A Web Service for Pre-segmenting Very Long Transcribed Speech Recordings

Nina Poerner*, **Florian Schiel**†
*Center for Information and Language Processing, University of Munich, Germany
†Bavarian Archive for Speech Signals, University of Munich, Germany
poerner@cis.uni-muenchen.de, schiel@bas.uni-muenchen.de

Abstract

The run time of classical text-to-speech alignment algorithms tends to grow quadratically with the length of the input. This makes it difficult to apply them to very long speech recordings. In this paper, we describe and evaluate two algorithms that pre-segment long recordings into manageable "chunks". The first algorithm is fast but cannot guarantee short chunks on noisy recordings or erroneous transcriptions. The second algorithm reliably delivers short chunks but is less effective in terms of run time and chunk boundary accuracy. We show that both algorithms reduce the run time of the MAUS speech segmentation system to under real-time, even on recordings that could not previously be processed. Evaluation on real-world recordings in three different languages shows that the majority of chunk boundaries obtained with the proposed methods deviate less than 100 ms from a ground truth segmentation. On a separate German studio quality recording, MAUS word segmentation accuracy was slightly improved by both algorithms. The chunking service is freely accessible via a web API in the CLARIN infrastructure, and currently supports 33 languages and dialects.

Keywords: text-to-speech alignment, speech segmentation, speech processing, MAUS

1. Introduction

Text-to-speech alignment plays an important role in Phonetic research, speech technology, and in the production of video subtitles. The Bavarian Archive for Speech Signals offers a free web service, Munich AUtomatic Segmentation (MAUS) (Schiel, 1999), to the scientific community. MAUS segments transcribed speech recordings into words and phones. Its main advantage is that, for many of its 33 languages and dialects, it cannot only perform forced alignment, but also model variable pronunciation. MAUS has recently been evaluated as the best aligner, out of ten candidates, on a recording by J.F. Kennedy (Oesch and Sidler, 2017).

One major disadvantage of MAUS is its run time on long recordings. This is because MAUS performs optimal Viterbi alignment, which, while more accurate than beam search, has quadratic run time with signal duration and transcription length (see Figure 1). Long run times mainly affect users who wish to use MAUS on certain types of real-world recordings, such as public speeches, interviews or audio books, as these tend to be longer than audio material produced for scientific research.

One way to combat MAUS's run time problem is to pre-segment long recording-transcription pairs into short "chunks", i.e., slices of matching audio and text material of a few seconds or minutes. Recently, we introduced the "chunker", an algorithm that does this job automatically (Poerner and Schiel, 2016). The chunker has since been made available as a web service by the Bavarian Archive for Speech Signals (Kisler et al., 2017).

In the following, we give an overview of the algorithm, including a number of new developments (Section 2.). Sections 3. and 4. describe an evaluation on three real-world educational and political recordings and a studio quality recording.

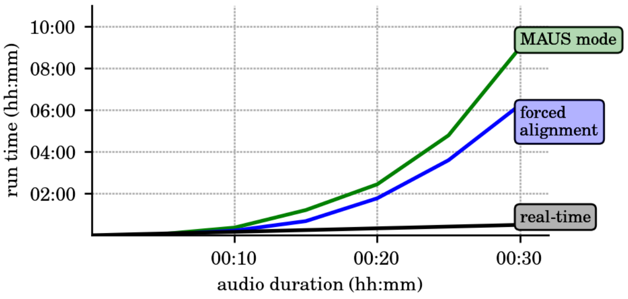

Figure 1: Run time of MAUS text-to-speech alignment with pronunciation modeling ("MAUS mode") and with simple forced alignment. Inputs are subsequences of the material used in Section 3.1.4.

2. Algorithms

The chunker offers two different technical solutions to the chunking problem: the so-called "standard" algorithm (default) and the "forced" algorithm.

2.1. Standard chunking algorithm

The standard chunking algorithm is a combination of the T (token-based) and P (phoneme-based) chunking algorithms presented in Poerner and Schiel (2016). The T algorithm builds on work by Moreno et al. (1998). It can be divided into the following steps:

Language model – A smoothed HTK (Young et al., 2006) bi-gram language model is trained on the transcription.

Recognition – The HTK-HVite speech recognition engine recognizes the signal, using the bi-gram model as well as language-specific acoustic models from MAUS.

