

Octra

Web-basierter Editor für die orthographische Transkription

Christoph Draxler

Julian Pömp

6. März 2018

1 Einleitung

Octra ist ein web-basierter Editor zum effizienten Erstellen zeitalignierter orthographischer Transkripte von Sprachaufnahmen [PD17]. Octra ordnet einer Audiodatei eine einzige Transkriptionsspur zu, die in der Regel ein orthographisches Transkript der Äußerung mit einigen wenigen Markersymbolen für bestimmte interessierende Phänomene enthält. Ergebnis einer Transkription ist eine Segmentliste, deren Elemente zeitaligniert sind und die den Transkriptionstext enthalten. Diese Segmentliste kann in verschiedenen Formaten exportiert werden.

Bei der Entwicklung von Octra wurde auf eine möglichst effiziente Bedienung geachtet. Der 2D-Editor stellt auch längere Signalabschnitte ohne Stauchung dar und erlaubt damit ein visuelles Setzen von Grenzen. Mit Tastaturkürzeln können die wichtigsten Editorfunktionen direkt erreicht werden. Octra unterstützt drei Modi: online-Modus zur automatischen Auswahl der zu transkribierenden Datei, lokaler Modus zur Verarbeitung lokaler Dateien, URL-Modus zur Einbettung in Verarbeitungspipelines.

Octra unterstützt nur eine Transkriptionsspur. Das bedeutet, dass Octra für die Annotation von Dialogen weniger gut geeignet ist.¹

2 Schnelleinstieg in Octra

- Octra in Google Chrome, Firefox oder Opera laden:
<https://phonetik.uni-muenchen.de/apps/octra/octra/login>
Der Octra-Startbildschirm besteht aus zwei Teilen: links für den Online- oder Crowdsourcing-Modus, rechts für den lokalen Modus (Abb. 1).
- Wählen Sie auf dem lokalen Rechner eine Audiodatei im `.wav`-Format aus, ziehen Sie sie auf das graue Feld in der rechten Hälfte des Octra-Fensters.

¹Natürlich kann man die Segmentgrenzen so setzen, dass sie einzelne Sprecherbeiträge enthalten, und dann in der Transkription ein passendes Kürzel setzen.

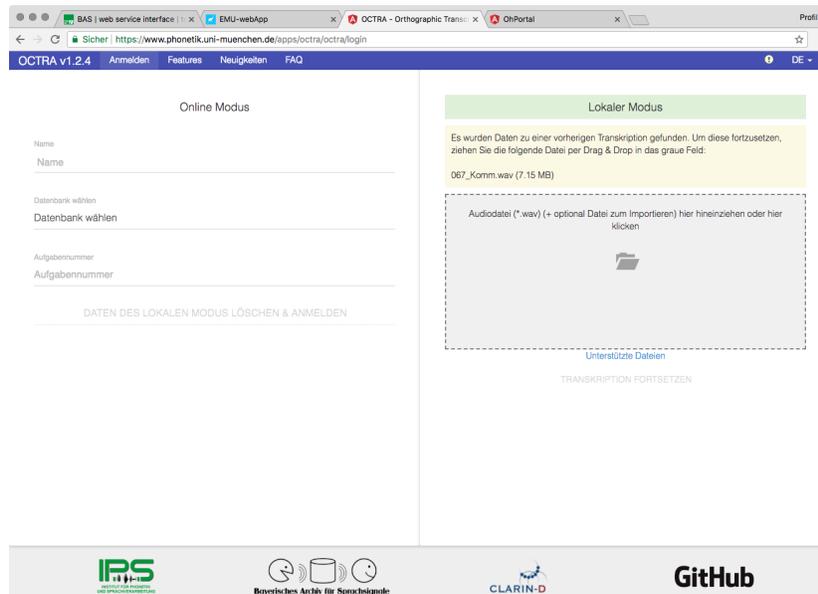


Abbildung 1: Octra Startbildschirm – Ablegen einer Audiodatei im .wav-Format auf dem grauen Feld öffnet Octra im lokalen Modus

Damit starten Sie die Transkription².

