Korpora Deutsch als Fremdsprache

Zeitschrift Korpora Deutsch als Fremdsprache

Jahrgang 4, Nummer 2 (Dezember 2024)
ISSN 2749-4411

DIE DULKO-TOOLS DES EXMARALDA-PARTITUR-EDITORS
Von einer externen Toolsammlung zum integrierten Bestandteil

Andreas Nolda
Berlin-Brandenburgische Akademie der Wissenschaften

Abstract

Die Dulko-Tools aus dem Dulko-Lernerkorpusprojekt für die semiautomatische Annotation von Lernertexten sind seit 2023 offizieller Teil des EXMARaLDA-Partitur-Editors. Für die im Juni 2024 erschienene Version 1.8 wurden sie weiter verbessert. In diesem Bericht aus der Praxis wird gezeigt, wie mit Hilfe der Dulko-Tools der aktuellen Version des EXMARaLDA-Partitur-Editors Lernertexte und andere geschriebene Daten annotiert werden können. Deren Annotationswerkzeuge werden anhand der Annotation eines Beispiels aus Dulko-Korpus eingeführt. Außerdem werden Hilfswerkzeuge für annotierte Daten vorgestellt, die sich für die Datenanzeige und Datenaufbereitung nutzen lassen.

Keywords: Lernerkorpus; Metadaten; Tokenisierung; POS-Tagging; Lemmatisierung; Fehlerannotation

Abstract

The Dulko tools for semi-automatic annotation of learner texts, originally developed for the Dulko learner-corpus project, are an official part of the EXMARaLDA Partitur Editor since 2023. For version 1.8 of the Partitur Editor, released in June 2024, the Dulko tools have been further improved. This report demonstrates how learner texts and other written data can be annotated by means of the Dulko tools from the current version of the EXMARaLDA Partitur Editor. The use of its annotation tools is illustrated by means of an example from the Dulko corpus, and auxiliary tools are presented for the visualisation and processing of annotated data.

Keywords: learner corpus; metadata; tokenisation; POS tagging; lemmatisation; error annotation

1. Einführung: Eine kurze Geschichte der Dulko-Tools des EXMARaLDA-Partitur-Editors

Obwohl ursprünglich für die Annotation gesprochener Korpusdaten konzipiert, wurde und wird der EXMARaLDA-Partitur-Editor (vgl. Schmidt / Wörner 2014)[1] auch für die Annotation von Lernertexten und anderen geschriebenen Daten eingesetzt. Nach mehrjähriger Entwicklung enthält der EXMARaLDA-Partitur-Editor seit dem Release von Juli 2023 (Version 1.7) offiziell Tools für die semiautomatische Annotation solcher Daten – die sogenannten *Dulko-Tools* aus dem Dulko-Lernerkorpusprojekt.

Im Dulko-Lernerkorpusprojekt wurde von 2017 bis 2021 am Lehrstuhl für Germanistische Linguistik an der Universität Szeged in Ungarn das deutsch-ungarische Lernerkorpus Dulko (vgl. Beeh et al. 2021) erstellt[2]. Dieses Korpus enthält fehlerannotierte deutschsprachige Essays und Übersetzungen vom Ungarischen ins Deutsche, die zusammen mit einschlägigen Metadaten kontrolliert

[1] https://exmaralda.org (18.06.2024).
[2] Das Dulko-Lernerkorpusprojekt war ein Teilprojekt einer von der Alexander-von-Humboldt-Stiftung geförderten Institutspartnerschaft namens „Deutsch-ungarischer Sprachvergleich: korpustechnologisch, funktional-semantisch und sprachdidaktisch“ (DeutUng) des Instituts für Germanistik der Universität Szeged mit dem Leibniz-Institut für Deutsche Sprache in Mannheim (vgl. https://www.ids-mannheim.de/gra/projekte/deutung/, 18.06.2024).

erhoben wurden. Probanden waren Szegeder Studenten[3] mit Ungarisch als Muttersprache und Deutsch als Fremdsprache auf dem Niveau B2 oder C1. Zu Projektende wurden die beiden Teilkorpora `DulkoEssay-v1.0` und `DulkoTranslation-v1.0` auf dem ANNIS-Server[4] des Lehrstuhls für Korpuslinguistik und Morphologie an der Humboldt-Universität zu Berlin für die Recherche verfügbar gemacht.

Für die Annotation der Lernertexte im Dulko-Projekt wurde das Standoff-Annotationsverfahren der Falko-Lernerkorpora (vgl. Reznicek et al. 2012) weiterentwickelt. Das Dulko-Annotationsverfahren (vgl. Hirschmann / Nolda 2019; Beeh et al. 2021; Nolda 2023) sieht vor, dass die Lernertexte wie beim Falko-Annotationsverfahren tokenisiert und mit Satzspannen, Wortklassen und Lemmata annotiert werden. Abweichungen zwischen Tokens des Lernertexts und Tokens einer Zielhypothese werden als Fehler interpretiert (vgl. Lüdeling / Hirschmann 2015), die als Spannen mit einem mehrdimensionalen Fehler-Tagset annotiert werden. Um auch einander überlappende Fehler repräsentieren zu können, erlaubt das Dulko-Annotationsverfahren die Angabe beliebig vieler Zielhypothesen.

