|  |  |  |
| --- | --- | --- |
|  |  |  |
| partitureditor2 | coma2 | exakt |
|  |  |  |
| **How to:**  **Segmentierung** | | |
|  | | |

Dieses Dokument erläutert die Verwendung intergrierter Segmentierungsalgorithmen der endlichen Maschinen des EXMARaLDA Partitur-Editors.

Vorkenntnisse über Algorithmen, endliche Maschinen oder reguläre Sprache sind für das Verständnis der ersten beiden Kapitel nicht vorrausgesetzt, da alle relevanten Aspekte in diesem Dokument erklärt werden. Dennoch sollten Sie vorab das Dokument „Understanding the basics of EXMARaLDA“ lesen.

Inhalte

[A. Über EXMARaLDA und Segmentierung 2](#_Toc372115773)

[1. Basis- und segmentierte Transkriptionen 2](#_Toc372115774)

[2. Segmente in Transkriptionssystemen 3](#_Toc372115775)

[B. Segmentierungsalgorithmen 5](#_Toc372115776)

[1. Funktionsweise 5](#_Toc372115777)

[2. Auswirkungen auf die Transkription 5](#_Toc372115778)

[C. Segmentierung im Partitur-Editor 6](#_Toc372115779)

[1. Segmentierungsoptionen 6](#_Toc372115780)

[2. Segmentierungsfehler 9](#_Toc372115781)

[3. Arbeiten mit Fehlerlisten 11](#_Toc372115782)

[4. Exportieren segmentierter Transkriptionen 12](#_Toc372115783)

[5. Segmentauszählung 12](#_Toc372115784)

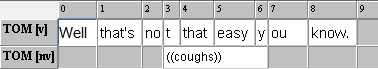
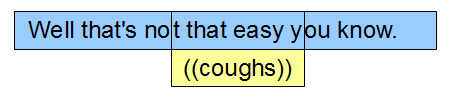
[6. Wortlistengenerierung 12](#_Toc372115785)

[Appendix: GAT2 Transkriptionskonventionen für Minimaltranskripte 13](#_Toc372115786)

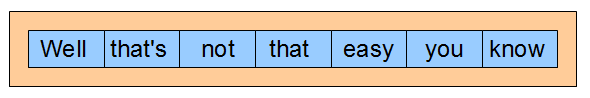
# A. Über EXMARaLDA und Segmentierung

## 1. Basis- und segmentierte Transkriptionen

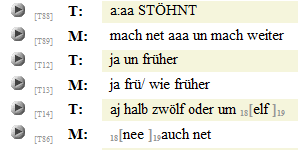
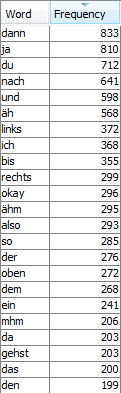
Streng genommen gibt es nicht *das eine* EXMARaLDA-Transkriptionsformat, sondern zwei: Die Basistranskription und die segmentierte Transkription. Bei der Transkription, die im Partitur-Editor erstellt wird, handelt es sich um eine EXMARaLDA Basistranskription mit der Dateierweiterung .exb[[1]](#footnote-1). Obwohl die Ereignisse mit ihren Grenzen als Segmente angezeigt werden, sind diese lediglich visueller Natur. Die Elemente auf der Zeitachse (bzw. die Zeitpunkte) dienen der Organisation verschiedener Ereignisse. Sie geben Auskunft über deren Reihenfolge, Gleichzeitigkeit und Überlappungen. Abgesehen von der organisatorischen Information über diese Ereignisse in der Basistranskription, gibt es keine sprachlichen oder anderweitig transkriptionsrelevanten Bedeutungen. Das hat zur Folge, dass die Eventgrenzen in der Basistranskription nicht als Wortgrenzen erkannt werden, selbst wenn für jedes Einzelwort ein separates Ereignis erzeugt wird. Wie Sie der folgenden Grafik entnehmen können, muss man sich für die Darstellung von Überlappungen von der Vorstellung lösen, dass eine Eventgrenze gleichzeitig eine Wortgrenze ist. Im Segmentierungsprozess werden ausschließlich Leer- und Interpunktionszeichen als Indikatoren für Wort- und Äußerungsgrenzen erkannt, darum ist es essentiell auch am Ende einer Eventgrenze ein Leerzeichen zu setzen.



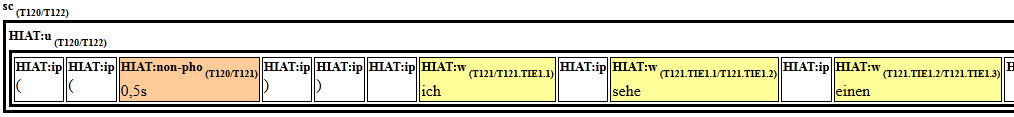
Wörter oder andere Einheiten, die durch verschiedene Symbole der jeweiligen Transkriptionskonvention gekennzeichnet werden, werden vor der Umwandlung in eine segmentierte Datei nicht erkannt. Fehlerfreie Basistranskriptionen lassen sich dann durch den Segmentierungsvorgang in das EXMARaLDA segmentierte Transkriptions-Format (mit der Dateierweiterung .exs) umwandeln. Über dieses Format werden explizit Informationen über verschiedene Einheiten der Transkription ausgedrückt, wie Wort- uns Äußerungsgrenzen. Das untenstehende Bild zeigt die Realisierung der Basistranskription des vorherigen Bildes als segmentierte Transkription (.exs), wobei die blauen Kästchen die Wortebene und das orange die Äußerungsebene darstellt.