- Octra öffnet die Signaldatei im 2D-Editor (Abb. 2).
- Wählen Sie einen Signalabschnitt aus und drücken Sie die Tabulator-Taste zum Abspielen des ausgewählten Signalabschnitts.³
- Markieren Sie Signalabschnitte zur Transkription durch Setzen von Grenzen – am besten in Pausen. Bewegen Sie dazu den Cursor an die passende Stelle im Signal und drücken Sie die Taste 's'. Durch nochmaliges Drücken der Taste 's' können Sie die eben gesetzte Grenze wieder löschen.
- Mit der Taste 'a' markieren Sie ein Segment als leeres Segment – ein solches Segment muss nicht transkribiert werden, es enthält nur einen Pausenmarker.
- Bewegen Sie den Cursor zwischen zwei Segmentgrenzen und drücken Sie die Eingabetaste. Octra öffnet das Segment im Transkriptionseditor (Abb. 3).

²Wenn schon eine Audiodatei in Bearbeitung ist, dann warnt Octra, dass die Transkription dieser Datei noch nicht abgeschlossen ist und verworfen wird

³Alternativ setzen Sie in den Einstellungen (im blauen Balken unter dem Zahnrad-Symbol) die Optionen **Lupe anzeigen** und **Audiowiedergabe bei Mausbewegung** – dann wird der Signalabschnitt unter der Lupe abgespielt.

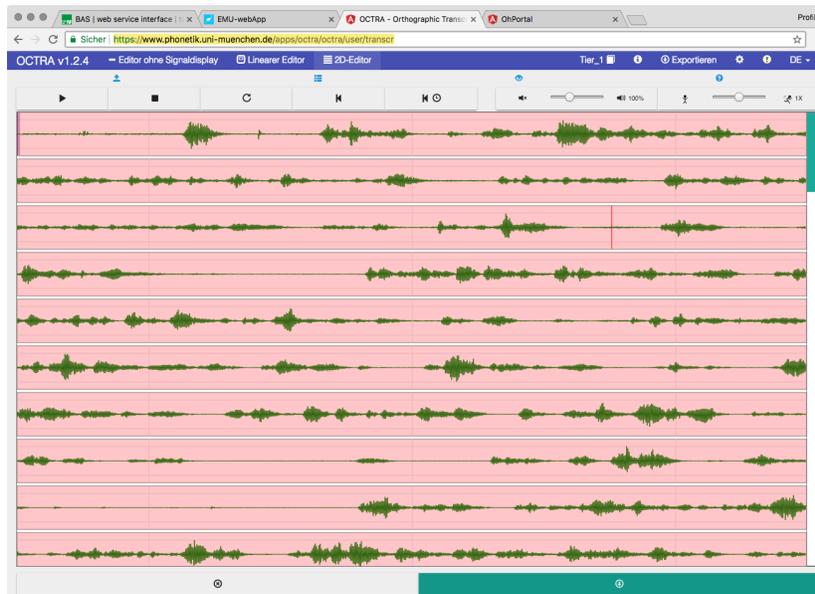


Abbildung 2: Octra 2D-Editor mit langem Sprachsignal

- Mit der Tabulatortaste starten und stoppen Sie die Audiowiedergabe. Mit der Kombination Umschalt- und Tabulatortaste startet die Audiowiedergabe ca. 0,5s vor der Cursorposition – damit kommt man leicht wieder an die richtige Stelle im Transkript. Das Abspielen kann auch durch Mausklicks auf die Abspielbuttons gesteuert werden.
- Wenn das Segment transkribiert ist gehen Sie mit der Tastenkombination 'ALT' und → zum nächsten Segment; das Transkript des zuletzt bearbeiteten Segments wird dabei automatisch gesichert.
- Sind alle Segmente transkribiert sichern Sie die Transkripte mit der Schaltfläche 'Daten exportieren'. Octra bietet Ihnen eine Reihe von Exportformaten für das Sichern der Transkription an (Abb. 4).

3 Weitere Optionen

Octra bietet eine Vielzahl weiterer Möglichkeiten:

- **weitere Editoren:** der *Editor ohne Signaldisplay* erlaubt die Steuerung des Audiosignals wie mit einem Diktiergerät, der *lineare Editor* stellt das gesamte Signal in einer Zeile dar.
- **Loggen:** Octra erlaubt ein Loggen der Editorfunktionen. Damit kann der Transkriptionsvorgang detailliert analysiert werden. Im lokalen Modus ist

