

Softwarelokalisierung und Kontext

Zur Rolle von Kontext und Referenzmaterial
bei der Lokalisierung von GUIs für
Anwendungsprogramme

Inauguraldissertation
zur Erlangung des Akademischen Grades
eines Dr. phil.,

vorgelegt dem Fachbereich Translations-, Sprach- und
Kulturwissenschaft (06) der Johannes Gutenberg-Universität
Mainz

von

Marcus Wiedmann
aus Memmingen

Germersheim 2025

Referentin: Univ.-Prof. Dr. Silvia Hansen-Schirra
Korreferent: PD Dr. Moritz Schaeffer

Tag des Prüfungskolloquiums: 18.12.2025
Veröffentlicht unter Verwendung der Lizenz CC-BY-4.0,
<https://creativecommons.org/licenses/by/4.0/deed.de>

Inhaltsverzeichnis

1	Einleitung	1
1.1	Schwerpunkte der Arbeit	3
1.2	Struktur der Arbeit	4
2	Softwarelokalisierung und GUI-Text	7
2.1	GUI-Text	7
2.1.1	Software	7
2.1.2	GILT, Lokalisierung und Softwarelokalisierung	8
2.1.3	Grafische Benutzeroberflächen (GUIs)	11
2.1.4	Lokalisierung von GUIs	12
2.2	Besonderheiten von GUI-Text	16
2.2.1	Steuerzeichen mit Bezug zur Programmfunktionalität	17
2.2.2	Steuerzeichen mit Bezug zur Formatierung	20
2.2.3	Zeichenbasierte Konventionen	21
2.3	Freie Software	22
3	Text, Fachtext, Kontext	25
3.1	Text	25
3.2	Textsorte	30
3.3	Fachtext	32
3.4	Kontext	34
3.4.1	Kontext und Bedeutung	36
3.4.2	Außersprachlicher Kontext	36
3.4.3	Sprachlicher Kontext, Kotext	39
3.4.4	Bühler und Coseriu	41
3.4.5	Ko-Text, Chron-Text, Rel-Text, Bi-Text und Nicht-Text	43
3.4.6	Kontext im multimodalen Text	46
3.4.7	Kontext im Fachtext	48
3.4.8	Zwischenfazit zum Kontext	49

4	GUI-Text: Multimodaler Fachtext, Kontext und Referenzmaterial	53
4.1	GUI-Text als multimodaler Text	53
4.2	GUI-Text als Fachtextsorte	59
4.3	GUI-Text und Programmfunktionen	62
4.4	GUI-Text und Kontext	64
4.5	GUI-Text und Referenzmaterial	72
4.6	Kontext und Probleme bei der Übersetzung von GUI-Text	77
4.6.1	Situativer Kontext und Probleme bei der Übersetzung von GUI-Text	77
4.6.2	Fachlicher Kontext und Probleme bei der Übersetzung von GUI-Text	81
4.6.3	Funktionaler Kontext und Probleme bei der Übersetzung von GUI-Text	83
4.7	Einschränkungen und offene Fragen	88
5	Korpuslinguistische Untersuchung von GUI-Text	91
5.1	Korpuslinguistik als Methodik	91
5.2	Kriterien für die Korpuserstellung	93
5.3	Auswahl der Textproben	97
5.3.1	Die Softwareprodukte im Korpus	99
5.3.2	Volumen der Softwareprodukte im Korpus	100
5.4	Vorbereitung des Korpus	101
5.5	Annotationsschema und Features	102
5.5.1	Situativer Kontext	106
5.5.1.1	PROBLEM_WORTART	108
5.5.1.2	PROBLEM_GUI-BEZUG	112
5.5.1.3	PROBLEM_FLEXIONSENDUNG	113
5.5.2	Fachlicher Kontext	115
5.5.2.1	Fachterminologie	118
5.5.2.2	FACHTERM_ANEINANDERREIHUNG	120
5.5.3	Funktionaler Kontext	123
5.5.3.1	UNKLARE_ANEINANDERREIHUNG	123
5.5.3.2	homonym_it-term	127
5.5.3.3	metaphorik	127
5.5.3.4	unklare_bedeutung	128
5.5.4	Weitere ergänzende Annotationen	128
5.5.4.1	LÄNGE	129
5.5.4.2	STEUERZEICHEN	129
5.6	Auswertung der Korpusstudie	131
5.6.1	Grundlegende Auswertung	132

5.6.2	Auswertung der Features des situativen Kontexts . . .	137
5.6.3	Auswertung der Features des fachlichen Kontexts . . .	139
5.6.4	Auswertung der Features des funktionalen Kontexts . .	140
5.6.5	Inter-Annotator-Agreement	140
5.7	Interpretation der Ergebnisse	144
5.8	Einschränkungen, offene Fragen und weitere Vorgehensweise .	154
6	Recherche und Übersetzungsalternativen bei der Übersetzung der Kontextarten	159
6.1	Hypothesen des Übersetzungstests	159
6.1.1	Hypothesen zum Referenzmaterial	160
6.1.2	Hypothese zur Anzahl der Übersetzungsalternativen . .	161
6.2	Konzeption des Übersetzungstests	162
6.3	Durchführung des Übersetzungstests und Probandinnen	165
6.4	Auswertung des Übersetzungstests	165
6.4.1	Vorbereitungen für die Auswertung	166
6.4.2	Ausgewertete Strings im Übersetzungstest	166
6.4.3	Auswertung hinsichtlich der Recherche	167
6.4.4	Auswertung hinsichtlich der Anzahl der Übersetzungsalternativen	170
6.5	Interpretation der Ergebnisse des Übersetzungstests	174
6.5.1	Recherche	175
6.5.2	Übersetzungsalternativen	176
6.5.3	Kontextkategorien	179
6.6	Einschränkungen und offene Fragen des Übersetzungstests . .	184
7	Recherche und kognitiver Aufwand bei der Übersetzung der Kontextarten	187
7.1	Hypothesen der Eye-Tracking- und Key-Logging-Studie	188
7.1.1	Hypothese zum Referenzmaterial	188
7.1.2	Hypothese zum kognitiven Aufwand	189
7.2	Konzeption der Eye-Tracking- und Key-Logging-Studie	195
7.3	Geplanter Ablauf und Zusatzmaterial	198
7.4	Probandinnen	201
7.5	Durchführung der Studie	205
7.6	Auswertung der Studie	206
7.6.1	Vorbereitung der erhobenen Daten	206
7.6.2	Nutzen der Zusatzmaterialien	208
7.6.3	Eye-Tracking und Key-Logging	217
7.7	Interpretation der Ergebnisse der Eye-Tracking- und Key-Logging-Studie	224

7.7.1	Zusatzmaterialien und Recherche	225
7.7.2	Eye-Tracking, Key-Logging und kognitiver Aufwand	234
7.8	Einschränkungen und Probleme	238
8	Fazit und Ausblick	241
8.1	Ergebnisse zu Recherche, Übersetzungsalternativen/Entropie und kognitivem Aufwand	242
8.2	Profile der Kontextkategorien	243
8.3	Forschungsdesiderata	244
	Bibliographie	247
	Anhänge	258
A	Anhang zur Korpuserstellung	259
A.1	Audacity	259
A.2	Avidemux	261
A.3	BlueGriffon	264
A.4	Claws Mail	266
A.5	Double Commander	267
A.6	GIMP	269
A.7	LibreOffice	271
A.8	Mozilla Firefox	272
A.9	Notepad++	273
A.10	VLC Media Player	275
B	Anhang zum Übersetzungstest	277
B.1	Anweisungsdatei für den Übersetzungstest	277
B.2	Excel-Tabelle mit dem eigentlichen Test	280
B.3	Kontextarten der Strings im Übersetzungstest	284
B.4	Änderungen vor Beginn der Auswertung	288
C	Anhang zur Eye-Tracking- und Key-Logging-Studie	293
C.1	Items aus der Eye-Tracking- und Key-Logging-Studie	293
C.2	Anweisungsdatei für die Eye-Tracking- und Key-Logging-Studie	296

Abbildungsverzeichnis

2.1	Notepad++ Version 7.6.4: Menü File und Ausschnitt der Datei <code>english.xml</code>	16
2.2	Access-Keys und Shortcuts in Audacity Version 2.2.2	18
3.1	Kontextschema nach Melby und Foster (vgl. Melby und Foster 2010, S. 5)	44
4.1	Dialogfeld Preferences , Bereich Devices in Audacity Version 2.2.2	56
4.2	Dialogfeld Preferences , Bereich Import / Export in Audacity Version 2.2.2	57
4.3	Ausschnitt aus der Datei <code>browser.dtd</code> von Mozilla Firefox Version 6.6.0.3	71
4.4	Ausschnitt aus der Datei <code>audacity.pot</code> von Audacity Version 2.2.2	72
4.5	Kontinuum der Kontextinformationen beim Übersetzen von GUI-Text	77
4.6	Dialogfeld Find files in Double Commander Version 0.9.2	79
4.7	Dialogfeld Compare files in Double Commander Version 0.9.2	80
5.1	Annotationsschema des Korpus	105
5.2	Entscheidungsbaum für den situativen Kontext	107
5.3	Entscheidungsbaum für den fachlichen Kontext	116
5.4	Entscheidungsbaum für den funktionalen Kontext	124
5.5	Strings und Annotationen nach Programmtyp	132
5.6	Prozentsatz der Strings mit Annotationen und der Annotationen gegenüber der Anzahl der Strings	133
5.7	Annotationen nach Kontextarten	134
5.8	Annotationen nach Kontextarten anteilig	135
5.9	Länge der annotierten Strings	135
5.10	Steuerzeichen in annotierten Strings	136
5.11	Verteilung der Features der Kategorie „Situativer Kontext“	138

5.12	Verteilung der Features der Kategorie „Fachlicher Kontext“	139
5.13	Verteilung der Features der Kategorie „Funktionaler Kontext“	141
6.1	Wurde eine Recherche durchgeführt?	168
6.2	War die Recherche erfolgreich?	169
6.3	Anzahl der spontanen Übersetzungsvorschläge pro Probandin	170
6.4	Ist der erste spontane Übersetzungsvorschlag identisch mit der endgültigen Übersetzung?	171
6.5	Sind mindestens 10 identische Vorkommen eines Übersetzungsvorschlags vorhanden?	171
6.6	Anzahl der Übersetzungsalternativen pro String, auf die sich die endgültigen Übersetzungsvorschläge verteilen	172
6.7	Anzahl der Übersetzungsalternativen aller Probandinnen bei den spontanen Übersetzungsvorschlägen	173
6.8	Anzahl der Übersetzungsalternativen aller Probandinnen bei den endgültigen Übersetzungsvorschlägen	174
6.9	Anzahl der Übersetzungsalternativen aller Probandinnen, die bei den endgültigen Übersetzungsvorschlägen neu hinzugekommen sind	175
7.1	Projekt 1 in Translog II	196
7.2	Bearbeitungsdauer nach Teilprojekten	197
7.3	Screenshot des Strings „First difference“, Markierung mit rotem Pfeil	199
7.4	Textliche Beschreibung des Strings „First difference“	200
7.5	Was ist Ihr höchster Bildungsabschluss? Was sind Sie im Moment hauptberuflich?	202
7.6	Wie gut übersetzen/lokalisieren Sie von Ihrer ersten Fremdsprache in Ihre Muttersprache? Von 1 (sehr schlecht) bis 100 (sehr gut)	203
7.7	Bearbeitungsdauer nach Probandinnen	204
7.8	Benötigen Sie für die korrekte Übersetzung mehr Informationen?	209
7.9	Waren diese Informationen für die korrekte Übersetzung hilfreich?	210
7.10	Sind die Antworten auf beide Fragen identisch?	211
7.11	Welche Kombination von Antworten wurde bei den verschiedenen Kontextarten gegeben?	212
7.12	Welche Antwortkombinationen wurden bei den verschiedenen Kontextarten für die unterschiedlichen Materialien gegeben?	213
7.13	Dauer zwischen Beantwortung von Frage 1 und Frage 2	214

7.14	Dauer und Beantwortung der Fragen nach Kontextarten und Materialien	215
7.15	Anzahl der manuell eingefügten bzw. gelöschten Zeichen	216
7.16	Bearbeitungsdauer der Segmente nach Kontextarten	219
7.17	Anzahl der Fixationen und Gesamtleседauer für AT- und ZT-Tokens	222
7.18	Dauer der ersten Fixation („FFDur“) und Verweildauer („FPDurS“) für AT-Tokens	223
7.19	Word Translation Entropy: „HTra“ und „NHTra“	224

Tabellenverzeichnis

5.1	Softwareprodukte und Volumen	101
6.1	Programmtypen und Kontextarten	167
7.1	Anzahl der manuell eingefügten bzw. gelöschten Zeichen	217
7.2	Bearbeitungsdauer der Segmente nach Kontextarten	219
7.3	Anzahl der Fixationen und Gesamtlesedauer	220
7.4	Dauer der ersten Fixation („FFDur“) und Verweildauer („FPDurS“) für AT-Tokens	223
7.5	Word Translation Entropy: „HTra“ und „NHTra“	225

Kapitel 1

Einleitung

Betrachtet man die Literatur zur Softwarelokalisierung, stellt man fest, dass nur sehr wenige Monografien (beispielsweise Astapenko 2008, Behrens 2016, Beste 2006, K. J. Dunne 2006a und Roturier 2015)¹ und ein paar Sammelbände (beispielsweise Esselink 2000, K.-D. Schmitz und Wahle 2000 sowie Reineke und K.-D. Schmitz 2005) vorliegen, die überdies weitgehend schon älteren Datums sind. Zudem fokussiert sich Behrens in der jüngsten dieser Monografien auf „solche Übersetzungsprobleme, die unmittelbar auf Barrieren des technologischen Umfelds zurückgeführt werden können“ (Behrens 2016, S. 39), bezieht also nur einen konkreten Teil der üblicherweise genannten Probleme bei der Softwarelokalisierung mit ein. Darüber hinaus finden sich nur Monografien aus verwandten Bereichen wie Weblokalisierung (Jiménez-Crespo 2013) und Projektmanagement bei der Softwarelokalisierung und Übersetzung (K. J. Dunne und E. S. Dunne 2011). Auch an einzelnen Artikeln gibt es nicht gerade viele aus diesem Bereich. Auch Behrens sieht dies so:

Ausführlich haben sich translationsnahe Fachperiodika wie *Multilingual, technische kommunikation* und *MDÜ* mit der Internationalisierung und Lokalisierung digitaler Medien auseinandergesetzt, dies zumindest bis in die ersten Jahre nach der Jahrtausendwende. [...] Schon bald aber schien das Thema erschöpft, und die Beiträge wurden seltener.

Dagegen gibt es beispielsweise im verwandten Bereich der Lokalisierung von Videospielen eine ganze Reihe auch jüngerer Monografien (wie Bernal-Merino 2015, Kim 2019, Mejías-Climent 2021, O’Hagan und Mangiron 2013,

¹Ergänzend befasst sich Jiménez-Crespo in seiner Monografie mit dem größeren Bereich der Lokalisierung, von dem die Softwarelokalisierung nur einen Teilbereich darstellt (Jiménez-Crespo 2024).

Pettini 2021 und Sanz López 2013), die sich mit verschiedenen Aspekten dieses Bereichs befassen, sowie auch eine Vielzahl von Artikeln.

Zudem kommen in den Veröffentlichungen zur Softwarelokalisierung selbst an vielen Stellen Praktikerinnen² zu Wort, die häufig eine exemplarische Herangehensweise wählen, um verschiedene Aspekte und Problembereiche der Softwarelokalisierung und des Lokalisierungsprozesses zu schildern (wie beispielsweise Esselink 2000 und Roturier 2015 oder viele der Beiträge in K.-D. Schmitz und Wahle 2000 sowie in Reineke und K.-D. Schmitz 2005), und auch die akademischen Veröffentlichung tragen wenig zur empirischen Erforschung der beteiligten Textsorten und der häufig genannten Problembereiche bei.³

Aus der geringen Zahl an Veröffentlichungen in den letzten Jahren lässt sich schließen, dass sich viele Aspekte der Softwarelokalisierung aus Praktikerinnensicht in den letzten Jahren und vielleicht sogar Jahrzehnten nicht in relevantem Maß geändert haben.⁴ Gleichzeitig wird aber deutlich, dass gerade auch im Hinblick auf die Relevanz von Lokalisierung allgemein und Softwarelokalisierung im Speziellen ein eklatanter Mangel an empirischer Beschäftigung mit diesem Themengebiet vorliegt, und das obwohl die Lokalisierung und dabei wiederum die Softwarelokalisierung einen großen Teil der professionellen Translation darstellt (vgl. Drewer und Ziegler 2014, S. 41), wobei Englisch gemeinhin als häufigste Ausgangssprache betrachtet wird (vgl. beispielsweise Drewer und Ziegler 2014, S. 50 oder für die Vergangenheit Esselink 2003, S. 84).⁵

Eine solche empirische Untersuchung einer der in der Softwarelokalisierung zentralen Textsorten, der Texte aus grafischen Benutzeroberflächen (GUIs) von Softwareprodukten, ist das Ziel der vorliegenden Arbeit. Konkret soll deren Bezug zu Kontext und Referenzmaterial, also zu zwei in der professionellen Praxis häufig als relevant genannten Aspekten dieser Textsorte, genauer betrachtet werden.

²Hier und im weiteren Verlauf dieser Arbeit wird ausschließlich die feminine Form verwendet. Es sind jedoch grundsätzlich alle Geschlechter gemeint.

³Vgl. dazu auch Jiménez-Crespo, der zahlreiche Forschungsfragen im Bereich der Lokalisierung aufführt, die noch untersucht werden können (Jiménez-Crespo 2024, S. 278–279).

⁴Als Hinweis dafür kann auch gewertet werden, dass viele der in diesen Veröffentlichungen genannten Aspekte auch bei den in der vorliegenden Arbeit untersuchten Softwareprodukten auffindbar waren, wie sich in den folgenden Kapiteln noch zeigen wird.

⁵Behrens stellt Letzteres allerdings in Frage, ohne dabei jedoch Gegenargumente anzuführen (vgl. Behrens 2016, S. 28).

1.1 Schwerpunkte der Arbeit

Im Fokus dieser Arbeit liegen zwei Bereiche. Der erste ist das Problem des Kontexts bzw. des fehlenden Kontexts, das in der Literatur zur Softwarelokalisierung an vielen Stellen thematisiert wird. Textelemente in grafischen Benutzeroberflächen von Softwareprodukten unterscheiden sich von anderen Texten, weil es darin keine Abfolge von Sätzen, Absätzen und Abschnitten gibt. In GUIs finden sich dagegen viele jeweils eigenständige Textelemente, die häufig aus nur einem oder wenigen Wörtern und nur in seltenen Fällen aus vollständigen Sätzen bestehen. Beispielfähig können dafür die Textelemente in Menüs oder Dialogfeldern genannt werden. Für die Übersetzung solcher Textelemente werden häufig zusätzliche Kontextinformationen als unabdingbar angesehen. Diese Vorstellung des fehlenden Kontexts in Bezug auf die Lokalisierung von GUIs soll in der vorliegenden Arbeit so konkretisiert werden, dass sie operationalisierbar und damit empirisch untersuchbar wird, so dass anschließend weitere Fragestellungen in diesem Zusammenhang untersucht werden können, wie Fragen der Entropie oder des kognitiven Aufwands.⁶

Der zweite Bereich ist die Frage des für die Recherche bei der Lokalisierung von GUIs genutzten Referenzmaterials. In der Literatur zur Softwarelokalisierung werden verschiedene notwendige bzw. wünschenswerte Referenzmaterialien genannt. In der vorliegenden Arbeit soll untersucht werden, ob sich diese Materialien auch verschiedenen Aspekten der Kontextfrage zuordnen lassen und sich für unterschiedliche Übersetzungsprobleme aus dem Fokusbereich des Kontexts in unterschiedlichem Maße eignen.

Für die Beantwortung dieser Fragen sind zunächst eine theoretische Herangehensweise und im Anschluss verschiedene aufeinander aufbauende empirische Studien erforderlich. Als Methoden dieser Studien wurden Korpuslinguistik zur Überprüfung des Ausgangstexts und anschließend ein Übersetzungstest zur Ermittlung der Anzahl von Übersetzungsalternativen und für erste Einblicke in Recherche bzw. Referenzmaterialien sowie eine Eye-Tracking- und Key-Logging-Studie gewählt, die auf dieser Basis Rückschlüsse im Hinblick auf den kognitiven Aufwand sowie auf hilfreiche Referenzmaterialien ermöglicht. Die genaue Struktur der vorliegenden Arbeit wird im nächsten Abschnitt beschrieben.

⁶Von Jiménez-Crespo werden als einer der zahlreichen noch zu untersuchenden translationswissenschaftlichen Forschungsbereiche in der Lokalisierung explizit kognitive Übersetzungsprozesse genannt (vgl. Jiménez-Crespo 2024, S. 279).

1.2 Struktur der Arbeit

Im Rahmen der Arbeit wird zunächst in Kapitel 2 der Bereich der Softwarelokalisierung allgemein und speziell der Untersuchungsgegenstand der vorliegenden Arbeit, GUI-Text⁷, beschrieben. In diesem Zusammenhang werden Informationen zu Software, GILT, GUIs sowie zur Lokalisierung von GUIs gegeben und anschließend Besonderheiten von GUI-Text beschrieben, die sich in den zu übersetzenden Strings in Form von zusätzlichen Zeichen für Programmfunktionalität oder Formatierung sowie als zeichenbasierte Konventionen zeigen. Danach wird der Themenbereich freie Software erläutert.

In Kapitel 3 stehen allgemeine Definitionen im Fokus. Zunächst sind dies Text, Textsorte und Fachtext. Anschließend wird umfassender das Gebiet Kontext beleuchtet, das im weiteren Verlauf der Arbeit speziell für GUI-Text untersucht werden soll, wobei verschiedene Aspekte des Begriffs *Kontext* betrachtet werden, wie dessen Beitrag zur Bedeutung, außersprachlicher und sprachlicher Kontext sowie die Frage des Kontexts in multimodalen Texten und in Fachtexten.

In Kapitel 4 werden die Bezüge zwischen den im vorangehenden Kapitel vorgestellten Definitionen und dem Themenbereich der vorliegenden Arbeit, GUI-Text, hergestellt. Zunächst wird GUI-Text dazu als multimodale Fachtextsorte identifiziert sowie deren Bezug zu Programmfunktionen dargestellt. Anschließend werden auf dieser Basis verschiedene Kontextaspekte von GUI-Text herausgearbeitet und diese in Bezug zu Referenzmaterialien bei der Lokalisierung von GUI-Text gesetzt. Zuletzt werden spezielle Probleme bei der Übersetzung von GUI-Text mit den drei identifizierten Kontextarten in Bezug gesetzt.

Auf dieser Grundlage folgen die empirischen Teile dieser Arbeit. Dies ist zunächst in Kapitel 5 eine Korpusstudie, in der auf Basis der vorangehenden Kapitel versucht wird, operationalisierbare Definitionen für das Problem des fehlenden Kontexts in der Softwarelokalisierung anhand eines Korpus zu identifizieren und anschließend in einer Annotation nicht nur qualitativ, sondern auch quantitativ zu erfassen, auszuwerten und zu interpretieren.

Im anschließenden Kapitel 6 wird eine Studie in Form eines Übersetzungstests mit Studierenden im Master Translationswissenschaft am Fachbereich 06, Translations-, Sprach- und Kulturwissenschaft, der Johannes Gutenberg-Universität Mainz beschrieben, deren Daten im Anschluss ausgewertet und interpretiert werden. Bei dieser Studie stand neben den Fragen des Kontexts und der Übersetzungsalternativen auch die Frage der Recherche im Fokus.

Zuletzt wird in Kapitel 7 eine weitere Studie beschrieben, die wieder-

⁷GUI steht hierbei für Graphical User Interface, also die grafische Benutzeroberfläche

um auf der Basis des Übersetzungstests konzipiert wurde. Bei dieser Studie handelt es sich um eine Eye-Tracking- und Key-Logging-Studie, mit deren Hilfe einerseits die Frage der Recherche und hilfreicher Referenzmaterialien und andererseits auf der Basis der Ergebnisse im Zusammenhang mit den Übersetzungsalternativen aus dem Übersetzungstest die Frage des kognitiven Aufwands untersucht wurde.

Im abschließenden Kapitel 8 werden die Ergebnisse der Arbeit zusammengefasst und wird ein Ausblick auf mögliche weiterführende Forschungsarbeiten gegeben.

Kapitel 2

Softwarelokalisierung und GUI-Text

Bevor die Rolle des Kontexts bei der Übersetzung von Benutzeroberflächen-texten in Anwendungsprogrammen empirisch untersucht werden kann, müssen einige Grundfragen diskutiert werden. In diesem Kapitel finden sich dazu zunächst Informationen zur Softwarelokalisierung allgemein und zu GUI-Text im Speziellen (Abschnitt 2.1). Im Anschluss daran werden alle technisch-textlichen Besonderheiten von GUI-Text aufgeführt, die im später untersuchten Korpus enthalten sind (Abschnitt 2.2). Zuletzt finden sich einige Informationen zum Themenbereich freie Software, aus dem die untersuchten GUI-Textauszüge stammen (Abschnitt 2.3).

2.1 GUI-Text

In diesem Abschnitt sollen folgende Aspekte aus dem Bereich Softwarelokalisierung thematisiert werden, die anschließend eine detailliertere Diskussion der Probleme bei der Übersetzung von GUI-Text ermöglichen:

- Software und dabei insbesondere Anwendungs- und Unterstützungssoftware: Abschnitt 2.1.1
- Softwarelokalisierung: Abschnitt 2.1.2
- Grafische Benutzeroberflächen: Abschnitt 2.1.3
- Lokalisierung von GUIs: Abschnitt 2.1.4

2.1.1 Software

Im Computerbereich wird zwischen Hardware und Software unterschieden. Von Gumm und Sommer wird *Hardware* als „Oberbegriff für alle materiellen

Komponenten eines Computersystems“ (Gumm und Sommer 2016, S. 39) bezeichnet. Im Unterschied dazu bezeichnen sie als *Software* die Programme, „die einen sinnvollen Betrieb eines Rechners erst möglich machen“ (Gumm und Sommer 2016, S. 63). Dabei teilen sie die Software in verschiedene Schichten auf, nämlich in das Betriebssystem und in das grafische Bediensystem (vgl. Gumm und Sommer 2016, S. 65). Beide dienen als Mittler zwischen Hardware und Anwenderinnen, die mit Anwendungsprogrammen, wie beispielsweise Textverarbeitungssystemen, Browsern und Spielen, arbeiten (vgl. Gumm und Sommer 2016, S. 2, 65).

Beste hingegen unterscheidet Software in Systemsoftware und Anwendungssoftware. Erstere setzt sich zusammen aus Betriebssystemsoftware und Dienstprogrammen, die für das Funktionieren eines Computers erforderlich sind, zu letzterer gehören Programme zur Ausführung bestimmter Aufgaben, wie Textverarbeitung, Tabellenkalkulation oder Datenbanken (vgl. Beste 2006, S. 25).

Im Eintrag „Software“ der Wikipedia wird neben System- und Anwendungssoftware noch eine weitere Kategorie von Software genannt:

Unterstützungssoftware, d. h. Programme, die bei der Entwicklung [und] Wartung helfen oder eine nicht-anwendungsspezifische Leistung erbringen (z. B. Editoren, Compiler, Virens Scanner, Datenbankmanagementsysteme, ...) (*Software* 2025)

Die Unterstützungssoftware bildet somit einen Teil der von Beste genannten Kategorie Systemsoftware. Für die Korpusuntersuchung (s. Kapitel 5) im Rahmen der vorliegenden Arbeit wird Software aus den Kategorien Anwendungssoftware laut Bestes Definition und Unterstützungssoftware laut Wikipedia-Definition genutzt.¹

2.1.2 GILT, Lokalisierung und Softwarelokalisierung

Die *Lokalisierung* wird als einer der Schritte im Rahmen der Globalisierungsstrategie von Softwareunternehmen betrachtet (vgl. z. B. K.-D. Schmitz 2005a, S. 1, Esselink 2000, S. 4, und Beste 2006, S. 27). Häufig findet sich als Überbegriff das Akronym GILT, das für die englischen Begriffe *Globalization*, *Internationalization*, *Localization* und *Translation* steht (vgl. Drewer und Ziegler 2014, S. 30). Die Globalisierung ist zunächst generell der Prozess, durch den die Weltwirtschaft zum globalen Markt wird (vgl. Drewer und Ziegler 2014, S. 30). Im Zusammenhang mit der GILT-Branche liegt

¹Ein Beispiel für eine weitere ausführliche Definition aus dem Bereich der Softwarelokalisierung findet sich in Jiménez-Crespo 2024, S. 148–149.

allerdings der Fokus des Begriffs *Globalisierung* einerseits auf der Entscheidung von Unternehmen, weltweit agieren zu wollen, und andererseits daran anschließend auf dem Prozess, der zum Erreichen dieser weltweiten Tätigkeiten erforderlich ist (vgl. Drewer und Ziegler 2014, S. 30–31, und Munday 2022, S. 240).

Ist die Globalisierungsentscheidung gefallen, erfolgt die der Lokalisierung vorgeschaltete *Internationalisierung*, die die „Grundlage für die anschließende Lokalisierung“ (Drewer und Ziegler 2014, S. 33) bildet. Produkte und Dienstleistungen sollen dabei so entwickelt werden, dass sie schnell und mit geringem Aufwand an andere Märkte angepasst, also lokalisiert, werden können (vgl. K.-D. Schmitz 2005a, S. 2; s. auch Beste 2006, S. 26, Drewer und Ziegler 2014, S. 33, und Behrens 2016, S. 46). Bei der Internationalisierung müssen sowohl technische, gesetzliche und kulturelle als auch sprachliche Eigenschaften der Produkte berücksichtigt werden (vgl. K.-D. Schmitz 2005a, S. 2; s. auch Beste 2006, S. 26, und Drewer und Ziegler 2014, S. 33). Sie erfolgt entsprechend bereits während der Produktentwicklung (vgl. Esselink 2000, S. 2; s. auch Beste 2006, S. 26). Behrens zählt als Gegenstände der Internationalisierung die Bereiche natürliche Sprache, Geschäftslogik, Kodierung, Formatierung, Farben, Formen und Gesten sowie Schallereignisse auf und stellt diese ausführlich dar (vgl. Behrens 2016, S. 46–59).

Im Rahmen der Lokalisierung erfolgt anschließend die Anpassung von Softwareprodukten an einen bestimmten Markt (vgl. K.-D. Schmitz 2005a, S. 2). Diesen Prozess, die Softwarelokalisierung, bezeichnen Drewer und Ziegler als einen der stärksten Bereiche der Lokalisierung allgemein und damit auch als einen wichtigen Markt (vgl. Drewer und Ziegler 2014, S. 41). Von der inzwischen nicht mehr aktiven Localisation Industry Standards Association (LISA) wurde die Lokalisierung im Bereich Software wie folgt definiert:²

Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold. (LISA zitiert nach Esselink 2000, S. 3)

Drewer und Ziegler nennen als im Rahmen der Lokalisierung anzupassende Aspekte konkret „die Sprache, die kulturellen und gesellschaftlichen Gegebenheiten und Konventionen sowie die rechtlichen und technischen Bedingungen im Zielland“ (Drewer und Ziegler 2014, S. 35). Dabei lassen sich

²Diese Definition wird auch in jüngerer Zeit noch häufig genannt, beispielsweise von Drewer und Ziegler (vgl. Drewer und Ziegler 2014, S. 35), von Jiménez-Crespo (vgl. Jiménez-Crespo 2013, S. 13) oder von Munday (vgl. Munday 2022, S. 240).

nach Drewer und Ziegler sprachliche und nichtsprachliche Aspekte der Lokalisierung unterscheiden. Der sprachliche Anteil entspricht der Übersetzung, wobei sie hierzu auch mit Verweis auf funktionale Übersetzungsmodelle die Anpassung von Datumsformaten, Maßeinheiten oder numerischen Angaben zählen (vgl. Drewer und Ziegler 2014, S. 36). Anpassungen im technischen und juristischen Bereich, wie z. B. andere Netzstecker oder eine andere Lizenzvereinbarung, aber auch Anpassung von Grafiken, Symbolen und Farben, gehören dagegen zum nichtsprachlichen Teil der Lokalisierung (vgl. Drewer und Ziegler 2014, S. 36). Ebenso können dazu Veränderungen an den bei der Bedienung der Software verwendeten Tonsignalen gehören (vgl. Zerfaß 2002, S. 214, und Behrens 2016, S. 59).

Entsprechend weisen Drewer und Ziegler darauf hin, dass die Anpassung kultureller Aspekte nur dann als Besonderheit des Bereichs Lokalisierung betrachtet werden kann, wenn man lediglich den Wort-für-Wort-Transfer aus einer in eine andere Sprache als Übersetzung bezeichnet und darüber hinausgehende Veränderungen, die nach funktionalistischem Übersetzungsverständnis ebenso Teil der Übersetzung sind, zum Beispiel die oben genannten Anpassungen von Maßeinheiten, nicht damit gemeint sind. Nach funktionalem Verständnis sind somit „die beiden Benennungen Übersetzung und Lokalisierung nahezu synonym“ (Drewer und Ziegler 2014, S. 39). Diese Argumentation wird auch in anderen Veröffentlichungen verfolgt, z. B. bei Göpferich (vgl. Göpferich 2002, S. 29–30).³

Für Esselink gehören noch weitere Tätigkeiten zur Lokalisierung:

Translation is only one of the activities in localization; in addition to translation, a localization project includes many other tasks such as project management, software engineering, testing, and desktop publishing. (Esselink 2000, S. 4; vgl. auch Esselink 2003, S. 69)

In der vorliegenden Arbeit wird mit *Softwarelokalisierung* (kurz *Lokalisierung*) die Gesamtheit aller Arbeiten beim Anpassen von Softwareprodukten und zugehörigen Materialien an eine Zielsprache und -kultur bezeichnet. *Übersetzen* bezeichnet hingegen den Teilprozess bei der Softwarelokalisierung, bei dem die in einem Softwarelokalisierungsprojekt enthaltenen Textelemente von einer Sprache in eine andere übertragen werden.

Die Übersetzung betrifft dabei verschiedene Bereiche von Softwarelokalisierungsprojekten, wie Benutzeroberfläche, Onlinehilfe, Produktdokumentation, sonstige Begleitmaterialien und Webseiten mit jeweils unterschiedlichen

³Bei Behrens findet sich eine ausführlichere Darstellung zu dieser Frage, an deren Ende Behrens sich im Unterschied dazu gegen eine klare Trennung dieser Begriffe ausspricht (vgl. Behrens 2016, S. 73–75).

Besonderheiten (vgl. z. B. K.-D. Schmitz 2005a, S. 3–6, Esselink 2000, S. 10–13, Beste 2006, S. 61–69, und Drewer und Ziegler 2014, S. 41). In der vorliegenden Arbeit wird aber nur der Bereich der grafischen Benutzeroberfläche untersucht.⁴

2.1.3 Grafische Benutzeroberflächen (GUIs)

Mit dem Begriff *grafische Benutzeroberfläche* (Graphical User Interface, GUI), der häufig verkürzt als *Benutzeroberfläche* wiedergegeben wird, wird der schon seit den 1990er Jahren gängigste Oberflächentyp bei Softwareprodukten bezeichnet (vgl. Schneider 1998, S. 364). Mit Oberfläche ist dabei „die *Bildschirmanzeige* gemeint. Grafisch steht im Gegensatz zu zeichenorientiert, welches die Eigenschaft klassischer *Terminals* (Bildschirme) ist.“ (Schneider 1998, S. 364, Hervorhebungen im Original) Gumm und Sommer verwenden für solche zeichenorientierten Oberflächen statt *Terminal* die Bezeichnung *Kommandozeile* und bezeichnen diese und GUIs als Bediensysteme (vgl. Gumm und Sommer 2016, S. 71), wobei letztere für sie „fensterorientierte grafische Bediensysteme“ (Gumm und Sommer 2016, S. 72) sind, die eine wesentliche Verbesserung der Bedienung mit sich bringen.

Laut Schneider kann eine GUI Fenster, Felder wie Schaltflächen, Pop-up-Fenster und Bildlaufleisten umfassen (vgl. Schneider 1998, S. 364; vgl. z. B. auch K.-D. Schmitz 2005a, S. 5). Esselink nennt zusätzlich Menüs, Dialogfelder sowie Fehler- und Statusmeldungen und Tooltips (vgl. Esselink 2000, S. 57–59; vgl. auch Wahle 2000b, S. 35, und Ottmann 2005, S. 101).⁵

Wahle bezeichnet die Oberflächenelemente als Interaktionselemente: „Die Interaktion kann sowohl über sprachliche als auch grafische Mittel stattfinden. Bei den zu lokalisierenden Interaktionselementen handelt es sich jedoch in erster Linie um Text“. (Wahle 2000b, S. 35) Beste ergänzt: „Der Text ist also Teil der Software“ (Beste 2006, S. 39) und dient zur Kommunikation zwischen Mensch und Maschine (vgl. Beste 2006, S. 39).

Von Reineke werden diese zu übersetzenden Textelemente, die Teil der

⁴Da auch die Onlinehilfe sowie die digital in der Software enthaltene Dokumentation zur Software gehören, werden auch diese Komponenten verschiedentlich als Teil der Benutzeroberfläche betrachtet (vgl. K.-D. Schmitz 2005a, S. 3, vgl. Ottmann 2005, S. 101, und vgl. Schneider 1998, S. 787; s. auch Beste 2006, S. 26). Bei der vorliegenden Untersuchung werden diese Komponenten aber außen vor gelassen, da sie nicht zur Interaktion mit dem Softwareprodukt dienen (s. Abschnitt 2.1.3). Ein kurzer Überblick zur Entwicklung der Softwarelokalisierung findet sich bei Esselink (vgl. Esselink 2003, S. 83–84), ein etwas aktuellerer bei Jiménez-Crespo (vgl. Jiménez-Crespo 2013, S. 7–11).

⁵Je nach Betriebssystem und Hersteller sind eine ganze Reihe verschiedener Oberflächenelemente mit teils unterschiedlichen Benennungen in Gebrauch. Dies spielt für die vorliegende Untersuchung allerdings keine Rolle und wird daher nicht vertieft.

grafischen Benutzeroberfläche sind, als „Softwarebenutzeroberflächentexte“ (Reineke 2005, S. 73) bezeichnet. Verschiedentlich findet sich dafür auch die Bezeichnung „Strings“ (z. B. bei Esselink 2000, S. 59), die allerdings primär auf textliche Elemente außerhalb von Menüs und Dialogfeldern angewendet wird, beispielsweise von Esselink und Ottmann. Ersterer nennt folglich als Beispiele für Strings Fehlernachrichten, Statusnachrichten, Fragen, Tooltips, Nachrichten der Statusleiste, Tastaturmodi, Listenelemente, Zeichenfolgen mit Variablen sowie dynamische Menü- oder Dialogfeldoptionen (vgl. Esselink 2000, S. 63). Letztere zählt zu den Strings Fehlernachrichten, Statusmeldungen, Eingabeaufforderungen und Tooltips (vgl. Ottmann 2005, S. 101). Beste verwendet die Begriffe GUI-Text und String dagegen weitgehend synonym, er stellt lediglich bei seiner Definition von Strings den technischen Aspekt in den Vordergrund: „Softwaretexte ersten Grades⁶ [...] bestehen in technischer Hinsicht aus Zeichenfolgen, so genannten *Strings*, die üblicherweise dem Übersetzer in Listenform vorliegen“ (Beste 2006, S. 73, Hervorhebung im Original). Behrens arbeitet in seiner Monografie mit der Benennung „User-Interface-Strings“ (vgl. Behrens 2016).

In der vorliegenden Arbeit wird für die Summe aller zu übersetzenden Textelemente in grafischen Benutzeroberflächen die Bezeichnung *GUI-Text* verwendet.

Die Bezeichnung *Strings* wird in der vorliegenden Arbeit gemäß Beste mit Fokus auf den technischen Aspekt genutzt, also für einzelne GUI-Textelemente im Untersuchungskorpus. Ein Beispiel für einen String wäre damit der Text für eine Menüoption oder für den Titel eines Dialogfelds oder auch der Text einer Fehlernachricht.

2.1.4 Lokalisierung von GUIs

Technisch gesehen kann GUI-Text im Rahmen von Softwarelokalisierungsprojekten in unterschiedlicher Form vorliegen:

- Als Teil des Quellcodes einer Software (vgl. z. B. Ottmann 2005, S. 103)
- In den Binärdateien, die zum lauffähigen Softwareprodukt gehören (vgl. z. B. Drewer und Ziegler 2014, S. 44)
- Von den Programmiererinnen für die Lokalisierung aus dem Quellcode extrahiert, beispielsweise in eine Tabelle (vgl. z. B. Ottmann 2005, S. 103)

⁶Bestes Definition der Softwaretexte ersten Grades entspricht dabei der hier gegebenen Definition für GUI-Text: „Softwaretexte ersten Grades umfassen die graphische Benutzeroberfläche mit ihren Bedienelementen, Befehlen und Dialogen.“ (Beste 2006, S. 39)

- In spezielle Dateien ausgelagert, die dann lokalisiert werden (vgl. z. B. Reineke 2005, S. 74)

Wenn Strings im Quellcode einer Software enthalten sind, kann für die Lokalisierung ein einfacher Texteditor zum Einsatz kommen. Allerdings ist die Gefahr dann groß, dass Teile des Quellcodes versehentlich verändert oder zu lokalisierende Strings übersehen werden. (vgl. Ottmann 2005, S. 103, und Wahle 2000b, S. 37)

Wenn GUI-Text in den Binärdateien eines Softwareprodukts enthalten ist, müssen spezielle Lokalisierungstools genutzt werden, um die Strings daraus zu extrahieren und zur Übersetzung bereitzustellen, sofern das Softwareprodukt mit einer Programmiersprache geschrieben ist, für die das Lokalisierungstool diesen Schritt ausführen kann (vgl. Freigang und Reinke 2005, S. 62, und Esselink 2000, S. 59). Roturier bezeichnet dies als „binary localization“ (vgl. Roturier 2015, S. 94–95). Nach der Lokalisierung wird dann eine zielsprachliche Binärdatei erzeugt (vgl. Freigang und Reinke 2005, S. 62, und Drewer und Ziegler 2014, S. 46). Im Unterschied zur Übersetzung im Quellcode ist bei dieser Vorgehensweise die visuelle Lokalisierung möglich, also die Übersetzung innerhalb der GUI-Elemente, in denen die Strings enthalten sind, wie Menüs und Dialogfelder (vgl. Ottmann 2005, S. 102–103, Jiménez-Crespo 2024, S. 163, und Drewer und Ziegler 2014, S. 45). In Lokalisierungstools kann zudem in der Regel auch die Größe der GUI-Elemente angepasst werden, wenn sie für die gewählte Übersetzung zu klein sind (vgl. Jiménez-Crespo 2024, S. 163). Außerdem bieten Lokalisierungstools Funktionen, mit denen beispielsweise Größenbeschränkungen und die korrekte Zuweisung von Access-Keys überprüft werden können (s. Abschnitt 2.2; vgl. Freigang und Reinke 2005, S. 62, und Wahle 2000b, S. 38). Die visuelle Lokalisierung ist allerdings nicht für Nachrichten und andere Strings möglich, die nicht in festen Bereichen der GUI angesiedelt sind, sondern während der Laufzeit je nach Situation angezeigt werden (vgl. Esselink 2000, S. 59).

Wenn Programmiererinnen Strings für die Lokalisierung in Dateien extrahieren, kann es sein, dass sie die Strings alphabetisch sortieren. Es sind also gar keine Zusammenhänge mehr zwischen den Strings oder auch Informationen dazu vorhanden, an welcher Stelle sie angeordnet sind. (vgl. Ottmann 2005, S. 103, Drewer und Ziegler 2014, S. 48, und Seewald-Heeg 2020, S. 15)

Im Hinblick auf die Internationalisierung ist es die beste Lösung, wenn sämtliche Strings eines Softwareprodukts dauerhaft in spezielle Dateien ausgelagert sind, die dann lokalisiert werden können und die im Folgenden als *Ressourcendateien* bezeichnet werden (vgl. K. J. Dunne 2009, S. 191, Esselink 2000, S. 3, und Jiménez-Crespo 2024, S. 148–149).⁷ In diesem Fall muss

⁷An einigen Stellen in der Literatur wird der Begriff *Ressourcendatei* allerdings für

also nicht im Quellcode gearbeitet werden und es sind keine speziellen Lokalisierungstools für die Lokalisierung erforderlich (vgl. Reineke 2005, S. 74, Freigang und Reinke 2005, S. 62, Behrens 2016, S. 46, und K. J. Dunne 2009, S. 192). Es müssen lediglich die Ressourcendateien übersetzt werden und über deren *String-IDs*⁸ werden die übersetzten Strings an den korrekten Stellen in der GUI angezeigt (vgl. Wahle 2000b, S. 40). Wird die GUI in mehrere Sprachen übersetzt, werden mehrere übersetzte Ressourcendateien in verschiedene Unterverzeichnisse bzw. unter verschiedenen Dateinamen gespeichert. In manchen Fällen können dann bereits im laufenden Softwareprodukt verschiedene GUI-Sprachen ausgewählt werden, teils müssen die lokalisierten Dateien allerdings auch gemeinsam mit dem Quellcode kompiliert und damit ein lauffähiges Softwareprodukt in der jeweiligen Sprache erstellt werden (vgl. Reineke 2005, S. 75, Freigang und Reinke 2005, S. 62, und Jiménez-Crespo 2024, S. 149). In der Regel ist auf diese Weise aber keine visuelle Lokalisierung inkl. Größenänderungen möglich. Solche Änderungen müssen nach der Kompilierung in der Entwicklungsumgebung vorgenommen werden (vgl. Reineke 2005, S. 75). Ein typisches Beispiel hierfür sind Properties-Dateien von Softwareprodukten, die in der Programmiersprache Java geschrieben wurden (vgl. Sachse 2005, S. 157–159).

Manche speziellen Dateitypen für ausgelagerte Strings, beispielsweise Windows-RC-Dateien, nehmen eine Zwitterstellung ein, da sie neben den Strings auch Informationen zu den GUI-Elementen, deren Anordnung und Größen enthalten können und damit in Lokalisierungstools auch visuell lokalisiert werden können. Auch hier trifft dies aber nicht für Nachrichten und ähnliche Strings zu. (vgl. Esselink 2000, S. 59–60, und Drewer und Ziegler 2014, S. 48)

Bei sämtlichen in das Korpus aufgenommenen Softwareprodukten (s. Abschnitt 5.3.1) sind die Strings in separate Ressourcendateien ausgelagert. Je nach Softwareprodukt und verwendetem Dateiformat lassen sich aber große Unterschiede in den Inhalten der Ressourcendateien sowie den String-IDs feststellen, die zum Einfügen der Strings in das laufende Softwareprodukt genutzt werden. Hier ein Beispiel aus einer Java-Properties-Datei, aus der Strings in das Untersuchungskorpus aufgenommen wurden:

- Mozilla Firefox, Datei `browser.properties`, Zeile 43:
`xpinstallDisabledButton=Enable`

`xpinstallDisabledButton` ist hier die String-ID, `Enable` ist der String.

einen bestimmten Dateityp, die Windows-RC-Dateien, verwendet, die im Folgenden separat besprochen werden (s. beispielsweise Drewer und Ziegler 2014, S. 44–46).

⁸Behrens bezeichnet diese als Ressourcenschlüssel (vgl. Behrens 2016, S. 43).

Der String-ID kann in diesem Fall die Information entnommen werden, dass es sich bei **Enable** um die Beschriftung einer Schaltfläche („Button“) handelt.

Ebenso wie die von Programmiererinnen extrahierten Strings haben auch die Strings in Ressourcendateien in der Regel nicht die Reihenfolge, in der sie im Softwareprodukt angezeigt werden, zumal es auch grundsätzlich nicht praktikabel sein dürfte, die Strings eines Softwareprodukts in ihrer Gesamtheit in eine sinnvolle Reihenfolge zu bringen. Teils werden zusammengehörige Strings, wie die eines Menüs oder eines Dialogfelds, gemeinsam angeordnet, technisch ist dies aber nicht erforderlich, weil den Ressourcendateien mithilfe der String-IDs die Strings entnommen und im laufenden Softwareprodukt angezeigt werden, was unabhängig davon geschieht, an welcher Stelle der Ressourcendateien die String-ID enthalten ist. Die String-IDs und Kommentare innerhalb der Ressourcendateien können allerdings Hinweise darauf enthalten, an welcher Stelle die Strings im Softwareprodukt angezeigt werden oder welche Funktion sie haben.

Ein Beispiel hierfür aus dem Softwareprodukt Notepad++ Version 7.6.4 findet sich in Abbildung 2.1: Links ist ein Ausschnitt der GUI mit einigen Hauptmenüoptionen und dem geöffneten Menü **File** zu sehen, rechts ein Ausschnitt aus der Ressourcendatei `english.xml`, in die die Strings dieses Softwareprodukts ausgelagert sind. Zu erkennen sind darin zwei Abschnitte, der erste mit vorangestelltem Kommentar `<!-- Main Menu Entries ->`, in dem die Hauptmenüoptionen in der Reihenfolge zu finden sind, die auch in der GUI sichtbar ist. Dem zweiten Abschnitt ist der Kommentar `<!-- Sub Menu Entries ->` vorangestellt. Er enthält allerdings nur einzelne der Menüoptionen aus dem Menü **File**, die im Screenshot zu sehen sind (jeweils mit einer String-ID, die mit der Zeichenfolge `file-` beginnt), und danach weitere aus dem Menü **Edit** (jeweils mit einer String-ID, die mit der Zeichenfolge `edit-` beginnt). Die ersten beiden der aufgeführten Menüoptionen aus dem Menü **File** im Screenshot, **Open Containing Folder** und **Close More**, sind dabei zwar in der richtigen Reihenfolge zu sehen, die weiteren Menüoptionen sind allerdings an dieser Stelle der Ressourcendatei nicht vorhanden und umgekehrt ist die Menüoption **Recent Files** aus der Ressourcendatei nicht im Menü **File** des Screenshots zu finden. Die an dieser Stelle der Ressourcendatei nicht enthaltenen Menüoptionen, die im Screenshot zu sehen sind, sind aber an anderer Stelle der Ressourcendatei zu finden, wobei vermutet werden kann, dass die String-IDs der betreffenden Strings ebenfalls einen Hinweis darauf geben, in welchem der Menüs die Menüoptionen angeordnet sind.

Wie an diesem Beispiel zu sehen ist, weisen einige Strings Besonderheiten auf, die entweder im technischen Bereich liegen, da die Strings als Teil von Softwareprodukten funktionieren müssen, die die Formatierung der Strings

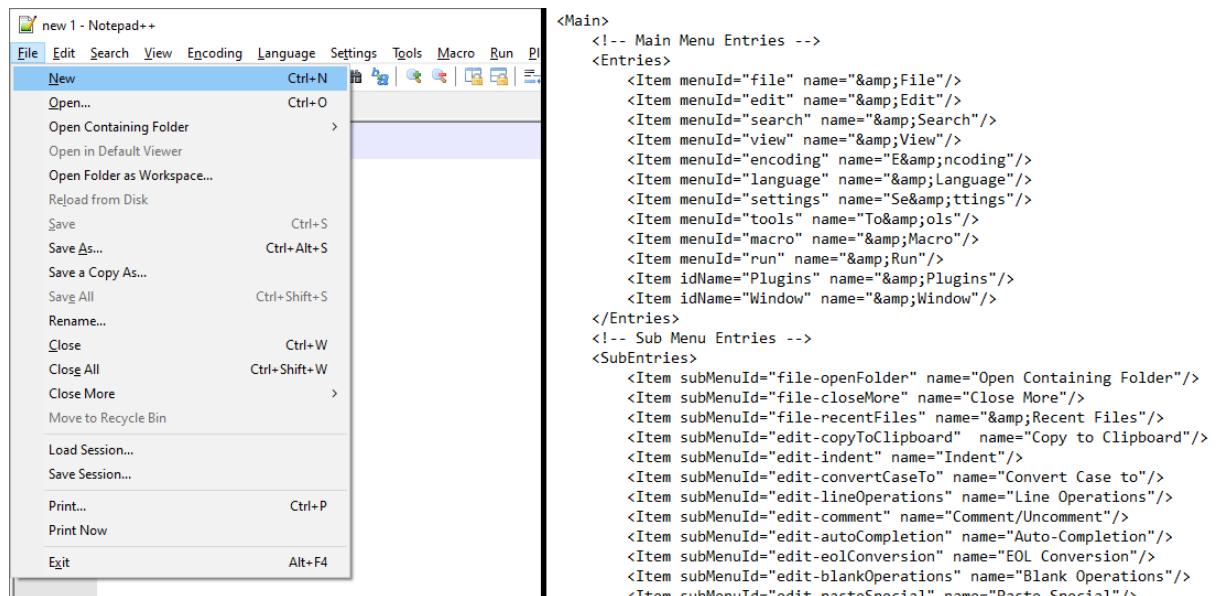


Abbildung 2.1: Notepad++ Version 7.6.4: Menü **File** und Ausschnitt der Datei `english.xml`

betreffen oder bei denen es sich um zeichenbasierte Konventionen handelt. Diese Merkmale unterscheiden sie von anderen Texten und müssen bei der Übersetzung berücksichtigt werden. Diejenigen dieser Merkmale, die im Untersuchungskorpus enthalten sind, werden im Folgenden erläutert.

2.2 Besonderheiten von GUI-Text

Strings können einige technische Besonderheiten aufweisen, die bei der Übersetzung berücksichtigt werden müssen. Zum einen enthalten sie Zeichen, die für die Benutzerinnen nicht sichtbar sind, weil sie bei der Ausführung des jeweiligen Softwareprodukts eine technische Funktion, wie die Auswahl einer Menüoption über die Tastatur, erfüllen (s. Abschnitt 2.2.1) oder eine Formatierung bewirken, beispielsweise die Einfügung eines Zeilenumbruchs (s. Abschnitt 2.2.2). Diese Zeichen dürfen beim Übersetzen nicht verändert oder nur gemäß den jeweiligen Vorgaben angepasst werden (vgl. Sachse 2005, S. 149) und werden als *Steuerzeichen* (vgl. Reineke 2005, S. 82) bzw. als *Steuercodes* (vgl. Esselink 2000, S. 69) bezeichnet. Darüber hinaus weisen Strings teils auch Konventionen auf der Zeichenebene auf, die zusätzliche Informationen

für Benutzerinnen bieten (s. Abschnitt 2.2.3).⁹ In den folgenden Abschnitten werden diejenigen dieser Besonderheiten von GUI-Text kurz beschrieben, die im Untersuchungskorpus enthalten sind.

2.2.1 Steuerzeichen mit Bezug zur Programmfunktionalität

*Access-Keys*¹⁰ stellen eine der Methoden dar, GUI-Elemente wie Menüoptionen oder Schaltflächen anstatt über die Maus über die Tastatur auszuwählen, und tragen damit zur Barrierefreiheit bei. In der Regel wird dafür die Alt-Taste plus ein Buchstabe gedrückt, der im auszuwählenden GUI-Element enthalten und durch Unterstreichung hervorgehoben ist (vgl. Esselink 2000, S. 70, und K. J. Dunne 2009, S. 194). Beispiele können Abbildung 2.2 entnommen werden: Über die Tastenkombination **Alt+F** kann das Menü **File** geöffnet werden, über die Tastenkombination **Alt+N** die Menüoption **New**. Die unterstrichenen Buchstaben „F“ und „N“ fungieren hierbei als Access-Keys.

Esselink nennt zwei gängige Konventionen für die Definition eines Access-Keys in einem String, zum einen das Et-Zeichen („&“), zum anderen die Tilde („~“), die jeweils vor dem Buchstaben platziert werden müssen, der als Access-Key definiert werden soll (vgl. Esselink 2000, S. 70–71). Im Korpus werden für Access-Keys das Et-Zeichen, der Unterstrich und die Tilde verwendet. Beispiele:

⁹Neben den im Folgenden genannten technischen und textlichen Besonderheiten finden sich noch eine Reihe weiterer Aspekte, die bei der Softwarelokalisierung von Bedeutung sind oder spezielle Schwierigkeiten mit sich bringen, für die vorliegende Untersuchung aber keine Rolle spielen. Hierzu gehören Platzprobleme und Resizing (vgl. beispielsweise Ottmann 2005, S. 107, Esselink 2000, S. 26, 67–68, Beste 2006, S. 71, K.-D. Schmitz 2005a, S. 16–17, und Drewer und Ziegler 2014, S. 43, 50–51), die Darstellung von Zeichen außerhalb des ASCII-Zeichensatzes (vgl. beispielsweise Esselink 2000, S. 26, Beste 2006, S. 70, und K.-D. Schmitz 2005a, S. 13–14), Formate für Datum, Währungen und Zahlen (vgl. beispielsweise Esselink 2000, S. 26, und Beste 2006, S. 71) sowie die Sortierfolge (vgl. beispielsweise Beste 2006, S. 70, und Ottmann 2002, S. 160) sowie Abkürzungen und nicht zu übersetzende Elemente (vgl. beispielsweise Ottmann 2005, S. 106–107, 108). Zu diesen Aspekten gehört ebenso das Problem der Concatenation/Konkatenierung (vgl. beispielsweise Ottmann 2005, S. 110, Esselink 2000, S. 34, 70, Beste 2006, S. 74, und Behrens 2016, der sich primär mit diesem Thema befasst), das ebenfalls in der vorliegenden Arbeit unberücksichtigt bleibt, weil es dabei nicht um die Strings und deren textliche Inhalte selbst geht, sondern darum, wie sie im laufenden Softwareprodukt zu größeren Sinneinheiten zusammengefügt werden.

¹⁰Werden u. a. auch als Mnemonics, Accelerator-Keys, Hotkeys oder Zugriffstasten bezeichnet (vgl. Wahle 2000b, S. 47, Ottmann 2005, S. 109, Esselink 2000, S. 70, Beste 2006, S. 64, Roturier 2015, S. 90, und Sachse 2005, S. 149).

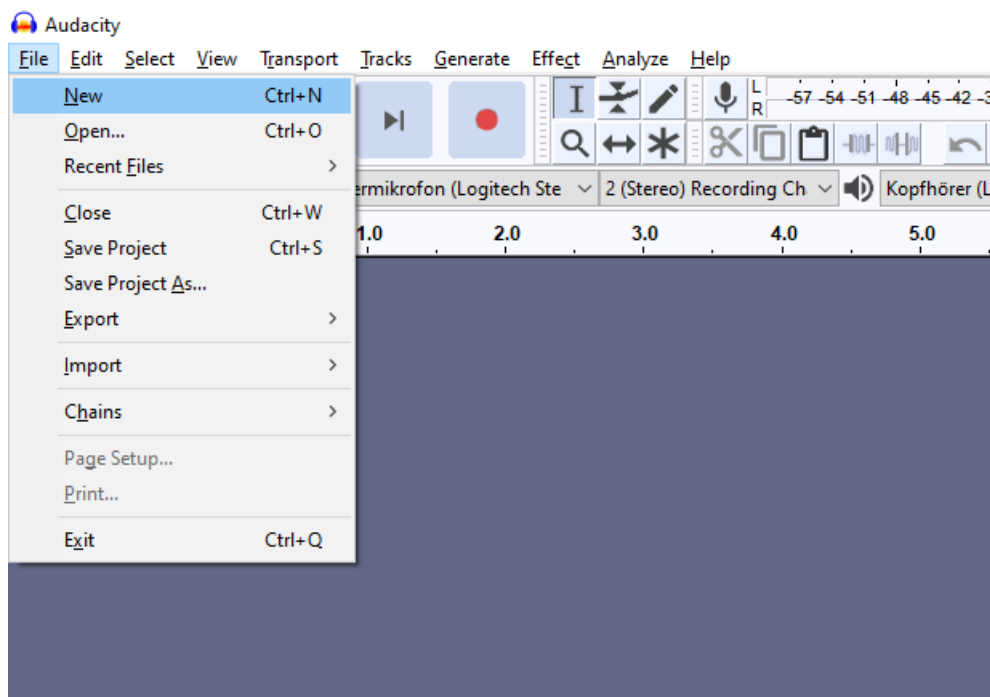


Abbildung 2.2: Access-Keys und Shortcuts in Audacity Version 2.2.2

- Datei `dateimanager.txt`, Zeile 284: `C&ase sensitive`¹¹
- Datei `email.txt`, Zeile 5: `_Set as default account`
- Datei `office-anwendung.txt`, Zeile 36: `Delete ~Record`

Verwandt mit den Access-Keys sind die *Shortcuts*, da auch mit ihnen die Auswahl von Programmfunktionen über die Tastatur ermöglicht wird. Beste spricht dabei von Tastenkombinationen für schnellen Zugriff auf einen Befehl, bei denen im Unterschied zu Access-Keys kein Umweg über ein Menü erforderlich ist. Shortcuts werden üblicherweise in Menüs rechts neben den zugehörigen Menüoptionen aufgelistet. (vgl. Beste 2006, S. 64) In Abbildung 2.2 finden sich mehrere Beispiele, etwa die Tastenkombination **Ctrl+N** zum Auswählen der Menüoption **New**. Shortcuts unterscheiden sich also technisch gesehen etwas von Access-Keys, müssen aber auch übersetzerisch anders behandelt werden: Bei Access-Keys muss der gewählte Buchstabe in der Übersetzung der Menüoption enthalten sein und damit ggf. beim Übersetzen geändert werden, bei Shortcuts ist dies nicht der Fall. Daher werden letztere häufig bei der Lokalisierung nicht verändert, wie beispielsweise der

¹¹Informationen zu den im Korpus verwendeten Softwareprodukten sowie den hier und bei den folgenden Beispielen angegebenen Dateien finden sich in Abschnitt 5.3.1 sowie in Anhang A.

häufige Shortcut Strg+P zum Drucken zeigt: Im Englischen gibt es einen Bezug zwischen dem Buchstaben „P“ und der Menüoption/Funktion **Print**, der aber bei der Lokalisierung beispielsweise ins Deutsche mit der Menüoption/Funktion **Drucken** verloren geht. Hier ein Beispiel für einen Shortcut aus dem Korpus:

- Datei `audiobearbeitung.txt`, Zeile 8: `&Undo\tCtrl+Z`¹²

Der letzte Steuerzeichentyp mit Bezug zur Programmfunktionalität sind die *Variablen* bzw. *Platzhalter* (vgl. beispielsweise Sachse 2005, S. 150, Ottmann 2005, S. 110–111, Roturier 2015, S. 28–32, und Jiménez-Crespo 2024, S. 154–156). Laut Esselink sind dies in der Regel Zeichen, denen ein Prozentzeichen vorangestellt wird. Wenn das Softwareprodukt läuft und ein String mit Variable angezeigt wird, wird diese durch ein Wort, einen Wert oder eine Zeichen ersetzt (vgl. Esselink 2000, S. 68). Sachse liefert dafür ein Beispiel:

Platzhalter werden verwendet, wenn z.B. dynamisch zur Laufzeit eine Meldung zusammengesetzt werden muss. Eine Meldung wie „Die Datei xyz.txt enthält 54 Wörter.“ kann nicht vom Entwickler fest vorgegeben werden, denn der Name der Datei und die Anzahl an Wörtern ergeben sich erst durch die Dateiauswahl des Anwenders. (Sachse 2005, S. 150, Hervorhebung im Original)

Auch hierfür finden sich im Korpus verschiedene Beispiele mit Variablen, die verschiedenen Konventionen folgen:

- Datei `audiobearbeitung.txt`, Zeile 1: `File '%s' already exists, do you really want to overwrite it?`
- Datei `browser.txt`, Zeile 41: `Sign in to {1}`
- Datei `html-editor.txt`, Zeile 293: `Abandon unsaved changes to "%title%" and reload page?`

Im ersten Beispiel wird die Variable mit der Zeichenfolge `%s` gebildet, in der zweiten ist es eine Zahl in geschweiften Klammern (`{1}`) und in der dritten besteht sie aus der Zeichenfolge `%title%`.

Die letzte Gruppe der Steuerzeichen mit Bezug zur Programmfunktionalität sind die *Escapezeichen*¹³, die im Unterschied zu den anderen nicht im Zusammenhang mit Funktionen stehen, sondern diese aufheben sollen. Sie

¹²Neben dem Shortcut sind in diesem Beispiel auch noch ein Access-Key und das Steuerzeichen `\t` (s. Abschnitt 2.2.2) enthalten.

¹³Von Sachse auch als Escape-Sequenzen bezeichnet (vgl. Sachse 2005, S. 149).

werden eingesetzt, um ein Zeichen in der GUI des Softwareprodukts anzeigen zu können, das in der betreffenden Datei eigentlich als Steuerzeichen definiert ist. Sachse liefert dafür ein Beispiel mit dem oben genannten Access-Key „&“: Wenn ein String wie „Meyer & Söhne“ im Softwareprodukt angezeigt werden soll, wird dem Programm durch ein zweites vorangestelltes „&“, das hier als Escapezeichen dient, signalisiert, dass das Zeichen „&“ in der GUI ein Mal angezeigt werden soll, anstatt als Steuerzeichen das nachfolgende Zeichen als Access-Key zu definieren. Im String selbst müsste also „Meyer && Söhne“ angegeben werden (vgl. Sachse 2005, S. 149). In diesem Beispiel ist also das erste Et-Zeichen das Escapezeichen, das die Funktion des zweiten Zeichens als Access-Key aufhebt. Auch hierfür finden sich im Korpus Beispiele:

- Datei `dateimanager.txt`, Zeile 1: `Comparing... %d%% (ESC to cancel)`¹⁴

2.2.2 Steuerzeichen mit Bezug zur Formatierung

Im Folgenden finden sich verschiedene Typen von Steuerzeichen, die im Korpus enthalten sind und sich auf die Formatierung von Strings in der GUI auswirken und im Korpus enthalten sind.

Den ersten Bereich bildet *HTML-Code*, mit dem die Funktionalität von HTML zur Formatierung von Strings eingesetzt wird. Beispielsweise kommt im folgenden String das HTML-Tagpaar `...` zum Einsatz, das bewirkt, dass der String in der GUI in Fettdruck angezeigt wird:

- Datei `videobearbeitung.txt`, Zeile 112: `Audio Output`

Es werden aber auch andere Steuerzeichen für die Formatierung eingesetzt, die nicht aus HTML-Tags bestehen. Ein Beispiel dafür ist die Zeichenfolge `\t`, die beispielsweise zum Einfügen von Tabulatorschritten, aber auch zum Abgrenzen eines Shortcuts von der zugehörigen Menüoption dient (vgl. Esselink 2000, S. 62, Jiménez-Crespo 2024, S. 155, und Sachse 2005, S. 150). Die Teile des Strings, die auf diese Steuerzeichen folgen, werden einen Tabulatorschritt nach rechts gerückt. Im Korpus finden sich diverse Beispiele für dieses Steuerzeichen wie etwa folgendes:

¹⁴Dieses Beispiel enthält mehrere der in diesem und den folgenden Abschnitten beschriebenen Besonderheiten: eine Variable (`%d`) und das Escapezeichen direkt im Anschluss an den Buchstaben „d“ in Form eines Prozentzeichens (`%`), mit dem sichergestellt wird, dass im angezeigten String ein Prozentzeichen zu sehen ist, sowie drei Punkte (s. Abschnitt 2.2.3).

- Datei `audiobearbeitung.txt`, Zeile 8: `&Undo\tCtrl+Z`¹⁵

Auf ganz ähnliche Weise wie das Steuerzeichen `\t` funktioniert das Steuerzeichen `\n`, mit dem Zeilenumbrüche in einen String eingefügt werden (vgl. Sachse 2005, S. 150, 159, und Jiménez-Crespo 2024, S. 155). Die Teile des Strings, die nach diesen Steuerzeichen folgen, werden in der nächsten Zeile dargestellt. Auch hierfür enthält das Korpus verschiedene Beispiele:

- Datei `email.txt`, Zeile 51: `Do you really want to replace all attribute names \n with the default set?`

Für die Benutzerinnen wird diese Nachricht in der GUI in zwei Zeilen untereinander angezeigt. Die erste enthält die Wörter bis zum Steuerzeichen, also „Do you really want to replace all attribute names“, die zweite die übrigen Wörter, also „with the default set?“.

2.2.3 Zeichenbasierte Konventionen

Neben den verschiedenen Typen von Steuerzeichen gibt es auch noch zeichenbasierte Konventionen, die den Benutzerinnen zusätzliche Informationen bieten sollen. Hiervon finden sich im Korpus nur *drei Punkte* am Ende von Menüoptionen, die Beste als typografisches Gestaltungsmittel (vgl. Beste 2006, S. 42) bezeichnet, da damit keine technische Funktion bzw. Formatierung verknüpft ist, sondern in Menüs signalisiert wird, dass nach Auswahl der Menüoption nicht ohne weitere Rückfrage eine Aktion ausgeführt wird, sondern zunächst ein Dialogfeld geöffnet wird, meist mit weiteren Auswahlmöglichkeiten (vgl. Esselink 2000, S. 62; vgl. auch Beste 2006, S. 42, und K. J. Dunne 2009, S. 194).¹⁶ In Abbildung 2.2 weist beispielsweise die Menüoption **Open...** drei Punkte auf. Wenn sie ausgewählt wird, wird ein Dialogfeld geöffnet, in dem die Benutzerin in der Verzeichnisstruktur zu einer zu öffnenden Datei navigieren kann.

Daneben werden drei Punkte auch dafür verwendet, einen laufenden Prozess zu signalisieren. An einzelnen Stellen dienen sie außerdem dazu, einen unvollständigen Satz bzw. ein unvollständiges Wort anzugeben. Hier jeweils ein Beispiel für diese unterschiedlichen Fälle aus dem Korpus:

- Datei `browser.txt`, Zeile 157: `Customize...`

¹⁵Neben dem Tabulator sind in diesem Beispiel auch noch ein Access-Key und ein Shortcut enthalten.

