DOI: 10.61782/fa.2025.0775

FORUM ACUSTICUM
EURONOISE 2025

SLICER – A TOOL FOR EFFICIENT STIMULI EXTRACTION FROM LARGE SPEECH CORPORA

Lucas Eckert[1] **Saskia Wepner**[1] **Barbara Schuppler**[1*]

[1] Signal Processing and Speech Communication Laboratory, Graz University of Technology, Austria

ABSTRACT

This paper introduces a tool for searching and extracting stimuli from speech corpora, allowing to manipulate both the audio and the annotation file simultaneously. SLICER is designed to complement existing software like Praat by offering additional functionalities, which include: 1) An advanced label search, allowing users to locate specific segments based on annotations. 2) Slices can subsequently be manipulated. 3) These slices can further be filed and exported to create a set of stimuli. When manipulating segments (e.g., phones, words), SLICER offers the possibility to insert noise with configurable signal-to-noise ratios, and apply smooth attack and decay transitions to ensure natural-sounding stimuli. 4) When exporting the set of stimuli, users can (a) choose which annotation levels to include, (b) set audio sample rates and formats, and (c) normalize the audio output for consistency between the stimuli. 5) The integrated file-naming conventions allow for locating the stimulus in the original corpus file. As an example from our own work, we used SLICER to extract disfluent utterances from a corpus of spontaneous conversations, which were manipulated (i.e., by deleting or substituting filler particles), normalized, and subsequently used in a perception experiment.

Keywords: *speech, textgrid manipulation, experimental design, corpus annotation tools*

**Corresponding author*: b.schuppler@tugraz.at.

1. INTRODUCTION

When preparing perception experiments, the selection and preparation of stimuli is a time-consuming process. Particularly, in corpus-based studies, where stimuli are extracted from longer stretches of speech with corresponding annotations, it includes many processing steps that have to be repeated for every stimulus. Ensuring that the resulting stimuli sound natural may require additional manipulation of the audio signals, such that the listeners in the experiment will not be influenced by artefacts that stem from manipulating the signal. Within a stimulus, they can not be cut out simultaneously from both the audio and the annotation file when using established annotation tools. Instead, using separate programmes for audio and annotations with multiple manipulations, the parts before and after the artefact need to be exported separately (for audio and annotation) and later joined again. Further, it may be necessary to extract a large number of files with already existing annotations (e.g., word and phone-level segmentation, prosody).

Established tools for corpus annotation are ELAN [1] and Praat [2]. While ELAN is particularly helpful for textual annotation of audio and video files, Praat offers versatile tools for speech analysis, such as labelling, segmenting and performing spectral analyses as well as speech synthesis and manipulations like filtering and pitch alteration. However, when it is necessary to extract multiple stimuli with corresponding annotations from long files, such as large (spontaneous) speech corpora, additional functionalities are required, which we describe in detail in Sec. 1.1.

This paper presents SLICER, a tool that incorporates these functionalities and provides an intuitive workflow for large-scale stimulus extraction and manipulation. We

do NOT try to replace other tools, instead, we suggest SLICER for complementing existing tools by suggesting a workflow to support working with established software. SLICER makes it more time-efficient to produce a large number of stimuli for listening and transcription experiments, while also allowing for controlling the acoustic quality of the stimuli as well as their comparability within a set of stimuli. We recommend creating all necessary annotations before stimulus extraction using ELAN or Praat and afterwards feeding them into SLICER. The file formats used by SLICER are fully compatible with both Praat and ELAN.