Alignment – The recognized text is symbolically aligned with the transcription using the Hirschberg algorithm

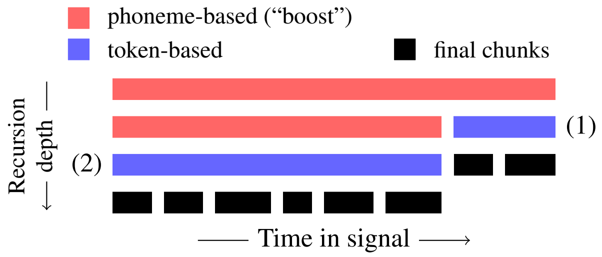

Figure 2: Schematic representation of standard algorithm with boost phase. (1) Switch from P to T chunker because chunk is short enough. (2) Switch from P to T chunker because P failed to find more boundaries.

(Hirschberg, 1975). Alternatively, users can opt for a faster, approximate algorithm with linear run time, similar to one proposed by Anguera et al. (2011).

Boundary selection – After symbolic alignment, the chunker searches for “anchor” regions where the transcription and the recognition result are perfectly aligned for a minimum number of words (e.g., 3). The signal and transcription are then split at a word boundary inside this region. The chunker prioritizes inter-word pauses.

Recursion – Unless all chunks are short enough, or no new boundaries were found, the entire process is repeated recursively on the discovered chunks.

Speech recognition run time grows with the duration of the recording and with the number of unique word types in the transcription. Hence, the run time of the T chunker – while faster than MAUS – was found to be unacceptable on recordings that exceed one hour.

The P chunker algorithm, which is based on work by Bordel et al. (2012), alleviates this problem. Its recognition engine and symbolic alignment run on the phoneme level instead of the word level. Since phoneme inventories are limited, recognition can thus be done in linear time. On the downside, the P chunker finds fewer boundaries and is less successful at locating inter-word pauses (Poerner and Schiel, 2016).

The “standard” algorithm combines the speed of the P chunker with the accuracy of the T chunker. More specifically, the P chunker is used in a so-called “boost phase” to break the signal into relatively long chunks (duration > 1 minute, red bars in Figure 2). After the boost phase, the T chunker does the fine-grained chunking (blue bars in Figure 2). In cases where the P chunker fails to find a sufficient number of boundaries, the T chunker is called as a back-up.

2.2. Forced chunking algorithm

There are cases where the standard algorithm fails to find a sufficient number of anchor regions, resulting in chunks that are too long for further processing. While we have found that this often indicates bad signal and/or transcription quality (which the user may want to deal with anyway), we decided to develop a second algorithm that guarantees a chunking result even in problematic cases.

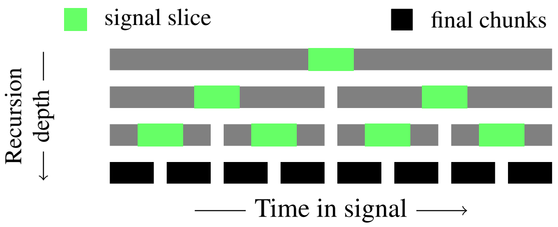

Figure 3: Schematic representation of the forced algorithm.

The forced chunking algorithm builds on a method of Moreno and Alberti (2009). They describe a factor automaton language model that is able to find any substring of a given text in the signal. We use this strategy in a divide-and-conquer way (see Figure 3):

Language model – A factor automaton language model is built on the transcription.

Recognition – A slice (e.g., 2 minutes) is cut from the middle of the signal. It is recognized by HTK-HVite using the factor automaton model. Since recognition with a factor automaton produces an alignment as a side-effect, no symbolic alignment is necessary.

Boundary selection – A word boundary from the recognized slice is used as a chunk boundary. As with the standard algorithm, inter-word pauses are prioritized.

Recursion – The process is recursively repeated on the resulting halves of the signal.

The size of the factor automaton grows linearly with the length of the transcription, and the necessary recursion depth grows logarithmically with audio duration. Hence, the forced algorithm tends to be slower than the (boosted) standard algorithm.