¹⁶Weitere der von Beste aufgeführten typografischen Gestaltungsmittel finden sich in Abschnitt 4.1. Bei ihnen wird die Formatierung über andere Methoden erzeugt, nicht durch in den Strings enthaltene Zeichen, weshalb sie separat diskutiert werden.

- Datei `email.txt`, Zeile 160: `Busy searching...`
- Datei `medienplayer.txt`, Zeile 47: `General input settings. Use with care...`

Beim ersten Beispiel handelt es sich um drei Punkte am Ende einer Menüoption, beim zweiten wird ein laufender Prozess angegeben und beim letzten ein unvollständiger Satz.

2.3 Freie Software

In Bezug auf den Quellcode und dessen Verfügbarkeit werden bei Softwareprodukten proprietäre und freie Software sowie Open-Source-Software unterschieden. Erstere ist Software, deren Quellcode nicht für die Allgemeinheit einsehbar ist. Proprietäre Software liegt Benutzerinnen lediglich in Form ausführbarer Programme, also bereits kompiliert vor. Diese ausführbaren Programme können nicht wieder in Quellcode konvertiert werden, wodurch Struktur, Aufbau und genaue Funktionen der Software nicht zu erkennen und auch keine Weiterentwicklungen durch Dritte möglich sind. Übliche Lizenzen gestatten Benutzerinnen lediglich, die Software auszuführen. (vgl. Zittrain 2004, S. 269, 271) Teils kann es allerdings auch sein, dass der Quellcode einsehbar und auch veränderbar ist, diese Veränderungen allerdings nicht weitergegeben werden dürfen (vgl. Zittrain 2004, S. 270).

Im Unterschied dazu hat Richard Stallman mit dem von ihm gegründeten Projekt namens GNU den Begriff *freie Software* eingeführt, mit der er vier Freiheiten verbindet:

- Die Freiheit, das Programm für jeden beliebigen Zweck auszuführen
- Die Freiheit, das Programm nach Bedarf zu ändern (umfasst auch Zugriff auf den Quellcode)
- Die Freiheit, Exemplare des Programms entweder gratis oder gegen Gebühr weiterzugeben
- Die Freiheit, geänderte Versionen des Programms weiterzugeben, damit die Community profitiert (vgl. Stallman 2002, S. 20)

Nach Stallman wurde 1998 neben dem Begriff *freie Software* auch der Begriff *Open Source* eingeführt, der letztlich einen etwas anderen Blickwinkel auf denselben Sachverhalt bedeutet: Für die Open-Source-Bewegung sei der Aspekt, ob der Quellcode frei verfügbar ist, eher eine praktische als eine ethische Frage (vgl. Stallman 2002, S. 57). Nach Sandrini und García González steht bei Open Source die Zusammenarbeit im Zentrum, bei freier Software

die Freiheit (vgl. Sandrini und García González 2015, S. 9). Weitgehend führt dies in der Praxis aber zu keinen Unterschieden.

Sowohl für freie Software als auch für Open-Source-Software gilt in der Regel etwa das Copyleft-Prinzip, das besagt, dass für veränderte Versionen eines Programms, die weitergegeben werden, dieselben vier Freiheiten wie für die ursprüngliche Programmversion gelten sollen (vgl. Zittrain 2004, S. 266). Ein Beispiel hierfür ist die General Public License (GPL) von GNU/Linux (vgl. Zittrain 2004, S. 269), die in verschiedenen Versionen für die im Korpus enthaltenen Softwareprodukte genutzt wird. Für eines der Softwareprodukte gilt die LGPL (GNU Lesser General Public License), bei der das Copyleft ausgeschlossen ist.¹⁷ Für drei wird die MPL (Mozilla Public License) mit eingeschränktem Copyleft genutzt.¹⁸ Informationen zu den Textproben, den dafür genutzten Softwareprodukten und deren Lizenzen finden sich in Abschnitt 5.3.1.

¹⁷https://de.wikipedia.org/wiki/GNU_Lesser_General_Public_License; letzter Zugriff: 20.06.2025

¹⁸https://de.wikipedia.org/wiki/Mozilla_Public_License; letzter Zugriff: 20.06.2025

Kapitel 3

Text, Fachtext, Kontext

Bereits im letzten Kapitel wurde im Zusammenhang mit Softwarelokalisierung und GUI-Text der Begriff *Text* genannt (Kapitel 2). Was genau ein Text ist bzw. was ihn ausmacht, wurde allerdings noch nicht thematisiert. Dieser Schwerpunkt wird in diesem Kapitel in Abschnitt 3.1 gesetzt. Im Abschnitt 3.2 finden sich daran anschließende Informationen zum Thema *Textsorte*. Abschnitt 3.3 beleuchtet dann den *Fachtext* und im letzten Abschnitt 3.4 finden sich Erläuterungen zum Thema *Kontext*.

Im folgenden Kapitel, Kapitel 4, werden diese einleitenden Informationen dann in Bezug zu GUI-Text gestellt, wodurch die Frage beantwortet werden kann, ob es sich bei GUI-Text um einen multimodalen Fachtext einer bestimmten Textsorte, nämlich Software, handelt. Im Anschluss können dann die hier erarbeiteten Kriterien zum Thema *Kontext* auf diese multimodale Fachtextsorte übertragen werden.

3.1 Text

In ihrer Monografie erläutert Adamzik ausführlich, warum die Definition des Begriffs *Text* aus textlinguistischer Sicht problematisch ist (vgl. Adamzik 2016, S. 40–48).¹ Dabei skizziert sie, dass das Definitionsproblem auf einer „eingengten Vorstellung (vom Sinn) von Definitionen“ (Adamzik 2016, S. 41) beruht und dabei der jeweilige Gebrauchskontext und der Wandel,

¹Überblicke finden sich auch an anderer Stelle, beispielsweise bei Gambier und Lautenbacher, die in ihrem Beitrag ebenfalls einen Abriss der Entwicklung des Textbegriffs in der Linguistik darstellen. Darin schließen sie an Adamziks Erläuterungen an, indem sie zunächst gängige Definitionen wie die bekannten sieben Textualitätskriterien von Beaugrande and Dressler (vgl. De Beaugrande und Dressler 1981, S. 3–13) nennen und dann ihrerseits die Tatsache herausstellen, dass keine Einigkeit bei der Definition des Textbegriffs erreicht wurde. (vgl. Gambier und Lautenbacher 2024, S. 3–4)

dem auch wissenschaftliche Begriffe unterliegen können, nicht berücksichtigt wird (vgl. Adamzik 2016, S. 41). Sie plädiert daher für eine prototypische Definition, bei der zwar ebenso Merkmale angegeben werden, diesen aber ein anderer Status zugewiesen wird, indem zwischen zentralen und peripheren Merkmalen unterschieden wird (vgl. Adamzik 2016, S. 42). Damit schließt sie an ein u. a. von Sandig (vgl. z. B. Sandig 2000) vertretenes Konzept an. Adamzik vertritt zudem die Ansicht, dass Texte als komplexe Phänomene unterschiedlichen Kategorien zugeordnet bzw. unter unterschiedlichen Aspekten betrachtet werden können, weshalb ihrer Ansicht nach verschiedene Definitionen erforderlich sind, beispielsweise je nach Aspekten, die betrachtet werden (vgl. Adamzik 2016, S. 42–43).

Mögliche Merkmale im Rahmen einer prototypischen Textdefinition lassen sich aber durchaus beispielsweise bei Beaugrandes und Dresslers bekanntem Ansatz finden, der darauf basiert, die kommunikative Funktion eines Textes in den Vordergrund zu stellen. Beaugrande und Dressler sprechen von sieben Textualitätskriterien, nämlich Kohäsion, Kohärenz, Intentionalität, Akzeptabilität, Informativität, Situationalität und Intertextualität (vgl. De Beaugrande und Dressler 1981, S. 3–13).² Werden nicht alle diese Kriterien erfüllt, sehen sie einen Text als nicht kommunikativ und damit als Nicht-Text an (vgl. De Beaugrande und Dressler 1981, S. 3). Auch Stolze geht in diese Richtung, indem sie „Text als eine deutlich eingegrenzte, einen Sinnzusammenhang bildende Einheit miteinander verknüpfter Sprachzeichen als Beziehungsgeflecht aus Information, Intention und Situation“ (Stolze 1985, S. 23) definiert. Dem schließt sich Nord an, die ebenfalls betont, dass Texte in eine kommunikative Situation eingebettet sind. Dabei ist für sie Voraussetzung „das Vorhandensein einer in Raum und Zeit fixierten oder fixierbaren Situation und mindestens zweier Kommunikationspartner, welche in der Lage sind und die Absicht (Intention) haben, miteinander zu einem bestimmten Zweck zu kommunizieren“ (Nord 2009, S. 13).

Göpferichs Definition des Begriffs *Text*, die wie auch die von Stolze und Nord aus Sicht der Translationswissenschaft erfolgt³, bezieht sich auf Beaugrandes und Dresslers Textualitätskriterien, setzt aber eigene Akzente und

²Adamzik sieht die Textualitätskriterien von Beaugrande und Dressler als „den einflussreichsten Versuch einer Zusammenschau [...], der bis heute eine Referenz bildet“ (Adamzik 2016, S. 57). Adamziks Ansicht nach handelt es sich dabei zwar um eine Nebeneinanderstellung verschiedener Aspekte, sie schätzt diese aber als positiv ein, weil das Modell von Beaugrande und Dressler dadurch offen sei für Ansätze unterschiedlicher Forschungsrichtungen (vgl. Adamzik 2016, S. 57). Eine detaillierte Darstellung der Rezeption der Textualitätskriterien von Beaugrande und Dressler findet sich bei Adamzik (vgl. Adamzik 2016, S. 98–101).

³Ebenso betonen Gambier und Lautenbacher die Relevanz des Textbegriffs für die Translationswissenschaft (vgl. Gambier und Lautenbacher 2024, S. 4).

ist weniger konkret, was eine prototypische Herangehensweise erleichtert:

Ein Text ist ein thematisch und/oder funktional orientierter, kohärenter sprachlicher oder sprachlich-figürlicher Komplex, der mit einer bestimmten Intention, der Kommunikationsabsicht, geschaffen wurde, eine erkennbare kommunikative Funktion [...] erfüllt und eine inhaltlich und funktional abgeschlossene Einheit bildet. (Göpferich 1995, S. 56)

Göpferich ist der Ansicht, dass Kohäsion keine Grundbedingung für das Vorliegen eines Textes ist. Von Relevanz seien hingegen „die thematische Orientierung, Intentionalität, eine erkennbare kommunikative Funktion, Kohärenz und Abgeschlossenheit“ (Göpferich 2006, S. 61). Mit Kohäsion ist dabei gemeint, dass zwischen den sprachlichen Einheiten auf der formal-grammatischen Ebene ein Zusammenhang besteht, die Kohärenz betrifft den Zusammenhang auf inhaltlich-logischer Ebene (vgl. Göpferich 1995, S. 42). Nicht sinnvoll sei zudem die Reduktion des Textbegriffs auf sprachliche Komplexe: „Oftmals liefern Bilder nämlich Komplementärinformationen zum rein verbal Vermittelten, so dass ihre Ausgrenzung aus dem Textbegriff zu mangelnder Kohärenz des rein verbalen Restes führen [...] würde“. (Göpferich 2006, S. 61)

Gerade dieser letzte Punkt ist gegenüber den anderen bisher genannten Definitionen neu, ist aber mit Blick auf die Softwarelokalisierung von Interesse und wird neben Göpferich auch von anderen Autorinnen wie Kvam et al. genannt. Diese heben einerseits die sprachlichen Zeichen als Träger der Bedeutung von Texten hervor, machen andererseits aber auch die Relevanz von Elementen aus anderen semiotischen Systemen, die Polysemiotik von Texten, deutlich. Dabei sind sowohl „nicht-verbale Zeichen als auch paraverbale Zeichen in der Form von etwa Schriftgröße und Schrifttyp [...] zusätzlich zu den verbalen Zeichen wichtige und für die Botschaft des Textes zentrale Bedeutungsträger“ (Kvam u. a. 2024, S. 14–15). Die paraverbalen Zeichen kann man aber auch wie Fix zusammen mit weiteren „Faktoren, die die sinnliche Wahrnehmbarkeit eines Textes betreffen“ als *Materialität* eines Textes bezeichnen, also „die formale Sichtbarmachung und Gestaltung der sprachlichen Zeichen“ (Fix 2008, S. 344).⁴ Fix nennt darüber hinaus auch noch die *Medialität*, „die technischen Mittel für die Übertragung und Speicherung von Zeichen“ (Fix 2008, S. 344).⁵ Fix ergänzt:

⁴An anderer Stelle zählt sie als Beispiele für nichtsprachliche Zeichen, die zusammen mit sprachlichen Zeichen das Sinnangebot eines Textes konstituieren, „Bilder, typographische Elemente, Farben, Papiersorten, Flächen, Linien“ (Fix 2001, S. 114) auf.

⁵Die Medialität stellt auch bei anderen Definitionen einen relevanten Aspekt dar, bei-

Aus semiotischer Sicht kann man die Mittel, mit deren Hilfe der Text überhaupt erst wahrnehmbar gemacht wird, nicht aus der Betrachtung – weder der des Einzeltextes noch der der Textsorte – ausschließen; schließlich sind alle diese Mittel an der Konstitution von Textsinn und an der Lenkung der Rezeption beteiligt – und dies zuweilen sogar in entscheidender Weise [...]. (Fix 2008, S. 345)

Den Aspekt der Medialität ergänzen Gambier und Lautenbacher, indem sie auf das Aufkommen digitaler Texte in den letzten drei Jahrzehnten und damit die Übersetzung bzw. Lokalisierung von Websites, YouTube-Videos, Videospielen, Blogs usw. hinweisen (vgl. Gambier und Lautenbacher 2024, S. 4):

From now on, „texts“ are fluid, with other „texts“ and other systems of signs (fixed or moving images, graphs, colors, fonts, sounds, etc.). A „text“ has become polysemiotic or multimodal and exists in a permanent intertextual relationship with other „texts.“ (Gambier und Lautenbacher 2024, S. 5)

Im digitalen Bereich zeigt sich die Materialität dabei teils ähnlich und teils erweitert gegenüber dem analogen: „To this change from ‚text‘ (verbal, linear, and unidirectionally produced) to hypertext is added the presence of other sign systems – non-alphabetic, symbolic, graphic, visual, and audial.“ (Gambier und Lautenbacher 2024, S. 6)

Klug und Stöckel nutzen den Begriff *Modalität*, um auf verschiedene dieser nichtsprachlichen Zeichensysteme Bezug zu nehmen, und sprechen in diesem Zusammenhang wie auch beispielsweise Gambier und Lautenbacher (vgl. Gambier und Lautenbacher 2024, S. 1) von *multimodalen Texten*, die auf der „semantischen und funktionalen Verknüpfung bzw. wechselseitigen Integration verschiedener Zeichenmodalitäten – wie Sprache, Bild, Musik, Geräusch in einem textuellen Rahmen (z. B. in Film, Comic oder illustrierter Zeitschrift)“ (Klug und Stöckl 2016, S. VII) beruhen.⁶

Hier wird also schon deutlich, dass die Modalität über Text-Bild-Gefüge wie von Göpferich oben genannt hinausgeht. Schmitz bringt als Beispiele die gesprochene Sprache mit Mimik, Gestik, Lautstärke, Prosodie und Stimmführung. Bei der geschriebenen Sprache nennt er mit Absatzgliederung, Layout, Typographie und Textdesign (vgl. U. Schmitz 2016, S. 328) Aspekte

spielsweise bei Halliday und Hasans Textdefinition: „[...] any instance of living language that is playing some part in a context of situation, we shall call a text. It may be either spoken or written, or indeed in any other medium of expression that we like to think of.“ (Halliday und Hasan 1989, S. 10).

⁶Laut Steinseifer findet in anderen Disziplinen statt des Begriffs *Multimodalität* der Begriff *Multimedialität* als Synonym Verwendung (vgl. Steinseifer 2011, S. 164).

der bereits zuvor genannten Materialität.⁷ Aufgrund der Vielfalt potenzieller Multimodalität hält er es für schwierig, dafür ein einziges taugliches Schema zu entwickeln (vgl. U. Schmitz 2016, S. 330). Zunächst schlägt er jedoch folgende Definitionen für Text und für multimodalen Text vor:

- Text: „[...] jedes nach außen (z. B. durch ungefüllte Zeiten oder Flächen) abgegrenzte Zeichengebilde, das mindestens ein sprachliches Zeichen enthält. Man unterscheidet visuelle (z. B. schriftliche) von akustischen (mündlichen) Texten.“ (U. Schmitz 2016, S. 331)
- Multimodaler Text: „[...] sämtliche nach außen (z. B. durch ungefüllte Zeiten oder Flächen) abgegrenzte Zeichengebilde, die mindestens ein sprachliches Zeichen enthalten und mindestens zwei der fünf Modi gesprochene Sprache, geschriebene Sprache, stehendes Bild (z. B. Foto), bewegte Bilder (z. B. Film) und Audio (z. B. Musik) nutzen.“ (U. Schmitz 2016, S. 333)⁸

Diese Definitionen machen deutlich, dass für Schmitz ebenso wie für Göpferich die Aspekte Kohärenz und Abgeschlossenheit Bedeutung haben, den Sprachzeichen im Vergleich bei Schmitz aber weit weniger Relevanz beigemessen wird. Ebenso verzichtet er auf die Aspekte Thema/Funktion und Kommunikationsabsicht.

Seine Definitionen für Text und multimodalen Text bettet Schmitz in eine multimodale Texttypologie ein, für die er folgende Aspekte als relevant betrachtet:

⁷Nach Schmitz' Einschätzung wird allerdings „die materielle Erscheinungsform des Textes größtenteils ignoriert“ (vgl. U. Schmitz 2016, S. 328). Schriftliche Texte gelten damit seiner Meinung nach im Unterschied zu gesprochener Sprache als monomodal, obgleich visuelle Gestaltungsmöglichkeiten „zur Übermittlung, Gliederung und ergonomischen Rezeptionserleichterung bei immens wachsenden Informationsmengen“ (U. Schmitz 2016, S. 329) durch Typografie, Layout und Textdesign genutzt werden. Zudem werden „herkömmliche Texte immer mehr mit anderen Botschaftsträgern und Modi verbunden“ (U. Schmitz 2016, S. 329). Schmitz führt in diesem Zusammenhang eine ganze Reihe von Textlinguistinnen an, die multimodale Elemente oder Einbettungen berücksichtigen (vgl. U. Schmitz 2016, S. 330). Bateman, Wildfeuer und Hiippala machen ihrerseits deutlich, dass das Konzept der Multimodalität auf der Vorstellung basiert, dass Ausdrucksformen in einer kommunikativen Situation nie alleine vorkommen. Kommunikationsmodi treten stets gemeinsam auf und Bedeutung wird aus dieser Kombination konstruiert, nicht aus den Ausdrücken selbst (vgl. Bateman, Wildfeuer und Hiippala 2017, S. 8, 17). Haapaniemi hebt diesen Sachverhalt und damit die Bedeutung des Konzepts der Multimodalität für die Translation hervor (vgl. Haapaniemi 2024, S. 29; weitere Informationen zu Haapaniemis Ausführungen zum Konzept der Materialität finden sich in Abschnitt 3.4.6).

⁸Schmitz weist explizit darauf hin, dass die akustische und optische Umsetzung, also Gestaltung, für sich betrachtet keine Multimodalität begründet (vgl. U. Schmitz 2016, S. 333–334). Er betont: „Nur wenn die Gesamtbedeutung eines Textes durch Zusammenwirken mindestens zweier Modi erzeugt wird, ist er multimodal.“ (U. Schmitz 2016, S. 334)

- *Text*: siehe oben
- *Kotext, Kontext, Situation*: s. Abschnitt [3.4.6](#)
- *Kode*: „ein System vereinbarter Zeichen (z. B. türkische vs. deutsche Standardsprache, Dialekt, Kleiderordnung)“ (U. Schmitz [2016](#), S. 332)
- *Modus*: „die Art der Präsentation und Wahrnehmung [...] insbesondere gesprochene Sprache, geschriebene Sprache, stehendes Bild (z. B. als Foto), bewegte Bilder (z. B. als Film) sowie Audio (z. B. Musik)“ (U. Schmitz [2016](#), S. 332)
- *Material*: „der Werkstoff, der zu Zeichen geformt wird oder sie trägt (z. B. Tinte oder Pixel bzw. Papier und Glas)“ (U. Schmitz [2016](#), S. 333)
- *Medium*: „ein technisches Hilfsmittel der Kommunikation (z. B. ein Kugelschreiber oder ein Smartphone)“ (U. Schmitz [2016](#), S. 333)
- *Kommunikationsform*: „der durch mediale Merkmale geprägte, kommunikativ-funktional aber nicht festgelegte Rahmen, innerhalb dessen kommuniziert wird (z. B. Telefongespräch, Fernsehsendung, Chat, Brief)“ (U. Schmitz [2016](#), S. 333; vgl. Brinker, Pappert und Cölfen [2024](#), S. 148)
- *Textsorte*: s. Abschnitt [3.2](#)

Wenn also Göpferichs Textdefinition ergänzt wird durch Schmitz' Angabe, dass für Multimodalität zwei der fünf Modi nach Schmitz' Definition vorhanden sein müssen, schließt dies die obigen Erläuterungen zu Materialität und Medialität implizit mit ein. Man könnte also den prototypischen multimodalen Text als ein Zeichengebilde mit mindestens einem sprachlichen Zeichen betrachten, das thematisch und/oder funktional orientiert und kohärent ist, eine erkennbare kommunikative Funktion erfüllt, eine abgeschlossene Einheit bildet und mindestens zwei Modi nutzt.

3.2 Textsorte

Bereits im vorangegangenen Abschnitt wurde das Merkmal *Textsorte* als Teil von Schmitz' multimodaler Texttypologie (vgl. U. Schmitz [2016](#), S. 333) aufgeführt. Im Rahmen dieser Texttypologie verweist Schmitz für die Definition des Begriffs *Textsorte* auf Brinker et al., die diesen als „komplexe Muster sprachlicher Kommunikation [...], die innerhalb einer Sprachgemeinschaft im Laufe der historisch-gesellschaftlichen Entwicklung aufgrund kommunikativer Bedürfnisse entstanden sind“ (Brinker, Pappert und Cölfen [2024](#), S. 160), definieren. Textsorten sind nach Brinker et al. darüber hinaus „immer an eine bestimmte (dominierende) kommunikative Funktion (die Textfunktion) geknüpft“ (Brinker, Pappert und Cölfen [2024](#), S. 149), „z. B. Liebesbrief,

Mahnschreiben, Hirtenbrief“ (U. Schmitz 2016, S. 333). Reiß und Vermeer betonen in ihrer Definition ebenfalls den Aspekt der Kommunikation. Sie definieren Textsorten als „überindividuelle Sprech- und Schreibakttypen [...], die an wiederkehrende Kommunikationsakte gebunden sind und bei denen sich eben aufgrund ihres wiederholten Auftretens in gleichartigen Kommunikationskonstellationen charakteristische Sprachverwendungs- und Textgestaltungsmuster herausgebildet haben“ (Reiß und Vermeer 1984, S. 149). Beaugrande und Dressler setzen Textsorten ergänzend in Zusammenhang mit ihrem Kriterium der Intertextualität, das für die Entwicklung von Textsorten „als Klassen von Texten mit typischen Mustern von Eigenschaften verantwortlich“ (De Beaugrande und Dressler 1981, S. 13) sei.⁹

Laut Brinker et al. lassen sich Textsorten als „typische Verbindungen von kontextuellen (situativen), kommunikativ-funktionalen und strukturellen (grammatischen und thematischen) Merkmalen beschreiben“ (Brinker, Pappert und Cölfen 2024, S. 145), den sogenannten Textsortenkonventionen. Diese dienen als Erkennungssignale, als Auslöser von Erwartungshaltungen und als Steuerungssignale für das Textverstehen (vgl. Reiß und Vermeer 1984, S. 189). Dadurch, dass sie den an der Kommunikation Beteiligten Orientierung für Textproduktion und -rezeption geben, erleichtern sie die Kommunikation (vgl. Brinker, Pappert und Cölfen 2024, S. 145). Kommunikation und Konvention gehen also Hand in Hand: „Text ist als Phänomen des kommunikativen Handelns letzten Endes eine Konventionskategorie: Irgendetwas wird von den an einer kommunikativen Handlung Beteiligten als Text (sowie auch als Exemplar einer besonderen Textsorte) interpretiert“ (Kvam u. a. 2024, S. 13–14).

Die hier angeführten Definitionen des Begriffs *Textsorte* beziehen sich zwar explizit auf sprachliche Kommunikation und setzen insbesondere bei den älteren Veröffentlichungen ein herkömmliches Verständnis des Begriffs *Text* voraus, der Bezug zu Schmitz' multimodaler Texttypologie, die das Merkmal *Textsorte* umfasst (vgl. U. Schmitz 2016, S. 333), zeigt aber, dass die *Textsorte* neben der Modalität auch ein Merkmal multimodaler Texte ist.

Zusammenfassend lässt sich also sagen, dass Textsorten konventionalisierte komplexe Kommunikationsmuster mit einer bestimmten kommunikativen Funktion sind, die sich innerhalb einer Sprachgemeinschaft im Lauf der Zeit gebildet haben und die mehrere Modi aufweisen können.

⁹Adamzik diskutiert Aspekte der Verwendung des Begriffs *Textsorte* und verwandter Begriffe sowie damit verbundener Probleme auf Basis der Literatur zum Thema (vgl. Adamzik 2016, S. 327–334), wobei sie hier ebenso wie bei der Beschäftigung mit dem Begriff *Text* auf die Möglichkeit einer Herangehensweise mit Prototypen aufmerksam macht (vgl. Adamzik 2016, S. 327).

3.3 Fachtext

Bevor Fachtexte definiert werden können, muss das Thema Fachsprache angesprochen werden, wobei Arntz et al. herausstellen, dass es „eine größere Zahl unterschiedlicher Fachsprachen“ (vgl. Arntz, Picht und K.-D. Schmitz 2014, S. 11) gibt, da es auch viele Fachgebiete mit jeweils eigenen Fachsprachen gibt. Fachsprache bildet die Basis für Fachtexte, allerdings wird die Definition von Fachsprache als schwierig angesehen. Eine beispielsweise auch von Arntz et al. angewendete Methode ist die Gegenüberstellung zur Gemeinsprache (vgl. Arntz, Picht und K.-D. Schmitz 2014, S. 21–25). Diese halten sie zwar auch nicht für unproblematisch, es drängen sich ihrer Ansicht nach aber keine alternativen Herangehensweisen auf (vgl. Arntz, Picht und K.-D. Schmitz 2014, S. 21–22). Ausgangspunkt ist für sie, dass die Fachsprache „nicht ohne Gemeinsprache denkbar“ (Arntz, Picht und K.-D. Schmitz 2014, S. 22) ist. Umgekehrt haben Fachsprachen auch Einfluss auf die Gemeinsprache, da verschiedene Gebiete von Wissenschaft und Technik im Alltag eine Rolle spielen (vgl. Arntz, Picht und K.-D. Schmitz 2014, S. 22–23).¹⁰ Auch Göpferich sieht die Einteilung in Gemeinsprache und Fachsprache problematisch, weil sie sich einer wissenschaftlichen Definition entziehe (vgl. Göpferich 1995, S. 28). Sie ist der Ansicht, dass sämtliche Texte beides enthalten, sowohl fachsprachliche Merkmale als auch nicht fachsprachenspezifische. Konkret sei sogar möglich, dass letztere auf untypische Weise in sehr fachspezifischen Texten enthalten sind (vgl. Göpferich 1995, S. 29). Sie ergänzt:

Welche als fachsprachlich markierte Einheiten mit welchen sprachlichen Einheiten anderer Fachsprachlichkeitsgrade oder Abstraktionsgrade oder gar mit sprachlichen Einheiten, die – isoliert betrachtet – fachsprachenuntypisch sind, kombiniert werden können, dürfte primär von der jeweiligen Textsorte abhängen. (Göpferich 1995, S. 29)

Eine scharfe Trennung zwischen eher fach- und eher gemeinsprachlichen Texten sieht sie somit nicht (vgl. Göpferich 1995, S. 29).

Aus diesem Grund werden Fachsprachen häufig als aus der Gemeinsprache entstanden und als Subsysteme der Gemeinsprache angesehen (vgl. Stolze 2013, S. 43).¹¹ In Fachtexten werden gemeinsprachliche und fachsprachliche

¹⁰ Ausführlicher wird diese Thematik von Stolze dargestellt (vgl. Stolze 2013, S. 47–49).

¹¹ Weitere Informationen dazu, wie verschiedene Fachsprachen einerseits sowie Gemeinsprache und Fachsprache andererseits in Verbindung stehen, finden sich beispielsweise bei Arntz et al. (vgl. Arntz, Picht und K.-D. Schmitz 2014, S. 11–25). Roelcke befasst sich ausführlich mit horizontalen und vertikalen Gliederungen von Fachsprachen (vgl. Roelcke 2020, S. 41–56).

Merkmale kombiniert (vgl. Stolze 2013, S. 45). Fachsprachliche Merkmale können dabei auf Lexemebene die Fachterminologie sowie auf weiteren Ebenen „bestimmte sprachlich-stilistische Besonderheiten (Syntagmen, Sätze, Textsorten)“ (Stolze 2013, S. 46) sein.

Damit wird deutlich, dass sich die Fachsprachlichkeit eines Textes oder einer Textsorte anhand der eingesetzten fachsprachlichen Merkmale überprüfen lässt, die je nach Textsorte unterschiedlich sein können. Die bereits von Stolze genannte Fachterminologie betrachten Hansen-Schirra et al. zusammen mit Abkürzungen auf der lexikalischen Ebene als das offensichtlichste Fachsprachenmerkmal (vgl. Hansen-Schirra, Hansen u. a. 2009, S. 110). Ebenso heben sie den ausgeprägten Nominalstil hervor, der beispielsweise zur verstärkten Verwendung von Substantiven, Kompositabildung und Nominalisierungen führe. Daneben zählen sie auch weitere Aspekte wie inhaltsarme Verben im Präsens auf, und im Bereich der Grammatik nennen sie „komplexe Nominalphrasen, Funktionsverbgefüge, komplexe Attribuierung sowie Infinitiv- und Partizipialsätze“ (Hansen-Schirra, Hansen u. a. 2009, S. 110).

Arntz et al. betonen zudem die Bedeutung der Terminologisierung, bei der gemeinsprachlichen Wortformen neue Begriffsinhalte zugewiesen werden, häufig über metaphorsische Prozesse, bei denen Ähnlichkeiten eine Rolle spielen (vgl. Arntz, Picht und K.-D. Schmitz 2014, S. 22). Albrecht führt drei Stufen der Terminologisierung auf:

- Im Fachterminus erfolgt eine Präzisierung der gemeinsprachlichen Bedeutung.
- Der Fachterminus weist gegenüber der gemeinsprachlichen Bedeutung eine eingeschränkte oder leicht verschobene Bedeutung auf.
- Der Fachterminus erhält „einen von der gemeinsprachlichen Bedeutung deutlich abweichenden Inhalt, meist auf dem Wege der Metaphorisierung“ (Albrecht 2013, S. 280).

Zudem zählt er als weitere wichtige Quellen für Fachtermini verschiedene Arten von Entlehnung, wie Lehnwort, Fremdwort, Lehnübersetzung, Lehnbedeutung und Lehn schöpfung, auf (vgl. Albrecht 2013, S. 280).

Welche der hier genannten fachsprachlichen Merkmale die Textsorte GUI-Text auszeichnen, wird in Abschnitt 4.2 erörtert. Davor sollen allerdings im Folgenden noch relevante Aspekte des Themenbereichs Kontext vorgestellt werden.

3.4 Kontext

Zur Forschung im Bereich Kontext sowohl allgemein als auch im Bereich Translationswissenschaft haben Aschenberg, Baker, Gambier und House Überblicke veröffentlicht (Aschenberg 1999, Baker 2006, Gambier 2021 und House 2006), in denen sie die Unterschiede der verschiedenen Denkschulen herausarbeiten, die sich mit dem Begriff *Kontext* befassen.

House unterscheidet in ihrer Abhandlung zu Text und Kontext in der Übersetzung die philosophische, die psychologische, die pragmatische, die soziolinguistische, anthropologische und konversationsanalytische und zuletzt die funktional-pragmatische und die systemisch-funktionale Tradition, die sie jeweils zusammenfassend darstellt (vgl. House 2006, S. 338–343). Aschenberg erweitert dies dadurch, dass sie die Beschäftigung verschiedener linguistischer Schulen mit dem Begriff *Kontext* hervorhebt (vgl. Aschenberg 1999, S. 7). Eine Ergänzung dazu bieten Gambier und Lautenbacher an, indem sie ebenfalls auf die Beschäftigung verschiedener Disziplinen mit dem Begriff *Kontext* sowie zahlreiche Veröffentlichungen und Journale in diesen Bereichen hinweisen (vgl. Gambier und Lautenbacher 2024, S. 6–7). Dabei stellen sie insbesondere die Vieldeutigkeit des Begriffs *Kontext* heraus:

Context often coexists with macro-/micro-context, (sociocultural) situation, (working) setting, circumstances, environment, landscape, location, zone, (institutional) reality, etc. This shows both the semantic ambiguity and polysemy of the term, covering aspects ranging from geopolitics to psychology. (Gambier und Lautenbacher 2024, S. 7)

Diese Menge der verschiedenen Ansätze und Arbeitsfelder mit ihren unterschiedlichen Akzentuierungen macht Aschenberg dafür verantwortlich, dass es im Rahmen der Sprachwissenschaft keine allgemein akzeptierte Definition des Begriffs *Kontext* gibt (vgl. Aschenberg 1999, S. 7).

Baker macht ergänzend deutlich, dass der Begriff *Kontext* sehr häufig in der Literatur zum Übersetzen und Dolmetschen genannt, aber nur sehr selten einer genauen Prüfung unterzogen wird (vgl. Baker 2006, S. 321, 322, 332). Sie selbst stellt unterschiedliche Perspektiven auf das Thema *Kontext* aus der Literatur zu Pragmatik und linguistischer Anthropologie vor. Dabei befasst sie sich mit den Unterschieden zwischen kognitiven einerseits und sozialen bzw. interaktiven Definitionen von Kontext andererseits, betrachtet statische und dynamische Kontextmodelle und diskutiert neutrale und machtsensitive Definitionen von Kontext, alles jeweils primär im Hinblick auf das Dolmetschen (vgl. Baker 2006, S. 322–332).

Wie auch Baker stellt House ebenfalls die Notwendigkeit einer Konkretisierung der Auffassung von Kontext speziell für die pragmatische Betrachtung von linguistischen Äußerungen heraus:

A pragmatic framework would then need to include a general representation of contextual features that determine the values of linguistic expressions, with context being represented by a body of information presumed to be available to the participants in the speech situation. Given the need to specify context as features of this situation, a distinction must be made between actual situations of utterance in all their manifold variety and the selection of only those features that are linguistically and socio-culturally relevant for both the speaker producing a particular utterance and the hearer who interprets it. (House 2006, S. 340–341)

Melby und Foster ergänzen dies, indem sie feststellen, dass der Begriff *Kontext* eine Definition vermissen lässt, die in der täglichen Arbeit professioneller Übersetzerinnen genutzt werden kann (vgl. Melby und Foster 2010, S. 1).

Aschenberg geht erste Schritte in Richtung einer Konkretisierung, indem sie drei „elementare Kontexttypen“ (Aschenberg 1999, S. 9) unterscheidet, die Situation, den sprachlichen Kontext und das Sprecherwissen, die auch schon in Houses Erläuterungen oben erkennbar sind und die Aschenberg in Anlehnung an Beaugrande und Dressler als notwendige Bedingungen textueller Sinnbildung (vgl. Aschenberg 1999, S. 9) bezeichnet (vgl. dazu auch Krüger 2013, S. 305). Ähnlich nennt Klos drei Kontextgrößen, die *situative* Dimension, die „Angaben über den Sprecher, den oder die Hörer, den Zeitpunkt und den Ort einer Äußerung“ (Klos 2011, S. 171) umfasst, die *sozial-kognitive* Dimension, die „u.a. das Hintergrundwissen der GesprächsteilnehmerInnen betrifft und je nach sozialer Konstellation sehr unterschiedlich ausfallen kann“ (Klos 2011, S. 171), und die *linguistische* Dimension, der „alle relevanten Inhalte, die innerhalb der Interaktion oder des Textes zu finden sind“ (Klos 2011, S. 171), umfasst.

Im Folgenden wird deutlich werden, dass die Situation und das Sprecherwissen, die von Aschenberg und Klos unterschieden werden, in der Literatur in vielen Fällen nicht bzw. nicht klar unterschieden werden. Sehr wohl allerdings wird in den verschiedenen Kontextdefinitionen in der Literatur jeweils eine weitere Unterscheidung klar identifiziert, nämlich die zwischen außersprachlichem (Abschnitt 3.4.2) und sprachlichem Kontext (Abschnitt 3.4.3). Für beides werden jedoch unterschiedlichen Bezeichnungen genutzt und beide lassen sich wiederum weiter unterteilen. Zunächst

finden sich aber im folgenden Abschnitt Informationen zur Frage, welche Relevanz der Kontext für die Bedeutung von Texten hat.

3.4.1 Kontext und Bedeutung

Aschenberg weist in ihrem Beitrag auf den Konsens in der Translationswissenschaft hin, dass nicht Wörter, sondern Texte übersetzt werden (vgl. Aschenberg 1999, S. 9). Kußmaul bestätigt dies, indem er ausführt, dass die Bedeutung von Wörtern nur innerhalb eines Textes klar werde (vgl. Kußmaul 2015, S. 28). Diesen Gedanken erläutert er wie folgt: „Die Bedeutungserschließung von Wörtern in Texten geschieht grundsätzlich aus zwei Richtungen: (a) durch das im Wort enthaltene Bedeutungspotential und (b) über den bedeutungsdeterminierenden Kontext.“ (Kußmaul 1989, S. 107) Das Bedeutungspotenzial der Wörter lässt sich nach seiner Ansicht über die Nutzung von Wörterbüchern recherchieren (vgl. Kußmaul 1989, S. 107), wobei Definition und Beispiele aus dem Wörterbuch in der Regel in den Text passen und durch den Kontext bestätigt werden (vgl. Kußmaul 1989, S. 107; vgl. hierzu auch Kußmaul 1995, S. 105).

Ebenso betonen Kvam et al. die Relevanz des Kontexts für die Bedeutung von Texten:

Der Kontext und der damit verbundene Prozess Kontextualisierung ist dabei als Bedeutungskonstitution zu verstehen. [...] Ein Text kann somit nicht ohne Kontext und den Prozess einer dahin führenden Kontextualisierung existieren. (Kvam u. a. 2024, S. 15–16).

Klos führt ergänzend dazu aus, dass innerhalb der Pragmatik weitgehend unbestritten sei, dass „die Interpretation einer sprachlichen Äußerung immer vor dem Hintergrund ihrer kontextuellen Einbettung erfolgt“ (Klos 2011, S. 170). Der Kontext sei ein wichtiger Referenzpunkt für die Bedeutungsbestimmung und diene zur Ermittlung des kommunikativen Sinns eines sprachlichen Ausdrucks (vgl. Klos 2011, S. 170).

3.4.2 Außersprachlicher Kontext

Vermeer subsummiert den sprachlichen und den außersprachlichen Kontext unter dem Terminus *Situation*, unter dem „alle Gegebenheiten außerhalb des aktualisierten Sprachzeichens selbst verstanden werden (also auch umgeben-

de Sprachzeichen=,Kontext‘), die bewusst¹² oder unbewußt auf das aktuelle Sprach-,Verhalten‘ einwirken“ (Vermeer 1972, S. 61). Das heißt, dass Vermeer die Situation in einen extralingualen Situationsteil und einen lingualen Kontextteil unterteilt (vgl. Vermeer 1972, S. 154).

Statt Vermeers Begriff *extralingualer Situationsteil* nutzt Kußmaul den Begriff *nonverbaler Kontext* für Kultur und Situation und damit „Bereiche, die unabhängig von der Sprache bestehen“ (Kußmaul 2015, S. 43). Stolze wiederum fasst unter dem Begriff *Kontext* die textexternen Merkmale, „die auf den Bereich Sender, Empfänger, außersprachliche Wirklichkeit und Mitteilungssituation verweisen“ (Stolze 1985, S. 23) zusammen. Stolze führt weiter aus:

Dazu gehört die Kommunikationssituation, auf die hin ein Text produziert ist, sowie der außersprachliche Kontext, der durch alle nichtsprachlichen Umstände konstituiert wird, die von den Sprechern entweder direkt wahrgenommen werden oder ihnen bekannt sind. Wir verstehen darunter die soziokulturelle Umgebung, die durch natürliche, historische, kulturelle, religiöse, soziale, wirtschaftliche, politische, technische, individuelle und okkasionelle Umstände gegeben ist. (Stolze 1985, S. 26–27)

Kvam et al. erkennen ebenfalls einen umfassenden außersprachlichen Kontextbezug. Nach ihnen spielt bei der „Rezeption eines Textes [...] der situative, soziale, historische, individuelle etc. Kontext für die Interpretation der Zeichen im jeweiligen Gewebe (und damit das Verständnis der Botschaft) eine zentrale Rolle“ (vgl. Kvam u. a. 2024, S. 15).

Im Unterschied zu Vermeers extralingualen Situationsteil und Kußmauls nonverbalem Kontext nutzt Reiß als Bezeichnung für die außersprachliche Situation den Begriff „situationeller Kontext“ oder auch „Situationskontext“ (vgl. Reiß 1986, S. 12, 70). Dieser umfasst nach ihrer Auffassung die außersprachlichen Faktoren (vgl. Reiß 1986, S. 70) oder auch die außersprachlichen Determinanten, genauer „den engeren Situationsbezug, den Sachbezug, den Zeitbezug, den Ortsbezug, den Empfängerbezug, die Sprecherabhängigkeit und die affektiven Implikationen“ (Reiß 1986, S. 71), die sie im Anschluss an diese Erläuterungen detailliert beschreibt.

Nach Kußmaul lässt sich im sprachwissenschaftlichen Sinne der Aspekt des Kontexts der umgebenden Wörter als Teil der Semantik und der Aspekt der Situation und Kultur als Teil der Pragmatik verorten (Kußmaul 2015,

¹²Bei direkten Zitaten aus älteren Veröffentlichungen wurde die damals gültige Schreibung und Zeichensetzung beibehalten, auch wenn sie nicht mehr den aktuellen Regeln entspricht.

S. 43). Beides trägt seiner Ansicht nach dazu bei, „das Gemeinte [zu] erschließen“ (Kußmaul 2015, S. 22). Speziell den starken Zusammenhang zwischen Pragmatik und Kontext betont auch House: „In the tradition of pragmatics, conceptualizations of context have played such an overridingly important role that the very definition of pragmatics is often bound up with the notion of context.“ (House 2006, S. 340) Sie erläutert, dass eine Theorie zum Zusammenhang zwischen linguistischen Äußerungen und dem, was damit ausgedrückt werden soll, den Kontext, d. h. in ihrem Sinne den außersprachlichen Kontext, betrachten muss, in dem diese Äußerungen getätigt werden. Umgekehrt beeinflusst der Inhalt auch den Kontext und der Kontext wiederum wirkt sich auf die syntaktische, semantische, lexikalische und phonologische Struktur der Sprache aus (vgl. House 2006, S. 340–341).

Auch Albrecht betont den Bezug von Kontext zur linguistischen Pragmatik. Diese beschäftigt sich nach seiner Aussage

[...] nicht nur – definitionsgemäß – mit dem Verhältnis von Zeichen und „Zeichenbenutzer“, sondern auch mit dem Verhältnis von tatsächlich Gesprochenem und dem, was zu dessen Verständnis hinzukommt: Sprechsituation, Hintergrundwissen von Sprecher und Hörer, kurz und gut mit dem „Kontext“ in einem sehr allgemeinen Sinn. (Albrecht 2013, S. 216)

An Beispielen zeigt Albrecht die verschiedenen Bedeutungen des Begriffs *Kontext* auf, indem er zwischen dem, „was im Text (im gesprochenen oder geschriebenen) selbst steht und zum Verständnis des jeweils untersuchten Ausdrucks beiträgt“ (Albrecht 2013, S. 216), und dem unterscheidet, „was nicht im Text selbst steht, sondern was aus der Situation, in der der Text geäußert wird, erschlossen werden kann“ (Albrecht 2013, S. 217). Den ersten Fall bezeichnet Albrecht als Kontext, den zweiten als Situation (vgl. Albrecht 2013, S. 217). Albrecht betont, dass diese Unterscheidung ursprünglich die gesprochene Sprache betrifft, weist aber gleichzeitig darauf hin, dass sie auch für die geschriebene Sprache und damit auch für das Übersetzen sinnvoll ist (vgl. Albrecht 2013, S. 217). Für ihn hat die Situation, also der außersprachliche Kontext, in der geschriebenen Sprache allerdings einen anderen Status als in der gesprochenen Sprache:

Sie erscheint in diesem Fall als „suppletiver Kontext“, als eine Art von Kontext zweiten Grades. Geschriebene Texte enthalten eine Menge von Angaben, die in gesprochenen Äußerungen fehlen, weil sie dort überflüssig wären. Mit diesen zusätzlichen Informationen wird die Situation künstlich aufgebaut, in die die Äußerungen des

„wirklichen Lebens“ stets eingebettet sind. Praktische Fachtexte (Bedienungsanleitungen, Anweisungen zur Einnahme eines Medikaments usw.) weisen ebenso wie literarische Texte große Unterschiede bezüglich der Konstruktion eines solchen Verstehenshorizontes auf. (Albrecht 2013, S. 217)

Den außersprachlichen Kontext betreffend könnte man also zusammenfassend sagen, dass er zur Pragmatik zählt, für geschriebene Texte von geringerer Bedeutung ist, verschiedene Bezeichnungen für ihn vorliegen und inhaltlich zahlreiche auch unterschiedliche Aspekte damit verbunden werden. Baker, deren Aufsatz (vgl. Baker 2006) sich ausschließlich mit dem außersprachlichen Kontext befasst, stellt diesen letzten Punkt explizit heraus:

We could, for instance, spend considerable time trying to enumerate all the possible elements that constitute context in any type of translational event and the various ways in which these elements constrain the behavior (linguistic and non-linguistic) of translators and interpreters. But this approach can only take us so far. (Baker 2006, S. 332)

Für die empirische Untersuchung des Kontexts bringt dieser Sachverhalt große Schwierigkeiten mit sich, da für die Operationalisierbarkeit geklärt sein muss, welcher Satz von Aspekten relevant ist, um deren Bedeutung zu untersuchen (siehe Abschnitt 3.4.8).

3.4.3 Sprachlicher Kontext, Kotext

Wie bereits in Abschnitt 3.4.2 deutlich wurde, gibt es Autorinnen, die den *sprachlichen* oder auch linguistischen Kontext mit dem Begriff *Kontext* bezeichnen, wie beispielsweise Albrecht (vgl. Albrecht 2013, S. 217), und andere, die den *außersprachlichen* Kontext als *Kontext* bezeichnen, wie beispielsweise Stolze (vgl. Stolze 1985, S. 23). Andere nutzen Adjektive zur Unterscheidung, wie Kußmaul, der die Begriffe *nonverbaler Kontext* und *verbaler Kontext* (vgl. Kußmaul 2015, S. 43) vorschlägt. Eine weitere Gruppe von Autorinnen arbeitet basierend auf einer Prägung von Catford mit dem Begriff *Kotext*, wenn sie vom sprachlichen Kontext redet (vgl. Catford 1978, S. 31). Catford schreibt: „By *context* we mean ‚context of situation‘, i.e. those elements of the extra-textual situation which are related to the text as being linguistically relevant: hence *contextual*. By *co-text* we mean items in the *text* which accompany the item under discussion: hence *co-textual*.“ (Catford 1978, S. 31; Hervorhebungen im Original).

Die Bezeichnung *Kotext* findet sich beispielsweise bei Schmitt, der damit das textinterne Umfeld im Unterschied zur situativen Einbettung des Texts (für ihn der Kontext) bezeichnet (vgl. Schmitt 2006, S. 81) und ebenso bei Göpferich, die den Textzusammenhang (bei ihr der Kotext) von der außersprachlichen Situation (bei ihr der Kontext) unterscheidet (vgl. Göpferich 2008, S. 57). Für Raasch ist ganz ähnlich der Kotext „die sprachliche Umgebung einer sprachlichen Äußerung“ (Raasch 1984, S. 112).¹³ Stolze, die die textexternen Merkmale als *Kontext* bezeichnet, betrachtet die textinternen Merkmale, „die auf den Bereich Sprachstruktur verweisen“ (Stolze 1985, S. 23), ebenso als Kotext.¹⁴ Weiter schreibt sie:

Mit J. C. Catford [...] bezeichnen wir die Verknüpfung syntaktischer und semantischer Elemente untereinander, aus denen der Text aufgebaut ist, als *Kotext*. Aufgrund der wechselseitigen Abhängigkeit von textinternen und textexternen Merkmalen berücksichtigt eine vollständige Textbeschreibung immer beide Kriterien. Damit orientiert sich unsere Textbestimmung an der semiotischen Triade der syntaktischen, semantischen und pragmatischen Dimensionierung, die einem jeden Text zu eigen ist. (Stolze 1985, S. 23; Hervorhebung im Original)

Damit fügt sie Kußmauls Differenzierung in Semantik und Pragmatik auch noch die Syntax hinzu. Eine weitere Unterteilung des sprachlichen Kontexts in Mikro- und Makrokotext nimmt Reiß vor:

Beide sind keine exakt und definitiv abgrenzbaren Einheiten. Sie variieren je nach sprachlicher und gedanklicher Einbettung des jeweils zu übertragenden Elements. Immerhin dürfte der Mikrokotext im allgemeinen nur die in unmittelbarer Nachbarschaft stehenden Wörter umfassen und nur selten über die Satzeinheit hinausgehen, während der Makrokotext vom Abschnitt bis zum Textganzen reichen kann. (Reiß 1986, S. 58)

Klos betont ergänzend die Tatsache, dass der Kotext „unabhängig von textexternen, situativen Faktoren gedacht werden kann“ (Klos 2011, S. 172)

¹³Für Raasch ist der analog der Kontext „der kommunikative Rahmen, in den eine sprachliche Äußerung eingebettet ist“ (Raasch 1984, S. 112).

¹⁴Auch Reiß verweist auf Catford, lehnt dessen Bezeichnung „co-text“ allerdings aus praktischen Gründen ab. Zu leicht käme es bei Tippfehlern zu Verwirrung. Außerdem sei in der deutschsprachigen Literatur zu Übersetzungsfragen mit „Kontext“ primär der sprachliche Kontext gemeint (vgl. Reiß 1986, S. 56).

und dass durch ihn „Deutungen, die in anderen thematischen Zusammenhängen denkbar wären“ (Klos 2011, S. 172), unter Umständen überlagert werden. Daraus schließt sie Folgendes:

[...] der Kotext einer sprachlichen Einheit [dient] als erste Interpretationsressource, denn er kontextualisiert sprachliche Äußerungen und Benennungseinheiten ganz automatisch. Während die Konzeption von *Kontext* ein sehr umfassendes Phänomen ist, halte ich den *Kotext* aus formaler und struktureller Sicht für greifbarer: Das Hintergrundwissen zwischen einzelnen Sprachteilnehmern mag stark voneinander abweichen, sofern allen Sprachbenutzern jedoch dieselbe textuelle Basis vorliegt, sollte das kotextuelle Wissen vergleichsweise kongruent sein. (Klos 2011, S. 171–172; Hervorhebungen im Original)

Der sprachliche Kontext befasst sich also mit der semantischen und syntaktischen Ebene. Auch hier zeigen sich verschiedene Bezeichnungen und eine gewisse definitorische Unklarheit, insbesondere in Bezug auf die Frage, was genau die sprachliche Umgebung ausmacht. Nur Reiß gibt mit ihrer Unterscheidung in Mikro- und Makrokontext eine klarere Antwort auf diese Frage. Allerdings macht die von Klos genannte Greifbarkeit und vergleichsweise Kongruenz des Wissens der Sprachbenutzerinnen zum sprachlichen Kontext diesen zu einer idealen Ansatzstelle für die Operationalisierung bei empirischen Untersuchungen, wie in Abschnitt 3.4.8 weiter ausgeführt wird.

Neben den in diesem und im vorangegangenen Abschnitt beschriebenen Herangehensweisen zum Thema *Kontext* lassen sich noch zwei weitere Systeme aufführen, die sich von diesen sowohl begrifflich als auch teils inhaltlich in einigen Aspekten unterscheiden und jeweils relevante Aspekte für die weitere Argumentation bieten. Daher werden sie im Folgenden etwas ausführlicher separat vorgestellt. Zum einen ist dies das Modell von Bühler, das später von Coseriu erweitert wurde und sich eher mit dem außersprachlichen Kontext befasst (s. Abschnitt 3.4.4), und zum anderen die Nomenklatur von Melby und Foster, die sich primär mit dem sprachlichen Kontext befasst und im Abschnitt 3.4.5 vorgestellt wird.

3.4.4 Bühler und Coseriu

Bühler nähert sich dem Kontextbegriff über den Begriff *Umfeld*, das er in das sympraktische, das symphysische und das synsemantische Umfeld unterscheidet (vgl. Bühler 1999, S. 154–168). Das sympraktische Umfeld entspricht dabei dem außersprachlichen Kontext, das synsemantische Umfeld

dem sprachlichen Kontext. Das symphysische Umfeld bezeichnet die „dingliche Anheftung“ (Bühler 1999, S. 159), also den Fall, dass sprachliche Zeichen physisch mit einem Gegenstand verbunden sind (vgl. Bühler 1999, S. 159).

Coseriu differenziert diese Unterscheidung weiter aus, indem er den Begriff *Umfeld* weiter unterteilt. Für ihn sind die Kategorien *Situation*, *Region*, *Kontext* und *Redeuniversum* von Belang. Bei der Situation „geht es ausschließlich um die Umstände und Beziehungen in Zeit und Raum, die durch das Sprechen selbst entstehen“ (Coseriu 2007, S. 126). Unter *Region* versteht Cosieru „den Raum, der ein in bestimmten Bedeutungssystemen funktionierendes Zeichen einschließt“ (Coseriu 2007, S. 121). Begrenzt wird die Region durch Tradition und Kenntnisse der Sprecherinnen (vgl. Coseriu 2007, S. 121). Als *Kontext* betrachtet Coseriu „die gesamte ein Zeichen umgebende Wirklichkeit“ (Coseriu 2007, S. 127), die aus Zeichen oder aus Nichtzeichen bestehen kann (vgl. Coseriu 2007, S. 127–128). Das Redeuniversum ist für ihn „das universelle System von Bedeutungen, zu dem ein Text gehört und durch das er seine Gültigkeit und seinen besonderen Sinn erhält“ (Coseriu 2007, S. 128).

Den Kontext selbst wiederum unterscheidet er in einzelsprachlichen Kontext („die Sprache selbst, in der gesprochen wird“), Rede-Kontext („den Text selbst als ‚Umfeld‘ eines jeden seiner Teile“) und Außer-Rede-Kontext, „der durch alle nicht-sprachlichen Umstände konstituiert wird, die von den Sprechern entweder direkt wahrgenommen werden oder ihnen bekannt sind“ (Coseriu 2007, S. 128). Letzterer wird von Coseriu nochmals in folgende Unterkategorien unterschieden:

- Physikalischer Außer-Rede-Kontext: entspricht ungefähr Bühlers symphysischem Umfeld (vgl. Coseriu 2007, S. 130)
- Empirischer Außer-Rede-Kontext: „Gegenstände und Sachverhalte, die den Kommunikationspartnern zum Zeitpunkt und am Ort des Sprechens bekannt sind“ (Coseriu 2007, S. 130)
- Natürlicher Außer-Rede-Kontext: „die ganze uns bekannte Welt als Kontext des Sprechens“ (Coseriu 2007, S. 130)
- Praktischer/Okkassioneller Außer-Rede-Kontext: „die ‚Gelegenheit‘ des Sprechens [...]. Bestimmend für diese Art des Kontexts sind z.B. Gesprächspartner und die genauen räumlichen und zeitlichen Umstände des Sprechens“ (Coseriu 2007, S. 132); entspricht damit in etwa dem sympraktischen Umfeld bei Bühler (vgl. Albrecht 2013, S. 223)
- Historischer Außer-Rede-Kontext: „die Gesamtheit der ‚historischen‘ Umstände, die den Sprechern bekannt sind“ (Coseriu 2007, S. 132)
- Kultureller Außer-Rede-Kontext: „all das, was zur kulturellen Tradition einer Gemeinschaft gehört“ (Coseriu 2007, S. 133)

Es zeigt sich also, dass Bühler dem Kontextbegriff mit dem symphysischen

Umfeld eine weitere Dimension hinzufügt, die in den vorangegangenen Definitionen keine Rolle gespielt hat. Albrecht erläutert, dass ein symphysisches Umfeld vorliegt, „wenn ein sprachliches Zeichen in unmittelbarem physischen Kontakt mit einem Gegenstand erscheint“ (Albrecht 2013, S. 220). Dafür nennt er folgende Beispiele:

Den Aufdruck *Ein schwarzes Schaf* auf einem Buch wird man nicht dahingehend interpretieren, daß man in dem Buch ein schwarzes Schaf zu sehen habe, sondern daß darin über ein schwarzes Schaf berichtet wird. Wenn auf einer Plastikpackung mit Salat in einem französischen Supermarkt zu lesen ist: *Déjà lavée. Prête à l'emploi*, so beziehen sich die weiblichen Formen auf etwas, das im Text gar nicht vorkommt, das jedoch deutlich sichtbar in der Packung enthalten ist: *la salade*. (Albrecht 2013, S. 220; Hervorhebungen im Original)

Coserius Differenzierungen sorgen hingegen insbesondere für eine weitere deutliche Auffächerung des außersprachlichen Bereichs des Kontexts und zeigen somit dessen Komplexität und Umfang, aber damit auch die Schwierigkeiten der Operationalisierung auf deutlichere Weise.

3.4.5 Ko-Text, Chron-Text, Rel-Text, Bi-Text und Nicht-Text

Melby und Fosters Fokus ist im Unterschied zu dem von Coseriu der sprachlich-textliche Bereich, den sie in Bezug zur professionellen Übersetzungspraxis setzen, nachdem sie wie oben bereits erwähnt als Ausgangspunkt ihrer Überlegungen feststellen, dass ihrer Meinung nach keine Kontextdefinition vorliegt, die für die tägliche Praxis professioneller Übersetzerinnen tauglich ist (vgl. Melby und Foster 2010, S. 1). Daher versuchen sie sich in ihrem Paper an einer differenzierteren und im professionellen Kontext ihrer Ansicht nach nutzbringenden Unterteilung und betonen, dass sie im Unterschied zu anderen Autorinnen, die eher die wissenschaftliche Analyse von Texten im Blick haben, die Produktion von Zieltexten und damit den Zweck in den Blick nehmen wollen (vgl. Melby und Foster 2010, S. 3).

In ihrer Darstellung sehen sie *Kontext* als Oberbegriff, den sie einerseits in die Kategorie *Text* mit den Unterkategorien *Ko-Text*, *Chron-Text*, *Rel-Text* und *Bi-Text* und andererseits in die Kategorie *Nicht-Text* unterteilen. Diese Kategorien sind ihrer Ansicht nach relevant für das Verstehen des Ausgangstexts und für die Produktion des Zieltexts (vgl. Melby und Foster 2010, S. 3). Im Fokus sollen ihrer Ansicht nach die Merkmale des Kontexts stehen, die

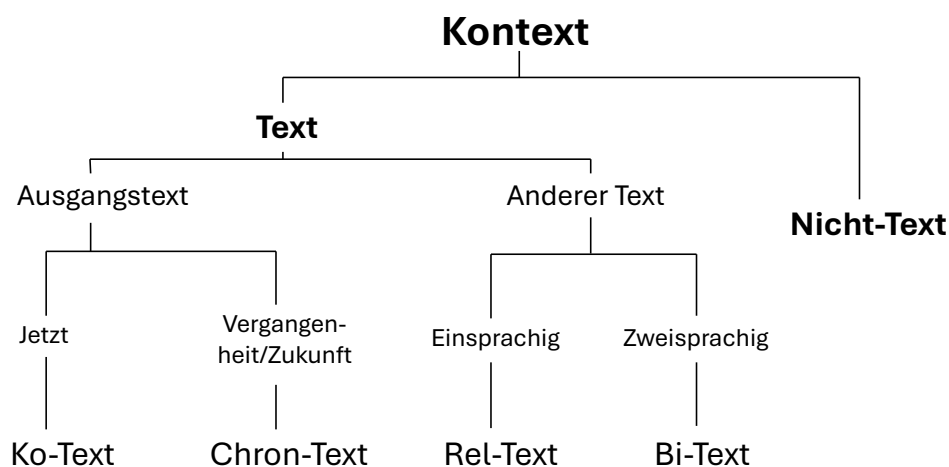


Abbildung 3.1: Kontextschema nach Melby und Foster (vgl. Melby und Foster 2010, S. 5)

sich auf die Bedeutung auswirken (vgl. Melby und Foster 2010, S. 4). Wie diese Kategorien zusammenhängen, lässt sich Abbildung 3.1 entnehmen, die auf einer Abbildung aus Melby und Fosters Veröffentlichung (vgl. Melby und Foster 2010, S. 5) basiert.

Die erste Kategorie, die sie definieren, ist der *Nicht-Text*¹⁵. Er umfasst für sie nichtlinguistische Variablen, wie die Verfasserin und deren Arbeitssituation sowie sie umgebende kulturelle Ereignisse, die zur Verdeutlichung ihrer Absichten beitragen können (vgl. Melby und Foster 2010, S. 4). Diese Kategorie fällt also in den oben genannten Bereich des außersprachlichen Kontexts (s. Abschnitt 3.4.2). Für sie schließt dies aber auch physische Objekte oder Ereignisse ein, die eine kulturelle oder sprachliche Bedeutung haben (vgl. Melby und Foster 2010, S. 7). Zur Verdeutlichung bringen sie folgendes Beispiel: „a translator will be better able to translate the documentation for a piece of machinery if first given the opportunity to operate that machine or at least see it in operation in person or through video.“ (Melby und Foster 2010, S. 7) Weitere Beispiele für Nicht-Text sind nach Melby und Foster Körpersprache, technisches Wissen und kulturelles Bewusstsein (vgl. Melby und Foster 2010, S. 7). In der Übersetzungspraxis muss ihrer Ansicht nach

¹⁵Nicht zu verwechseln mit dem in Abschnitt 3.1 von Beaugrande und Dressler definierten Nicht-Text als dem Fall, dass nicht sämtliche ihrer Textualitätskriterien erfüllt werden (vgl. De Beaugrande und Dressler 1981, S. 3)

der Nicht-Text als die Aspekte des Kontexts betrachtet werden, auf die im Verlauf eines Übersetzungsprojekts nicht über schriftliche Texte zugegriffen wird, die aber dennoch für die Arbeit der Übersetzerinnen relevant sind (vgl. Melby und Foster 2010, S. 8). Ebenso ist der Übersetzungsauftrag Teil des Nicht-Texts (vgl. Melby und Foster 2010, S. 11).

Alle anderen Kategorien bestehen demnach aus Text und gehören damit der zweiten Kategorie *Text* an. Dies sind der Ausgangstext selbst (Unterkategorie *Ko-Text*) sowie andere Texte. Diese anderen Texte wiederum können in einsprachige Dokumente, die für das Verständnis des Ausgangstexts und die Produktion des Zieltexts relevant sind (Unterkategorie *Rel-Text*) und zweisprachige Dokumente unterteilt werden, also Texte mit Übersetzungen (Unterkategorie *Bi-Text*) (vgl. Melby und Foster 2010, S. 4). Der Rel-Text besteht dabei aus zugehörigen Dokumenten und Ressourcen, wie z. B. einem einsprachigen Wörterbuch oder einsprachiger Terminologie (vgl. Melby und Foster 2010, S. 6). Ein typischer Bi-Text wäre ein Translation-Memory oder auch ein zweisprachiges Korpus, könnte aber auch ein zweisprachiges Glossar oder eine zweisprachige Termbank sein (vgl. Melby und Foster 2010, S. 7).

Vom Ko-Text selbst werden frühere und spätere Versionen desselben AT-Dokuments unterschieden, für die Melby und Foster die Unterkategorie *Chron-Text* einführen (vgl. Melby und Foster 2010, S. 4–5): „Co-text is synchronic while chron-text is diachronic.“ (Melby und Foster 2010, S. 6) Diese Definition konkretisieren sie wie folgt:

For our purposes, the co-text of a word or phrase is limited to surrounding text within a particular version of one document but not limited to the current sentence. Definitional text (text within a document that reveals the meaning of a lexical item or a term) will also be considered co-text. (Melby und Foster 2010, S. 6)

Deutlich wird bei diesem Modell wie von den Autorinnen erhofft, dass gewisse Aspekte aus der professionellen Übersetzungspraxis einfließen. Der außersprachliche Kontext wird in Bezug auf die Übersetzungspraxis etwas konkretisiert, bleibt aber dennoch weiterhin in Teilen unklar und lässt sich damit nur schwer operationalisieren. Zudem wird auch hier beim Ko-Text nicht klar herausgearbeitet, ob es sich nur um umgebende Textteile oder um den gesamten Ausgangstext handelt. Implizit liegt allerdings die Vermutung nahe, dass der gesamte aktuell zu übersetzende Text von Melby und Foster als Ko-Text betrachtet wird, da sie für ihr Modell in Anspruch nehmen, dass es umfassend ist (vgl. Melby und Foster 2010, S. 3–4) und ansonsten Teile des Ausgangstexts gar nicht im Modell enthalten wären. Melbys und Fosters oben zitierte Definition selbst bleibt allerdings eher schwammig, weil nur Teile des

Ausgangstexts explizit eingeschlossen werden, nämlich der umgebende Text und weitere Textteile, die zur Klärung der Bedeutung der aktuellen Textstelle beitragen (vgl. Melby und Foster 2010, S. 6).

3.4.6 Kontext im multimodalen Text

Bei der bisherigen Betrachtung fällt auf, dass die Multimodalität, die bereits zuvor in Abschnitt 3.1 als wichtiger Faktor für viele Texte herausgearbeitet wurde, im überwiegenden Teil der bisher genannten Kontextdefinitionen keine Rolle spielt. Der sprachliche Kontext zeigt sich in den Sprachzeichen (s. Abschnitt 3.4.3), der außersprachliche Kontext bezieht sich auf die Situation, nicht auf die physische oder mediale Präsentationsform (s. Abschnitt 3.4.2). Die einzige Ausnahme bildet Bühlers symphysisches Umfeld (vgl. Bühler 1999, S. 159), das in Abschnitt 3.4.4 dargestellt wird und das Bezug auf physische Aspekte eines Texts nimmt.

In der Literatur gibt es aber sehr wohl Beiträge, in denen die Relevanz von Multimodalität für den Kontext thematisiert wird. Beispielsweise definiert Schmitz im Rahmen seiner multimodalen Texttypologie *Kotext* als „die engere oder weitere textliche Umgebung eines Textstückes innerhalb eines Textes“ und *Kontext* als „die semiotische Umgebung des Textes“ (U. Schmitz 2016, S. 331, 332). Als Beispiel für ersteres nennt er Sätze, aber auch Abschnitte und für letzteres das gesamte Handbuch, in dem sein Beitrag erscheint. *Situation* ist für ihn „die lebensweltliche Umgebung, in deren Rahmen der jeweilige Text eine Rolle spielt“ (U. Schmitz 2016, S. 332), z. B. ein Seminar, in dem sein Beitrag diskutiert wird. Damit bewegt er sich im Rahmen der oben genannten Definitionsansätze für den Begriff *Kontext* und gleichzeitig passen seine Beispiele auf herkömmliche Texte. Allerdings nennt er für seine Kontextdefinition explizit die semiotische Umgebung, was mit Blick auf die weiteren Aspekte seiner multimodalen Texttypologie, die bereits in Abschnitt 3.1 aufgeführt wurden, sehr wohl Multimodalität einschließt. Insbesondere seine Textdefinition (vgl. U. Schmitz 2016, S. 331) sowie der von ihm genannte Aspekt *Modus* (vgl. U. Schmitz 2016, S. 332–333) machen dies deutlich.

Auch Gambier und Lautenbacher betonen speziell mit Bezug auf Unterschiede zwischen Onlinetexten und herkömmlichen Texten, dass über die sprachlichen Zeichen hinaus weitere Aspekte eine Rolle spielen: „When localizing a website, for example, we immediately recognize that the layout, colors, and typographic characters, all with culturally marked value, transform our reading.“ (Gambier und Lautenbacher 2024, S. 8) Parallel verlieren gewisse Aspekte des außersprachlichen Kontexts bei Websites an Bedeutung:

Websites are rarely dated and, apart from their topic (referring to artifacts, goods, machines, services, local places, etc.), users do not bother to search for the producer of the site to discover when and why it has been produced. The text in the title bar, keywords, menus, hyperlinks, images, sound and audiovisual files, and the main body text are simply there, with no explicit justification or crossreferences. Parts of the context collapse in online texts. (Gambier und Lautenbacher 2024, S. 8)

Auf dieser Basis entwickeln sie folgende Kontextdefinition für multimediale Texte:

The context could be understood as the set of elements (linguistic and other, such as those visual and audial) necessary for the production and reception of the text (oral or written) – it would be both the cotext (discursive environment), the communication situation *hic et nunc* (participatory framework, socio-spatio-temporal framework and purposes of the text), encyclopaedic, cultural and discursive knowledge that makes our interpretation oriented, for instance, by the rules related to this or that genre [...]. (Gambier und Lautenbacher 2024, S. 8–9)

In dieser Definition finden sich viele Aspekte wieder, die bereits in Bezug auf sprachlichen und außersprachlichen Kontext (s. Abschnitte 3.4.3 und 3.4.2) genannt wurden, es werden aber auch explizit der visuelle und auditive Modus genannt, die auch Schmitz aufführt (vgl. U. Schmitz 2016, S. 332).