Dank der Informationen über die verschiedenen Segmente, ergeben sich aus ein und derselben Transkription viele verschiedene Visualisierungsmöglichkeiten. Während die Eingabe, die im Partitur-Editor vorgenommen wird, auf der musikalischen Partitur basiert, kann die Ausgabe in einem komplett anderen Format erfolgen. Enthält die Segmentierung Ihrer Transkription Äußerungen wie z. B. in HIAT, können Sie eine zeitlich sortierte Äußerungsliste mit Markierung der Überlappungen generieren. Diese Äußerungsliste entspricht einem vertikalen Transkriptionsformat und hat nur wenig Ähnlichkeit mit dem der musikalischen Partitur. Stylesheets für verschiedene Visualisierungen finden Sie im Download Bereich auf [www.exmaralda.org](http://www.exmaralda.org).

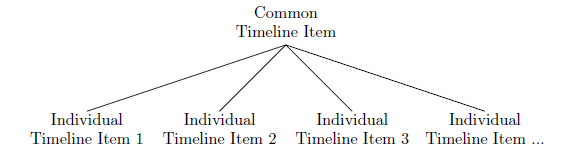
Sie können auch andere Einheiten segmentierter Transkriptionen für Visualisierungen und Darstellungsformate verwenden: Das Werkzeug EXACT kreiert, sammelt und sortiert die Wörter, zählt diese aus und erstellt für jedes Korpus mit Wortsegmentierung Wortlisten mit entsprechender Frequenz.



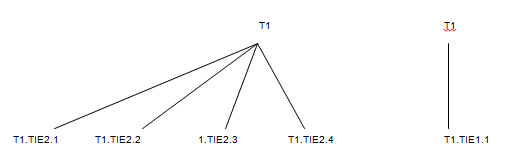
Segmentierte Transkriptionen lassen sich im Partitur-Editor weder öffnen, noch einsehen. Demnach besteht, abgesehen vom Arbeiten mit der XML-Datei, z. Z. nicht die Möglichkeit das Resultat der Segmentierung zu überprüfen oder gar Änderungen vorzunehmen. Falls Sie das Resultat der Segmentierung trotzdem einsehen möchten, finden Sie im Download Bereich auf [www.exmaralda.org](http://www.exmaralda.org) ein Stylesheet für die HTML-Visualisierung („Segmentierte Transkription als HTML-Darstellung“), welches auf die segmentierte Transkription angewendet werden kann.

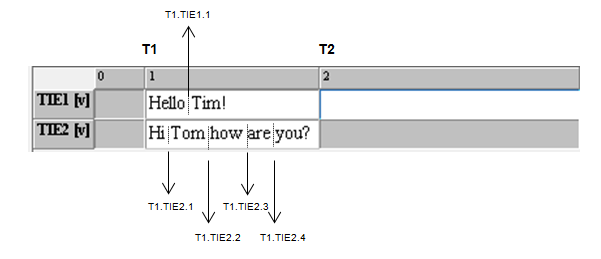


Die sogenannten *„Zeitachsenverzweigungen“* (time line forks) sind ein wichtiges Merkmal segmentierter Transkriptionen. Der Partitur-Editor erfordert keine manuelle Alignierung parallel verlaufender Ereignisse, wie zum Beispiel bei gleichzeitigem Sprechen. Aus diesem Grund müssen Äußerungen unterschiedlicher Sprecher nicht den selben Start- und Endpunkt haben, was eine detailliertere Alignierung ermöglicht. Jeder Sprecher hat seine eigene individuelle Zeitlinie, die (unabhängig von den anderen individuellen Zeitlinien der anderen Sprecher) an die allgemeine Zeitlinie gekoppelt ist.

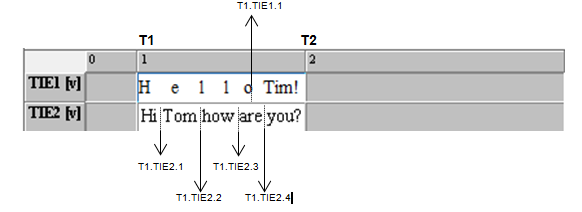
Wenn Tom „Hello Tim!“ und Tim „Hi Tom, how are you?“ sagt, wissen wir zwar, dass die Äußerungen zur gleichen Zeit beginnen und enden, können aber keine Informationen über Überlappungen und deren Ausmaß geben. Der gemeinsame Start- und Endzeitpunkt ist Teil der gemeinsamen Zeitachse, und für individuelle Elemente auf der Zeitachse gibt es für jedes Segment in jeder Spur eine Zeitachsenverzweigung. Bedenken Sie, dass der Segmentierungsalgorithmus diese Vorgehensweise vorschreibt, sobald keine gemeinsamen Elemente auf der Zeitachse existieren.