3. Evaluation

3.1. Data

We tested the chunker system on three real-world recordings of German, Italian and American English, whose durations range between 58 and 89 minutes, as well as on a 55 minute “pseudo recording” made up of concatenated studio quality audio clips (see Table 1 for details). MAUS on its own failed to produce a segmentation for any of these recordings within 48 hours (at which point the processes had to be interrupted).

3.1.1. Lecture recording (German)

The German real-world recording is a university lecture on Aristotelian philosophy from the euroWiss project (Heller et al., 2013)[1]. The lecture is held by a male native German speaker, and there are short contributions by students. Audio quality is high for the main speaker, but other speakers are difficult to understand as they speak far away from the microphone. There are frequent instances of background noises and reverberations. The signal was downloaded as a

[1] hdl.handle.net/11022/0000-0001-7DBA-2

Recording	Signal	Transcription	Standard algorithm		Forced algorithm	
	duration	# words	Chunker run time	MAUS run time	Chunker run time	MAUS run time
Lecture	88:27	11969	41:44	23:60	165:59	24:10
Seminar	73:47	9046	23:55	15:18	100:00	18:30
Address	58:29	7562	23:10	12:11	60:01	13:33
Concat	55:08	8231	25:38	16:21	76:00	16:29

Table 1: Signal duration, number of words, chunker run time and subsequent MAUS run time in minutes

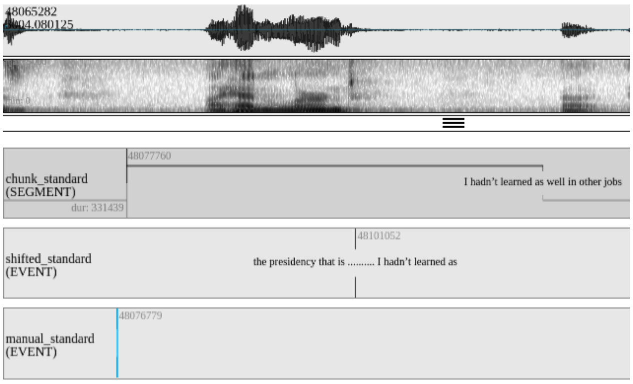

Figure 4: Manually annotated chunk boundary. Annotation levels from top to bottom: chunker prediction (invisible during annotation), randomly shifted prediction (visible during annotation), manually annotated boundary.

stereo MP3 file and converted into a mono 16 kHz WAVE file. The orthographic transcription was extracted from the accompanying EXMARaLDA annotation. The transcription is narrow and includes hesitations and false starts. We deliberately did not use the chunk segmentation provided.

3.1.2. Seminar recording (Italian)

The Italian real-world recording also stems from the euroWiss project. It contains a seminar on the author Walter Benjamin, held by a female native Italian speaker, again with short contributions by students. Remarks made on the audio and transcription quality of the German recording apply to the Italian recording too. Note that while the audio contains a transcribed and an untranscribed part, we only used the transcribed part (the first 73:47 minutes).

3.1.3. Address (American English)

The American English real-world recording is an address given by former US president Barack Obama (Obama, 2015)[2]. Most of the material is spoken by the main speaker, but there are short contributions by other speakers, including many non-native speakers. Audio quality is high for all speakers, but there are background noises and reverberations. The recording was downloaded as a mono MP3 file and converted into a mono 16 kHz WAVE file. The transcription was crawled from the corresponding web site. It is relatively broad, meaning that it does not contain hesitations and false starts. Furthermore, a number of sentences in the audio differ or are missing from the transcription.

3.1.4. Concatenated recording (German)

The concatenated recording is made up of 401 clips from the German Verbmobil corpus (Burger et al., 2000)[3]. The clips are studio quality, 16 kHz mono WAVE. They contain speaker turns from timetabling role-plays (6 male, 16 female speakers), and they come with a manual broad phonetic segmentation. To mimic a long recording, the clips were concatenated into a single 55 minute file.

3.2. Processing

All recording-transcription pairs were converted into EMU databases (Winkelmann et al., 2017)[4]. We used the G2P[5] grapheme-to-phoneme converter (Reichel, 2012) to produce a phonemic (SAMPA) representation of every word in the transcription. In the next step, we ran the chunker in two different settings:

standard – Boost phase with anchor length 4 phonemes, followed by word-based phase with anchor length 3 words. Minimum chunk duration 5 seconds, alignment by Hirschberg algorithm.

forced – Forced algorithm with slice duration 2 minutes. Minimum chunk duration 5 seconds.