Meloni betont in ihrer Untersuchung der Übersetzungen von Heinrich Hoffmanns *Struwelpeter* ins Italienische darüber hinaus die Bezüge zwischen den verschiedenen Modi, wenn sie drei Kontextaspekte nennt: „1) die Wechselwirkung von Text und Bild (verbaler Kontext und nonverbaler/visueller Kontext); 2) die akustische Ebene des Textes (paraverbaler/prosodischer Kontext); 3) die semantische Dimension des Textes (semantischer Kontext)“ (Meloni 2024, S. 165). Im weiteren Verlauf ihrer Darstellung benutzt sie den Begriff „situativer Kontext“, um speziell den ersten Aspekt zu erläutern. Es sei „ein komplementärer Text-Bild-Bezug zu beobachten, d. h. der Text erzählt die Geschichte und beschreibt ergänzend dazu den situativen Kontext“ (Meloni 2024, S. 165), oft mit explizitem Bezug auf die Bilder, in denen mehrere Handlungen gleichzeitig dargestellt würden (vgl. Meloni 2024, S. 166).

Auf ähnliche Weise betont Moreno García, dass in Videospielen Multimodalität und Interaktivität zu einem komplexen Artefakt kombiniert werden

und eine immer weiter steigende Zahl an Kommunikationsmodi umfassen, zu denen er Text, Bewegtbilder, Soundeffekte, Musik und kinästhetische Informationen zählt, die gemeinsam die Spielerfahrung ausmachen und in Kombination Bedeutung haben (vgl. Moreno García 2024, S. 65).

Man kann also sagen, dass die Multimodalität sich auf die Frage des Kontexts auswirkt, einerseits, indem zum schriftlichen Text weitere Modi und damit auch semiotische Ebenen hinzukommen, und andererseits, indem zwischen diesen Ebenen Bezüge entstehen, die sich auf die Gesamtaussage auswirken und damit weitere Kontextinformationen darstellen, die zur Gesamtbedeutung beitragen. Je nach Textsorte – Website, gereimte Bildergeschichte, Videospiele etc. – können diese weiteren Kontextaspekte unterschiedlich sein.

Allgemeiner formuliert diese Haltung, dass neben den linguistischen Elementen auch die diese umgebenden nicht linguistischen textlichen Elemente allesamt von Bedeutung sind, Haapaniemi. Er versteht dies als Teil seines Ansatzes der Materialität, bei dem betont wird, dass jegliche Art von Zeichen oder Zeichensystem, wie verbale Sprache, stets ein materielles Medium nutzen und eine materielle Form einnehmen muss, damit Empfängerinnen daraus Bedeutung ableiten können. Die Materialität wirke dabei in zwei Richtungen: Aus ihr ergäben sich Einschränkungen für die Produktion von Zeichen und gleichzeitig auch für die Konstruktion der Bedeutung aus Zeichen. (vgl. Haapaniemi 2024, S. 22)¹⁶ Mit Verweis auf Marais betont er, dass die Nutzung der Sprache in eine allgemeine Zeichentheorie eingebettet werden und nicht nur sprachliche Zeichen betreffen sollte (vgl. Haapaniemi 2024, S. 26; s. auch Marais 2019, S. 119). Haapaniemi weist ebenso darauf hin, dass das Zeichen eine Form aufweisen muss, die durch das genutzte Medium übertragen werden kann und durch die Empfängerinnen erkennbar ist. Dies bezeichnet er als kotextuelle Einschränkung (vgl. Haapaniemi 2024, S. 27).

Es lässt sich also schließen, dass es notwendig ist, auch die multimodalen Spezifika eines bestimmten Textes (bzw. einer bestimmten Textsorte) bei Diskussionen zum Kontext mit einzubeziehen.

3.4.7 Kontext im Fachtext

Auch für den Bereich der Fachtexte wird die Relevanz des Kontexts verschiedentlich herausgearbeitet. Beispielsweise nennt Engberg den fachlichen Kontext als eine Hauptherausforderung der Fachübersetzung (vgl. Engberg 2024, S. 245). Konkret nennt er in Bezug auf die maschinelle Übersetzung auf dem Gebiet des Rechts „den fehlenden Einbezug von Fachkontext, insbesondere Wissen, Terminologie und andere[r] Formulierungskonventionen“,

¹⁶Vgl. dazu auch Adamzik 2016, S. 64

als Risiken (vgl. Engberg 2024, S. 245). Das Fachwissen betrachtet er in seinem Modell als kognitiven Kontext von Kommunikation (vgl. Engberg 2024, S. 247). Unter *Fachwissen* versteht er Folgendes:

[...] die Einsichten, die von Individuen erlangt und geteilt werden, die einer Fachgemeinschaft angehören, die sich gleichzeitig durch das Teilen und die wechselseitige Konstruktion dieses Wissens in kommunikativer Interaktion konstituiert (Engberg 2024, S. 249)¹⁷

Er nennt dies verstehensrelevantes Wissen. Dieses „beschränkt sich nicht auf das direkt z. B. in Wörterbüchern und Lexika auffindbare Wissen, sondern umfasst alles Wissen, das man braucht, um Elemente eines Textes so zu interpretieren, wie dies in der jeweiligen Situation und für den jeweiligen Wissens- und Kommunikationsbereich angemessen ist“ (Engberg 2024, S. 250) Dieses Wissen bilde den kognitiven Kontext für das Verstehen kommunikativer Äußerungen, wozu aber auch „Merkmale aus dem Text selbst“ (vgl. Engberg 2024, S. 251) gehören.

Ebenso weist Schmitt in Bezug auf die Terminologearbeit für Fachtexte darauf hin, dass „[...] die aktuelle Bedeutung nicht nur von Wörtern im allgemeinen, sondern auch von Fachausdrücken oft entscheidend von der Verwendungssituation (Kontext) und dem textuellen Umfeld (Kotext)“ (Schmitt 2006, S. 302) abhängt. Damit zeigt er wie Engberg auch für den fachtextlichen Bereich die Zerteilung in außersprachlichen und sprachlichen Kontext auf, die bereits weiter oben allgemein herausgearbeitet wurde.

3.4.8 Zwischenfazit zum Kontext

Zusammenfassend lässt sich auf Basis dieser Erläuterungen also sagen, dass der Kontext von großer Relevanz für das Erfassen der Bedeutung von Wörtern verschiedener Texte ist, auch beispielsweise bei Fachtexten. Es zeigen sich allerdings Unterschiede in der Bezeichnung: Diejenigen Autorinnen, die mit dem Begriff *Kotext* arbeiten, sehen darin die sprachliche Umgebung und nutzen *Kontext* zur Bezeichnung der außersprachlichen Umgebung. Dieses Begriffspaar erscheint damit eine Alternative zu sein für die in Abschnitt 3.4.2 dargestellte Unterscheidung zwischen extralingualem Situationsteil und intralingualem Kontextteil (Vermeer), verbalem und non-verbalem Kontext (Kußmaul), Mikro-/Makrokontext und Situationskontext (Reiß). Ebenso wird klar, dass der außersprachliche Kontext schwerer zu greifen ist und es schwierig wäre, eine abschließende Menge an Aspekten

¹⁷Bei dieser Definition verweist er auch auf Kalverkämper 1997, S. 14–15.

aufzuzählen, die den außersprachlichen Kontext bei translatorischen Tätigkeiten umfassen und sich auf das Verhalten der Translatorinnen auswirken, wie Baker betont (vgl. Baker 2006, S. 332).

Nach Aussage von Albrecht ist der sprachliche Kontext bei schriftlichen Texten von größerer Bedeutung (vgl. Albrecht 2013, S. 217) und gleichzeitig auch greifbarer und im Bezug auf das Wissen im Unterschied zum Sprecherwissen vergleichsweise kongruent, wie Klos deutlich macht (vgl. Klos 2011, S. 172). Der sprachliche Kontext wird nach Klos „vom Text selbst mitgegeben“ und kann „unabhängig von textexternen, situativen Faktoren gedacht werden“ (Klos 2011, S. 172).

Allerdings geben die aufgeführten Definitionen des sprachlichen Kontexts keine klare Auskunft auf die Frage, ob der sprachliche Kontext den gesamten Text umfasst oder nur die (mehr oder weniger umfangreiche) direkte sprachliche Umgebung einer Äußerung. Lediglich Reiß geht diese Frage mit ihrer Unterscheidung zwischen Mikro- und Makrokontext gezielt an. Zudem muss der sprachliche Kontext noch je nach Textsorte durch multimodale Aspekte ergänzt werden, für die die Kriterien der Greifbarkeit und Kongruenz von Klos ebenfalls gültig sind.

Diese Kriterien sind es auch, durch die sich der sprachliche Kontext für empirische Untersuchungen wie beispielsweise die von Klos zur semantischen Dekodierung von Substantivkomposita (vgl. Klos 2011) oder die von Burkhardt zur Bedeutungserschließung in fremdsprachigen Texten anbietet, in der sie die Rolle des sprachlichen Kontexts überprüft (vgl. Burkhardt 1995). Auch Prior et al. arbeiten bei ihrer Untersuchung mit der Übersetzungsambiguität einzelner Wörter „in and out of context“ mit dem sprachlichen Kontext in Form von Auszügen aus Parallelkorpora (vgl. Prior u. a. 2011). Analog dazu untersuchen Rydning und Lachaud das Verstehen und Produzieren von Übersetzungen bei polysemen Wörtern mit und ohne Kontext, wobei der Kontext in diesem Fall aus dem ersten Absatz einer Vorlesung besteht. (vgl. Rydning und Lachaud 2010) Sie nutzen also ebenfalls den sprachlichen Kontext.

In der aktuellen Translationsprozessforschung wird ebenso mit dem sprachlichen Kontext gearbeitet. Beispielsweise weist Wei darauf hin, dass es eine große Einigkeit hinsichtlich der Frage gibt, dass der sprachliche Kontext zur Interpretation ambiger Wörter beiträgt und auch die Anzahl geeigneter Übersetzungen für ein AT-Wort reduziert. Er beschreibt, wie die Unsicherheit bei Wörtern mit hoher Entropie mit Hilfe zusätzlicher Kontextinformationen, „information which is provided by the surrounding items in the textual material“ (Wei 2021, S. 171), reduziert wird. In seiner Untersuchung stellt er fest, dass der sprachliche Kontext zum Sammeln weiterer Informationen für Klarstellung oder Disambiguierung genutzt wird (vgl. Wei 2021, S. 197).

Daher besteht die begründete Vermutung, dass auch für die vorliegende Untersuchung der sprachliche und ergänzend dazu der relevante multimodale Kontext genutzt werden kann. Voraussetzung dafür ist, dass für das hier diskutierte Thema GUI-Text Aschenbergs Forderung nach „begrifflicher Reduktion“ (Aschenberg 1999, S. 8) erfüllt wird. Diese stehe in Zusammenhang damit, welche Theoriebildung der Kontextfrage vorausgeht und welche deskriptiven Ansprüche daraus folgen (vgl. Aschenberg 1999, S. 8). Im folgenden Kapitel wird dazu zunächst erläutert, wie und welche Aspekte von Kontext sich bei GUI-Text zeigen. Auf dieser Basis soll der Kontext von GUI-Text operationalisiert werden, wodurch die weiteren Untersuchungen im vorliegenden Buch vorbereitet werden.

Kapitel 4

GUI-Text: Multimodaler Fachtext, Kontext und Referenzmaterial

In diesem Kapitel sollen die Erkenntnisse aus dem vorangegangenen Kapitel 3 auf den Bereich der Softwarelokalisierung, konkret auf GUI-Text übertragen werden. Das geschieht in mehreren Schritten: Zunächst wird in Abschnitt 4.1 ermittelt, ob es sich bei GUI-Text um multimodalen Text handelt, und anschließend in Abschnitt 4.2, ob es sich bei GUI-Text um eine Fachtextsorte handelt. Im Anschluss daran wird in Abschnitt 4.3 der Zusammenhang zwischen GUI-Text und Programmfunktionen erörtert, woraufhin in Abschnitt 4.4 die Frage ins Zentrum rückt, welche Kontextaspekte bei GUI-Text vorhanden sind und operationalisiert werden können. Danach werden in Abschnitt 4.5 Zusammenhänge zwischen diesen operationalisierbaren Kontextaspekten und übliche Referenzmaterial bei der Softwarelokalisierung herausgearbeitet. Abschnitt 4.6 liefert dann erste Anhaltspunkte für die Features der Korpusstudie im nächsten Kapitel, Kapitel 5, und in Abschnitt 4.7 werden einige der Einschränkungen und offenen Fragen dieser Herangehensweise diskutiert.

4.1 GUI-Text als multimodaler Text

In Abschnitt 3.1 wurde auf Basis von Adamziks Wunsch nach einer prototypischen Textdefinition (vgl. Adamzik 2016, S. 41) und ihrer Meinung, dass je nach unterschiedlichen zu untersuchenden Aspekten unterschiedliche Textdefinitionen erforderlich sind (vgl. Adamzik 2016, S. 43), Göpferichs Textdefinition (vgl. Göpferich 1995, S. 56) mit der von Schmitz für multimodalen

Text (vgl. U. Schmitz 2016, S. 333) kombiniert. Auf dieser Basis kann für GUI-Text die Frage beantwortet werden, ob dieser im Sinne der erarbeiteten Definition auch ein multimodaler Text ist, wobei parallel ein konkreter Blick auf die multimodalen Aspekte von GUI-Text geworfen werden kann, die in der Literatur zur Softwarelokalisierung genannt werden.

Die erarbeitete Definition weist folgende Aspekte eines multimodalen Texts auf:

- Zeichengebilde mit mindestens einem sprachlichen Zeichen
- Thematisch und/oder funktional orientiert
- Kohärent
- Erkennbare kommunikative Funktion
- Abgeschlossene Einheit
- Mindestens zwei Modi

Alle diese Punkte treffen auf GUI-Text zu: Alle GUIs sind Zeichengebilde mit mindestens einem sprachlichen Zeichen. Wenn man als Beispiel eine E-Mail-Software betrachtet, so ist dieses Zeichengebilde sowohl thematisch als auch funktional orientiert und kohärent und weist darüber hinaus eine erkennbare kommunikative Funktion auf, nämlich Benutzerinnen das Verfassen, Senden, Empfangen und Verwalten von E-Mails zu ermöglichen. Zudem bildet das Softwareprodukt eine abgeschlossene Einheit. Und zuletzt weist es neben sprachlichen Zeichen auch andere Zeichen auf, wie sich beispielsweise an Abbildung 2.2 erkennen lässt, auf der neben Text auch verschiedene Symbole zu sehen sind. Auf dieser Basis lässt sich zweifelsfrei feststellen, dass es sich bei GUI-Text um multimodalen Text handelt.¹ Auch Beste bezeichnet GUI-Text in seiner Analyse der Fachsprachlichkeit als Text und betont dabei, dass GUI-Text insbesondere auch von Rezipienten als Text wahrgenommen wird, wodurch Beaugrandes und Dresslers Textualitätskriterium der Akzeptabilität erfüllt werde (vgl. Beste 2006, S. 43).

In der Literatur zur Softwarelokalisierung werden Merkmale der Multimodalität an verschiedenen Stellen beschrieben. Im Folgenden werden diese Merkmale aufgeführt, um die Tatsache, dass es sich bei GUI-Text um

¹Wie bereits in Abschnitt 3.1 beschrieben, fehlt in Göpferichs Textdefinition (vgl. Göpferich 1995, S. 56) und damit auch in der hier untersuchten Textdefinition das Merkmal der Kohäsion, das beispielsweise bei Beaugrande und Dressler genannt wird (vgl. De Beaugrande und Dressler 1981, S. 3–5). Kohäsion ist in GUI-Text tatsächlich nur sehr eingeschränkt vorhanden, da viele Strings aus maximal zwei Wörtern (vgl. Beste 2006, S. 40) und nur wenige aus vollständigen Sätzen (vgl. Ottmann 2005, S. 105) bestehen, die auch nicht wie herkömmliche Texte eine zusammenhängende Abfolge von Sätzen, Absätzen und Abschnitten bilden, bei der in der Regel Kohäsion vorliegt. GUI-Text würde also nicht alle von Beaugrandes und Dresslers Textualitätskriterien (vgl. De Beaugrande und Dressler 1981, S. 3–13) erfüllen.

multimodalen Text handelt, zu untermauern und gleichzeitig die weitere Argumentation vorzubereiten, bei der die Multimodalität eine wichtige Rolle spielt.

Bei GUIs sind neben dem zu übersetzenden Text ebenso grafische Mittel für die Interaktion von Bedeutung (vgl. Wahle 2000b, S. 35) und damit für das Verständnis. Typische Beispiele dafür sind Symbole und Abbildungen (vgl. Esselink 2000, S. 26–27, Beste 2006, S. 71–72, K.-D. Schmitz 2005a, S. 10–12, und Behrens 2016, S. 53–58), die von Beste als nonverbale Informationsträger bezeichnet werden, wobei er konkret Symbole in Symbolleisten nennt (vgl. Beste 2006, S. 43). Auch der Einsatz von Farben ist in diesem Zusammenhang zu nennen (vgl. Esselink 2000, S. 34, K.-D. Schmitz 2005a, S. 12–13, und Behrens 2016, S. 52–53).

Neben diesen grafischen Mitteln führt Beste noch typografische Gestaltungsmittel insbesondere in Menüs auf, die laut Beste „zusätzliche Informationen [...] vermitteln“ (Beste 2006, S. 61–62) sollen:

- Als Access-Keys markierte Buchstaben werden unterstrichen dargestellt (vgl. Beste 2006, S. 64).²
- Menüpunkte, nach deren Auswahl ein Dialogfeld geöffnet wird, weisen am Ende drei Punkte auf (vgl. Beste 2006, S. 62).
- Nicht verfügbare Menüpunkte werden grau hinterlegt und für einzelne Menüpunkte gibt es Pfeile, mit denen auf zugehörige Untermenüs hingewiesen wird (vgl. Beste 2006, S. 62).
- Die momentan angewählte Menüoption wird farbig hinterlegt (vgl. Beste 2006, S. 42).

Hinzu kommt die audiale Ebene, beispielsweise über Tonsignale wie Pieptöne, die bei fehlerhaften Eingaben zum Einsatz kommen (vgl. Zerfaß 2002, S. 214, und Behrens 2016, S. 59).

Des Weiteren nennt Beste die hierarchische Menüeinteilung, mit deren Hilfe zusammengehörige Menüoptionen in gemeinsamen Haupt- oder Untermenüs angeordnet werden:

So finden sich im Menü „Datei“ beispielsweise die Befehle „Datei öffnen“, „Datei drucken“ und „Datei schließen“. Diese formale Unterteilung ist gleichzeitig funktionsorientiert und folgt in den meisten (jedoch nicht allen) Fällen logisch nachvollziehbaren Kriterien, wie der Zusammenfassung von Tabellenoperationen im Menü „Tabelle“. (Beste 2006, S. 41; vgl. auch Beste 2006, S. 61)

²Detaillierte Erläuterungen zu diesem und dem nächsten Punkt finden sich in Abschnitt 2.2.

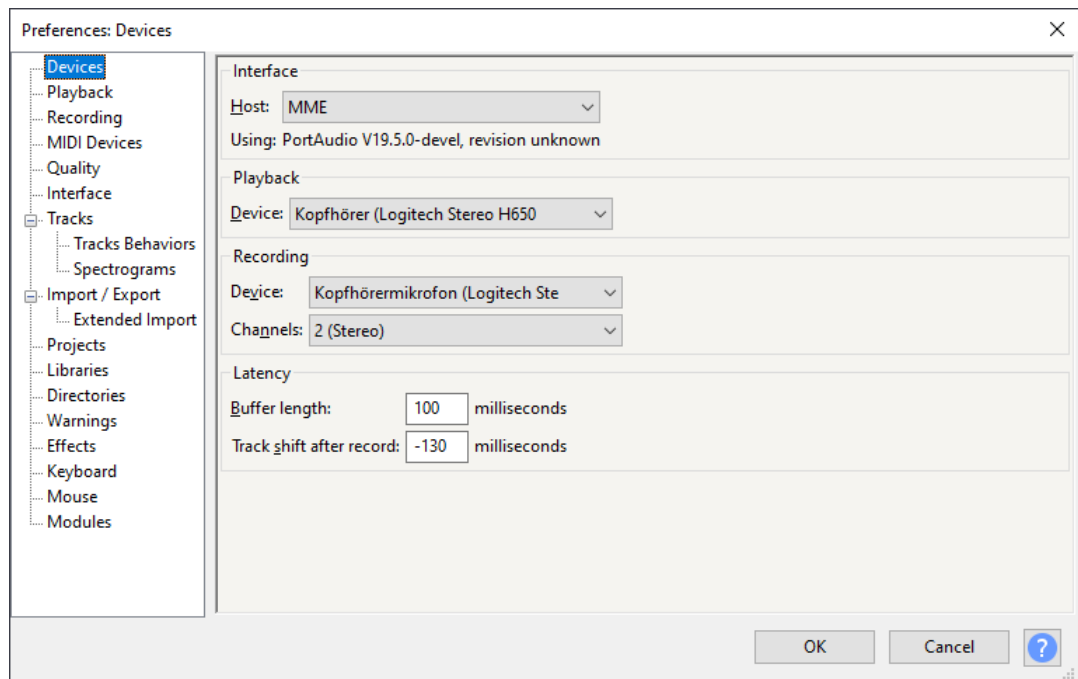


Abbildung 4.1: Dialogfeld **Preferences**, Bereich **Devices** in Audacity Version 2.2.2

Diese funktionsorientierte Unterteilung trifft neben Menüs auch auf Dialogfelder zu, die zudem verschiedene Typen von GUI-Elementen³ enthalten können. Beides dient zur Bereitstellung von weiteren Informationen, die mit außersprachlichen Mitteln kommuniziert werden.

Ein Beispiel: Im rechten Teil von Abbildung 4.1, in dem ein Dialogfeld mit Einstellmöglichkeiten für den Audioeditor Audacity 2.2.2 zu sehen ist, sind verschiedene Bereiche mit den Überschriften **Interface**, **Playback**, **Recording** und **Latency** voneinander abgetrennt. In diesen Bereichen sind jeweils zugehörige Einstelloptionen gruppiert. In den ersten drei Bereichen sind Dropdown-Listen mit vorgegebenen Auswahlmöglichkeiten zu finden, im letzten Bereich Textfelder, in die Benutzerinnen gewünschte Werte eingeben können.

Ein weiteres Beispiel: Im rechten Teil von Abbildung 4.2 finden sich ebenfalls verschiedene Bereiche und zudem Optionsfelder, von denen jeweils nur eines ausgewählt werden kann, sowie Kontrollkästchen, die jeweils unabhängig voneinander aktiviert oder deaktiviert werden können. Auch mit diesen unterschiedlichen GUI-Elementen gehen also verschiedene Bedeutungen ein-

³Unter Windows beispielsweise Schaltflächen, Optionsfelder, Kontrollkästchen, Schieberegler sowie Text-, Listen-, Kombinations- und Drehfelder (vgl. Beste 2006, S. 62–63).

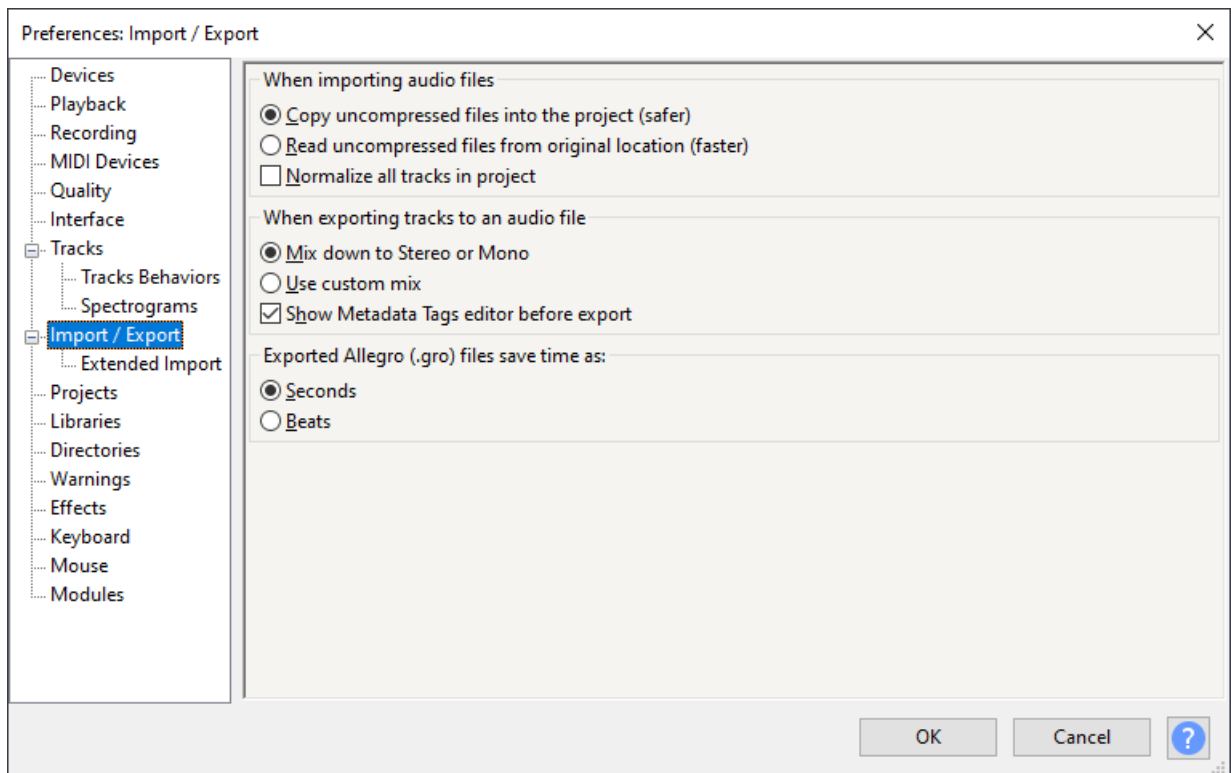


Abbildung 4.2: Dialogfeld **Preferences**, Bereich **Import / Export** in Audacity Version 2.2.2

her, die sich optisch erkennen lassen, ohne die Funktionalität ausprobieren zu müssen.

Diese Aspekte, grafische Mittel wie Symbole, typografische Gestaltungsmittel wie farbiges Hinterlegen von Strings, Tonsignale, die hierarchische Struktur von Strings und verschiedene zugehörige Bereiche und Felder, mit denen unterschiedliche Bedeutungen einhergehen, befinden sich alle außerhalb der Ebene der sprachlichen Zeichen und tragen dennoch zur Bedeutung bei.

Betrachtet man bei der in Abschnitt 3.1 erarbeiteten Definition die Voraussetzung für Multimodalität – mindestens zwei Modi –, so kann man nach der hier aufgeführten detaillierten Auflistung aus der Literatur zur Softwarelokalisierung die obige Schlussfolgerung bestätigen, dass es sich bei GUI-Text um multimodalen Text handelt, da sich in GUI-Text geschriebene Sprache, Bilder und Audio wiederfinden lässt. Auch wenn Schmitz selbst betont, dass „die bedienungsanleitenden Ikons im Formular des Textverarbeitungsprogramms“ (U. Schmitz 2016, S. 334) seiner Ansicht nach für sich betrachtet

einen Text nicht multimodal machen, muss konstatiert werden, dass sich in den Beispielen oben deutlich verschiedene Modi zeigen, deren Bedeutung im Zusammenspiel sichtbar wird. Noch deutlicher macht dies Fix mit ihrer Auflistung nichtsprachlicher Zeichen, wie „Bilder, typographische Elemente, Farben, Papiersorten, Flächen, Linien“ (Fix 2001, S. 114), die sich in den Beispielen oben weitestgehend finden lassen und nach ihrer Deutung Zeichenkomplexe bilden, bei denen alle diese Zeichen gemeinsam das Sinnangebot eines Textes bilden (vgl. Fix 2001, S. 114, 118). Schäler macht diese Feststellung, dass GUI-Text multimodal ist, ebenfalls, indem er auf Folgendes hinweist und dabei u. a. Softwareprodukte aufführt:

[...] the material localisers deal with is multimodal, i.e., the files they localise can contain text, graphics, audio and video applied to a large variety of services or products, from websites to desktop applications, video games and courseware. (Schäler 2008, S. 196)

Behrens führt in Ergänzung zu dieser Argumentation eine eigene Argumentation aus, deren Ziel es ist, den zugrunde liegenden Textbegriff nicht mehr statisch, sondern prozedural zu sehen. Auch für ihn ist die Basis, dass Text mehr als nur verbale Inhalte umfasst. Als Besonderheit von GUI-Text sieht er zudem, dass dieser aus mindestens drei Teilrepräsentationen besteht, dem „permanenten Stringvorrat“, also den in Ressourcendateien ausgelagerten Strings, dem Quellcode als „Anweisungen für die logische Verarbeitung dieses Stringvorrats“⁴ und der „flüchtige[n] Ausgabe für den Anwender (UI)“ (Behrens 2016, S. 76). Für ihn entsteht Text aus der Interaktion dieser drei Teilrepräsentationen, weshalb für ihn auch „typografische Strukturdaten (Markup-Zeichen, Operatoren, Separatoren usw.) und der Quellcode zu den nonverbalen Textinformationen gezählt werden“ (vgl. Behrens 2016, S. 158). Lytje will Software aus dem Blickwinkel des Softwaredesigns ebenso als dynamischen Text betrachten, weil die Software auf diese Weise als dynamisches semiotisches Produkt betrachtet werden kann, das die Intention der Designerin ausdrückt und die intendierten Bedeutungen über das Produkt an die Benutzerin kommuniziert (vgl. Lytje 1996, S. 143, 157–158). Lytje bezeichnet es in diesem Zusammenhang sogar als offensichtlich, dass Software sowohl multimodal als auch multimedial ist. Die Multimodalität sieht sie dadurch als gegeben an, dass mehrere Sinne und kognitive Fähigkeiten aktiviert werden müssen, um sie „zu lesen“. Die Multimedialität sieht sie darin bestätigt, dass verschiedene Arten von Instrumenten für die Darstellung verwendet werden. (vgl. Lytje 1996, S. 158)

⁴Beides s. Abschnitt 2.1.4

Darüber hinaus wird auch im mit der Softwarelokalisierung verwandten Bereich der Lokalisierung von Videospiele Multimodalität thematisiert, wie an Schäfers Aussage oben bereits zu sehen war.⁵ In Bezug darauf findet sich beispielsweise bei Dietz der Begriff „blind localization“ für den Fall, dass die Lokalisierenden nicht das zu lokalisierende Spiel spielen können (vgl. Dietz 2007, S. 5), der später auch beispielsweise von O’Hagan und Mangiron aufgegriffen wird (vgl. O’Hagan und Mangiron 2013, S. 119) und von Moreno García dahingehend ergänzt wird, dass bei „blind localization“ die Strings analog zu der Beschreibung in Abschnitt 2.1.3 in separate Ressourcendateien ausgelagert werden. Dadurch müssen sich nach seinen Worten die Lokalisierungsexpertinnen mit „disassembled ‚texts‘ forced back to monomodality“ (Moreno García 2024, S. 64) auseinandersetzen, da die weiteren Modi und damit also die Multimodalität auf diese Weise verloren geht. Speziell bei mobilen Spielen spricht er bei dieser Vorgehensweise davon, dass diese „a vast entity seen by the translator from an incredibly limited and limiting element, that of the monomodal ‚text‘“ (Moreno García 2024, S. 64) sind. Nach Kim sind die extrahierten Textinhalte außerhalb der multimodalen Umgebung, wie sie häufig bei der Lokalisierung von Videospiele vorliegen, „Nekrotext“ (Kim 2019, S. 120), der nur einen Teil des Informationsangebots des Videospiele bietet. Alle diese Betrachtungsweisen zeigen die Multimodalität auf indirekte Art, nämlich aus dem Blickwinkel der Übersetzerinnen: Wenn die Multimodalität bei der Übersetzung von extrahierten GUI-Strings verloren geht, geht damit auch das vollständige semiotische Angebot verloren.

Es lässt sich also sagen, dass sowohl gemäß der erarbeiteten Definition für multimodalen Text, aufgrund der in der Literatur zur Softwarelokalisierung geschilderten Aspekte von GUIs als auch aufgrund der Situation im verwandten Bereich der Lokalisierung von Videospiele GUI-Text als multimodaler Text betrachtet werden muss. Im nächsten Abschnitt werden die Fragen diskutiert, ob GUI-Text als eigene Textsorte und darüber hinaus als Fachtext betrachtet werden kann.

4.2 GUI-Text als Fachtextsorte

Analog zur Vorgehensweise im vorangehenden Abschnitt soll in diesem Abschnitt überprüft werden, ob es sich bei GUI-Text um eine multimodale Text-

⁵Im Bereich der Multimodalität zählt Pérez González neben der Lokalisierung von Videospiele noch weitere audiovisuelle Medien auf: „Audiovisual textualities – including films, dramas, or videogames – represent another crucial locus of interaction between verbal and non-verbal signifiers.“ (Pérez González 2014, S. 121)

sorte und zudem um Fachtext handelt. Zunächst werden dabei die verschiedenen Aspekte der in Abschnitt 3.2 erarbeiteten Definition für multimodale Textsorten überprüft. Es wird also festgestellt, ob es sich bei GUI-Text um konventionalisierte komplexe Kommunikationsmuster mit einer bestimmten kommunikativen Funktion handelt, die sich innerhalb einer Sprachgemeinschaft im Lauf der Zeit gebildet haben und die mehrere Modi aufweisen können (vgl. beispielsweise Brinker, Pappert und Cölfen 2024, S. 149, 160, und U. Schmitz 2016, S. 333).

Zunächst kann davon ausgegangen werden, dass das Kriterium der Entwicklung der Textsorte GUI-Text im Lauf der Zeit zutrifft, da GUIs wie in Abschnitt 2.1.3 beschrieben schon seit den 1990er Jahren der gängigste Oberflächentyp bei Software sind. Zudem handelt es sich wie schon aus den Ausführungen in Abschnitt 4.1 ersichtlich, um komplexe Kommunikationsmuster mit einer bestimmten kommunikativen Funktion (s. das dort genannte Beispiel *E-Mail-Programm*), die auch mehrere Modi aufweisen. Ebenso liegt bei GUIs Konventionalisierung vor, wie sich an den Ähnlichkeiten der GUIs verschiedener Softwareprodukte, nicht nur für ein Betriebssystem wie Windows, sondern auch im Vergleich mehrerer Desktop-Betriebssysteme wie Windows, Linux und Mac OS zeigt, wie Beste exemplarisch anhand der plattformübergreifenden Standardisierung typischer Menüoptionen feststellt (vgl. Beste 2006, S. 40, 42). GUIs weisen in der Regel eine ähnliche Struktur und je nach Betriebssystem typische Elemente auf, die unter Windows beispielsweise als Schaltflächen, Optionsfelder, Kontrollkästchen, Schieberegler sowie Text-, Listen-, Kombinations- und Drehfelder (vgl. Beste 2006, S. 62–63) bezeichnet werden. GUI-Text als Textsorte erleichtert durch diese Konventionen die Rezeption, weil sich Benutzerinnen rasch mit ihnen unbekanntem Softwareprodukten vertraut machen können.

Dass GUI-Text selbstverständlich als Textsorte betrachtet werden kann, zeigt sich auch darin, dass verschiedene Autorinnen aus dem Bereich Softwarelokalisierung den Begriff *Textsorte* ohne weitere Erläuterungen mit Bezug auf GUI-Text verwenden. Beispielsweise spricht Behrens von der „Textsorte UI“ (vgl. beispielsweise Behrens 2016, S. 41, 158)⁶, und Beste arbeitet in Bezug auf GUI-Text ebenfalls mit dem Begriff *Textsorte* (vgl. Beste 2006, S. 50). Behrens teilt außerdem den Bereich der Softwarelokalisierung explizit in „grob drei Textsortenklassen“ ein, von denen einer GUI-Text ist, nämlich „Texte, die die Bedienung der Software ermöglichen: die Ressourcen, d. h. Dialoge, Menüs, Accelerator- und String-Tabelle, das Versionsfenster“ (Behrens 2016, S. 71).⁷

⁶Behrens verweist an mehreren Stellen auch explizit auf die „Spezifika der Textsorte UI“ (s. z. B. Behrens 2016, S. 22, 164), also die Konventionen dieser Textsorte.

⁷Die weiteren Textsortenklassen sind einerseits Hilfetexte und Dokumentation und an-

Wie in Abschnitt 3.3 gezeigt, hängen die fachsprachlichen Merkmale eines Fachtexts primär von der Textsorte ab (vgl. Göpferich 1995, S. 29). Wenn GUI-Text als eigene Textsorte eingestuft wird, können auf dieser Basis die fachsprachlichen Merkmale von GUI-Text untersucht werden. Je nachdem, welche dies sind und welchen Raum sie einnehmen, kann dann von GUI-Text als Fachtext geredet werden. Diese Frage nimmt Beste in seiner Arbeit in den Blick. Dazu untersucht er detailliert die von ihm identifizierten Textsorten der Softwarelokalisierung dahingehend, ob sie Kriterien der Fachsprachlichkeit auf den Ebenen Lexik, Syntax und Text erfüllen (vgl. Beste 2006, S. 37). In Bezug auf GUI-Text stellt er insbesondere fest, dass für den Bereich der Lexik die Merkmale „Kontextuelle Eineindeutigkeit der Terminologie“, „Erhöhte Konstanz der lexikalischen Referenz“, „Flexionsmorphologische Paradigmen-selektion“ und „Erhöhte Ausschöpfung bestimmter Wortbildungsverfahren“ erfüllt oder teilweise erfüllt werden (Beste 2006, S. 39–43, 47). Das Merkmal im Bereich der Syntax, „Syntaktische Paradigmenselektion der Fachsprache“, ist dagegen nicht anwendbar, weil in GUIs keine oder nur wenige vollständige Sätze enthalten sind (Beste 2006, S. 41, und Ottmann 2005, S. 105).⁸ Auf Basis dieser Untersuchungen scheint es Beste gerechtfertigt, Softwaretexte ersten Grades, also GUI-Text, „als technische Fachtexte mit einem hohen Grad an Fachsprachlichkeit zu klassifizieren“ (Beste 2006, S. 43).

Damit zeigt sich, dass sich die Fachsprachlichkeit von GUIs primär im Bereich der Lexik manifestiert, weil in aller Regel nur wenige vollständige Sätze mit fachsprachlicher Syntax vorliegen, und hier in der Nutzung von Fachterminologie verschiedener Bildungsarten wie Komposition, Derivation und Konversion (vgl. Beste 2006, S. 41). Deren Festlegung und konsistente Nutzung wird in der Literatur zur Softwarelokalisierung auch an verschiedenen Stellen thematisiert (vgl. beispielsweise K.-D. Schmitz 2002, S. 14, und K.-D. Schmitz 2005a, S. 9), wobei „je nach Fachlichkeitsgrad des Produktes [...] der Anteil der fachspezifischen Terminologie unterschiedlich groß sein kann“ (K.-D. Schmitz 2002, S. 14). Zu beachten ist dabei allerdings, dass

dererseits Begleitmaterialien (vgl. Behrens 2016, S. 71), wobei Behrens auf Bestes Einteilung in Softwaretexte ersten, zweiten und dritten Grades (vgl. Beste 2006, S. 39) verweist.

⁸Beste untersucht darüber hinaus auch die Merkmale im Bereich des Texts, „Isomorphie von Form und Funktion auf makro- und mikrostruktureller Ebene“, „Charakteristische textsortenübergreifende Textbausteine“, „Makrostrukturelle textsortenspezifische Fachtextbaupläne“, „Einsatz typographischer Gestaltungsmittel“ und „Nonverbale Informationsträger“, die er ebenso als erfüllt ansieht (vgl. Beste 2006, S. 47). In der vorliegenden Untersuchung sind die letzten dieser Merkmale aber im Bereich der Multimodalität angesiedelt und die anderen eher im Bereich der Textsorte GUI-Text zu sehen. Daher spielen sie für die weitere Untersuchung keine Rolle. Details zur Herausarbeitung der Merkmale (vgl. Beste 2006, S. 32–38) sowie zur detaillierten Überprüfung in Bezug auf GUI-Text (vgl. Beste 2006, S. 39–43) finden sich in Bestes Veröffentlichung.

sich viele Softwareprodukte an Laien richten, weil Computer in sämtlichen Lebens- und Arbeitsbereichen zum Einsatz kommen (vgl. Kemmann 2002, S. 88, und Beste 2006, S. 38), wodurch der Adressatenkreis sehr heterogen wird (vgl. Kemmann 2002, S. 89).

Im Bereich der Softwarelokalisierung lässt sich die Terminologie je nach Zweck der zu lokalisierenden Software (s. Abschnitt 2.1.1) in verschiedene Teilgebiete aufteilen:

- Terminologie aus dem Betriebssystem, unter dem die Software läuft, wie beispielsweise die Benennungen der GUI-Elemente des verwendeten Betriebssystems (vgl. K.-D. Schmitz 2005a, S. 10, und Kemmann 2002, S. 89).
- Terminologie aus dem Verwendungsbereich des Produkts. Esselink nennt als Beispiel dafür eine Finanzsoftware, bei deren Übersetzung die Terminologiestandards der Finanzbranche berücksichtigt werden müssen (vgl. Esselink 2000, S. 402, 407), und Kemmann zählt als Beispiele „bautechnische, medizinische, versicherungsmathematische oder automobiltechnische Terminologie“ (Kemmann 2002, S. 89) auf, die in GUIs enthalten sein kann (vgl. dazu auch Seewald-Heeg 2020, S. 16). Behrens geht überdies davon aus, „dass die Textsorte UI immer mehr Anwendungs- und Fachgebiete erobern wird“ (Behrens 2016, S. 41).⁹

Somit kann GUI-Text als eigene Textsorte betrachtet werden, die fachsprachliche Merkmale primär im Bereich der Lexik aufweist. Im nächsten Abschnitt wird neben der Multimodalität und der Fachsprachlichkeit der Textsorte GUI eine weitere Besonderheit von GUI-Text erläutert, der Zusammenhang zwischen Strings und Funktionen im Softwareprodukt.

4.3 GUI-Text und Programmfunktionen

Neben den eben behandelten Aspekten Multimodalität und Fachtextlichkeit weist die Textsorte GUI-Text noch eine weitere Besonderheit auf, nämlich den Bezug zu Programmfunktionen. Wie in Abschnitt 2.1.3 gezeigt, können Strings als Interaktionselemente betrachtet werden, die die Kommunikation zwischen den Benutzerinnen und Software, also zwischen Mensch und Maschine, ermöglichen. Beste weist explizit auf diese Besonderheit hin:

⁹Irmeler und Hartwig weisen darüber hinaus darauf hin, dass auch in einem einzigen Softwareprodukt verschiedene Verwendungszwecke vorkommen können: „[...] besonders die zunehmend beliebten ‚Suite‘-Softwarepakete vereinigen Anwendungen aus den verschiedensten Bereichen, wie z.B. Publizieren, Buchhaltung, Datenbanken etc., deren Terminologie oft sehr fachbezogen ist.“ (Irmeler und Hartwig 2000, S. 98)

Im Unterschied zu anderen Arten von Fachtexten besitzen Softwaretexte ersten Grades neben der inhaltlichen und sprachlichen Ebene zusätzlich eine technische Komponente in Form des eigentlichen Programms. (Beste 2006, S. 66)

Auch Behrens betont die Interaktivität von GUI-Text und folgert daraus, dass der verwendete Textbegriff „auch die Gesamtheit der Kommandos für die Mensch-Maschine-Interaktion einschließen [muss], also etwa Zugriffstasten und Accelerators“ (Behrens 2016, S. 159). Speziell stellt er also Access-Keys und Shortcuts und im weiteren Verlauf als Beispiel den Umgang mit Texteingaben in Microsoft Excel heraus, seine Aussage lässt sich aber auch allgemeiner lesen, nämlich mit Bezug auf sämtliche GUI-Elemente, ob sie nun Textelemente umfassen oder nur Symbole sind. Dies schließt auch an seine bereits oben (s. Abschnitt 4.1) erwähnte Argumentation an, dass GUI-Text aus Strings, Quellcode und der Ausgabe für Benutzerinnen besteht, die Strings also mit der Funktionalität im Rahmen der Ausgabe für Benutzerinnen eine Einheit bilden (vgl. Behrens 2016, S. 76).

Aus diesen Punkten lässt sich schließen, dass einzelne Strings stellvertretend für eine bestimmte Programmfunktion stehen und mit dieser verknüpft sind. Die zugehörige Programmfunktion wird ausgeführt, wenn das mit dem String versehene Interaktionselement (bzw. das Symbol) ausgewählt wird. Wahle macht diesen Zusammenhang deutlich, indem sie angibt, dass „die einzelnen Menüs, Befehle, Schaltflächen usw. in der Regel mit Text beschriftet sind, der die jeweilige Funktion beschreibt.“ (Wahle 2000b, S. 35)

Ein Beispiel: In einem Textverarbeitungsprogramm wird durch Auswahl des Menüpunkts **Speichern unter...** ein Dialogfeld geöffnet, in dem eine Speicherposition sowie ein Dateiname für die bearbeitete Datei gewählt werden kann. Durch Klicken auf die Schaltfläche **OK** wird die Datei anschließend gespeichert. Der Menüpunkt **Speichern unter...** ist also mit der Funktion zum Öffnen des entsprechenden Dialogfelds verknüpft, die Schaltfläche **OK** mit der Funktion zum Speichern mit den angegebenen Einstellungen (Speicherposition, Dateiname).

Ebenso können im Dialogfeld für den Drucker in vielen Fällen Kontrollkästchen aktiviert werden, mit denen beispielsweise eingestellt wird, ob die betreffende Datei in Farbe oder in Graustufen gedruckt wird oder ob pro Blatt eine oder zwei Seiten gedruckt werden. Diese Einstellungen werden beim Klicken auf die Schaltfläche **OK** an den Drucker übermittelt.

Primär gilt diese Verknüpfung von Strings mit Funktionen allerdings für Menüpunkte und Schaltflächen, mit denen die Software zum Ausführen bestimmter Schritte angewiesen wird, sowie für Kontrollkästchen und ähnliche Elemente, über die bestimmte Einstellungen oder Werte an

die Software übermittelt werden. Bei Symbolen, die Tooltips aufweisen, gibt es eine solche Verknüpfung ebenfalls, allerdings mit dem Symbol als Zwischenschritt: Der Text des Tooltips beschreibt die Funktion, die mit dem Symbol verknüpft ist. Nachrichten dienen hingegen eher zur Information. Mithilfe von Nachrichten werden Benutzer beispielsweise über Probleme, falsche Eingaben oder den Verarbeitungsfortschritt informiert (vgl. Beste 2006, S. 63). Derartige Strings sind also auf andere Weise und häufig nur indirekt mit den zugehörigen Funktionen verknüpft.

Die in diesem und den vorangehenden Abschnitten geschilderten Charakteristika von GUI-Text, Multimodalität, Fachtextlichkeit und Zusammenhang mit Programmfunktionen, werden in den nächsten Abschnitten mit dem Problem des Kontexts bzw. des fehlenden Kontexts bei der Softwarelokalisierung in Verbindung gebracht.

4.4 GUI-Text und Kontext

Bei den Ausführungen zum Thema Kontext in Abschnitt 3.4 zeigte sich, dass sich außersprachlicher und sprachlicher bzw. multimodaler Kontext unterschiedlich gut zur Operationalisierbarkeit eignen. Gambier und Lautenbacher ergänzen diese Problematik des außersprachlichen Kontexts durch weitere Aspekte, die auf digitale Texte zutreffen. Sie tun dies am Beispiel von Websites, indem sie wie in Abschnitt 3.4.6 schon ausgeführt, darauf hinweisen, dass bei diesen in der Regel weder eine genaue Datierung oder eine explizit genannte Verfasserin vorhanden sind, wodurch Teile des Kontexts in Onlinetexten nicht vorhanden sind (vgl. Gambier und Lautenbacher 2024, S. 8).

Ganz Ähnliches kann auch für die GUI von Softwareprodukten gesagt werden. Für den multimodalen Kontext verhält es sich allerdings genau umgekehrt, er ist sofort ersichtlich und auch relevant, wie Gambier und Lautenbacher wieder mit Bezug auf das Beispiel von Websites ausführen: „When localizing a website, for example, we immediately recognize that the layout, colors, and typographic characters, all with culturally marked value, transform our reading.“ (Gambier und Lautenbacher 2024, S. 8)

Vor diesem Hintergrund bietet sich also für die weitere Untersuchung ein Fokus einerseits auf den sprachlichen Kontext mit der semantischen und syntaktischen Ebene und andererseits auf den multimodalen Kontext an, also die außersprachlichen semiotischen Ebenen. Zunächst soll aber die Literatur zur Softwarelokalisierung daraufhin überprüft werden, welche Aspekte des (fehlenden) Kontexts darin thematisiert werden. Die folgende Liste enthält

Beispiele für diese verschiedenen Aspekte:

1. „When translating string sections of software applications, translators will frequently be faced with many issues and problems that need to be resolved, such as *lack of context*, fixed sentence orders, unclear variables, etc.“ (Esselink 2000, S. 73)¹⁰
2. „Aus einem kompletten Fenster oder Dialog lassen sich ggf. *Kontextinformationen* ablesen.“ (Ottmann 2005, S. 103)
3. „Items used in menus and dialog boxes are usually grouped so the translators can view *the context of these items*. An example of this kind of problem is where all the options used within one particular dialog box would be displayed together.“ (Esselink 2000, S. 59)
4. „Werden die Textelemente mit Hilfe eines Lokalisierungstools übersetzt, wird der *Kontext* in der Regel durch die WYSIWYG-Darstellung deutlich [...].“ (Wahle 2000b, S. 45)
5. „When working with software strings, translators very often have to guess if, how, and where a particular string may be displayed in the software application.“ (Esselink 2000, S. 59)
6. „Um ein Textelement in die Zielsprache übertragen zu können, muss der Übersetzer den *Kontext* kennen. Neben dem *sprachlichen Kontext* muss auch ersichtlich sein, ob es sich bei dem Text um einen Befehl, eine Meldung oder z.B. um die Beschreibung einer Schaltfläche handelt [...].“ (Wahle 2000b, S. 44)
7. „Die Textelemente müssen aber *kontextabhängig* übersetzt werden. Ein Beispiel für fehlende Eindeutigkeit ist das Wort ‚Bestellung‘, das im Englischen mit ‚purchase order‘, ‚manufacturing order‘ oder ‚customer order‘ übersetzt werden kann, je nachdem, in welchem Bereich eine Bestellung aufgegeben wird.“ (Ottmann 2005, S. 105–106)

Diese Beispiele nehmen einerseits häufig Bezug auf die verschiedenen Arten der GUI-Lokalisierung, die in Abschnitt 2.1.4 beschrieben wurden. Andererseits nehmen sie aber auch Bezug auf verschiedene Aspekte des Begriffs *Kontext*. Das erste Zitat in dieser Liste ist ein Beispiel für Fälle, bei denen gar nicht deutlich wird, wie das Wort *Kontext* zu verstehen ist. Informationen dazu sind nur sehr implizit vorhanden. Bei Zitat 2 spielt die Anordnung der Strings in der GUI eine Rolle, es wird aber nicht konkret gesagt, welche Kontextinformationen im genannten Fall konkret genutzt werden können. In Zitat 3 wird als Kontextinformation primär der direkte sprachliche Kontext in Form der Strings genannt, die in der GUI zusammen gruppiert sind,

¹⁰Die Hervorhebung in Kursivschrift hier und in den folgenden Beispielen wurde vom Verfasser der vorliegenden Arbeit zur Verdeutlichung vorgenommen.

beispielsweise in einem Menü oder Dialogfeld. Bei Zitat 4 liegt der Fokus wiederum auf dem direkten sprachlichen Kontext, allerdings ist über den Verweis auf die WYSIWYG-Darstellung der multimodale Kontext ebenso vorhanden. Das fünfte Zitat betrifft Strings, die nicht in Menüs oder Dialogfeldern angesiedelt sind (s. Abschnitt 2.1.4), für die also weder direkter sprachlicher Kontext noch multimodaler Kontext bereitgestellt werden kann. Zitat 6 legt den Fokus klar auf den multimodalen Kontext in Form der Typen von GUI-Elementen, für die ein String zum Einsatz kommt, der sprachliche Kontext wird aber ebenfalls genannt. Das letzte Zitat, Zitat 7, beschreibt primär den fachlichen Kontext.

Zusammenfassend kann also gesagt werden, dass an verschiedenen Stellen der Literatur aus dem Bereich der Softwarelokalisierung folgende Aspekte thematisiert werden:

- Der direkte sprachliche Kontext, also die Strings, die in der GUI gleichzeitig und in direkter Umgebung des zu übersetzenden Strings angesiedelt sind
- Der multimodale Kontext in Form des relevanten GUI-Bereichs und der Typen von GUI-Elementen, in denen der zu übersetzende String angezeigt wird
- Der fachliche Kontext

Allerdings sind dies nicht die einzigen Arten von Kontext bei der GUI-Lokalisierung. Die erste der bislang noch nicht genannten Kontextarten befasst sich mit dem sprachlichen Kontext, für den es in Bezug auf GUI-Text zwei Betrachtungsarten gibt: Zunächst den direkten sprachlichen Kontext, von dem die Zitate oben in Teilen handeln, also alle Strings, die in einem einzigen GUI-Bereich zu einem bestimmten Zeitpunkt angezeigt werden. In diesem Fall ist der direkte sprachliche Kontext jeweils eingegrenzt auf einen konkreten GUI-Bereich. Beispielsweise hat jedes Dialogfeld oder jedes Menü seinen eigenen direkten sprachlichen Kontext, der auch exakt abgegrenzt ist vom sprachlichen Kontext der anderen Dialogfelder oder Menüs. Andererseits könnte man aber auch die Gruppe aller Strings aus einem Softwareprodukt insgesamt als sprachlichen Kontext betrachten. Dies geht zum einen auf abstrakte Weise, indem alle Strings eines Softwareprodukts als sprachlicher Kontext betrachtet werden, andererseits aber konkret über die Inhalte der Ressourcendateien, in die die Strings zur Übersetzung ausgelagert sind – sofern dies der Fall ist (vgl. beispielsweise Reineke 2005, S. 74, Freigang und Reinke 2005, S. 62, und Behrens 2016, S. 46). Wie rechts in Abbildung 2.1 zu sehen ist, ergibt sich daraus eine zweite Art von direktem sprachlichen Kontext, der aus den in den Ressourcendateien vor und nach dem betreffenden

String aufgeführten Strings besteht, aber auch noch weitere Informationen aufweist. Diese beiden Arten von Kontext, der sprachliche Kontext in der GUI und der sprachliche Kontext in der Ressourcendatei, können in manchen Fällen sehr ähnlich sein, in anderen aber sehr unterschiedlich, wie die Abbildung zeigt. Weitere Beispiele und Details zum sprachlichen Kontext in der Ressourcendatei finden sich unten.

Als zweite weitere Kontextart, die in den obigen Zitaten nicht genannt wurde, können bei der Softwarelokalisierung auch die in Abschnitt 4.3 oben beschriebenen Programmfunktionen betrachtet werden, wie Melby deutlich macht:

Instead of only referring to the next and previous sentences in a document, context may include the purpose of a button in a user interface. For example, a button labelled „start“ on a user interface that controls a medical device may cause it to begin scanning a patient, while a „start“ button on a video editing software package may capture the beginning time code of a video segment. (Melby und Foster 2010, S. 5)

Bei GUI-Text lassen sich also insgesamt vier Kontextarten unterscheiden.¹¹

Der situative Kontext Die Aspekte der Multimodalität von GUIs, die in Abschnitt 4.1 erläutert wurden, führen dazu, dass als Kontext für einzelne Strings in der GUI neben den sie umgebenden Strings als sprachlichem Kontext auch Informationen verfügbar sind, die aus den verwendeten grafischen und typografischen Gestaltungsmitteln sowie der hierarchischen Anordnung der Strings und den verwendeten GUI-Elementen hervorgehen. Diese Art von Kontext setzt sich also aus sprachlichen und außersprachlichen Aspekten zusammen und wird von Freigang und im weiteren Verlauf dieser Arbeit als *situativer Kontext* (vgl. Freigang 1997, S. 124) bezeichnet.¹²

¹¹Vgl. dazu auch die Masterarbeit von Kamila Gagola, die eine Untersuchung auf Basis einer vergleichbaren Einteilung durchführt (vgl. Gagola 2017, S. 31–34).

¹²Schmitz gibt dafür im Zusammenhang mit Terminologearbeit folgende Erklärung: „Bei den meisten sprachlichen Elementen in Menüs und Dialogfeldern gibt keinen sprachlichen Kontext wie er sonst bei Benennungen in Fachtexten üblich ist. Dafür ist aber der situative, d. h. der software-technische Umgebungskontext wichtig und für die Differenzierung von Begriffen und die korrekte Übersetzung der Einheiten ausschlaggebend.“ (K.-D. Schmitz 2008, S. 273–274) Auch Beste nennt den Begriff *situativer Kontext*, allerdings weniger konkret, da er ihn u. a. in Zusammenhang mit der Funktion stellt, die in der vorliegenden Arbeit einer anderen Kontextart zugewiesen wird: „Die zu überset-

Damit ist der situative Kontext auch der Aspekt des Kontexts bei der Lokalisierung von GUIs, der häufig im Zusammenhang mit der Nutzung von Softwarelokalisierungstools gemeint ist. Diese ermöglichen in der Regel das Übersetzen im situativen Kontext, also beispielsweise in den Menüs oder Dialogfeldern (vgl. beispielsweise Ottmann 2005, S. 102–103, Beste 2006, S. 93–94, Reineke 2005, S. 79, Wahle 2000b, S. 37–38, 40, K.-D. Schmitz 2005a, S. 9, Schildhauer 2005, S. 124, Roturier 2015, S. 97, und Austermühl 2001, S. 146).¹³

Der fachliche Kontext GUI-Text weist als Fachtext (s. Abschnitt 4.2) einen fachlichen Kontext auf, der insbesondere für die Übersetzung von Fachtermini relevant ist (s. dazu ebenso Abschnitt 3.4.7). Melby und Foster führen dazu das Beispiel „The transaction isolation level must be dirty read for local databases“ an. Mit Bezug auf die darin genannten Benennungen „transaction isolation level“ und „dirty read“ erläutern sie: „In technical translation, not only is there sometimes a lack of context for strings that are presented to translators, there is often a lack of rel-text in the form of explanatory material.“ (Melby und Foster 2010, S. 9)¹⁴ Hier sprechen sie also konkret von Fachübersetzung als Kontext bzw. Rel-Text und betonen auch, dass es in diesem Fall nicht ausreicht, zu sehen, wo in der Anzeige die Nachricht positioniert ist (vgl. Melby und Foster 2010, S. 9).¹⁵ Der situative Kontext ist hier also weder ausreichend noch hilfreich. Sie verweisen dagegen auf bestimmte Arten von Rel-Text, im genannten Beispiel auf Hintergrunddokumente zur

zenden Einheiten sind aus dem Zusammenhang gerissen, und der Übersetzer kann ihren situativen Kontext nicht direkt aus dem Text erschließen. In zahlreichen Fällen ist daher die Funktion unklar oder ambivalent, existieren in der Zielsprache mehrere Möglichkeiten der Übersetzung, können logische Fehler entstehen.“ (Beste 2006, S. 73) Auch andere Autorinnen nutzen diesen Begriff, allerdings in anderen Zusammenhängen, beispielsweise Adamzik für den außersprachlichen Kontext in der in Abschnitt 3.4.2 erläuterten Form (vgl. Adamzik 2016, S. 114).

¹³An manchen Stellen wird auch der umgekehrte Fall thematisiert, dass beim Übersetzen in den Quelldateien der situative Kontext verloren geht (vgl. Ottmann 2005, S. 103, und Wahle 2000b, S. 37). In Softwarelokalisierungstools können allerdings nicht alle Strings im situativen Kontext übersetzt werden, weil nicht für alle Strings situativer Kontext vorhanden ist (vgl. Schildhauer 2005, S. 124; s. auch Wahle 2000b, S. 41, und Sachse 2005, S. 147).

¹⁴Schmitz stellt wie oben schon angegeben analog fest, dass bei den meisten Strings in der GUI kein sprachlicher Kontext wie in anderen Fachtexten bei Benennungen vorhanden ist (vgl. K.-D. Schmitz 2008, S. 273–274).

¹⁵Zudem stellen sie klar, dass Übersetzerinnen im Bereich der Softwarelokalisierung derartige Probleme regelmäßig begegnen (vgl. Melby und Foster 2010, S. 10), sie stellen damit also den expliziten Bezug zur Softwarelokalisierung her, auch wenn solche terminologischen Probleme grundsätzlich in verschiedenen Bereichen der Fachübersetzung auftreten.

thematisierten Datenbank, sowie auf substanzielle Vorkenntnisse aus dem Bereich IT (also in ihrer Terminologie auf Nicht-Text), aber ebenso auf eine Terminologiedatenbank bzw. ein projektspezifisches Glossar (vgl. Melby und Foster 2010, S. 9–10). Der fachliche Kontext kann somit sowohl sprachlich (in Form von Rel-Text) als auch außersprachlich (in Form von Nicht-Text) vorliegen.¹⁶

Der funktionale Kontext Der Zusammenhang, der zwischen Strings und zugehörigen Programmfunktionen (s. Abschnitt 4.3) besteht, stellt einen eigenen Typ von Kontext dar, wie bereits oben anhand von Melbys und Fosters Beispiel mit der Schaltfläche **start** in verschiedenen Softwareprodukten festgestellt wurde (vgl. Melby und Foster 2010, S. 5). Schmitz nutzt dafür die auch im weiteren Verlauf dieser Arbeit eingesetzte Bezeichnung *funktionaler Kontext* (K.-D. Schmitz 2021, S. 10)¹⁷ und betont an anderer Stelle ergänzend, dass alle Elemente beispielsweise eines Dialogfeldes „jeweils eine genau definierte Funktionalität innerhalb der Software“ (K.-D. Schmitz 2008, S. 271) aufweisen. Esselink weist Übersetzerinnen auf diese Art von Kontext hin, indem er ihnen die folgende Arbeitsanweisung gibt: „Always verify the meaning or function of each software option, and use a word in the target language that accurately describes that feature.“ (Esselink 2000, S. 66) Diese Art von Kontext ist somit primär außersprachlich, kann aber, wie in Abschnitt 4.5 erläutert werden wird, auch in Form von Beschreibungen der jeweiligen Funktionen vorliegen (also in der Terminologie von Melby und Foster als Rel-Text).

Der Kontext in der Ressourcendatei In Abschnitt 2.1.4 wurde bereits beschrieben, dass die zu übersetzenden Strings bei lokalisierten Softwareprodukten häufig in separate Ressourcendateien ausgelagert werden (s. auch beispielsweise Reineke 2005, S. 74, Freigang und Reinke 2005, S. 62, und Behrens 2016, S. 46). In diesem Fall geht die Multimodalität verloren, was diese Dateien nach Kim zu Nekrotexten macht (s. Abschnitt 4.1 und vgl. Kim 2019, S. 120). Neben allen Strings, die in der GUI Verwendung finden, enthalten diese Ressourcendateien aber auch weitere Informationen:

¹⁶Konkret zu dem aufgeführten Beispiel erläutern sie weiter: „For example, ‚dirty‘ in this context means that a change has been made to a row of information in a database but that change has not yet been committed (i.e. it is not yet final).“ (Melby und Foster 2010, S. 10)

¹⁷Diese Veröffentlichung ist inzwischen allerdings nicht mehr online zu finden (vgl. dazu auch Gagola 2017, S. 31).

- Die String-IDs (s. Abschnitt 2.1.4)
- Kommentare und Anweisungen für Übersetzerinnen (s. Abschnitt 2.1.4)
- Die in GUI-Text enthaltenen Steuerzeichen und zeichenbasierten Konventionen (s. Abschnitt 2.2)
- Unter Umständen Teile des Quellcodes des Softwareprodukts¹⁸

Bei allen diesen Informationen handelt es sich um sprachlichen Kontext in Form von Zeichen. Ein Beispiel dafür findet sich in Abbildung 2.1, in der ein paar Strings, Steuerzeichen, String-IDs und Kommentare zur Zuordnung der Strings in der GUI enthalten sind. Ein weiteres Beispiel ist in Abbildung 4.3 zu sehen, das einen Ausschnitt aus einer der Ressourcendateien von Mozilla Firefox Version 6.6.0.3 zeigt. Darin finden sich einige Kommentare (jeweils zwischen den Zeichen `<!--` und `-->`), von denen einige als Inhalt `LOCALIZATION NOTE` und ggf. weitere Informationen, z. B. `DONT_TRANSLATE` oder auch weitere Erläuterungen zur Übersetzung wie `"Pin" is being used as a metaphor for expressing the fact that these tabs are "pinned" to the left edge of the tabstrip` haben. Bei den String-IDs, wie `mainWindow.title`, `appmenu.tooltip`, `pinTab.label` und `pinTab.accesskey`, gibt der Teil nach dem Punkt einen Hinweis auf das GUI-Element, den GUI-Bereich oder die Funktion des Strings. Ebenso können die einander zugeordneten Strings selbst auch als Kontext dienen, wie das Beispiel in Abbildung 4.4 zeigt. Der erste String darin, `Open`, ist potenziell mehrdeutig, damit kann das Verb „(to) open“ oder das Adjektiv „(something is) open“ gemeint sein, die unterschiedlich übersetzt werden müssten.¹⁹ Betrachtet man die weiteren Strings, `Open script`, `Save`, `Save script`, `Save as`, `Save script as...`, so lässt sich daraus die Vermutung ableiten, dass es sich jeweils um Befehle, vielleicht mit zugeordnetem Tooltip handelt, der etwas ausführlicher gestaltet ist. Also ist beim String `Open` vermutlich das Verb gemeint und die Alternative des Adjektivs ist sehr wahrscheinlich falsch. Diese Information lässt sich selbstverständlich auch aus der GUI beziehen, sofern die genannten Strings in einem einzigen situativen Kontext angeordnet sind. Sie ist aber in der Ressourcendatei allein, also ohne situativen Kontext, der weitere Informationen liefert, vorhanden und zudem auch dann, wenn diese Strings in der GUI in verschiedenen situativen Kontexten zu finden sind. Auf ähnliche Weise ist denkbar,

¹⁸Je nachdem, wie umfassend die Internationalisierung und damit die Trennung von Quellcode und Strings ist (s. Abschnitt 2.1.4).

¹⁹Weitere Informationen zu diesem Problem finden sich in Abschnitt 4.6.1.

```

<!-- This Source Code Form is subject to the terms of the Mozilla Public
- License, v. 2.0. If a copy of the MPL was not distributed with this
- file, You can obtain one at http://mozilla.org/MPL/2.0/. -->

<!-- LOCALIZATION NOTE : FILE This file contains the browser main menu items -->
<!-- LOCALIZATION NOTE : FILE Do not translate commandkeys -->

<!-- LOCALIZATION NOTE (mainWindow.title): DONT_TRANSLATE -->
<ENTITY mainWindow.title "&brandFullName;">
<!-- LOCALIZATION NOTE (mainWindow.titlemodifier) : DONT_TRANSLATE -->
<ENTITY mainWindow.titlemodifier "&brandFullName;">
<!-- LOCALIZATION NOTE (mainWindow.titlemodifiermenuseparator): DONT_TRANSLATE -->
<ENTITY mainWindow.titlemodifiermenuseparator " - ">
<!-- LOCALIZATION NOTE (mainWindow.titlePrivateBrowsingSuffix): This will be appended to the window's title
inside the private browsing mode -->
<ENTITY mainWindow.titlePrivateBrowsingSuffix "(Private Browsing)">

<ENTITY appmenu.tooltip "Open menu">
<ENTITY navbarOverflow.label "More tools...">

<!-- Tab context menu -->

<!-- LOCALIZATION NOTE (pinTab.label, unpinTab.label): "Pin" is being
used as a metaphor for expressing the fact that these tabs are "pinned" to the
left edge of the tabstrip. Really we just want the string to express the idea
that this is a lightweight and reversible action that keeps your tab where you
can reach it easily. -->
<ENTITY pinTab.label "Pin Tab">
<!-- LOCALIZATION NOTE(pinTab.accesskey, pinSelectedTabs.accesskey,
unpinTab.accesskey, unpinSelectedTabs.accesskey): These share the
same accesskey but will never be visible at the same time. -->
<ENTITY pinTab.accesskey "P">
<ENTITY unpinTab.label "Unpin Tab">
<!-- LOCALIZATION NOTE(pinTab.accesskey, pinSelectedTabs.accesskey,
unpinTab.accesskey, unpinSelectedTabs.accesskey): These share the
same accesskey but will never be visible at the same time. -->

```

Abbildung 4.3: Ausschnitt aus der Datei `browser.dtd` von Mozilla Firefox Version 6.6.0.3

```

#: lib-src/mod-nyq-bench/NyqBench.cpp:1716
msgid "Open"
msgstr ""

#: lib-src/mod-nyq-bench/NyqBench.cpp:1716
msgid "Open script"
msgstr ""

#: lib-src/mod-nyq-bench/NyqBench.cpp:1717
msgid "Save"
msgstr ""

#: lib-src/mod-nyq-bench/NyqBench.cpp:1717
msgid "Save script"
msgstr ""

#: lib-src/mod-nyq-bench/NyqBench.cpp:1718
msgid "Save As"
msgstr ""

#: lib-src/mod-nyq-bench/NyqBench.cpp:1718
msgid "Save script as..."
msgstr ""

```

Abbildung 4.4: Ausschnitt aus der Datei `audacity.pot` von Audacity Version 2.2.2

dass in der Ressourcendatei ein Wort, das Verständnisschwierigkeiten verursacht, an mehreren Stellen vorkommt, die in der Summe Rückschlüsse auf seine Bedeutung enthalten oder von denen zumindest eine weitere Informationen bietet (vergleichbar mit dem Beispiel, das Melby und Foster nennen, bei dem an einer Stelle eine informelle Definition eines Worts vorkommt: vgl. Melby und Foster 2010, S. 6).

