Annotation von Lernerdaten mit EXMARaLDA (Dulko)

Andreas Nolda
http://andreas.nolda.org

Stand: Mai 2019

Inhaltsverzeichnis

1 Download, Installation und Konfiguration

EXMARaLDA (Dulko) ist eine Toolsammlung für den EXMARaLDA-Partitureditor (https://exmaralda.org/en/partitur-editor-en/) für die Annotation von Lernerdaten in Lernerkorpora. Mit Hilfe der dort als Transformationsszenarien zur Verfügung gestellten XSLT 2-Stylesheets können Lernertexte tokenisiert, mit Wortarten, Lemmata und Satzspannen annotiert, Zielhypothesen editiert sowie Abweichungen zwischen Zielhypothesen und dem Lernertext mit Fehlerkategorien klassifiziert werden. Außerdem unterstützt EXMARaLDA (Dulko) den Annotator bei der Verwaltung von Metadaten. Der Name der Toolsammlung ist motiviert durch das Lernerkorpusprojekt Dulko (‚Deutsch-ungarisches Lernerkorpus‘; http://www1.ids-mannheim.de/gra/projekte/deutung/dulko.html), für das EXMARaLDA (Dulko) entwickelt wurde.

EXMARaLDA (Dulko) ist als Open Source unter https://bitbucket.org/nolda/exmaralda-dulko/ frei verfügbar. Unter „Downloads“ kann eine ZIP-Datei mit der letzten Version von EXMARaLDA (Dulko) heruntergeladen werden, die ausführbare Dateien für Microsoft Windows, Linux und Mac-OS enthält.

EXMARaLDA (Dulko) setzt eine erfolgreiche Installation des TreeTaggers (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/) sowie der Release-Version von EXMARaLDA (https://exmaralda.org/en/release-version/) voraus. Zu den Details der Installation und Konfiguration siehe die „Installation instructions“ und „Configuration instructions“ auf https://bitbucket.org/nolda/exmaralda-dulko/ bzw. in der Datei `README.md`, die in der erwähnten ZIP-Datei enthalten ist.

Lernertext (Auszug):

Wie in der ganzen Gesellschaft, *auch in der Regierung* ***sollte der Anzahl*** *der Frauen 50 % sein* [...].

intermediäre Zielhypothese:

Wie in der ganzen Gesellschaft ***sollte*** *auch in der Regierung* ***die Anzahl*** *der Frauen 50 % sein* [...].

Fehler: Zeichensetzung, Verbstellung, Genus

finale Zielhypothese:

Wie in der ganzen Gesellschaft sollte auch in der Regierung ***der Anteil*** *der Frauen 50 % sein* [...].

Fehler: Lexik

Abbildung 1: Lernertext, Zielhypothesen, Fehlerkategorien und Fehlerbereiche

2 Grundgedanken des Annotationsverfahrens

2.1 Dulko und Falko

Das im Dulko-Lernerkorpusprojekt verwendete Annotationsverfahren folgt im Prinzip den im Falko-Lernerkorpusprojekt an der Humboldt-Universität zu Berlin (https://www.linguistik.hu-berlin.de/de/institut/professuren/korpuslinguistik/forschung/falko/) entwickelten Annotationsrichtlinien (Reznicek *et al.* 2012). In einigen zentralen Aspekten weicht das Dulko-Annotationsverfahren jedoch davon ab (vgl. Hirschmann und Nolda 2019):

1. Beim Dulko-Annotationsverfahren können beliebig viele Zielhypothesen angegeben werden.
2. Fehler und ihre Bereiche werden beim Dulko-Annotationsverfahren explizit mit Hilfe von Fehlerkategorien unterschiedlicher sprachlicher Dimensionen annotiert.
3. Jeder Zielhypothese können beim Dulko-Annotationsverfahren Fehlerkategorien beliebiger sprachlicher Dimensionen zugeordnet werden.

2.2 Zielhypothesen

Im Dulko-Lernerkorpusprojekt werden zwei Hauptarten von Zielhypothesen unterschieden:

1. Eine *finale Zielhypothese* zu einer lernersprachlichen Einheit E bei einer Lesart I ist eine mit E bei I synonyme muttersprachliche Einheit mit minimalen Abweichungen.
2. Eine *intermediäre Zielhypothese* zu einer lernersprachlichen Einheit E bei einer Lesart I enthält weniger Abweichungen als eine finale Zielhypothese zu E bei I.