Finally, we used MAUS to produce a phonetic segmentation based on the outcomes of the two chunker variants. To compare the accuracy of MAUS with and without pre-chunking, we also ran it directly on the concatenated recording. Since MAUS on its own could not process the full 55 minutes in a reasonable amount of time, the recording was manually pre-segmented into ten minute chunks, effectively giving MAUS a head start over the chunker.

3.3. Ground truth

3.3.1. Word boundaries

In the German concatenated recording, the existing manual phonetic annotation was used to locate the ground truth start and end times of words. In the three real-world recordings, no such annotation exists; hence, word segmentation accuracy could not be evaluated on them.

3.3.2. Chunk boundaries

A predicted chunk boundary has the form (w_1, t, w_2), where w_1 and w_2 are the word tokens to the left and right of the boundary, and t is the predicted time. Its true time is

[2] www.americanrhetoric.com/speeches/barackobama/barackobamaYSEALIWashingtonDC.htm

[3] hdl.handle.net/11022/1009-0000-0000-EB31-0, sessions m112d-m117d, m119d, m222d, m224d, m230d, m231d

[4] ips-lmu.github.io/EMU.html

[5] All tools were called using EMU's interface to the BAS web services (Poerner and Winkelmann, 2017). Hence, run times reported in Table 1 contain a slight overhead for signal uploads.

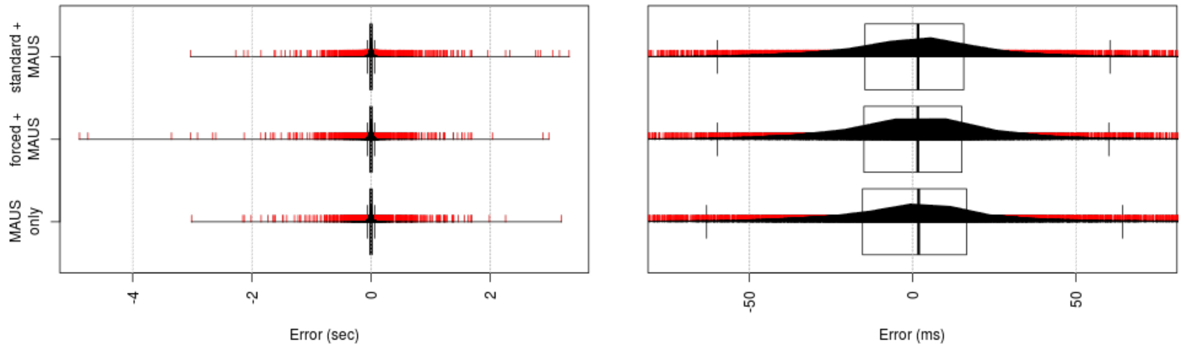

Figure 5: MAUS word segmentation errors relative to manual annotation, with and without pre-chunking, on concatenated German recording. Box plots with error distributions. Red dashes: Single data points. Full range (left) and zoomed view (right).

the mid-point between the true end time of w_1 and the true start time of w_2. If there is no pause between w_1 and w_2, end time, start time and mid-point coincide.

In the concatenated recording, true chunk boundaries were derived from the existing manual annotation. In the real-world recordings, true boundaries were annotated by the first author using the EMU-webApp (Winkelmann and Raess, 2014). To avoid confirmation bias, predicted boundaries were randomly shifted to the left or right by up to two seconds before annotation, while the actual predictions were hidden and later restored (see Figure 4).

3.4. Results

Chunk boundary errors are defined as the difference between predicted chunk boundary times and true chunk boundary times (see Section 3.3.2.). Figure 6 shows distributions of chunk boundary errors on all recordings. Word segmentation errors are defined as the difference between word start times according to MAUS and word start times according to the manual phonetic segmentation. Figure 5 shows distributions of word segmentation errors for the concatenated recording. Table 2 lists the percentages of chunk boundaries that were correctly placed inside inter-word pauses of duration 100 ms or more. Figure 8 shows distributions of chunk durations on all recordings.