Als Annotationswerkzeug wählte das Dulko-Projekt den EXMARaLDA-Partitur-Editor. Dieser war bereits im Falko-Projekt für manuelle Annotationsschritte eingesetzt worden, bevor man diese dort stattdessen in Microsoft Excel mit Hilfe eines von Marc Reznicek entwickelten Add-ins durchführte (vgl. Reznicek et al. 2012: 4). Im Unterschied zu der Excel-Lösung hat der EXMARaLDA-Partitur-Editor jedoch mehrere Vorteile: Er ist frei verfügbar, läuft stabil auf allen Desktop-Betriebssystemen, verwendet ein vertrautes horizontales Partiturformat und besitzt zahlreiche Funktionen für die effektive Bearbeitung von Spuren und Ereignissen.

Damit die Annotatoren im Dulko-Projekt sämtliche manuellen und automatischen Annotationsschritte direkt im EXMARaLDA-Partitur-Editor durchführen können, entwickelte der Verfasser des vorliegenden Berichts ab 2016 eine Reihe von XSLT-basierten Tools für den EXMARaLDA-Partitur-Editor und machte sie unter dem Namen *EXMARaLDA (Dulko)* zunächst auf Bitbucket und später auf Sourcehut als Open Source verfügbar[5]. Da die Dulko-Tools zwischenzeitlich von anderen Projekten nachgenutzt wurden, entschloss sich ihr Entwickler, sie in Kooperation mit Thomas Schmidt in das EXMARaLDA-Quellcode-Repositorium[6] zu integrieren. Mit dem EXMARaLDA-Release von 2023 erübrigt sich die komplexe separate Installation der Dulko-Tools. Außerdem sind die Dulko-Tools seitdem auch für die Annotation von Texten jenseits deutschsprachiger Lernertexte nutzbar. Die aktuelle, im Juni 2024 erschienene Version 1.8 des EXMARaLDA-Partitur-Editors[7] bringt weitere Verbesserungen der Dulko-Tools.

In diesem Bericht aus der Praxis soll gezeigt werden, wie Lernertexte und andere geschriebene Daten mit Hilfe der Dulko-Tools der aktuellen Version des EXMARaLDA-Partitur-Editors annotiert werden können. Die Annotationswerkzeuge dafür werden in Abschnitt 2 anhand einer Beispielannotation eingeführt. Darüber hinaus enthalten die Dulko-Tools Hilfswerkzeuge für die Datenanzeige und Datenaufbereitung annotierter Daten, die in Abschnitt 3 vorgestellt werden.

[3] Nicht-movierte Rollenbezeichnungen sind im vorliegenden Text entsprechend ihrer lexikalischen Bedeutung sexusneutral zu verstehen, sofern sich aus dem pragmatischen Kontext nichts anderes ergibt.
[4] https://korpling.org/annis/ (18.06.2024).
[5] https://sr.ht/~nolda/exmaralda-dulko/ (18.06.2024). Diese Arbeit wurde 2018 mit dem Innovationspreis der Universität Szeged ausgezeichnet.
[6] https://github.com/Exmaralda-Org/exmaralda (18.06.2024).
[7] https://exmaralda.org/de/offizielle-version/ (18.06.2024).

2. Die Dulko-Tools des EXMARaLDA-Partitur-Editors: Annotationswerkzeuge

In diesem Abschnitt werden die von den Dulko-Tools des EXMARaLDA-Partitur-Editors zur Verfügung gestellten Annotationswerkzeuge eingeführt: das Dulko-Template, Dulko-Transformationsszenarien für die Annotation sowie das Dulko-Annotationspanel. Als Beispiel dafür wird die Annotation eines Auszugs aus einem Lernertext des Dulko-Korpus (Lernertext `Feminismus_4`, 1. Teil der Satzspanne `s17`) gemäß dem Dulko-Annotationsverfahren herangezogen.

Illustriert wird dies durch eine Reihe von Abbildungen mit Screenshots aus einer aktuellen Linux-Installation des EXMARaLDA-Partitur-Editors (Version 1.8). Entsprechend den Voreinstellungen sind die Menü-Einträge und Dialoge auf Englisch.

2.1 Das Dulko-Template

Für die Annotation von Lernertexten empfiehlt es sich, zunächst das Dulko-Template zu laden („New from Dulko template“ im „File“-Menü). Das Dulko-Template – eine interne XML-Datei in EXMARaLDAs EXB-Format – definiert eine Partitur mit einer leeren `word`-Spur. In diese Spur ist der zu annotierende Text einzutragen; dies kann der vollständige Text sein oder zunächst nur ein satzförmiger Teil davon (vgl. Abbildung 1).

Abbildung 1
`word`-Spur mit Lernertext

Darüber hinaus definiert das Dulko-Template lernerkorpusspezifische Metadatenattribute gemäß den „Core metadata for learner corpora“ (vgl. Granger / Paquot 2017, Draft 1.0)[8]. Die korpus- und dokumentspezifischen Attribute findet man in den Meta-Informationen des EXMARaLDA-Partitur-Editors („Meta information“ im „Transcription“-Menü oder der entsprechende Button auf der Toolbar; vgl. Abbildung 2), die lernerspezifischen Attribute in der Sprechertabelle des EXMARaLDA-Partitur-Editors („Speakertable“ im „Transcription“-Menü oder der entsprechende Button auf der Toolbar; vgl. Abbildung 3). Dort lassen sich die entsprechenden Attributwerte eintragen. Die Werte annotationsbezogener Attribute wie `corpus_size` (Tokenzahl), `pos_tagged` (Wortklassen-Tagging), `error_annotated` (Fehlerannotation) oder `annotation_other` (weitere Annotationen) werden von Dulko-Transformationsszenarien automatisch ergänzt bzw. aktualisiert.