Intermediäre Zielhypothesen dienen dazu, Fehler zu repräsentieren, die bei der finalen Zielhypothese aufgrund überlappender Fehler ‚unsichtbar' bleiben. Ein solcher Fall liegt in dem Beispiel in Abbildung 1 vor (einem Auszug aus einem an der Universität Szeged im Wintersemester 2017/18 erhobenen Essays eines ungarischen Muttersprachlers), wo durch die Korrektur des lexikalischen Fehlers im Lernertext bei *Anzahl* in der finalen Zielhypothese der Genus-Fehler bei *der Anzahl* ansonsten unrepräsentiert bleiben würde. Abbildung 2 zeigt dasselbe Beispiel im EXMARaLDA-Partitureditor unter Verwendung von Fehlerkategorien aus dem Fehler-Tagsets des Dulko-Lernerkorpusprojekts.

Zu intermediären oder finalen Zielhypothesen können *alternative Zielhypothesen* angegeben werden, die andere Lesarten der lernersprachlichen Einheit repräsentieren (vgl. die bei Lüdeling und Hirschmann 2015 diskutierten Beispiele). Eine explizite Auszeichnung alternativer Zielhypothesen

Wie	in	der	ganzen	Gesellschaft	,		auch	in	der	Regierung	sollte	der	Anzahl	der	Frauen	50	%	sein	.
Wie	in	der	ganzen	Gesellschaft		sollte	auch	in	der	Regierung		die	Anzahl	der	Frauen	50	%	sein	.
					ZS														
						StV													
												Gen							
Wie	in	der	ganzen	Gesellschaft		sollte	auch	in	der	Regierung		der	Anteil	der	Frauen	50	%	sein	.
													Lex						

Abbildung 2: Annotation des Beispiels in Abbildung 1 im EXMARaLDA-Partitureditor

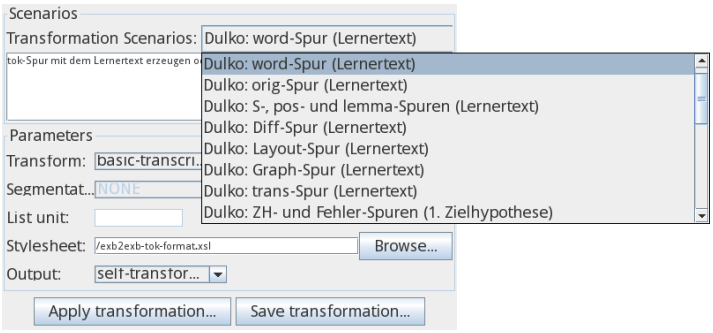

Abbildung 3: Menü mit Transformationsszenarien von EXMARaLDA (Dulko)

wird von EXMARaLDA (Dulko) gegenwärtig nicht unterstützt, kann aber bei Bedarf im EXMARaLDA-Partitureditor auf manuelle Weise erfolgen.

3 Das Dulko-Annotationsverfahren

3.1 Transformationsszenarien in EXMARaLDA (Dulko)

EXMARaLDA (Dulko) stellt Transformationsszenarien zur Verfügung, die im EXMARaLDA-Partitureditor unter „Transcription“ > „Transformation …“ aufgerufen werden können (vgl. Abbildung 3). Jedes Transformationsszenario ruft ein XSLT 2-Stylesheet im `share`-Verzeichnis von EXMARaLDA (Dulko) auf, das auf dem EXB-Datenformat des Partitureditors operiert. Diese Stylesheets können auch im Linux-Terminal mittels des Shell-Skripts `exb2exb.sh` aufgerufen werden.

Zu diesen Transformationsszenarien gehören insbesondere:

Dulko: `word`-Spur (Lernertext): `word`-Spur mit dem Lernertext erzeugen oder aktualisieren.

Dulko: `orig`-Spur (Lernertext): `orig`-Spur zur `word`-Spur hinzufügen oder aktualisieren.

Dulko: `S`-, `pos`- und `lemma`-Spuren (Lernertext): `pos`- und `lemma`-Spuren zur `word`-Spur hinzufügen oder aktualisieren.

Dulko: `Diff`-Spur (Lernertext): `Diff`-Spur zur `word`-Spur hinzufügen oder aktualisieren.

Dulko: `Layout`-Spur (Lernertext): `Layout`-Spur zur `word`-Spur hinzufügen oder aktualisieren.