Im folgenden Abschnitt werden diese vier Kontextarten in Zusammenhang mit dem Referenzmaterial gebracht, das üblicherweise als notwendig erachtet wird, um GUI-Text mit hoher Qualität übersetzen zu können.

4.5 GUI-Text und Referenzmaterial

Im Bereich der Softwarelokalisierung ist die Beziehung zwischen Kontext und Referenzmaterial sehr eng, was sich schon allein an der von Ottmann in ihrem Beitrag mit dem Titel „Software-Lokalisierung“ gewählten Abschnittsüberschrift „Kontextinformationen und Referenzmaterial“ (Ottmann 2002, S. 149) zeigt. Einleitend schreibt sie in diesem Abschnitt: „Damit eine sinnvolle Übersetzung möglich ist, müssen den Übersetzern Kontextinformationen und Referenzmaterial zur Verfügung gestellt werden.“ (Ottmann 2002,

S. 149)²⁰ Ebenso verwendet Wahle in ihrem Beitrag „Wie wird Software lokalisiert?“ als Abschnittsüberschrift „Mittel zur Kontextbestimmung“ (Wahle 2000b, S. 45). In diesem Abschnitt zählt sie verschiedene der im Folgenden genannten Referenzmaterialien auf.

Auf diese Weise machen Ottmann und Wahle deutlich, dass Informationen zum Kontext über die Nutzung von geeignetem Referenzmaterial ermittelt werden können. Auf ganz ähnliche Weise argumentieren Melby und Foster mit Verweis auf Baker:

She suggests that instead of treating context as a constraint, „a set of restrictions on what we can or cannot achieve in translation and other communicative events,“ it might be more productive to „recognize context as a resource.“ Although it may not be exactly what Baker had in mind, the present article indeed treats context as a set of resources that need to be available to translators. (Melby und Foster 2010, S. 3)²¹

Im Folgenden wird dieser Zusammenhang zwischen Kontext und Referenzmaterial herausgearbeitet, indem zunächst das in der Literatur zur Softwarelokalisierung genannte Referenzmaterial aufgeführt wird. Anschließend wird aufgezeigt, für welche der im vorangegangenen Abschnitt herausgearbeiteten Kontextarten (s. Abschnitt 4.4) das Referenzmaterial hilfreich sein kann. Eine Sonderstellung nimmt dabei allerdings der Kontext in der Ressourcendatei ein, da Inhalte wie Kommentare oder String-IDs in den Ressourcendateien selbst als Referenzmaterial genannt werden, was aufgrund der Erläuterungen oben wenig verwunderlich erscheint. Daher wird diese Kontextart im weiteren Verlauf dieser Arbeit nicht separat aufgeführt, sondern als weiteres Referenzmaterial behandelt.

Nachfolgend findet sich die Liste der verschiedenen Referenzmaterialien, die in der Literatur zur Softwarelokalisierung genannt werden. Jeweils zugeordnet sind Angaben dazu, für welche Kontextarten sie jeweils sinnvoll sein können:²²

²⁰In einer späteren Veröffentlichung wählt sie unter derselben Überschrift sehr ähnliche Worte: „Nur wenn ausreichend Kontextinformationen und Referenzmaterial zur Verfügung stehen, ist eine sinnvolle Übersetzung möglich. Übersetzer müssen wissen, in welchem Zusammenhang die einzelnen Textelemente stehen, die sie übersetzen sollen.“ (Ottmann 2005, S. 104)

²¹Gemeint ist hier Mona Bakers Veröffentlichung Baker 2006. Für die von Melby und Foster übernommenen Zitate aus dieser Veröffentlichung vgl. Baker 2006, S. 332.

²²Ottmann (vgl. Ottmann 2005, S. 104) und insbesondere Esselink (Esselink 2000, S. 53–54) führen neben den im Folgenden genannten Referenzmaterialien weitere auf, die nicht zur Bereitstellung von Kontextinformationen, sondern für andere Zwecke dienen, z. B.

Lauffähige Version der zu übersetzenden Software (vgl. Ottmann 2005, S. 102, und Esselink 2000, S. 53) bzw. Simulationen und Demos, falls dies nicht möglich ist (vgl. Ottmann 2005, S. 102). Es lässt sich annehmen, dass dieses Referenzmaterial für folgende Kontextarten und Zwecke sinnvoll ist:

- **Situativer Kontext:** In der Software kann sowohl festgestellt werden, welche Strings gemeinsam gruppiert sind, als auch, in welchen GUI-Elementen sie in welcher hierarchischen Anordnung vorliegen.
- **Fachlicher Kontext:** In der Software können relevante Fachgebiete für das Softwareprodukt ermittelt werden.
- **Funktionaler Kontext:** In der Software können die jeweils mit den Strings verknüpften Funktionen identifiziert bzw. ausprobiert werden.

Dokumentation des Softwareprodukts in Form von Hilfeinformationen bzw. Onlinehilfe (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, und Wahle 2000b, S. 45) oder auch „Referenzhandbücher mit Beschreibung der einzelnen [Text-]Elemente“ (Ottmann 2005, S. 104). Es lässt sich annehmen, dass dieses Referenzmaterial für folgende Kontextarten und Zwecke sinnvoll ist:

- **Situativer Kontext:** Diesem Material können Informationen dazu entnommen werden, welche Strings gemeinsam gruppiert sind und in welchen GUI-Elementen sie in welcher hierarchischen Anordnung vorliegen.
- **Fachlicher Kontext:** Anhand dieses Materials können die relevanten Fachgebiete für das Softwareprodukt ermittelt werden.
- **Funktionaler Kontext:** Diesem Material können Beschreibungen der jeweils mit den Strings verknüpften Funktionen entnommen werden.

Kommentare der Entwicklerinnen in den Ressourcendateien (vgl. Esselink 2000, S. 34, 64, Wahle 2000b, S. 37–38, 40, 45, Reineke 2005, S. 85, Roturier 2015, S. 64–65, und Sachse 2005, S. 148, 158). Es lässt

Übersetzungen früherer Produktversionen oder anderer Produkte desselben Herstellers (vgl. dazu auch Melbys und Fosters Kontextmodell in Abschnitt 3.4.5). Seewald-Heeg weist außerdem darauf hin, dass aufgrund des mangelnden Kontexts Rückfragen an die Auftraggeberin notwendig sind (vgl. Seewald-Heeg 2020, S. 15). Da solche Rückfragen aber grundsätzlich eine Lösung für alle Arten von Problemen sein können und somit keine weiteren Rückschlüsse zulassen, werden sie in der folgenden Liste nicht separat aufgeführt.

sich annehmen, dass dieses Referenzmaterial für folgende Kontextarten und Zwecke sinnvoll ist (s. auch die Beispiele für den Kontext in Ressourcendateien in Abschnitt 4.4):

- **Situativer Kontext:** In Kommentaren können Informationen dazu enthalten sein, welche Strings gemeinsam gruppiert sind und in welchen GUI-Elementen sie in welcher hierarchischen Anordnung vorliegen.
- **Fachlicher Kontext:** In Kommentaren können Informationen zum fachlichen Kontext der zu übersetzenden Strings enthalten sein.
- **Funktionaler Kontext:** In Kommentaren können Informationen zu den mit den jeweiligen Strings verknüpften Funktionen enthalten sein.

Andere Arten von Informationen in den Ressourcendateien (vgl. Wahle 2000b, S. 37), wie z. B. die verwendeten String-IDs (vgl. Wahle 2000b, S. 40, und Sachse 2005, S. 147–148, 158). Es lässt sich annehmen, dass dieses Referenzmaterial für folgende Kontextart und folgenden Zweck sinnvoll ist (s. auch die Beispiele für den Kontext in Ressourcendateien in Abschnitt 4.4):

- **Situativer Kontext:** Den String-IDs in vielen Softwareprodukten können Informationen dazu entnommen werden, welche Strings gemeinsam gruppiert sind und in welchen GUI-Elementen sie in welcher hierarchischen Anordnung vorliegen. Ein Beispiel dazu findet sich in Abschnitt 2.1.3 bei der Erläuterung zu String-IDs, ein weiteres in Abschnitt 4.4.²³

Glossare, z. B. für Betriebssystem, Softwareprodukt oder Fachgebiet (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, 401–402). Es lässt sich annehmen, dass dieses Referenzmaterial für folgende Kontextart und folgenden Zweck sinnvoll ist:

- **Fachlicher Kontext:** Glossare decken in der Regel zumindest in Teilen die Bereiche ab, aus denen sich die Fachterminologie des jeweiligen Softwareprodukts zusammensetzt.

²³Freigang und Reinke weisen in diesem Zusammenhang in Bezug auf RC-Dateien darauf hin, dass man sich „– vorausgesetzt man verfügt über Grundkenntnisse in der Interpretation von Programmcode in Ressourcendateien – zumindest eine grobe Vorstellung vom unmittelbaren Kontext eines Textelements machen kann“ (Freigang und Reinke 2005, S. 65).

Hintergrundinformation zum Produkt (vgl. Esselink 2000, S. 53). Es lässt sich annehmen, dass dieses Referenzmaterial für folgende Kontextart und folgenden Zweck sinnvoll ist:

- **Fachlicher Kontext:** Diese Informationen lassen Rückschlüsse auf die relevanten Fachgebiete zu.

Diese Auflistung sagt allerdings nichts darüber aus, ob im Einzelfall auch alle Kontextinformationen verfügbar sind, die theoretisch über dieses Referenzmaterial eingeholt werden können. Beispielsweise kann es schwierig sein, in der lauffähigen Version der zu übersetzenden Software die Stelle zu finden, deren situativer Kontext benötigt wird. Ebenso werden in der Dokumentation nicht immer alle Strings beschrieben und die Ressourcendateien enthalten in der Regel nur für einzelne der zu übersetzenden Strings hilfreiche Informationen, in manchen Fällen auch gar keine.

Für Softwarelokalisierungsprojekte und das dafür verfügbare Referenzmaterial bzw. die dafür verfügbaren Kontextinformationen kann man also von einem Kontinuum ausgehen: Am einen Ende, wenn lediglich extrahierte Strings ohne weitere Angaben alphabetisch sortiert als Liste vorliegen (s. Abschnitt 2.1.4), haben Übersetzerinnen praktisch keine Kontextinformationen. Der Extremfall davon liegt vor, wenn nur ein kleines Update mit wenigen Strings übersetzt werden muss, ohne dass Zugriff auf die weiteren Strings besteht, so dass nicht einmal über die Menge der neu zu übersetzenden Strings oder der bereits übersetzten Strings, auf die Zugriff besteht, kontextrelevante Informationen bezogen werden können. Wenn sämtliche der genannten Referenzmaterialien vorliegen, also eine lauffähige Version der Software, Dokumentation, Kommentare und weitere Informationen in den Ressourcendateien, Glossare und Hintergrundinformationen, und diese Referenzmaterialien auch alle benötigten Informationen enthalten, haben Übersetzerinnen alle relevanten Kontextinformationen. Dies wird in Abbildung 4.5 verdeutlicht.

Im folgenden Abschnitt werden nun typische Probleme bei der Übersetzung von Strings beschrieben, die exemplarisch in der Literatur zur Softwarelokalisierung genannt werden, und mithilfe der hier vorgestellten Kontextarten kategorisiert.

chen je nach Position in der GUI drei oder mehr Übersetzungen möglich sein (vgl. Esselink 2000, S. 63). Wenn die Position in der GUI, also der situative Kontext, bekannt ist, können diese Probleme gelöst werden.

Ein weiterer Ausgangspunkt für einen Teil dieser Probleme ist die Nutzung von Englisch als der Sprache, in der die meisten Softwareprodukte erstellt werden (vgl. Drewer und Ziegler 2014, S. 50, und Esselink 2003, S. 84), was auch an den Beispielen dieses Abschnitts deutlich wird.

Die Übersetzungsprobleme, für deren Lösung der situative Kontext hilfreich ist, können in die folgenden Kategorien unterteilt werden:

Ambige Wortarten Wenn eines der Wörter in einem kurzen String einem der „universell einsetzbaren Wortstämme“ (Beste 2006, S. 73) im Englischen angehört, führt dies in vielen Fällen zu Verständnis- und damit Übersetzungsproblemen (vgl. Beste 2006, S. 73), weil die intendierte Wortart nicht zweifelsfrei erkennbar ist.²⁴ Als Beispiel gibt Beste einen String an, bei dem das Wort „record“ als Verb oder als Substantiv betrachtet werden kann: „*Record error* kann sowohl *Fehler aufzeichnen* als auch *Aufzeichnungsfehler* bedeuten“ (Beste 2006, S. 73, Hervorhebung im Original). Ottmann nennt zudem noch das Wort „open“, das als Verb oder Adjektiv betrachtet werden kann (vgl. Ottmann 2005, S. 103).²⁵ Es ist allerdings anzunehmen, dass es mehr als diese beiden Varianten ambiger Wortarten gibt. In Abschnitt 5.5.1.1 finden sich sämtliche Varianten ambiger Wortarten, die im Korpus identifiziert wurden, zusammen mit Beispielen aus dem Korpus.

Bezug auf andere GUI-Elemente Melby und Foster diskutieren ein Beispiel, bei dem ein String nur aus dem Wort „in“ besteht: „How does one translate a preposition without knowing what verb it collocates with and what object it has?“ (Anhang 1 in Melby und Foster 2010, S. 1)²⁶

²⁴Für Albrecht ist im Englischen „der Übergang vom Verb zum Substantiv oder umgekehrt eher die Regel als die Ausnahme“ (Albrecht 2013, S. 107).

²⁵Weitere Beispiele zu der Variante Verb-Substantiv finden sich bei Esselink (das Wort „copy“, das als Verb oder Substantiv betrachtet werden kann; vgl. Esselink 2000, S. 63), bei Ottmann („Run Counter“ und „Change Order“, wo „Run“ bzw. „Change“ als Verb oder Substantiv betrachtet werden kann; vgl. Ottmann 2005, S. 106) und bei Wahle („Display Options“ mit „Display“ als Verb oder Substantiv; vgl. Wahle 2000b, S. 44). Drewer und Ziegler greifen Ottmanns Beispiel „Change Order“ auf und weisen explizit auf den situativen Kontext als Basis für die Lösung dieser Ambiguität hin (vgl. Drewer und Ziegler 2014, S. 48–49). Dunne nennt das Beispiel der Schaltfläche „Install Help“, bei der „Install“ Verb oder Substantiv sein kann, verweist dabei aber auf das Ausprobieren, also den funktionalen Kontext, mit dem solche Probleme ebenso gelöst werden können (vgl. K. J. Dunne 2009, S. 203).

²⁶Dieses Beispiel übernehmen sie von Dunne, der seinerseits am Beispiel von „in“ auf

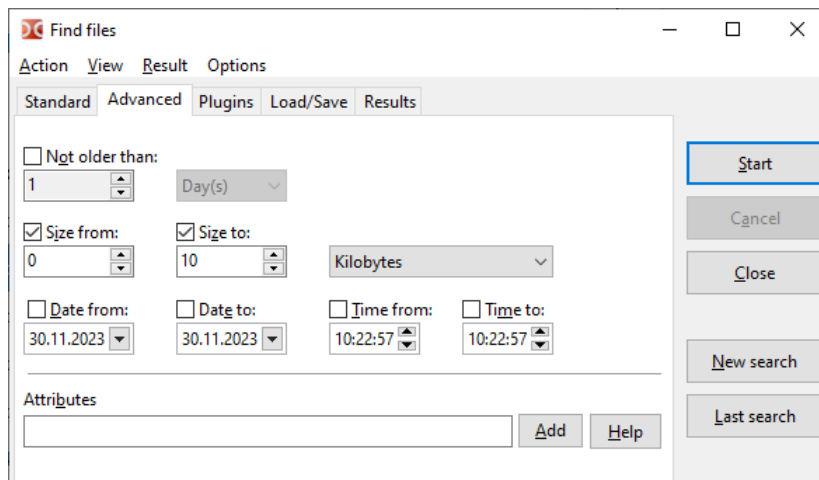


Abbildung 4.6: Dialogfeld **Find files** in Double Commander Version 0.9.2

Unabhängig davon, ob die Präposition mit einem Verb verbunden ist und mit diesem eine Kollokation bildet oder welche Art von Objekt auf die Präposition folgt, liegt eine grammatische Unvollständigkeit vor, die die Übersetzung behindert. Vollständigkeit wird durch den situativen Kontext in der GUI hergestellt, indem diesem String in der GUI ein anderer String oder ein bestimmtes GUI-Element zugeordnet wird. Der andere String liefert entweder das kollokierende Verb oder das Objekt, das GUI-Element tritt an die Stelle des Objekts. Ein Beispiel für letzteren Fall findet sich in Abbildung 4.6, in dem das Dialogfeld **Find files** in Double Commander Version 0.9.2 und darin verschiedene Dropdown-Listen zu sehen sind, auf die sich Strings mit einer Präposition am Ende (**Not older than**, **Size from ...**) beziehen.

Eine weitere Möglichkeit des Bezugs auf andere GUI-Elemente stellen deiktische Ausdrücke dar:

Deiktische Ausdrücke sind eine spezielle Klasse von sprachlichen Ausdrücken, mit denen Sprecher referieren können. Ihre Referenz wird nicht darüber vermittelt, dass sie bestimmte Eigenschaften von etwas beschreiben, sondern ergibt sich aus der jeweiligen Äußerungssituation. Deiktische Ausdrücke fungieren wie Variablen, die erst durch die Situierung in ei-

die verschiedenen Aspekte der Übersetzung einer einzelnen Präposition eingeht, die nach seiner Ansicht allerdings auch Teil einer Concatenation sein kann (vgl. K. J. Dunne 2006b, S. 103–104).

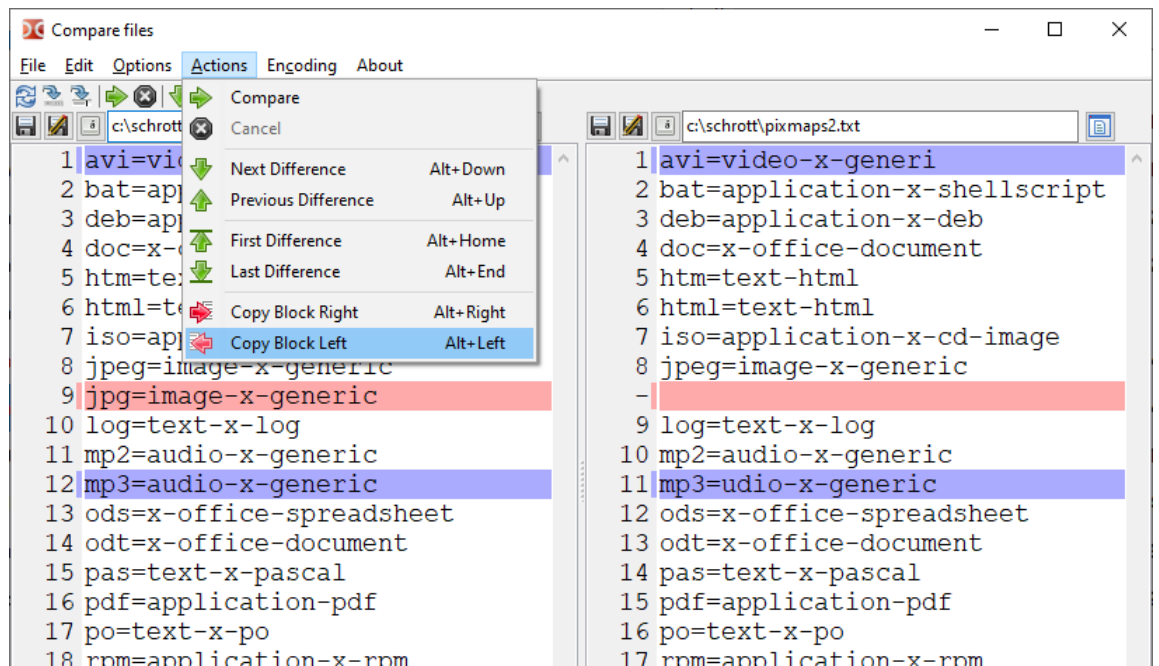


Abbildung 4.7: Dialogfeld **Compare files** in Double Commander Version 0.9.2

nem bestimmten Kontext „gefüllt“ werden können. (Finkbeiner 2015, S. 35)

Um deiktische Ausdrücke verstehen und übersetzen zu können, muss neben der semantischen Bedeutung also der Kontext bekannt sein. Dabei lassen sich drei Arten von Deixis unterscheiden, Personaldeixis, Lokaldeixis und Temporaldeixis (vgl. Finkbeiner 2015, S. 37). Von diesen drei Arten ist die Lokaldeixis für GUI-Text die relevanteste, weil damit räumliche Orientierungen, Richtungen und relative Positionen in der GUI ausgedrückt werden können. Allgemein wird Lokaldeixis „über Lokaladverbien (*hier, dort, da, oben, unten, ...*), lokale Präpositionen (*vor, hinter, neben, auf, in, ...*) und Demonstrativpronomen (*dieser, jener*) hergestellt.“ (Finkbeiner 2015, S. 38) In einem String können so andere Strings oder auch andere GUI-Elemente referiert werden. Beispielsweise sind in Abbildung 4.6 die beiden Menüoptionen **Copy Block Right** und **Copy Block Left** in Double Commander Version 0.9.2 zu sehen, mit denen markierte Textblöcke vom linken Teilfenster in das rechte und umgekehrt kopiert werden können.

In Abschnitt 5.5.1.2 sind diese beiden verschiedenen Arten von Bezügen

auf andere GUI-Elemente mit Beispielen aus dem Korpus enthalten.

Allein stehende Pronomen und Adjektive Bei der Übersetzung von allein stehenden Pronomen und Adjektiven kann die Festlegung der genauen grammatischen Form in flektierenden Sprachen problematisch sein, da zur Herstellung grammatischer Kongruenz das Geschlecht des Bezugsworts erforderlich ist (vgl. Beste 2006, S. 73). Esselink gibt dafür die Beispiele „All“ (vgl. Esselink 2000, S. 75–76) und „None“. Bei letzterem wäre beispielsweise der Bezug zu „File“ oder „Directory“ denkbar. Im Deutschen würde „None“ in Fall 1 mit „Keine“ (nämlich „Keine Datei“, also Femininum Singular) übersetzt, in Fall 2 mit „Kein“ (nämlich „Kein Verzeichnis“, also Neutrum Singular) (vgl. Esselink 2000, S. 63).²⁷ Beste nennt dabei explizit auch längere Strings, die allein stehende Pronomen oder Adjektive enthalten (vgl. Beste 2006, S. 73), die Beispiele in den anderen Veröffentlichungen betreffen allerdings lediglich Ein-Wort-Strings. In Abschnitt 5.5.1.3 finden sich sämtliche Varianten dieser Kategorie, die im Korpus identifiziert wurden, zusammen mit Beispielen aus dem Korpus.

4.6.2 Fachlicher Kontext und Probleme bei der Übersetzung von GUI-Text

In diesem Abschnitt finden sich Kategorien von Übersetzungsproblemen, für deren Lösung der fachliche Kontext (s. Abschnitt 4.4) benötigt wird. Wie in Abschnitt 4.2 dargestellt, zeigt sich die Fachlichkeit bei GUI-Text primär im Bereich der Lexik, also der Fachterminologie. In Teilen spielt dabei wie auch bei den Problemen des situativen Kontexts (s. Abschnitt 4.6.1) die Nutzung von Englisch als Ausgangssprache im überwiegenden Teil der Softwareprodukte (vgl. Drewer und Ziegler 2014, S. 50, und Esselink 2003, S. 84) eine Rolle, was an einigen der Beispiele in diesem Abschnitt deutlich wird. Ein weiterer Aspekt liegt in der Tatsache begründet, dass viele Softwareprodukte innovative Fachgebiete und Themen umfassen, für die in der Zielsprache

²⁷Wahle nennt dasselbe Beispiel und gibt dazu noch Übersetzungen ins Deutsche, bei denen die Flexionsproblematik vermieden wird, wie „**ohne, kein Eintrag** oder **keine Angabe**“ (Wahle 2000b, S. 45, Hervorhebungen im Original). Auch Reineke thematisiert die Übersetzung von „None“ (vgl. Reineke 2005, S. 84), sieht dafür aber die Hersteller der Software in der Bringschuld, ausreichende Informationen zur Übersetzung bereitzustellen. Diese Informationen könnten beispielsweise der situative Kontext sein. Dunne gibt ein Beispiel mit dem String „Global“ als Titel einer Registerkarte im Dialogfeld „Preferences“ an, wobei letzteres die sprachliche Ergänzung des allein stehenden Adjektivs ist, was nur im Kontext des Titels des Dialogfelds, also des situativen Kontexts, ersichtlich ist (vgl. K. J. Dunne 2009, S. 208–209).

noch keine etablierte Terminologie vorhanden ist (vgl. K.-D. Schmitz 2005a, S. 9).²⁸

Die Probleme im Bereich des fachlichen Kontexts lassen sich in verschiedene Kategorien aufteilen:

Fachterminologie In Strings kann Fachterminologie vorkommen, „wobei je nach Fachlichkeitsgrad des Produktes oder der Dienstleistung der Anteil der fachspezifischen Terminologie unterschiedlich groß sein kann“ (K.-D. Schmitz 2005a, S. 9). Die Fachterminologie kann dabei aus verschiedenen Teilgebieten kommen (s. Abschnitt 4.2) und zudem verschiedene Wortarten oder auch Mehrwortbenennungen umfassen (vgl. Drewer und Ziegler 2014, S. 178, Albrecht 2013, S. 108, und K.-D. Schmitz 2008, S. 272). Zudem sind Kurzformen oder auch Akronyme gängig (vgl. Drewer und Ziegler 2014, S. 179, Albrecht 2013, S. 281, und Beste 2006, S. 35).

In Softwareprodukten kommt außerdem eine Vielzahl von Eigennamen vor, beispielsweise von Unternehmen, Produkten (vgl. Drewer und Ziegler 2014, S. 139), Programmiersprachen und Technologien. Diese zu erkennen und mit ihnen auf korrekte Weise umzugehen, ist ebenso Teil des fachlichen Hintergrundwissens und somit des fachlichen Kontexts.

In Abschnitt 5.5.2.1 sind Beispiele aus dem Korpus für diese Kategorie und die darin zu findenden Teilkategorien enthalten.

Aneinanderreihungen mit Fachterminologie Komposition ist eine typische Wortbildungsmöglichkeit von Fachterminologie (vgl. Beste 2006, S. 34, Drewer und Ziegler 2014, S. 177–178, und Albrecht 2013, S. 107).²⁹ Sie ist insbesondere im Englischen und im Deutschen weiter als in anderen Sprachen, beispielsweise den romanischen Sprachen, verbreitet (vgl. Albrecht 2013, S. 107). Beste gibt die Beispiele „*Gitternetzlinie, Onlinezusammenarbeit, AutoKorrektur, Textfeld, Formatvorlage, Webseitenvorschau, Tabstopp*“ (Beste 2006, S. 41, Hervorhebungen im Original) an, Irmeler und Hartwig nennen für das Englische „File

²⁸In diesem Zusammenhang zeigen sich viele Verfahren der Benennungsbildung, von denen u. a. Entlehnung für den IT-Bereich von großer Relevanz ist (vgl. beispielsweise Drewer und Ziegler 2014, S. 180). Die Bildungsmethoden für Benennungen spielen im Rahmen der vorliegenden Arbeit allerdings keine Rolle.

²⁹Diese und weitere Wortbildungsmöglichkeiten aus dem Bereich von Aneinanderreihungen zeigen sich bei GUI-Text sowohl bei Fachterminologie als auch außerhalb der Fachterminologie. Aus diesem Grund findet sich in Abschnitt 4.6.3 eine Kategorie für unklare Aneinanderreihungen für Fälle aus diesem Bereich, bei denen es sich nicht um Fachterminologie handelt.

Folder Synchronisation Wizard“ (Irmeler und Hartwig 2000, S. 95–96). Schmitz diskutiert die unklare Beziehung zwischen Bestandteilen eines fachterminologischen Kompositums, die zu Problemen beim Übersetzen führen kann, am Beispiel „data network identification code“ (vgl. K.-D. Schmitz 2007, S. 53) und Chang erläutert am Beispiel von „Internet Protocol“, welche Interpretationsmöglichkeiten dieses Kompositum bietet und wie die Disambiguierung mithilfe von Fachkenntnissen erfolgen kann (vgl. Chang 2005, S. 174–175).³⁰

Darüber hinaus sind auch Zusammensetzungen mit Eigennamen oder Abkürzungen möglich (für letzteres vgl. Beste 2006, S. 34). In Abschnitt 5.5.2.2 sind Teilkategorien dieser Kategorie zu finden, die im Korpus enthalten sind. Ergänzt werden diese durch Beispiele aus dem Korpus.

4.6.3 Funktionaler Kontext und Probleme bei der Übersetzung von GUI-Text

In diesem Abschnitt werden Kategorien von Übersetzungsproblemen beschrieben, für deren Lösung der funktionale Kontext (s. Abschnitt 4.4) hilfreich ist. Auch in diesem Bereich spielt wie bei den Problemen des situativen Kontexts (s. Abschnitt 4.6.1) die Nutzung von Englisch als Ausgangssprache der Mehrzahl der Softwareprodukte (vgl. Drewer und Ziegler 2014, S. 50, und Esselink 2003, S. 84) eine Rolle, was an den Beispielen in diesem Abschnitt deutlich wird.

Die Übersetzungsprobleme, für deren Lösung der funktionale Kontext hilfreich ist, lassen sich in die folgenden Kategorien einteilen:

Unklare Aneinanderreihung Insbesondere längere Aneinanderreihungen von Substantiven führen zu Problemen. Dunne bezeichnet dieses Phänomen als „noun stacking (i.e. modifier + noun + noun + noun, etc.)“ (K. J. Dunne 2009, S. 203). Die durch einen zusätzlichen „modifier“ ergänzte Variante wird nachfolgend besprochen, aber auch die Varianten lediglich aus Substantiven bringen Verständnis- und damit Übersetzungsprobleme mit sich (vgl. K. J. Dunne 2009, S. 203).

Ein Typ der Aneinanderreihung von Substantiven sind Komposita. Komposition ist nicht nur wie oben beschrieben eine typische Wortbildungsmöglichkeit von Fachterminologie (vgl. Beste 2006, S. 34, Drewer und Ziegler 2014, S. 177–178, und Albrecht 2013, S. 107), sondern wird auch zur Wortbildung außerhalb des Bereichs der Fachtermini genutzt. Bei

³⁰Weitere Informationen zu Unklarheiten bei Aneinanderreihungen finden sich in Abschnitt 4.6.3.

Komposita liegt ein Aspekt der Übersetzungsprobleme laut Albrecht darin, dass die Wortbildung durch Komposition einerseits zwar dazu dient, komplexe Sachverhalte knapp zu benennen, damit andererseits aber eine erhebliche semantische Vagheit einhergeht, da formal gleich gebildete Komposita auf unterschiedlichen semantisch-syntaktischen Relationen beruhen können (vgl. Albrecht 2013, S. 112). Als Beispiele nennt Albrecht „Schweineschnitzel, Jägerschnitzel, Kinderschnitzel“ (Albrecht 2013, S. 112). Auch Dunne nennt diese Ambiguität, daneben aber auch „hidden plurals“ wie im Beispiel „user data“, das als Daten über eine Benutzerin, Daten über die Benutzerin, Daten über Benutzerinnen allgemein und von der Benutzerin generierte Daten verstanden werden kann (vgl. K. J. Dunne 2009, S. 203).³¹ Was genau gemeint ist, kann ohne Rücksprache mit Klientin oder Entwicklerin nicht ermittelt werden (vgl. K. J. Dunne 2009, S. 203). Hierfür ist also der funktionale Kontext erforderlich. Ergänzend stellt Schmitz fest, dass sowohl in der englischen Sprache als auch im Bereich der Softwarelokalisierung besonders häufig „Mammutkomposita“ vorkommen, deren korrekte Auflösung „meist eine sehr genaue Kenntnis der Sachlage“ (K.-D. Schmitz 2002, S. 22) erfordert. Als Beispiele nennt er die auch schon von Wahle genannten Aneinanderreihungen von Substantiven „Network Printing Services Setup“ und „the Windows 98 operating system“ (K.-D. Schmitz 2002, S. 22)³², wobei ersteres ein Kompositum ist, letzteres aber ein Fall von Generalisierung-Präzisierung, was im Folgenden separat diskutiert wird. Melby und Foster nennen als Beispiele „alert rule settings“ und „event console purge policy“ (Anhang 1 in Melby und Foster 2010, S. 2).

Die von Schmitz genannte Sachkenntnis³³ kann je nach Kompositum Kenntnis der Funktionalität sein, bei Fachterminologie aber auch die Kenntnis des entsprechenden Fachgebiets, weshalb Fachkomposita separat betrachtet werden (s. Abschnitt 4.6.2).

Als weiteren Typ von unklaren Aneinanderreihungen nennt Wahle Komposita mit Adjektiv (also mit einem der von Dunne oben genannten „modifiers“), weil auf grammatischer Ebene nicht zu erkennen ist, welche Bestandteile zusammengehören, wodurch die Gefahr besteht, dass das Adjektiv dem falschen Substantiv zugeordnet wird (vgl. Wahle

³¹Eine umfassendere Diskussion zur Frage der Mehrdeutigkeit von Komposita und deren Disambiguierung im Deutschen findet sich bei Chang, der für die Disambiguierung auf den Kontext verweist (vgl. Chang 2005, S. 172–175).

³²Vgl. Wahle 2000a, S. 118

³³Ebenso wie Schmitz verweist auch Wahle auf Sachkenntnis und den gesunden Menschenverstand, um diese Art von Übersetzungsproblem zu lösen (vgl. Wahle 2000a, S. 116).

2000a, S. 116). Als Beispiel nennt sie „serial interface error“, das nicht mit „serieller Schnittstellenfehler“ zu übersetzen ist.³⁴ Esselink nennt in diesem Zusammenhang das Beispiel „required disk slot“ (vgl. Esselink 2000, S. 28)³⁵, bei Beste findet sich wie auch bei Wahle „unknown font error“ (vgl. Beste 2006, S. 73). Diese beiden letzten Beispiele weisen mit „required“ und „unknown“ allerdings keine Adjektive, sondern Partizipien auf, für die dieselben Übersetzungsprobleme gelten. Im Korpus wurden beide Varianten als separate Features annotiert (s. Abschnitt 5.5.3.1).

Hintergrund des Beispiels „serial interface error“ ist, dass sich ein vorangestelltes Adjektiv bei einem Kompositum im Deutschen immer auf das Grundwort bezieht. Im Englischen kann es sich dagegen auch wie hier auf das Bestimmungswort beziehen, es ist also die serielle Schnittstelle gemeint, bei der ein Fehler vorliegt. Dies kann im Deutschen nicht in Form eines Kompositums ausgedrückt werden, weil mit der Übersetzung „serieller Schnittstellenfehler“ ein serieller Fehler benannt wird, da „serieller“ sich auf das Grundwort, also „Fehler“ bezieht. Eine geeignete Übersetzungsvariante könnte also „Fehler bei der seriellen Schnittstelle“ sein.

Eine weitere Form von unklaren Aneinanderreihungen von Substantiven stellen im Englischen Fälle von Generalisierung-Präzisierung dar. Damit wird die Aneinanderreihung einer allgemeinen Information (Generalisierung) und einer damit verbundenen spezifischen Information (Präzisierung) bezeichnet. Im Englischen steht die Präzisierung häufig an erster Stelle und danach folgt erst die Generalisierung. Im Deutschen wird aber umgekehrt formuliert, z. B. wird „the ABC software“ als „die Software ABC“ übersetzt (vgl. Fischer 2011, S. 38).³⁶ Oberflächlich betrachtet sehen Komposita und Fälle von Generalisierung-Präzisierung identisch aus. Für die korrekte Übersetzung muss also die Beziehung zwischen den Bestandteilen der Aneinanderreihung bekannt sein. Diese bildet die Basis dafür, festzustellen, ob ein Kompositum oder ein Fall von Generalisierung-Präzisierung vorliegt.

³⁴Im Anschluss gibt Wahle noch weitere Beispiele.

³⁵Er bezeichnet dieses Phänomen als „long modifier chain“ (Esselink 2000, S. 28).

³⁶Schmitz sieht dies fälschlicherweise nur als stilistische Forderung, „im Deutschen zuerst die Art des Elements und dann das Element zu nennen statt Komposita zu verwenden (z.B. ‚das Menü Datei‘, statt ‚das Datei-Menü‘)“ (vgl. K.-D. Schmitz 2005a, S. 15). Ebenso ist sein von Wahle adaptiertes Beispiel „the Windows 98 operating system“ (K.-D. Schmitz 2002, S. 22) ein Fall von Generalisierung-Präzisierung, daher auch sein korrekter Übersetzungsvorschlag „das Betriebssystem Windows 98“.

Falls eine unklare Aneinanderreihung Fachterminologie enthält, spielt wie bereits festgestellt der fachliche Kontext auch eine wichtige Rolle beim Ermitteln der korrekten Übersetzung, Ottmann nennt aber ein Beispiel, an dem deutlich wird, warum der funktionale Kontext für die Übersetzung unklarer Aneinanderreihungen primäre Relevanz hat. Sie führt für den String „Approve Planning Alert“ die Übersetzungsvorschläge „Planungsmeldung genehmigen“ und „Meldung für Planungsgenehmigung“ an (vgl. Ottmann 2005, S. 106). Es ist aber noch eine dritte Übersetzung denkbar, „Meldung ‚Planung genehmigen‘“. Letzteres wäre ein Fall von Generalisierung-Präzisierung: Der Titel der Meldung, d. h. die Präzisierung, würde „Approve Planning“ lauten, die Art des Titels, also die Generalisierung, wäre entsprechend „Alert“. In diesem Fall wäre also denkbar, dass es verschiedene Meldungen mit unterschiedlichen Titeln gibt, wovon hier einer genannt wird. Möglich wären noch weitere Meldungen mit Titeln wie „Reject Planning“ oder „Cancel Planning“. Ottmanns zweiter Übersetzungsvorschlag wäre eine Paraphrase dieses Sachverhalts. Welche der Übersetzungen korrekt ist, lässt sich wesentlich auf Basis des funktionalen Kontexts beurteilen.

Denkbar ist, dass es bei den unklaren Aneinanderreihungen noch weitere Kategorien gibt. Abschnitt 5.5.3.1 enthält alle im Korpus identifizierten Kategorien der unklaren Aneinanderreihungen aus dem Korpus mit Beispielen aus dem Korpus.

Homonymie Laut Schmitz bezeichnet die Homonymie „eine Beziehung zwischen identischen Benennungen in derselben Sprache für unterschiedliche Begriffe“ (K.-D. Schmitz 2005b, S. 41; vgl. auch Beste 2006, S. 31).³⁷ Weiter weist er auf die Kommunikationsprobleme hin, die bei Nutzung von Homonymen innerhalb eines Fachgebiets entstehen (vgl. K.-D. Schmitz 2005b, S. 42). Laut Stolze sind Homonyme in „technischen Texten aufgrund der Terminologisierung gemeinsprachlicher Wörter sogar alles andere als selten [...] und müssen jeweils sachbezogen aufgelöst werden“ (Stolze 2013, S. 146). Explizit weist sie dabei darauf hin, dass die Homonyme „durch den Einbezug der Kontextsignale des Textganzen und des Wissenschaftshintergrunds“ (Stolze 2013, S. 147) eindeutig werden, also den in GUI-Text in der Regel nur unzureichenden sprach-

³⁷Dunne bezeichnet dasselbe Phänomen als „Homographs“ – seine Definition lautet „words with the same lexical form but different meanings“ (K. J. Dunne 2009, S. 202). Stolze weist darüber hinaus darauf hin, dass es neben Homonymie auch das Phänomen der Polysemie gibt, die nur schwer zu unterscheiden sind (vgl. Stolze 2013, S. 148). Schmitz zufolge ist die Unterscheidung dieser beiden Phänomene aber „in der Terminologielehre und praktischen Terminologearbeit nicht notwendig“ (K.-D. Schmitz 2005b, S. 41).

lichen, aber auch den fachlichen Kontext. Ebenso weist Albrecht in Bezug auf die korrekte Interpretation homophoner bzw. polysemer Wörter ganz allgemein u. a. auf den sprachlichen Kontext hin (vgl. Albrecht 2013, S. 217).

Dunne nennt als Beispiel den Einzelwortstring „Archive“, der das Speichersystem oder auch die Position der gespeicherten Elemente bezeichnen kann, und betont, dass solche Probleme ohne den genauen Kontext, in dem der String angezeigt wird, und die genaue Funktion, der er zugeordnet ist, nicht gelöst werden können (vgl. K. J. Dunne 2009, S. 202), also den situativen und den funktionalen Kontext.³⁸ An diesem Beispiel zeigt sich, dass Homonyme mit ihren verschiedenen Bedeutungen durchaus in einem einzigen GUI-Text vorkommen können. Wenn dem so ist, reicht der fachliche Kontext nicht aus, dann ist der funktionale Kontext für die eindeutige Klärung der Bedeutung im betreffenden String zwingend erforderlich. Ein ähnlicher Fall wird von Ottmann beschrieben: „Das englische ‚Copy‘ ist im Menü Edit mit ‚Kopieren‘ zu übersetzen, im Druckdialog dagegen mit ‚Exemplar‘.“ (Ottmann 2002, S. 149). Hier liegt zwar auch ein Wortartwechsel vor, gleichzeitig aber auch Homonymie an unterschiedlichen Stellen desselben Softwareprodukts. In Abschnitt 5.5.3.2 werden weitere Beispiele für diese Kategorie aus dem Korpus genannt.

Metaphorik Wie bereits im Zusammenhang mit dem fachlichen Kontext erwähnt, betreffen Softwareprodukte häufig „innovative Fachgebiete und Themen, bei denen die zu verwendende Terminologie in der Zielsprache (noch) nicht existiert oder etabliert ist“ (K.-D. Schmitz 2005a, S. 9), weil die Entwicklerinnen neue Terminologie für neue Funktionen und somit neue Begriffe und zugehörige neue Benennungen finden müssen (vgl. Drewer und Ziegler 2014, S. 177). Neben Komposition ist eines der dazu genutzten Verfahren die Terminologisierung, bei der eine Benennung „innerhalb einer Sprache in einen anderen Bereich [wandert], z.B. von der Gemeinsprache in eine Fachsprache oder von einer Fachsprache in eine andere“ (Drewer und Ziegler 2014, S. 178; s. auch Albrecht 2013, S. 280, und Arntz, Picht und K.-D. Schmitz 2014, S. 22). Nach Drewer und Ziegler stellt Metaphorik „den klassischen Fall der Terminologisierung [dar]: Ein einzelnes Wort wird einem sprachlichen Bereich

³⁸Er nennt allerdings auch explizit die verbale Form, „to file“ als weitere Möglichkeit, den Einwortstring „Archive“ zu verstehen, was den Hinweis auf den situativen Kontext als Verstehenshilfe erklärt. In der vorliegenden Arbeit wird dieses Problem aber an anderer Stelle, nämlich unter den ambigen Wortarten im situativen Kontext, kategorisiert (s. Abschnitt 4.6.1).

entnommen (meist der Gemeinsprache) und metaphorisch in die Fachsprache übertragen, um dort eine lexikalische Lücke zu füllen.“ (Drewer und Ziegler 2014, S. 140)³⁹ Abschnitt 5.5.3.3 enthält Beispiele für diese Kategorie aus dem Korpus.

Unklare Bedeutungen Insbesondere Nachrichten, aber auch andere Strings sind häufig sehr knapp gehalten und machen detaillierte Kenntnisse über das Produkt, aber teils auch die Programmierung oder den weiteren Kontext erforderlich (vgl. Esselink 2000, S. 63). Dunne bezeichnet dies als „Telegraphic Style“, der u. a. bei Fehler- und Statusnachrichten vorkomme und bei dem beispielsweise Relativpronomen und andere Wörter aus Gründen der Sprachökonomie oder zum Reduzieren der Übersetzungskosten weggelassen werden (vgl. K. J. Dunne 2009, S. 202). Bei dieser Problemkategorie ist weder der situative Kontext für das Verständnis hilfreich, noch ist die Fachterminologie der für das Verständnis relevanteste Aspekt. Stattdessen wird der funktionale Kontext benötigt: „[...] the meaning is likely to be lost on those who are not intimately familiar with the workings of the system“ (vgl. K. J. Dunne 2009, S. 202). Dunne nennt das Beispiel „Problem checking variables: {0}“ (vgl. K. J. Dunne 2009, S. 202), bei dem unklar sei, ob es sich um eine Status- oder Fehlernachricht handelt und ob die Anwendung gerade eine Problemprüfung der Variablen durchführt oder beim Prüfen der Variablen ein Problem aufgetreten ist (vgl. K. J. Dunne 2009, S. 202). Auch hier verweist Dunne zur Klärung auf das Entwicklungsteam (vgl. K. J. Dunne 2009, S. 203). In Abschnitt 5.5.3.4 finden sich Beispiele für diese Kategorie aus dem Korpus.

4.7 Einschränkungen und offene Fragen

Die Ausführungen in den vorangehenden Abschnitten könnten den Anschein erwecken, dass die Zuordnung zwischen den beschriebenen Problemen und den verschiedenen Kontextarten eindeutig ist. In der Praxis können die Probleme teils aber unter Nutzung verschiedener Kontextarten gelöst werden. Beispielsweise können ambige Wortarten an vielen Stellen auch über Informationen zur betreffenden Funktion disambiguiert werden. Die hier vorge-

³⁹Den Prozess der Metaphorisierung begründet Kemmann damit, dass Terminologie für Benutzerinnen intuitiv verständlich sein muss. Um dies zu erreichen, wird der Bezug auf Konzepte genutzt, die Benutzerinnen aus ihrem täglichen Leben vertraut sind. Als Beispiele nennt er den Erfolg von Office-Produkten, bei denen Bürotätigkeiten für die Metaphorisierung genutzt wurden, wie Ausschneiden und Einfügen, das Anlegen von Ordnern und Dateien, die Nutzung von Lineal, Stift und Taschenrechner (vgl. Kemmann 2002, S. 92).

nommene Einteilung kann allerdings dennoch dazu beitragen, die verschiedenen Problembereiche beim Übersetzen von GUI-Text klarer zu erkennen. Auf dieser Basis können dann die dafür jeweils erforderlichen Lösungswege, also das jeweils benötigte Referenzmaterial und mögliche Recherchewege, ermittelt werden.

Zudem muss einschränkend gesagt werden, dass mit diesen drei Kontextarten nicht alle Probleme, die bei der Übersetzung von Strings vorkommen, abgedeckt werden. Nur eine Auswahl typischer Probleme, für die der Kontext relevant ist, wird damit erfasst. Beispielsweise beschreibt Wahle anhand von möglichen Übersetzungen für den String „Close All Windows“ noch ein anderes Phänomen, bei dem sich der situative Kontext auf die Übersetzung auswirkt, das aber in der vorliegenden Untersuchung keine weitere Berücksichtigung findet (vgl. Wahle 2000b, S. 44; vgl. auch Wahle 2000a, S. 114, wo das analoge Beispiel „Close All Objects and Log Off“ genannt wird). Bei diesem Problem steht im Vordergrund, wie sich der situative Kontext darauf auswirkt, ob der entsprechende String im Imperativ („Schließen Sie alle Fenster“, falls er eine Aufforderung an den Benutzer ist), im Infinitiv („Alle Fenster schließen“, falls er die Beschriftung einer Schaltfläche oder ein Menüpunkt ist) oder in der dritten Person Singular („Schließt alle Fenster“, falls er ein Tooltip für ein Symbol ist) übersetzt werden soll (vgl. Wahle 2000b, S. 44).

Darüber hinaus gibt es bei der Übersetzung von Strings auch Probleme, für die der Kontext keine oder zumindest eine andere Rolle spielt, z. B. textliche Besonderheiten, wie die Verwendung von Variablen und Concatenations (s. Abschnitt 2.2; zum Problem der Übersetzung von Strings mit Variablen s. auch Reineke 2005, S. 86, und speziell zu Concatenations Behrens 2016, der diesem Phänomen eine ganze Monografie widmet).

Über die Einteilung der in der Literatur zur Softwarelokalisierung thematisierten Übersetzungsprobleme in verschiedene Kontextarten und darin in verschiedene Kategorien zeigt sich aber ein Ausgangspunkt für eine mögliche Operationalisierung. Offen bleibt aber, ob sich diese empirisch auch tatsächlich bestätigt. Dieser Schritt wird im nächsten Kapitel in Angriff genommen. Auf diese Weise lässt sich dann von der qualitativen Ebene auf die quantitative Ebene wechseln, es kann also die Frage beantwortet werden, wie häufig die verschiedenen beschriebenen Kategorien in einem Korpus mit Strings vorkommen. Die exemplarische Herangehensweise in den allermeisten der Veröffentlichungen zur Softwarelokalisierung lässt nämlich keine Rückschlüsse zur Relevanz der diskutierten Probleme zu, weshalb die Operationalisierung und auf dieser Basis die quantitative Auswertung dringend angebracht sind und auch Erkenntnisse erwarten lassen, die relevant für die weitere Forschungs- und Praxisarbeit in diesem Bereich sein können. Ebenso kann es von Nutzen

sein, den Bezug zwischen Kontextproblemen und Referenzmaterial näher zu untersuchen, um auf Basis aussagekräftiger Daten den in der Literatur häufig genannten Bedarf an verschiedenem Referenzmaterial für hochqualitative Lokalisierungen empirisch zu bestätigen. Wenn klar ist, dass tatsächlich verschiedene Arten von Referenzmaterial zur Lösung verschiedener Kontextprobleme nötig ist, kann diese Information dazu genutzt werden, bei Lokalisierungsprojekten auf möglichst umfassenden Zugang zu Referenzmaterialien zu drängen, um die verschiedenen Kontextprobleme möglichst vollständig lösen und damit höchstmögliche Lokalisierungsqualität erreichen zu können.

Kapitel 5

Korpuslinguistische Untersuchung von GUI-Text

Wenn festgestellt werden soll, ob die im vorangegangenen Kapitel herausgearbeitete Einteilung in die Kontextarten „Situativer Kontext“, „Fachlicher Kontext“ und „Funktionaler Kontext“ tatsächlich in GUI-Text vorkommt und welche Rolle sie dort spielt, werden für eine solche Untersuchung zunächst Daten und eine taugliche Methodik benötigt, die dazu dient, das Phänomen der Kontextarten operationalisierbar zu machen.

Zu diesem Zweck werden in diesem Kapitel zunächst Informationen zur gewählten Methodik, einer korpuslinguistischen Untersuchung (Abschnitte 5.1 und 5.2), und zu den verwendeten Textproben (Abschnitt 5.3) gegeben. Daran schließen sich Details zur Vorbereitung des Korpus (Abschnitt 5.4) sowie zum Annotationsschema (Abschnitt 5.5) und zu den Ergebnissen (Abschnitte 5.6 und 5.7) an. Zuletzt werden einige der Einschränkungen dieser Vorgehensweise, die bereits am Ende des letzten Kapitels genannt wurden, anhand einiger Beispiele konkretisiert (Abschnitt 5.8). Dies alles bildet die Grundlage für die weiteren Kapitel der vorliegenden Arbeit.

5.1 Korpuslinguistik als Methodik

Wenn die Strings einer Software untersucht werden sollen, entspricht dies einer produktorientierten Herangehensweise. Eine der gängigen produktorientierten Untersuchungsmethoden ist die **Korpuslinguistik**, die laut Lemnitzer und Zinsmeister die „Beschreibung von Äußerungen natürlicher Sprachen, ihrer Elemente und Strukturen, und die darauf aufbauende Theoriebildung auf der Grundlage von Analysen authentischer Texte, die in Korpora zusammengefasst sind“ (Lemnitzer und Zinsmeister 2015, S. 14), bezeichnet.

Korpuslinguistik kommt u. a. für deskriptive Zwecke zum Einsatz (vgl. Saldanha und O'Brien 2013, S. 50). Ihr Ziel ist die Beschreibung von Sprache, die als empirische Untersuchung des tatsächlichen Sprachgebrauchs erfolgt (vgl. Scherer 2014, S. 1). Korpuslinguistische Studien weisen quantitative und qualitative Aspekte auf, wobei über die quantitative Auswertung empirisch die Relevanz beobachteter Phänomene aufgezeigt werden kann, während die qualitative Herangehensweise zur Aufbereitung und Interpretation der Daten sowie zur Datenexploration und zur Gewinnung erster Arbeitshypothesen dienen kann (vgl. Hirschmann 2019, S. 6–7).

Im Rahmen einer deskriptiven korpuslinguistischen Untersuchung wird ein **Korpus** erstellt, das Bowker und Pearson als eine große Sammlung authentischer Texte verstehen, die auf Basis eines bestimmten Satzes von Kriterien in elektronischer Form gesammelt wurden (vgl. Bowker und Pearson 2002, S. 9). Lemnitzer und Zinsmeister konkretisieren diese Definition wie folgt:

Ein Korpus ist eine Sammlung schriftlicher oder gesprochener Äußerungen. Die Daten des Korpus sind typischerweise digitalisiert, d.h. auf Rechnern gespeichert und maschinenlesbar. Die Bestandteile des Korpus bestehen aus den Daten selber sowie möglicherweise aus Metadaten, die diese Daten beschreiben, und aus linguistischen Annotationen, die diesen Daten zugeordnet sind. (Lemnitzer und Zinsmeister 2015, S. 13)

Die Daten selber, also die im Korpus enthaltenen Texte, werden als **Primärdaten** bezeichnet und stellen die eigentlichen Texte ohne die nicht sprachlich motivierten Eigenschaften, wie Worttrennung oder Schrifttyp, dar (vgl. Lemnitzer und Zinsmeister 2015, S. 43). Die Texte können dabei neben schriftlichen Dokumenten aus mündlichen Äußerungen bestehen (vgl. Scherer 2014, S. 3). Die **Metadaten** enthalten weitere Informationen über die Texte im Korpus, wie Medium, Autorinnen und Zeitpunkt der Entstehung (vgl. Lemnitzer und Zinsmeister 2015, S. 44).

Neben dem authentischen Sprachmaterial sind in Korpora Informationen zu Frequenzen sowie beispielsweise dazu enthalten, wie Wörter und grammatische Kategorien verwendet werden (vgl. Scherer 2014, S. 10). Ist ein Korpus erst einmal erstellt, bleibt es in aller Regel unverändert, es erfüllt somit „das Kriterium der **Beständigkeit**“ (vgl. Scherer 2014, S. 6, Hervorhebung im Original).

Mithilfe **linguistischer Annotationen** werden linguistische Einheiten „mit ihren linguistischen Beschreibungen verbunden“ (Lemnitzer und Zinsmeister 2015, S. 43; vgl. auch Saldanha und O'Brien 2013, S. 77, und Hirschmann 2019, S. 22). Der Vorteil der Annotation besteht laut Scherer darin,

dass sie „implizite Informationen explizit macht und somit eine einfachere und schnellere Erhebung der benötigten Informationen ermöglicht“ (Scherer 2014, S. 21). Die linguistische Annotation kann dabei „auf Wort-, Satz-, Text-, Laut- oder Bedeutungsebene eingefügt werden“ (Scherer 2014, S. 21). Mithilfe von Annotationen können auf effiziente Weise geeignete Beispiele gefunden werden und sie erhöhen gleichzeitig die Überprüfbarkeit durch andere Forschende (vgl. Lemnitzer und Zinsmeister 2015, S. 57).

Die Summe der Annotationen bildet das **Annotationsschema**, „die systematische Beschreibung von Annotationskategorien und ihre Anwendung auf Korpusdaten“ (Lemnitzer und Zinsmeister 2015, S. 196). Für ein solches müssen die Kategorien sehr gut definiert werden, damit sie „nachvollziehbar und objektiv definiert sind“ (Lemnitzer und Zinsmeister 2015, S. 102). Das Annotationsschema sollte daher neben den Tagnamen und den Kategoriebezeichnungen u. a. auch Definitionen der Kategorien sowie prototypische und problematische Beispiele enthalten, Entscheidungsbäume sind ebenfalls hilfreich (vgl. Lemnitzer und Zinsmeister 2015, S. 102). Das Annotationsschema selbst sollte laut Lemnitzer und Zinsmeister durch und mit der Arbeit an den Korpusdaten entwickelt werden (vgl. Lemnitzer und Zinsmeister 2015, S. 103).

Einschränkend weist Hirschmann aber darauf hin, dass „die Anreicherung durch Annotationen [...] immer durch eine streitbare und fehleranfällige Interpretation der zu annotierenden Daten“ (Hirschmann 2019, S. 22) erfolgt. Mithilfe eines Inter-Annotator-Agreement (IAA) kann allerdings die Tauglichkeit der für die Annotationen festgelegten Kategorien überprüft werden (vgl. Hirschmann 2019, S. 97).

Die Annotation selbst kann mithilfe von Annotationstools erfolgen, durch die die Annotation vereinfacht wird und die den Datenexport in standardisierten Ausgabedateien, beispielsweise im XML-Format, ermöglichen (vgl. Lemnitzer und Zinsmeister 2015, S. 105). Ein solches Annotationstool ist UAM CorpusTool¹, das in Version 3.3 zur Annotation genutzt wurde (s. Abschnitt 5.4).

5.2 Kriterien für die Korpuserstellung

Im Folgenden werden einige der Kriterien erläutert, die bei der Erstellung von Korpora berücksichtigt werden müssen.

Repräsentativität und Ausgewogenheit

¹<http://www.corpustool.com/>; letzter Zugriff: 17.06.2025

Scherer weist in ihrer Korpusdefinition explizit darauf hin, dass die Auswahl der Texte bzw. Textteile „bewusst nach bestimmten sprachwissenschaftlichen Kriterien“ (vgl. Scherer 2014, S. 3) erfolgen soll. Das Korpus dient als zu untersuchender Ausschnitt der Sprache, was die Sprache ganz allgemein oder eine bestimmte Varietät sein kann (vgl. Scherer 2014, S. 4). Wird mit einem Korpus, einem sogenannten **Spezialkorpus**, eine Sprachvarietät untersucht, muss angestrebt werden, dass es repräsentativ für diese Varietät ist (vgl. Saldanha und O’Brien 2013, S. 59, 78; vgl. auch Scherer 2014, S. 28). Solche Spezialkorpora können für spezielle Forschungsfragen genutzt werden (vgl. Saldanha und O’Brien 2013, S. 78).

Die Repräsentativität ist bei der Erstellung eines Korpus das wichtigste Ziel. Dazu muss allerdings die zu untersuchende Grundgesamtheit festgelegt werden (vgl. Scherer 2014, S. 5), also welche Texte die Sprachvarietät ausmachen. Das Korpus stellt dabei die Stichprobe dar, die die Grundgesamtheit repräsentiert und die so Aussagen über die Grundgesamtheit ermöglicht, wobei letztere beispielsweise bei der deutschen Sprache der Gegenwart nicht präzise definiert werden kann (vgl. Lemnitzer und Zinsmeister 2015, S. 48). Zudem kann laut Saldanha und O’Brien die Repräsentativität auch deshalb in der Praxis nicht immer erreicht werden, da sich häufig äußere Umstände auf die Korpuserstellung auswirken (vgl. Saldanha und O’Brien 2013, S. 76). Lemnitzer und Zinsmeister sprechen in diesem Kontext von „*opportunistischen* Datenzusammenstellungen“ (Lemnitzer und Zinsmeister 2015, S. 141, Hervorhebung im Original) als Extremfall dieser Problematik. Dieser Fall trifft beispielsweise zu, wenn Daten nur dann Teil des Korpus sind, weil dafür keine Copyright-Beschränkungen bestehen und sie verfügbar sind.

Ein weiterer Aspekt in diesem Zusammenhang ist die Ausgewogenheit: In der Regel wird Heterogenität der Textproben bevorzugt, da ein ausgewogenes Korpus „der Heterogenität einer Sprache gerecht wird“ (Lemnitzer und Zinsmeister 2015, S. 141). Es muss also dafür gesorgt werden, dass die Textproben, die in ein Korpus aufgenommen werden, auch die Breite der sprachlichen Phänomene abdecken, die untersucht werden. Häufig kann dies durch Aufnahme verschiedener Textsorten in ein Korpus erreicht werden, wobei externe und interne Kriterien berücksichtigt werden (vgl. Lemnitzer und Zinsmeister 2015, S. 50). Die externen Kriterien beziehen sich dabei auf das quantitative Verhältnis verschiedener Textsorten zueinander, das im Korpus berücksichtigt wird, die internen Kriterien auf die zu beobachtenden linguistischen Phänomene, die für eine Textsorte charakteristisch sind (vgl. Lemnitzer und Zinsmeister 2015, S. 50).

Um Verlässlichkeit und Replizierbarkeit der Ergebnisse einer Studie sicherzustellen, betonen Saldanha und O’Brien daher, wie wichtig es ist, genau zu beschreiben, wie die Stichprobe ausgewählt wurde, und die Details

der Korpusinhalte detailliert zu veröffentlichen sowie alle Entscheidungen zu dokumentieren, damit für die Lesenden nachvollziehbar bleibt, was wie gemacht wurde und wie die Schlüsse daraus gezogen wurden (vgl. Saldanha und O'Brien 2013, S. 76). Lemnitzer und Zinsmeister weisen ergänzend darauf hin, dass insbesondere quantitative, aber auch qualitative Erkenntnisse aus einem Korpus nur sehr eingeschränkt auf die Grundgesamtheit übertragen werden können oder zunächst nur für das Korpus selbst gelten und durch die weitere Forschung auch widerlegt werden können (vgl. Lemnitzer und Zinsmeister 2015, S. 49–51).

Texte und Textgröße

Wenn große Datenvolumina untersucht werden, können damit Diskurse in einer Gemeinschaft untersucht werden, nicht nur die der Autorin eines Texts oder einer kleinen Gruppe von Sprachnutzenden (vgl. Saldanha und O'Brien 2013, S. 59). Laut Bowker und Pearson ist ein erster Indikator für die Angabe *groß*, dass die Anzahl der Texte größer ist, als sie einfach in gedruckter Form gesammelt und gelesen werden könnten (Bowker und Pearson 2002, S. 10). Die Größe muss aber stets in Relation dazu betrachtet werden, welche Sprachvarietät repräsentiert werden soll, weshalb genaue Angaben dazu schwierig sind (vgl. Saldanha und O'Brien 2013, S. 73). Gemessen wird die Größe laut Scherer in Wörtern. Sie weist darauf hin, dass die häufig geäußerte Meinung, dass größere Korpora besser wären, nicht zutreffe. Zehn- bis zwanzigtausend Wörter könnten je nach Untersuchungsgegenstand schon relevante Ergebnisse erbringen (vgl. Scherer 2014, S. 6–7).

Eine einfache Lösung scheint in diesem Zusammenhang der Rückgriff auf die im Internet in großer Zahl vorhandenen Texte zu sein. Diese Vorgehensweise wurde zunächst unter dem Stichwort *Web as Corpus* bekannt (vgl. Kilgarriff und Grefenstette 2003, S. 333), weist aber Probleme im Hinblick auf das Urheberrecht auf (vgl. Perkuhn, Keibel und Kupietz 2012, S. 46). Ein aktuelleres Stichwort mit ähnlichem Bezug und ähnlichen Problemen ist *Big Data* (vgl. Calderón Campos und Vaamonde 2024, S. 1). Ob das Internet oder Big Data für ein Forschungsvorhaben geeignet sind, muss allerdings einerseits in Bezug auf Repräsentativität und Ausgewogenheit und andererseits in Bezug auf die Forschungsfrage diskutiert werden: Kleinere Korpora werden wie oben dargestellt in der Regel für spezielle Forschungsfragen erstellt und die Annotation kann detaillierter erfolgen, allerdings ist die Repräsentativität möglicherweise eingeschränkt, wodurch ihr Wert als Ausgangspunkt für allgemeingültige Aussagen geringer wird. Größere Referenzkorpora weisen größere Repräsentativität für eine Sprache oder Sprachvarietät auf und sind für eine größere Bandbreite an Forschungsprojekten nützlich, für spezielle Fragen enthalten sie aber unter Umständen keine ausreichenden Daten.

Korpora, bei denen das Internet als Korpus oder Big Data genutzt werden, sind nützlich für die quantitative Analyse in großen Dimensionen, weisen aber häufig Probleme in Bezug auf Ausgewogenheit und Repräsentativität auf und die Annotation ist in vielen Fällen fehlerhaft. (vgl. Calderón Campos und Vaamonde 2024, S. 1)²

Auch die Frage, ob ganze Texte oder nur Textausschnitte, als Proben bzw. **Textproben** (vgl. Scherer 2014, S. 7) bezeichnet, in das Korpus aufgenommen werden sollen, muss in Betracht gezogen werden. Scherer spricht von zwei Varianten, der Aufnahme der gesamten Texte oder der Aufnahme einer vorher festgelegten Anzahl von Wörtern (vgl. Scherer 2014, S. 7). Im letzteren Fall müssen zu lange Texte gekürzt werden, zu kurze müssen mit anderen zusammengefasst werden (vgl. Scherer 2014, S. 20). Die Verwendung von Textausschnitten mit gleicher Anzahl an Wörtern vereinfacht den statistischen Vergleich, allerdings wird dadurch auch das Prinzip der Datenintegrität verletzt (vgl. Saldanha und O'Brien 2013, S. 74).

Biber weist ergänzend darauf hin, dass für die Beantwortung einer bestimmten Forschungsfrage die Auswahl der Textproben relevant ist, da Textteile jeweils verschiedene Zwecke, Themen und teils auch Adressaten haben. Werden Textproben ohne Rücksicht darauf aus längeren Texten entnommen, ermögliche dies die Charakterisierung des Gesamttexts oder Registers. Wenn jedoch Stichproben entnommen werden, die homogen im Hinblick auf Zweck, Thema usw. sind, beispielsweise nur Methodikabschnitte in Forschungsartikeln, ermögliche dies die linguistische Analyse der speziellen Charakteristika bestimmter Subregister. (vgl. Biber 1995, S. 131–132)

Biber argumentiert mit Verweis auf frühere Arbeiten weiter, dass die Repräsentativität eines Korpus sowohl im Hinblick auf Größe als auch im Hinblick auf Diversität betrachtet werden müsse, wobei die Diversität nach seiner Meinung die größere Bedeutung hat. Die Frequenz vieler verbreiteter linguistischer Phänomene lässt sich nach seinen Erkenntnissen schon anhand kurzer Textproben mit einer Größe von je 1.000 Wörtern und mit insgesamt zehn verschiedenen Texten zuverlässig ermitteln (vgl. Biber 1995, S. 131; s. auch Biber 1993, S. 249), wodurch die benötigte Diversität entsteht. Dagegen kann die Frequenz seltenerer Phänomene weniger zuverlässig anhand kurzer Textproben festgestellt werden (vgl. Biber 1993, S. 249).

Copyright

Saldanha und O'Brien stellen darüber hinaus die Frage des Copyrights der im Korpus verwendeten Textproben heraus. Festzustellen, wer der Copyright-

²Weitere relevante Aspekte für die Nutzung des Internets als Korpus finden sich beispielsweise bei Lemnitzer und Zinsmeister (vgl. Lemnitzer und Zinsmeister 2015, S. 42).

Inhaber ist und welche Vorgaben beispielsweise bei Freigabe des Zugriffs auf ein elektronisches Korpus über das Internet für Texte in verschiedenen Sprachen und aus verschiedenen Ländern bestehen, ist sehr schwierig und zeitaufwändig (vgl. Saldanha und O’Brien 2013, S. 75–76; s. auch Lemnitzer und Zinsmeister 2015, S. 141).

5.3 Auswahl der Textproben

Um Forschungsfragen zur Textsorte GUI-Text (s. Abschnitt 4.2) zu untersuchen, ist ein Spezialkorpus nötig (vgl. Saldanha und O’Brien 2013, S. 78), das für diesen Zweck erstellt werden muss, da hierfür keine allgemein zugänglichen Korpora verfügbar sind. Das Internet als Korpus oder auch Big Data (vgl. Kilgarriff und Grefenstette 2003, S. 333, und Calderón Campos und Vaamonde 2024, S. 1) kommen für diesen Bereich nicht in Frage, weil nur diese eine sehr spezielle Textsorte im Korpus enthalten sein soll.

Auf Basis von Bibers Argumentation in vorangehenden Abschnitt wurden daher zehn Textproben mit Strings aus zehn verschiedenen Softwareprodukten mit jeweils ca. 1.000 Wörtern in das Korpus aufgenommen (vgl. Biber 1995, S. 131). Für die Auswahl der Softwareprodukte wurden folgende Kriterien berücksichtigt:

- Gemäß der Definition in Abschnitt 2.1.1 sollten Softwareprodukte aus den Bereichen Anwendungs- und Unterstützungssoftware für das Korpus berücksichtigt werden, um sowohl die Diversität (vgl. Biber 1995, S. 131) als auch die Repräsentativität (vgl. Saldanha und O’Brien 2013, S. 59) und die Ausgewogenheit (vgl. Lemnitzer und Zinsmeister 2015, S. 50) zu erhöhen.
- Diese Softwareprodukte sollten ebenfalls aus Gründen von Diversität, Repräsentativität und Ausgewogenheit zehn unterschiedliche Anwendungsgebiete abdecken.
- Alle Softwareprodukte sollten über eine freie bzw. eine Open-Source-Lizenz lizenziert sein, einerseits um Copyright-Probleme zu vermeiden (vgl. Saldanha und O’Brien 2013, S. 75–76), andererseits, weil in solchen Projekten der Quellcode der Softwareprodukte und damit auch die Strings verfügbar sind (s. Abschnitt 2.3 und vgl. Stallman 2002, S. 20, 57 und Jiménez-Crespo 2024, S. 283).
- Wie in Abschnitt 2.1.4 festgestellt, können Strings in Softwarelokalisierungsprojekten als Teil des Quellcodes, in den Binärdateien des lauffähigen Programms, von Programmierinnen in Tabellen extrahiert und ausgelagert in Ressourcendateien zur Übersetzung bereitgestellt

werden (vgl. beispielsweise Ottmann 2005, S. 103, Drewer und Ziegler 2014, S. 44, und Reineke 2005, S. 74). Die Strings, die in das Korpus aufgenommen wurden, sollten in Ressourcendateien ausgelagert sein, weil sie dadurch gut in das Korpus aufgenommen werden konnten.