In der Sprechertabelle ist Deutsch als verwendete Zweitsprache voreingestellt („Language(s) used“ und „Second language(s)“); die Erstsprache („First language(s)“) ist nicht spezifiziert. Mit Hilfe des Buttons „Edit languages“ lassen sich diese Angaben anpassen. In der Sprechertabelle kann außerdem das Geschlecht des Lerners angegeben und eine Abkürzung als Lerner-ID eingetragen werden. Die Meta-Informationen enthalten Felder für Projektname, Transkriptionsname und Transkriptionskonventionen[9].

[8] Zu den Metadaten-Konventionen des Dulko-Lernerkorpusprojekts vgl. Beeh et al. (2021: Abschnitt 4).
[9] Die Eintragungen in Meta-Informationen und Sprechertabelle bei „Fixed attributes“ und „Languages“ werden beim Daten-Export ebenfalls als „Core metadata“-Attribute ausgegeben (vgl. Abschnitt 3.2).

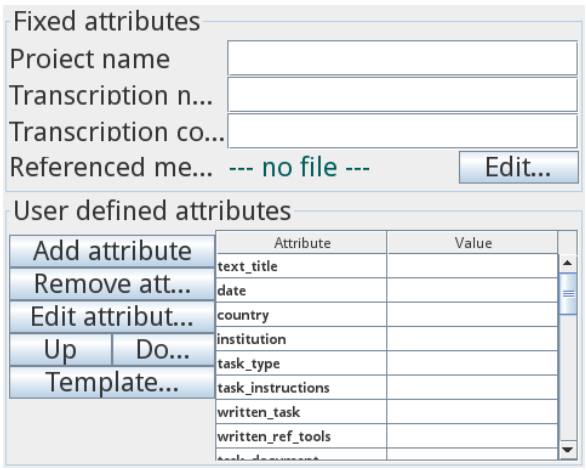

Abbildung 2
Meta-Informationen des Dulko-Templates

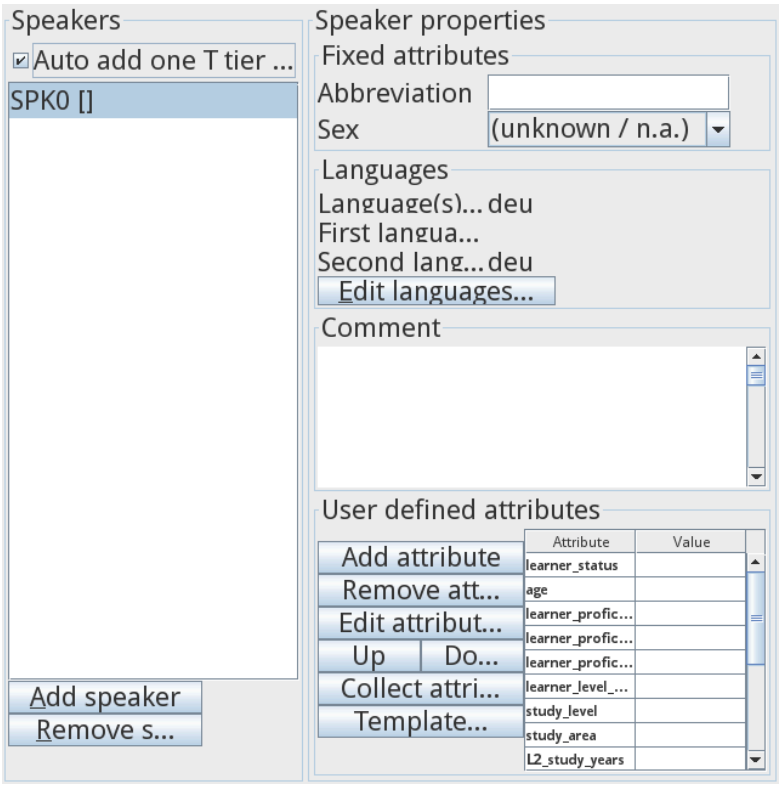

Abbildung 3
Sprechertabelle des Dulko-Templates

2.2 Dulko-Transformationsszenarien für die Annotation

Zur semiautomatischen Annotation von Lernertexten stellen die Dulko-Tools des EXMARaLDA-Partitur-Editors eine Reihe von Transformationsszenarien zur Verfügung. Diese können im Transformationsdialog des EXMARaLDA-Partitur-Editors ausgewählt werden („Transformation“ im „Transcription“-Menü oder der entsprechende Button auf der Toolbar; vgl. Abbildung 4). Manche

der Dulko-Transformationsszenarien (wie diejenigen für die Fehlerannotation) sind lernerkorpusspezifisch; andere lassen sich auch für die Annotation von geschriebenen Daten anderer Art nutzen. Jedes Transformationsszenario ruft ein internes XSLT-Stylesheet auf, das die XML-Repräsentation von Partitur und Metadaten im EXB-Format manipuliert.