Dulko: `Graph`-Spur (Lernertext): `Graph`-Spur zur `word`-Spur hinzufügen oder aktualisieren.

Dulko: `trans`-Spur (Lernertext): `trans`-Spur zur `word`-Spur hinzufügen oder aktualisieren.

Dulko: ZH- und Fehler-Spuren (1. Zielhypothese): ZH- und Fehler-Spuren für die 1. Zielhypothese hinzufügen oder aktualisieren.

Dulko: ZHS-, ZHpos- und ZHlemma-Spuren (1. Zielhypothese): ZHS-, ZHpos- und ZHlemma-Spuren zur 1. ZH-Spur hinzufügen oder aktualisieren.

Dulko: ZHDiff-Spur (1. Zielhypothese): ZHDiff-Spur zur 1. ZH-Spur hinzufügen oder aktualisieren.

Dulko: ZH- und Fehler-Spuren (2. Zielhypothese): ZH- und Fehler-Spuren für die 2. Zielhypothese hinzufügen oder aktualisieren.

Dulko: ZHS-, ZHpos- und ZHlemma-Spuren (2. Zielhypothese): ZHS-, ZHpos- und ZHlemma-Spuren zur 2. ZH-Spur hinzufügen oder aktualisieren.

Dulko: ZHDiff-Spur (2. Zielhypothese): ZHDiff-Spur zur 2. ZH-Spur hinzufügen oder aktualisieren.

Weitere Transformationsszenarien erlauben die Übertragung von Metadaten-Variablen aus dem Dulko-Template dulko.template.exb, die Durchnummerierung von Satzspannen nach einer manuellen Änderung, das Löschen überflüssiger Zeitpunkte auf der Zeitachse des Partitureditors, den Export der annotierten Daten ins HTML-Format sowie die Erzeugung einer ANNIS-kompatiblen Version der annotierten Daten und einer Pepper-kompatiblen Metadaten-Liste.

Im Folgenden wird die Anwendung der in der obigen Liste aufgeführten Transformationsszenarien am bereits eingeführten Beispiel illustriert.

3.2 Tokenisierung und Annotation am Beispiel

Zunächst wird im EXMARaLDA-Partitureditor das Dulko-Template dulko.template.exb geöffnet und unter einem neuen Namen gespeichert. Dieses Template enthält einschlägige Metadaten-Attribute nach dem von Granger und Paquot (2017) vorgeschlagenen Standard, die unter „Transcription“ > „Meta information …“ sowie unter „Transcription“ > „Speakertable …“ bearbeitet werden können. Diese Metadaten-Attribute können aus dem Dulko-Template auch in anderen Dateien mit Hilfe des Transformationsszenarios „Dulko: Metadaten“ übertragen werden; dabei werden fehlende Metadaten-Attribute ergänzt, ohne bereits vorhandene Attribute und Werte zu überschreiben. Annotationsbezogene Metadaten wie die Tokenzahl werden von einschlägigen Transformationsszenarien automatisch hinzugefügt bzw. aktualisiert.

Der Lernertext oder ein satzförmiger Teil davon wird nun in das erste Feld der word-Spur eingetragen (vgl. Abbildung 4). Anschließend wird dieser Text mit Hilfe des Transformationsszenarios „Dulko: word-Spur (Lernertext)“ tokenisiert (Abbildung 5). Im Stylesheet exb2exb-word.xsl sind die folgenden Tokenisierungsregeln implementiert:

- Interpunktionszeichen werden für den TreeTagger normalisiert[1] und in der Regel einzeln tokenisiert.
- Einfache URLs werden einzeln tokenisiert.
- Komponenten numerischer Spannen werden einzeln tokenisiert.
- Ordinalia mit finalem Punkt werden gemeinsam tokenisiert.

[1] Bei der Normalisierung werden typographische Symbole wie „ (U+201E), “ (U+201C), ‚ (U+201A), ‘ (U+2018), – (U+2013) und … (U+2026) durch entsprechende ASCII-Zeichen ersetzt.