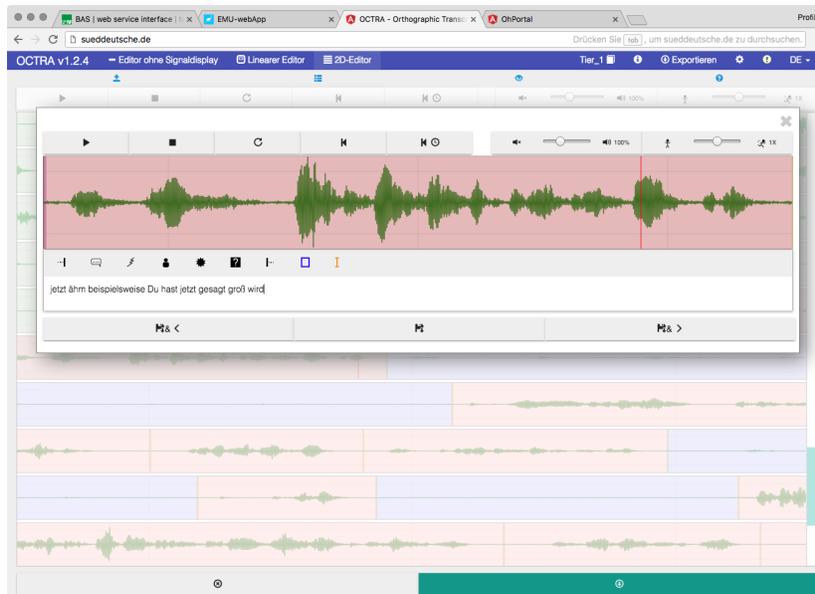


Abbildung 3: Octra Transkriptionseditor

das Loggen standardmäßig ausgeschaltet.

- **Crowdsourcing:** Octra kann als Editor in Crowdsourcing-Projekten verwendet werden. Dazu werden die zu transkribierenden Dateien in einer Datenbank auf dem Server abgelegt. Transkribierer melden sich bei Octra an, wählen das ihnen zugeordnete Projekt auf der linken Seite des Anmeldefensters aus. Octra lädt dann die nächste zu bearbeitende oder die ihnen zugewiesene Datei aus und öffnet sie im Editor. Nach Ende der Transkription wird das Transkript automatisch an den Server übermittelt und die nächste Datei geöffnet.⁴
- **Einstellungen:** mit der Einstellung *einfacher Modus* werden Beschriftungen der Symbole angezeigt, mit *Audioausgabe der Lupe* wird das Audiosignal im Lupenfenster abgespielt.
- **Segmenteliste:** durch einen Klick auf das Augensymbol (👁️) wird eine farbcodierte Liste aller Segmente angezeigt: grün für transkribierte, rot für noch nicht transkribierte Segmente, blau für Pausensegmente. Rote Schrift zeigt formale Fehler im Transkript an, z.B. das Verwenden von Satzzeichen.

⁴Wenn Sie selbst ein Crowdsourcing-Transkriptionsprojekt einrichten wollen schreiben Sie bitte an draxler@phonetik.uni-muenchen.de.

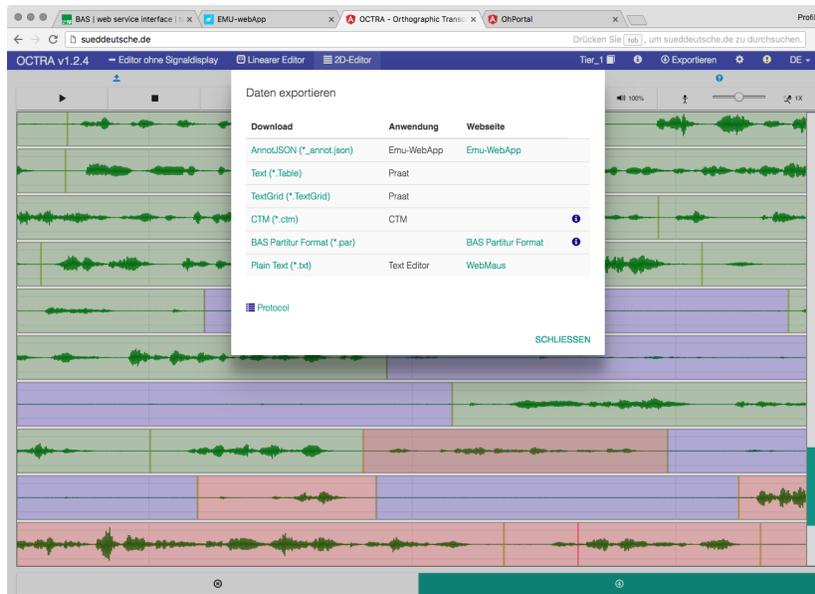


Abbildung 4: Export-Dateiformate in Octra

- **Hilfestellung:** Octra zeigt Hilfetexte zu den einzelnen Bedienelementen an, wenn man mit der Maus darüberfährt.