- Die Softwareprodukte sollten in englischer Sprache entwickelt worden sein, weil dies die zentrale Sprache für die Softwareentwicklung ist (vgl. Drewer und Ziegler 2014, S. 50, und Esselink 2003, S. 84). Da bei Open-Source-Projekten allerdings Freiwillige ohne Einschränkungen mitarbeiten, ist mit Ausnahme von sehr kleinen Projekten davon auszugehen, dass Englisch nicht die Muttersprache aller Beteiligten ist und dass die Strings daher auch nicht oder nicht nur von Muttersprachlerinnen geschrieben werden.³
- Es sollte eine deutsche Übersetzung der Strings vorliegen, einerseits aus praktischen Gründen, nämlich als Hinweis, dass der Zugriff auf die Strings technisch möglich ist, andererseits aber auch als Hinweis darauf, dass für das Softwareprodukt Lokalisierungsbedarf ins Deutsche besteht.
- Die Softwareprodukte sollten möglichst Anwendungsgebiete abdecken, die für ein breiteres Publikum von Nutzen sind, was ebenfalls ein Grund für einen höheren Lokalisierungsbedarf ist. Aufgrund der eher technischen Natur der Unterstützungsprogramme trifft dies für diese Programmgruppe aber weniger zu, weshalb aus dieser Kategorie auch weniger Programme in das Korpus aufgenommen wurden. Es ist anzunehmen, dass Unterstützungsprogramme aufgrund ihrer Einsatzgebiete und Verbreitung einen geringeren Anteil an der Grundgesamtheit der Computerprogramme für ein breiteres Publikum bilden, weshalb ein deutlich geringerer Anteil am Korpus der Repräsentativität sowie der Ausgewogenheit dient (vgl. dazu beispielsweise Saldanha und O'Brien 2013, S. 59, und Lemnitzer und Zinsmeister 2015, S. 50).
- Die Softwareprodukte im Korpus sollten für alle drei Desktop-Betriebssysteme Windows, Linux und Mac OS verfügbar sein, da Diversität, Repräsentativität und Ausgewogenheit auch dadurch erhöht werden, dass es keine Konzentration auf ein Betriebssystem gibt.
- Es kann vorkommen, dass Open-Source-Projekte aufgegeben und nicht mehr weiterentwickelt werden. Dann ist zwar häufig das Softwareprodukt in seiner letzten Version noch verfügbar, es werden aber keine neuen Versionen mehr veröffentlicht, beispielsweise weil die zentralen

³Eine ähnliche Problematik sehen beispielsweise auch Lemnitzer und Zinsmeister in Bezug auf die Nutzung von Texten aus dem Internet, bei denen ebenfalls vorkommen kann, dass sie nicht von Muttersprachlerinnen verfasst wurden (vgl. Lemnitzer und Zinsmeister 2015, S. 42).

Organisatorinnen das Interesse verloren oder keine Zeit mehr haben, gewisse Ziele erreicht wurden oder ähnliche Ziele bereits bei anderen Projekten verwirklicht wurden. Für dieses Korpus wurden daher speziell Projekte gesucht, für die es zum Zeitpunkt der Korpuserstellung in jüngerer Vergangenheit noch neue Versionen bzw. Releases gab, die sich also mit großer Wahrscheinlichkeit noch in Entwicklung befanden.⁴

- Wenn im Rahmen eines Open-Source-Projekts, wie beispielsweise Mozilla, mehrere Softwareprodukte mit unterschiedlichen Anwendungsgebieten programmiert werden, sollte nur ein einziges dieser Programme in das Korpus aufgenommen werden, um Diversität, Repräsentativität und Ausgewogenheit zu erhöhen.

5.3.1 Die Softwareprodukte im Korpus

Im Folgenden finden sich zwei Listen mit den verwendeten Softwareprodukten mit Versionen sowie den dabei genutzten Lizenzen (s. Abschnitt 2.3). Weitere Details zur Vorbereitung des Korpus sowie zu den verwendeten Softwareprodukten, wie Downloadlinks, Informationen zur Extraktion der Texte für das Korpus und dabei aufgetretenen Problemen, finden sich in Anhang A. Zunächst folgt eine Liste mit den Anwendungsprogrammen:

- Audiotbearbeitung: Audacity Version 2.2.2 (GNU GPL Version 2)⁵
- Browser: Mozilla Firefox Version 66.0.3 (MPL Version 2.0)
- Bildbearbeitung: GIMP Version 2.10.8 (GNU GPL Version 3+)
- Dateimanager: Double Commander Version 0.9.2 (GNU GPL Version 2)
- E-Mail-Programm: Claws Mail Version 3.17.3 (GNU GPL Version 3)
- Medienplayer: VLC Media Player Version 3.0.6 (GNU LGPL Version 2.1+)
- Office-Anwendung: LibreOffice Version 6.2.2.2 (MPL Version 2.0)
- Videobearbeitung: Avidemux Version 2.7.3 (GNU GPL)

Hinzu kamen folgende Unterstützungsprogramme:

- Editor: Notepad++ Version 7.6.4 (64 Bit) (GNU GPL Version 2)

⁴Die Korpuserstellung erfolgte 2019, die Entwicklung des HTML-Editors BlueGriffon, dessen GUI-Strings Teil des Korpus waren, wurde im Jahr 2024 eingestellt (vgl. <https://de.wikipedia.org/wiki/BlueGriffon>; letzter Zugriff: 05.07.2025).

⁵In den Klammern wird in dieser Liste jeweils die genutzte Lizenz angegeben. Informationen zu GNU GPL: https://de.wikipedia.org/wiki/GNU_General_Public_License; letzter Zugriff: 20.06.2025. Informationen zu MPL: https://de.wikipedia.org/wiki/Mozilla_Public_License; letzter Zugriff: 20.06.2025.

- HTML-Editor: BlueGriffon Version 3.0.1 (MPL Version 2.0)

Die im vorangehenden Abschnitt genannten Kriterien konnten damit bei der Korpuserstellung weitgehend umgesetzt werden. Allerdings hatte die Kombination der Kriterien in verschiedenen Fällen, wie im Zusammenhang mit der Repräsentativität oben (s. Abschnitt 5.2) erwähnt, einen großen Einfluss auf die tatsächlich in das Korpus aufgenommenen Textproben, wenn beispielsweise interessante Korpuskandidaten nicht ins Deutsche lokalisiert waren oder die Entwicklung eines Projekts anscheinend zum Stillstand gekommen war (vgl. dazu beispielsweise Lemnitzer und Zinsmeister 2015, S. 141 und die von ihnen angeführten opportunistischen Datenzusammenstellungen). Die auffälligste Auswirkung der Kriterien auf die Textproben besteht darin, dass in manchen Fällen anstelle von bekannteren Softwareprodukten weniger bekannte Alternativen in das Korpus aufgenommen werden konnten. Ein Beispiel dafür stellt das E-Mail-Programm Claws Mail dar, das anstelle verbreiteter Softwareprodukte aus diesem Bereich wie Thunderbird in das Korpus aufgenommen wurde, weil Thunderbird wie schon der im Korpus verwendete Browser Firefox ein Mozilla-Projekt ist. Die einzige Abweichung von den Kriterien bildet der Editor Notepad++, der lediglich unter Windows verfügbar ist, nicht aber unter Linux und Mac, aufgrund seiner Verbreitung und der langjährigen kontinuierlichen Entwicklung dennoch in das Korpus aufgenommen wurde.

5.3.2 Volumen der Softwareprodukte im Korpus

Tabelle 5.1 enthält einen Überblick über die AT-Volumen der im Korpus enthaltenen Strings der einzelnen Softwareprodukte.

Das Volumen wurde jeweils in Microsoft Word gezählt, wobei Bibers Vorgaben beachtet wurden (vgl. Biber 1995, S. 131, s. auch Abschnitt 5.3).⁶ Da zusammenhängende Zeichenfolgen in Word immer als ein Wort gezählt werden, mussten teils vor oder nach Steuerzeichen Leerzeichen eingefügt oder gelöscht werden, um eine korrekte Zählung zu ermöglichen. Diese Einfügungen wurden aber nicht in das Korpus übernommen. Ein Beispiel:

⁶Die Gesamtzahl an Ressourcendateien sowie das Gesamtvolumen an Wörtern bzw. Strings war bei diesen Softwareprodukten allerdings sehr unterschiedlich. Anteilig am Gesamtvolumen wurden also sehr unterschiedliche Prozentsätze der Strings eines Softwareprodukts in das Korpus aufgenommen. Dadurch waren aber auch die im Korpus berücksichtigten Inhalte sehr unterschiedlich, was sich positiv auf die Diversität (vgl. Biber 1995, S. 131) und damit auch auf die Repräsentativität (vgl. Saldanha und O'Brien 2013, S. 59) und die Ausgewogenheit (vgl. Lemnitzer und Zinsmeister 2015, S. 50) auswirkte.

Tabelle 5.1: Softwareprodukte und Volumen

Name der Software	Volumen in AT-Wörtern	Volumen in Strings
Audacity	1.001 Wörter	213 Strings
Avidemux	1.002 Wörter	363 Strings
BlueGriffon	1.001 Wörter	424 Strings
Claws Mail	1.001 Wörter	233 Strings
Double Commander	1.004 Wörter	381 Strings
Gimp	1.001 Wörter	84 Strings
LibreOffice	1.000 Wörter	470 Strings
Mozilla Firefox	1.004 Wörter	315 Strings
Notepad++	1.005 Wörter	387 Strings
VLC Media Player	1.009 Wörter	137 Strings
Gesamt	10.028 Wörter	3.007 Strings

- Audacity could not find a place to store temporary files.\nPlease enter an appropriate directory in the preferences dialog.

Die Stelle mit dem Zeilenumbruch ([...] files.\nPlease [...]) wurde von Microsoft Word als ein Wort gezählt, also wurde zum Zählen ein Leerzeichen eingefügt, wodurch dann die tatsächlich vorhanden zwei Wörter gezählt wurden: [...] files.\n Please [...].

Außerdem wurden Strings, die in den Ressourcendateien so formatiert waren, dass sie mehrere Zeilen umfassten, für das Korpus in eine einzige Zeile eingefügt. Die Zahl der Zeilen im Korpus ist also identisch mit der Zahl der Strings.

Je mehr Strings eine Textprobe umfasst, desto weniger Wörter sind im Schnitt pro Zeile enthalten, desto kürzer sind also die Strings. Enthält die Textprobe umgekehrt mehr längere Strings, nimmt dadurch die Zahl der Zeilen ab. Der Median der Strings für das gesamte Korpus liegt bei 339, die Standardabweichung beträgt 128,2, der Interquartilsabstand beträgt 167,5.

5.4 Vorbereitung des Korpus

Wenn die Strings eines Softwareprodukts in nur einer einzigen Datei enthalten waren, wurden die ersten ca. 1.000 Wörter dieser Datei für das Korpus verwendet. In Fällen, in denen die Strings in mehreren Dateien enthalten waren, wurde eine Datei verwendet, die Strings aus zentralen Bereichen der

GUI enthielt. Falls in einer Datei weniger als 1.000 Wörter enthalten waren, wurden Inhalte weiterer Dateien hinzugezogen. In allen Fällen wurden die String-IDs nicht gezählt und vor Aufnahme in das Korpus entfernt. Im Unterschied zu den String-IDs wurden etwaige Steuerzeichen und beispielsweise HTML-Tags und ähnliches, die Teil der Strings waren, mit in das Korpus aufgenommen. Die gewählten Strings wurden für jedes Programm in eine Textdatei eingefügt, die in UAM CorpusTool Version 3.3⁷ eingelesen wurde. Falls Strings aus mehr als einer Datei genutzt wurden, wurden sie alle fortlaufend in diese Datei kopiert.

Sämtliche Details zu Vorgehensweise und Problemen bei den einzelnen Textproben finden sich in Anhang A.

5.5 Annotationsschema und Features

Wie bereits in Abschnitt 5.1 erwähnt, sind bei korpuslinguistischen Studien quantitative und qualitative Aspekte relevant (vgl. Hirschmann 2019, S. 6–7). Beide spielen auch bei der vorliegenden Untersuchung eine Rolle. In einem ersten Schritt wurden die Korpusdaten im Hinblick auf die im vorangehenden Kapitel beschriebenen Kontextarten (s. Abschnitt 4.4) explorativ untersucht. Parallel wurde ein detailliertes Annotationsschema erstellt. Anschließend wurde eine Annotation des gesamten Korpus vorgenommen, die die Basis für eine nachfolgende quantitative Auswertung bildet (s. Abschnitt 5.6). Mit vorab formulierten Hypothesen wurde in diesem Fall also nicht gearbeitet. Im Einzelnen wurden folgende Kontextarten identifiziert, die im weiteren Verlauf als Basis für die Annotation des Korpus dienen:

Situativer Kontext: Diese Kontextart betrifft die Multimodalität von GUIs (s. Abschnitt 4.1). Einzelne Strings in der GUI haben als Kontext neben den sie umgebenden Strings, die als sprachlicher Kontext dienen können, auch Kontext, der aus den verwendeten grafischen und typografischen Gestaltungsmitteln (vgl. beispielsweise Wahle 2000b, S. 35, Beste 2006, S. 61–62, und Behrens 2016, S. 52–53) sowie der hierarchischen Anordnung der Strings (vgl. Beste 2006, S. 41) und den verwendeten GUI-Elementen (vgl. Beste 2006, S. 62–63) hervorgeht.

Fachlicher Kontext: Diese Kontextart beruht auf der Tatsache, dass es sich bei GUI-Text um Fachtext handelt (s. Abschnitt 4.2), der in erster Linie Fachtermini verschiedener Wortbildungsarten enthalten kann (vgl. beispielsweise Beste 2006, S. 41).

⁷<http://www.corpustool.com/>; letzter Zugriff: 17.06.2025

Funktionaler Kontext: Diese Kontextart basiert auf dem Zusammenhang zwischen Strings und zugehörigen Programmfunktionen (s. Abschnitt 4.3). Alle Strings bzw. GUI-Elemente in einem Softwareprodukt weisen eine „genau definierte Funktionalität innerhalb der Software“ (K.-D. Schmitz 2008, S. 271) auf.

Im Rahmen der Korpusuntersuchung wurde der Versuch unternommen, diese drei Kontextarten mithilfe der in Abschnitt 4.6 beschriebenen Probleme zu operationalisieren, um so eine Annotation des Korpus zu ermöglichen. In den folgenden Abschnitten werden diese Probleme nochmals erläutert sowie die dafür entwickelten Features definiert und anhand von Beispielen eingegrenzt, um für nachvollziehbare und objektive Kategorien zu sorgen (vgl. Lemnitzer und Zinsmeister 2015, S. 102).

Ein paar Aspekte, die für die Annotation generell gelten, finden sich im Folgenden (die Details zu den dabei beispielhaft genannten Features werden in den weiteren Abschnitten erläutert):

- Strings können gleichzeitig mit verschiedenen Kontextarten annotiert werden. Beispiel: `Save log to:` (*Audiobearbeitung*⁸): Für „Save log to“ wurde das Feature `über_präposition` (situativer Kontext) und für „log“ das Feature `substantivischer_fachterm` (fachlicher Kontext) annotiert.
- Manche der Annotationen umfassen komplette Strings, manche nur Teile von Strings. Das kann von der Länge der Strings abhängen, aber auch vom annotierten Feature (s. obiges Beispiel `Save log to:`).
- Ein Teil der annotierten Probleme kommt in der Praxis häufig vor, weshalb es dafür in aller Regel auch gängige Übersetzungen gibt, mit denen die annotierten Probleme umgangen bzw. gelöst werden. In der Praxis sind diese Probleme also nicht relevant, sofern die Übersetzerinnen die gängigen Übersetzungen kennen oder recherchieren können. Im Korpus wurden solche Stellen dennoch annotiert. Beispiel: `Save As` (u. a. im Subkorpus *Dateimanager*). Dieser String wird laut Microsoft-Terminologiedatenbank (*Microsoft Terminology 2025*) standardmäßig mit „Speichern unter“ übersetzt.⁹

⁸Im Folgenden wird beim Zitieren von Strings aus dem Korpus jeweils in Klammer der Programmtyp angegeben, der in den Listen in Abschnitt 5.3.1 zu finden ist. Dieser Programmtyp wurde auch als Name der zugehörigen Datei im Korpus verwendet.

⁹Wahle gibt außerdem das Beispiel `None` an, das im Subkorpus *HTML-Editor* enthalten ist (Feature `flexion_pronomen`). Dafür gibt sie verschiedene Übersetzungsmöglichkeiten an, die die Problematik umgehen, dass für die Herstellung grammatischer Kongruenz das Geschlecht des Bezugsworts erforderlich ist (vgl. Wahle 2000b, S. 44–45).

- Einige der annotierten Probleme existieren nur, wenn die Strings jeweils für sich betrachtet werden, was in der Praxis in der Regel nicht der Fall ist. Sowohl durch Übersetzen weiterer Strings im Rahmen des Gesamtprojekts als auch der konkreten Strings in direkter Umgebung des aktuellen Strings entsteht Projektwissen, auch wenn die Übersetzung in Ressourcendateien erfolgt. Zudem können beispielsweise über String-IDs oder Kommentare (s. Abschnitt 4.5 sowie die Erläuterungen und Beispiele zum Kontext in der Ressourcendatei in Abschnitt 4.4) weitere Informationen bereitstehen (vgl. beispielsweise Wahle 2000b, S. 38, 40, 45, und Sachse 2005, S. 147–148). Bei der Annotation des Korpus wurden solche Informationen bewusst außer Acht gelassen. Es wurde also jeder String nur für sich selbst betrachtet und alle etwaigen weiteren hilfreichen Informationen wurden ausgeblendet. Beispielsweise folgen im Subkorpus *Dateimanager* ab Zeile 6 eine Reihe von Strings aufeinander, wobei das Wort „Copy“ jeweils mit einem Feature des Untersystems `PROBLEM_WORTART` annotiert ist: `Choose template...`, `C&heck free space`, `Cop&y attributes`, `Copy o&wnership` usw. Die ersten beiden der Strings bestehen eindeutig aus Verb und Substantiv/Kompositum, daher liegt die Vermutung nahe, dass dies für die letzten beiden ebenso zutrifft. Für sich betrachtet, könnte das Wort „Cop&y“ bei den letzten beiden allerdings auch ein Substantiv sein. Daher wurde für diese Wörter ein Feature des Untersystems `PROBLEM_WORTART` annotiert.
- Die im Folgenden beschriebenen Features können nicht als vollständig für die Grundgesamtheit angesehen werden, sondern primär für das untersuchte Korpus. Beispielsweise sind weitere Varianten von unklaren Aneinanderreihungen in der Kategorie „Funktionaler Kontext“ denkbar.

Vor Beginn der eigentlichen Annotation, die in UAM CorpusTool Version 3.3¹⁰ durchgeführt wurde, wurde ein Annotationsschema erstellt. Dieses Schema umfasst die Subnetze `KONTEXTART`, `LÄNGE` und `STEUERZEICHEN` (s. Abbildung 5.1). Das Subnetz `KONTEXTART` wiederum teilt sich entsprechend den oben beschriebenen Kontextarten in die Unterschemas `SITUATIVER_KONTEXT`, `FACHLICHER_KONTEXT` und `FUNKTIONALER_KONTEXT` auf. Die anderen beiden Subnetze weisen keine Unterschemas auf.

In den folgenden Abschnitten werden zunächst die Systeme und Features des Subnetzes `KONTEXTART` erläutert, daran schließen sich Ausführungen zu den Subnetzen `LÄNGE` und `STEUERZEICHEN` an (s. Abschnitt 5.5.4, wo sich auch weitere Erläuterungen zu diesen beiden Subnetzen finden), mit deren Hilfe weitere Daten zu den annotierten Strings erhoben werden sollten.

Abbildung 5.1 gibt einen Überblick zum gesamten Annotationsschema.

¹⁰<http://www.corpustool.com/>; letzter Zugriff: 17.06.2025



Abbildung 5.1: Annotationsschema des Korpus

5.5.1 Situativer Kontext

In Abschnitt 4.6.1 wurden folgende Kategorien als Problembereiche des situativen Kontexts herausgearbeitet:

- Ambige Wortarten: Im Englischen gibt es zahlreiche Wörter, die verschiedenen Wortarten angehören können, was insbesondere bei kurzen Strings zu Problemen führt, wenn die intendierte Wortart nicht zweifelsfrei zu erkennen ist (vgl. beispielsweise Beste 2006, S. 73).
- Bezug auf andere GUI-Elemente: Manche Strings weisen einen Bezug auf ein anderes GUI-Element oder einen Bereich der GUI auf. Dies kann über eine Präposition am Anfang oder am Ende eines Strings erfolgen (vgl. beispielsweise Anhang 1 in Melby und Foster 2010, S. 1) oder über Lokaldeixis (vgl. Finkbeiner 2015, S. 38).
- Allein stehende Pronomen und Adjektive: Wenn Strings aus allein stehenden Pronomen oder Adjektiven bestehen bzw. solche Wörter enthalten, kann bei der Übersetzung in flektierende Sprachen die Festlegung der korrekten grammatischen Form problematisch sein, da dazu das Geschlecht des Bezugsworts benötigt wird (vgl. beispielsweise Beste 2006, S. 73).

Im Unterschema `SITUATIVER_KONTEXT` wurden auf Basis dieser Kategorien für die Problematik mit ambigen Wortarten das Untersystem `PROBLEM_WORTART`, für die Problematik mit dem Bezug auf andere GUI-Elemente das Untersystem `PROBLEM_GUI-BEZUG` und für die Problematik mit allein stehenden Pronomen und Adjektiven das Untersystem `PROBLEM_FLEXIONSENDUNG` erstellt.

Erläuterungen und Beispiele zu diesen Untersystemen mit ihren Features sind in den folgenden Abschnitten enthalten (vgl. dazu auch den in Abbildung 5.2 gezeigten Entscheidungsbaum für den situativen Kontext).

Zusätzlich zu den oben genannten generellen Aspekten der Annotation wurden bei der Annotation der Features in der Kategorie „Situativer Kontext“ folgende Punkte beachtet:

- Einzelne Strings können auch Probleme aus zwei Features des situativen Kontexts enthalten. Beispiel: `Search for (Browser)`: „Search“ kann hier verschiedenen Wortarten (Substantiv, Verb) angehören und am Ende des Strings steht eine Präposition, daher wurden hierfür die beiden betreffenden Features annotiert.
- Zweifelsfälle sollten annotiert werden.
- Es wurden jeweils ganze Strings annotiert. Lediglich bei Wortartproblemen wurde nur das entsprechende Wort annotiert (s. Angaben im Entscheidungsbaum).

Situativer Kontext – Entscheidungsbaum

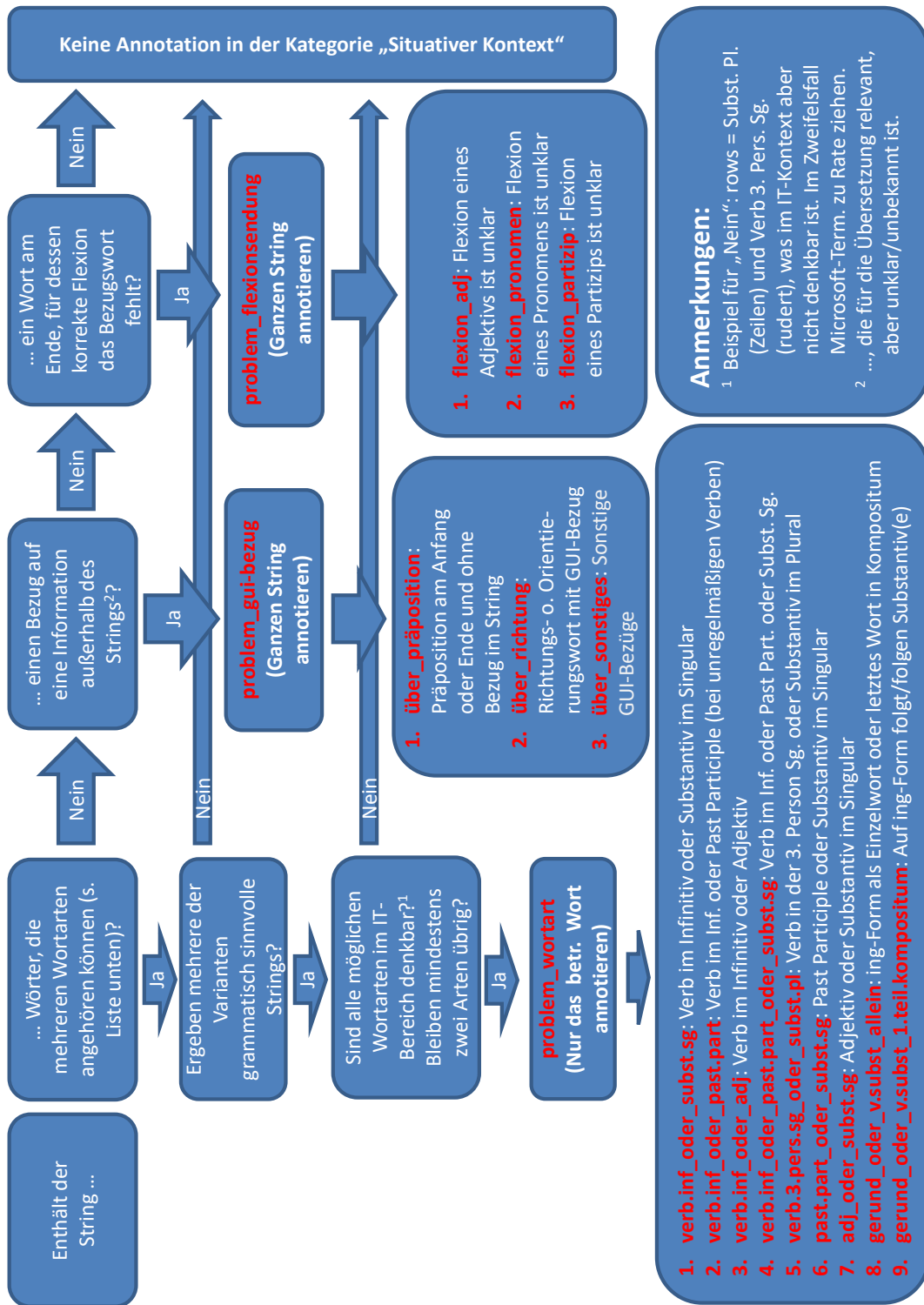


Abbildung 5.2: Entscheidungsbaum für den situativen Kontext

5.5.1.1 PROBLEM_WORTART

Für die Problematik mit ambigen Wortarten (s. Abschnitt 4.6.1 und vgl. beispielsweise Beste 2006, S. 73) wurde das Untersystem PROBLEM_WORTART genutzt, das die im Folgenden detailliert beschriebenen Features umfasst. Dabei wurden folgende Aspekte berücksichtigt:

- Es wurden generell nur im IT-Bereich benutzte Wortbedeutungen berücksichtigt, also nicht beispielsweise „show = Show“, „select = ausgewählt/erlesen“, „file = abheften“, „save = sicher“, „mute = stumm“. Referenzquelle hierfür waren die Inhalte der Microsoft-Terminologiedatenbank (vgl. *Microsoft Terminology* 2025).
- Das betreffende Wort wurde auf Basis dessen annotiert, was im untersuchten String grammatisch möglich war, nicht auf Basis dessen, was theoretisch bei diesem Wort für sich allein betrachtet möglich wäre. Für die Ermittlung der prinzipiell möglichen Wortarten wurden die Angaben im englischsprachigen Onlinewörterbuch Merriam Webster (vgl. *Merriam-Webster* 2025) herangezogen. Beispielsweise werden im Merriam Webster für das Wort „output“ die Wortarten Substantiv und Verb angegeben, wobei eine der angegebenen Formen des Past Participle ebenfalls „output“ lautet.¹¹ Wenn ein String lediglich aus dem Wort „output“ besteht (z. B. im Subkorpus *Videobearbeitung*), wurde dafür also das Feature `verb.inf_oder_past.part_oder_subst.sg` annotiert. Im String `Access output (Medienplayer)` wurde dagegen für das Wort „output“ das Feature `past.part_oder_subst.sg` annotiert. Grund dafür ist, dass „output“ in diesem String als Substantiv und somit Teil eines Kompositums und auch als Past Participle an Stelle eines Nebensatzes mit Verb betrachtet werden kann, der sich auf das Substantiv „access“ bezieht (vgl. Aarts 2011, S. 157). In diesem String kann es allerdings aufgrund der Wortstellung nicht als Verb im Infinitiv fungieren. Wenn dem so wäre, würde es an erster Stelle des Strings stehen.
- Bei der Annotation sollte wie oben allgemein erläutert jeder String für sich betrachtet werden. Zudem sollte nicht danach gesucht werden, welche Übersetzung am plausibelsten erscheint, sondern es wurden jeweils wie im vorangehenden Punkt beschrieben alle im String grammatisch möglichen Varianten identifiziert und annotiert, um die Annotation weniger subjektiv zu machen. Zwei Beispiele:

¹¹<https://www.merriam-webster.com/dictionary/output>; letzter Zugriff: 21.06.2025

1. Der String `Access modules` (*Medienplayer*): Es ist sehr wahrscheinlich, dass „Access“ in diesem String als Substantiv betrachtet werden muss, da im nächsten String die Information `Settings related to the various access methods` enthalten ist. Die korrekte Übersetzung wäre also „Zugriffsmodule“. Grammatisch möglich wäre in diesem String aber auch, „Access“ als Verb im Infinitiv zu betrachten, also „(to) access modules“ = „Auf Module zugreifen“.¹² Für diesen String wurde also das Feature `verb.inf_oder_subst.sg` annotiert. Im Unterschied dazu der String `Last access:` (*Dateimanager*). Hier kann „access“ grammatisch nicht als Verb betrachtet werden, sondern nur als Substantiv mit dem vorangestellten Adjektiv „last“. Dieser String wurde also nicht annotiert.

2. Der String „Warning“ (z. B. Subkorpus *Office-Anwendung*) ist grammatisch gesehen mit sehr hoher Wahrscheinlichkeit das Verbalsubstantiv, theoretisch ist aber auch denkbar, dass hier via Verb im Gerund ein Prozess ausgedrückt wird, der gerade läuft, also „Es wird (gerade) gewarnt/Die Warnung läuft“ (analog zu „Executing ...“ oder „Installing ...“) und anderen Statusmeldungen (vgl. beispielsweise Beste 2006, S. 63, und Esselink 2000, S. 63). Entsprechend wurde „Warning“ mit dem Feature `gerund_oder_v.subst_allein` annotiert.¹³

Wie sich bei der explorativen Untersuchung des Korpus zeigte, fanden sich beim Prüfen auf die eben genannten Kriterien deutlich mehr Mehrdeutigkeiten in Bezug auf die Wortarten, als die in Abschnitt 4.6.1 aus der Literatur zitierten Varianten Verb-Substantiv (vgl. beispielsweise Beste 2006, S. 73) und Verb-Adjektiv (vgl. Ottmann 2005, S. 103). Dies zeigt sich in der folgenden Liste der Features aus diesem Untersystem. Es ist allerdings möglich, dass es noch weitere Wortartenprobleme gibt, die nicht im vorliegenden Korpus zu finden waren.

verb.inf_oder_subst.sg: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Verb im Infinitiv als auch als Substantiv betrachtet werden können. Beispiele:

- **Copy** (z. B. *Audiobearbeitung*): to copy; the copy
- **Search** (z. B. *Browser*): to search; the search
- **Change directory** (*Dateimanager*): to change directory; the change directory

¹²Siehe <https://www.merriam-webster.com/dictionary/access>; letzter Zugriff: 21.06.2025

¹³Siehe <https://www.merriam-webster.com/dictionary/warn> und <https://www.merriam-webster.com/dictionary/warning>; letzter Zugriff: 21.06.2025

Wenn auf das betreffende Wort ein Adjektiv oder Partizip folgt, wurde dieses Feature nicht annotiert, weil hier eine eindeutig verbale Nutzung vorliegt. Beispiel: `~Insert Named Range or Expression...` (*Office-Anwendung*).

verb.inf_oder_past.part: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Verb im Infinitiv als auch als Past Participle betrachtet werden können. Entsprechend sind alle Wörter in dieser Kategorie unregelmäßige Verben. Beispiele:

- `Split Cell` (*HTML-Editor*): to split a cell; the split cell
- `_Set as default account` (*E-Mail*): to set as the default account; (something was/is) set as default account

verb.inf_oder_adj: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Verb im Infinitiv als auch als Adjektiv betrachtet werden können. Beispiele:

- `&Open...` (*Videobearbeitung*): to open; something is open; ebenso `Open offline storage message panel` (*Browser*): „Geöffnete Anzeige ...“ und „Anzeige ... öffnen“. Aber: `No open tabs` (*Browser*) wurde nicht annotiert, weil hier nur die adjektivische Verwendung von `open` möglich ist.
- `Clear` (*Office-Anwendung*): to clear; a clear view (o. ä.)

verb.inf_oder_past.part_oder_subst.sg: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle als Verb im Infinitiv, als Past Participle sowie als Substantiv im Singular betrachtet werden können. Auch hier müssen die Verben unregelmäßig sein. Beispiele:

- `Cut` (*Videobearbeitung*): to cut; something is cut; the cut
- `Reset` (*Browser*): to reset; something is reset; the reset

verb.3.pers.sg_oder_subst.pl: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Verb in der dritten Person Singular als auch als Substantiv im Plural betrachtet werden können. Beispiele:

- `Bookmarks` (*Browser*): somebody bookmarks a webpage; the bookmarks
- `Filters` (*Videobearbeitung*): somebody filters a list; the filters

past.part_oder_subst.sg: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Past Participle als auch als Substantiv im Singular betrachtet werden können. Beispiel:

- `Video Output` (*Videobearbeitung*): the video output; the video that was output
- `There is no filtering action set` (*E-Mail*): no action was set; a set that comprises filtering actions

adj_oder_subst.sg: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Adjektiv als auch als Substantiv im Singular betrachtet werden können. Beispiele:

- `Close Duplicate Tabs` (*Dateimanager*): the tabs that are duplicates, the tabs that are duplicate
- `Variable width` (*HTML-Editor*): a width that is variable; the width of a variable

`Medium` (z. B. *Videobearbeitung*) wurde nicht mit diesem Feature annotiert, weil die adjektivische (z. B. „medium size“) und die substantivische Variante (mit dem Plural „media“) Homonyme mit unterschiedlicher Bedeutung sind.

gerund_oder_v.subst_allein: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Gerund als auch als Verbalsubstantiv betrachtet werden können, das alleine bzw. am Ende einer Reihe von Substantiven steht. Beispiele:

- `Painting in GIMP` (*Bildbearbeitung*): the process of painting in GIMP is running; the painting in GIMP
- `Warning` (z. B. *Office-Anwendung*): the process of warning is running; the warning

Ein Beispiel für einen Fall, der in diesem Feature nicht annotiert wurde:

- `Decoding failed` (*Audiobearbeitung*): Die ing-Form wird hier als Subjekt eines verkürzten Satzes ohne Artikel, also substantivisch, gebraucht: „The decoding failed“.

gerund_oder_v.subst_1.teil.kompositum: Mit diesem Feature wurden Wörter annotiert, die an der betreffenden Stelle sowohl als Gerund als auch als Verbalsubstantiv betrachtet werden können, das im letzteren Fall dadurch zum ersten Teil eines Kompositums wird. Beispiele:

- **Bookmarking Tools** (*Browser*): the process in which tools are bookmarked is running; the tools used for bookmarking
- **Decoding video...** (*Videobearbeitung*): the process in which a video is decoded is running; the video used for decoding

5.5.1.2 PROBLEM_GUI-BEZUG

Für die Problematik mit dem Bezug von Strings auf andere GUI-Elemente wurde das Untersystem **PROBLEM_GUI-BEZUG** genutzt (s. Abschnitt 4.6.1). Dieser Bezug kann über eine Präposition am Anfang oder am Ende eines Strings (vgl. beispielsweise Anhang 1 in Melby und Foster 2010, S. 1) oder über Lokaldeixis (vgl. Finkbeiner 2015, S. 38) erfolgen. Bei der explorativen Untersuchung zeigte sich zudem, dass in diesem Ursystem noch ein drittes Feature für Fälle benötigt wurde, bei denen weder eine Präposition noch Lokaldeixis vorlag. Alle diese Features werden im Folgenden beschrieben.

über Präposition: Mit diesem Feature wurden nicht nur Strings annotiert, die lediglich aus einer Präposition bestehen, sondern auch alle weiteren Strings, an deren Anfang oder Ende eine Präposition steht, da auch in diesen Fällen wie von Melby beschrieben das zugehörige Objekt oder das zugehörige Verb fehlt (vgl. Anhang 1 in Melby und Foster 2010, S. 1), das für eine korrekte Übersetzung erforderlich ist. Der Bezug und somit das Objekt entsteht entweder durch Anordnung der GUI-Elemente zueinander oder implizit, indem durch Auswahl des entsprechenden Strings im Menü ein Dialog mit weiteren Informationen geöffnet wird. Letzteres ist beispielsweise typisch für die verbreitete Menüoption „Save as“. Wenn diese Menüoption ausgewählt wird, wird ein Dialog geöffnet, in dem die Benutzerin zur Eingabe eines Dateinamens für die Speicherung der Datei aufgefordert wird. Das fehlende Objekt ist hier also der einzugebende Dateiname. Beispiele:

- **You are securely connected to this site owned by:** (*Browser*)
- **with:** (*Browser*)
- **To the &directory:** (*Dateimanager*)

Anmerkungen zur Vorgehensweise:

- Es wurden auch Phrasal Verbs annotiert wie **Rep&lance with :** (*Editor*).
- Nicht annotiert wurden Strings wie **&Destination that the link will point to** (*Dateimanager*). „to“ bezieht sich in diesem Fall auf „destination“, das Bezugswort ist also vorhanden.

- Nicht annotiert wurden Strings wie `Sign in to {1}` (*Browser*). In diesem Fall ist das Objekt in Form einer Variablen vorhanden. Das Übersetzungsproblem entsteht also dadurch, dass der genaue Inhalt der Variablen nicht bekannt ist, der zur Laufzeit als Wort in den String eingefügt wird.
- Nicht annotiert wurden Strings wie `Audacity is starting up ...` (*Audiobearbeitung*). In diesem Fall steht zwar die Präposition am Ende des Strings, es fehlt aber kein Objekt.

über_richtung: Mit diesem Feature wurden Strings annotiert, die eine Lokaldeixis in Form von Richtungs- bzw. Orientierungswörtern enthalten, wie „left“, „right“, „up“, „down“, „forward“, die einen Bezug zu einem anderen GUI-Element herstellen, beispielsweise zu einem Dialogfeld, einem Bereich darin oder zu einem anderen String im Menü. Dieser deiktische Bezug zur GUI muss für die korrekte Übersetzung bekannt sein. Beispiele:

- `Move Tab Backward` (*Editor*): Tab nach hinten/zurück/an den Anfang verschieben?
- `Save Right` (*Dateimanager*): Nach rechts speichern/Rechts speichern/Das rechte Element speichern?

Wenn aus dem String ersichtlich war, dass es sich bei einem Wort nicht um ein Richtungswort handelt, sondern um einen Teil einer anderen Wendung, wie bei `Audacity is starting up...` (*Audiobearbeitung*), wo „up“ zum Verb „start“ gehört, wurde nicht annotiert. Ebenso wurde nicht annotiert, falls das betreffende Wort nicht der Richtung oder Orientierung dient. Beispiel: „Open checksum file after job is completed“ (*Dateimanager*). Hier wird „after“ nicht als Orientierungsangabe, sondern als Konjunktion genutzt.

über_sonstiges: Mit diesem Feature wurde ein String annotiert, bei dem es einen Bezug auf ein anderes GUI-Element gibt, der durch die anderen Features nicht abgedeckt wird:

- `&Find what` : (*Editor*)

5.5.1.3 PROBLEM_FLEXIONSENDUNG

Für Probleme mit Flexionsendungen von Strings wurde das Untersystem `PROBLEM_FLEXIONSENDUNG` genutzt, das die im Folgenden detailliert beschriebenen Features umfasst. Als Basis dafür dienten die in Abschnitt 4.6.1 vorgestellten und der Literatur entnommenen Problembereiche allein stehende

Pronomen und Adjektive (vgl. beispielsweise Beste 2006, S. 73). Bei der explorativen Untersuchung zeigte sich, dass zu diesem Problembereich wie von Beste angegeben auch längere Strings gehören, bei denen Adjektive oder Pronomen am Ende und in einzelnen Fällen auch in der Mitte von Strings stehen und ihr Bezugswort fehlt, dessen Genus für die korrekte Flexion des Adjektivs bzw. Pronomens erforderlich ist.

In Bezug auf die Übersetzung ins Deutsche gibt es wie oben bereits erwähnt für einige der Fälle, die in diesem Schema annotiert wurden, konventionelle Lösungen, die die Übersetzungsprobleme umgehen. Beispielsweise kann in eine unflektierte Wortart oder in den Plural gewechselt werden:

- **Previous** (*Audiobearbeitung*): „Zurück“ statt beispielsweise „Letzte/r/s“
- **Next** (*Audiobearbeitung*): „Weiter“ statt beispielsweise „Nächste/r/s“

Solche Strings wurden aber dennoch annotiert. Im Folgenden ist die Liste der Features in diesem Untersystem mit Beispielen zu finden:

flexion_adj: Mit diesem Feature wurden Strings mit einem Adjektiv annotiert, dessen Flexion wegen des unklaren Bezugs und dadurch des unklaren Genus problematisch ist. Beispiele:

- **Open &Recent...** (*Audiobearbeitung*): Kürzliches/Kürzliche ... öffnen
- **LIBVA (best)** (*Videobearbeitung*): bester/beste/bestes
- **Close All to the Left** (*Editor*): alles/alle

Nicht annotiert wurden allein stehende Adjektive, die in der Grundform bzw. adverbial übersetzt werden können, beispielsweise der String **Full**: „Vollständig“ (*Dateimanager*).

flexion_pronomen: Mit diesem Feature wurden Strings mit einem Pronomen annotiert, dessen Flexion wegen des unklaren Bezugs und dadurch des unklaren Genus problematisch ist. Beispiele:

- **Close All BUT This** (*Editor*): dieser/diese/dieses
- **Yes, save it** (*HTML-Editor*): ihn/sie/es

flexion_partizip: Mit diesem Feature wurden Strings mit einem Partizip annotiert, dessen Flexion wegen des unklaren Bezugs und dadurch des unklaren Genus problematisch ist. Beispiele:

- **Import selected** (*Dateimanager*): Ausgewählte/Ausgewähltes/Ausgewählten importieren
- **Play filtered** (*Videobearbeitung*): Gefilterte/Gefiltertes/Gefilterten wiedergeben

5.5.2 Fachlicher Kontext

In Abschnitt 4.6.2 wurden folgende Kategorien als Problembereiche des fachlichen Kontexts genannt:

- Fachterminologie: In Strings kann Fachterminologie verschiedener Fachgebiete und verschiedener Wortarten oder auch Mehrwortbenennungen vorkommen (vgl. Drewer und Ziegler 2014, S. 178, und K.-D. Schmitz 2008, S. 272). Auch Kurzformen oder Akronyme sind typisch (vgl. Drewer und Ziegler 2014, S. 179, Albrecht 2013, S. 281, und Beste 2006, S. 35). Ebenso finden sich Eigennamen, beispielsweise von Unternehmen, Produkten (vgl. Drewer und Ziegler 2014, S. 139), Programmiersprachen und Technologien.
- Aneinanderreihungen mit Fachterminologie: Komposition als Wortbildungsmöglichkeit von Fachterminologie ist in GUI-Text weit verbreitet (vgl. Beste 2006, S. 34, Drewer und Ziegler 2014, S. 177–178, und Albrecht 2013, S. 107).

Im Unterschema `FACHLICHER_KONTEXT` wurde für die Kategorie „Fachterminologie“ eine Reihe von Features für die verschiedenen Wortarten, Abkürzungen und Eigennamen erstellt. Darüber hinaus enthält dieses Unterschema das Untersystem `FACHTERM_ANEINANDERREIHUNG` mit verschiedenen Features für Komposita und sonstigen Aneinanderreihungen aus dem fachterminologischen Bereich.

Erläuterungen und Beispiele zu diesen Features sowie zu dem Untersystem mit den zugehörigen Features sind in den folgenden Abschnitten enthalten (vgl. dazu auch den in Abbildung 5.3 gezeigten Entscheidungsbaum für den fachlichen Kontext).

Neben den generell für die Annotation berücksichtigten Vorgaben oben wurden bei der Annotation der Features in der Kategorie „Fachlicher Kontext“ folgende Punkte beachtet:

- Wie bereits erwähnt, wurden die Fachterminologie bei der Annotation in verschiedene Wortarten, wie Substantive, Adjektive und Verben, unterteilt. Die Details dazu finden sich unten bei den relevanten Features.
- Die Fachterminologie aus dem Betriebssystem (s. Abschnitt 4.2) wurde annotiert, wenn sie zum Zeitpunkt der Annotation nicht mit der relevanten IT-Definition im englischsprachigen Onlinewörterbuch Merriam Webster (*Merriam-Webster* 2025) enthalten war.
 1. Beispiele für IT-Fachterminologie, deren Definition im Merriam

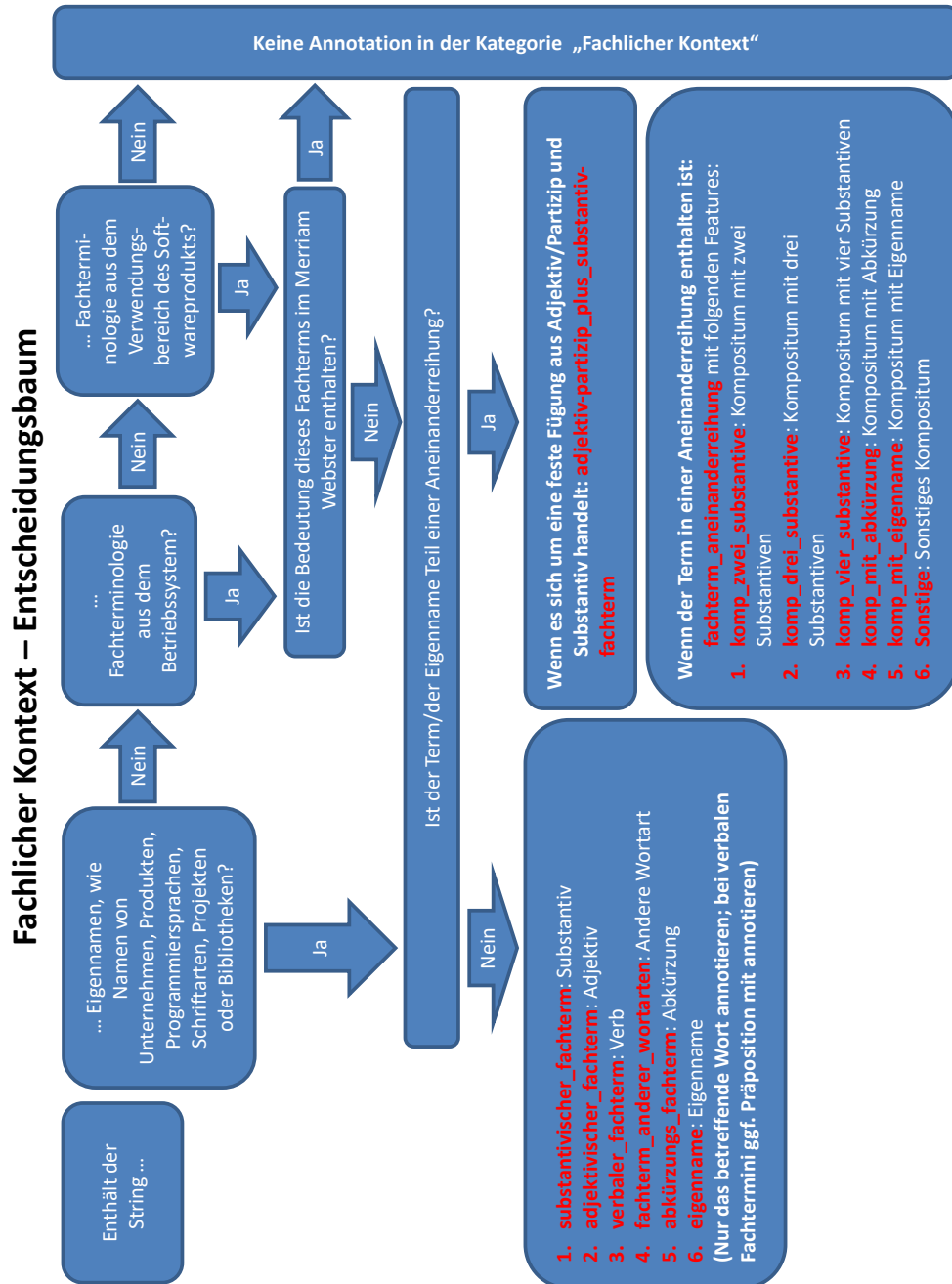


Abbildung 5.3: Entscheidungsbaum für den fachlichen Kontext

Webster enthalten ist: **directory** (*Dateimanager*¹⁴), **folder** (*E-Mail*), **file** (*Videobearbeitung*).

2. Beispiele für IT-Fachterminologie, deren Definition nicht im Merriam Webster enthalten ist: **panel** (*HTML-Editor*), **dialog** (*Office-Anwendung*), **extract** (*Dateimanager*; in der Definition laut Microsoft-Terminologie, *Microsoft Terminology 2025*: „To decompress, or pull out, files from a compressed form.“), **property** (*Dateimanager*; in der Definition laut Microsoft-Terminologie, *Microsoft Terminology 2025*: „An attribute of an object that is used to define its state, appearance, or value.“), **attribute** (*Dateimanager*; in der Definition laut Microsoft-Terminologie, *Microsoft Terminology 2025*: „A characteristic of an object.“).

- Die Fachterminologie aus dem Verwendungsbereich des jeweiligen Softwareprodukts (s. Abschnitt 4.2) wurde annotiert, falls deren Bedeutung nicht in den Definitionen im Merriam Webster enthalten ist.

1. Beispiele für Fachterminologie aus dem Fachgebiet des jeweiligen Softwareprodukts, deren Definition nicht im Merriam Webster enthalten ist: **demuxer** (*Medienplayer*), **sample rate conversion** (*Audiobearbeitung*).

2. Beispiele für Fachterminologie aus dem Fachgebiet des jeweiligen Softwareprodukts, deren Definition im Merriam Webster enthalten ist: **device** (*Browser*), **refresh** (*Office-Anwendung*).

- Wenn Fachterminologie als einer der Bestandteile in einem Kompositum bzw. in einer sonstigen Aneinanderreihung vorkommt, wurde die gesamte Aneinanderreihung mit dem betreffenden Aneinanderreihungsfeature für Fachterminologie annotiert, nicht nur einzelne Bestandteile davon, weil davon auszugehen ist, dass nicht nur zahlreiche Fachwörter (vgl. Albrecht 2013, S. 274), sondern insbesondere auch fachterminologische Komposita nicht in Wörterbüchern enthalten sind. Auf diese Weise konnte daher besser abgegrenzt werden zwischen Aneinanderreihungen mit Fachwortanteilen und Aneinanderreihungen, deren einzelne Bestandteile keine Fachwörter sind, deren Übersetzung aber dennoch problematisch ist (s. die Informationen zum funktionalen Kontext in den Abschnitten 4.6.3 und 5.5.3).
- Typischerweise in GUI-Text vorkommende Wörter, die aber keine spezielle IT-Fachbedeutung haben, wurden nicht annotiert. Beispiele: **undo** (*Videobearbeitung*), **redo** (*Editor*), **cut** (*Browser*), **copy** (*Audiobear-*

¹⁴Bei vielen der Beispiele in dieser Liste handelt es sich um Wörter, die in mehreren der im Korpus enthaltenen Softwareprodukte vorkommen. Die Quellenangaben können daher in aller Regel als Beispiele betrachtet werden.

beitung), *project* (*Editor*). Ob bei solchen Wörtern eine spezielle IT-Fachbedeutung vorliegt, wurde im Vergleich zwischen den Informationen im Onlinewörterbuch Merriam Webster ([Merriam-Webster 2025](#)) und der Definitionen in der Microsoft-Terminologiedatenbank ([Microsoft Terminology 2025](#)) überprüft.

- In die Kategorie „Fachlicher Kontext“ fällt keine Fachterminologie, die für das jeweilige Softwareprodukt geprägt wurde. Diese wurde ggf. mit einem Feature des funktionalen Kontexts (s. Abschnitt [5.5.3](#)) annotiert. Die im fachlichen Kontext annotierten Begriffe müssen also im Merriam Webster bzw. in den weiteren verfügbaren Quellen für Fachterminologie zu finden sein.
- Annotiert wurden jeweils die relevanten Wörter, also bei Fachterminologie die einzelnen Wörter, bei fachterminologischen Aneinanderreihungen die gesamte Aneinanderreihung.
- Die Einteilung in eines bzw. mehrere Wörter erfolgte auf Basis der Schreibung im AT. Beispiele: *sidebar* (*Browser*), *dashboard* (*Bildbearbeitung*), *checksum* (*Dateimanager*) wurden jeweils als ein Wort gezählt und daher im Feature `substantivischer_fachterm` annotiert. Im grammatischen Sinne könnten solche Wörter aber auch als Komposita aus zwei Bestandteilen betrachtet werden.
- In einigen Fällen wurden Wörter mit einem Feature des fachlichen Kontexts annotiert, die gleichzeitig auch mit einem der Features für ambige Wortarten aus dem situativen Kontext annotiert wurden. In solchen Fällen wurde für den fachlichen Kontext die aufgrund der verfügbaren Kontextinformationen korrekte Wortart gewählt. Beispiel: *Anchor* (*HTML-Editor*) wurde mit dem Feature `verb.inf_oder_subst.sg` und mit dem Feature `substantivischer_fachterm` annotiert. Das Feature `verbaler_fachterm` wurde nicht verwendet, weil im String danach die Information „Insert or edit an anchor“ steht, die die eindeutige Identifizierung der Wortart als Substantiv ermöglicht.

5.5.2.1 Fachterminologie

Für die Fachterminologie wurden die im Folgenden detailliert beschriebenen Features definiert. Diese wurden genutzt, um die Relevanz der verschiedenen Wortarten im Bereich Fachterminologie quantitativ zu erfassen (vgl. beispielsweise Drewer und Ziegler [2014](#), S. 139, 178–179, Albrecht [2013](#), S. 281, und K.-D. Schmitz [2008](#), S. 272).

substantivischer_fachterm: Mit diesem Feature wurden substantivische Fachtermini annotiert. Beispiele:

- After recovery, save the project to save the changes to disk. (*Audiobearbeitung*): „recovery“¹⁵
- Improved support for many image formats, in particular better PSD importing (*Bildbearbeitung*): „support“

Auch Wörter mit Vorsilben wie „auto-“ (z. B. Auto-Completion im Subkorpus *Editor*) oder „multi-“ (z. B. Multi-threading im Subkorpus *Videobearbeitung*) wurden zu substantivischen Fachtermini gezählt.

adjektiv-partizip_plus_substantiv-fachterm: Mit diesem Feature wurden Fachtermini annotiert, die aus einer festen Kombination aus Adjektiv bzw. Partizip und Substantiv bestehen, die also eine Mehrwortbenennung bilden (vgl. Drewer und Ziegler 2014, S. 178). Beispiele:

- Ordered list (*HTML-Editor*): „ordered list“
- Regular expression (z. B. *Editor*): „regular expression“

Zur Orientierung, ob eine Mehrwortbenennung vorliegt, wurde die Microsoft-Terminologiedatenbank (*Microsoft Terminology 2025*) genutzt.

adjektivischer_fachterm: Mit diesem Feature wurden adjektivi-sche Fachtermini annotiert. Beispiele:

- Recoverable projects (*Audiobearbeitung*): „recoverable“
- GIMP 2.10.0-RC1 is the first release candidate before GIMP 2.10.0 stable release, with a focus on debugging and stability. Other than the many bug fixes, most notable improvements are: (*Bildbearbeitung*): „stable“

verbaler_fachterm: Mit diesem Feature wurden verbale Fachtermini annotiert. Beispiele:

- Unpin from Overflow Menu (*Browser*): „unpin“
- &Extract Files... (*Dateimanager*): „extract“

Wenn eine Präposition zum Verb gehört, wurde diese mit annotiert, wie bei Zoom In (z. B. Office-Anwendung) und Wrap around (*Editor*).

fachterm_anderer_wortarten: Mit diesem Feature wurden Fachtermini anderer Wortarten annotiert. Weitgehend waren dies Partizipien. Beispiele:

¹⁵Bei längeren Strings wird jeweils anschließend das Wort angegeben, das mit dem betreffenden Feature annotiert wurde. Wenn nichts angegeben wird, wurde der gesamte String mit dem Feature annotiert.

- Color management is a core feature now, most widgets and preview areas are color-managed (*Bildbearbeitung*): „color-managed“
- A hidden string (*HTML-Editor*): „hidden“

abkürzungs-fachterm: Mit diesem Feature wurden fachterminologische Abkürzungen annotiert. Beispiele:

- External Audacity module which provides a simple IDE for writing effects. (*Audiobearbeitung*): „IDE“: Integrated Development Environment
- Windows (CRLF) (*Dateimanager*): „CRLF“: carriage return, line feed

Auch im Feature `eigennamen` sind einige Abkürzungen enthalten: Wenn die Abkürzungen für Technologien oder auch Formate stehen, z. B. HTML (*HTML-Editor*) oder XML (*Office-Anwendung*), wurden die Wörter dem Feature `eigennamen` zugewiesen (s. a. die Erläuterungen zum Feature `eigennamen`).

eigennamen: Mit diesem Feature wurden Eigennamen aus dem fachterminologischen Bereich annotiert. Dies umfasste beispielsweise Eigennamen aus den Bereichen Unternehmen, Produkte, Programmiersprachen, Dateiformate, Schriftarten, Projekte, Bibliotheken, Befehle oder Dateien. Beispiele:

- Dolby Pro Logic (*Videobearbeitung*)
- Unix (LF) (z. B. *Editor*): „Unix“
- log.txt (*Audiobearbeitung*)
- Save as PNG (*Videobearbeitung*): „PNG“

Wie oben geschildert, wurden eine Reihe von Abkürzungen, beispielsweise MP3 (*Videobearbeitung*), ID3 (*Audiobearbeitung*) und FFmpeg (*Audiobearbeitung*), in diesem Feature (bzw. im Feature `komp_mit_eigennamen`) annotiert, weil es sich dabei um die Benennung einer Technologie o. ä. handelt. Ein weiteres Beispiel: GIMP (*Bildbearbeitung*) = GNU Image Manipulation Program, also eine Abkürzung, die gleichzeitig der Name des Programms ist.

5.5.2.2 FACHTERM_ANEINANDERRHEIUNG

Für fachterminologische Aneinanderreihungen wurde das Untersystem `FACHTERM_ANEINANDERRHEIUNG` genutzt, das die im Folgenden detailliert beschriebenen Features umfasst. Für sämtliche Features in diesem Untersystem wurden folgende Vorgaben berücksichtigt:

- Analog zur Vorgehensweise oben wurde auch in diesem Untersystem das Onlinewörterbuch Merriam Webster (*Merriam-Webster 2025*) zu Rate gezogen: Wenn mindestens eines der Wörter, aus denen sich die Aneinanderreihung zusammensetzt, ein IT-Fachterminus ist und diese Bedeutung nicht im Online-Merriam-Webster enthalten ist, wurde das gesamte Kompositum mit dem passenden Feature des Untersystems `FACHTERM_ANEINANDERRREIHUNG` annotiert. Wenn alle Bestandteile der Aneinanderreihung entweder keine IT-Fachtermini sind oder mit ihrer IT-Fachbedeutung im Online-Merriam-Webster enthalten sind, wurde die gesamte Aneinanderreihung nicht annotiert. Beispiele:

1. `Insert or edit an audio file (HTML-Editor)`: „audio file“: Sowohl „audio“ als auch „file“ sind mit den passenden IT-Definitionen im Online-Merriam-Webster zu finden. Dieses Kompositum wurde nicht annotiert.

2. `Open install message panel (Browser)`: Die IT-Definition von „panel“ ist nicht im Online-Merriam-Webster zu finden, die anderen beiden („install“ und „message“) sind enthalten. Das gesamte Kompositum „install message panel“ wurde annotiert.

- Zur Feststellung, wie viele Wörter ein Kompositum umfasst, wurde die Schreibung im Korpus verwendet. Beispiel (*Editor*): `Edit Popup ContextMenu`: „Popup ContextMenu“ wurde mit dem Feature `komp_zwei_substantive` annotiert.

`komp_zwei_substantive`: Mit diesem Feature wurden Komposita mit zwei Substantiven annotiert, die mindestens einen Fachterminus enthalten. Beispiele:

- `Edit the Processing Instruction (HTML-Editor)`: „Processing Instruction“
- `Full File Path to Clipboard (Editor)`: „File Path“

Im Unterschied zum Untersystem `UNKLARE_ANEINANDERRREIHUNG` der Kategorie „Funktionaler Kontext“ (s. Abschnitt 5.5.3.1) wurden in der Kategorie „Fachlicher Kontext“ auch Komposita mit zwei Substantiven (bzw. Komposita mit Abkürzung/Eigenname plus Substantiv) annotiert, da hier der fachterminologische Aspekt im Vordergrund steht, nicht die Übersetzungsschwierigkeiten, die bei Komposita aus zwei Bestandteilen in der Regel deutlich geringer sind als bei umfangreicheren Komposita.

`komp_drei_substantive`: Mit diesem Feature wurden Komposita mit drei Substantiven annotiert, die mindestens einen Fachterminus enthalten. Beispiele:

- Pivot Table Filter (*Office-Anwendung*)
- Column rule width: (*HTML-Editor*)

komp_vier_substantive: Mit diesem Feature wurden Komposita mit vier Substantiven annotiert, die mindestens einen Fachterminus enthalten. Beispiele:

- Sheet Area Input Field (*Office-Anwendung*)
- Configuration of Tree View Menu Colors (*Dateimanager*): „Tree View Menu Colors“

komp_mit_abkürzung: Mit diesem Feature wurden Komposita unabhängig von ihrer Länge annotiert, die eine fachterminologische Abkürzung enthalten. Beispiele:

- Manage use of DRM software (*Browser*): „DRM software“
- GUI Rendering Options (*Videobearbeitung*)

Für Abkürzungen galten dabei die oben beim Feature `abkürzungs-fachterm` genannten Vorgaben.

komp_mit_eigenname: Mit diesem Feature wurden Komposita unabhängig von ihrer Länge annotiert, die einen Eigennamen enthalten. Beispiele:

- Nyquist Effect Workbench (*Audiobearbeitung*)
- CSS Properties (*HTML-Editor*)

Für Eigennamen galten dabei die oben beim Feature `eigenname` genannten Vorgaben.

sonstige: Mit diesem Feature wurden sonstige Aneinanderreihungen annotiert, die mindestens einen Fachterminus enthalten und nicht in die obigen Features fallen. Beispielsweise können dies komplexere Aneinanderreihungen mit „und“ und „oder“ bzw. Aneinanderreihungen mit unterschiedlichen Wortarten sein. Beispiele:

- View Synced Tabs Sidebar (*Browser*): „Synced Tabs Sidebar“
- Re_verse UP and DOWN arrow keys for navigation (*Videobearbeitung*): „UP and DOWN arrow keys“

5.5.3 Funktionaler Kontext

Im Abschnitt 4.6.3 wurden folgende Kategorien als Problembereiche des funktionalen Kontexts herausgearbeitet:

- Unklare Aneinanderreihungen: Längere Aneinanderreihungen von Substantiven führen zu Übersetzungsproblemen (vgl. K. J. Dunne 2009, S. 203). Dazu gehören Komposita (vgl. beispielsweise K.-D. Schmitz 2002, S. 22) sowie Komposita mit Adjektiven oder Partizipien (vgl. beispielsweise Wahle 2000a, S. 116, und Esselink 2000, S. 28) und Fälle von Generalisierung-Präzisierung (vgl. Fischer 2011, S. 38).
- Homonymie: Die Verwendung von Homonymen innerhalb eines Fachgebiets führt zu Kommunikationsproblemen (vgl. K.-D. Schmitz 2005b, S. 42).
- Metaphorik kommt häufig als Verfahren der Terminologisierung zum Einsatz, wenn für neue Begriffe bei neuer Funktionalität eines Softwareprodukts neue Benennungen erforderlich sind (vgl. Drewer und Ziegler 2014, S. 178–179).
- Unklare Bedeutungen entstehen dadurch, dass Strings zu knapp formuliert werden, wodurch für das Verständnis genaue Kenntnisse des Softwareprodukts erforderlich sind (vgl. Esselink 2000, S. 63), insbesondere auch bei Fehler- und Statusnachrichten (vgl. K. J. Dunne 2009, S. 202).

Im Unterschema `FUNKTIONALER_KONTEXT` wurden für erste diese Kategorien das Untersystem `UNKLARE_ANEINANDERREIHUNG` und für die übrigen die Features `homonym_it-term`, `metaphorik` und `unklare_bedeutung` erstellt.

Erläuterungen und Beispiele zu diesen Features sowie zu dem Untersystem mit den zugehörigen Features sind in den folgenden Abschnitten enthalten (vgl. dazu auch den in Abbildung 5.4 gezeigten Entscheidungsbaum für den funktionalen Kontext).

In den Features dieses Unterschemas wurden jeweils die relevanten Wörter annotiert. Lediglich beim Feature `unklare_bedeutung` wurden die ganzen Strings annotiert.

5.5.3.1 UNKLARE_ANEINANDERREIHUNG

Das Untersystem `UNKLARE_ANEINANDERREIHUNG` weist mehrere Features auf, die verschiedene Aspekte der in Abschnitt 4.6.3 beschriebenen Probleme mit der Übersetzung von Komposita und Generalisierung-Präzisierung umfassen. Bei allen diesen Features stehen daher Substantive im Fokus, wobei Eigennamen und Abkürzungen im Rahmen dieser Features auch als

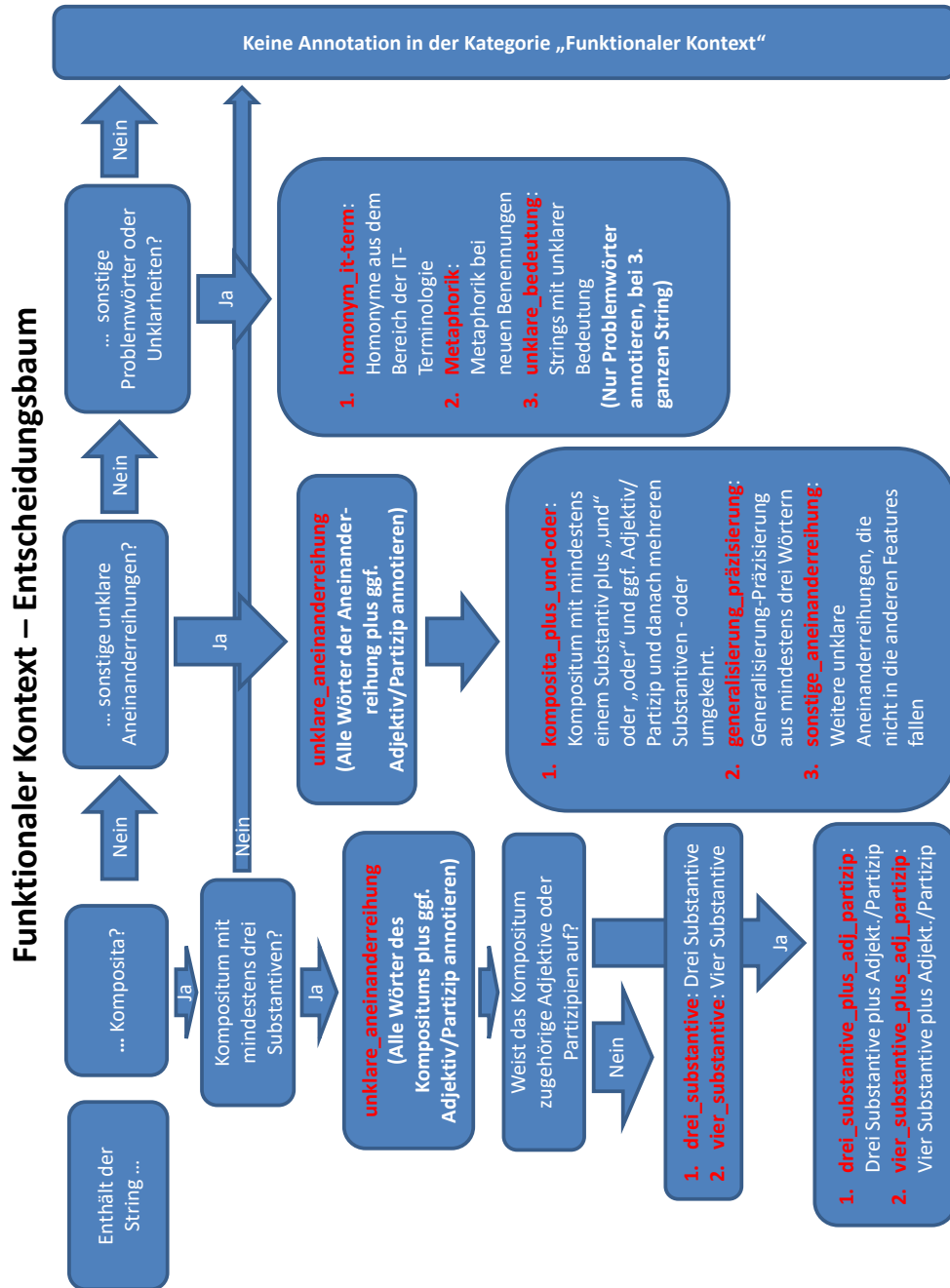


Abbildung 5.4: Entscheidungsbaum für den funktionalen Kontext

Substantive betrachtet werden. Die Features werden im Folgenden beschrieben. Zu beachten ist dabei, dass es eine gewisse Schnittmenge zwischen einzelnen Features dieses Untersystems und Features des Untersystems `FACHTERM_ANEINANDERREIHUNG` aus Abschnitt 5.5.2.2 gibt: Wenn fachterminologische Aneinanderreihungen auch die Anforderungen der Features `komp_drei_substantive`, `komp_vier_substantive`, `komp_mit_abkürzung`, `komp_mit_eigenname` oder `sonstige` erfüllen, wurden sie für beide Kategorien, „Fachlicher Kontext“ und „Funktionaler Kontext“, annotiert. Für die ersten beiden dieser Features mit drei und vier Substantiven trifft dies in allen Fällen zu, für die weiteren Features nur, wenn sie sich aus mindestens drei Bestandteilen zusammensetzen und die weiteren Bedingungen erfüllen, die im Folgenden erläutert werden.