Abbildung 4: Lernertext

Abbildung 5: Tokenisierter Lernertext

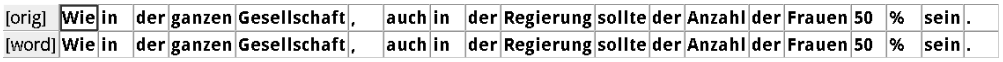

Abbildung 6: Vorlage für die Bearbeitung der unbearbeiteten Originalfassung

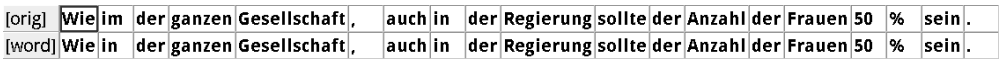

Abbildung 7: Tokenisierter Lernertext inklusive unbearbeiteter Originalfassung

- Wortartige Komponenten von Abkürzungen werden einzeln tokenisiert.
- Abkürzungen mit finalem Punkt werden gemeinsam tokenisiert.
- Wörter mit internem Apostroph werden vor dem Apostroph geteilt.
- Wörter mit Bindestrich ohne intervenierende Leerzeichen werden nicht geteilt.

Davon nicht erfasste Fälle können im EXMARaLDA-Partitureditor von Hand korrigiert werden.

Optional können eigenhändige Bearbeitungen durch den Lerner in der folgenden Weise repräsentiert werden. Zunächst wird mittels des Transformationsszenarios „Dulko: `orig`-Spur (Lernertext)“ eine weitere Spur angelegt, in die das Stylesheet `exb2exb-orig.xsl` die `word`-Spur als Vorlage kopiert (Abbildung 6). Anschließend werden in dieser Vorlage alle Bearbeitungen durch den Lerner rückgängig gemacht. Abweichungen zwischen der sich daraus ergebenden unbearbeiteten Originalfassung in der `orig`-Spur und dem bearbeiteten Lernertext in der `word`-Spur repräsentieren eigenhändige Bearbeitungen durch den Lerner – im vorliegenden Beispiel die Ersetzung des Wortes *im* durch das Wort *in* (vgl. Abbildung 7).

Darüber hinaus kann die `orig`-Spur genutzt werden, um die logisch-visuelle Gliederung des Lernertexts zu annotieren:

- Das Symbol _ (`U+5F`) auf der `orig`-Spur ohne entsprechendes Ereignis auf der `word`-Spur repräsentiert Leerraum im Lernertext (zum Beispiel eine Auslassung).
- Das Symbol ¶ (`U+B6`) auf der `orig`-Spur ohne entsprechendes Ereignis auf der `word`-Spur repräsentiert einen Absatzumbruch im Lernertext.
- Das Symbol | (`U+7C`) auf der `orig`-Spur ohne entsprechendes Ereignis auf der `word`-Spur repräsentiert einen Zeilenumbruch im Lernertext.

Möchte man die Trennung eines Wortes am Zeilenende angeben, so wird dieses Wort mit Hilfe des entsprechenden Werkzeugs des EXMARaLDA-Partitureditors auf der `orig`-Spur in drei Ereignisse geteilt:

1. ein Ereignis für den Wortteil am Ende der ersten Zeile,
2. ein Ereignis für den Trennstrich - (`U+2D`) und
3. ein Ereignis für den Wortteil am Anfang der zweiten Zeile.

[orig]	**Wie**	**im**	**der**	**ganzen**	**Gesellschaft**	**,**	**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[word]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**	**,**	**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[S]	s1																		
[pos]	KOUS	APPR	ART	ADJA	NN	$,	ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,	auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.

Abbildung 8: Tokenisierter Lernertext inklusive unbearbeiteter Originalfassung mit Satzspannen, Wortarten und Lemmata

[orig]	**Wie**	**im**	**der**	**ganzen**	**Gesellschaft**	**,**	**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[word]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**	**,**	**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[Diff]		CHA																	
[S]	s1																		
[pos]	KOUS	APPR	ART	ADJA	NN	$,	ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,	auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.

Abbildung 9: Tokenisierter Lernertext inklusive unbearbeiteter Originalfassung mit Abweichungen, Satzspannen, Wortarten und Lemmata

Das Symbol | braucht in diesem Fall nicht verwendet zu werden, da der Zeilenumbruch von – impliziert wird.