In unserem Beispiel:



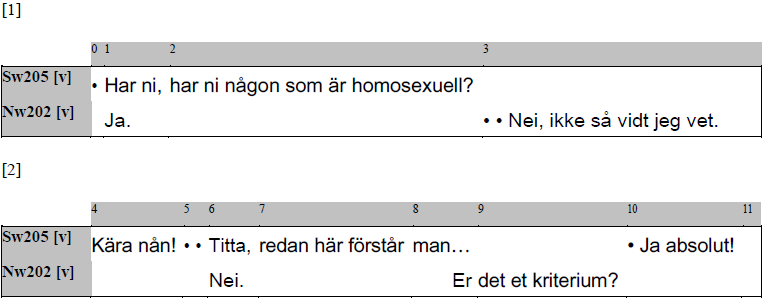


Es ist möglich, dass „Hello“ von Sprecher 1 stark in die Länge gezogen wird (im Folgenden durch Leerzeichen kenntlich gemacht und T1.TIE1.1 nicht mehr zwischen T1.TIE2.1 und T1.TIE2.2, sondern nunmehr zwischen T1.TIE2.3 und T1.TIE2.4 liegt.

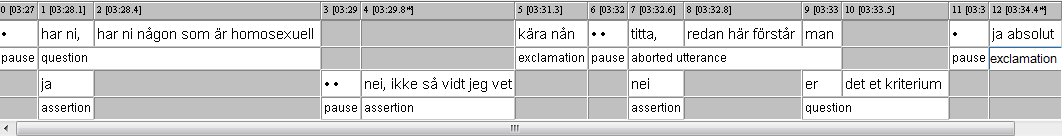


## 2. Segmente in Transkriptionssystemen

Transkriptionssysteme haben verschiedene Analyseeinheiten und unterschiedliche Symbole für deren Markierung. Das Transkriptionssystem HIAT nutzt unter anderem Satzpunkte, Ausrufe- und Fragezeichen für die Markierung segmentaler Äußerungen und für die Angabe der Äußerungsmodi.



Die Kodierung über den Umfang und die Äußerungsart würde ohne die Unterstützung von HIAT im Partitur-Editor in etwa so aussehen:



Mit der integrierten Unterstützung von Transkriptionskonventionen, lässt Sie EXMARaLDA bekannte Symbole der jeweiligen Transkription verwenden. Für alle Funktionen, die unter dem Punkt Segmentierung im Transkriptionsmenü aufgelistet sind, verwendet der Partitur-Editor dann die Symbole des Transkriptionssystems für die Segmentierung.

Die Segmentierung kann generell als eine Umwandlung der Transkriptionsspuren in mehrere einfache, flache Bäume, (wie in der formellen/ generativen Grammatik) verstanden werden, deren Strukturen vom jeweiligen Transkriptionssystem abhängen:

Die größte Einheit ist ausnahmslos die inhärente EXMARaLDA Segmentkette, die ununterbrochene Folge von Ereignissen, die zu einem Sprecher gehören. Innerhalb einer Segmentkette kann es verschiedene Äußerungsarten geben, die wiederrum Wörter, Pausen und nicht-phonologische Phänomene aufweisen können.

# treesB. Segmentierungsalgorithmen

Die Verwendung endlicher Maschinen ist nicht spezifisch für EXMARaLDA. Sollten Sie die Segmentierung lediglich für unterstützte Transkriptionssysteme verwenden wollen, können Sie den folgenden Abschnitt außer Acht lassen. Dennoch kann ein Grundverständnis über die Funktionsweise der Segmentierung helfen, solche Fehler von vornherein zu vermeiden.

## 1. Funktionsweise

Die Grundidee besteht darin, die „Maschine“ die Transkription Symbol für Symbol lesen zu lassen, damit sie überprüfen kann, ob alle Eingaben entsprechend der Konventionen getätigt worden sind, um aus den Eingaben die segmentierte Transkription erstellen zu können .Für die Erstellung einer korrekten Transkription, dürfen in dieser nur Symbole verwendet werden, die der Gruppe erlaubter Symbole dieser Transkriptionsform angehören. Ihre Verwendung muss gemäß den Regeln des Lexikons und der Syntax erfolgen. Die Maschine muss demnach in der Lage sein, alle vorstellbaren Transkriptionen entsprechend der Transkriptionskonventionen zu erkennen, ähnlich wie bei den natürlichen (menschlichen) Sprachen.

Glücklicherweise lassen sich Transkriptionen einfacher beschreiben als die natürlichen Sprachen. Aus diesem Grund können auch die sogenannten *endlichen Maschinen* für die Erkennung und Erstellung von Transkriptionen herangezogen werden. Die Maschine startet bei der Suche nach Segmentierungsfehlern bei der ersten Segmentkette einer Transkriptionsspur und arbeitet sich entweder bis an das Ende der Transkriptionsspur vor, oder stoppt an einer ungültigen Sequenz. Sie ist nicht in der Lage sich jedes gelesene Symbol zu merken, sondern erkennt nur den *state* des Fehlers, d.h., ob er sich „innerhalb eines Wortes einer Äußerung in einer Segmentkette“ oder „in einem nicht phonologischen Phänomen innerhalb einer Äußerung in einer Segmentkette“ befindet. In Abhängigkeit von dem ursprünglichen *state* und dem Input, bewegt sich die Maschine in einen anderen *state* ,d.h. sie liest Leerzeichen und ändert eine Einheit mit vom Wortstatus in den Äußerungsstatus.