4. Discussion

4.1. Run time

On all recordings, both chunking algorithms enabled MAUS to run in less than real-time (see Table 1). The standard chunking algorithm ran in under real-time too, while the forced algorithm took up to four times as long. Still, both methods reduced overall run time compared to MAUS-only segmentation, which took more than 48 hours.

4.2. Effects on word segmentation accuracy

Figure 5 suggests that MAUS word segmentation accuracy was not harmed by pre-chunking. Quite the contrary, segmentation accuracy slightly improved relative to the baseline. It is worth keeping in mind that word segmentation accuracy could only be evaluated on the studio quality recording, which might not be representative of real-world recordings.

	Lecture	Seminar	Address	Concat
Standard	58.4%	52.6%	51.5%	69.5%
Forced	75.6%	75.1%	65.1%	79.0%

Table 2: Percentage of chunk boundaries that were placed inside a pause of 100 ms or more between their previous and next word.

4.3. Finding pauses

As mentioned in Section 2.1., the chunker aims to place boundaries inside inter-word pauses to reduce the risk of cutting into the previous or following word. Table 2 shows that the forced algorithm was more successful at finding inter-word pauses than the standard algorithm, which is probably due to its greater choice of potential boundary locations.

4.4. Chunk durations

The standard algorithm produced chunks with a duration of up to 1:47 minutes (see Figure 8). While this is sufficiently short for fast MAUS processing, it suggests that there were regions where the standard method failed to find any boundaries. Manual inspection suggests that these regions often coincided with low-quality stretches in the recording (e.g., prolonged background noise, turns by speakers far away from the microphone, regions with audio-transcription mismatches).

The forced algorithm, on the other hand, returned shorter chunks, because it is forced to cut inside low-confidence regions.

4.5. Chunk boundary errors

The majority of absolute chunk boundary errors made by the standard algorithm were below 100 ms (see Figure 6). We revisited all absolute errors beyond 500 ms and found that the standard algorithm has a tendency to get confused whenever a speaker utters the same (or approximately the same) phrase several times in a row (e.g., *Ja,* ***wir wollen die*** *Nahrung,* ***wir wollen die*** *äh* ***wir wollen die*** *Ernährung*, German lecture). A possible explanation is that the recognition engine correctly recognizes one repetition but not the others, resulting in a situation where the recognized phrase gets aligned to the wrong instance in the transcription.