1.1 Why use SLICER?

Terminology. We refer to a *filed* (listed, but not exported) or displayed time frame as a *slice* when worked on in SLICER and a *stimulus* after export. A segment that is either silenced, replaced with noise or cut out from the audio is referred to as *manipulation segment*.
The key features of SLICER include:

1. Advanced label search using regular expressions, allowing the search of multiple labels at once.
2. Filing and naming of slices for later editing, allowing fast scanning of long files and accumulating of stimulus candidates.
3. Bulk export of multiple stimuli with consistent settings.
4. Selection of annotation-tiers to export (e.g., changing multiple speaker annotation files to single speaker files.)
5. Simultaneous manipulation of annotation and audio for any number of segments:
    - Removing manipulation segments and optionally replacing them by noise of chosen length and SNR (e.g., the noise floor of the surrounding recording).
    - Fading in and out at the start and end of slices and manipulation segments (e.g., to avoid audible artefacts from cutting).
    - Automatic mapping of time stamps in the annotations of manipulated slices.
6. Workspace saving for later loading and editing.
7. Playback and export of normalized audio (e.g., to ensure stimuli are played with similar level in the experiment).

Some of this functionality is also included in Praat, ELAN and DAWs like REAPER [3]. Their combination in SLICER makes it possible to efficiently work on large speech corpora while keeping a good overview, editing multiple stimuli with the same properties and exporting them with automatic consistent naming.

2. WORKFLOW

In our own work, we so far presented SLICER at a practical session at the "3rd Graz-Vienna Speechworkshop. Connecting with Health Sciences" [4]. We further used SLICER to prepare stimuli for a transcription experiment to investigate the effect of filler particles (FP) on human and automatic speech recognition [5]. For this purpose, we extracted stimuli from a large corpus of conversational speech containing one-hour-long conversations between two speakers each (i.e., GRASS corpus [6]). From utterances that originally all contained an FP, we used SLICER to create a set of stimuli with and without the FP. Here, we show an exemplary workflow from start to export for creating the stimuli. Fig. 1 shows a schematic representation of the stimuli that we describe here. Fig. 2 shows the graphical user interface (GUI) of SLICER.

2.1 Label Search and Stimuli Selection

First, we imported the audio and annotation files via the file menu (brown pane in Fig. 2. After loading audio and annotation, the available annotation tiers were listed and we selected those tiers needed for our experiment (blue pane), i.e., the word level annotation of the first speaker. We searched through the annotations of the whole one-hour long conversation using the label search (red pane) with regular expressions. To find FPs in our data, we searched for `^(a|ä)hm?$`. The label search box then showed how many instances were found in the active tiers. We could also change start and end of a slice shown to us (yellow pane) and listen to the examples (upper left in pink pane). The main tool for setting time stamps at the right time is the SLICER figure window (pink pane). There, the audio and the selected tiers are shown in separate subplots. Using the navigation bar (lower left in pink pane), we zoomed in to set boundaries precisely and also save the figure. A slice was then filed in *Selected slices* (green pane) and named (e.g., to mark utterances of different kinds) [1].

[1] Note that filing time frames in *Selected time frames* does neither export nor save them.