## 2. Auswirkungen auf die Transkription

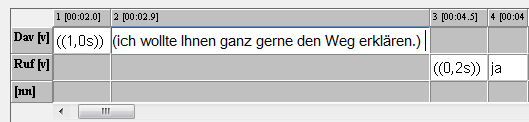
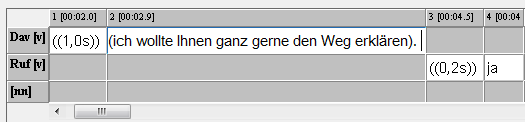
Da jedes Symbol und auch jede Kombination von Symbolen, ungeachtet des Kontextes, als Teil der Transkriptionskonvention gelesen wird, ist die Bedeutung der Symbole nicht so flexibel wie es die menschliche Interpretation erlauben würde. Ein Beispiel: Für die Markierung des Endes eines Deklarativsatzes wird nach HIAT ein Satzpunkt gesetzt. Die Bedeutung dieses Zeichens ist also vergeben und kann in solchen Transkriptionskonventionen nicht zusätzlich als Abkürzungssymbol auftreten. Der Segmentierungsalgorithmus des Partitur-Editors liest die Zeichen ohne den sprachlichen Kontext einzubeziehen. Sprachspezifische Abkürzungen wie in dem Satz „Das sind z.B. Hühner“ würden mit dem Satzpunkt als Abkürzungssymbol demnach in drei Äußerungen segmentiert werden:

1. Das sind z.
2. B.
3. Hühner.

Für die Markierung eines abgebrochenen Satzes ist die Unterscheidung zwischen drei aneinandergereihten Satzpunkten (…) und dem Ellipsen Symbol (…) wichtig. Reihen Sie lediglich Satzpunkte aneinander, kreieren Sie damit leere deklarative Äußerungen. Für den Leser ist die Unterscheidung klar, der Segmentierungsalgorithmus benötigt eindeutig identifizierbare Zeichen.

Einige Details, die Ihnen unwichtig erscheinen mögen, sind für die Funktionsweise von Segmentierungsalgorithmen von Bedeutung. Wenn Sie beispielsweise HIAT verwenden und eine Äußerung mit zwei undeutlichen Wörtern am Äußerungsende zu transkribieren, setzen Sie diese in einfache Klammern. Beachten Sie, die geschlossene Klammer vor dem Äußerungsendzeichen zu setzen. Schließen Sie zuerst den unverständlichen Part *innerhalb* der Äußerung, bevor Sie die komplette Äußerung abschließen.

Das gleiche Vorgehen wird bei einer Äußerung gebraucht, die im Ganzen unverständlich ist, da die Äußerung vor dem unverständlichen Teil in Klammern beginnt. Bei der linken Option in folgendem Beispiel würden sich Segmentierungsfehler ergeben, bei der rechten Option hingegen nicht.

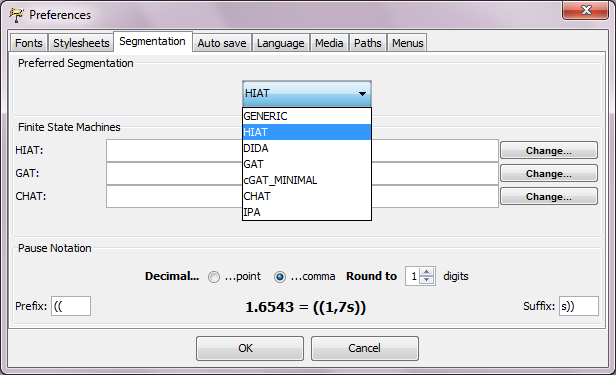


# C. Segmentierung im Partitur-Editor

## 1. Segmentierungsoptionen

Der EXMARaLDA Partitur-Editor stellt Segmentierungen für die meistgenutzten Transkriptionskonventionen wie HIAT, DIDA, GAT, CHAT, als auch für das IPA bereit. Ihre bevorzugte Segmentierung wählen Sie im Dropdown-Menü unter Bearbeiten > Voreinstellungen… > Segmentierung aus. Des Weiteren liefert EXMARaLDA eine generische Segmentierung für Transkriptionen ohne lineare Symbole (wie Äußerungsendzeichen) und schriftliche Daten.

Für die Beschreibung solcher Einheiten können Sie mit Annotationen arbeiten. Alle Segmentierungen nutzen die Notation in Segmentketten[[2]](#footnote-2) als Toplevel, d.h., wenn eine Segmentkette irrtümlich unterbrochen wird, wird durch das gesplittete Segment auch die Segmentierung fehlerhaft.