Bitte melden Sie Fehler und Verbesserungsvorschläge über den Fehlermeldungsbutton oder per Mail an octra@phonetik.uni-muenchen.de.

4 Zitieren

Wenn Sie Octra für Ihre Arbeiten benutzen, dann zitieren Sie bitte [PD17]. Vielen Dank!

```
@inproceedings{PoempDraxler_2017,
  Address = {Berlin},
  Author = {Julian Pömp and Christoph Draxler},
  Booktitle = {Proceedings Phonetik und Phonologie},
  Title = {{OCTRA} -- A configurable browser-based editor for orthographic transcription},
  Pages = {145-148},
  Year = {2017}}
```

5 Anhang

Dateiformate

Octra unterstützt verschiedene Dateiformate für den Im- und Export von Transkripten. Einige der Formate unterstützen Annotationen auf mehreren Ebenen,

z. B. AnnotJSON, TextGrid, usw. Werden Inhalte aus diesen Dateiformaten importiert fragt Octra, welche Ebene zur Transkription verwendet werden soll; der Inhalt dieser Ebene wird dann in Octra angezeigt und kann verändert werden. Beim Export werden alle vorhandenen Ebenen exportiert.

AnnotJSON

AnnotJSON ist ein JSON-Format in UTF-8 Kodierung, das in Emu Sprachdatenbanken verwendet wird. Es unterscheidet drei Arten von Segmenten: das **Item** ohne Samplepunkt, ein **Event** mit einem Samplepunkt und das **Segment** mit Anfangssamplepunkt und Dauer in Samplepunkten. Aus den Samplepunkten kann mithilfe der Samplerate ein Zeitpunkt berechnet werden.

In AnnotJSON sind Segmente zu Annotationsebenen zusammengefasst, und Links geben die Beziehungen zwischen Segmenten an. Diese Beziehungen können zwischen Elementen innerhalb einer Ebene bestehen, oder zwischen Elementen verschiedener benachbarter Ebenen.

AnnotJSON ist in [WHJ17] beschrieben.

```
{
  "name": "014_Komm",
  "annotates": "014_Komm.wav",
  "levels": [
    {
      "name": "Tier_1",
      "type": "SEGMENT",
      "items": [
        {
          "id": 1,
          "labels": [
            {
              "name": "Tier_1",
              "value": "ja"
            }
          ],
          "sampleStart": 0,
          "sampleDur": 35120
        },
        {
          "id": 2,
          "labels": [
            {
              "name": "Tier_1",
              "value": "erstmal Hallo ich bin total beeindruckt über eure Vorträge ähm
                ich habe mir natürlich auch überlegt was ich machen soll "
            }
          ],
          "sampleStart": 35120,
          "sampleDur": 123408
        }
      ],
      ...
    }
  ]
}
```

TextGrid

TextGrid ist das Standardformat von Praat [Boe01]. TextGrid hat sich trotz einiger Mängel (ausschließlich zeitbasierte Elemente, keine expliziten Beziehungen zwischen Elementen, keine Kennzeichnung der Kodierung, keine Referenz

auf das annotierte Sprachsignal, schwierig zu parsen) als Austauschformat für zeitalignierte Mehrebenen-Annotationen durchgesetzt.

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 352.3413125
tiers? <exists>
size = 1
item []:
  item [1]:
    class = "IntervalTier"
    name = "Tier_1"
    xmin = 0
    xmax = 352.3413125
    intervals: size = 97
    intervals [1]:
      xmin = 0
      xmax = 2.195
      text = "ja"
    intervals [2]:
      xmin = 2.195
      xmax = 9.908
      text = "erstmal Hallo ich bin total beeindruckt über eure Vorträge ähm
ich habe mir natürlich auch überlegt was ich machen soll "
...

```

Tabellenformat (CSV)

Im Tabellenformat wird jedes Segment als Zeile mit vier Tabulator-getrennten Spalten in UTF-8 Kodierung gespeichert. Die Reihenfolge der Spalten ist **tmin**, **tier**, **text** und **tmax**. Dieses Format eignet sich besonders zum Austausch mit Datenbanken, Statistikpaketen und Tabellenkalkulationsprogrammen.