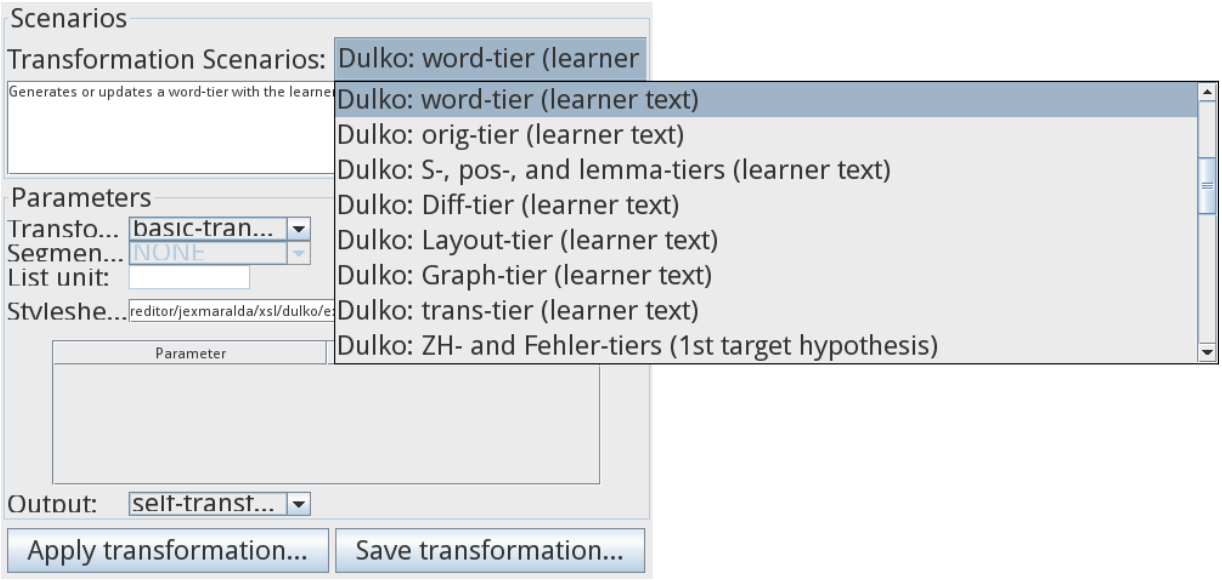

Abbildung 4
Dulko-Transformationsszenarien im Transformationsdialog

Um den Lernertext in Abbildung 1 gemäß dem Dulko-Annotationsverfahren zu annotieren, führt man zunächst das Transformationsszenario „Dulko: `word`-tier (learner text)“ aus, indem man dieses im Transformationsdialog auswählt und anwendet („Apply transformation“). Dieses Transformationsszenario tokenisiert den Text in der `word`-Spur, indem es pro Token ein eigenes Ereignis anlegt (vgl. Abbildung 5). Außerdem normalisiert es typographische Interpunktionszeichen zu ASCII-Zeichen.

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

Abbildung 5
Tokenisierter Lernertext

Für die Tokenisierung von Abkürzungen wird auf eine sprachspezifische Abkürzungsliste des TreeTaggers (vgl. Schmid 1997)[10] zurückgegriffen. Abkürzungslisten für zahlreiche Sprachen sind auf der TreeTagger-Homepage als Teil der „tagging scripts“ verfügbar. Benötigt wird eine Abkürzungsliste für die erste Sprache, die in der Sprechertabelle als verwendete Sprache eingestellt ist (vgl. Abschnitt 2.1). Für die im Dulko-Template voreingestellte Sprache Deutsch wird eine Abkürzungsliste des Namens `german-abbreviations` benötigt. Der Pfad zu dieser Abkürzungsliste ist zusammen mit dem Pfad zur TreeTagger-Installation in den TreeTagger-Einstellungen des EXMARaLDA-Partitur-Editors anzugeben („Preferences“ im „Edit“-Menü, „Paths“-Reiter, „TreeTagger directory“[11]

[10] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (18.06.2024).
[11] Alternativ kann der Pfad zur TreeTagger-Installation auch in einer Umgebungsvariable namens TREETAGGER_HOME gesetzt werden.

bzw. „Abbreviations file“; vgl. Abbildung 6). Existiert einer dieser Pfade nicht oder stimmt die Sprache der Abkürzungsliste nicht mit der ersten verwendeten Sprache in der Sprechertabelle überein, so wird ein Fehler ausgegeben.

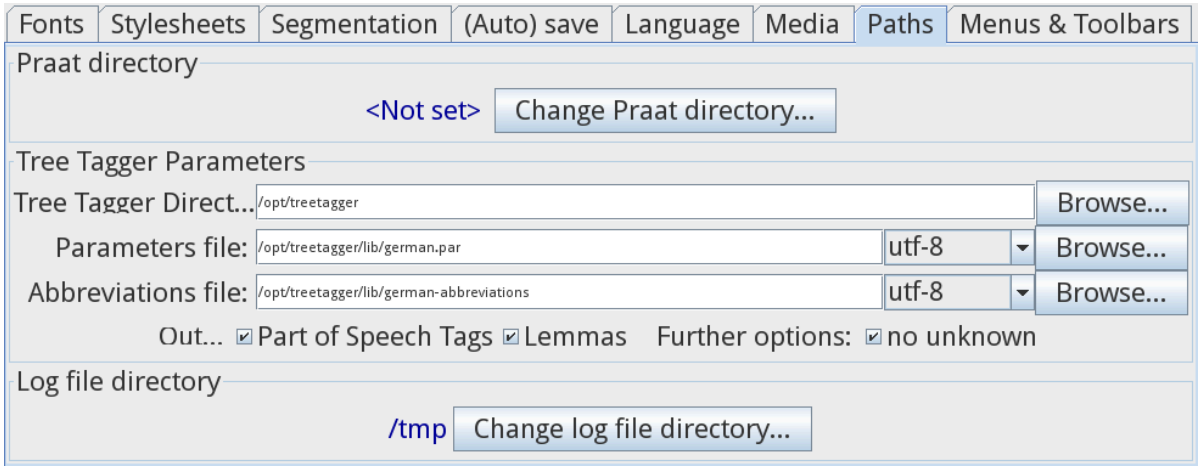

Abbildung 6
Einstellungen für den TreeTagger

Unbefriedigende Tokenisierungen lassen sich bei Bedarf manuell korrigieren. Der EXMARaLDA-Partitur-Editor stellt hierfür spezielle Funktionen zur Verfügung („Split“ und „Merge“ im „Event“-Menü oder die entsprechenden Buttons auf der Toolbar).