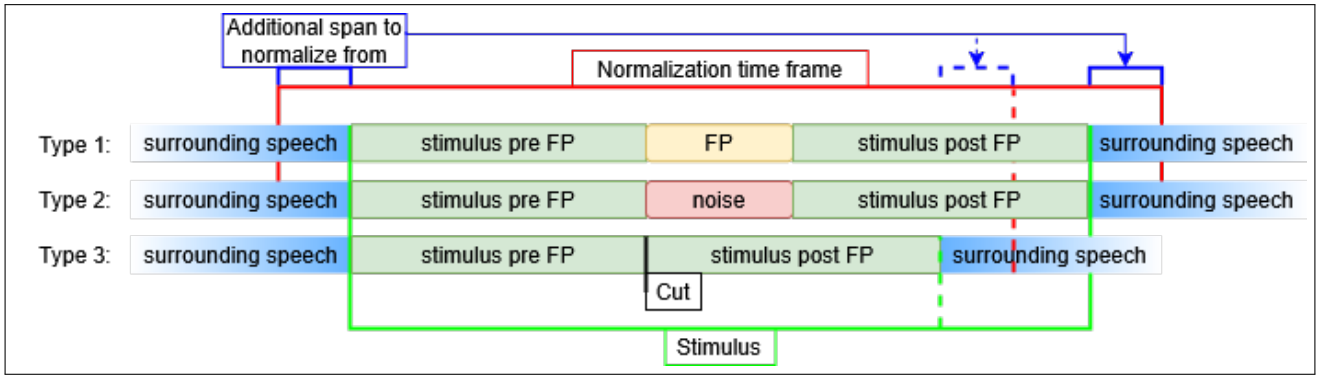

Figure 1. Schematic representation of different manipulations. From one utterance, we created three stimuli: 1) the original utterance, including the filler particle (FP), 2) FP replaced by noise and 3) FP removed. The green frame shows the resulting stimulus for each case. The blue frame shows the time span around the slice used for normalizing the audio.

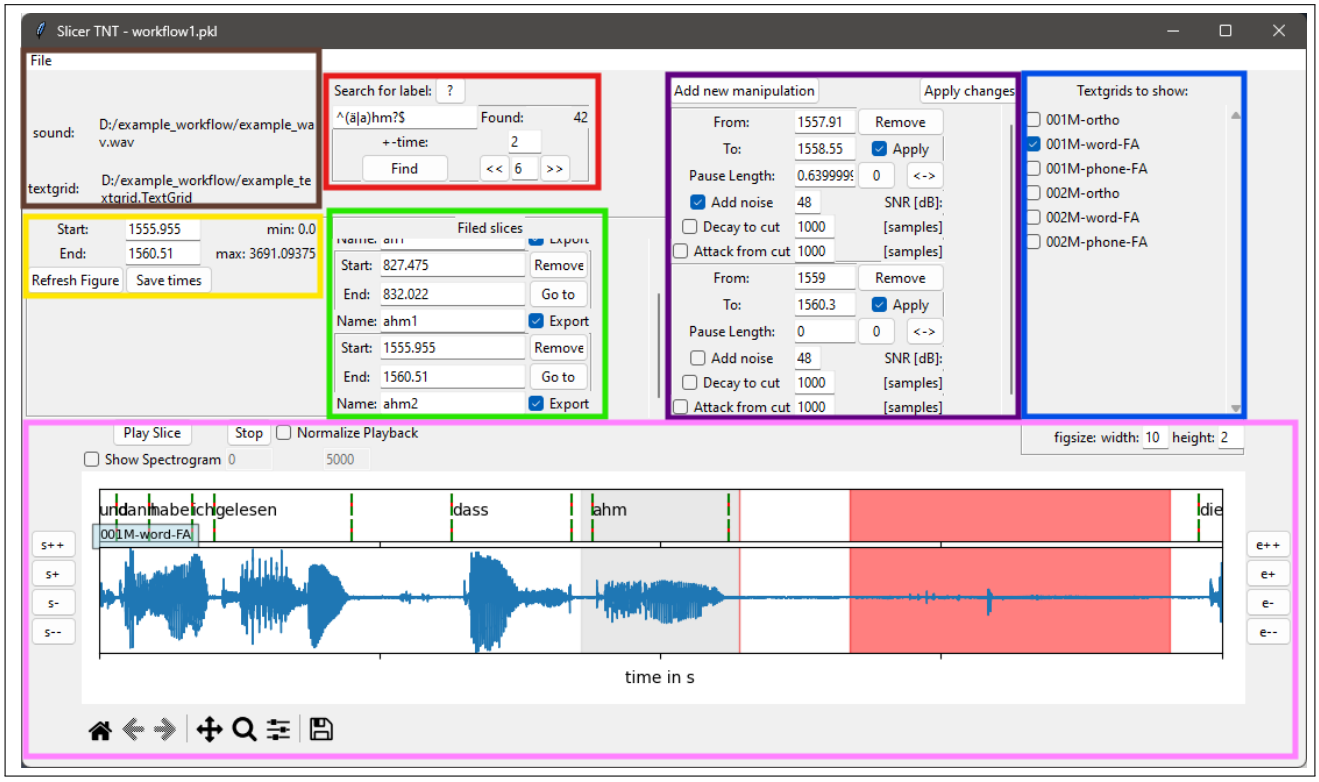

Figure 2. GUI (main window) of SLICER. The coloured frames are referenced throughout Sec. 2, where we describe the exemplary workflow.