Im Korpus waren keine längeren Komposita als mit vier Substantiven bzw. mit vier Substantiven plus Adjektiv oder Partizip enthalten. Daher wurden für solche Komposita keine Features in das Untersystem aufgenommen. Allerdings zeigten sich ein paar in der Literatur nicht explizit genannte Aneinanderreihungsphänomene, für die die Features `komposita_plus_und-oder` und `sonstige_aneinanderreihung` erstellt wurden.

drei_substantive: Mit diesem Feature wurden Komposita annotiert, die aus drei Substantiven bestehen. Beispiele:

- `Pivot Table Filter` (*Office-Anwendung*)
- `Socket IO Timeout` (*E-Mail*)
- `General settings for audio output modules`. (*Medienplayer*):
„audio output modules“

drei_substantive_plus_adj_partizip: Mit diesem Feature wurden Komposita annotiert, die aus drei Substantiven plus zugehörigem Adjektiv oder Partizip bestehen. Beispiele:

- `An image-based form submission button` (*HTML-Editor*): „image-based form submission button“
- `Open offline storage message panel` (*Browser*): „offline storage message panel“

vier_substantive: Mit diesem Feature wurden Komposita annotiert, die aus vier Substantiven bestehen. Beispiele:

- `Cycle Cell Reference Types` (*Office-Anwendung*)

- **Open add-on installation message panel** (*Browser*): „add-on installation message panel“

vier_substantive_plus_adj_partizip: Mit diesem Feature wurden Komposita annotiert, die aus vier Substantiven plus zugehörigem Adjektiv oder Partizip bestehen. Beispiel:

- **High bit depth color processing (16/32-bit per color channel)** (*Bildbearbeitung*): „high bit depth color processing“

komposita_plus_und-oder: Mit diesem Feature wurden Aneinanderreihungen annotiert, bei denen auf mindestens ein Substantiv ein „und“ oder „oder“ (o. ä.) folgt, woran sich wiederum mindestens zwei Substantive in Form eines Kompositums anschließen oder umgekehrt: Auf mehrere Substantive folgt ein „und“ oder „oder“ und daran schließt sich noch ein weiteres Substantiv an. Diese Aneinanderreihungen können darüber hinaus auch Adjektive oder Partizipien enthalten, die ebenfalls mit annotiert wurden. Beispiele:

- **Ogg Vorbis Import and Export** (*Audiobearbeitung*)
- **Pitch and Tempo Change support** (*Audiobearbeitung*)

Ebenso wie in Abschnitt 4.6.3 für die Komposita beschrieben, sind auch hier die Bezüge unklar, weshalb die Übersetzung problematisch ist. Dies liegt bei dieser Problemgruppe daran, dass im Englischen unklar ist, ob sich einzelne Wörter dieser Wortgruppen als Kompositabestandteile auf die Teile vor dem „und“ bzw. „oder“ beziehen oder nur auf einen dieser Teile. Im Englischen ist diese Konstruktion also mehrdeutig, im Deutschen wird die Ellipse dagegen durch einen Bindestrich eindeutig markiert. Die Übersetzung im ersten Beispiel könnte also **Ogg Vorbis-Import und -Export** oder auch **Ogg Vorbis-Import und Export** lauten. Im ersten Fall würde sich das Audioformat, Ogg Vorbis, auf den Import und auf den Export, im letzteren nur auf den Import beziehen.

generalisierung_präzisierung: Mit diesem Feature wurden Aneinanderreihungen aus mindestens drei Wörtern annotiert, die Fälle von Generalisierung-Präzisierung darstellen, wobei alle Substantive plus etwaige Adjektive oder Partizipien mit annotiert wurden. Beispiel:

- **Improvements for the Wavelet Decompose filter** (*Bildbearbeitung*): „Wavelet Decompose filter“

sonstige_aneinanderreihung: Mit diesem Feature wurden weitere unklare Aneinanderreihungen annotiert, die nicht in die obigen Features fallen. Beispiele:

- `~Named Range or Expression...` (*Office-Anwendung*)
- `Multi-threaded and hardware-accelerated rendering, processing and painting` (*Bildbearbeitung*)

Bei diesen Beispielen bestehen wie oben Unklarheiten bei Bezügen, bei diesen Beispielen aber von Partizipien auf Substantive.

5.5.3.2 homonym_it-term

Mit diesem Feature wurden Homonyme aus dem Bereich der IT-Fachterminologie annotiert, wobei die Homonyme derselben Wortart angehören. Beispiele:

- `Copy Image` (*Browser*): Die Benennung „image“ kann zu dem Begriff „Bild“ oder zu dem Begriff „Image“ gehören.¹⁶
- `Hidden Tabs` (*Browser*): Die Benennung „tab“ kann zu dem Begriff „Tabulator“ oder zu dem Begriff „Registerkarte“ gehören.

Falls der sprachliche Kontext in einem String ausreicht, um zu ermitteln, welcher der möglichen Begriffe gemeint ist, wurde nicht annotiert. Beispiel: Im String `Open In New Tab(s)` (*Dateimanager*) kann mit der Benennung „tab“ nur der Begriff „Registerkarte“ gemeint sein, die Übersetzung mit „Tabulator“ ist ausgeschlossen, da ein Tabulator nicht geöffnet werden kann.

5.5.3.3 metaphorik

Mit diesem Feature wurden Wörter annotiert, die weder im IT-Bereich noch im Fachgebiet des jeweiligen Softwareprodukts spezielle Fachterminologie darstellen, aber aus der Gemeinsprache bekannt sind. Die Verwendung dieser Wörter dient also in metaphorischem Sinne zur Benennung spezieller Funktionen des jeweiligen Softwareprodukts. Beispiele:

- `Save to Pocket` (*Browser*): „Pocket“

¹⁶Vgl. dazu die Definitionen in der Microsoft-Terminologie: Bild: „A pictorial graphic such as a digital photograph or illustration.“ Image: „A file that contains the operating system and application configuration that can be used to create any number of virtual machines.“ Und: „A collection of files and folders (sometimes compressed into one file) that duplicates the original file and folder structure of an operating system. It often contains other files added by the OEM or corporation.“ (*Microsoft Terminology 2025*)

- ~Detective (*Office-Anwendung*)
- Deck (*HTML-Editor*)
- Ludicrous (2×) (*Browser*)

Nicht annotiert wurden die im IT-Bereich oder dem jeweiligen Fachgebiet gängigen Benennungen, bei denen ebenfalls Metaphorik zum Einsatz kommt, wie „library“, sondern solche, die nur in einzelnen Projekten Verwendung finden. Zum Abgleich wurden die Microsoft-Terminologie (*Microsoft Terminology 2025*) in Bezug auf allgemeine IT-Fachterminologie und relevante Quellen für die jeweiligen Fachgebiete für diesen Typ von Fachterminologie genutzt.

5.5.3.4 unklare_bedeutung

Mit diesem Feature wurden Strings annotiert, deren Bedeutung auch nach der Terminologierecherche unklar bleibt. Das heißt, entweder enthalten sie keine IT-Fachterminologie bzw. keine spezielle Fachterminologie oder, falls doch, hilft diese Bedeutung nicht beim Erfassen der konkreten Bedeutung des betreffenden Strings. Solche Unklarheiten können beispielsweise daraus resultieren, dass ein Sachverhalt in einem String zu kurz oder grammatisch unklar dargestellt ist oder dass der String Wörter mit einer größeren semantischen Bandbreite enthält. Beispiele:

- **Brief view** (*Dateimanager*): Welche genaue Bedeutung hat in diesem String das Wort „brief“?
- **Keep Scrolling** (*Dateimanager*): Die Bedeutung der einzelnen Wörter ist hinreichend klar, unklar aber ist der Sachverhalt, der mit diesem String geschildert wird.
- **Common addresses** (*E-Mail*): „Common“ kann viele unterschiedliche Bedeutungen haben, wie „allgemein“, „gemeinsam“, „üblich“ (vgl. beispielsweise *Merriam-Webster 2025*). Welche dieser Bedeutungen in diesem String gemeint ist, ist unklar.
- **Status Expanded Selection** (*Office-Anwendung*): Die Bezüge der Wörter zueinander sind in diesem String unklar.

5.5.4 Weitere ergänzende Annotationen

Ergänzend zur Annotation der Kontextarten wurden zwei weitere Subnetze eingefügt, die zusätzliche Informationen für die Auswertung erbringen sollten: die Anzahl der Wörter im jeweiligen String (Abschnitt 5.5.4.1) sowie die in den annotierten Strings enthaltenen Steuerzeichen (Abschnitt 5.5.4.2).

5.5.4.1 LÄNGE

Als zweites Subnetz umfasst das Annotationsschema das System LÄNGE, mit dem parallel zu den Features für die Kontextarten annotiert wurde, wie lange der String mit der zugehörigen Kontextannotation ist. Folgende Features wurden in diesem Subnetz annotiert: `ein_wort`, `zwei_wörter`, `drei_wörter` und `länger`. Das letzte Feature wurde also für alle Strings verwendet, die mindestens vier Wörter umfassen, und stellt damit im Unterschied zu den anderen Features eine sehr grobe Kategorie dar. Für die Annotation dieses Systems galten folgende Vorgaben:

- Bei der Wortzählung wurde grundsätzlich die in den Strings enthaltene Schreibung als Basis angenommen, wobei Leerzeichen jeweils als Trennzeichen für verschiedene Wörter interpretiert wurden. In Einzelfällen wurden dadurch bei unterschiedlichen Schreibweisen dieselben Wörter unterschiedlich gezählt. Beispiel: Die Schreibung „addressbook“ und die Schreibung „address book“ kommen im Subkorpus *E-Mail* vor, was entsprechend der Schreibung im ersten Fall als ein Wort und im zweiten als zwei Wörter gezählt wurde.
- Separat als Teil der Strings aufgeführte Shortcuts wurden nicht als Wörter gezählt. Beispiel (*Audiobearbeitung*): `Select A&11\tCtrl+A`: 2 Wörter
- Variablen wurden jeweils als Wörter mitgezählt. Beispiel: `Search '%s'` (*E-Mail*): 2 Wörter
- `SHA-256` (*Editor*) und ähnliche Strings wurden als ein Wort gezählt.
- Schlüsselwörter aus dem IT-Bereich wurden als ein Wort gezählt. Beispiel (*Editor*): `Monitoring (tail -f)`: 3 Wörter
- Zahlen wurden als ein Wort gezählt. Beispiel (*Editor*): `Project Panel 1`: 3 Wörter
- Leerzeichen wurden in allen Fällen als Worttrenner aufgefasst. Beispiel (*Editor*): `Windows (CR LF)`: 3 Wörter
- Schrägstriche wurden in allen Fällen, in denen sie zum Trennen von zwei Wörtern genutzt wurden, wie Leerzeichen behandelt. Beispiel (*Editor*): `Comment/Uncomment`: 2 Wörter
- Unterstriche wurden nicht als Leerzeichen interpretiert. Beispiel (*Audiobearbeitung*): `translator_credits`: 1 Wort

5.5.4.2 STEUERZEICHEN

Als drittes Subnetz ist im Annotationsschema das System STEUERZEICHEN enthalten, mit dem annotiert wurde, ob in dem String mit der zugehörigen Kontextannotation ein Steuerzeichen enthalten ist und um welches es sich

dabei handelt (s. Abschnitt 2.2). Folgende Features wurden in diesem Subnetz annotiert:

- **accesskey**: Strings mit Access-Key. Je nach Computerprogramm „&“ (bzw. „&“), „_“ oder „~“.¹⁷
- **accesskey_drei_punkte**: Strings mit Access-Key und drei Punkten. Beispiel: `&Open...` (*Audiobearbeitung*)
- **accesskey_tabulator_shortcut**: Strings mit Access-Key, Tabulator und Shortcut. Beispiel: `Cu&t\tCtrl+X` (*Audiobearbeitung*)
- **drei_punkte**: Strings mit drei Punkten („...“) mit folgender Unterteilung:
 1. **laufender_vorgang**: Drei Punkte, die einen laufenden Vorgang angeben. Beispiel: `Audacity is starting up...` (*Audiobearbeitung*)
 2. **menüpunkte**: Drei Punkte, die angeben, dass durch Klicken auf den betreffenden Menüpunkt ein Dialogfeld geöffnet wird. Beispiel: `Information...` (*Medienplayer*)
 3. **sonstiges**: Sonstige Verwendungsarten von drei Punkten. Beispiel: `Advanced settings. Use with care...` (*Medienplayer*)
- **html-code**: Strings mit HTML-Codes. Beispiel: `Video Output` (*Videobearbeitung*)
- **shortcut**: Strings mit Shortcut, der aus einer Taste oder auch aus einer Tastenkombination bestehen kann. Beispiel: `Exit Full Screen (Esc)` (*Browser*)
- **variable**: Strings mit Variable. Beispiel: `Right: Delete %d file(s)` (*Dateimanager*)¹⁸
- **zeilenumbruch**: Strings mit Zeilenumbruch. Beispiel: `Socket IO timeout.\n` (*E-Mail*)
- **zeilenumbruch_variable**: Strings mit Zeilenumbruch und Variable. Beispiel: `The selected action cannot be used in the compose window\nbecause it contains %f, %F, %as or %p.` (*E-Mail*)
- **nicht_zutreffend**: Im betreffenden String ist kein Steuerzeichen enthalten.

Anmerkungen zu dieser Liste sowie zur Annotation mit Steuerzeichen:

¹⁷In den Subkorpora *Bildbearbeitung*, *Browser* und *HTML-Editor* waren keine Access-Keys enthalten, im ersten Fall primär aufgrund des in das Korpus aufgenommenen Ausschnitts, in den anderen Fällen, weil die Access-Keys in den bei diesen Programmen verwendeten DTD-Dateien in separate Strings ausgelagert sind, die nicht in das Korpus aufgenommen wurden.

¹⁸Je nach Softwareprodukt werden unterschiedliche Variablentypen verwendet. Weitere Beispiele aus dem Korpus finden sich in Abschnitt 2.2.1.

- Wie bereits in Abschnitt 2.2 deutlich wurde, sind nicht alle technischen und textlichen Besonderheiten von GUI-Text auch Steuerzeichen im eigentlichen Sinn. Dennoch wurden alle zugehörigen Features im System STEUERZEICHEN zusammengefasst. Weitere Details zu Steuerzeichen mit Bezug zur Programmfunktionalität finden sich in Abschnitt 2.2.1, zu Steuerzeichen mit Bezug zur Formatierung in Abschnitt 2.2.2 und zu zeichenbasierten Konventionen in Abschnitt 2.2.3.
- Wie an der Liste der Features zu sehen ist, wurden nicht nur Features für die verschiedenen Typen von Strings definiert, sondern auch Features für Strings, die mehrere verschiedene Steuerzeichen aufweisen. Auf diese Weise war es möglich, diesen Aspekt im Korpus jeweils mit einer einzigen Annotation zu markieren (s. dazu auch die Auswertung in Abschnitt 5.6).
- Im Subkorpus *HTML-Editor* waren HTML-Codes entsprechend dem Zweck dieses Programms teils auf der Metaebene als Strings vorhanden, wurden also nicht zu Formatierung genutzt. Solche Vorkommen wurden nicht annotiert.
- Die Shortcuts wurden nur dann annotiert, wenn sie als Teil eines Menünamens genannt wurden. Die Angabe von Tasten bzw. Tastenkombinationen allein im String, wie z. B. in einer Fehlernachricht, wurde nicht annotiert.
- Wenn drei Punkte im Zusammenhang mit Gerunds vorkamen, wie bei *Applying...* (*Audiobearbeitung*) wurde ein laufender Vorgang angenommen, ansonsten wurden die drei Punkte als Menüpunkte annotiert. Ausnahme: *Wait...* (*Dateimanager*).
- Im Subkorpus *Editor* wurde der Access-Keys („&“) als HTML-Code codiert („&“). Diese Vorkommen wurden als Access-Keys annotiert, nicht als HTML-Code.

Tabulatoren und Escapezeichen waren zwar im Gesamtkorpus enthalten, aber nicht in den annotierten Strings. Daher werden sie in dieser Liste nicht mit aufgeführt, werden aber im Abschnitt 2.2 mit aufgeführt.

5.6 Auswertung der Korpusstudie

In diesem Abschnitt werden Informationen zur Auswertung der Korpusstudie gegeben. Diese unterteilt sich in eine grundlegende Auswertung (Abschnitt 5.6.1) sowie in Details zu den verschiedenen Kontextarten (Kategorie „Situativer Kontext“: 5.6.2, „Fachlicher Kontext“: 5.6.3, „Funktionaler Kontext“: 5.6.4). Danach folgen in Abschnitt 5.6.5 Informationen zum Inter-Annotator-

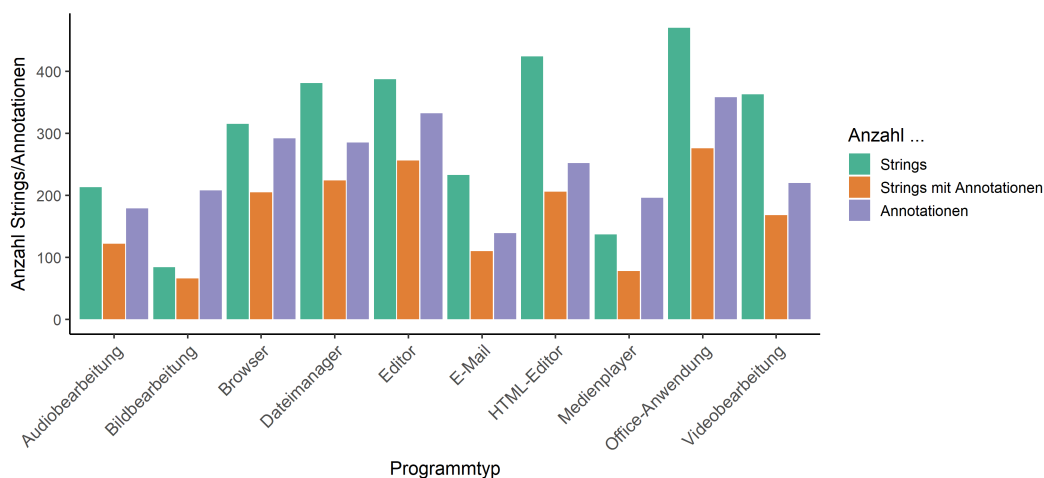


Abbildung 5.5: Strings und Annotationen nach Programmtyp

Agreement. Der daran anschließende Abschnitt (5.7) enthält die Interpretation der Ergebnisse auf Basis dieser Auswertung.

5.6.1 Grundlegende Auswertung

In Abbildung 5.5 ist als erster Schritt der Auswertung ein Überblick über die Anzahl der Annotationen im Korpus zu finden.

In dieser Abbildung sind die Anzahl an Strings (s. auch Abschnitt 5.3.2) sowie die Anzahl der Strings nach Programmtyp enthalten, die mindestens eine Annotation aufweisen. Zudem wird die Anzahl der Annotationen insgesamt nach Programmtyp angegeben, die von dieser Anzahl abweicht, da einzelne Strings auch mehrere Annotationen enthalten können. Die Anzahl an Strings liegt zwischen 84 (*Bildbearbeitung*) und 470 (*Office-Anwendung*), die Anzahl an annotierten Strings zwischen 66 (*Bildbearbeitung*) und 276 (*Office-Anwendung*). Der Median der annotierten Strings liegt bei 186,5, die Standardabweichung beträgt 74,0, der Interquartilsabstand beträgt 106,5. Die Anzahl der Annotationen liegt zwischen 139 (*E-Mail*) und 358 (*Office-Anwendung*). Der Median der Annotationen beträgt 236,0, die Standardabweichung 70,1 und der Interquartilsabstand 91,3.

Insgesamt weisen damit 56,9 Prozent der Strings mindestens eine Annotation auf, wobei der prozentuale Anteil von Strings mit Annotationen von 46,3 Prozent (*Videobearbeitung*) bis 78,6 Prozent (*Bildbearbeitung*) reicht, wie Abbildung 5.6 zu entnehmen ist. In dieser Abbildung ist auch der Prozentsatz der Annotationen gegenüber der Anzahl der Strings zu sehen, der deutlichere Unterschiede zeigt und von 247,6 Prozent (*Bildbearbeitung*) zu

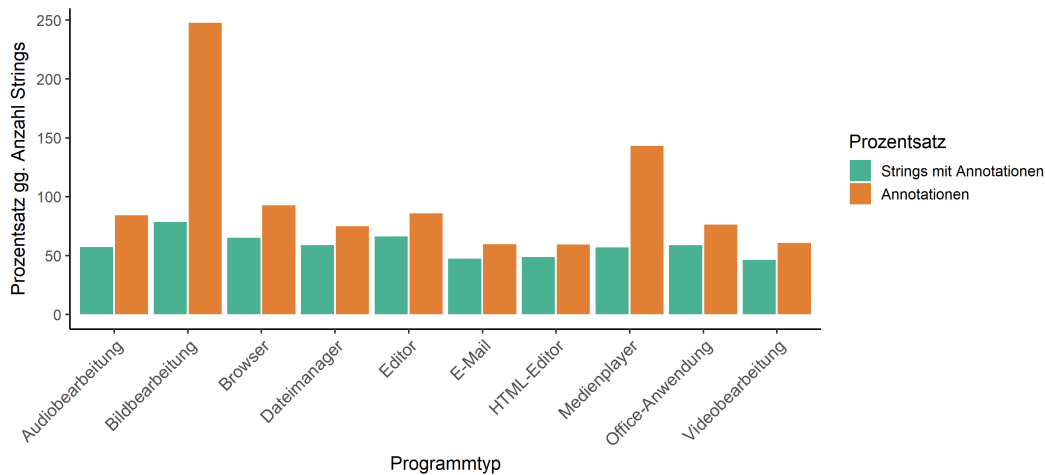


Abbildung 5.6: Prozentsatz der Strings mit Annotationen und der Annotationen gegenüber der Anzahl der Strings

59,4 Prozent (*HTML-Editor*) reicht.

Dabei besteht ein starker positiver linearer Zusammenhang zwischen der Anzahl der Strings und der Anzahl der Strings mit Annotationen (Korrelationskoeffizient: 0,94) und ein etwas weniger starker positiver linearer Zusammenhang zwischen der Anzahl der Strings mit Annotationen und der Anzahl der Annotationen (Korrelationskoeffizient: 0,89).

Aufgeschlüsselt nach Kontextarten ergibt sich die in Abbildung 5.7 zu sehende Verteilung. Insgesamt finden sich in der Kategorie „Situativer Kontext“ 811 Annotationen, in der Kategorie „Fachlicher Kontext“ 1.330 Annotationen und in der Kategorie „Funktionaler Kontext“ 320 Annotationen. Die Annotationen in der Kategorie „Situativer Kontext“ reichen von 7 (*Bildbearbeitung*) bis 180 (*Office-Anwendung*), die in der Kategorie „Fachlicher Kontext“ von 85 (*E-Mail*) bis 165 (*Editor*) und die in der Kategorie „Funktionaler Kontext“ von 22 (*Dateimanager*) bis 49 (*Editor*). Der Median in der Kategorie „Situativer Kontext“ liegt damit bei 78,5, die Standardabweichung beträgt 54,6 und der Interquartilsabstand beträgt 76,8. Für die Kategorie „Fachlicher Kontext“ liegt der Median bei 135,5, die Standardabweichung bei 25,4, der Interquartilsabstand bei 22. Für die Kategorie „Funktionaler Kontext“ beträgt der Median 32,5, die Standardabweichung 10,5 und der Interquartilsabstand 14,8.

Dabei zeigt sich zwischen der Anzahl der Strings und der Anzahl der Annotationen in der Kategorie „Situativer Kontext“ ein starker positiver linearer Zusammenhang (Korrelationskoeffizient: 0,89). Zwischen der Anzahl der Strings und der Anzahl der Annotationen der beiden anderen Kategorien

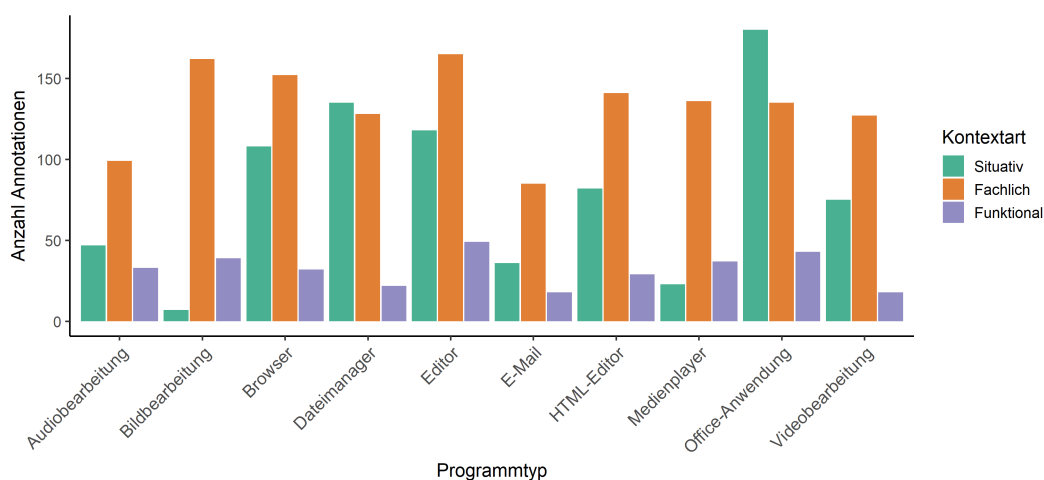


Abbildung 5.7: Annotationen nach Kontextarten

ist dagegen kein Zusammenhang vorhanden (Korrelationskoeffizient: 0,10 bei der Kategorie „Fachlicher Kontext“ und -0,02 bei der Kategorie „Funktionaler Kontext“).

Die prozentuale Verteilung der Kontextarten bei den einzelnen Programmtypen kann Abbildung 5.8 entnommen werden. Der Anteil der Annotationen der Kategorie „Situativer Kontext“ an der Gesamtmenge der Annotationen liegt bei 33,0 Prozent und bewegt sich bei den Subkorpora zwischen 3,4 Prozent (*Bildbearbeitung*) und 50,3 Prozent (*Office-Anwendung*), der Anteil der Annotationen der Kategorie „Fachlicher Kontext“ liegt insgesamt bei 54,0 Prozent und bewegt sich bei den Subkorpora zwischen 37,7 (*Office-Anwendung*) und 77,9 Prozent (*Bildbearbeitung*), der Anteil der Annotationen der Kategorie „Funktionaler Kontext“ liegt insgesamt bei 13,0 Prozent und bewegt sich bei den Subkorpora zwischen 7,7 (*Dateimanager*) und 18,9 Prozent (*Medienplayer*).

In Abbildung 5.9 ist die Länge der Strings mit Annotationen der verschiedenen Kontextarten für das Gesamtkorpus zu sehen, wobei die Einteilungen ein Wort, zwei Wörter, drei Wörter und länger als drei Wörter verwendet wurden. Im linken Diagramm sind die absoluten Werte zu sehen, rechts die prozentuale Verteilung. Für die Kategorie „Situativer Kontext“ fallen 36,5 Prozent und damit der größte Anteil auf Zwei-Wort-Strings. Den kleinsten Anteil weisen Strings mit vier und mehr Wörtern mit 15,4 Prozent auf. Die Kategorie „Fachlicher Kontext“ hat ihren größten Anteil mit 49,1 Prozent bei den Strings mit mindestens vier Wörtern, ihren kleinsten mit 8,7 Prozent bei den Ein-Wort-Strings, und die Kategorie „Funktionaler Kontext“ den größten mit 50,0 Prozent ebenfalls bei den Strings ab vier Wörtern und den kleinsten

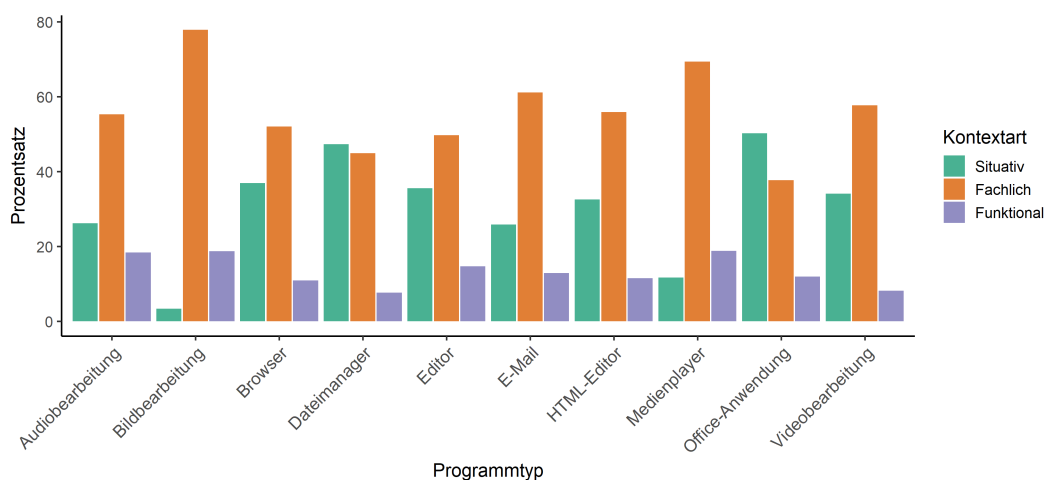


Abbildung 5.8: Annotationen nach Kontextarten anteilig

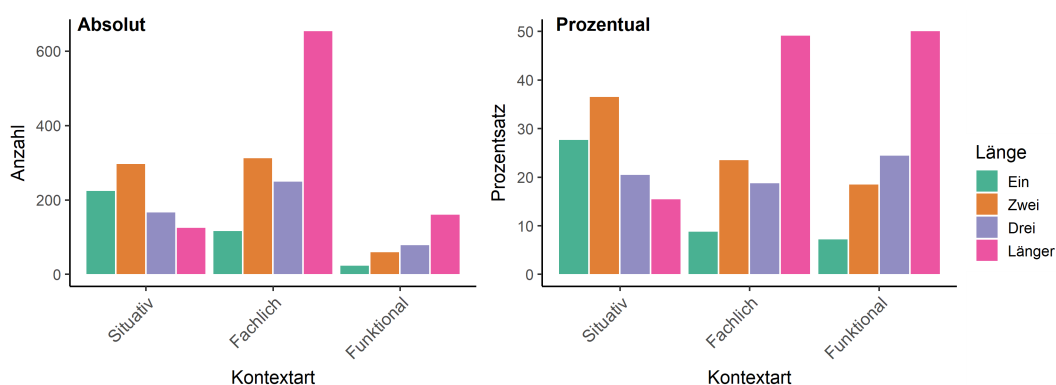


Abbildung 5.9: Länge der annotierten Strings

mit 7,2 Prozent bei den Ein-Wort-Strings. Dabei ist allerdings zu beachten, dass die letzte der Gruppen eine Sammelkategorie ist, in die alle Strings mit Annotationen fallen, die mehr als drei Wörter umfassen.

Für die Betrachtung der Steuerzeichen¹⁹ wurden die Subkorpora *Bildbearbeitung*, *Browser* und *HTML-Editor* nicht berücksichtigt, da sie keine Access-Keys enthalten (s. Abschnitt 5.5.4.2), was die Ergebnisse verzerren würde. Die zugehörigen Diagramme sind in Abbildung 5.10 enthalten. Die ersten beiden Diagramme enthalten die absolute und prozentuale Verteilung der Annotationen mit und ohne Steuerzeichen, bei der Kategorie „Situati-

¹⁹Wie bereits in Abschnitt 5.5.4.2 erläutert, umfasst dieses Feature neben den eigentlichen Steuerzeichen mit Bezug zur Programmfunktionalität und mit Bezug zur Formatierung auch zeichenbasierte Konventionen (s. Abschnitt 2.2.3).

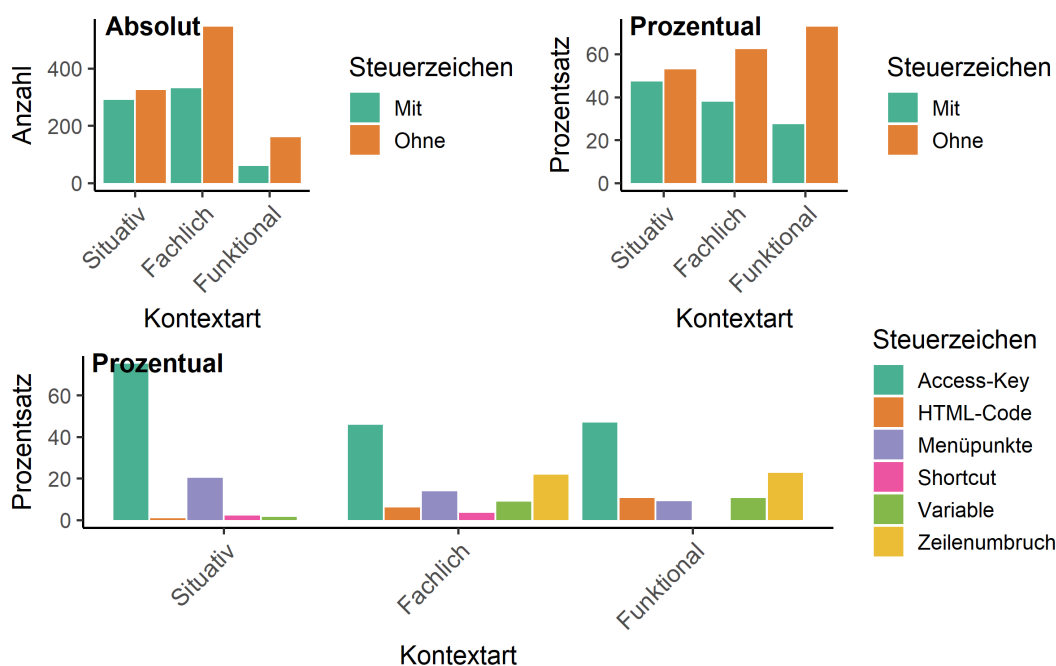


Abbildung 5.10: Steuerzeichen in annotierten Strings

ver Kontext“ 290 mit und 324 ohne (47,2 Prozent vs. 52,8 Prozent), bei der Kategorie „Fachlicher Kontext“ 330 mit und 545 ohne (37,7 Prozent vs. 62,3 Prozent) und bei der Kategorie „Funktionaler Kontext“ 60 mit und 160 ohne Steuerzeichen (27,3 Prozent vs. 72,7 Prozent).

Das dritte Diagramm in Abbildung 5.10 zeigt die prozentuale Verteilung auf verschiedene Typen von Steuerzeichen aufgeschlüsselt nach Kontextarten. Dabei wurden einzelne Features teils mehrfach gezählt:²⁰ Die Kategorie „Access-Keys“ bildet die Summe aus den Features `accesskey`, `accesskey_drei_punkte` und `accesskey_tabulator_shortcut`. Die Kategorie „Menüpunkte“ umfasst die Features `menüpunkte` und `accesskey_drei_punkte`, die Kategorie „Shortcuts“ die Features `shortcut` und `accesskey_tabulator_shortcut`. In der Kategorie „Variablen“ sind die Features `variable` und `zeilenumbruch_variable` und in der Kategorie „Zeilenumbruch“ die Features `zeilenumbruch` und `zeilenumbruch_variable` enthalten. Die Features `laufender_vorgang` und `sonstiges` wurden für dieses Diagramm nicht berücksichtigt, um den Fokus der Auswertung auf die

²⁰Grund dafür ist, dass im Subnetz `STEUERZEICHEN` eigene Features für Strings definiert wurden, die mehrere verschiedene Steuerzeichen aufweisen (s. Abschnitt 5.5.4.2). Entsprechend mussten bei dieser Auswertung beispielsweise alle Features gemeinsam erfasst werden, die einen Access-Key aufweisen.

Menüpunkte als Teilgruppe der Steuerzeichen mit drei Punkten zu setzen.

In diesem Diagramm ist zu sehen, dass bei 75,2 Prozent der Strings aus der Kategorie „Situativer Kontext“, die mindestens ein Steuerzeichen aufweisen, eines der Steuerzeichen ein Access-Key ist, und bei 20,4 Prozent finden sich Menüpunkte. Die anderen Steuerzeichen liegen zwischen 0 und 2,1 Prozent. Bei der Kategorie „Fachlicher Kontext“ ist es ebenfalls so, dass die meisten Strings mit Steuerzeichen einen Access-Key enthalten, nämlich 45,8 Prozent. Der Zeilenumbruch kommt mit 21,8 Prozent am zweithäufigsten vor. Die übrigen Steuerzeichen kommen in der Kategorie „Fachlicher Kontext“ in 3,5 bis 13,9 Prozent der Strings mit Steuerzeichen vor. Auch in der Kategorie „Funktionaler Kontext“ finden sich bei den Strings mit Steuerzeichen Access-Keys am häufigsten (47,0 Prozent), am zweithäufigsten kommen Zeilenumbrüche mit 22,7 Prozent vor. Die übrigen Steuerzeichen bewegen sich zwischen 0 und 10,6 Prozent.

In den folgenden Abschnitten werden die Features der einzelnen Kontextarten detaillierter betrachtet.

5.6.2 Auswertung der Features des situativen Kontexts

Abbildung 5.11 können Informationen zu den verschiedenen Untersystemen und Features der Kategorie „Situativer Kontext“ entnommen werden. Die ersten beiden Diagramme enthalten absolute und prozentuale Angaben zu den Untersystemen: 67,4 Prozent der Annotationen gehören dem Untersystem `PROBLEM_WORTART` an (547 Vorkommen), 23,8 Prozent dem Untersystem `PROBLEM_GUI-BEZUG` (193 Vorkommen) und 8,8 Prozent dem Untersystem `PROBLEM_FLEXIONSENDUNG` (71 Vorkommen).

Im Untersystem `PROBLEM_GUI-BEZUG`, das im dritten Diagramm zu sehen ist, wurde das Feature `über_präposition` in 37,8 Prozent der Fälle (73 Vorkommen) annotiert, das Feature `über_richtung` in 61,7 Prozent (119 Vorkommen) und das Feature `über_sonstiges` in 0,5 Prozent (1 Vorkommen).

Im Untersystem `PROBLEM_FLEXION` (s. viertes Diagramm) wurde das Feature `flexion_adj` in 77,5 Prozent der Fälle (55 Vorkommen) annotiert, das Feature `flexion_pronomen` in 14,1 Prozent (10 Vorkommen) und das Feature `flexion_partizip` in 8,5 Prozent (6 Vorkommen).

Im Untersystem `PROBLEM_WORTART` (s. fünftes Diagramm) wurde das Feature `verb.inf_oder_subst.sg` in 60,9 Prozent der Fälle (333 Vorkommen) annotiert und das Feature `verb.inf_oder_adj` in 15,0 Prozent (82 Vorkommen). Die übrigen Features weisen zwischen 1,1 Prozent (6 Vorkommen, `adj_oder_subst.sg`) und 7,7 Prozent (42 Vorkommen, `verb.inf_oder_past.part_oder_subst.sg`) auf.

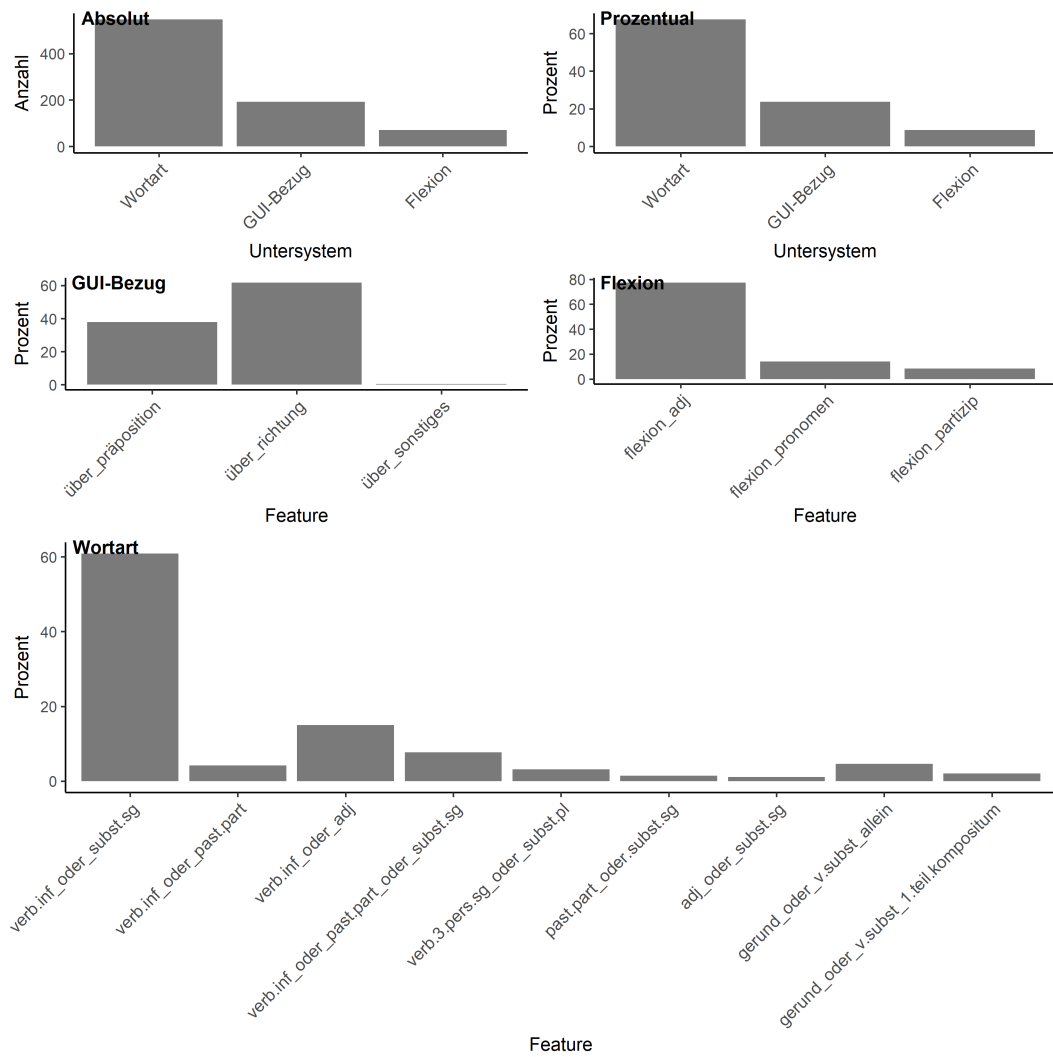


Abbildung 5.11: Verteilung der Features der Kategorie „Situativer Kontext“

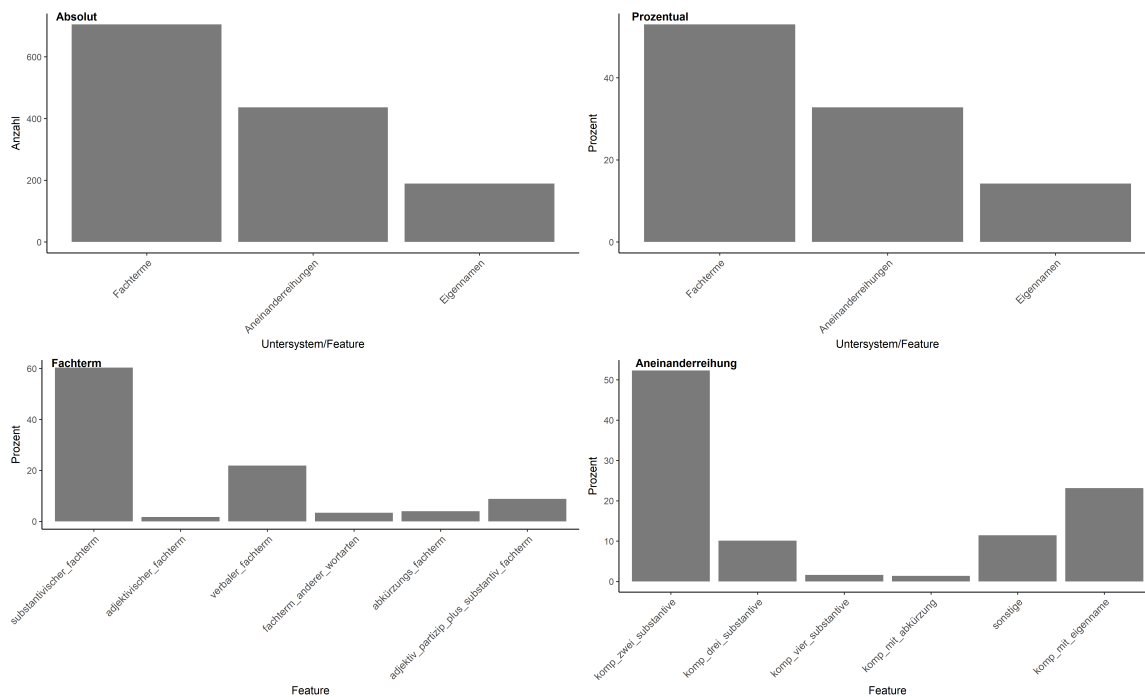


Abbildung 5.12: Verteilung der Features der Kategorie „Fachlicher Kontext“

5.6.3 Auswertung der Features des fachlichen Kontexts

In Abbildung 5.12 finden sich Informationen zu den verschiedenen Untersystemen und Features der Kategorie „Fachlicher Kontext“. Die ersten beiden Diagramme enthalten absolute und prozentuale Angaben zur Gruppe der fachterminologischen Features, die in 53,0 Prozent der Fälle annotiert wurden (705 Vorkommen), sowie des Features `eigenname`²¹, das in 14,2 Prozent der Fälle annotiert wurde (189 Vorkommen), und des Untersystems `FACHTERM_ANEINANDERREIHUNG`, dem 32,8 Prozent der Annotationen angehören (436 Vorkommen).

In der Gruppe der fachterminologischen Features, zu der im dritten Diagramm Details zu sehen sind, wurde das Feature `substantivischer_fachterm` in 60,3 Prozent der Fälle (425 Vorkommen) annotiert und das Feature `verbaler_fachterm` in 21,8 Prozent (154 Vorkommen). Die übrigen Features liegen zwischen 1,7 (12 Vorkommen, `adjektivischer_fachterm`) und 8,8 Prozent (62 Vorkommen,

²¹Bei der Annotation zeigte sich, dass es eine recht große Zahl an Eigennamen gab. Da diese sich von den anderen fachterminologischen Features, die Fachwörter verschiedener Wortarten sind, unterscheiden, wurden die Eigennamen bei dieser Auswertung separat aufgeführt.

adjektiv_partizip_plus_substantiv_fachterm).

Im Untersystem `FACHTERM_ANEINANDERREIHUNG` (s. viertes Diagramm) wurde das Feature `komp_zwei_substantive` in 52,3 Prozent der Fälle (228 Vorkommen) annotiert und das Feature `komp_mit_eigenname` in 23,2 Prozent (101 Vorkommen). Die übrigen Features weisen zwischen 1,4 Prozent (6 Vorkommen, `komp_mit_abkürzung`) und 11,5 Prozent (50 Vorkommen, `sonstige`) auf.

5.6.4 Auswertung der Features des funktionalen Kontexts

In Abbildung 5.13 sind Informationen zu den verschiedenen Untersystemen und Features der Kategorie „Funktionaler Kontext“ enthalten. Die ersten beiden Diagramme enthalten absolute und prozentuale Angaben zum Untersystem `UNKLARE_ANEINANDERREIHUNG`, dem 50,9 Prozent der Annotationen (163 Vorkommen) angehören. Daneben finden sich die Features `homonym_it-term` mit 22,2 Prozent (71 Vorkommen), `metaphorik` mit 6,9 Prozent (22 Vorkommen) und `unklare_bedeutung` mit 20,0 Prozent (64 Vorkommen).

Im Untersystem `UNKLARE_ANEINANDERREIHUNG`, das im dritten Diagramm zu sehen ist, wurde das Feature `drei_substantive` in 46,6 Prozent der Fälle (76 Vorkommen) annotiert und das Feature `drei_substantive_plus_adj_partizip` in 17,2 Prozent (28 Vorkommen). Die übrigen Features weisen zwischen 1,2 Prozent (2 Vorkommen, `vier_substantive_plus_adj_partizip`) und 14,7 Prozent (24 Vorkommen, `komposita_plus_und-oder`) auf.

5.6.5 Inter-Annotator-Agreement

Das für diese Arbeit erstellte Korpus wurde für eine Zweitannotation bereitgestellt, die im Rahmen einer Masterarbeit am Fachbereich 06 der Johannes Gutenberg-Universität Mainz angefertigt wurde (vgl. Hildebrand 2020). Mithilfe dieser Zweitannotation sollte ein Inter-Annotator-Agreement errechnet und damit die Tauglichkeit des Annotationsschemas festgestellt werden (vgl. Hirschmann 2019, S. 97; s. auch Abschnitt 5.1). Wie Lemnitzer und Zinsmeister vorschlagen, wurden dazu neben den Textproben auch sämtliche der in Abschnitt 5.5 aufgeführten Entscheidungsbäume und Anweisungen mit Beispielen weitergegeben, um eine möglichst hohe Übereinstimmung zwischen den beiden Annotationen zu erreichen (vgl. Lemnitzer und Zinsmeister 2015, S. 61).

Bei Vorliegen von zwei Annotationen kann ein Messwert namens Cohens Kappa berechnet werden, mit dem der Grad der Übereinstimmung zwischen

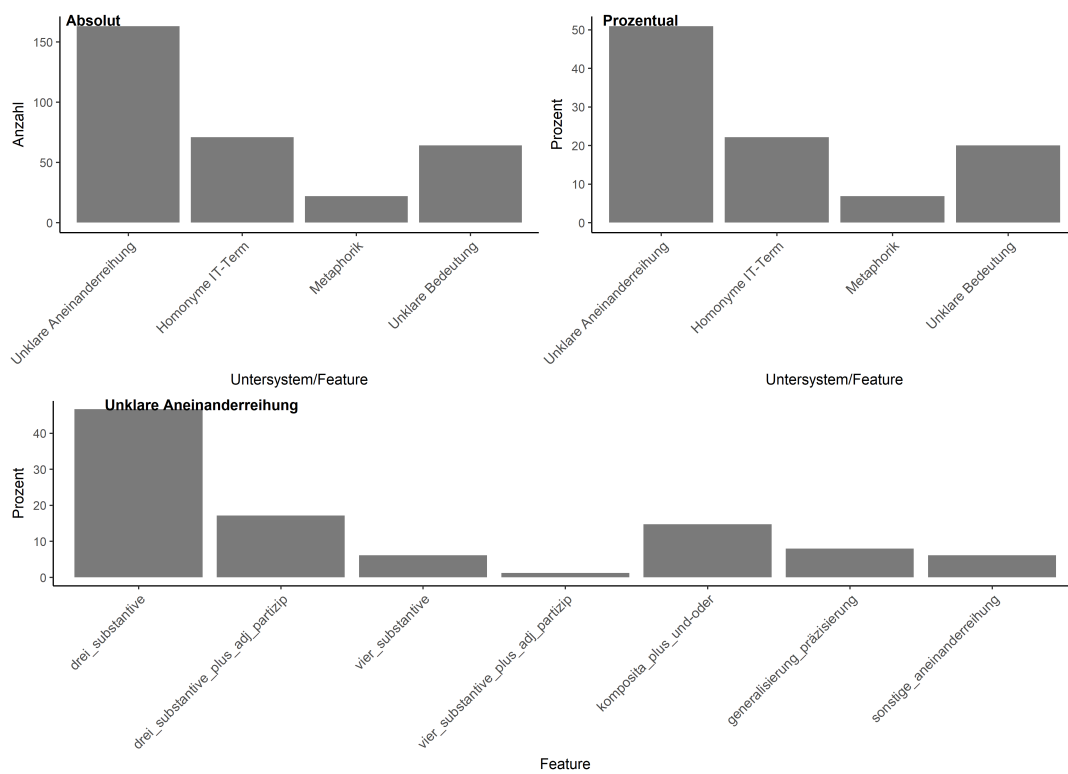


Abbildung 5.13: Verteilung der Features der Kategorie „Funktionaler Kontext“

zwei Annotationen ermittelt werden kann (vgl. Saldanha und O'Brien 2013, S. 106). Bei einer vollständigen Übereinstimmung der beiden Annotationen weist Cohens Kappa den Wert 1 auf, wenn es lediglich zufällige Übereinstimmungen gibt, weist Cohens Kappa den Wert 0 auf (vgl. Saldanha und O'Brien 2013, S. 106–107). Laut Saldanha und O'Brien wird eine Übereinstimmung von 50 Prozent als niedrig und eine von 95 Prozent als hoch angesehen (vgl. Saldanha und O'Brien 2013, S. 131).

Die Zweitannotation wurde allerdings nicht mit UAM CorpusTool erstellt, sondern in Microsoft Excel. Die dafür in Microsoft Excel genutzte Tabelle enthält das komplette Korpus, das so aufgeteilt ist, dass in jeder Zeile der Tabelle ein String enthalten ist. Bei der Zweitannotation wurde jeweils pro String überprüft, ob ein Feature einer der definierten Kontextarten vorhanden ist. Falls ja, wurde in die zutreffende Spalte von drei für diesen Zweck genutzten Spalten für die drei Kontextarten der Wert „1“ eingetragen, so dass am Ende die Anzahl von Strings mit Features der verschiedenen Kontextarten ermittelt werden konnte.

Gegenüber der Annotation in UAM CorpusTool ist diese Herangehensweise deutlich weniger detailliert: Statt ca. 10.000 Wörtern werden die ca. 3.000 Strings des Korpus (s. Abschnitt 5.3.2) betrachtet. Die durchschnittliche Stringlänge im Korpus liegt also bei etwa 3,3 Wörtern, wobei bei einzelnen Features ganze Strings annotiert wurden, bei anderen nur die jeweils relevanten Wörter (s. Abschnitt 5.5).²² Bei der Annotation im Rahmen der vorliegenden Arbeit weisen ca. 57 Prozent der Strings mindestens eine Annotation auf (s. Abschnitt 5.6.1), im Rahmen der Zweitannotation ca. 61 Prozent (vgl. Hildebrand 2020, S. 85).

Damit sich die beiden Annotationen vergleichen ließen, wurden die in UAM CorpusTool annotierten Daten zur Erstellung einer Tabelle mit Spalten genutzt, in die analog zur Zweitannotation pro String Angaben dazu eingefügt wurden, ob der betreffende String Annotationen einer der drei Kontextarten aufweist. Beim anschließenden Vergleich der beiden Annotationen wurden folgende Varianten überprüft:

- Für beide Annotationen wurden jeweils die Inhalte der drei Spalten für die verschiedenen Kontextarten verglichen, die jeweils den Wert „1“ enthielten, wenn im String (mindestens) eine Annotation der Kontextart vorhanden ist, und ansonsten den Wert „0“. Wenn beide Annotationen auf diese Weise verglichen werden, ergibt sich für die Kategorie „Situati-

²²Grob geschätzt wurden im Rahmen der vorliegenden Arbeit in der Kategorie „Situativer Kontext“ ca. 10 Prozent der Wörter annotiert, in der Kategorie „Fachlicher Kontext“ ca. 15 Prozent der Wörter und in der Kategorie „Funktionaler Kontext“ ca. 5 Prozent der Wörter. Weitere Informationen zu diesen Schätzungen finden sich in Abschnitt 5.7.

ver Kontext“ ein Cohens Kappa von 0,66, für die Kategorie „Fachlicher Kontext“ ein Cohens Kappa von 0,08 und für die Kategorie „Funktionaler Kontext“ ein Cohens Kappa von 0,11.

- In einer zweiten Variante wurde für jeden String ermittelt, ob und welche Kombination von Kontextarten dafür annotiert wurde, und daraus eine neue Spalte mit einem eindeutigen Wert für jede der möglichen sieben Kombination erstellt (Annotation einer einzigen Kontextart, Kombination von zwei der drei Kontextarten, Kombination der drei Kontextarten). Bei dieser Auswertung ergibt sich ein Cohens Kappa von 0,34.
- In einer dritten Variante wurde lediglich betrachtet, ob für einen String eine Annotation vorhanden ist oder nicht, also unabhängig von der jeweiligen Kontextart. Bei dieser Auswertung wurde ein Cohens Kappa von 0,61 ermittelt.

Mit den genannten Einschränkungen, dass diese Vergleiche nur auf Stringebene und auf Basis der Grobeinteilung in die drei Kontextarten erfolgten, lässt sich Folgendes daraus feststellen:

- In der ersten Auswertungsvariante ist die Übereinstimmung für die Kategorie „Situativer Kontext“ deutlich am höchsten. Dies deutet darauf hin, dass sich die darin subsummierten Features der Untersysteme `PROBLEM_WORTART`, `PROBLEM_GUI-BEZUG` und `PROBLEM_FLEXIONSENDUNG` am besten in Strings identifizieren lassen. Dagegen sind die Übereinstimmungen für die beiden anderen Kategorien deutlich niedriger. Für die Kategorie „Fachlicher Kontext“ scheint dies eher unerwartet zu sein, da die die Nutzung des Onlinewörterbuchs Merriam Webster einen sehr klaren Satz an Anweisungen darstellen müsste, der sich gut von verschiedenen Annotatorinnen nutzen lässt. Die geringe Übereinstimmung in der Kategorie „Funktionaler Kontext“ ist eher nachvollziehbar, weil diese Kategorie insbesondere mit den Features `homonym_it-term` und `unklare_bedeutung` eher schwer zu eingrenzende Annotationsbereiche umfasst.
- Kombiniert man die drei Kontextkategorien wie in Variante 2, zeigt sich ebenfalls ein niedriger Wert für die Übereinstimmung, weil die Wahrscheinlichkeit, dass mehrere Annotationen in einem String vollständig übereinstimmen, geringer ist als der Vergleich innerhalb der Kategorien in Variante 1.
- Aus Variante 3 könnte man herauslesen, dass es bei den beiden Annotationen insgesamt eine recht große Übereinstimmung dahingehend gab, welche der Strings ganz allgemein Kontextprobleme aufweisen.

Detailliertere Schlüsse aus diesen Ergebnissen ließen sich ziehen, wenn die beiden Annotationen genauer überprüft werden würden. Möglicherweise müssten dann auf dieser Basis die Anweisungen und Entscheidungsbäume für die Annotation optimiert werden. Hierfür könnte auch eine detailliertere Zweitannotation unter Berücksichtigung sämtlicher Features hilfreich sein.

Der nächste Abschnitt enthält die Interpretation der Ergebnisse dieser Auswertung.

5.7 Interpretation der Ergebnisse

Wie bereits in Abschnitt 5.3.2 aufgezeigt wurde, sind die Textproben im Hinblick darauf, wie viele Strings sie enthalten und welchen Anteil an der Gesamtmenge der Strings der genutzten Softwareprodukte sie abdecken, sehr divers, wobei diese Diversität für ausgewogene Korpora explizit gewünscht ist (vgl. Biber 1995, S. 131, und vgl. Lemnitzer und Zinsmeister 2015, S. 50). Dies zeigt sich sehr deutlich an der hohen Standardabweichung von 128,2 bei der Anzahl an Strings der verschiedenen Textproben und wird auch von den ersten Werten in Abschnitt 5.6.1 bestätigt, beispielsweise der ebenfalls hohen Standardabweichung von 74,0 bei der Anzahl der annotierten Strings und einer hohen Standardabweichung von 70,1 bei der Anzahl der Annotationen. Diese Punkte können in Summe als deutliche Hinweise darauf interpretiert werden, dass das gewählte Korpus sowohl divers als auch ausgewogen ist.

Wie ebenfalls bereits in Abschnitt 5.3.2 erläutert wurde, bedeutet die unterschiedliche Anzahl der Strings, dass in den Subkorpora mit weniger Strings mehr längere Strings und umgekehrt enthalten sind. Das Subkorpus *Bildbearbeitung*, das die wenigsten Strings aufweist, enthält beispielsweise ein paar einführende Informationen zum Programm selbst sowie überwiegend Informationen zu Neuerungen und Verbesserungen verschiedener Programmversionen, von denen viele in Satzform sind oder aus Listen mit längeren Listenpunkten bestehen. Das Subkorpus *Medienplayer* weist die zweitwenigsten Strings auf und enthält u. a. eine Gruppe von längeren Strings, in denen in ganzen Sätzen die Funktion einzelner anderer Strings aus der GUI erläutert werden. Zudem enthält es einen sehr langen String mit Informationen für den Zugriff auf verschiedene Themen der Hilfefunktion und einen weiteren sehr langen String mit einer Beschreibung der Befehlszeilenoptionen des Programms. Umgekehrt enthält das Subkorpus mit den meisten Strings, *Office-Anwendung*, ausschließlich sehr kurze Strings mit Texten für Menüs oder weitere Auswahlmöglichkeiten, von denen die überwiegende Anzahl drei und weniger Wörter umfasst. Da die Subkorpora alle dasselbe Volumen aufwei-

sen, kann der starke positive lineare Zusammenhang zwischen der Anzahl der Strings und der Anzahl der Strings mit Annotationen dahingehend interpretiert werden, dass kürzere Strings mehr kontextbezogene Übersetzungsprobleme mit sich bringen. Ebenso weist der starke positive lineare Zusammenhang zwischen der Anzahl an Strings mit Annotationen und der Anzahl an Annotationen darauf hin, dass kürzere Strings mehr Mehrfachannotationen aufweisen, wodurch sich ebenfalls die kontextbezogenen Übersetzungsprobleme einzelner Strings erhöhen.

Deutlich wird bei diesen ersten Auswertungen ebenfalls, dass sich Annotationen aus einer der drei identifizierten Kontextarten in über der Hälfte der Strings des Gesamtkorpus finden, im Extremfall des Subkorpus *Bildbearbeitung* sogar in fast vier Fünfteln aller Strings. Auffällig ist zudem, dass dies auch weitgehend bei den Subkorpora so ist: Trotz ihrer Diversität weisen bei lediglich drei Subkorpora weniger als die Hälfte der Strings Annotationen auf und beim Rest über 50 Prozent, mit der deutlichen Spitze beim Subkorpus *Bildbearbeitung*. Sowohl im Gesamtkorpus als auch bei den einzelnen Subkorpora zeugt dies von einer großen Relevanz der Kontextarten.

Des Weiteren fällt auf, dass gerade bei den beiden Subkorpora mit den wenigsten Strings die Anzahl der Annotationen deutlich höher ist als die Anzahl der Strings mit Annotationen. Dies wird insbesondere dann sichtbar, wenn man den Prozentsatz der Annotationen gegenüber der Anzahl von Strings mit Annotationen betrachtet: Außer bei *Bildbearbeitung* und *Medienplayer* liegen die Anteile bei allen Subkorpora teils deutlich unter 100 Prozent, bei *Bildbearbeitung* bei 247,6 Prozent und bei *Medienplayer* bei 143,1 Prozent. Diese beiden Subkorpora weisen auch den höchsten Anteil an Annotationen in der Kategorie „Fachlicher Kontext“ auf, der mit 77,9 Prozent (*Bildbearbeitung*) bzw. 69,4 Prozent (*Medienplayer*) deutlich über dem Durchschnitt von 54,0 Prozent für alle Subkorpora liegt. Auch hierfür liegt der Grund in den Inhalten dieser beiden Subkorpora: In längeren Strings kann mehr Fachterminologie enthalten sein und insbesondere im Subkorpus *Bildbearbeitung* weisen die Informationen zu Neuerungen und Verbesserungen zahlreiche Benennungen und Beschreibungen von Funktionen des Softwareprodukts auf, die in vielen Teilen aus Fachterminologie bestehen. Zudem enthalten sie auch einen relativ großen Anteil an komplexen Aneinanderreihungen und damit an Annotationen in der Kategorie „Funktionaler Kontext“, die teils gleichzeitig auch als längere Aneinanderreihungen aus dem Bereich Fachterminologie annotiert sind oder auch andere Annotationen aus dem Bereich Fachterminologie enthalten. Beispielstrings aus dem Subkorpus *Bildbearbeitung* mit hoher Annotationsdichte:

- Multi-threaded and hardware-accelerated rendering,

- processing and painting
- Metadata viewing and editing for Exif, XMP, IPTC, and DICOM
- Paste in place feature

Beispielstrings aus dem Subkorpus *Medienplayer*:

- Settings for the video, images or video+audio decoders and encoders.
- Settings for subtitle, teletext and CC decoders and encoders.
- Sout stream modules allow to build a sout processing chain.

Diese beiden Subkorpora sind damit die einzigen, bei denen der Anteil von Annotationen der Kategorie „Funktionaler Kontext“ höher als der Anteil von Annotationen der Kategorie „Situativer Kontext“ ist, die wiederum vermehrt bei kurzen Strings vorkommen, was im Folgenden noch weiter ausgeführt wird. Die speziellen Inhalte des Subkorpus *Bildbearbeitung* führen auch dazu, dass dieses Subkorpus mit 78,6 Prozent einen deutlich höheren Anteil an annotierten Strings als das Gesamtkorpus aufweist. Im Unterschied zum Programmtyp, der wie oben festgestellt keinen großen Einfluss auf die Annotationen hat, haben die konkreten Inhalte der gewählten Textproben nach diesen Ergebnissen sehr wohl einen deutlichen Einfluss auf die Annotationen.

Betrachtet man das Korpus als Ganzes, dann ergibt die Verteilung der Annotationen auf die drei Kontextarten ein deutliches Bild: Über die Hälfte der Annotationen betrifft Features aus der Kategorie „Fachlicher Kontext“. Die 1.330 Annotationen Kategorie der „Fachlicher Kontext“ auf 10.000 Wörter bedeuten, dass vermutlich sogar mehr als 15 Prozent der Wörter des Korpus einen fachtextlichen Hintergrund haben, wenn man die Vorkommen der annotierten Fachtermini und Eigennamen (894 Vorkommen) sowie die jeweils mehreren Wörter der annotierten fachterminologischen Aneinanderreihungen (allein 456 Wörter bei 228 Vorkommen von Komposita aus zwei Substantiven) addiert. Diese hohen Werte sind auch im Hinblick darauf interessant, dass acht der verwendeten Textproben Anwendungsgebiete abdecken, die für ein eher breiteres Publikum gedacht sind (s. Abschnitt 5.3) und gleichzeitig die Fachterminologie, die Eingang in die Gemeinsprache gefunden hat, über die vorhandenen Definitionen im Onlinewörterbuch Merriam Webster eliminiert wurde.

Die beiden Subkorpora mit Unterstützungsprogrammen, *Editor* und *HTML-Editor*, bilden im Hinblick auf die prozentuale Verteilung der Kontextarten keine Ausnahme, sondern bewegen sich im Rahmen der anderen

Subkorpora. Es fällt lediglich auf, dass das Subkorpus *Editor* die größte Zahl an Annotationen aus der Kategorie „Fachlicher Kontext“ aufweist, wobei der Unterschied mit 165 Vorkommen gegenüber den 127 bis 162 Vorkommen von weiteren sieben Subkorpora, darunter 141 Vorkommen beim Subkorpus *HTML-Editor*, nicht sehr groß ist. Dies zeigt sich auch darin, dass die Standardabweichung für die Subkorpora bei der Kategorie „Fachlicher Kontext“ mit 25,4 im Vergleich zum Median (135,5 Annotationen) sehr niedrig ist, insbesondere wenn man diese Werte mit den beiden anderen Kategorien vergleicht, bei denen die Standardabweichungen bei deutlich niedrigeren Medianen in beiden Fällen höher sind. Annotationen aus der Kategorie „Fachlicher Kontext“ sind damit deutlich gleichmäßiger über die Subkorpora verteilt als aus den anderen Kategorien, bei denen es eine deutlich höhere Streuung gibt. Aus diesen Beobachtungen lässt sich schließen, dass sich der Typ des Softwareprodukts im Korpus nicht sehr stark auf die Anzahl der fachterminologischen Annotationen auswirkt. Ergänzend könnte man sagen, dass sämtliche Subkorpora eine erhöhte Fachsprachlichkeit aufweisen, wenn man die Anzahl der Annotationen aus der Kategorie „Fachlicher Kontext“ als Indikator für die Fachsprachlichkeit nimmt.

Die nächste Kategorie, „Situativer Kontext“, weist einen Anteil von knapp einem Drittel der Gesamtzahl der Annotationen auf. Dies macht immer noch 811 Annotationen und damit ebenfalls einen Anteil aus, der in Richtung 10 Prozent der untersuchten Wörter geht, wenn man berücksichtigt, dass davon in vielen Fällen ganze Strings betroffen sind, die in vielen Fällen mehr als ein Wort umfassen, wobei sie jedoch häufig nicht sehr lange sind (s. u.). Dies zeugt von der Relevanz der darunter subsummierten Problematiken, auch wenn diese teils auf kleine Gruppen von Problemwörtern zurückzuführen sind, wie weiter unten noch ausgeführt wird. Zudem bestätigt der starke positive lineare Zusammenhang zwischen der Anzahl der Strings und der Anzahl der Annotationen in der Kategorie „Situativer Kontext“, dass die in dieser Kategorie zusammengefassten Problematiken insbesondere bei kurzen Strings auftreten.