Als Nächstes ruft man das Transformationsszenario „Dulko: `S`, `pos`, and `lemma`-tiers (learner text)“ auf. Dieses Transformationsszenario annotiert die Ereignisse in der `word`-Spur mit Satzspannen, Wortklassen und Lemmata und legt dafür die Spuren `pos`, `lemma` und `S` an (vgl. Abbildung 7)[12].

[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	@card@	%	sein	.

Abbildung 7
Tokenisierter Lernertext mit Satzspannen, Wortklassen und Lemmata

Wortklassen-Tagging und Lemmatisierung erfolgen mit Hilfe einer lokalen TreeTagger-Installation. Dafür ist eine sprachspezifische Parameterdatei erforderlich, die von der TreeTagger-Homepage heruntergeladen werden kann. Wie bei der Abkürzungsliste für die Tokenisierung muss die Parameterdatei zur ersten Sprache passen, die in der Sprechertabelle als verwendete Sprache eingestellt ist. Im Fall der im Dulko-Template voreingestellten Sprache Deutsch ist dies die Parameterdatei `german.par`, die das Wortklassen-Tagset STTS (vgl. Schiller et al. 1999) verwendet. Der Pfad zur Parameterdatei ist ebenfalls in den TreeTagger-Einstellungen des EXMARaLDA-Partitur-Editors einzutragen. Es wird ein Fehler ausgegeben, falls der Pfad nicht existiert oder die Sprache der Parameterdatei nicht mit der ersten verwendeten Sprache in der Sprechertabelle übereinstimmt.

In der aktuellen Version 1.8 des EXMARaLDA-Partitur-Editors setzt die Satzspannen-Bestimmung noch voraus, dass satzbeendende Interpunktionszeichen mit dem STTS-Tag `$.` annotiert sind. Das EXMARaLDA-Quellcode-Repositorium enthält jedoch bereits eine generalisierte Version

[12] Einleitende Überschriften bleiben ohne Satzspanne, insofern die Überschrift in den Meta-Informationen als Wert des Attributs `text_title` eingetragen ist.

des entsprechenden XSLT-Stylesheets, die diese faktische Beschränkung auf das Deutsche aufhebt und die Teil des nächsten EXMARaLDA-Releases sein wird. Bei Bedarf können Satzspannen wie bei der Tokenisierung manuell korrigiert und mit Hilfe des Transformationsszenarios „Dulko: sentence spans“ neu durchnummeriert werden.

Handelt es sich bei dem zu annotierenden Text um einen Lernertext, so fügt man nach dessen Annotation eine oder mehrere Zielhypothesen hinzu, die auf analoge Weise annotiert werden. Für die erste Zielhypothese ist das Transformationsszenario „Dulko: `ZH` and `Fehler`-tiers (1st target hypothesis)“ vorgesehen, das eine Zielhypotesenspur (`ZH`) sowie leere Spuren für die Fehlerannotation erstellt (`FehlerOrth`, `FehlerMorph`, `FehlerSyn`, `FehlerLex` und `FehlerSem`). Außerdem kopiert es den Lernertext aus der `word`-Spur als Vorlage in die ZH-Spur, die dann manuell bearbeitet wird. Dabei können Funktionen des „Event“-Menüs („Split“, „Merge“, „Remove“) und des „Timeline“-Menü („Insert timeline item“) bzw. die entsprechenden Buttons auf der Toolbar genutzt werden.

Diese Zielhypothese wird ebenfalls mit Satzspannen, Wortklassen und Lemmata annotiert, wozu das Transformationsszenario „Dulko: `ZHS`, `ZHpos`, and `ZHlemma`-tiers (1st target hypothesis)“ aufgerufen wird. Dabei werden die Spuren `ZHS`, `ZHpos` und `ZHlemma` angelegt.

Abweichungen zwischen Tokens des Lernertexts in der `word`-Spur und Tokens der Zielhypothese in der `ZH`-Spur werden vom Transformationsszenario „Dulko: `ZHDiff`-tier (1st target hypothesis)“ berechnet und in einer `ZHDiff`-Spur als `CHA`, `SPLIT`, `MERGE`, `DEL`, `INS`, `MOVS` oder `MOVT` getaggt[13]. Insoweit die Abweichungen als Fehler und nicht als Folgefehler interpretiert werden, wird der linguistische Bereich des Fehlers manuell oder mit Hilfe des Dulko-Annotationspanels (vgl. Abschnitt 2.3) als Spanne mit einem Fehlertag in der entsprechenden Fehlerspur annotiert (vgl. Abbildung 8). Dabei erweisen sich wieder Funktionen des „Event“-Menüs als nützlich („Extend to the left“, „Extend to the right“ usw. bzw. die entsprechenden Toolbar-Buttons).