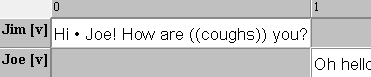


**Generisch**

Auf die generische Segmentierung können Sie immer zurückgreifen. Sie erkennt sowohl Wörter, die durch Leerzeichen oder Nicht-Wortzeichen (Interpunktionszeichen etc., die keine linguistische Information ausdrücken) als auch Segmentketten. Mit dieser Form der Segmentierung lassen sich Wort- und Segmentkettenlisten erstellen.

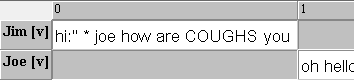
**HIAT**

Die Segmentierung nach HIAT erkennt Äußerungen, Wörter, Pausensymbole und nicht-phonologische Segmente entsprechend der HIAT-Konventionen. Andere bedeutungstragende Interpunktionszeichen wie Anführungszeichen für Zitate, Schrägstriche für Reparaturen, Bindestriche für Wortfragmente und Kommata für Wiederholungen werden zwar erkannt, aber in der segmentierten Transkription nur als Interpunktionszeichen (IP) transferiert und nicht wie Wörter oder Äußerungen gewertet. Es gibt also keine „Zitat-“ oder „Abbruch“-Segmente in der segmentierten Transkription. Im Folgenden sehen Sie die Segmentierung einer Segmentkette, die zwei Äußerungen beinhaltet, von denen die eine eine interne Pause und die andere ein internes nicht-phonologisches Ereignis beinhaltet, nämlich das Husten des Sprechers:



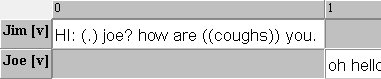
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Segment Chain | | | | | | | | | | | | | |
| Utterance | | | | | | Utterance | | | | | | | |
| W[[3]](#footnote-3) | IP | Non-Pho | IP | W | IP | W | IP | W | IP | Non-Pho | IP | W | IP |
| Hi |  | • |  | Joe | ! | How |  | are | (( | coughs | )) | you | ? |

**DIDA**

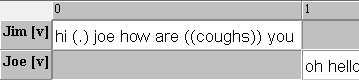
Die Segmentierung nach DIDA segmentiert Wörter, Pausen und nicht-morphemische Äußerungen. Sie beinhaltet aber auch noch andere Symbole, die in der Transkription vorkommen. Dennoch gibt es in DIDA keine Äußerungen oder gleichwertige „Brocken“. Im Folgenden sehen Sie erneut die Segmentierung einer Segmentkette, dieses Mal mit einer zusätzlichen Dehnung und Betonung auf dem ersten Wort:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Segment Chain | | | | | | | | | | | | |
| W | IP | PAUSE | IP | W | IP | W | IP | W | IP | NMÄ | IP | W |
| Hi:” |  | \* |  | joe |  | how |  | are |  | COUGHS |  | you |

**GAT und cGAT Minimal**

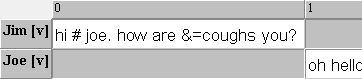
Bei der traditionellen GAT-Transkriptionskonvention wird auf Basis von Äußerungsendzeichen nach Phrasierungseinheiten segmentiert. Die Begrüßung von Jim bestünde demnach lediglich aus zwei Segmenten innerhalb der Segmentkette:

|  |  |
| --- | --- |
| Segment Chain | |
| PE | PE |
| HI: (.) joe? | how are ((coughs)) you. |

Für eine detaillierte Segmentierung wurde ein neues Set von Konventionen entwickelt: die [cGAT Minimal](http://agd.ids-mannheim.de/download/FOLKER-Transkriptionshandbuch.pdf)[[4]](#footnote-4) Konventionen. Sie sind im [Folker- Transkriptionshandbuch](http://agd.ids-mannheim.de/download/FOLKER-Transkriptionshandbuch.pdf)[[5]](#footnote-5) in deutscher Sprache detailliert beschrieben. Eine englische Übersetzung ist in Planung. Eine Kurzanleitung auf Deutsch finden Sie im Anhang dieses Dokuments. Folgen Sie dieser Beschreibung und Sie werden in der Lage sein, Segmente und Segmentketten noch genauer zu segmentieren. Die cGAT Minimalkonvention unterstützt keine Endzeichen für Intonationsphrasen. Zur Zeit gibt es also noch keine Möglichkeit sich die Vorteile beider Segmentierungsoptionen zunutze zu machen:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Segment Chain | | | | | | | | | | | | | |
| W | S[[6]](#footnote-6) | Pause | S | W | S | W | S | W | S | Non-Pho | S | W | S |
| hi |  | (.) |  | joe |  | how |  | are |  | ((coughs)) |  | you |  |

**CHAT**

Transkriptionen, die dem CHAT-Format folgen, welches im [CHAT manual](http://childes.psy.cmu.edu/manuals/chat.pdf)[[7]](#footnote-7) beschrieben wird, werden nach Äußerungen entsprechend der Äußerungsterminatoren segmentiert. Betrachtet man eine ähnliche Segmentkette wie im vorigen Abschnitt, erhält man eine Segmentierung in zwei Äußerungen:

|  |  |
| --- | --- |
| Segment Chain | |
| U[[8]](#footnote-8) | U |
| Hi # joe. | how are &=coughs you? |

**IPA**

Es besteht die Möglichkeit das Internationale Phonetische Alphabet für Transkriptionen im Partitur-Editor heranzuziehen. Die Segmentierung erfolgt bei Transkriptionen, die entsprechend der Konventionen angefertigt werden, in Wörter und Silben. Die Details dieser Konventionen wurden in:

* Thoma, Dieter & Tracy, Rosemarie (2005): L1 and Early L2: What's the difference? Talk, DGfS-Jahrestagung in Köln

veröffentlicht.