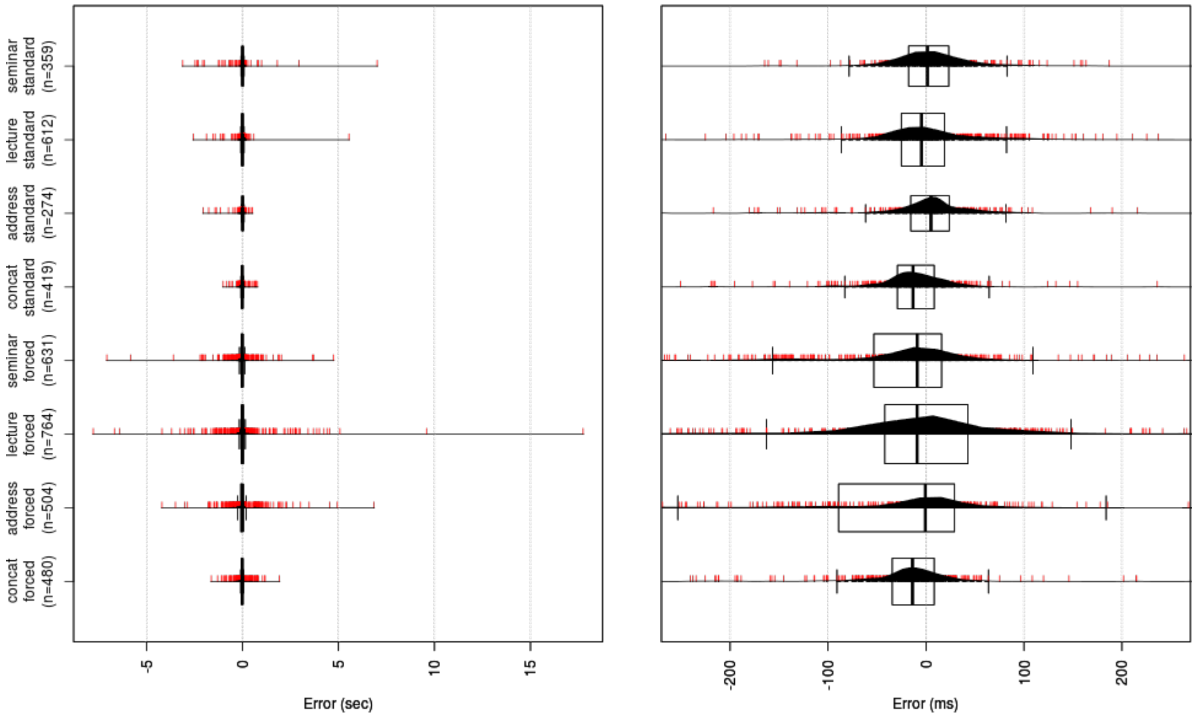

Figure 6: Chunk boundary errors relative to manual annotation. Box plots with error distributions. Red dashes: Single data points. Full range (left) and zoomed view (right).

While the forced algorithm guarantees short chunks, it turned out to be more prone to errors than the standard algorithm, with a greater range of error outliers and broader error distributions. Manual analysis of these outliers revealed that most of them come down to one of two patterns:

- The chunker misinterprets the above-mentioned low-quality regions, e.g., by "imagining" a subsequence of the transcription inside a long noisy pause. This kind of mistake can lead to major mis-alignments, especially if it happens early in the recursion.

- In a scenario where a pause is preceded or followed by a short or clipped word (e.g., *I* or *of*), the forced chunker sometimes locates that word on the wrong side of the pause (see Figure 7). The difference between predicted and true chunk boundary is half the duration of the pause, which may be several seconds. However, the error is less severe than its magnitude suggests, since all words except one end up in the correct chunk.

5. Summary

We have presented two algorithms that automatically break long transcribed speech recordings into "chunks", i.e., shorter text-audio pairs. They are intended as a preprocessing tool to the Viterbi-based phonetic segmentation system MAUS, which cannot cope with very long recordings on its own. The combination of chunker and MAUS allows researchers to align and segment very long speech recordings (e.g., interviews, speeches or audio books) quickly and without the need for time-consuming manual pre-segmentation.

The standard chunking algorithm applies a combination of fast phoneme-based and accurate word-based recognition. The forced chunking algorithm is slower and less accurate, but it guarantees a result even in the face of poor signal or transcription quality. Hence, it can be used as a fall-back in extreme cases where the standard algorithm fails to deliver chunks that are short enough.

We evaluated both algorithms on long recordings in three different languages. The comparison with a manual annotation showed that most chunk boundaries deviate from the ground truth by less than 100 ms. MAUS word segmentation accuracy does not seem to be negatively affected by pre-chunking; instead, there is a slightly positive impact.

Both algorithms are offered as free-to-use web services within the CLARIN infrastructure.[6]

[6] http://hdl.handle.net/11022/1009-0000-0001-232C-7

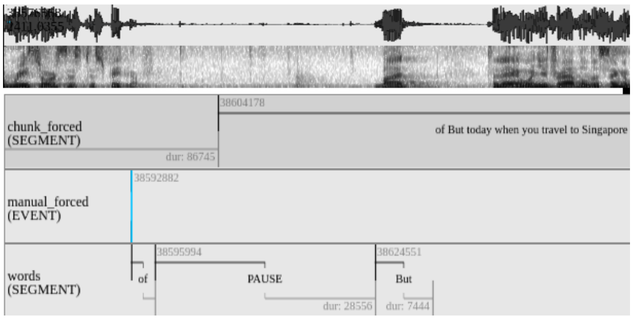

Figure 7: Forced algorithm locates short word on wrong side of a pause. Annotation levels from top to bottom: chunker prediction, manual boundary, manual locations of the words *of* and *But*, and their inter-word pause.