[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	@card@	%	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	@card@	%	sein	.
[FehlerOrth]						ZS														
[FehlerMorph]																				
[FehlerSyn]							StV													
[FehlerLex]													Gen							
[FehlerSem]																				

Abbildung 8
Erste Zielhypothese mit Abweichungen, Satzspannen, Wortklassen, Lemmata und Fehlerkategorien

Zur Repräsentation von einander überlappenden Fehlern kann es erforderlich sein, mehr als eine Zielhypothese zu annotieren. Im vorliegenden Beispiel gibt es im Lernertext bei *der Anzahl* zwei solche Fehler: einen Genus-Fehler, der in Abbildung 8 in der ersten Zielhypothese zu *die Anzahl* korrigiert und in der `FehlerLex`-Spur als Gen annotiert wurde, sowie einen Wortwahl-Fehler: Die erwähnten 50 % sind keine Anzahl. Würde man den zweiten Fehler in derselben Zielhypothese zu *der Anteil* korrigieren, so könnte der Genus-Fehler nicht annotiert werden, da es beim genusanzeigenden Artikel keine Abweichung mehr gäbe. Gemäß dem Dulko-Annotationsverfahren wird stattdessen mit Hilfe des Transformationsszenarios „Dulko: `ZH` and `Fehler`-tiers (2nd target hypothesis)“ eine zweite `ZH`-

[13] In Zweifelsfällen werden disjunktive Tags wie `MOVS/DEL` oder `MOVT/INS` ausgegeben, die vom Annotator manuell zu disambiguieren sind.

Spur mit entsprechenden `Fehler`-Spuren angelegt und der Inhalt der ersten `ZH`-Spur als Vorlage in die zweite ZH-Spur kopiert. In dieser Spur wird dann der Wortwahl-Fehler manuell korrigiert und in der `FehlerLex`-Spur mit dem Fehlertag `Lex` annotiert (vgl. Abbildung 9).

[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	@card@	%	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	@card@	%	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	@card@	%	sein	.
[FehlerOrth]																				
[FehlerMorph]																				
[FehlerSyn]																				
[FehlerLex]														Lex						
[FehlerSem]																				

Abbildung 9
Zweite Zielhypothese mit Abweichungen, Satzspannen, Wortklassen, Lemmata und Fehlerkategorien

Analog zur ersten, intermediären Zielhypothese wird die zweite, finale Zielhypothese außerdem mit Satzspannen, Wortklassen und Lemmata sowie mit Abweichungen zwischen der ersten und zweiten Zielhypothese annotiert[14]. Dafür werden die Transformationsszenarien „Dulko: `ZHS`, `ZHpos`, and `ZHlemma`-tiers (2nd target hypothesis)“ und „Dulko: `ZHDiff`-tier (2nd target hypothesis)“ auf die zweite Zielhypothese angewendet.

Bei der Annotation der beiden Zielhypothesen werden von den Transformationsszenarien intern dieselben XSLT-Stylesheets aufgerufen. Der Parameter `zh-number` im Transformationsdialog legt dabei fest, auf welche Zielhypothese sich das Transformationsszenario bezieht: Der Wert `1` steht für die erste Zielhypothese, der Wert `2` für die zweite Zielhypothese usw.[15] Transformationsszenarien, die sich auf den Lernertext in der `word`-Spur beziehen, setzen den `zh-number`-Parameter auf den Wert `0`.

Für besondere Anwendungsfälle existieren weitere Transformationsszenarien. Ist der zu annotierende Text eine Übersetzung, so kann mit Hilfe des Transformationsszenarios „Dulko: `trans`-tier (learner text)“ eine leere `trans`-Spur angelegt werden, die mit der `S`-Spur aligniert ist. In diese Spur wird der übersetzte Ausgangstext manuell eingetragen.

Zur Annotation von Selbstkorrekturen des Lerners verfährt man folgendermaßen. Zunächst legt man mit dem Transformationsszenario „Dulko: `orig`-tier (learner text)“ eine `orig`-Spur mit einer Kopie des Lernertexts aus der `word`-Spur an. Sodann editiert man die `orig`-Spur und macht

[14] Zur Unterscheidung zwischen *intermediären* und *finalen* Zielhypothesen vgl. Beeh et al. (2021: Abschnitt 2.2).
[15] Bei bestimmten Transformationsszenarien ist Parameter `zh-number` leer. In diesem Fall wird vom Transformationsszenario „Dulko: `ZH` and `Fehler`-tiers (additional target hypothesis)“ eine neue Zielhypothese angelegt und von den Transformationsszenarien „Dulko: `ZHS`, `ZHpos`, and `ZHlemma`-tiers (additional target hypothesis)“ und „Dulko: `ZHDiff`-tier (additional target hypothesis)“ die jeweils letzte Zielhypothese annotiert.

dort Selbstkorrekturen des Lerners rückgängig. Schließlich führt man das Transformationsszenario „Dulko: `Diff`-tier (learner text)“ aus, das Abweichungen zwischen Tokens der Originalfassung in der `orig`-Spur und Tokens der Endfassung in der `word`-Spur berechnet und in einer `Diff`-Spur mit Tags wie `CHA`, `DEL` oder `INS` annotiert (vgl. Abbildung 10).

[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	@card@	%	sein	.

Abbildung 10
Tokenisierter Lernertext mit Originalfassung, Abweichungen, Satzspannen, Wortklassen und Lemmata

Die `orig`-Spur kann man außerdem nutzen, um Leerraum, Absatzumbrüche, Zeilenumbrüche oder Trennstriche anzugeben. Zu diesem Zweck fügt man in dieser Spur jeweils ein Token mit einem der Symbole `_`, `¶`, `|` oder `-` ein; die `word`-Spur lässt man dabei unverändert[16]. Anschließend ruft man das Transformationsszenario „Dulko: `Layout`-tier (learner text)“ auf, das diese Tokens in einer `Layout`-Spur als `SPACE`, `PARB`, `LB` bzw. `HYPH` taggt.