Als Nächstes wird der tokenisierte Lernertext in der `word`-Spur mittels des Transformationsszenarios „Dulko: `S-`, `pos-` und `lemma`-Spuren (Lernertext)“ mit Wortarten, Lemmata und Satzspannen annotiert (Abbildung 8). Intern werden dabei die Stylesheets `exb2exb-tag.xsl` und `exb2exb-s.xsl` aufgerufen, deren Anwendungsbereich durch den XSLT-Parameter `zh-number` gesteuert wird; steht er auf `0` wie im vorliegenden Fall, wird die `word`-Spur annotiert. Wortarten-Tagging und Lemmatisierung erfolgen mit dem lokal installierten TreeTagger (Schmid 1997), der für das Deutsche das STTS-Tagset verwendet (Schiller *et al.* 1999).[2] Satzspannen werden vom Stylesheet `exb2exb-s.xsl` nach den folgenden Regeln identifiziert:

- Eine Satzspanne endet mit einem Token, das der TreeTagger als `$.` (‚satzbeendende Interpunktion‘) taggt, oder mit einer Abkürzung, auf die ein Großbuchstabe folgt.
- Eine Überschrift am Anfang des Lernertexts, die in den Metadaten unter „Transcription“ > „Meta information …“ als Wert von `text_title` eingetragen ist, bleibt wie im Falko-Lernerkorpus ohne Satzspannen-Tag.

In Fällen, in denen die Satzspannen-Grenzen nicht automatisch erkannt werden, können die Satzspannen mit den Werkzeugen des EXMARaLDA-Partitureditors manuell korrigiert werden. Wird dabei die Zahl der Satzspannen geändert, können die Satzspannen mit dem Transformationsszenario „Dulko: Satzspannen“ erneut durchnummeriert werden.

Falls wie oben beschrieben eine `orig`-Spur angelegt wurde, wird nun mit dem Transformationsszenario „Dulko: `Diff`-Spur (Lernertext)“ eine weitere Spur erstellt, auf der automatisch Abweichungen zwischen der `word`-Spur und der `orig`-Spur mit den Abweichungstags `INS`, `DEL`, `CHA`, `SPLIT`, `MERGE`, `MOVS` bzw. `MOVT` markiert werden (vgl. Abbildung 9).[3] Diese Abweichungen werden vom Style-

[2] Der TreeTagger wird in `exb2exb-tag.xsl` über die Java-Klasse `TreeTagger.java` mittels der in EXMARaLDA enthaltenen TreeTagger-Anbindung „TreeTagger for Java“ (https://reckart.github.io/tt4j/) angesprochen. Dafür wird der XSLT 2-Prozessor Saxon in der Open-Source-Version 9.1.0.8J (https://sourceforge.net/projects/saxon/) mitgeliefert, der es erlaubt, Java-Methoden als XSLT-Funktionen aufzurufen.

[3] Dieselben Tags werden im Falko-Lernerkorpus zur Markierung von Abweichungen zwischen den Zielhypothesen und der Token-Ebene verwendet (vgl. Reznicek *et al.* 2012: 60 f.).

[orig]	In	diesem	Essay	wird	die	heutige	Situation	in	Ungarn	darge	-	stellt	,	die		mit	anderen	Ländern	verglichen	.	¶
[word]	In	diesem	Essay	wird	die	heutige	Situation	in	Ungarn	dargestellt			,		und	mit	anderen	Ländern	verglichen	.	
[Diff]														DEL	INS						
[Layout]											HYPH										PARB
[Graph]						UL															
[S]	s1																				
[pos]	APPR	PDAT	NN	VAFIN	ART	ADJA	NN	APPR	NE	VVPP			$,		KON	APPR	PIS	NN	VVPP	$.	
[lemma]	in	dies	Essay	werden	die	heutig	Situation	in	Ungarn	darstellen			,		und	mit	andere	Land	vergleichen	.	

Abbildung 10: Tokenisierter Lernertext inklusive unbearbeiteter Originalfassung mit Abweichungen, Textlayout, graphischen Auszeichnungen, Satzspannen, Wortarten und Lemmata

[orig]	Wie	im	der	ganzen	Gesellschaft	,	auch	in	der	Regierung	sollte	der	Anzahl	der	Frauen	50	%	sein	.
[word]	Wie	in	der	ganzen	Gesellschaft	,	auch	in	der	Regierung	sollte	der	Anzahl	der	Frauen	50	%	sein	.
[Diff]		CHA																	
[S]	s1																		
[pos]	KOUS	APPR	ART	ADJA	NN	$,	ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,	auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.
[ZH]	Wie	in	der	ganzen	Gesellschaft	,	auch	in	der	Regierung	sollte	der	Anzahl	der	Frauen	50	%	sein	.
[FehlerOrth]																			
[FehlerMorph]																			
[FehlerSyn]																			
[FehlerLex]																			
[FehlerSem]																			

Abbildung 11: Vorlage für die Bearbeitung der ersten Zielhypothese

sheet `exb2exb-diff.xsl` für jede Satzspanne getrennt berechnet.[4] Tokens, bei denen eine automatische Bestimmung nicht möglich ist, werden mit den vorläufigen Tags `MOVS/DEL` oder `MOVT/INS` versehen, die manuell zu disambiguieren sind.