```
tmin tier text tmax
0 Tier_1 ja 2.195
2.195 Tier_1 erstmal Hallo ich bin total beeindruckt über eure Vorträge ähm
ich habe mir natürlich auch überlegt was ich machen soll 9.908
...

```

Der Umbruch in der zweiten Zeile ist nur zur besseren Lesbarkeit eingefügt. Tatsächlich darf ein Annotationslabel weder Zeilenumbruch noch Tabulatorzeichen enthalten.

BAS Partiturformat

Das BAS-Partiturformat ist ein Mehrebenen-Annotationsformat für symbolische und zeitbezogene Segmentdaten. Eine BPF-Datei hat mindestens die Referenzebene **ORT** für den orthographischen Wortlaut der transkribierten Äußerung. In dieser Ebene sind die Elemente durchnummeriert. Andere Annotationsebenen können sich auf die Elemente der Referenzebene beziehen.

In BAS Partiturd Dateien, die aus Octra exportiert wurden, stehen die Daten in der zeitlosen Ebene **ORT** bzw. der zeitalignierten Ebene **TRN**, die die Segmentgrenzen und die darin enthaltenen Wortsegmente beschreibt.

```

LHD: Partitur 1.3
SAM: 16000
NCH: 1
LBD:
ORT: 0 ja
ORT: 1 erstmal
ORT: 2 Hallo
ORT: 3 ich
ORT: 4 bin
ORT: 5 total
ORT: 6 beeindruckt
ORT: 7 über
ORT: 8 eure
ORT: 9 Vorträge
ORT: 10 ähm
ORT: 11 ich
ORT: 12 habe
ORT: 13 mir
ORT: 14 natürlich
ORT: 15 auch
ORT: 16 überlegt
ORT: 17 was
ORT: 18 ich
ORT: 19 machen
ORT: 20 soll
...
TRN: 0 35120 0 ja
TRN: 35120 123408 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 erstmal Hallo
    ich bin total beeindruckt über eure Vorträge ähm ich habe mir natürlich auch überlegt was
    ich machen soll

```

Der Zeilenumbruch in den letzten Zeilen wurde zur Verbesserung der Lesbarkeit eingefügt. Tatsächlich darf ein Transkriptionslabel keinen Zeilenumbruch enthalten.

Das BAS Partiturformat ist in [SBGW98] beschrieben. Online ist das Format unter

<http://www.bas.uni-muenchen.de/forschung/Bas/BasFormatsdeu.html>

beschrieben.

Protokolldaten

Oetra erlaubt das Protokollieren des Transkriptionsvorgangs. Dies ist allgemein für die Analyse der komplexen Tätigkeit der Transkription nützlich als auch hilfreich bei der Ausbildung von Transkribierern. Das Protokollieren kann im Einstellungen-Menü an- und ausgeschaltet werden.

Transkriptionsprotokolle werden als JSON-Objekte in UTF-8 Kodierung gespeichert.

```

{
  "version": "1.0",
  "encoding": "UTF-8",
  "projectname": "local",
  "lastupdate": "Wed, 28 Feb 2018 12:24:31 GMT",
  "duration": 5637461,
  "samplerate": 16000,
  "audiofile": "014_Komm.wav",
  "logs": [
    {

```

```

    "timestamp": 1519812702887,
    "type": "editor",
    "context": "2D-Editor",
    "value": "changed"
  },
  {
    "timestamp": 1519812825591,
    "type": "shortcut",
    "context": "multi-lines-viewer",
    "value": "boundary:add",
    "playerpos": 0,
    "caretpos": -1
  },
  {
    "timestamp": 1519812832036,
    "type": "shortcut",
    "context": "multi-lines-viewer",
    "value": "boundary:add",
    "playerpos": 0,
    "caretpos": -1
  },
  {
    "timestamp": 1519812833877,
    "type": "shortcut",
    "context": "multi-lines-viewer",
    "value": "boundary:add",
    "playerpos": 0,
    "caretpos": -1
  },
  ...

```

Literatur

- [Boe01] Paul Boersma. Praat, a system for doing phonetics by computer. *Glott International*, 5(9/10):341–345, 2001.
- [PD17] Julian Pömp and Christoph Draxler. OCTRA – A configurable browser-based editor for orthographic transcription. In *Proceedings Phonetik und Phonologie*, pages 145–148, Berlin, 2017.
- [SBGW98] F. Schiel, S. Burger, A. Geumann, and K. Weilhammer. The partitur format at BAS. In *Proc. LREC*, Granada, 1998.
- [WHJ17] Raphael Winkelmann, Jonathan Harrington, and Klaus Jänsch. Emu-SDMS: Advanced Speech Database Management and Analysis in R. *Computer Speech and Language*, 2017.