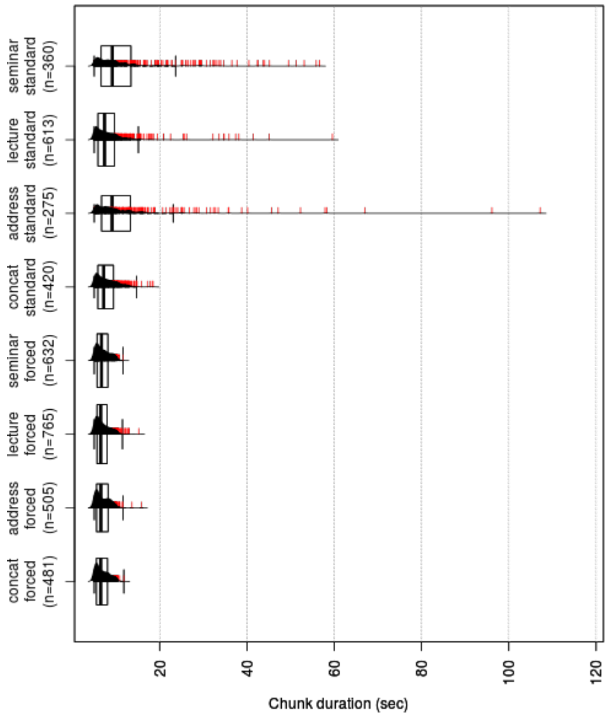

Figure 8: Chunk durations. Box plots with distributions. Red dashes: Single data points.

6. Bibliographical References

Anguera, X., Perez, N., Urruela, A., and Oliver, N. (2011). Automatic synchronization of electronic and audio books via TTS alignment and silence filtering. In *Proc. ICME*, Barcelona, Spain, July.

Bordel, G., Peñagarikano, M., Rodríguez-Fuentes, L. J., and Varona, A. (2012). A simple and efficient method to align very long speech signals to acoustically imperfect transcriptions. In *Proc. Interspeech*, pages 1840–1843, Portland, USA, September.

Hirschberg, D. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343.

Kisler, T., Reichel, U., and Schiel, F. (2017). Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326–347.

Moreno, P. J. and Alberti, C. (2009). A factor automaton approach for the forced alignment of long speech recordings. In *Proc. ICASSP*, pages 4869–4872, Taipei, Taiwan, April.

Moreno, P. J., Joerg, C. F., Van Thong, J.-M., and Glickman, O. (1998). A recursive algorithm for the forced alignment of very long audio segments. In *Proc. ICSLP*, pages 2711–2714, Sydney, Australia, December.

Oesch, J. and Sidler, A. (2017). Videx: Indexing videos for the "official bulletin" of the swiss federal parliament. *Haute école d'ingénierie et de gestion du canton de Vaud*, www.jonasoesch.ch/portfolio/videx.

Poerner, N. and Schiel, F. (2016). An automatic chunk segmentation tool for long transcribed speech recordings. In *Proc. Phonetik und Phonologie*, pages 144–146, Munich, Germany, October.

Poerner, N. and Winkelmann, R. (2017). Interfacing the BAS speech science web services and the EMU speech database management system. In *Proc. Phonetik und Phonologie*, Berlin, Germany, September.

Reichel, U. (2012). PermA and Balloon: Tools for string alignment and text processing. In *Proc. Interspeech*, pages 1874–1877, Portland, USA, September.

Schiel, F. (1999). Automatic phonetic transcription of non-prompted speech. In *Proc. ICPhS*, pages 607–610, San Francisco, USA, August.

Winkelmann, R. and Raess, G. (2014). Introducing a web application for labeling, visualizing speech and correcting derived speech signals. In *Proc. LREC*, pages 4129–4133, Reykjavik, Iceland, May.

Winkelmann, R., Harrington, J., and Jänsch, K. (2017). EMU-SDMS: Advanced speech database management and analysis in R. *Computer Speech & Language*, 45:392–410.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., and Povey, D. (2006). *The HTK Book*. Cambridge University.

7. Language Resource References

Burger, Susanne and Weilhammer, Karl and Schiel, Florian and Tillmann, Hans G. (2000). *Verbmobil data collection and annotation*. Springer. hdl.handle.net/11022/1009-0000-0000-EB31-0.

Heller, Dorothee and Hornung, Antonie and Redder, Angelika and Thielmann, Winfried. (2013). *euroWiss: linguistic profiling of European academic education*. Hamburger Zentrum für Sprachkorpora. hdl.handle.net/11022/0000-0001-7DBA-2.

Obama, Barack. (2015). *Southeast Asian Youth Initiative Fellows Address*. American Rhetoric. www.americanrhetoric.com.