Falls im Korpus auf der `orig`-Spur Leerraum, Absatzumbrüche, Zeilenumbrüche und/oder Trennstriche wie oben erläutert mit den Symbolen _, ¶, | bzw. - annotiert werden, kann mit dem Transformationsszenario „Dulko: `Layout`-Spur (Lernertext)“ eine `Layout`-Spur angelegt werden, auf der das Stylesheet `exb2exb-layout.xsl` diese Symbole automatisch den Tags `SPACE`, `PARB`, `LB` und `HYPH` markiert. Außerdem kann mit dem Transformationsszenario „Dulko: `Graph`-Spur (Lernertext)“ und dem Stylesheet `exb2exb-graph.xsl` eine `Graph`-Spur erstellt werden, auf der graphische Auszeichnungen des Lernertexts manuell annotiert werden können. Dafür stellt das Annotationswerkzeug (siehe unten) die Tags `UL` (Unterstreichung) und `UP` (durchgängige Großschreibung) zur Verfügung. Das Ergebnis dieser Bearbeitungsschritte illustriert das (gekürzte und leicht modifizierte) Beispiel in Abbildung 10.

Im Anschluss an diese Vorarbeiten werden mit Hilfe des Transformationsszenarios „Dulko: `ZH`- und `Fehler`-Spuren (1. Zielhypothese)“ Spuren für die erste Zielhypothese und die darauf bezogenen Fehlerkategorien angelegt (vgl. Abbildung 11). Dabei kopiert das Stylesheet `exb2exb-zh.xsl` die `word`-Spur als Vorlage für die manuelle Bearbeitung; das Stylesheet `exb2exb-fehler.xsl` erzeugt leere Fehler-Spuren für orthographische, morphologische, syntaktische, lexikalische und semantische Fehlerkategorien.

Bei der Bearbeitung der Zielhypothese können die Werkzeuge des EXMARaLDA-Partitureditors zur Manipulation von Ereignissen verwendet werden. Dazu gehört auch das Annotationswerkzeug unter „View“ > „Annotation panel“, mit dem die in der Datei `annotation-panel.xml` definierten Fehlerkategorien durch Doppelklick eingefügt werden können (Abbildung 12). Das Ergebnis der Bearbeitung der `ZH`- und `Fehler`-Spuren sieht im vorliegenden Beispiel wie in Abbildung 13 aus.

[4] Wurde das Transformationsszenario „Dulko: `S`-, `pos`- und `lemma`-Spuren (Lernertext)“ noch nicht aufgerufen und damit noch keine `S`-Spur angelegt, so bricht das Stylesheet `exb2exb-diff.xsl` mit einer Fehlermeldung ab.

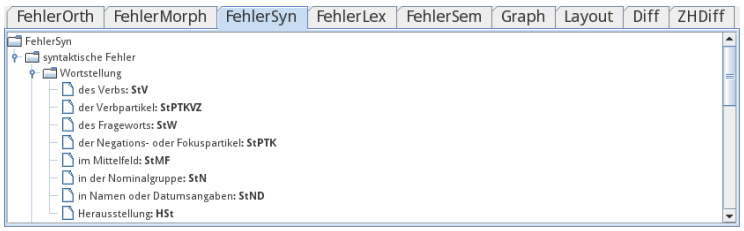

Abbildung 12: Annotationswerkzeug mit Fehlerkategorien

[orig]	**Wie**	**im**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[word]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[Diff]		CHA																		
[S]	s1																			
[pos]	KOUS	APPR	ART	ADJA	NN	$,		ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,		auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.
[ZH]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**		**sollte**	**auch**	**in**	**der**	**Regierung**		**die**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[FehlerOrth]						ZS														
[FehlerMorph]																				
[FehlerSyn]							StV													
[FehlerLex]													Gen							
[FehlerSem]																				