Graphische Auszeichnungen des Lernertexts wie Unterstreichung oder durchgängige Großschreibung können manuell auf einer Graph-Spur annotiert werden, die mit dem Transformationsszenario „Dulko: Graph-tier (learner text)“ angelegt wird.

2.3 Das Dulko-Annotationspanel

Zur Unterstützung bei der Fehlerannotation kann der Annotator das Annotationspanel des EXMARaLDA-Partitur-Editors öffnen („Annotation panel“ im „View“-Menü) und die Dulko-Spezifikation dafür laden („Dulko: Annotation“ in der Drop-down-Liste; vgl. Abbildung 11[17]). Dieses Dulko-Annotationspanel enthält mehrere hierarchisch angeordnete Tagsets, die sich durch Mausklick auf den jeweiligen Reiter oder durch Mausklick in die gleichnamige Spur auswählen lassen. Durch Doppelklick auf einen Tag wird dieser an der aktuellen Position in die Partitur eingefügt.

[16] Für die Angabe eines Trennstrichs unterteilt man das zu trennende Wort auf der `orig`-Spur in drei Tokens: den ersten Teil des Worts, den Trennstrich und den zweiten Teil des Worts. Auf der `word`-Spur umfasst das ungetrennte Wort dann dementsprechend eine Spanne aus drei Ereignissen.

[17] Alternativ kann durch Mausklick auf „Open specification“ eine eigene Tagset-Spezifikation geladen werden.

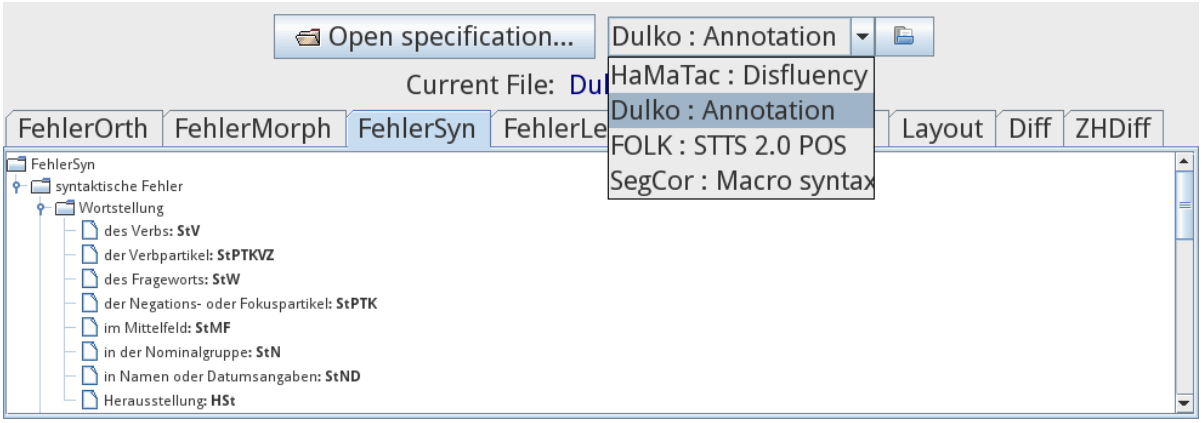

Abbildung 11
Annotationspanel mit Fehlerkategorien

3. Die Dulko-Tools des EXMARaLDA-Partitur-Editors: Hilfswerkzeuge

In diesem Abschnitt werden Hilfswerkzeuge aus den Dulko-Tools des EXMARaLDA-Partitur-Editors für annotierte Daten vorgestellt werden. Dazu zählen das Dulko-Formatierungsstylesheet für die Datenanzeige sowie die Dulko-Transformationsszenarien für die Datenaufbereitung.

3.1 Das Dulko-Formatierungsstylesheet

Wie aus Abbildung 9 ersichtlich ist, enthält der Output der Dulko-Transformationsszenarien neben der linguistischen Annotation auch Formatierungsangaben: `word` und `ZH`-Spuren sind fett gesetzt, `ZHDiff`-Spuren grün, `Fehler`-Spuren rot etc. Diese Formatierungsangaben gehen bei der Speicherung verloren, sofern nicht explizit angegeben wird, dass die Formatierungsangaben mit gespeichert werden sollen („Save as“ im „File“-Menü, „Save formats“). Wenn man eine solche EXB-Datei ohne Formatierungsangaben öffnet, kann die Formatierung wiederhergestellt werden, indem man das Dulko-Formatierungsstylesheet darauf anwendet („Built-in stylesheets“ im „Format“-Menü, „Dulko“).

3.2 Dulko-Transformationsszenarien für die Datenaufbereitung

Neben Transformationsszenarien für die Annotation stellen die Dulko-Tools auch Transformationsszenarien für verschiedene Aufgaben der Datenaufbereitung zur Verfügung. Dazu zählen insbesondere die Optimierung annotierter Daten und deren Export in verschiedene Formate.