Eine schriftlich publizierte Version dieser Konventionen existiert z.Z. noch nicht. Die Konventionen sind aber, was die für die Segmentierung relevanten Zeichen anbelangt, denkbar einfach: Wörter werden mit einem Leerzeichen abgeschlossen, verschiedene Silben eines Wortes durch Punkte voneinander getrennt. Da Jim nur einsilbige Wörter verwendet, wird der Segmentierungsalgorithmus an einem anderen Beispiel verdeutlicht:

IPAsc

Mit der (optionalen) Segmentierung in Silben sieht die Segmentierung folgendermaßen aus:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Segment Chain | | | | | | | | | | | | | | |
| W | | | WB[[9]](#footnote-9) | W | | | WB | W | | | WB | W | | |
| SL[[10]](#footnote-10) | SB[[11]](#footnote-11) | SL |  | SL | SB | SL |  | SL | SB | SL |  | SL | SB | SL |
| næ: | . | ra |  | ɧʉ: | . | tər |  | ɪŋ: | . | ən |  | hɑ: | . | rɛ |

## 

## 2. Segmentierungsfehler

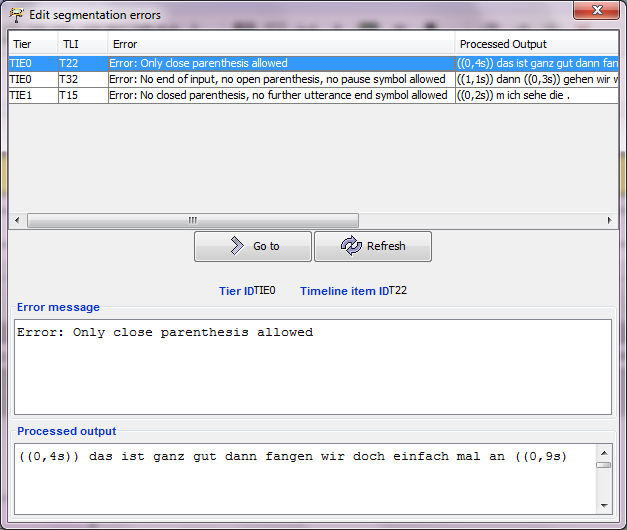
Wenn die Basistranskription nicht den Konventionen des Transkriptionssystems folgt, kann auch keine korrekte segmentierte Transkription ausgegeben werden. Bevor Sie eine solche Transkription exportieren können, müssen alle Fehler behoben werden. Um den Segmentierungsalgorithmus, den Sie unter Bearbeiten > Voreinstellungen… > Segmentierung festgelegt haben, anzuwenden und die Liste mit Segmentierungsfehlern zu laden, wählen Sie Transkription > Segmentierungsfehler….

Gehen Sie sicher, dass die gewünschte Segmentierung aktiviert ist. Sollten keine Segmentierungsfehler vorliegen, können Sie den Segmentierungsalgorithmus entweder für eine Segmentauszählung oder für den Export segmentierter Transkriptionen nutzen.

Sollten Segmentierungsfehler auftreten, erscheint eine Liste mit folgenden Informationen zu jedem Fehler:

* Tier: Die Spur, in der der Fehler auftritt
* TLI: Der Zeitpunkt des Fehlers in der Transkription
* Error: Beschreibung des Fehlers
* Processed output: Der letzte Abschnitt, der bearbeitet und als Ausgabe umgewandelt werden konnte

Damit Ihnen diese Informationen angezeigt werden können, klicken Sie die entsprechende Zeile an. Um die Transkription von Segmentierungsfehlern zu bereinigen, gehen Sie wie folgt vor: Markieren Sie den zu bearbeitenden Fehler, indem Sie die entsprechende Zeile der Tabelle anklicken. Klicken Sie auf Go to, um die Partitur an die Stelle zu bewegen, wo der Fehler auftritt. Falls nötig, konsultieren Sie vorliegendes Dokument für Informationen über Transkriptionskonventionen und Segmentierung. Beheben Sie den Fehler. Klicken Sie auf Refresh, um die noch verbleibenden Segmentierungsfehler anzuzeigen. Bedenken Sie Enter zu drücken, sich „aus dem Event zu bewegen” oder die Transkription zu speichern, damit die Korrekturen Anwendung finden.



**Generisch**

Die generische Segmentierung verursacht niemals Segmentierungsfehler.