Abbildung 13: Erste Zielhypothese mit Fehlerkategorien

[orig]	**Wie**	**im**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[word]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[Diff]		CHA																		
[S]	s1																			
[pos]	KOUS	APPR	ART	ADJA	NN	$,		ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,		auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.
[ZH]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**		**sollte**	**auch**	**in**	**der**	**Regierung**		**die**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[ZHS]	s1																			
[ZHpos]	KOUS	APPR	ART	ADJA	NN		VMFIN	ADV	APPR	ART	NN		ART	NN	ART	NN	CARD	NN	VAINF	$.
[ZHlemma]	wie	in	die	ganz	Gesellschaft		sollen	auch	in	die	Regierung		die	Anzahl	die	Frau	50	%	sein	.
[FehlerOrth]						ZS														
[FehlerMorph]																				
[FehlerSyn]							StV													
[FehlerLex]													Gen							
[FehlerSem]																				

Abbildung 14: Erste Zielhypothese mit Satzspannen, Wortarten, Lemmata und Fehlerkategorien

Die fertige Zielhypothese wird wie der tokenisierte Lernertext mit Wortarten, Lemmata und Satzspannen annotiert (vgl. Abbildung 14). Dafür ruft das Transformationsszenario „Dulko: `ZHS-`, `ZHpos-` und `ZHlemma`-Spuren (1. Zielhypothese)“ wieder die Stylesheets `exb2exb-tag.xsl` und `exb2exb-s.xsl` auf, wobei es den XSLT-Parameter `zh-number` auf `1` setzt.

Abweichungen zwischen der `ZH`-Spur und der `word`-Spur werden mit Hilfe des Transformationsszenarios „Dulko: `ZHDiff`-Spur (1. Zielhypothese)“ mit den Abweichungstags `INS`, `DEL`, `CHA`, `SPLIT`, `MERGE`, `MOVS` bzw. `MOVT` markiert (Abbildung 15). Abweichungen, die vom Stylesheet `exb2exb-diff.xsl` nicht innerhalb einer Satzspanne automatisch auflösbar sind,[5] werden wieder mit den provisorischen Tags `MOVS/DEL` oder `MOVT/INS` versehen und müssen manuell disambiguiert werden.

[5] Das Stylesheet `exb2exb-diff.xsl` bricht mit einer Fehlermeldung ab, wenn das Transformationsszenario „Dulko: `ZHS-`, `ZHpos-` und `ZHlemma`-Spuren (1. Zielhypothese)“ noch nicht aufgerufen wurde und somit keine `ZHS`-Spur vorhanden ist.

[orig]	**Wie**	**im**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[word]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[Diff]		CHA																		
[S]	s1																			
[pos]	KOUS	APPR	ART	ADJA	NN	$,		ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,		auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.
[ZH]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**		**sollte**	**auch**	**in**	**der**	**Regierung**		**die**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[ZHDiff]						DEL	MOVT					MOVS	CHA							
[ZHS]	s1																			
[ZHpos]	KOUS	APPR	ART	ADJA	NN		VMFIN	ADV	APPR	ART	NN		ART	NN	ART	NN	CARD	NN	VAINF	$.
[ZHlemma]	wie	in	die	ganz	Gesellschaft		sollen	auch	in	die	Regierung		die	Anzahl	die	Frau	50	%	sein	.
[FehlerOrth]						ZS														
[FehlerMorph]																				
[FehlerSyn]							StV													
[FehlerLex]													Gen							
[FehlerSem]																				

Abbildung 15: Erste Zielhypothese mit Abweichungen, Satzspannen, Wortarten, Lemmata und Fehlerkategorien