Der übrige Teil der Annotationen mit 13 Prozent und 320 Vorkommen betrifft Annotationen aus der Kategorie „Funktionaler Kontext“, die ebenfalls, u. a. durch unklare Aneinanderreihungen (z. B. 76 Vorkommen von Komposita mit drei Wörtern, also allein schon 228 Wörter), ein Volumen von mehr als 5 Prozent der Wörter betreffen. Grob geschätzt wurden demnach 30 Prozent des Volumens des untersuchten Korpus in Wörtern für mindestens eine der Kontextarten annotiert, auch wenn es zwischen bestimmten Teilen der Aneinanderreihungen aus den Kategorien „Fachlicher Kontext“ und „Funktionaler Kontext“ Überschneidungen gibt. Über diese Vorgehensweise, das Volumen der annotierten Wörter zu schätzen, lässt sich also ebenfalls fest-

stellen, dass die Kontextproblematik generell ein im Korpus sehr relevantes Phänomen darstellt.

Betrachtet man die Anteile der drei Kontextarten bei den Subkorpora, so zeigt sich, dass bei sechs Subkorpora der Anteil der Annotationen aus der Kategorie „Fachlicher Kontext“ am höchsten ist, gefolgt von Annotationen der Kategorie „Situativer Kontext“ und dann der Kategorie „Funktionaler Kontext“, was auch grob der Verteilung der Kontextarten im Gesamtkorpus entspricht. Die erste Ausnahme bilden wie erwähnt die genannten Subkorpora *Bildbearbeitung* und *Medienplayer*. Neben der Tatsache, dass bei diesen Subkorpora der zweitgrößte Anteil der von Annotationen der Kategorie „Funktionaler Kontext“ ist, weisen sie auch mit 3,4 Prozent bzw. 11,7 Prozent den mit Abstand niedrigsten Anteil von Annotationen der Kategorie „Situativer Kontext“ auf. Die zweite Ausnahme bilden die Subkorpora *Dateimanager* und *Office-Anwendung*, bei denen der Anteil von Annotationen der Kategorie „Situativer Kontext“ am höchsten ist, gefolgt von Annotationen der Kategorie „Fachlicher Kontext“. Für das Subkorporum *Office-Anwendung* kann der Grund dafür in der großen Anzahl an sehr kurzen Strings gesehen werden, die potenziell mehr Annotationen der Kategorie „Situativer Kontext“ mit sich bringen (dazu im Folgenden mehr). Dieser Sachverhalt wird auch durch den bereits genannten starken positiven linearen Zusammenhang zwischen der Anzahl Strings und der Anzahl an Annotationen der Kategorie „Situativer Kontext“ bestätigt. Für das Subkorporum *Dateimanager* bietet sich allerdings keine Begründung für die Besonderheit an, dass es darin mehr Annotationen der Kategorie „Situativer Kontext“ als der Kategorie „Fachlicher Kontext“ gibt.

Bei der Verteilung der Annotationen auf Strings mit unterschiedlicher Länge zeigt sich, dass insbesondere bei der Kategorie „Situativer Kontext“ die deutliche Mehrzahl der Annotationen in den Ein- bis Drei-Wort-Strings enthalten sind, nämlich 84,6 Prozent, was auch dem hohen Korrelationskoeffizienten bei dieser Kategorie entspricht. Bei der Kategorie „Fachlicher Kontext“ sind dies nur 50,9 Prozent und bei der Kategorie „Funktionaler Kontext“ nur 50,0 Prozent. In beiden letzteren Kategorien spielt dabei eine Rolle, dass Aneinanderreihungen primär in längeren Strings vorkommen können, da sich beispielsweise das Feature `vier_substantive` aus der Kategorie „Funktionaler Kontext“ nur in Strings findet, die mindestens vier Wörter umfassen. Dagegen gehen die Phänomene der Kategorie „Situativer Kontext“ eher mit kurzen Strings einher, die keine vollständigen Sätze darstellen, beispielsweise wenn die Wortart unklar ist oder das Bezugswort eines Adjektivs oder Pronomens fehlt. Beispiele aus dem Subkorporum *Medienplayer*: Kurzer String mit ambiger Wortart beim Wort „control“ (Feature `verb_inf_oder_subst_sg` annotiert): `Control interfaces`. Längerer String, bei dem diese Ambigui-

tät aufgelöst ist: `Settings for VLC's control interfaces` („control“ ist der erste Teil des Kompositums „control interfaces“, also ein Substantiv).

Auch bei den Steuerzeichen, die allerdings nur bei einem Subset des Korpus betrachtet wurden, zeigt sich ein ähnliches Bild: Bei der Kategorie „Situativer Kontext“ enthält etwas weniger als die Hälfte der Strings Steuerzeichen, bei der Kategorie „Fachlicher Kontext“ weisen nur noch 37,7 Prozent der Strings Steuerzeichen auf und bei der Kategorie „Funktionaler Kontext“ lediglich 27,3 Prozent. Dies hat mit großer Wahrscheinlichkeit ebenfalls mit der Länge der betreffenden Strings zu tun. Hier könnte allerdings bei der Kategorie „Funktionaler Kontext“ noch hinzukommen, dass Strings mit unklarer Bedeutung in Menüs und zugänglicheren Bereichen der GUI, bei denen bestimmte Steuerzeichen gängiger sind, idealerweise eher vermieden werden, sondern vielmehr in Nachrichten usw. vorkommen, die weniger Steuerzeichen aufweisen. Auf Basis der vorliegenden Daten kann dies aber nicht abschließend geklärt werden.

Bei allen Kontextarten sind Access-Keys als Steuerzeichen anteilig am häufigsten vorhanden, im Fall des situativen Kontexts sogar deutlich überwiegend. Access-Keys, Menüpunkte und Shortcuts lassen sich in der Regel Menüoptionen und auch Inhalten von Dialogfeldern zuordnen (vgl. beispielsweise Beste 2006, S. 42, 64), HTML-Codes, Variablen und Zeilenumbrüche eher längeren Strings wie Nachrichten (vgl. beispielsweise Esselink 2000, S. 63, 68–69). Daraus lässt sich ableiten, dass die Strings der Kategorie „Situativer Kontext“, die Steuerzeichen aufweisen, weit überwiegend aus Menüs und Dialogfeldern stammen (97,7 Prozent). Die Kategorie „Fachlicher Kontext“ zeigt mit 63,2 Prozent Annotationen aus dem Bereich der Steuerzeichen, die sich Menüs und Dialogfeldern zuordnen lassen, ebenfalls einen hohen Anteil, weist aber auch 21,8 Prozent Strings mit Zeilenumbrüchen, also längere und eher den Nachrichten zugehörige Strings auf. Bei der Kategorie „Funktionaler Kontext“ befinden sich nur 56,1 Prozent der Annotationen in Strings mit Access-Keys, Menüpunkten oder Shortcuts. Auch hier gibt es vermehrt Zeilenumbrüche, aber auch HTML-Codes und Variablen, die eher in Richtung Nachrichten u.ä. deuten.

Für die Kategorie „Situativer Kontext“ lässt sich aus Basis dieser Interpretationen schließen, dass die darin subsummierten Kontextprobleme im Korpus überwiegend in kurzen Strings vorkommen, die vermehrt in Menüs und Dialogfeldern angeordnet sind. Dies entspricht beispielsweise dem, was Beste in Bezug auf die „universell einsetzbaren Wortstämme“ in kurzen Strings sagt, die in vielen Fällen zu Verständnis- und damit Übersetzungsproblemen führen (vgl. Beste 2006, S. 73), passt aber auch zu seinen Aussagen zur Kürze und zum Vorhandensein von Steuerzeichen von beispielsweise Menüoptionen (vgl. beispielsweise Beste 2006, S. 40, 64).

Wenn man die Kategorie „Situativer Kontext“ detailliert betrachtet, lässt sich feststellen, dass die Problematik ambiger Wortarten, auf die Beste in Zusammenhang mit seiner eben genannten Äußerung primär abzielt, klar am häufigsten im Korpus vorkommt. In der Literatur werden allerdings lediglich die beiden Untergruppen `verb.inf_oder_subst.sg` und `verb.inf_oder_adj` als Beispiele für diese Problematik genannt (vgl. beispielsweise Beste 2006, S. 73, Ottmann 2005, S. 103, 106, Esselink 2000, S. 63, Wahle 2000b, S. 44, oder K. J. Dunne 2009, S. 203; siehe dazu auch Abschnitt 4.6.1). Diese beiden Varianten sind auch die Varianten ambiger Wortarten, die im Korpus am häufigsten enthalten sind, wobei zu ersterer Gruppe letztlich sogar noch mehr Vorkommen gehören, da das Feature `verb.3.pers.sg_oder_subst.pl` damit verwandt ist, weil es dieselben Wörter wie das Feature `verb.inf_oder_subst.sg` umfasst. Insgesamt kommen diese drei Features auf einen Anteil von 79,0 Prozent der Features für ambige Wortarten. Bei diesen Features wurden zahlreiche verschiedene Wörter annotiert, weil es im Englischen, wie auch beispielsweise Albrecht betont (vgl. Albrecht 2013, S. 107), sehr viele Wörter gibt, bei denen das Verb im Infinitiv und das Substantiv im Singular – und damit auch das Verb in der 3. Person Singular und das Substantiv im Plural – gleich lauten (wie „the copy, to copy, copies“, „the match, to match, matches“), wohingegen sich die 82 Vorkommen des Features `verb.inf_oder_adj` auf sieben verschiedene Wörter verteilen: „open“, „clear“, „correct“, „duplicate“, „empty“, „forward“ und „reverse“. In den drei miteinander verwandten Features `verb.inf_oder_past.part`, `verb.inf_oder_past.part_oder_subst.sg` und `past.part_oder_subst.sg`, die nur unregelmäßige Verben bzw. unregelmäßige Verben mit gleichlautendem Substantiv betreffen, verteilen sich die 73 Vorkommen auf insgesamt zehn Wörter: „autoinput“, „cut“, „input“, „output“, „quit“, „read“, „reset“, „run“, „set“ und „split“. Es zeigt sich also, dass ein gewisser, wenn auch im Korpus wenig relevanter Teil der Problematik ambiger Wortarten in der Literatur zur Softwarelokalisierung, die für die vorliegende Arbeit rezipiert wurde, nicht explizit genannt wird, dieser aber ebenfalls Beachtung verdient.

Die Probleme, die aus dem Bezug von Strings auf andere GUI-Elemente entstehen, kommen zwar nur in etwas weniger als einem Viertel der Fälle aus der Kategorie „Situativer Kontext“ vor, aber dennoch bilden die Features `über_präposition` und `über_richtung` eine zahlenmäßig relevante Untergruppe des Korpus. Dass in lediglich zwei Veröffentlichungen (vgl. Anhang 1 in Melby und Foster 2010, S. 1, und K. J. Dunne 2006b, S. 103–104), die für die vorliegende Arbeit rezipiert wurden, die Problematik mit Präpositionen genannt wird und dies auch nur anhand desselben Beispiels und ohne Blick auf längere Strings als Ein-Wort-Strings mit lediglich einer Präposition, ist

daher durchaus verwunderlich. Die Probleme mit deiktischen Ausdrücken, die aus der Multimodalität von GUIs entstehen, bilden mit einem Anteil von 61,7 Prozent an den Problemen mit dem Bezug von Strings auf andere GUI-Elemente und 119 Vorkommen im Korpus sogar die deutlich größte Teilgruppe der Probleme mit GUI-Bezug, werden aber dennoch in der für die Arbeit rezipierten Literatur zur Softwarelokalisierung nicht thematisiert, wohingegen verschiedene Arten von Deixis ein typisches Thema der Linguistik bzw. Pragmatik sind (vgl. beispielsweise Finkbeiner 2015, S. 34–41).

Verschiedene Varianten der Flexionsproblematik zeigen sich im Korpus insgesamt als am wenigsten relevant, wobei das Feature `flexion_adj` mit drei Vierteln dieser Fälle die nennenswerteste Untergruppe ist. Hier zeigt sich ähnlich wie beim überwiegenden Teil der anderen Problembereiche aus der Kategorie „Situativer Kontext“, dass dieser Problembereich in der Literatur zwar angesprochen wird (vgl. beispielsweise Beste 2006, S. 73, Esselink 2000, S. 63, 75–76, Wahle 2000b, S. 44–45, Reineke 2005, S. 84, oder K. J. Dunne 2009, S. 209), allerdings auch nur in Teilen, nämlich mit Ausnahme von Beste (vgl. Beste 2006, S. 73) mit dem Fokus auf Ein-Wort-Strings. Im Korpus finden sich aber auch diverse Beispiele von Mehrwort-Strings mit allein stehendem Adjektiv oder Pronomen am Ende oder sogar in der Mitte.

Allen Untergruppen der Kategorie „Situativer Kontext“ ist gemein, dass sie sich relativ einfach und mit großer Zuverlässigkeit maschinell identifizieren lassen, sei es durch konkrete Wortlisten (unregelmäßige Verben, Termini, die mit verschiedenen Wortarten in der Termbank hinterlegt sind, Richtungswörter) oder die Stellung im String (Präposition am Anfang oder Ende, Adjektiv, Pronomen oder Partizip am Ende), um sie so vorab für besondere Beachtung bei der Übersetzung zu markieren. Zudem bestätigt sich die Notwendigkeit, für die Übersetzung relevantes Referenzmaterial bereitzustellen, um die Probleme dieser Kontextart beim Übersetzen lösen zu können (s. Abschnitt 4.5), wie beispielsweise eine lauffähige Version der zu übersetzenden Software (vgl. Ottmann 2005, S. 102, und Esselink 2000, S. 53), Dokumentation des Softwareprodukts (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, und Wahle 2000b, S. 45) sowie Kommentare der Entwicklerinnen und weitere Informationen in den Ressourcendateien (vgl. Esselink 2000, S. 34, 64–65, Wahle 2000b, S. 37–38, 40, 45, Reineke 2005, S. 85, und Sachse 2005, S. 147–148, 158).

Bei der Kategorie „Fachlicher Kontext“ sind die Fachterme der verschiedenen Wortarten die deutlich größte Gruppe, deren Hauptfokus wiederum auf substantivischen Fachtermini liegt. Diese bilden aber auch eine zentrale Grundlage für viele der fachterminologischen Aneinanderreihungen, die in

erster Linie aus Substantiven gebildet werden.²³ Die Eigennamen spielen in der Gesamtkategorie zwar eine untergeordnete Rolle, kommen aber auch in Aneinanderreihungen in größerer Frequenz vor. In Bezug auf Lexik bestätigt sich im Korpus damit Bestes Einschätzung, der GUI-Text „als technische Fachtexte mit einem hohen Grad an Fachsprachlichkeit“ (Beste 2006, S. 43) klassifiziert. Gerade auch vor dem Hintergrund des großen Anteils von Annotationen der Kategorie „Fachlicher Kontext“ im Korpus wird zudem deutlich, wie wichtig bei der Lokalisierung eine gute Terminologiarbeit und die Nutzung umfassender Terminologiedatenbanken ist, die auch Eigennamen enthalten (s. Abschnitt 4.5). Darauf weisen beispielsweise Ottmann und Esselink explizit hin (vgl. Ottmann 2005, S. 104, und Esselink 2000, S. 53, 401–402).

Die anteilsmäßig kleinste Kategorie, „Funktionaler Kontext“, hat ihren deutlichen Schwerpunkt mit über der Hälfte der Fälle bei unklaren Aneinanderreihungen, wobei nachvollziehbar ist, dass längere und komplexere Komposita seltener als kürzere und weniger komplexe vorkommen. Einleuchtend ist ebenso, dass Fälle von Metaphorik die klar kleinste Gruppe darstellen, da es sich dabei häufig um Benennungen für neue und spezielle Funktionen handelt, bei denen davon ausgegangen werden kann, dass es nur wenige gibt. Fälle von Homonymie bei der IT-Fachterminologie und von unklarer Bedeutung liegen beide bei 20 Prozent der Annotationen der Kategorie „Funktionaler Kontext“ bzw. etwas darüber, bringen in der Praxis aber sehr große übersetzerische Probleme mit sich. Für das Korpus lässt sich für erstere vielleicht nicht unbedingt stolze Meinung, dass sie in Fachtexten recht verbreitet sind (vgl. Stolze 2013, S. 146), bestätigen. Allerdings wurden im Korpus die Vorkommen homonymer Wörter nicht annotiert, deren Begriff im sprachlichen Kontext des Strings klar erkennbar war (s. die Vorgaben im Abschnitt 5.5.3.2), wo also analog zu Albrecht eine korrekte Interpretation aufgrund des sprachlichen Kontexts möglich war (vgl. Albrecht 2013, S. 217). Annotiert wurden im Korpus nur mehrdeutige Vorkommen von einer begrenzten Anzahl an Homonymen, nämlich „check“, „control“, „image“, „line“, „offset“ und „tab“. Im Korpus scheint dieses Problem also sehr eingeschränkt vorhanden

²³Von Liang gibt es eine Korpusuntersuchung aus dem Jahr 1985 zur deutschen Fachsprache der EDV (vgl. Liang 1985), aus der Wichter zitiert (vgl. Wichter 1997, S. 1179). Da diese allerdings mit deutschsprachigen Texten durchgeführt wurde und älteren Datums ist, wurde sie für die vorliegende Arbeit nicht rezipiert. Ähnlich verhält es sich mit einer Studie von Chang, der anhand eines Korpus mit Texten aus der deutschsprachigen Computerzeitschrift c't Anglizismen in der deutschen Fachsprache der Computertechnik untersucht (vgl. Chang 2005). Ebenso nicht berücksichtigt wurde eine Korpusanalyse von Astapenko, die ebenfalls deutschsprachige Texte aus GUI und Dokumentation über Lemmatisierung und Wortartenbestimmung analysiert, wobei beide Textsorten gemeinsam betrachtet werden (vgl. Astapenko 2008).

zu sein.

Vielleicht lässt sich sagen, dass die Features der Kategorie „Funktionaler Kontext“ überwiegend – mit Ausnahme der Metaphorik – die Qualität des Ausgangstexts betreffen. Wenn beispielsweise alle Strings klar und verständlich formuliert, Homonyme in der IT-Fachterminologie eliminiert oder nur eindeutig verwendet und unklare Aneinanderreihungen vermieden werden, ist die Qualität des Ausgangstexts höher. Wenn man dies tatsächlich als einen möglichen Indikator für Textqualität bzw. für einen Teilaspekt der Textqualität nehmen kann, wären das Subkorpus *Dateimanager* mit einem Anteil von 7,7 Prozent Annotationen der Kategorie „Funktionaler Kontext“ gefolgt vom Subkorpus *Videobearbeitung* mit einem Anteil von 8,2 Prozent Annotationen der Kategorie „Funktionaler Kontext“ die Subkorpora mit der höchsten Qualität im Hinblick auf diese Aspekte. Die mit der niedrigsten Qualität wären *Audiobearbeitung*, *Bildbearbeitung* und *Medienplayer* mit je über 18 Prozent, wobei sich wie oben schon festgestellt auch die konkreten Inhalte der Textprobe auf die Anzahl der Annotationen der verschiedenen Kontextarten auswirken. Ebenso könnte die Feststellung, dass nur etwas mehr als ein Viertel der Strings in der Kategorie „Funktionaler Kontext“ Steuerzeichen enthält und davon wiederum nur etwas mehr als die Hälfte aus der Gruppe der Steuerzeichen stammt, die vermehrt bei Menüs und Dialogfeldern vorkommen, ein Zeichen dafür sein, dass zumindest die GUI-Bereiche, die relevanter für die alltägliche Arbeit sind, im Hinblick auf die Features der Kategorie „Funktionaler Kontext“ eine höhere Qualität aufweisen, weil ein Großteil der Probleme der Kategorie „Funktionaler Kontext“ in Strings wie Nachrichten vorkommen, die für die wesentlichen Aspekte der Arbeit in einem Softwareprodukt nicht zentral sind.

Zuletzt gilt auch für diese Kategorie, dass das in Abschnitt 4.5 genannte Referenzmaterial für die Übersetzung benötigt wird, um die Probleme dieser Kontextart lösen zu können. Auch hier wären das primär eine lauffähige Version der zu übersetzenden Software (vgl. Ottmann 2005, S. 102, und Esselink 2000, S. 53), Dokumentation des Softwareprodukts (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, und Wahle 2000b, S. 45) sowie Kommentare der Entwicklerinnen und weitere Informationen in den Ressourcendateien (vgl. Esselink 2000, S. 34, 64–65, Wahle 2000b, S. 37–38, 40, 45, Reineke 2005, S. 85, und Sachse 2005, S. 147–148, 158).

5.8 Einschränkungen, offene Fragen und weitere Vorgehensweise

Wie sich im Verlauf der Annotation des Korpus zeigte, wies die gewählte Methode Nachteile auf, die bei zukünftigen Untersuchungen vermieden werden sollten: Zum einen enthalten verschiedene Subkorpora Wiederholungen derselben Strings (teils mit und teils ohne oder mit anderen Steuerzeichen, aber mit denselben Wörtern). Es könnte sinnvoll sein, vorab solche Wiederholungen zu entfernen, um möglichst viele unterschiedliche Strings in das Korpus aufnehmen zu können. Zum anderen wäre es sinnvoller, anstatt die Textproben wie in der vorliegenden Studie eher willkürlich auszuwählen, zunächst die Gesamtmenge der Strings eines Softwareprodukts zu erfassen und aus dieser Menge dann per Zufallsverfahren die gewünschte Anzahl von Strings bzw. von Wörtern zu wählen. Dies würde einen klareren Blick auf die durchschnittlichen Inhalte der Strings des jeweiligen Softwareprodukts ermöglichen.

In der vorliegenden Untersuchung lag der Fokus auf Programmen der drei Desktop-Betriebssysteme Windows, Linux und Mac OS. Damit wurde der immer wichtiger werdende Bereich der mobilen Betriebssysteme, wie Android oder iOS, komplett außen vor gelassen. Es ist zu erwarten, dass bei solchen Apps ebenfalls Probleme im Zusammenhang mit (fehlendem) Kontext bestehen, die aber aufgrund anderer Anforderungen wie vorhandener Platz auf dem Display anders geartet sind. Auch hierfür könnte eine Korpusstudie hilfreich sein.

In Abschnitt 4.7 wurden bereits gewisse Einschränkungen im Hinblick auf die Zuordnung von Übersetzungsproblemen zu einzelnen Kontextarten genannt, die bei der weiteren Konkretisierung der Problembereiche und der darin vorzufindenden Unterkategorien im Rahmen einer Korpusstudie zwangsläufig noch deutlicher zu Tage treten. Diese Konkretisierung ist jedoch unabdingbar, um klare Vorgaben für die Annotation zu machen. Dies bildet dann die Basis für objektivierbare und reproduzierbare Ergebnisse (vgl. beispielsweise Lemnitzer und Zinsmeister 2015, S. 57, zur Überprüfbarkeit durch andere Forschende und Lemnitzer und Zinsmeister 2015, S. 102, zur Definition nachvollziehbarer und objektiver Kategorien bei der Annotation). Wenn allerdings einzelne annotierte Strings für sich betrachtet werden, kann es durchaus sein, dass eine darin annotierte Kontextproblematik subjektiv betrachtet irrelevant ist oder gar kein Übersetzungsproblem im eigentlichen Sinne darstellt. Dies wird insbesondere dann deutlich, wenn im Korpus alle Treffer für einzelne annotierte Features gleichzeitig angezeigt und verglichen werden und man den Eindruck gewinnen kann, dass sich die Fälle in verschiedene Schweregrade einteilen lassen. Dieses Dilemma ist allerdings nicht

vermeidbar, wenn eine möglichst hohe Objektivierbarkeit der Ergebnisse erreicht werden soll. In diesem Zusammenhang sollten auch die teils geringen Übereinstimmungswerte mit der Zweitannotation als Ausgangspunkt dienen, die einzelnen Features nochmals genauer zu betrachten und ggf. das Annotationsschema, die Anweisungen oder die Entscheidungsbäume zu optimieren.

Wie beispielsweise in Abschnitt 5.5.1.1 in Bezug auf die Problematik mit ambigen Wortarten erwähnt, basieren die entwickelten Features neben der Einteilung und Herleitung auf der explorativen Untersuchung des vorliegenden Korpus, wobei qualitative und quantitative Verfahren zum Einsatz kamen (vgl. Hirschmann 2019, S. 6). Dabei wurden in einem ersten Schritt auf Basis der Einteilungen in verschiedene Kontextarten in Abschnitt 4.4 und der Zuordnung der in der Literatur zur Softwarelokalisierung beschriebenen Problembereiche zu diesen Kontextarten in Abschnitt 4.6 Features entwickelt und teils anhand der im Korpus vorgefundenen Phänomene ergänzt und konkretisiert. Wie von Lemnitzer und Zinsmeister beschrieben, wurde das Annotationsschema also „in Auseinandersetzung mit den zu annotierenden Daten“ (vgl. Lemnitzer und Zinsmeister 2015, S. 103) entwickelt. Durch diese Herangehensweise besteht jedoch, beispielsweise in den Bereichen ambige Wortarten und unklare Aneinanderreihungen, aber auch potenziell in anderen Bereichen, die Möglichkeit, dass in anderen Korpora mit ähnlichen Inhalten noch weitere Phänomene zu finden sind, die in dieser Korpusstudie nicht erfasst wurden. Von einer umfassenden Dokumentation der vorliegenden Phänomene kann also nicht ausgegangen werden.

Zudem wurden mit diesem Korpus nur bestimmte Bereiche untersucht, die im Zusammenhang mit der Kontextfrage stehen. Es stellt also keine Untersuchung aller bei der Softwarelokalisierung vorkommender Probleme dar. Neben manchen bereits in Abschnitt 4.7 genannten weiteren Problemen, die mit technischen Aspekten zu tun haben, wie Concatenations und Variablen (s. Abschnitt 2.2 und vgl. beispielsweise Reineke 2005, S. 86, zu Variablen und Behrens 2016 zu Concatenations), gibt es auch noch weitere Aspekte, die in der vorliegenden Untersuchung außer Acht gelassen wurden, die aber u. U. ebenfalls in die Kontextproblematik fallen und mit der Kürze vieler Strings zu tun haben. Beispielsweise gibt es auch Fälle von Homonymie außerhalb der Fachterminologie, wie bei „left“ (links; verbleibend) (vgl. K. J. Dunne 2006b, S. 104), „right“ (rechts; Berechtigung) oder „medium“ (Medium, mittel). Solche Fälle wurden nicht annotiert, weil neben dem Bedeutungswechsel auch ein Wortartwechsel vorliegt.

Ein weiteres potenzielles Problem findet sich bei den Features der fachterminologischen Aneinanderreihungen in der Kategorie „Fachlicher Kontext“ und den Features der unklaren Aneinanderreihungen in der Kategorie „Funktionaler Kontext“: In diesem Bereich wurde wie in Abschnitt 5.5.3.1 erläu-

tert eine ganze Anzahl von Stellen parallel für beide Kontextarten annotiert, nämlich längere Aneinanderreihungen mit Fachtermini. Bei der gewählten Vorgehensweise ließ sich diese Dopplung nicht vermeiden.

In Abschnitt 2.2 wurden Steuerzeichen mit Bezug zur Programmfunktionalität (s. Abschnitt 2.2.1), Steuerzeichen mit Bezug zur Formatierung (s. Abschnitt 2.2.2) und zeichenbasierte Konventionen (s. Abschnitt 2.2.3) als Besonderheiten von GUI-Text vorgestellt, die teils als Indikatoren dafür dienen, in welchen GUI-Bereichen Strings anzusiedeln sind, die solche Besonderheiten aufweisen (s. auch Abschnitt 5.7 und vgl. beispielsweise Beste 2006, S. 42, 64, und Esselink 2000, S. 63, 68–69). Diese Zuordnung kann wiederum dazu genutzt werden, die Ambiguität von Wortarten aufzulösen. Der String `~Open...` (*Office-Anwendung*), der mit dem Feature `verb.inf_oder_adj` müsste somit also in einem Menü enthalten sein. Da Benutzerinnen über Menüs die gewünschte Funktion eines Softwareprodukts aufrufen können (vgl. Beste 2006, S. 61), kann davon ausgegangen werden, dass bei diesem String ein Verb angegeben ist, mit dem diese Funktion beschrieben wird, da mit dem Adjektiv lediglich ein Status ausgedrückt wird. Diese Art von Indikatoren für Strings wurde bei der Annotation des Korpus ignoriert.

Außerdem lässt diese Untersuchung Fragen offen, die auch in den folgenden Kapiteln nicht beantwortet werden, wie beispielsweise: Wenn in einem String eine Annotation vorhanden ist, bedeutet dies dann, dass er schwerer zu übersetzen ist als Strings ohne Annotation? Steigt die Schwierigkeit, wenn ein String mehrere Annotationen enthält, und steigt sie weiter an, wenn diese Annotationen verschiedenen Kontextarten angehören? Besteht die Möglichkeit, dass die Gesamtzahl der Annotationen eines Subkorpus etwas über den Schweregrad des Subkorpus bei der Übersetzung aussagt? Insbesondere bei der Kategorie „Funktionaler Kontext“ steht wie oben angesprochen auch noch die Qualität des Ausgangstexts zur Diskussion: Kann die Anzahl der Vorkommen von Annotationen der Kategorie „Funktionaler Kontext“ wirklich ein Maß für zumindest bestimmte Teilaspekte der Qualität des Ausgangstexts sein?

Bei den meisten der eben genannten offenen Fragen rückte bereits die Übersetzung als nächster Schritt in den Mittelpunkt. Bislang wurde lediglich der Ausgangstext untersucht, bei der Übersetzung sind aber weitere Erkenntnisse zu den hier beschriebenen, annotierten und ausgewerteten Features möglich. Dabei kann einerseits das Produkt, also die Übersetzung (vgl. Saldanha und O'Brien 2013, S. 50), in den Blick genommen werden, beispielsweise indem untersucht wird, wie sich die verschiedenen Kontextarten auf die Übersetzung auswirken. Dieser Bereich wird im nächsten Kapitel, Kapitel 6, betrachtet. Andererseits kann aber auch der Prozess des

Übersetzens in den Blick genommen werden, in diesem Fall das Verhalten und die kognitiven Prozesse der Übersetzerinnen (vgl. Saldanha und O'Brien 2013, S. 109), die sich beim Umgang mit Übersetzungsproblemen zeigen, die aus den verschiedenen Kontextarten ergeben. Fragen aus diesem Kontext wird in Kapitel 7 nachgegangen.

Wenn einerseits eine produkt- und andererseits eine prozessorientierte Herangehensweise gewünscht ist, werden verschiedene Forschungsmethoden benötigt, wie in den folgenden Kapiteln erläutert wird. Bei beiden Untersuchungen kann aber als weiterer Themenbereich die Nutzung der Referenzmaterialien in den Blick genommen werden, die in Abschnitt 4.5 bereits den drei Kontextarten zugeordnet wurden: Erkennen Übersetzerinnen Recherchebedarf, wie wird bei der Recherche vorgegangen, wie wirkt sich das auf die erstellten Übersetzungen bzw. auf den Übersetzungsprozess aus und für welche Kontextprobleme sind welche Arten von Materialien nützlich? Auch dieser Fragenkomplex wird in den weiteren Kapiteln angegangen, zunächst primär produktorientiert anhand eines Übersetzungstests in Kapitel 6 und danach prozessorientiert mithilfe einer Eye-Tracking- und Key-Logging-Studie in Kapitel 7.

Kapitel 6

Recherche und Übersetzungsalternativen bei der Übersetzung der Kontextarten

Das im letzten Kapitel annotierte Korpus bildet die Grundlage für einen Übersetzungstest, bei dem der Blick weg vom Ausgangstext und hin zur Übersetzung gerichtet werden soll, wobei der Fokus weiterhin die Produktorientierung bleibt. Insbesondere die Fragen, ob sich für die verschiedenen Kontextarten hinsichtlich der Recherche bzw. des Referenzmaterials einerseits sowie in Bezug auf die Anzahl der Übersetzungsalternativen andererseits Unterschiede ergeben, stehen dabei im Mittelpunkt. Hierzu werden in diesem Kapitel zunächst die Hypothesen (s. Abschnitt 6.1) und im Anschluss daran die Konzeption des dafür durchgeführten Übersetzungstests (s. Abschnitt 6.2) dargestellt. Danach finden sich Angaben zur Durchführung des Übersetzungstests und zu den Probandinnen (s. Abschnitt 6.3). Die Auswertung, die Angaben zu Vorbereitungen einerseits und die ermittelten Daten hinsichtlich Recherche sowie Anzahl der Übersetzungsalternativen andererseits umfasst, wird im Anschluss daran dargestellt (s. Abschnitt 6.4) und danach interpretiert (s. Abschnitt 6.5). Zum Ende folgt ein Abschnitt mit Einschränkungen dieses Übersetzungstests (s. Abschnitt 6.6).

6.1 Hypothesen des Übersetzungstests

In den folgenden Abschnitten werden in zwei Bereichen Hypothesen herausgearbeitet, zum einem zum Referenzmaterial in Abschnitt 6.1.1 und zum anderen zur Anzahl der Übersetzungsalternativen in Abschnitt 6.1.2.

6.1.1 Hypothesen zum Referenzmaterial

In Abschnitt 4.5 wurden verschiedene Referenzmaterialien genannt, die für die drei in Abschnitt 4.4 identifizierten Kontextarten „Situativer Kontext“, „Fachlicher Kontext“ und „Funktionaler Kontext“ von Nutzen sein können. Betrachtet man diese genauer, so zeigt sich zweierlei: Zum einen ist ein Großteil dieser Materialien nur im Rahmen eines konkreten Softwarelokalisierungsprojekts über die Auftraggeberin und nicht für Außenstehende verfügbar, zum anderen sind die Materialien nicht für alle Kontextarten gleichermaßen als hilfreich einzuschätzen. Hier ein Überblick dazu aus Perspektive der Kontextarten:

Kategorie „Situativer Kontext“:

- Eine lauffähige Version der zu übersetzenden Software (vgl. Ottmann 2005, S. 102, und Esselink 2000, S. 53) bzw. Simulationen und Demos, falls dies nicht möglich ist (vgl. Ottmann 2005, S. 102)
- Die Dokumentation des Softwareprodukts in Form von Hilfeinformationen bzw. Onlinehilfe (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, und Wahle 2000b, S. 45) oder auch Referenzhandbücher, die Beschreibungen der einzelnen Strings enthalten (vgl. Ottmann 2005, S. 104)
- In den zu übersetzenden Dateien enthaltene Informationen wie Kommentare der Entwicklerinnen und weitere Informationen in den Ressourcendateien (vgl. Esselink 2000, S. 34, 64–65, Wahle 2000b, S. 37–38, 40, 45, Reineke 2005, S. 85, und Sachse 2005, S. 147–148, 158)

Kategorie „Fachlicher Kontext“:

- Lauffähige Version der zu übersetzenden Software bzw. Simulationen und Demos
- Dokumentation des Softwareprodukts
- In den zu übersetzenden Dateien enthaltene Kommentare
- Verschiedene Glossare, wie etwa Terminologie für die relevanten Betriebssysteme, die Software oder das Fachgebiet (vgl. Ottmann 2005, S. 104, und Esselink 2000, S. 53, 401–402)
- Hintergrundinformationen zum Produkt (vgl. Esselink 2000, S. 53)

Kategorie „Funktionaler Kontext“:

- Lauffähige Version der zu übersetzenden Software bzw. Simulationen und Demos
- Dokumentation des Softwareprodukts
- In den zu übersetzenden Dateien enthaltene Kommentare

Orientierung am AT geringer wird (vgl. Rydning und Lachaud 2010, S. 99). Dass auch bei der kontextlosen Übersetzung polysemer Wörter eine gewisse Varianz vorliegt, liegt im Falle ihrer Untersuchung an der Wahl polysemer Wörter: Ohne Kontext wird von den Probandinnen eine der ihnen bekannten möglichen Übersetzungsalternativen genutzt, durch den Kontext erfolgt dagegen die Disambiguierung bzw. Monosemierung (vgl. Kußmaul 2015, S. 21). Es wird als ersichtlich, welche der Übersetzungsalternativen zu verwenden ist, und gleichzeitig kann bei der Suche nach freieren Übersetzungen kreativer vorgegangen werden.

Wenn man dieses Ergebnis insbesondere auf die typischen kurzen Strings in der Softwarelokalisierung (vgl. beispielsweise Beste 2006, S. 40) überträgt, die aus wenigen Wörtern bestehen und ohne weiteren Kontext übersetzt werden müssen, könnte es zu einem ähnlichen Effekt kommen: Ohne Kontext orientieren sich die Probandinnen eines Übersetzungstests stärker am AT. Es werden also weniger Übersetzungsalternativen produziert. Im umgekehrten Fall werden durch dieselbe Probandinnengruppe viele Übersetzungsalternativen produziert, was im nächsten Schritt zur Berechnung der Entropie genutzt werden kann (vgl. beispielsweise Bangalore u. a. 2016, S. 213 und s. 7.6.3).

Nicht auszuschließen ist allerdings, dass sich die verschiedenen Kontextarten auf unterschiedliche Weise auf die Anzahl der Übersetzungsalternativen auswirken, insbesondere da in Abschnitt 4.6 für die verschiedenen Kontextarten unterschiedliche Übersetzungsprobleme beschrieben wurden. Dies lässt sich mit folgender Nullhypothese untersuchen:

- Für alle Strings der drei Kontextarten gibt es jeweils eine eindeutige Lösung, die die Mehrzahl der Probandinnen angibt.

6.2 Konzeption des Übersetzungstests

Um diese Hypothesen zu prüfen, wurde ein Übersetzungstest entwickelt, bei dem die Probandinnen 100 Strings aus dem Korpus übersetzen mussten, von denen je 25 dem situativen, dem funktionalen und dem fachlichen Kontext zugeordnet waren. Hinzu kamen weitere 25 Strings als Referenzgruppe, bei denen keine Annotation vorliegt.

Für die Vergleichbarkeit der Ergebnisse bei der im Anschluss an den Übersetzungstest mit denselben Strings geplanten Eye-Tracking-Studie (s. Kapitel 7) war es wichtig, dass sich die Länge der Strings im Übersetzungstest nicht zu sehr unterscheidet. Deshalb wurden nur Strings aus zwei oder drei Wörtern in den Test aufgenommen. Für diese Herangehensweise sprach zudem auch die Tatsache, dass ein deutlicher Anteil der Features im Korpus in kürzeren

Strings annotiert wurde.¹

Die für den Übersetzungstest gewählten Strings sollten möglichst aus allen zehn Subkorpora stammen, wobei das Korpus „Bildbearbeitung“ deutlich unterrepräsentiert ist, da darin fast nur lange Strings enthalten sind (s. Abschnitt 5.3.1 und Abschnitt 5.6).

Der Test besteht aus einer Excel-Tabelle (s. Anhang B.2), bei der jede Zeile einen zu übersetzenden String enthält. Die Strings wurden zufällig ausgewählt und wurden nach Programmtyp gruppiert. Auf diese Weise wurde die eine grundlegende Kontextinformation, die im Minimalfall bei einem Softwarelokalisierungsprojekt vorhanden ist, nämlich das grundsätzliche Anwendungsgebiet des Softwareprodukts, für die Probandinnen bereitgestellt.² Innerhalb der Programmtypen sind die Kontextarten aber gemischt, weil die Strings darin alphabetisch sortiert sind. Die Tabelle enthält folgende Spalten, in die die Probandinnen jeweils ihre Ergebnisse eintragen mussten:

- „Zu übersetzende Elemente“: Die zu übersetzenden AT-Strings
- „Erster spontaner Einfall für die Übersetzung sowie ggf. alle weiteren Ü-Vorschläge ohne Recherche (in der Reihenfolge des Einfallens)“: Zur Eingabe von Übersetzungsvorschlägen ohne vorherige Recherche
- „Endgültiger Ü-Vorschlag“: Für den finalen Übersetzungsvorschlag, der mit oder ohne Recherche aus den spontanen Vorschlägen ausgewählt bzw. durch die Recherche neu identifiziert wurde
- „Recherche durchgeführt (ja/nein)“: Für die Angabe, ob für die Übersetzung des Strings eine Recherche durchgeführt wurde
- „Wo haben Sie recherchiert? (Mehrfachnennungen möglich)“: Für die Angabe der Quellen, in denen recherchiert wurde³
- „War die Recherche erfolgreich? (ja/nein/weiß nicht)“: Für eine subjektive Einschätzung dazu, ob die Recherche erfolgreich war

Für alle Strings wurden vor Aufnahme in den Test folgende Vereinheitlichungsmaßnahmen vorgenommen:

- Aus allen Strings wurden die enthaltenen Steuerzeichen entfernt.
- Die Groß-/Kleinschreibung der Strings wurde vereinheitlicht.

¹84,6 Prozent der Annotationen der Kategorie „Situativer Kontext“ betreffen Strings mit ein bis drei Wörtern Länge. Bei der Kategorie „Fachlicher Kontext“ sind dies 50,9 Prozent und bei der Kategorie „Funktionaler Kontext“ 50,0 Prozent (s. Abschnitt 5.6.1).

²Der String „Decoding video“ wurde dabei versehentlich der Kategorie „Office-Anwendung“ zugeordnet, nicht der Kategorie „Videobearbeitung“. Dies wurde bei der Auswertung (s. Abschnitt 6.4) korrigiert.

³Diese Spalte wurde bei der Auswertung im Abschnitt 6.4 allerdings nicht berücksichtigt.

- Etwaige Sonderformatierungen und Satzzeichen wurden entfernt.

Die Liste der Strings aus dem Übersetzungstests mit der jeweiligen Kontextart plus annotiertem Feature findet sich in Anhang [B.3](#). Darin ist zu erkennen, dass in der Kategorie „Situativer Kontext“ sämtliche Untersysteme aus dem Annotationsschema enthalten sind, wobei im Untersystem `PROBLEM_WORTART` fünf von neun Features vertreten sind und bei den anderen beiden Untersystemen alle Features außer `über_sonstiges`, das nur in einem Fall annotiert wurde. Die Features des Untersystems `PROBLEM_WORTART` sind dabei am öftesten enthalten, danach die des Untersystems `PROBLEM_GUI-BEZUG` und zuletzt die des Untersystems `PROBLEM_FLEXIONSENDUNG`, was grob der anteilmäßigen Verteilung der Untersysteme der Kategorie „Situativer Kontext“ im Korpus entspricht (s. Abschnitt [5.6.2](#)).

In der Kategorie „Fachlicher Kontext“ sind von sieben Features für die verschiedenen fachterminologischen Wortarten vier enthalten, von den sechs Features im Untersystem `FACHTERM_ANEINANDERREIHUNG` drei. Bei diesen drei handelt es sich lediglich um zweiteilige Aneinanderreihungen, nämlich Komposita aus zwei Substantiven, Komposita mit Abkürzung und Komposita mit Eigenname, weil die weiteren Aneinanderreihungen in aller Regel in Strings enthalten sind, die länger als zwei bis drei Wörter sind. Dies entspricht aber grob den Anteilen im Korpus, in dem Komposita aus zwei Substantiven über 60 Prozent der Annotationen im Untersystem `FACHTERM_ANEINANDERREIHUNG` darstellen.

In der Kategorie „Funktionaler Kontext“ ist das Untersystem `UNKLARE_ANEINANDERREIHUNG` nur in einem Fall vertreten, weil unklare Aneinanderreihungen analog zu den fachterminologischen Aneinanderreihungen primär in längeren als den für den Übersetzungstest gewählten Strings mit zwei bis drei Wörtern vorkommen. Die übrigen Features sind alle vertreten, wobei das Feature `unklare_bedeutung` mit Abstand am häufigsten vorkommt. Dies entspricht nicht den Anteilen im Korpus, allerdings sind im Korpus praktisch keine Vorkommen des Feature `metaphorik` zu finden, die in Zwei- bis Drei-Wort-Strings ohne weitere Annotationen enthalten sind. Beim Feature `homonym_it-term` gibt es ebenfalls nur wenige taugliche Strings, weil im Korpus zwar einige Fälle vorhanden sind, diese sich aber auf nur ein paar wenige Wörter verteilen (s. Abschnitt [5.7](#)).

Alle Quellen waren für die Recherche zugelassen, auch Suchmaschinen, wobei jedoch nur einzelne Wörter in die Suchmaschinen eingegeben werden durften, nicht die vollständigen Strings (s.o.). Maschinelle Übersetzung war ebenso ausgeschlossen.

6.3 Durchführung des Übersetzungstests und Probandinnen

Vor Durchführung des eigentlichen Übersetzungstests wurde mit zwei Probandinnen ein Pilottest durchgeführt. Deren Feedback wurde zur Optimierung der Excel-Datei mit dem Übersetzungstest (s. Anhang [B.2](#)) und der Anweisungsdatei für den Übersetzungstest (s. Anhang [B.1](#)) genutzt.

An dem Test nahmen 20 Studierende teil, von denen 18 im Master Translation am Fachbereich 06 (FTSK) der Johannes Gutenberg-Universität Mainz eingeschrieben waren. Eine weitere Probandin hatte bereits kurz vor Durchführung der Studie den Master Translation erfolgreich abgeschlossen und eine andere war noch im Bachelor Sprache, Kultur, Translation eingeschrieben, hatte aber im Bachelor an der Fachübersetzungsübung Internettechnologien teilgenommen und war im Prozess der Bewerbung zum Master Translation. Alle Teilnehmenden hatten aufgrund der Belegung der Fachübersetzungsübung Internettechnologien in diesem Studiengang bereits Erfahrung mit IT-Fachtexten und mit der Recherche in relevanten Terminologiedatenbanken. Teils hatten die Probandinnen schon Veranstaltungen zur Softwarelokalisierung besucht. Falls dies nicht der Fall war, wurden in die Anweisungsdatei des Übersetzungstests Erläuterungen dazu eingefügt, dass die zu übersetzenden Elemente aus GUIs verschiedener Softwareprodukte stammen.

Die Probandinnen durften den Übersetzungstest zu Hause nach eigener Zeiteinteilung durchführen, sollten aber versuchen, ihn in 90 Minuten vollständig zu bearbeiten. Den Test bekamen sie per Mail mit einem festen Abgabetermin und den Anweisungen zugeschickt und sollten den ausgefüllten Test per Mail an die Studienleitung zurückschicken. In Anhang [B.1](#) ist die Anweisungsdatei für den Übersetzungstest enthalten. Bei der Anwerbung der Probandinnen wurden diese über den Daten- und Persönlichkeitsschutz bei der durchzuführenden Studie aufgeklärt. Im Anschluss an die Teilnahme an der Studie erhielten sie eine Aufwandsentschädigung von 20 Euro.

6.4 Auswertung des Übersetzungstests

Vor der Auswertung des Übersetzungstests wurden einige für die Auswertung erforderliche Vorarbeiten ausgeführt, die in Abschnitt [6.4.1](#) beschrieben werden. Danach erfolgte die eigentliche Auswertung, für die zuvor noch einige Strings ausgeschlossen wurden (s. Abschnitt [6.4.2](#)). Danach folgen eine Teilauswertung im Hinblick auf die Recherche (s. Abschnitt [6.4.3](#)) und eine zweite Teilauswertung im Hinblick auf die Anzahl der Übersetzungsalterna-

tiven (s. Abschnitt 6.4.4). Im Anschluss daran erfolgt in Abschnitt 6.5 die Interpretation der ermittelten Ergebnisse.

6.4.1 Vorbereitungen für die Auswertung

Zunächst wurden sämtliche ausgefüllten Übersetzungstests in einer Excel-Tabelle zusammengeführt sowie durch die Spalten „StringID“, „Programmtyp“, „Kontextart“ und „Proband“ ergänzt. Danach wurden zahlreiche Vereinheitlichungen und Fehlerkorrekturen durchgeführt, die die maschinelle Auswertung ermöglichen sollten. Details dazu finden sich in Anhang B.4.

Hatten die Probandinnen in der Spalte „Erster spontaner Einfall“ (und in einem Fall bei P17 für den String „Manage track changes“ sowie in einem Fall bei P08 für den String „Report generated to“ in der Spalte „Endgültiger Ü-Vorschlag“) mehr als einen Übersetzungsvorschlag eingefügt, wurden diese Vorschläge in eigene, neue Zeilen übernommen, so dass jeder Übersetzungsvorschlag in einer eigenen Zeile enthalten war. In diesen neuen Zeilen wurde in die Spalte „Zu übersetzende Elemente“ der AT-String plus ein Zahlenwert eingefügt, beispielsweise „File Format Support 2“ für den zweiten Übersetzungsvorschlag der Probandin für diesen AT-String. In den Spalten „Endgültiger Ü-Vorschlag“, „Recherche durchgeführt?“, „Wo haben Sie recherchiert?“ und „War die Recherche erfolgreich?“ wurde in diese Zeilen jeweils die Zeichenfolge „XXXXX“ eingefügt und in die übrigen Spalten die Werte der Stammzeile übernommen.

6.4.2 Ausgewertete Strings im Übersetzungstest

Tabelle 6.1 enthält eine Übersicht über die Anzahl der Strings in den verschiedenen Programmtypen und Kontextarten, die nach Abschluss des Tests in die Auswertung einfließen. Diese Zahlen sind niedriger als die ursprünglich für den Übersetzungstest festgelegten 100 Strings (s. Abschnitt 6.2), da sich einige der ausgewählten Strings nachträglich als ungeeignet erwiesen, beispielsweise weil sie mehrere Annotationen unterschiedlicher Kontextarten enthielten. Alle nachfolgenden Auswertungen beziehen sich auf die in dieser Tabelle angegebenen Werte, die Ergebnisse werden daher nicht in absoluten Zahlen, sondern in Prozentsätzen im Vergleich zur Gesamtmenge angegeben.

Tabelle 6.1: Programmtypen und Kontextarten

Programmtyp	Kein	Situativ	Funktional	Fachlich	Gesamt
Audiobearbeitung	1	2	1	1	5
Bildbearbeitung	1	0	0	0	1
Browser	2	1	1	1	5
Dateimanager	1	7	1	3	12
E-Mail	2	2	2	3	9
Editor	3	1	4	2	10
HTML-Editor	1	0	3	6	10
Medienplayer	3	2	1	1	7
Office-Anwendung	4	5	3	1	13
Videobearbeitung	4	2	3	0	9
Gesamt	22	22	19	18	81

6.4.3 Auswertung hinsichtlich der Recherche

Zunächst wurde ausgewertet, wie oft und in welchen Fällen eine Recherche durchgeführt wurde (s. Abbildung 6.1).⁴ In ca. 0,7 Prozent aller Fälle wurden allerdings keine Angaben gemacht. Diese Fälle sind im Diagramm als „Nicht zutreffend“ markiert.

Insgesamt wurde für 47,7 Prozent aller Strings eine Recherche durchgeführt, bei 51,6 Prozent verzichteten die Probandinnen auf eine Recherche. Für die vier Kontextkategorien ergaben sich dagegen unterschiedliche Anteile: Während bei der Kategorie „Kein Kontext“ in 18,6 Prozent der Fälle eine Recherche durchgeführt wurde, wurde in 80,9 Prozent der Fälle nicht recherchiert. Für die weiteren Kategorien, „Situativer Kontext“, „Fachlicher Kontext“ und „Funktionaler Kontext“, nimmt die Recherchequote kontinuierlich zu, von 45,2 Prozent über 57,7 Prozent auf 74,9 Prozent, während parallel die Fälle abnehmen, in denen nicht recherchiert wurde, von 54,1 Prozent über 42,1 Prozent auf 23,5 Prozent.

In Bezug auf die Frage, ob die Probandinnen die Recherche in den Fällen als erfolgreich einschätzten, in denen sie auch recherchiert haben, zeigen sich ebenfalls Unterschiede für die vier Kontextkategorien (s. Abbildung 6.2). In den Anweisungen zum Übersetzungstest wurden zur Beantwortung der Frage, wann genau eine Recherche als erfolgreich gelten sollte, keine Angaben gemacht, hier war also eine subjektive Einschätzung zu erwarten. Der dies-

⁴Die Interpretation der in diesem Abschnitt beschriebenen Daten erfolgt in Abschnitt 6.5.3 unten.

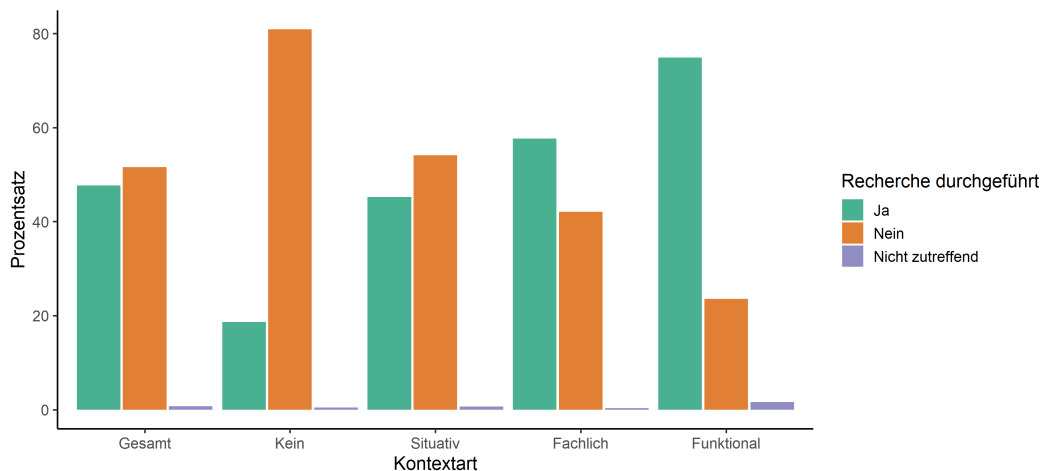


Abbildung 6.1: Wurde eine Recherche durchgeführt?

bezügliche Passus der Anweisungsdatei lautet wie folgt: „Bitte führen Sie nur dann eine Recherche durch, wenn Sie dies für erforderlich halten, und geben Sie entsprechend in den **Spalten 4, 5 und 6** an, **ob Sie eine Recherche durchgeführt** haben, **wo Sie recherchiert haben** (Mehrfachnennungen möglich) und **ob die Recherche erfolgreich war.**“ (s. Anhang B.1; Hervorhebungen im Original).

Insgesamt waren die Probandinnen bei 55,7 Prozent der Strings, bei denen sie eine Recherche durchführten, der Meinung, dass ihre Recherche erfolgreich war, bei 22,4 Prozent schätzten sie die Recherche als nicht erfolgreich ein und bei 21,9 Prozent gaben sie „Weiß nicht“ an. Bei den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ wurden mit 70,7 Prozent bzw. 76,2 Prozent die größten Anteile der Recherchevorgänge als erfolgreich betrachtet, wobei gleichzeitig ebenso wie bei der Kategorie „Situativer Kontext“ der Prozentsatz der Fälle, bei denen die Recherche als nicht erfolgreich angesehen wurde, geringer war als der, bei denen „Weiß nicht“ angegeben wurde. Für die Kategorie „Funktionaler Kontext“ ergibt sich an anderes Bild: Der Prozentsatz der erfolgreichen Recherchevorgänge liegt mit 42,6 Prozent sogar noch etwa 4 Prozent niedriger als beim situativen Kontext und die Einschätzung „Nein“ für den Rechercheerfolg wurde in 36,1 Prozent der Fälle angegeben. Demgegenüber wurde in 21,3 Prozent der Fälle „Weiß nicht“ angegeben. Diese Kategorie ist also die einzige, bei der die Angabe „Nein“ häufiger gemacht wurde als die Angabe „Weiß nicht“.

Ergänzend zu diesen Werten lässt sich betrachten, ob und wie viele spontane Vorschläge für die unterschiedlichen Kategorien angegeben wurden (s. Abbildung 6.3). Diese Auswertung lässt sich einerseits als Hinweis dazu be-

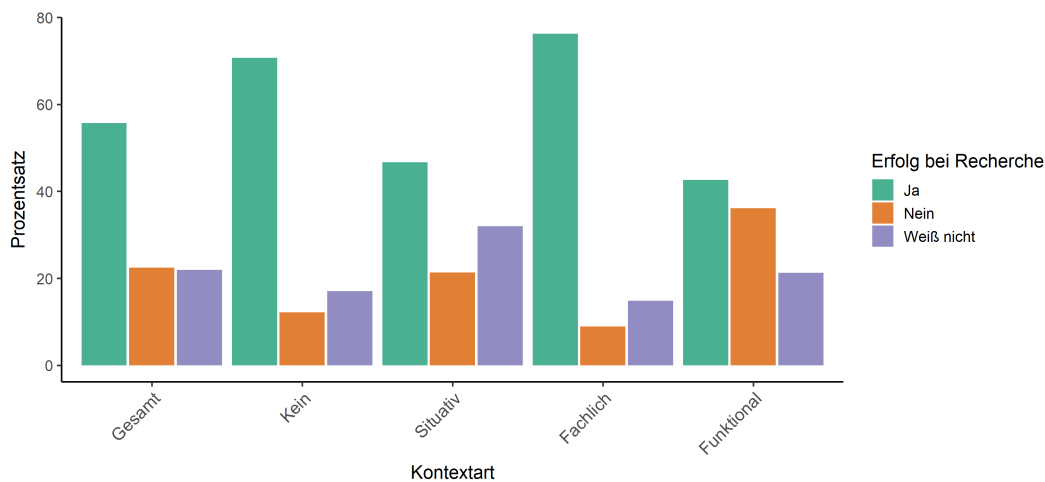


Abbildung 6.2: War die Recherche erfolgreich?

trachten, ob die Übersetzungsprobleme bei einem String so hoch waren, dass sich den Probandinnen vor der Recherche gar kein Übersetzungsvorschlag angeboten hat. Andererseits könnten mehrere verschiedene spontane Übersetzungsvorschläge darauf hindeuten, dass es bei einem String noch Unklarheiten entweder beim Verstehen, beispielsweise durch fehlenden Kontext, oder bei der Kenntnis fachterminologischer Benennungen gab.

Im Schnitt für alle Kategorien wurde für 16,7 Prozent der Strings kein spontaner Übersetzungsvorschlag angegeben, für 69,4 Prozent einer, für 10,4 Prozent zwei, für 3,0 Prozent drei und für 0,6 Prozent vier. Für die vier Kontextkategorien ist zu sehen, dass der Prozentsatz der Strings, bei denen kein spontaner Übersetzungsvorschlag vorliegt, von „Kein Kontext“ mit 3,0 Prozent über „Situativer Kontext“ mit 9,5 Prozent und „Fachlicher Kontext“ mit 18,9 Prozent auf „Funktionaler Kontext“ mit 38,7 Prozent ansteigt. Umgekehrt sinkt die Anzahl der Strings mit einem einzigen Übersetzungsvorschlag zwischen diesen Kontextarten von 84,3 Prozent bei der Kategorie „Kein Kontext“ bis auf 49,2 Prozent bei der Kategorie „Funktionaler Kontext“.

Abschließend wurde ermittelt, ob es Änderungen zwischen dem spontanen und dem endgültigen Übersetzungsvorschlag gab, sofern beide vorhanden waren (s. Abbildung 6.4), bei vielen Strings vermutlich auf Basis der erfolgten Recherche.⁵ Bei Strings, für die mehrere spontane Übersetzungsvorschläge eingegeben wurden, wurde dabei nur der zuerst eingegebene Vorschlag mit

⁵Für 20 Strings wurde kein endgültiger Übersetzungsvorschlag angegeben, also für ca. 1 Prozent der übersetzten Strings. Überwiegend waren dies Strings mit funktionalem Kontext, hinzu kamen ein paar mit situativem sowie einer mit fachlichem und einer ohne Kontext. Da dies eine zu vernachlässigende Menge darstellt, wurden diese Fälle ignoriert.

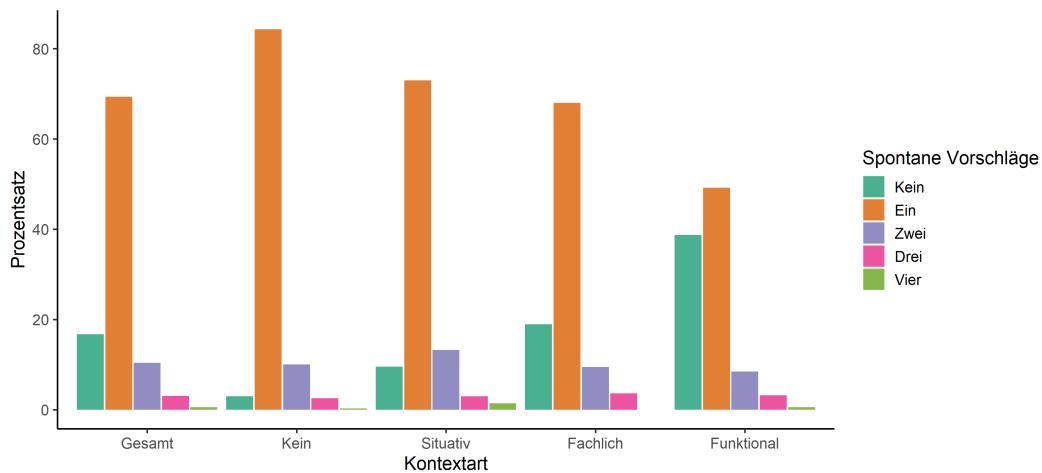


Abbildung 6.3: Anzahl der spontanen Übersetzungsvorschläge pro Probandin

dem endgültigen Vorschlag abgeglichen.

Aus diesem Diagramm wird ersichtlich, dass in der Gesamtbetrachtung 71,4 Prozent der endgültigen Übersetzungsvorschläge mit dem ersten spontanen Vorschlag identisch waren. Mit Abstand der größte Prozentsatz der Übereinstimmungen zeigt sich mit 84,1 Prozent für die Kategorie „Kein Kontext“, dann folgen die Kategorien „Fachlicher Kontext“ mit 74,6 Prozent und „Situativer Kontext“ mit 65,7 Prozent und zuletzt die Kategorie „Funktionaler Kontext“ mit 54,1 Prozent.

6.4.4 Auswertung hinsichtlich der Anzahl der Übersetzungsalternativen

Im nächsten Schritt wird die Anzahl der Übersetzungsalternativen betrachtet, die von den verschiedenen Probandinnen für die Strings angegeben wurden.⁶ In Abbildung 6.5 sind dafür zunächst die Prozentsätze der Strings zu sehen, bei denen mindestens zehn der von den Probandinnen eingegebenen endgültigen Übersetzungsvorschläge identisch sind.

In der Gesamtbetrachtung lassen sich bei 65,4 Prozent der Strings zehn identische Übersetzungsvorschläge finden, bei 34,6 nicht. Diese verteilen sich wie folgt auf die vier Kategorien: Bei 90,9 Prozent der Strings in der Kategorie „Kein Kontext“ gibt es mindestens zehn identische Übersetzungsvorschläge, bei der Kategorie „Fachlicher Kontext“ in 88,9 Prozent der Fälle,

⁶Die in diesem Abschnitt beschriebenen Daten werden in Abschnitt 6.5.3 unten interpretiert.

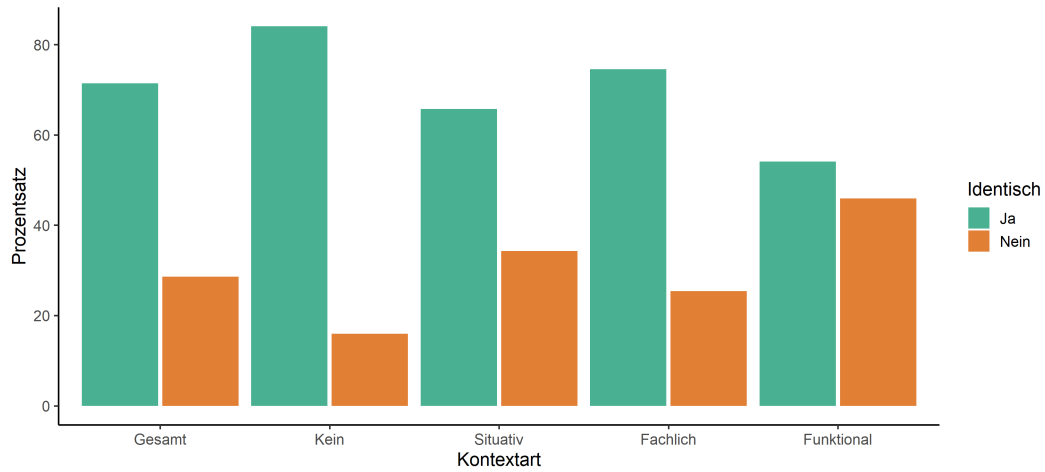


Abbildung 6.4: Ist der erste spontane Übersetzungsvorschlag identisch mit der endgültigen Übersetzung?

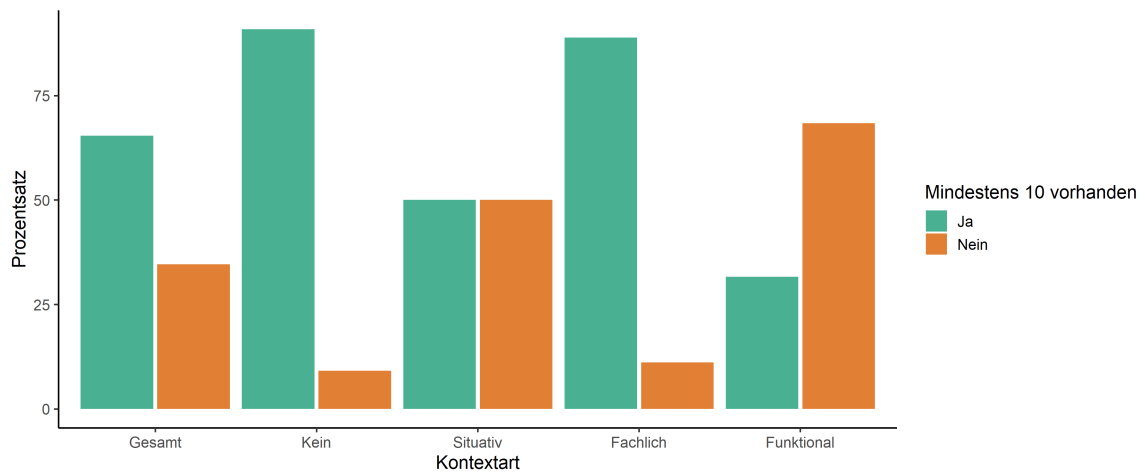


Abbildung 6.5: Sind mindestens 10 identische Vorkommen eines Übersetzungsvorschlags vorhanden?

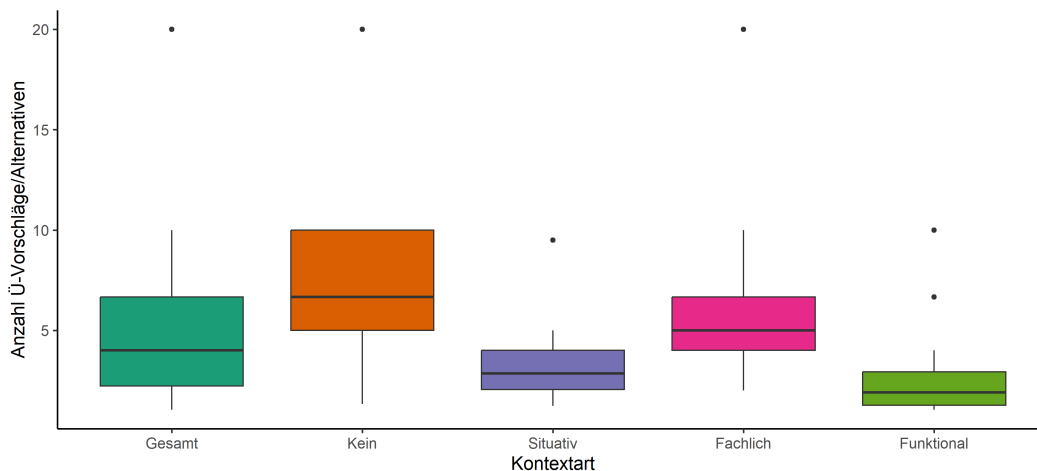


Abbildung 6.6: Anzahl der Übersetzungsalternativen pro String, auf die sich die endgültigen Übersetzungsvorschläge verteilen

bei der Kategorie „Situativer Kontext“ in 50 Prozent und bei der Kategorie „Funktionaler Kontext“ in 31,6 Prozent.

Bei der Antwort auf die Frage nach den Übersetzungsalternativen erbringt auch eine andere Herangehensweise aussagekräftige Ergebnisse: Abbildung 6.6 zeigt einen Box-Whisker-Plot, für den zunächst die Anzahl der endgültigen Übersetzungsvorschläge für einen String ermittelt wurde, da nicht in allen Fällen ein endgültiger Vorschlag angegeben wurde. Diese Zahl wurde anschließend durch die Anzahl der Übersetzungsalternativen für diesen String geteilt. Wenn viele Probandinnen für einen String denselben Übersetzungsvorschlag eingegeben haben, wenn also die Anzahl der Übersetzungsalternativen gering ist, ist der Teiler kleiner und das errechnete Ergebnis somit größer. Umgekehrt gilt: Viele unterschiedliche Übersetzungsvorschläge führen zu einem großen Teiler, wodurch das Ergebnis kleiner wird.

Die Werte aus dieser Berechnung ergänzen die Daten aus Abbildung 6.5: Der Median liegt für alle Strings insgesamt bei 4,0, der Interquartilsabstand beträgt 4,4, die Standardabweichung liegt bei 4,1. Für die Kategorie „Kein Kontext“ liegt der Median bei 6,7 und der Interquartilsabstand beträgt 5,0, die Standardabweichung hat den Wert 4,7. Die Kategorie „Fachlicher Kontext“ weist einen Median von 5,0 und einen Interquartilsabstand von 2,7 auf, die Standardabweichung liegt bei 4,2. Die Kategorien „Situativer Kontext“ mit den Werten 2,9, 1,9 und 1,8 und „Funktionaler Kontext“ mit 1,9, 1,6 und 2,2 weisen dagegen deutlich niedrigere Werte auf. Dies bedeutet, dass in diesen Kategorien die Anzahl der unterschiedlichen Übersetzungsvorschläge höher war.