**HIAT**

Die gängigsten Segmentierungsfehler in HIAT werden verursacht durch:

* Den Gebrauch von drei Satzpunkten anstelle des Ellipsen Symbols. Mit In Ereignissen ersetzen… können Sie diese Fehler beheben.
* Das Setzen der schließenden Klammern eines unverständlichen Teils nach dem Äußerungsendzeichen. Die schließende Klammer muss immer vor dem Äußerungsendzeichen stehen.
* Das Auslassen einer öffnenden- oder schließenden Klammer.
* Das Auslassen des abschließenden Anführungszeichens eines Zitates.

Fehler, die nicht erkannt werden, beinhalten:

* Irrtümlicherweise in mehrere Teile gesplittete Äußerungen. Solche Äußerungen entstehen, wenn die Toplevel Segmentkette unterbrochen wird, d.h. wenn ein leeres, („graues“) Ereignis inmitten der Äußerung auftritt.
* Äußerungen, die mit dem Ende einer Segmentkette aufhören, anstatt ein eigenes Äußerungsendzeichen zugewiesen zu bekommen.
* An Ereignisgrenzen verbundene Wörter, die entstehen, wenn das finale Leerzeichen unterschlagen wird. Erinnern Sie sich daran, dass Sie Ereignisgrenzen innerhalb von Wörtern setzen können, ohne diese zu splitten!

**DIDA**

Wenn Sie mit der DIDA-Konvention arbeiten, bedenken Sie bitte, dass Großbuchstaben nur in nicht-morphemischen Äußerungen zulässig sind.

**GAT und cGAT Minimal**

Segmentierungsfehler entstehen in der traditionellen *GAT*-Konvention nur, wenn Endzeichen für Intonationsphrasen falsch angewendet werden. Diese sollten nicht Bestandteil von Wörtern sein.

Der Gebrauch von *cGAT* bedarf mehr Genauigkeit. Stellen Sie die korrekte Anwendung der Konventionen sicher. Ein Unterschied zwischen GAT und cGAT liegt in der Unterscheidung zwischen Groß- und Kleinschreibung und dem Gebrauch von Symbolen für Dehnungen und Betonungen.

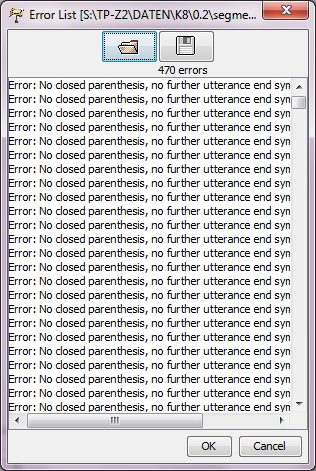
**CHAT**

Auch die CHAT Konvention ist für die Segmentierung irrelevant, da Segmentierungsfehler nur aufgrund fehlerhafter Anwendung von Äußerungsendzeichen entstehen. Alle Äußerungsabschlüsse müssen gemäß den Konventionen verwendet werden.

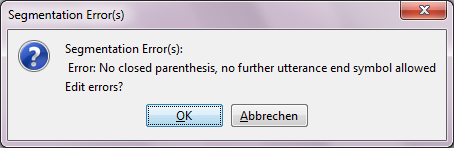
**IPA**

Denken Sie daran Leerzeichen zu setzen, wenn Sie Wörter voneinander trennen möchten. Punkte sollten nur für die Markierung von Silbengrenzen herangezogen werden.

## 3. Arbeiten mit Fehlerlisten

Die Fehlerliste kann sowohl individuell im Partitur-Editor als auch kollektiv für ein Korpus in Coma generiert werden. Genauer gesagt werden hierbei separate Fehlerlisten für Segmentierungs- und Strukturfehler erstellt. Bei der Fehlerliste handelt es sich um eine XML-Datei, die Informationen über die Fehler und die Transkription, in der sie auftreten, bereitstellt. Sie können die Fehlerliste in den Partitur-Editor laden, indem Sie auf den Ordner in dem Dialog klicken, der sich öffnet wenn Sie Datei > Fehlerliste… anwählen. Nach dem Öffnen der Fehlerliste sind alle Fehler des aktuellen Korpus ersichtlich. Mit einem Doppelklick auf den Fehler, öffnet sich die entsprechende Transkription an der Stelle, an der der Fehler auftritt. Ist der Fhler korrigiert, erscheint er nicht mehr in der Liste. Wenn es zu viele Fehler sind, um sie auf einmal zu korrigieren, können Sie die Liste der übrigen Fehler jederzeit durch einen Klick auf das Disketten-Symbol abspeichern.

## 4. Exportieren segmentierter Transkriptionen

Um eine segmentierte Transkription mit dem spezifischen Segmentierungsalgorithmus zu erstellen, den Sie unter Bearbeiten > Voreinstellungen gewählt haben, klicken Sie auf Transkription > Segmentierte Transkription exportieren…. Sollten keine Segmentierungsfehler auftreten, speichern Sie die Datei vorzugweise unter einem Namen, der mit der Basistranskription übereinstimmt. Hierfür empfiehlt sich das Suffix \_s. Wenn die Transkription Fehler enthält, werden Sie eine Fehlermeldung wie die folgende bekommen:

Klicken Sie OK, damit Ihnen die Liste der Segmentierungsfehler angezeigt wird.