2.2 Audio and Annotation Manipulation

Labels occurring within the borders of the manipulation segment can either be deleted or substituted. In our scenario, the same stimuli had to be produced three times, with the FP and without the FP, once replaced by noise,

once cut completely. Upon clicking *Add new manipulation*, we added a new entry to the manipulations list (purple box in Fig. 2): the first manipulation replaced the *ahm* by noise at 48 dB SNR. This is shown as a gray area in the SLICER figure window (pink box). The second manipulation was a complete cut of an artefact from both audio and annotation, so the start and end timestamps of successive events were shifted by the length of the manipulation segment.

Manipulating pause lengths. The pause length can be a minimum of 0 for complete cuts and a maximum of the full length of the manipulation segment. Everything in between is indicated by in gray (for the part that will be replaced by noise of silence) and in red (for the part that will be cut from the signal).

Softening the edges. If segmentation is challenging, e.g., because of overlapping sounds, a *Decay to cut* and an *Attack from cut* can be applied. These fades are not necessarily calculated towards complete silence but instead towards the chosen noise floor, which we selected at 48 dB SNR for the first manipulation segment. For playback or export of the example without the manipulation, it is possible to uncheck the *Apply* box for each manipulation. Doing so, we can compare the stimulus with and without the manipulation quickly without having to redo the manipulation.

2.3 Stimulus Export

When all manipulations were done, we exported the stimuli in the file menu (brown box in Fig. 2). Therefore, we checked all tiers that should be exported into the resulting annotation files. We chose to normalize the audio in a defined normalization time frame as explained in Fig. 1. The additional time span is especially useful for short slices, where normalization on the slice only would be applied to a very short time span of more similar amplitude, resulting in a higher average amplitude.

Choosing audio parameters. It is possible to change the values for bit depth and sample rate for different purposes, such as reducing the sample rate to 16kHz, as typically used in automatic speech recognition.

Bulk export. For quickly exporting multiple stimuli at once, SLICER provides the option to *export all checked slices*. Fade in and out can be applied here similarly to the manipulations. But this fade will not tend towards a noise floor now, and go to 0 instead. This will then export every filed slice where *Export* is checked, each using the specified start and end times. The name of the resulting files is given according to the start time and the name that is specified for the slice in the list of filed slices.

3. CONCLUSION

This paper presented SLICER and an example workflow introducing how it can be used for extracting and editing multiple stimuli for perception experiments from long audio and annotation files. We plan to continue incorporating additional features to SLICER, for instance, extending pauses that are longer than the manipulation segments. SLICER is available for your research at `https://github.com/SPSC-TUGraz/SLICER`, including a detailed documentation of all mentioned functionalities.

4. ACKNOWLEDGEMENTS

This research was funded by the Austrian Science Fund (FWF) [10.55776/P32700].

5. REFERENCES

[1] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "ELAN: a professional framework for multimodality research," in *Proceedings of LREC*, pp. 1556 – 1559, 2006.

[2] P. Boersma and D. Weenink, "Praat: doing phonetics by computer." `https://www.fon.hum.uva.nl/praat/`, 2025. V. 6.4, retrieved 2025-03-03.

[3] G. Francis, "The essential guide to recording, editing and mixing with REAPER," 2025.

[4] L. Eckert, "How to use SLICER for stimuli extraction from large speech corpora," in *Practical session at "3rd Graz-Vienna Speechworkshop. Connecting with Health Sciences"*, 2025.

[5] S. Wepner, *(When) Does it Harm to Be Incomplete? — Comparing Human and Automatic Speech Recognition of Syntactically Disfluent Utterances*. PhD thesis, Graz University of Technology, 2025.

[6] B. Schuppler, M. Hagmüller, and A. Zahrer, "A corpus of read and conversational Austrian German," *Speech Communication*, vol. 94, pp. 62–74, 2017.