In Abschnitt 2.1 wurde empfohlen, für die Annotation von Lernerdaten zunächst das Dulko-Template zu laden. Das Dulko-Template kann aber auch nachträglich angewendet werden, indem man das Transformationsszenario „Dulko: metadata“ ausführt. Erwähnt wurde dort auch, dass man in der Sprechertabelle eine Abkürzung als Sprecher-ID eintragen kann. Von den in Abschnitt 2.2 genannten Transformationsszenarien wird der Anfang dieser Sprecher-ID als Teil des angezeigten Spurennamens übernommen. Diese Anzeige kann ebenfalls nachträglich geschehen durch Aufruf des Transformationsszenarios „Dulko: tier names“. Zwei weitere Hilfswerkzeuge zur Datenoptimierung sind das Transformationsszenario „Dulko: sentence spans“, mit dessen Hilfe sich manuell korrigierte Satzspannen neu durchnummerieren lassen, und das Transformationsszenario „Dulko: timeline“, das

überzählige Zeitpunkte entfernt, zu denen es kein Ereignis in der `word`-Spur, einer eventuellen `orig`-Spur oder einer `ZH`-Spur gibt.

Für den Export annotierter Daten stellen die Dulko-Tools die folgenden Transformationsszenarios zur Verfügung. Die Transformationsszenarien „Dulko: text (learner text)“, „Dulko: text (1st target hypothesis)“, „Dulko: text (2nd target hypothesis)“ und „Dulko: text (last target hypothesis)“ exportieren den Text in der `word`-Spur bzw. der jeweiligen `ZH`-Spur. Das Transformationsszenario „Dulko: HTML version“ gibt Annotation und Metadaten als HTML-Datei aus, die sich insbesondere für die Qualitätskontrolle eignet, da der annotierte Lernertext dort in übersichtlicher Weise satzspannenweise umgebrochen ist. Die Transformationsszenarien „Dulko: ANNIS-compatible version“ und „Dulko: Pepper-compatible metadata list“ schließlich bereiten Annotation und Metadaten so auf, dass sie mittels Pepper[18] in das Format des Korpusrecherchesystems ANNIS[19] konvertiert werden können. Dieser Konvertierungsprozess kann mit Hilfe des Korpusbausystems *makeDulko* automatisiert werden, das der Entwickler der Dulko-Tools ebenfalls als Open Source verfügbar gemacht hat[20].

Literatur und Ressourcen

Beeh, Christoph / Drewnowska-Vargáné, Ewa / Kappel, Péter / Modrián-Horváth, Bernadett / Nolda, Andreas / Rauzs, Orsolya / Scheibl, György (2021): *Dulko-Handbuch: Aufbau und Annotationsverfahren des deutsch-ungarischen Lernerkorpus. Version 1.0*. Szeged: Institut für Germanistik. https://doi.org/10.14232/dulko-handbuch-v1.0.

Granger, Sylviane / Paquot, Magali (2017): *Core metadata for learner corpora. Draft 1.0*. Bozen: Eurac Research CLARIN Centre. http://hdl.handle.net/20.500.12124/61 (18.06.2024).

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

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

Nolda, Andreas (2023): Fehlerannotation und Fehleranalyse am Beispiel des deutsch-ungarischen Lernerkorpus Dulko. In: Auteri, Laura / Barrale, Natascia / Di Bella, Arianna / Hoffmann, Sabine (Hrsg.): *Jahrbuch für internationale Germanistik: Beihefte*. Bd. 10. Bern: Peter Lang, 747–755. https://www.peter-lang.com/document/1277913 (18.06.2024).

Reznicek, Marc / Lüdeling, Anke / Krummes, Cedric / Schwantuschke, Franziska / Walter, Maik / Schmidt, Karin / Hirschmann, Hagen / Andreas, Torsten (2012): *Das Falko-Handbuch: Korpusaufbau und Annotationen. Version 2.01*. Berlin: Humboldt-Universität zu Berlin. https://www.linguistik.hu-berlin.de/de/institut/professuren/korpuslinguistik/forschung/falko/FalkoHandbuchV2 (18.06.2024).

Schiller, Anne / Teufel, Simone / Stöckert, Christine / Thielen, Christine (1999): *Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset)*. Stuttgart: Universität Stuttgart, Institut für maschinelle Sprachverarbeitung / Tübingen: Universität Tübingen, Seminar für Sprachwissenschaft. https://www.ims.uni-stuttgart.de/documents/ressourcen/lexika/tagsets/stts-1999.pdf (18.06.2024).

[18] https://corpus-tools.org/pepper/ (18.06.2024).
[19] https://corpus-tools.org/annis/ (18.06.2024).
[20] https://sr.ht/~nolda/makedulko/ (18.06.2024).

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

Schmidt, Thomas / Wörner, Kai (2014): EXMARaLDA. In: Durand, Jacques / Gut, Ulrike / Kristoffersen, Gjert (eds.): *The Oxford Handbook of Corpus Phonology*. Oxford: Oxford University Press, 402-419.

Biographische Notiz: Andreas Nolda promovierte 2005 an der Freien Universität Berlin und habilitierte sich 2013 in den Fächern allgemeine Sprachwissenschaft und germanistische Linguistik an der Humboldt-Universität zu Berlin. Von 2013 bis 2019 war er DAAD-Lektor am Lehrstuhl für germanistische Linguistik des Instituts für Germanistik der Universität Szeged und baute dort das Dulko-Lernerkorpusprojekt mit auf. Seit 2019 ist er wissenschaftlicher Mitarbeiter im Korpusbereich am Zentrum für digitale Lexikographie der deutschen Sprache in Berlin.

Kontaktanschrift:
Dr. Andreas Nolda
Berlin-Brandenburgische Akademie der Wissenschaften
Jägerstraße 22/23
10117 Berlin
andreas@nolda.org