## 5. Segmentauszählung

Um eine Liste der Häufigkeiten der verschiedenen Segmenttypen der Transkription mit dem gewählten Segmentierungsalgorithmus zu erstellen, wählen Sie Transkription > Transkription auszählen…. Sollten Sie eine Fehlermeldung erhalten, bearbeiten Sie diese wie im vorigen Abschnitt erklärt.

## 6. Wortlistengenerierung

Um eine Wortliste zu generieren, wählen Sie Wortliste…im Menüpunkt Transkription. Die Wortliste wird in einem neuen Fenster angezeigt und beinhaltet alle Einheiten als segmentierte Wörter in der Transkription. Durch Anklicken von Sprecher und Wörter können Sie die Liste je nach Auswahl alphabetisch sortieren lassen Zum Speichern der Liste als HTML-Datei, gehen Sie auf Speichern unter…. Sie haben folgende Optionen: Wählen Sie Simple word list (HTML) für eine Liste mit den alphabetisch geordneten Wörtern oder Word list by speaker (HTML) für eine Liste, in der die Worten anhand der Sprechern geordnet werden. Sollten Sie eine Meldung über Segmentierungsfehler bekommen, bearbeiten Sie diese wie im vorigen Abschnitt erklärt.

# Appendix: GAT2 Transkriptionskonventionen für Minimaltranskripte

*Thomas Schmidt*

**N.B.:** *GAT2 ist eine Transkriptionskonvention, die für das Deutsche entwickelt wurde. Einige der Konventionen lassen sich nicht auf das Englische übertragen, da die Entwicklung auf der deutschen Orthographie basiert. Dieses Dokument dient als praktischer Überblick über die Transkriptionskontrolle von GAT2 fürs Englische in Folker und stellt keine „richtige“ Transkriptionskonvention dar. Wenn Sie anderen Konventionen folgen möchten, wählen Sie* Bearbeiten > Voreinstellungen > Transkriptionsstufe *an, um die Kontrolle auszuschalten.*

1. **Wörter**

* Keine Großschreibung **wayne rooney**, nicht: Wayne Rooney
* Zahlen werden ausgeschrieben. Abkürzungen werden nicht gebraucht. **elf**, nicht: 11

**Doktor**, nicht: Dr.

* Keine Zeichensetzung (d.h.: keine Bindestriche in Wörtern, keine Satzpunkte in Abkürzungen, keine Apostrophe, wählen Sie für Kontraktionen stattdessen einen Unterstrich):

**they\_re**, nicht: they’re

1. **Pausen**

* Pausen (ausgenommen Mikropausen) werden vorzugweise in separaten Segmenten transkribiert
* Pausen (ausgenommen Mikropausen) sind vorzugweise keinem Sprecher zugeordnet
* Gemessene Pausen werden als Zahl mit Dezimalpunkt und zwei Nachkommastellen in Klammern gesetzt.

**(0.85)**

* Mikropausen (Pausen, die zu kurz für eine Messung sind) werden geschrieben als

**(.)**

1. **Unverständliches/ schwer Verständliches**

* Vermutungen (werden durch einfache Klammern angezeigt. Es dürfen nur ganze Wörter oder Wortketten in Klammern gesetzt werden. **(whatever), not what(ever)**
* Alternativvorschläge können, durch Schrägstrich getrennt, in die Klammern eingefügt werden.

**(cat/bat/mat)**

* Unverständliche Passagen werden als +++ transkribiert. Die Anzahl der +++ entspricht der Anzahl der (vermuteten) Silben.

**++++++ +++**

* Längere unverständliche Passagen oder Passagen, in denen die Silben nicht ausgemacht werden können, werden transkribiert als

**((unverständlich))**

1. **Gemischtes**

* Hörbares Atmen wird mit dem Grad-Symbol transkribiert (beim Einatmen vorgestellt, beim Ausatmen nachgestellt) und ein bis drei Einheiten des Buchstabens H entsprechend der Länge des Ereignisses.

**°hhh hh°**

* Kontraktionen werden durch einen Unterstrich angezeigt

**we\_re**

* Nicht phonologische Segmente werden in doppelte Klammern gesetzt.

**((lacht))**

1. Die Erweiterungen .exb und .exs sind nur für die EXMARaLDA-Instrumente von Interesse. Alle EXMARaLDA-Dateien sind XML- Dateien. [↑](#footnote-ref-1)
2. Ununterbrochene Folge von Ereignissen, die zu einem Sprecher gehören [↑](#footnote-ref-2)
3. = Wort [↑](#footnote-ref-3)
4. <http://agd.ids-mannheim.de/html/FOLKER-Transkriptionshandbuch.pdf> [↑](#footnote-ref-4)
5. http://agd.ids-mannheim.de/html/FOLKER-Transkriptionshandbuch.pdf [↑](#footnote-ref-5)
6. = Leerzeichen [↑](#footnote-ref-6)
7. http://childes.psy.cmu.edu/manuals/chat.pdf [↑](#footnote-ref-7)
8. = Äußerung [↑](#footnote-ref-8)
9. = Wortgrenze [↑](#footnote-ref-9)
10. = Silbe [↑](#footnote-ref-10)
11. = Silbengrenze [↑](#footnote-ref-11)