[orig]	**Wie**	**im**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[word]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**	**,**		**auch**	**in**	**der**	**Regierung**	**sollte**	**der**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[Diff]		CHA																		
[S]	s1																			
[pos]	KOUS	APPR	ART	ADJA	NN	$,		ADV	APPR	ART	NN	VMFIN	ART	NN	ART	NN	CARD	NN	VAINF	$.
[lemma]	wie	in	die	ganz	Gesellschaft	,		auch	in	die	Regierung	sollen	die	Anzahl	die	Frau	50	%	sein	.
[ZH]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**		**sollte**	**auch**	**in**	**der**	**Regierung**		**die**	**Anzahl**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[ZHDiff]						DEL	MOVT					MOVS	CHA							
[ZHS]	s1																			
[ZHpos]	KOUS	APPR	ART	ADJA	NN		VMFIN	ADV	APPR	ART	NN		ART	NN	ART	NN	CARD	NN	VAINF	$.
[ZHlemma]	wie	in	die	ganz	Gesellschaft		sollen	auch	in	die	Regierung		die	Anzahl	die	Frau	50	%	sein	.
[FehlerOrth]						ZS														
[FehlerMorph]																				
[FehlerSyn]							StV													
[FehlerLex]													Gen							
[FehlerSem]																				
[ZH]	**Wie**	**in**	**der**	**ganzen**	**Gesellschaft**		**sollte**	**auch**	**in**	**der**	**Regierung**		**der**	**Anteil**	**der**	**Frauen**	**50**	**%**	**sein**	**.**
[ZHDiff]													CHA	CHA						
[ZHS]	s1																			
[ZHpos]	KOUS	APPR	ART	ADJA	NN		VMFIN	ADV	APPR	ART	NN		ART	NN	ART	NN	CARD	NN	VAINF	$.
[ZHlemma]	wie	in	die	ganz	Gesellschaft		sollen	auch	in	die	Regierung		die	Anteil	die	Frau	50	%	sein	.
[FehlerOrth]																				
[FehlerMorph]																				
[FehlerSyn]																				
[FehlerLex]														Lex						
[FehlerSem]																				

Abbildung 16: Zweite Zielhypothese mit Abweichungen, Satzspannen, Wortarten, Lemmata und Fehlerkategorien

Für die zweite Zielhypothese wird mit dem Transformationsszenario „Dulko: `ZH-` und `Fehler`-Spuren (2. Zielhypothese)“ eine Vorlage auf Grundlage der ersten Zielhypothese erstellt. Nach deren Bearbeitung werden die Transformationsszenarien „Dulko: `ZHS-`, `ZHpos-` und `ZHlemma`-Spuren (2. Zielhypothese)“ und „Dulko: `ZHDiff`-Spur (2. Zielhypothese)“ aufgerufen, um die zweite Zielhypothese ihrerseits mit Wortarten, Lemmata, Satzspannen und Abweichungstags zu annotieren. Das Ergebnis ist in Abbildung 16 wiedergegeben. Man beachte, dass die Abweichungstags auf der zweiten `ZHDiff`-Spur Abweichungen zwischen der 2. Zielhypothese und der 1. Zielhypothese und nicht etwa zwischen der 2. Zielhypothese und dem Lernertext markieren.

Damit ist die Annotation des vorliegenden Beispiels abgeschlossen. Sollten in einem späteren Arbeitsschritt Änderungen am Lernertext oder an den Zielhypothesen vorgenommen werden, so können die obigen Transformationsszenarien erneut aufgerufen werden, um die automatisch erzeugten Annotationen zu aktualisieren. Außerdem können weitere satzförmige Teile des Lernertexts in der `word`-Spur ergänzt und in analoger Weise tokenisiert und annotiert werden.

Literatur

Granger, Sylviane und Magali Paquot (2017). Core metadata for learner corpora. Draft 1.0. Manuskript, Louvain-la-Neuve: Université catholique de Louvain.

Hirschmann, Hagen und Andreas Nolda (2019). Dulko – auf dem Weg zu einem deutsch-ungarischen Lernerkorpus. In *Neues vom heutigen Deutsch: Empirisch – methodisch – theoretisch*, hg. v. Ludwig Eichinger und Albrecht Plewnia, Institut für Deutsche Sprache: Jahrbuch 2018, Berlin: de Gruyter, 339–342.

Lüdeling, Anke und Hagen Hirschmann (2015). Error annotation systems. In *The Cambridge Handbook of Lerner Corpus Research*, hg. v. Sylviane Granger, Gaëtanelle Gilquin und Fanny Meunier, Cambridge: Cambridge University Press, 135–157.

Reznicek, Marc *et al.* (2012). Das Falko-Handbuch: Korpusaufbau und Annotationen. Version 2.01. Manuskript, Humboldt-Universität zu Berlin. https://www.linguistik.hu-berlin.de/de/institut/professuren/korpuslinguistik/forschung/falko/FalkoHandbuchV2 [26. Jan. 2017].

Schiller, Anne *et al.* (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset). Manuskript, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung und Universität Tübingen, Seminar für Sprachwissenschaft. http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf [11. März 2018].

Schmid, Helmut (1997). Probabilistic part-of-speech tagging using decision trees. In *New Methods in Language Processing*, hg. v. Daniel B. Jones und Harold L. Somers, London: Routledge, 154–164.