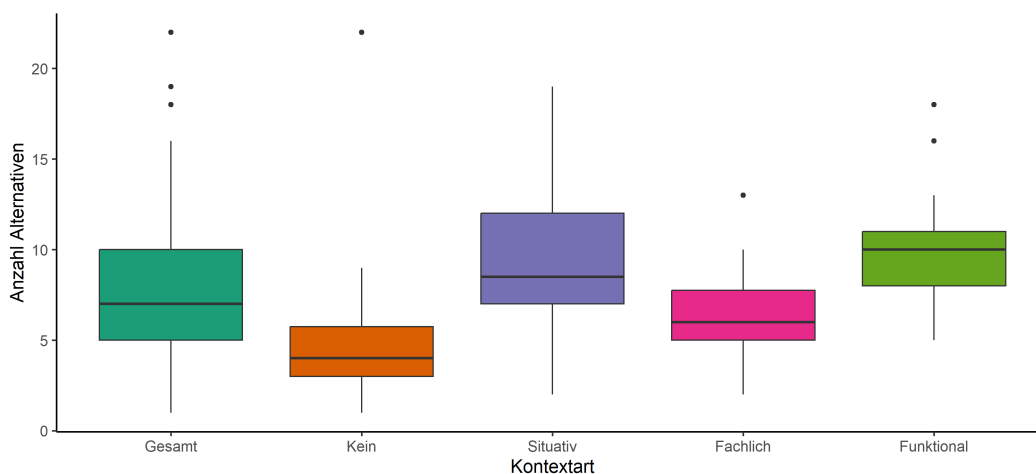


Abbildung 6.7: Anzahl der Übersetzungsalternativen aller Probandinnen bei den spontanen Übersetzungsvorschlägen

Weitere Aufschlüsse ergibt die Betrachtung der Anzahl der Übersetzungsalternativen, die die Probandinnen für die einzelnen Strings in der Spalte für spontane Übersetzungsvorschläge angegeben haben (s. Abbildung 6.7).

Der Median für die Anzahl der Übersetzungsalternativen für die spontanen Übersetzungsvorschläge insgesamt liegt bei 7,0, der Interquartilsabstand bei 5,0 (Standardabweichung: 4,3). Die Kategorien „Kein Kontext“ mit 4,0 und 2,75 (Standardabweichung: ebenfalls 4,3) und „Fachlicher Kontext“ mit 6,0 und 2,75 (Standardabweichung: 2,8) weisen die niedrigsten Werte für Median und Interquartilsabstand auf. Die Werte der Kategorien „Situativer Kontext“ mit 8,5 und 5,0 (Standardabweichung: 4,3) und „Funktionaler Kontext“ mit 10,0 und 3,0 (Standardabweichung: 3,2) sind höher, wobei der erstere den größten Interquartilsabstand aufweist, bei letzterem aber der Median im oberen Bereich der Box liegt.

Eine Ergänzung hierfür ist in Abbildung 6.8 zu finden, der die Anzahl der Übersetzungsalternativen in der Spalte für den endgültigen Übersetzungsvorschlag entnommen werden kann.

Hier ist zu sehen, dass die Mediane insgesamt und bei allen Kontextarten niedriger als bei den spontanen Übersetzungsvorschlägen sind und auch die Standardabweichung bei den Kontextarten jeweils unter den obigen Werten liegt. Lediglich die Kategorie „Funktionaler Kontext“ weicht hiervon ab: Der Median bleibt unverändert, allerdings sind der Interquartilsabstand (mit 8,5) und die Standardabweichung (mit 4,9) deutlich höher als bei der vorangegangenen Auswertung.

Betrachtet man nun nur die Übersetzungsalternativen, die in der Spalte

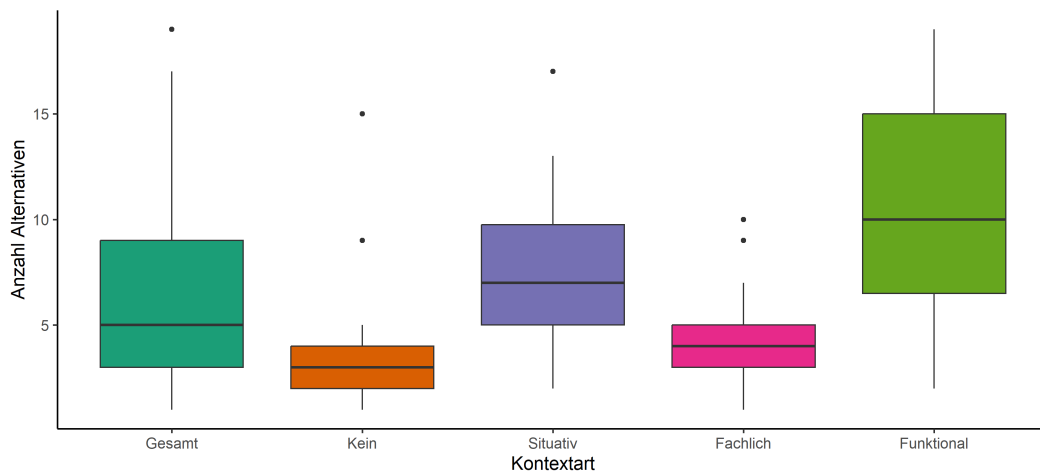


Abbildung 6.8: Anzahl der Übersetzungsalternativen aller Probandinnen bei den endgültigen Übersetzungsvorschlägen

für den endgültigen Übersetzungsvorschlag neu hinzugekommen sind, also noch nicht als spontane Übersetzungsvorschläge genannt wurden, ergibt sich das Bild in Abbildung 6.9.

Hier liegen der Median und die Standardabweichung deutlich niedriger als bei den vorangegangenen Box-Whisker-Plots. Sowohl Median als auch Interquartilsabstand sind insbesondere bei den Kategorien „Kein Kontext“ (Standardabweichung: 2,5) und „Fachlicher Kontext“ (Standardabweichung: 1,7) sehr niedrig, haben sich aber auch bei der Kategorie „Situativer Kontext“ (Standardabweichung: 2,2) verringert. Bei der Kategorie „Funktionaler Kontext“ fällt hingegen auf, dass der Median mit 4,0 deutlich niedriger ist als bei den vorangegangenen Plots, der Interquartilsabstand aber mit dem Wert 8,0 und die Standardabweichung mit dem Wert 4,7 fast so hoch sind wie zuvor.

6.5 Interpretation der Ergebnisse des Übersetzungstests

Die Ergebnisse lassen sich sowohl im Hinblick auf die Nullhypothesen aus Abschnitt 6.1 als auch auf die vier Kontextkategorien analysieren. Beides erfolgt in den weiteren Abschnitten, aufgeteilt nach den Hauptthemen Recherche (s. Abschnitt 6.5.1), Übersetzungsalternativen (s. Abschnitt 6.5.2) und Kontextkategorien (s. Abschnitt 6.5.3).

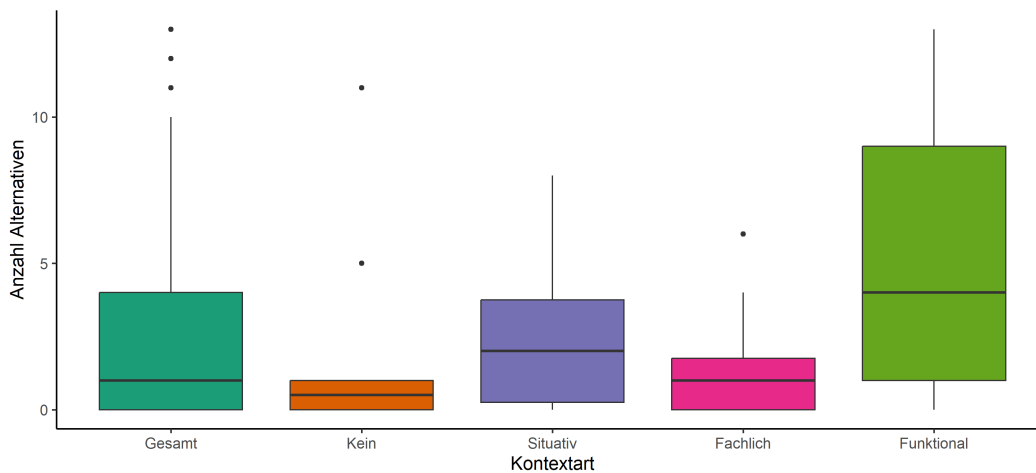


Abbildung 6.9: Anzahl der Übersetzungsalternativen aller Probandinnen, die bei den endgültigen Übersetzungsvorschlägen neu hinzugekommen sind

6.5.1 Recherche

Im Hinblick auf die Recherche wurden in Abschnitt 6.1.1 zwei Nullhypothesen formuliert:

- Für alle Strings der drei Kontextarten ist ein gleich hoher Recherchebedarf erforderlich.
- Für alle Strings der drei Kontextarten ist der Rechercheerfolg identisch.

Die obigen Daten (s. Abschnitt 6.4.3) zeigen, dass beim Übersetzungstest insgesamt bei knapp der Hälfte der Strings eine Recherche durchgeführt wurde, dass sich für die vier Kontextkategorien aber deutliche Unterschiede ergeben. Ebenso unterscheidet sich bei der Auswertung hinsichtlich dessen, ob die Probandinnen die Recherche als erfolgreich eingeschätzt haben, die Gesamtbetrachtung klar von den Ergebnissen für die vier Kontextkategorien und diese wiederum voneinander. Die Recherche wird von den Probandinnen bei den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ in fast bzw. mehr als drei Vierteln der Fälle als erfolgreich angesehen, wobei die Anzahl der Fälle, bei denen vom Misserfolg bei der Recherche ausgegangen wird, jeweils niedriger liegt als die Anzahl der Fälle, bei denen die Probandinnen nicht einschätzen konnten, ob die Recherche erfolgreich war oder nicht und daher die Antwort „Weiß nicht“ angegeben haben. Bei der Kategorie „Situativer Kontext“ wird weniger als die Hälfte der Recherchevorgänge als erfolgreich betrachtet, bei der Kategorie „Funktionaler Kontext“ sogar nur knapp über 40 Prozent, wobei bei letzterer Kategorie im Unterschied zu den übrigen

Kategorien die Angabe „Nein“ deutlich öfter genannt wurde als „Weiß nicht“. Durchaus denkbar ist hierbei, dass die Vorkenntnisse im Bereich Fachterminologie der Studierendengruppe unterschiedlich waren, insbesondere mit Bezug auf die verschiedenen Programmtypen. Falls sich einzelne Probandinnen privat oder im Rahmen des Studiums bereits mit einem oder mehreren dieser Themengebiete beschäftigt haben, verfügten sie vielleicht über Vorkenntnisse im Bereich Fachterminologie, andere wiederum nicht.

Die weiteren Auswertungen bestätigen die Erkenntnisse zur Recherche: In der Kategorie „Kein Kontext“ wird in nur sehr wenigen Fällen gar kein spontaner Übersetzungsvorschlag angegeben, bei der Kategorie „Funktionaler Kontext“ liegt dieser Wert dagegen nahe bei 40 Prozent, während gleichzeitig bei nur knapp 50 Prozent der Fälle ein spontaner Übersetzungsvorschlag angegeben wird, was dem klar niedrigsten Wert aller Kategorien entspricht. Wenn aber kein spontaner Übersetzungsvorschlag angegeben wird, muss zwangsläufig eine Recherche erfolgen, um einen Übersetzungsvorschlag anbieten zu können. Diese Werte entsprechen also den Verhältnissen bei der Antwort auf die Frage, ob eine Recherche durchgeführt wurde.

Umgekehrt zeigen sich bei Auswertung im Hinblick auf die Frage, ob der spontane mit dem endgültigen Übersetzungsvorschlag identisch ist, die meisten Veränderungen zwischen spontanem und endgültigem Übersetzungsvorschlag in der Kategorie „Funktionaler Kontext“. Auch hieran lässt sich ablesen, dass in vielen Fällen eine Recherche durchgeführt wurde und diese auch einen Effekt dahingehend hatte, dass Änderungen am spontanen Vorschlag vorgenommen wurden.

Aufgrund dieser Ergebnisse können also beide Nullhypothesen zu Recherche und Rechercheerfolg für den in dieser Studie durchgeführten Übertest zurückgewiesen werden: Bei den vier Kontextkategorien zeigt sich unterschiedlicher Recherchebedarf und die durchgeführten Recherchevorgänge werden je nach Kategorie von den Probandinnen als unterschiedlich erfolgreich betrachtet.

6.5.2 Übersetzungsalternativen

In Bezug auf die Anzahl der Übersetzungsalternativen wurde folgende Nullhypothese überprüft:

- Für alle Strings der drei Kontextarten gibt es jeweils eine eindeutige Lösung, die die Mehrzahl der Probandinnen angibt.

Die Werte für den Anteil der Strings, bei dem mindestens zehn identische endgültige Übersetzungsvorschläge vorliegen, bilden einen ersten Hinweis zur

Bewertung dieser Hypothese. In der Gesamtbetrachtung trifft dies auf etwa 65 Prozent der Strings zu. Auch hier zeigen sich für die Kontextarten aber Unterschiede: Die Kategorien „Kein Kontext“ und „Fachlicher Kontext“ mit knapp über bzw. knapp unter 90 Prozent der Fälle weisen die höchsten Werte auf, was sich im ersteren Fall dadurch erklären lässt, dass die Strings wenig übersetzerische Probleme enthalten (s. hierzu auch die in Abschnitt 6.5.3 diskutierten Beispiele), und im letzteren dadurch, dass die Recherche in vielen Fällen vermutlich in denselben Terminologiedatenbanken erfolgte und somit häufig identische Übersetzungen erbrachte. In der Kategorie „Situativer Kontext“ wurden bei lediglich 50 Prozent der AT-Strings 10 identische Übersetzungsvorschläge angegeben und in der Kategorie „Funktionaler Kontext“ sogar nur bei etwas mehr als 30 Prozent. Diese Unterschiede bestätigen sich durch die Berechnung des Quotienten aus der Anzahl der endgültigen Übersetzungsvorschläge geteilt durch die Anzahl der unterschiedlichen Übersetzungsalternativen. Bei den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ zeigen sich auch hier die höchsten Werte, wobei erstere die Höchstwerte bietet, während die Kategorie „Situativer Kontext“ und insbesondere die Kategorie „Funktionaler Kontext“ deutlich darunter liegen.

Die Berechnungen zur Anzahl der Übersetzungsalternativen bestätigen dieses Bild: Bei den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ ergeben sich die niedrigsten Werte, sowohl bei den spontanen als auch bei den endgültigen Übersetzungsvorschlägen, und es kommen durch die Recherche nur sehr wenige neue Vorschläge hinzu. Daran lässt sich erkennen, dass in vielen Fällen der oder die tauglichsten Übersetzungsvorschläge bereits zu Beginn bekannt ist und die weitere Recherche ebenfalls zu diesen Vorschlägen führt bzw. sie bestätigt, sofern überhaupt Recherche erforderlich ist.

Bei der Kategorie „Situativer Kontext“ ist die Anzahl der Übersetzungsalternativen zunächst ein ganzes Stück höher, sinkt aber durch die Recherche und bei der Recherche werden ebenfalls nur recht wenige neue Übersetzungsvorschläge generiert. Die Übersetzungsprobleme und die damit einhergehenden Übersetzungsalternativen sind hier also ebenso häufig zu Beginn bekannt, die Recherche ist aber weniger effektiv darin, die Anzahl der Übersetzungsalternativen weiter zu verringern als bei der Kategorie „Fachlicher Kontext“. Weitere Anhaltspunkte hierzu liefern die in Abschnitt 6.5.3 erläuterten Beispiele.

Dem umgekehrten Bild, dass sich bei der Kategorie „Funktionaler Kontext“ zeigt, liegen die größeren Übersetzungsprobleme dieser Kategorie und die ebenfalls wenig effektive Recherche zugrunde. Diese Gründe führen dazu, dass im Vergleich wenige spontane Übersetzungsvorschläge genannt werden, wobei sich hier schon viele Übersetzungsalternativen zeigen, und dass durch die Recherche die Anzahl an Übersetzungsalternativen zumindest im Hin-

blick auf Interquartilsabstand und Standardabweichung zunimmt, wobei der Median allerdings unverändert bleibt. Auch hier lohnt sich ein Blick auf die in Abschnitt 6.5.3 vorgestellten Beispiele.

Damit zeigen sich in dieser Untersuchung für die vier Kontextkategorien unterschiedliche Varianzen, die für die vier Kontextkategorien in unterschiedlichem Maße durch Recherche beeinflusst werden. Abweichend von Rydnings und Lachauds Erkenntnissen (vgl. Rydning und Lachaud 2010, S. 99) ändert sich durch die Recherche nicht in allen Fällen die Varianz dahingehend, dass die Anzahl der Übersetzungsalternativen zunimmt. In drei der vier Kontextkategorien nimmt sie ab, in einer, der Kategorie „Funktionaler Kontext“, zwar nicht hinsichtlich des Medians, aber hinsichtlich des Interquartilsabstands und der Standardabweichung zu. Ebenso zeigt sich hier eine deutlich höhere Zahl an durch die Recherche neu hinzugekommenen Übersetzungsalternativen als bei den anderen Kategorien.

Diese Unterschiede lassen zwei Rückschlüsse zu. Der erste relevante Aspekt ist die Fachtextlichkeit: Diese bringt die Anforderung nach einheitlicher Übersetzung der Fachterminologie mit sich, wobei die Nutzung von Terminologiedatenbanken zu einer geringeren Anzahl an Übersetzungsalternativen führt. Ein zweiter Aspekt ist die Frage, welche Art von Kontext für die Übersetzung relevant und damit bei der Recherche nötig ist. Man könnte vermuten, dass die Anzahl der Übersetzungsalternativen noch mehr zurückgehen würde, wenn für alle Kontextkategorien geeignete Kontextinformationen bereitgestellt werden würden. Auch hier geht also im Unterschied zu den Erkenntnissen von Rydning und Lachaud vermutlich eher nicht mit mehr Kontextinformationen eine größere Varianz einher. Speziell die Ergebnisse für die Kategorie „Funktionaler Kontext“ widersprechen den Erkenntnissen von Rydning und Lachaud: Bei den hier annotierten Features ist Polysemie durchaus eine relevante Größe, wenn man die einzelnen Features `homonym_it-term`, `metaphorik` und `unklare_bedeutung` sowie die damit einhergehenden Übersetzungsprobleme (vgl. beispielsweise K.-D. Schmitz 2005b, S. 42, vgl. Drewer und Ziegler 2014, S. 178–179, Esselink 2000, S. 63, und K. J. Dunne 2009, S. 202) betrachtet. Wenn davon ausgegangen wird, dass die verfügbaren Recherchequellen bei dieser Kategorie nicht die relevanten Kontextinformationen enthalten, also die Probandinnen weiterhin ohne Kontext übersetzt haben, wirken die Ergebnisse überraschend. Allerdings kann ein wichtiger Faktor hierfür sein, dass im vorliegenden Übersetzungstest nicht einzelne Wörter überprüft wurden, sondern Strings aus zwei bis drei Wörtern, bei denen sich zur Polysemie noch weitere Aspekte gesellen, die für die Bedeutung relevant sind.

Auch im Hinblick auf die Anzahl der Übersetzungsalternativen zeigt sich also, dass die Kontextarten große Unterschiede aufweisen. Somit kann auch

die diesbezügliche Nullhypothese für den in dieser Studie durchgeführten Übersetzungstest zurückgewiesen werden.

6.5.3 Kontextkategorien

Die Ergebnisse des Übersetzungstests lassen sich für die vier Kontextkategorien zu folgenden Profilen konkretisieren:

Bei Strings der Kategorie „Kein Kontext“ gaben die Probandinnen in den allermeisten Fällen einen, in 10 Prozent der Fälle zwei und in nicht einmal 3 Prozent der Fälle keinen spontanen Übersetzungsvorschlag an und führten weit überwiegend keine Recherche durch. In den wenigen Fällen, in denen sie eine Recherche durchgeführt haben, war diese in der deutlichen Mehrzahl der Fälle nach ihrer Einschätzung erfolgreich. Umgekehrt gab es damit also eine sehr geringe absolute Anzahl an Recherchevorgängen, die von den Probandinnen als nicht erfolgreich betrachtet wurden. Gegenüber dem ersten spontanen Übersetzungsvorschlag wurden entsprechend in nur wenigen Fällen noch Änderungen vorgenommen. Gleichzeitig war die Anzahl der Übersetzungsalternativen nur sehr gering: Bei über 90 Prozent der Strings aus der Kategorie „Kein Kontext“ waren mindestens 10 identische Übersetzungsvorschläge vorhanden und der niedrige Median von 4,0 Übersetzungsalternativen bei den spontanen Übersetzungen sank bei den endgültigen Übersetzungen noch. Gleichzeitig kamen bei den endgültigen Übersetzungen kaum weitere Übersetzungsalternativen hinzu, wobei Interquartilsabstand und Standardabweichung ebenfalls sanken.

Ein gutes Beispiel hierfür ist String 2 aus dem Übersetzungstest, „File type“: Alle Probandinnen gaben für diesen String mindestens einen spontanen Übersetzungsvorschlag an, wobei insgesamt drei Übersetzungsalternativen vorgeschlagen wurden, „Dateityp“, „Dateiformat“ und „Dateiart“. 18 Probandinnen nannten dabei „Dateityp“ als einzige spontane Übersetzungsalternative bzw. als eine ihrer spontanen Übersetzungsalternativen, nur zwei Probandinnen gaben spontan zwei bzw. drei Übersetzungsalternativen an. Zwei Probandinnen recherchierten erfolgreich, die übrigen recherchierten nicht. „Dateityp“ mit 17 Nennungen und „Dateiformat“ mit drei Nennungen waren die beiden endgültigen Übersetzungsalternativen. Die Probandinnen hatten beim Übersetzen des Strings also weit überwiegend keine Probleme, die beiden Wörter „file,“ und „type“ waren ihnen vertraut, ersteres als gängiger IT-Terminus wahrscheinlich schon aus privater oder schulischer Arbeit am Computer mit der gängigen Übersetzung „Datei“. Bei „type“ wurden verschiedene Übersetzungsalternativen vorgeschlagen, „Typ“ und „Art“ als unterschiedliche Benennungen im Deutschen und „Format“ als mögliche Konkretisierung der Bedeutung des Kompositums im IT-Kontext. Da bei diesem

Übersetzungstest außer der Nennung der Programmtypen kein weiterer Kontext vorlag, bestätigt sich so augenscheinlich Rydnings und Lachauds bereits genannte Feststellung aus ihrer Studie, dass ohne sprachlichen Kontext weniger Übersetzungsalternativen produziert werden (vgl. Rydning und Lachaud 2010, S. 99), wobei hier allerdings keine polysemen Wörter vorliegen. Anders gesagt: Weitestgehend waren die Strings in der Studie ohne Kontextannotation einfach zu übersetzen und für die Übersetzung waren keine weiteren Informationen erforderlich.

Bei Strings der Kategorie „Situativer Kontext“ haben die Probandinnen in zehn Prozent der Fälle keinen spontanen Übersetzungsvorschlag angegeben, in fast drei Vierteln der Fälle einen und in 13 Prozent zwei. In fast der Hälfte der Fälle wurde eine Recherche durchgeführt, von der wiederum nur fast die Hälfte als erfolgreich eingeschätzt wurde. Auf der Basis der Rechercheergebnisse gab es etwa ein Drittel an Fällen, bei denen der spontane Übersetzungsvorschlag überarbeitet wurde. Der Median der spontanen Übersetzungsalternativen war mehr als doppelt so hoch wie bei der Kategorie „Kein Kontext“, verringerte sich aber bei den endgültigen Übersetzungen, bei denen nur sehr wenige neue Übersetzungsalternativen hinzukamen. Interquartilsabstand und Standardabweichung sanken parallel, die Standardabweichung bewegte sich dabei in etwa auf dem Niveau der Standardabweichung der Kategorie „Kein Kontext“.

Ein Beispiel hierfür ist String 9, „Start script“, bei dem „start“ im Korpus als ambige Wortart (Substantiv und Verb, Feature `verb.inf_oder_subst.sg`) annotiert wurde. Dieser String weist die spontanen Übersetzungsalternativen „Skript starten“, „Skript ausführen“, „Programm ausführen“, „Programm starten“ und „Startskript“ auf, wobei alle Probandinnen mindestens einen spontanen Übersetzungsvorschlag angegeben haben und drei Probandinnen zwei bis vier. Bei den endgültigen Vorschlägen kamen noch zwei hinzu, „Transkription starten“ und „Skriptausführung“. Damit gab es wie beim spontanen Vorschlag auch fünf Übersetzungsalternativen. 10 Probandinnen haben recherchiert, vier davon schätzten die Recherche als erfolgreich ein, zwei als nicht erfolgreich. Drei Probandinnen gaben „Weiß nicht“ an, eine machte keine Angabe. Hier spielte die ambige Wortart nur eine kleine Rolle, Probleme waren eher die deutschen Benennungen für „start“ und „script“.

Ein weiteres Beispiel in dieser Kategorie ist String 29, „Save to“, der im Korpus mit GUI-Bezug über die Präposition (Feature `über_präposition`) annotiert wurde. Bei diesem String wurden die spontanen Übersetzungsalternativen „Speichern unter“, „Speichern in Ordner“, „Speichern in“, „Speichern an“, „Speichern als“ und „Speichern auf“ angegeben. Alle Probandinnen haben mindestens einen spontanen Übersetzungsvorschlag angegeben und eine

Probandin zwei. Zwei Probandinnen haben recherchiert, beide erfolgreich. Von den sechs spontanen Vorschlägen blieben bei der endgültigen Übersetzung noch vier übrig, es kam keiner hinzu.

Die Phänomene, die in der Kategorie „Situativer Kontext“ zusammengefasst sind, nämlich ambige Wortarten, GUI-Bezug und Flexionsendungen, lassen in der Regel wenige Übersetzungsalternativen zu, die sich ohne den multimodalen Kontext nicht lösen lassen, weshalb eine gewisse Grundmenge an Übersetzungsalternativen verbleibt, die teils nicht mit dem Phänomen selbst zu tun haben, sondern wie im ersten Beispiel aus der Verwendung verschiedener Benennungen resultieren, teils aber auch wie im zweiten Beispiel auf das annotierte Phänomen zurückgehen. Anzunehmen ist, dass die Probandinnen im ersten Beispiel oben („Start script“) aufgrund der Übersetzungsalternativen für die Übersetzung von „script“ in vielen Fällen dessen Übersetzung recherchiert haben und nicht die Frage, ob „start“ in diesem String nominal oder verbal verwendet wird. Diese Frage und die Frage, wie im zweiten Beispiel das Wort „to“ übersetzt werden muss, ließen sich aber nicht mit Internetrecherche oder Recherche in Datenbanken beantworten, sondern hätten mit geeigneten Referenzmaterialien, aus denen der multimodale Kontext des jeweiligen Softwareprodukts hervorgeht, gelöst werden müssen. Diese Besonderheiten in dieser Kategorie könnten auch die Ursache dafür sein, dass lediglich in weniger als einem Zehntel der Fälle in der Kategorie „Situativer Kontext“ kein spontaner Übersetzungsvorschlag angegeben wurde, aber in knapp der Hälfte der Fälle recherchiert wurde: Die Übersetzungsprobleme verhindern in vielen Fällen nicht, dass eine spontane Übersetzung angegeben wird, lassen aber auch keine eindeutige Übersetzungslösung zu (s. auch die Auswertungen zur Anzahl der Übersetzungsalternativen in Abschnitt 6.4.4). Die Probandinnen versuchen, dieses Problem über Recherche zu lösen, sind dabei aber in vielen Fällen nicht erfolgreich. Kurz gesagt: Die Strings in der Studie aus der Kategorie „Situativer Kontext“ waren im Grunde einfach zu übersetzen, die letzten Details zum korrekten Übersetzungsvorschlag ließen sich in der Studie aber nicht zweifelsfrei klären.

Damit kann auch gesagt werden, dass die in der Kategorie „Situativer Kontext“ erfassten Phänomene, die in der Literatur zur Softwarelokalisierung beschrieben werden (vgl. beispielsweise Beste 2006, S. 73, zu ambigen Wortarten, Anhang 1 in Melby und Foster 2010, S. 1, zum Bezug auf andere GUI-Elemente und Beste 2006, S. 73, zu allein stehenden Pronomen und Adjektiven), von Relevanz für die Übersetzung von GUI-Strings sind und zu recht in der Literatur verschiedentlich genannt werden. Ebenso ist allerdings auch die Lokaldeixis relevant, die jedoch nicht explizit in der für diese Arbeit rezipierten Literatur zur Softwarelokalisierung, sondern in allgemeinen Werken zur Pragmatik beschrieben wird (vgl. beispielsweise Finkbeiner 2015,

S. 38).

Bei Strings der Kategorie „Fachlicher Kontext“ gaben die Probandinnen in knapp zwanzig Prozent der Fälle keinen spontanen Übersetzungsvorschlag an. Entsprechend wurde in fast 60 Prozent der Fälle eine Recherche durchgeführt, die in etwas mehr als drei Vierteln der Fälle als erfolgreich angesehen wurde. In fast drei Vierteln der Fälle blieb es dennoch beim ursprünglichen Übersetzungsvorschlag. Bei der Recherche wurden also vermutlich in vielen Fällen erste Vorschläge bestätigt, bei denen sich die Probandinnen nicht sicher waren, deren Tauglichkeit sie aber bei der Recherche bestätigen konnten. Nur in einem kleinen Teil der Fälle wurde durch die Recherche eine andere, als besser tauglich erachtete Übersetzungsalternative identifiziert. Dies wird dadurch bestätigt, dass der Median der spontanen Übersetzungsalternativen etwas höher als bei der Kategorie „Kein Kontext“ war und sich wie bei der Kategorie „Situativer Kontext“ bei den endgültigen Übersetzungen verringerte, wobei gleichzeitig nur sehr wenige neue Übersetzungsalternativen hinzukamen. Interquartilsabstand und Standardabweichung sanken parallel, die Standardabweichung wies die niedrigsten Werte aller Kontextkategorien auf.

String 27, „Regular expressions“, kann als Beispiel für diesen Sachverhalt dienen. Dieser String ist im Korpus als Fachterminus annotiert, der aus einer festen Kombination aus Adjektiv bzw. Partizip und Substantiv besteht (Feature `adjektiv-partizip_plus_substantiv-fachterm`). Der String wurde spontan in den Übersetzungsalternativen „Reguläre Ausdrücke“, „Normale Ausdrücke“ und „Regular Expressions“ übersetzt. Drei Probandinnen haben keinen spontanen Übersetzungsvorschlag angegeben, zwei andere zwei. Bei den endgültigen Vorschlägen kam noch eine Übersetzungsalternative hinzu („Gewöhnliche Ausdrücke“), allerdings nutzten alle anderen Probandinnen als endgültigen Übersetzungsvorschlag die Übersetzungsalternative „Reguläre Ausdrücke“, es gab also nur noch zwei Übersetzungsalternativen. 10 Probandinnen haben recherchiert, acht davon schätzten die Recherche als erfolgreich ein, eine Probandin als nicht erfolgreich, eine andere gab die beiden Optionen „Nein“ und „Weiß nicht“ an.

Die an verschiedenen Stellen der Literatur zur Softwarelokalisierung genannte Bedeutung der Terminologiearbeit (vgl. beispielsweise K.-D. Schmitz 2008, S. 267, und Seewald-Heeg 2016, S. 154) wird dadurch im Rahmen dieser Studie bestätigt. Zusammenfassend könnte man damit sagen: Die Probandinnen waren sich bei der Studie in vielen Fällen unsicher bei der korrekten Übersetzung von Fachterminologie in der Kategorie „Fachlicher Kontext“, konnten diese aber mithilfe von Recherche in geeigneten Terminologiedatenbanken überwiegend gut und eindeutig lösen.

Bei Strings der Kategorie „Funktionaler Kontext“ konnten die Proban-

dinnen in fast 40 Prozent der Fälle keinen spontanen Übersetzungsvorschlag angeben, weshalb auch in beinahe drei Vierteln der Fälle recherchiert wurde. Der Erfolg bei dieser Recherche war aber mit etwas mehr als 40 Prozent nur sehr überschaubar, wobei die Probandinnen bei den übrigen Recherchevorgängen überwiegend zu der klaren Einschätzung kamen, dass die Recherche nicht erfolgreich war. Da diese Einschätzung im Unterschied zu den anderen Kategorien häufiger vorkam als die Einschätzung „Weiß nicht“, kann man annehmen, dass den Probandinnen in vielen Fällen klar war, dass sie ein Problem hatten, dass sie allerdings gleichzeitig auch den Eindruck hatten, dass die Recherche ihr Problem nicht lösen konnte. Dennoch wurden in fast der Hälfte der Fälle noch Änderungen am ursprünglichen Übersetzungsvorschlag vorgenommen, sofern denn einer vorlag. Ergänzt werden diese Erkenntnisse dadurch, dass der Median der spontanen Übersetzungsalternativen deutlich höher als bei den anderen Kategorien lag und für die spontanen und endgültigen Übersetzungen unverändert blieb, wobei gleichzeitig Standardabweichung und Interquartilsabstand im Unterschied zu den anderen Kontextkategorien noch anstiegen. Es erfolgte also keine mehr oder weniger starke Eingrenzung durch die Recherche, sondern im Gegenteil nahm die Anzahl der Übersetzungsalternativen noch zu.

Betrachtet man String 14, „Save to pocket“, bei dem das Wort „pocket“ im Korpus mit dem Feature *metaphorik* annotiert ist, lassen sich die acht Übersetzungsalternativen „In Pocket speichern“, „Für später speichern“, „In Tasche speichern“, „Bei Pocket speichern“, „In der Tasche speichern“, „Speichern“, „In Zwischenablage speichern“ und „In die Zwischenablage kopieren“ unterscheiden, bei denen weitgehend die Frage der Übersetzung von „pocket“ im Fokus ist. Sieben Probandinnen haben keinen spontanen Übersetzungsvorschlag angegeben, eine Probandin drei. Bei den endgültigen Vorschlägen zeigt sich eine neue Übersetzungsalternative („Speichern in Pocket“) und es sind insgesamt fünf Übersetzungsalternativen vorhanden, von denen eine, „In Pocket speichern“, von 16 Probandinnen gewählt wurde. Dies könnte daran liegen, dass „Pocket“ damals eine Funktion im Browser Firefox war, die die Probandinnen teils möglicherweise kannten und die auch im Internet recherchierbar war.⁷ Dies ist also ein gutes Beispiel für die Prägung neuer Fachterminologie für neue Funktionen, für die auch in der Zielsprache neue Benennungen gefunden werden müssen (vgl. Drewer und Ziegler 2014, S. 177).

Bei Strings 48, „On losing focus“, der im Korpus mit dem Feature *unklare_bedeutung* annotiert ist, ist die Lage etwas anders: Bei den spontanen Übersetzungsvorschlägen zeigen sich 11 Übersetzungsalternativen, von

⁷Inzwischen wurde diese Funktion aber wieder eingestellt, siehe <https://support.mozilla.org/de/kb/Zukunft-von-Pocket> (letzter Zugriff: 15.08.2025).

denen nur eine von zwei Probandinnen gewählt wurde und bei der sich eine große Bandbreite zeigt von „Bei Fokusverlust“ über „Über Konzentrationsverlust“ bis hin zu „Wenn nicht mehr aktiv“. 11 Probandinnen machen keine Angabe, zwei Probandinnen machen zwei Vorschläge, eine drei. Nach der Recherche zeigen sich 10 neue Übersetzungsalternativen bei insgesamt 14 Übersetzungsalternativen, nur fünf dieser Übersetzungsalternativen wurden von mehr als einer Probandin gewählt, wobei zwar das Wort „Fokus“ (bzw. „focus“) in allen Übersetzungsalternativen vorkommt, aber immer noch Unterschiede deutlich werden, wie beispielsweise an den Übersetzungsalternativen „Fokus entfernen“, „Über Verlieren des Fokus“ und „Wenn der Fokus verloren wird“ sichtbar wird. 17 Probandinnen haben recherchiert, nur fünf betrachteten die Recherche als erfolgreich.

Diese Ergebnisse für die Kategorie „Funktionaler Kontext“ lassen sich so zusammenfassen, dass es bei den Probandinnen in vielen Fällen eine große Unsicherheit hinsichtlich der korrekten Übersetzung für Strings aus dieser Kategorie gab, die in vielen Fällen dazu führte, dass ohne Recherche keine Lösung angeboten werden konnte. Die Recherche in den verfügbaren Quellen hat wenige Erkenntnisse gebracht und konnte die Unsicherheit nicht zerstreuen. Diese Unsicherheit zeigt sich in einer großen Zahl an Übersetzungsalternativen. Dies lässt sich dadurch begründen, dass die Probandinnen die Bedeutung der Strings häufig nicht verstanden haben, wodurch sich Esselinks und Dunnes Annahmen im Rahmen der Studie bestätigten, dass für den von Dunne als „Telegraphic Style“ bezeichneten Problembereich (vgl. K. J. Dunne 2009, S. 202) detaillierte Kenntnisse über das Produkt, aber teils auch die Programmierung oder den weiteren Kontext erforderlich sind (vgl. Esselink 2000, S. 63). Gleichzeitig zeigt sich in dieser Kategorie wie bereits in Abschnitt 6.5.2 erläutert die größte Abweichung zu Rydnings und Lachauds Erkenntnissen im Hinblick auf die Übersetzung polysemer Wörter mit und ohne sprachlichen Kontext (vgl. Rydning und Lachaud 2010, S. 99).

Basierend auf diesen Einzelerkenntnissen zu den vier Kontextkategorien kann man abschließend zusammenfassend sagen, dass die Ergebnisse der Studie einen deutlichen Zusammenhang zwischen den drei Aspekten Kontextart, Recherche(erfolg) und Anzahl der Übersetzungsalternativen zeigen.

6.6 Einschränkungen und offene Fragen des Übersetzungstests

Aus der Konzeption und Durchführung der Studie ergeben sich ein paar Einschränkungen, die bei der Bewertung der Ergebnisse berücksichtigt werden

müssen.

Eine dieser Einschränkungen liegt in der Probandinnengruppe dieser Studie, nämlich Studierenden im Master Translation. Die Frage der Erfahrung spielt für Studien mit Probandinnen im Bereich der Translation eine große Rolle, allerdings werden aus Gründen der Verfügbarkeit in zahlreichen Studien Studierende aus translationswissenschaftlichen Studiengängen als Probandinnen genutzt (vgl. Hansen-Schirra und Nitzke 2021). Die für die vorliegende Studie ausgewählte Gruppe an Probandinnen war zwar recht homogen, es fehlte ihr aber die professionelle Erfahrung im Bereich Translation allgemein und insbesondere auch die professionelle Erfahrung mit der speziellen Textsorte GUI-Text. Es ist davon auszugehen, dass diese Erfahrung bei der Übersetzung der GUI-Strings eine wichtige Rolle spielt und sich daher auch auf die Ergebnisse des Übersetzungstests auswirken würde. In der Eye-Tracking- und Key-Logging-Studie, die im Anschluss an diese Studie durchgeführt wurde (s. Kapitel 7), wurden daher professionelle Übersetzerinnen aus dem Bereich der Softwarelokalisierung als Probandinnen gesucht.

Ein weiteres Problem liegt in der Konzeption der Studie als Excel-Tabelle mit einer festen Reihenfolge der Strings. Es besteht die Möglichkeit, dass Probandinnen im Verlauf der Studie ihre Herangehensweise ändern, beispielsweise in Bezug auf die Frage, welche Strings sie auf welche Weise und mit welchem Zeitaufwand recherchieren. O'Brien und Saldanha bezeichnen dies als „Carry-over Effect“, dem durch Randomisierung der Reihenfolge der Aufgaben entgegengewirkt werden kann (vgl. Saldanha und O'Brien 2013, S. 114). Da eine solche Randomisierung nicht vorgenommen wurde und gleichzeitig die Zeit für die Bearbeitung des Übersetzungstests sehr knapp bemessen war, kann es beispielsweise sein, dass im letzten Teil des Tests weniger (gründlich) recherchiert wurde und weniger Alternativen eingegeben wurden. Da die verschiedenen Kontextarten aber über den ganzen Übersetzungstest hinweg verteilt waren, wirkt sich dieser Effekt vermutlich auf alle gleichermaßen aus.

Ein weiterer Aspekt, der bei dieser Studie problematisiert werden könnte, ist die ökologische Validität, die sich auf die Anforderung bezieht, dass Studien die professionelle Realität von Probandinnen widerspiegeln (vgl. Saldanha und O'Brien 2013, S. 33). In diesem Fall könnte das beispielsweise die Nutzung von Softwarelokalisierungstools oder auch die Übersetzung fortlaufender Ausschnitte von Ressourcendateien sein. Da jedoch bei der vorliegenden Studie lediglich Studierende mit wenig oder keiner Erfahrung im Bereich Softwarelokalisierung als Probandinnen dienten, zeigen sich keine Einschränkungen hinsichtlich der ökologischen Validität. Zudem ermöglichte das Design der Studie als kontrolliertes Experiment und damit die Manipulation der zu übersetzenden Strings mit Einbezug von vier Kontextkategorien in größerer

Zahl eine höhere Zuverlässigkeit im Vergleich zu einem möglichst professionellen Setting, was wiederum zu aussagekräftigen Ergebnissen führte. Diese Aussagekraft wurde nämlich nicht dadurch eingeschränkt, dass die zu überprüfenden Phänomene in einem längeren Ausschnitt einer Ressourcendatei nur vereinzelt vorkamen (vgl. Heilmann u. a. 2019, S. 100).⁸

Wie in Abschnitt 6.2 beschrieben, wurden für die Studie lediglich Strings aus zwei oder drei Wörtern ausgewählt. Dadurch konnte nur ein Teil der im vorangehenden Kapitel herausgearbeiteten Aspekte der verschiedenen Kontextarten bzw. der Features aus dem Annotationsschema im Übersetzungstest berücksichtigt werden. Insbesondere die fachterminologischen und anderen Aneinanderreihungen konnten praktisch nicht in die Studie integriert werden, weil die zugehörigen Strings nahezu ausnahmslos länger als drei Wörter waren. Um alle Features mit einzubeziehen, hätte der Test umfangreicher sein und mit anderen Beschränkungen erstellt werden müssen.

Neben der bereits genannten fehlenden Erfahrung und Professionalität der Probandinnen blieben noch weitere Fragen offen, insbesondere hinsichtlich des Nutzens anderer Referenzmaterialien, vor allem solcher, die im Rahmen von Softwarelokalisierungsprojekten zur Verfügung stehen, aber auch hinsichtlich der Anzahl der Übersetzungsalternativen und darauf basierend der Entropie, die in der Translationsprozessforschung als einer der Indikatoren für kognitiven Aufwand identifiziert wurde. Alle diese Aspekte wurden in der Eye-Tracking- und Key-Logging-Studie untersucht, die im folgenden Kapitel vorgestellt wird.

⁸Siehe hierzu auch Abschnitt 7.8, in dem dieser Punkt im Hinblick auf die Eye-Tracking- und Key-Logging-Studie nochmals aufgegriffen wird, und Heilmann et al., die sich im Zusammenhang mit Ausgangstextlänge mit dem Thema der ökologischen Validität befassen (vgl. Heilmann u. a. 2019).

Kapitel 7

Recherche und kognitiver Aufwand bei der Übersetzung der Kontextarten

Nachdem im Übersetzungstest in Kapitel 6 die Unterschiede der vier Kontextkategorien hinsichtlich Recherche und Anzahl der Übersetzungsalternativen ermittelt wurden, sollen nun weitere Informationen zur Recherche gesammelt und gleichzeitig überprüft werden, ob sich parallel zu den Unterschieden in der Anzahl der Übersetzungsalternativen auch Unterschiede im kognitiven Aufwand zeigen. Für diese prozessorientierte Herangehensweise, bei der Verhalten und kognitive Prozesse der Übersetzerinnen untersucht werden sollen (vgl. Saldanha und O'Brien 2013, S. 109), wurde eine Eye-Tracking- und Key-Logging-Studie konzipiert, deren Planung, Durchführung und Ergebnisse in diesem Kapitel vorgestellt werden.¹ Dazu enthält Abschnitt 7.1 Informationen zu den zu untersuchenden Hypothesen. In den anschließenden Abschnitten finden sich Informationen zu Konzeption (Abschnitt 7.2) sowie zu Ablauf und verwendeten Zusatzmaterialien für die Eye-Tracking- und Key-Logging-Untersuchung (Abschnitt 7.3). Der darauf folgende Abschnitt 7.4 enthält Informationen zu den Probandinnen, Abschnitt 7.5 zur Durchführung der Untersuchung. In Abschnitt 7.6 finden sich Informationen zur Auswertung der Studie, unterteilt nach den Schritten zur Vorbereitung der erhobenen Daten für die Auswertung, dem Nutzen der Zusatzmaterialien und den Ergebnissen aus Eye-Tracking und Key-Logging. In Abschnitt 7.7 erfolgt eine Interpretation dieser Ergebnisse und der abschließende Abschnitt 7.8 enthält

¹Vgl. dazu Jiménez-Crespo, der als einen der zahlreichen noch zu untersuchenden translationswissenschaftlichen Forschungsbereiche in der Lokalisierung explizit kognitive Übersetzungsprozesse nennt, die mittels Eye-Tracking und Key-Logging untersucht werden (vgl. Jiménez-Crespo 2024, S. 279).

Einschränkungen und Probleme dieser Studie.

7.1 Hypothesen der Eye-Tracking- und Key-Logging-Studie

In den folgenden Abschnitten werden zwei Hypothesen herausgearbeitet, eine zum Referenzmaterial in Abschnitt 7.1.1 und eine zum kognitiven Aufwand in Abschnitt 7.1.2.

7.1.1 Hypothese zum Referenzmaterial

Beim Übersetzungstest (s. Abschnitt 6.4.3) zeigten sich Unterschiede sowohl bei der Frage, ob bei Strings der vier Kontextkategorien Recherche als notwendig erachtet wurde, als auch bei der Frage, ob die Recherche in den im Test verfügbaren Internetquellen als erfolgreich betrachtet wurde. Die typischen und in der Literatur zur Softwarelokalisierung genannten Referenzmaterialien, die bereits in Abschnitt 4.5 vorgestellt wurden und in folgender Liste enthalten sind, waren in diesem Test allerdings nicht verfügbar:²

- Eine lauffähige Version der zu übersetzenden Software (vgl. Ottmann 2005, S. 102, und Esselink 2000, S. 53) bzw. Simulationen und Demos, falls dies nicht möglich ist (vgl. Ottmann 2005, S. 102)
- Die Dokumentation des Softwareprodukts in Form von Hilfeinformationen bzw. Onlinehilfe (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, und Wahle 2000b, S. 45) oder auch Referenzhandbücher, die Beschreibungen der einzelnen Strings enthalten (vgl. Ottmann 2005, S. 104)
- In den zu übersetzenden Dateien enthaltene Informationen wie Kommentare der Entwicklerinnen und weitere Informationen in den Ressourcendateien (vgl. Esselink 2000, S. 34, 64–65, Wahle 2000b, S. 37–38, 40, 45, Reineke 2005, S. 85, und Sachse 2005, S. 147–148, 158)
- Verschiedene Glossare, wie etwa Terminologie für die relevanten Betriebssysteme, die Software oder das Fachgebiet (vgl. Ottmann 2005, S. 104, und Esselink 2000, S. 53, 401–402)

²Im Übersetzungstest war lediglich Internetrecherche möglich und die genauen Softwareprodukte, aus denen die zu übersetzenden Strings stammten, waren nicht bekannt. Dadurch waren nur allgemeine und in Internet verfügbare Recherchequellen nutzbar, die von den im Folgenden genannten Referenzmaterialien lediglich Terminologie abdeckten, beispielsweise Terminologie für Betriebssysteme in Form der Microsoft-Terminologiedatenbank (vgl. *Microsoft Terminology* 2025) oder auch Terminologie des jeweiligen Fachgebiets.

- Hintergrundinformationen zum Produkt (vgl. Esselink 2000, S. 53)

Darüber hinaus wird die Nutzung von Softwarelokalisierungstools, in denen die Lokalisierung in Menüs und Dialogfeldern erfolgt (s. Abschnitt 2.1.4), insbesondere in Bezug auf den situativen Kontext in der Literatur als vorteilhaft angesehen (vgl. z. B. Ottmann 2005, S. 102–103, Beste 2006, S. 93–94, Wahle 2000b, S. 37–38, 40, K.-D. Schmitz 2005a, S. 9, und Schildhauer 2005, S. 124). Die Nutzung solcher Tools für die Lokalisierung kann allerdings als Alternative zur Recherche in der lauffähigen Version der zu übersetzenden Software betrachtet werden, da der Nutzwert jeweils derselbe ist: Der situative Kontext, also in welchen GUI-Bereichen und mit welchen umgebenden Strings und GUI-Elementen der zu übersetzende String angeordnet ist, kann ermittelt werden und daraus können dann Rückschlüsse auf die korrekte Übersetzung gezogen werden.

Zwei der hier aufgeführten Referenzmaterialien wurden für die vorliegende Studie als typische Materialien für Lokalisierungsprojekte ausgewählt, sowohl weil sie verschiedentlich in der Literatur genannt werden als auch weil sie, wie in Abschnitt 4.5 dargestellt, Nutzen für alle Kontextarten haben können. Zum einen ist dies die lauffähige Version bzw. eine Simulation oder ein Demo bzw. die Übersetzung in einem Softwarelokalisierungstool, zum anderen die Dokumentation des zu übersetzenden Softwareprodukts. Erstere wurde in dieser Studie in Form von Screenshots der GUI des zu übersetzenden Programms nachgestellt, in denen der zu übersetzende String enthalten ist, letztere in Form von textlichen Beschreibungen der zu übersetzenden Strings (s. Abschnitt 7.3 unten). Dabei ist anzunehmen, dass die Probandinnen den Nutzen dieser beiden Referenzmaterialien in Bezug auf die Übersetzung unterschiedlich einschätzen. Entsprechend ergibt sich folgende Nullhypothese für das Referenzmaterial:

- Der Nutzen von Screenshots und alternativ textlichen Beschreibungen ist für alle Strings der vier Kontextkategorien gleichermaßen groß.

7.1.2 Hypothese zum kognitiven Aufwand

Den Ausgangspunkt für die weitere Hypothese bildet der kognitive Aufwand (*cognitive effort*³) beim Übersetzungsprozess, der in zahlreichen Studien aus

³Verwiesen sei hier auf Wei, der feststellt, dass die Termini „cognitive load“, also kognitive Belastung, und „cognitive effort“, also kognitiver Aufwand, häufig synonym verwendet werden. Er hingegen definiert die kognitive Belastung als die Schwierigkeit, die durch eine Aufgabe oder einen Prozess entsteht, also die erforderliche Menge an kognitivem Aufwand, und den kognitiven Aufwand als den tatsächlichen Aufwand, der bei der Aufgabe oder dem

dem Bereich der Translationsprozessforschung im Fokus steht (in jüngerer Zeit beispielsweise in den Sammelbänden Carl, Bangalore und Schaeffer 2016, Jakobsen und Mesa-Lao 2017 und Carl 2021; weitere Veröffentlichungen werden im Folgenden genannt).⁴ Als Ausgangspunkt für die Frage des Aufwands nimmt Krings eine Unterteilung in zeitlichen, kognitiven und technischen Aufwand vor (vgl. Krings 2001, S. 178), wobei sich seine Darstellung wie auch diverse der im Folgenden zitierten Veröffentlichungen zwar speziell auf das Postediting beziehen, sich allerdings auch auf übersetzerische Prozesse übertragen lässt. Der zeitliche Aufwand ist für ihn die Zeit, die für das Beheben von Defiziten und das Herstellen einer Qualität erforderlich ist, die mit der einer Humanübersetzung vergleichbar ist (vgl. Krings 2001, S. 178–179). Bei der Übersetzung wäre das die Zeit, die für die Herstellung der Übersetzung nötig ist. Der kognitive Aufwand setzt sich nach Krings aus Typ und Umfang der kognitiven Prozesse zusammen, die zum Beheben eines Defekts in der MÜ aktiviert werden müssen (vgl. Krings 2001, S. 179), bei der Übersetzung entsprechend die kognitiven Prozesse bei der Entwicklung einer Übersetzungslösung. Der technische Aufwand umfasst laut Krings das Löschen, Einfügen und Umstellen einer MÜ-Lösung, nachdem Defizite erkannt und deren Behebung geplant wurden (vgl. Krings 2001, S. 179). Beim Übersetzen entspräche dies der eigentlichen Eingabearbeit der Übersetzung über die Computertastatur.

Der kognitive Aufwand kann durch Ermitteln der Augenbewegungen mithilfe eines Eye-Trackers erfasst werden, z. B. über die Dauer der Fixationen, also der Zeit, für die das Auge auf ein bestimmtes Wort fokussiert bleibt: Je länger dieser Zeit dauert, desto höher ist der kognitive Aufwand (vgl. Schaeffer, Nitzke und Hansen-Schirra 2019, S. 3). Neben dieser Fixationsdauer stellen auch die Anzahl der Fixationen pro Wort, Satz oder Text Messwerte für die kognitive Verarbeitung dar (vgl. Schaeffer, Nitzke und Hansen-Schirra 2019, S. 3). Dragsted stellt solche Zusammenhänge zwischen Produkt- und Prozessdaten erstmalig mit Blick auf ganze Texte fest. Speziell für die Prozessdaten Gesamtlesezeit (*Total Reading Time*) für AT-Wörter, die Anzahl der Fixationen auf AT-Wörter und Pausen bei der ZT-Produktion ermittelt

Prozess zum Einsatz kommt, wobei dieser Aufwand durch Optimierung der Zuweisung beschränkter kognitiver Ressourcen realisiert wird (vgl. Wei 2022, S. 76). Letzteres ist für die vorliegende Studie relevant, weshalb im weiteren Verlauf der Terminus „kognitiver Aufwand“ verwendet wird.

⁴Dabei fällt allerdings auf, dass sich die Translationsprozessforschung nur in wenigen Fällen mit Fachtexten, also dem Themenbereich der vorliegenden Studie, befasst, was sich im weiteren Verlauf an den rezipierten Arbeiten zeigen wird. Ausnahmen bilden beispielsweise Czulo und Nitzke 2016 und Czulo, Hansen-Schirra und Nitzke 2017, die aber beide für die Fragen der vorliegenden Studie inhaltlich nicht relevant sind.

sie hochsignifikante Korrelationen zu den Produktdaten: Bei Wörtern mit höherer Varianz, also bei ambigeren Wörtern, die von verschiedenen Übersetzerinnen unterschiedlich übersetzt werden, erhöhen sich bei dieser Studie all diese Werte (vgl. Dragsted 2012).

Die Nutzung solcher Verhaltensdaten ermöglicht es der Translationsprozessforschung, kognitive Prozesse sowie den kognitiven Aufwand empirisch zu untersuchen, zu messen und zu quantifizieren und damit auch Methoden und Modelle für Vorhersagen zu Qualität, Effizienz und Produktivität zu entwickeln (vgl. Schaeffer, Nitzke und Hansen-Schirra 2019, S. 5). Schaeffer et al. beschreiben dies als Wende hin zu Vorhersagen, als *predictive turn*, in der Übersetzungswissenschaft (vgl. Schaeffer, Nitzke und Hansen-Schirra 2019).

Mishra et al. arbeiten einen Zusammenhang zwischen kognitivem Aufwand und Übersetzungsschwierigkeit heraus. Dazu entwickeln sie den *Translation Difficulty Index* (TDI), der sich aus der Summe der Fixationen und Sakkaden zusammensetzt und mit drei Aspekten des AT-Satzes, nämlich Länge in Wörtern, Grad der Polysemie der einzelnen Wörter und strukturelle Komplexität des Satzes, in Beziehung steht (vgl. Mishra, Bhattacharyya und Carl 2013), was teils an die Varianz bzw. Ambiguität bei Dragsted (vgl. Dragsted 2012) anknüpft. Der TDI lässt sich allerdings nicht auf die vorliegende Untersuchung anwenden: Da jeweils nur Zwei- und Drei-Wort-Strings verwendet werden, sind die Unterschiede in der Länge der Items sehr gering und syntaktische Komplexität ist kaum vorhanden, der Fokus liegt also stark auf der Polysemie der Wörter. Diese Einschränkung des TDI, nur Vorhersagen auf Satzebene und nicht auf niedrigeren Ebenen wie Phrasen und Wörtern zu liefern, wird auch von Schaeffer et al. angesprochen (vgl. Schaeffer, Nitzke und Hansen-Schirra 2019, S. 11).

Tokowicz und Kroll schließen mit ihrer Forschung an Dragsteds oben geschilderte Erkenntnisse an, wobei gleichzeitig die Probleme von Mishra et al. vermieden werden: Sie untersuchen die Übersetzung einzelner Wörter vom Englischen ins Spanische (vgl. Tokowicz und Kroll 2007). Dabei stellen sie fest, dass die Reaktionszeiten von bilingualen Probandinnen beim Übersetzen langsamer sind, wenn die Wörter mehr als eine mögliche Übersetzung haben, als beim Übersetzen einzelner Wörter mit nur einer einzigen Übersetzung. Dies lässt Tokowicz und Kroll vermuten, dass es eine Konkurrenz zwischen verschiedenen Übersetzungsalternativen gibt. Die Auswahl aus diesen Alternativen führt zu höherem kognitiven Aufwand, weil ein Vergleich durchgeführt wird, welche Übersetzung am besten geeignet ist, und anschließend die anderen Alternativen ausgeblendet werden müssen (vgl. Tokowicz und Kroll 2007). Ähnliche Ergebnisse erbrachten auch nachfolgende Untersuchungen von beispielsweise Laxén und Lavaur (vgl. Laxén und Lavaur 2010) und Boada et al. (Boada u. a. 2013) unter Bilingualen für Cognates, also

Wörter mit sehr ähnlicher oder identischer Rechtschreibung und identischer Bedeutung in zwei Sprachen (vgl. Laxén und Lavaur 2010, S. 158), und Non-Cognates.

Carl und Schaeffer überführen diese Erkenntnisse aus der Bilingualenforschung in die Translationswissenschaft und führen dazu zunächst eine operationalisierbare Definition der Übersetzungsliteralität (*Translation Literality*) ein, für die sie drei Kriterien mit Fokus auf dem Übersetzungsprodukt entwickeln: Erstens ist die Reihenfolge der Wörter in AT und ZT identisch, zweitens sind AT- und ZT-Wörter Übersetzungsäquivalente, entsprechen sich also eins zu eins, und drittens gibt es im jeweiligen Kontext für jedes AT-Wort nur eine einzige mögliche Übersetzung (vgl. Carl und Schaeffer 2017a, S. 85).

Bei letzterem Kriterium zeigt sich erneut eine Verwandtschaft zu Dragstedts Übersetzungsambiguität und der von Mishra et al. genannten Polysemie, da mit ihm eine Einschätzung der semantischen Ähnlichkeit eines Wortes in beiden Sprachen einhergeht: „if an ST word is consistently translated in the same way we assume that both languages also have similar semantic representation with respect to this word“ (Carl und Schaeffer 2017a, S. 85). Dieses Konzept, das auf der Anzahl und Verteilung der gewählten Übersetzungen in einem Korpus mit alternativen Übersetzungen beruht, bezeichnen sie als „Word Translation Entropy“ (vgl. Carl und Schaeffer 2017a, S. 85–86), die sie mit dem Messwert „HTra“ berechnen (vgl. Carl, Schaeffer und Bangalore 2016, S. 29–33, und Schaeffer, Dragsted u. a. 2016, S. 191).

Word translation entropy describes the degree of uncertainty regarding which lexical TT item(s) are chosen given the sample of alternative translations for a single ST word: if the probabilities are distributed equally over a large number of items, the word translation entropy is high and there is a large degree of uncertainty regarding the outcome of the translation process. If, however, the probability distribution falls unto just one or a few items, entropy is low and the certainty of the TT item(s) to be chosen is high. (Schaeffer, Dragsted u. a. 2016, S. 191)

Carl und Schaeffer zeigen, dass ihr Messwert für Translation Literality mit dem Übersetzungsaufwand korreliert (vgl. Carl und Schaeffer 2017a, S. 102). Dass konkret auch die Word Translation Entropy als einzelner Aspekt der Translation Literality als Messwert für den kognitiven Aufwand der Übersetzerinnen dienen kann, bestätigen Schaeffer et al. (vgl. Schaeffer, Dragsted u. a. 2016) in Bezug auf die Dauer der ersten Fixation (*First Fixation Duration*, FFDur), indem sie zwischen FFDur und der Word Translation Entropy in Form des Messwerts „HTra“ eine positive und signifikante Korrelation

feststellen. Nach ihren Erkenntnissen ist bei Wörtern, die von verschiedenen Übersetzerinnen unterschiedlich übersetzt werden, die Wahrscheinlichkeit höher, dass die Verarbeitung bereits sehr früh, also während der ersten Fixation, erschwert ist (vgl. Schaeffer, Dragsted u. a. 2016, S. 206). Wenn keine nach Definition der Translation Literality wörtliche Übersetzung möglich ist, weil der vorliegende Kontext, Zielnormen oder andere Ursachen dagegen sprechen, steigt nach ihren Erkenntnissen der kognitive Aufwand (vgl. Schaeffer, Dragsted u. a. 2016, S. 199). Ebenso wurde zuvor schon von Rydning und Lachaud beim Vergleich zwischen professionellen Übersetzerinnen und bilingualen Nicht-Übersetzerinnen festgestellt, dass erstere polyseme Wörter ohne Kontext mit signifikant kleineren FFDur-Werten als die Bilingualen übersetzten (vgl. Rydning und Lachaud 2010, S. 96).

Konkret ist die Word Translation Entropy im Translationsprozess die Menge an ko-aktivierten Translationsalternativen, die alle gleichermaßen taugliche Varianten für die Übersetzung eines AT-Elements sind (vgl. Bangalore u. a. 2016, S. 213). Die Entropie steigt mit zunehmender Variation an (vgl. Bangalore u. a. 2016, S. 213–214). Der Wert der Word Translation Entropy ist also höher, wenn alle Übersetzerinnen unterschiedliche Übersetzungen produzieren, und niedrig, wenn die Übersetzerinnen keine oder nur sehr wenige Übersetzungsalternativen produzieren. Bangalore et al. betonen in diesem Zusammenhang die Frage der Wahrscheinlichkeit, die bereits in der Definition von Schaeffer et al. oben (vgl. Schaeffer, Dragsted u. a. 2016, S. 191) angesprochen wird: Die reine Anzahl verschiedener Übersetzungsalternativen spiegelt nicht direkt den Einfluss der einzelnen Alternativen auf das Verhalten der Übersetzerinnen wider, da es jeweils dominantere und weniger dominante Alternativen geben kann (vgl. Bangalore u. a. 2016, S. 212). Mithilfe des Entropiewerts erfolgt daher eine Gewichtung der Übersetzungsalternativen, nicht nur eine reine Zählung. Die Gewichtung basiert auf der Wahrscheinlichkeit der einzelnen Übersetzungsoptionen, wodurch der kognitive Aufwand besser wiedergegeben werden kann (vgl. Bangalore u. a. 2016, S. 214). Laut Wei geben höhere Entropiewerte für ein Element eine größere Unsicherheit bei der Entscheidungsfindung der Übersetzerinnen an, die eine Auswahl aus den Alternativen treffen müssen. Er nimmt an, dass diese größere Unsicherheit entweder aus der größeren Anzahl an aktivierten Übersetzungsalternativen resultiert oder einen Mangel an sehr wahrscheinlichen Wahlmöglichkeiten aus diesen Alternativen zeigt (vgl. Wei 2021, S. 169).

Der Messwert „HTra“ für die Word Translation Entropy wird zwar häufig im Kontext von Studien zum Postediting verwendet, allerdings weisen Ogawa et al. in ihrer Studie nach, dass die „HTra“-Werte bei maschineller Übersetzung und bei Humanübersetzung bei denselben ATs stark korrelieren (Ogawa, Gilbert und Almazroei 2021). Daraus schließt Wei, dass die Übersetzungsam-

biguität, die über den „HTra“-Wert ermittelt wird und die Schwierigkeit für Humanübersetzung tendenziell erhöht, eine größtenteils ähnliche Auswirkung auf maschinelle Übersetzung und damit auch auf den Postediting-Prozess hat (vgl. Wei 2021, S. 178).

In ihrer Studie befassen sich Schaeffer et al. für die Ermittlung des kognitiven Aufwands neben FFDur auch mit der Verweildauer für AT-Wörter (FPDurS), der Wahrscheinlichkeit einer Regression (Reg), der Dauer des Regressionspfads (RPDur), der Wahrscheinlichkeit einer Fixation oder der Wahrscheinlichkeit des Überspringens sowie der Gesamtlesedauer für den AT (TrtS) (vgl. Schaeffer, Dragsted u. a. 2016; zu den genannten Messwerten s. a. Carl, Schaeffer und Bangalore 2016). Nicht all diese Werte versprechen bei der vorliegenden Studie aussagekräftige Ergebnisse, da sich die hier verwendeten Zwei- und Drei-Wort-Strings stark von vollständigen Sätzen wie bei Schaeffer et al. unterscheiden. Insbesondere die Regressionswerte wurden bei dieser Studie nicht überprüft, da sie stark vom linguistischen Kontext abhängen, der bei der vorliegenden Studie nur sehr eingeschränkt vorhanden ist.

Neben diesen Werten zeigen Carl et al. auch starke Auswirkungen von „HTra“ auf die Dauer der Übersetzung (*Duration*, Dur) und somit ebenfalls Auswirkungen auf den kognitiven Aufwand (vgl. Carl und Schaeffer 2017b, S. 51).

Diese Erkenntnisse zum kognitiven Aufwand aus den genannten Studien können mit den Ergebnissen des Übersetzungstests in Kapitel 6 kombiniert werden: Die vier Kontextkategorien weisen in dieser Studie eine unterschiedliche Anzahl von Übersetzungsalternativen auf (s. Abschnitt 6.4.4), daher kann in Bezug auf den kognitiven Aufwand für die weitere Untersuchung also folgende Nullhypothese als Ausgangspunkt genommen werden:

- Der kognitive Aufwand ist bei allen vier Kontextkategorien gleich hoch.

Auf Basis der ermittelten unterschiedlichen Anzahl der Übersetzungsalternativen der vier Kontextkategorien lässt sich folgende Alternativhypothese zu dieser Nullhypothese formulieren:

- Der kognitive Aufwand unterscheidet sich bei allen vier Kontextkategorien dahingehend, dass er bei der Kategorie „Kein Kontext“ am geringsten ist und über die Kategorien „Fachlicher Kontext“ und „Situativer Kontext“ bis hin zur Kategorie „Funktionaler Kontext“ ansteigt.

Diese Nullhypothese bzw. die Alternativhypothese kann mit den genannten Eye-Tracking-Werten „Dur“, „TrtS“, „FFDur“, „FPDurS“ und „HTra“ untersucht werden, die alle Rückschlüsse auf den kognitiven Aufwand bei der Übersetzung einzelner Strings ermöglichen.

Auf Basis dieser Hypothesen zum kognitiven Aufwand und der davor erläuterten Hypothese zum Referenzmaterial wurde die im Folgenden beschriebene Eye-Tracking-Studie konzipiert. Details zur Konzeption der Studie finden sich in Abschnitt 7.2 und Informationen zum geplanten Ablauf sowie dem verwendeten Zusatzmaterial in Abschnitt 7.3.

7.2 Konzeption der Eye-Tracking- und Key-Logging-Studie

In der Eye-Tracking- und Key-Logging-Studie sollten die bereits im Übersetzungstest getesteten Items (s. Abschnitte 6.2 und 6.4.2) erneut genutzt werden, um die bereits ermittelten Daten bezüglich Recherche und Anzahl der Übersetzungsalternativen als Ausgangspunkt für die weitere Untersuchung nutzen zu können. Für jede Kontextart sowie für die Testgruppe „Kein Kontext“ wurden dazu 18 der Strings aus dem Übersetzungstest ausgewählt. Auf diese Weise wies jede Gruppe eine gleich große Anzahl auf, die sich an den für die Kategorie „Fachlicher Kontext“ vorhandenen 18 Strings orientierte. Die Auswahl der Strings für die Gruppen, bei denen mehr als 18 Strings vorhanden waren, erfolgte dabei primär unter praktischen Gesichtspunkten, da für jeden String noch Zusatzmaterial erstellt werden musste, was für einige Strings problematisch war (s. Abschnitt 7.3).

Die Reihenfolge dieser Strings wurde randomisiert, danach wurden in Translog II vier Teilprojekte mit je 18 Strings erstellt, denen jeweils die Angabe zum Programmtyp vorangestellt wurde, damit die Probandinnen sie wie beim Übersetzungstest auch schon zur Verfügung hatten (s. rechtes Fenster in Abbildung 7.1). Das Translog II-Fenster wies noch ein zweites Teilfenster rechts neben dem mit Programmtyp und AT-Strings auf, das pro Zeile jeweils bereits den Programmtyp aus dem linken Teilfenster enthielt. In dieses Fenster sollten die Probandinnen ihre Übersetzungsvorschläge eingeben. Die vollständige Liste der verwendeten Strings findet sich in Tabellenform in Anhang C.1.

Ein Grund für die Aufteilung auf vier Teilprojekte war, dass so jeweils alle zu übersetzenden Strings gleichzeitig auf dem Bildschirm zu sehen waren und die Probandinnen also nicht nach unten blättern mussten, was die Datenerhebung via Eye-Tracker erschwert hätte. Zudem konnte die Kalibrierung nach jedem Teilprojekt wiederholt werden, wodurch die Qualität der erhobenen Daten optimiert werden konnte, und die Probandinnen erhielten dadurch die Möglichkeit, zwischen den Teilprojekten bei Bedarf Pausen einzulegen.

Um festzustellen, ob die Randomisierung und anschließende Aufteilung

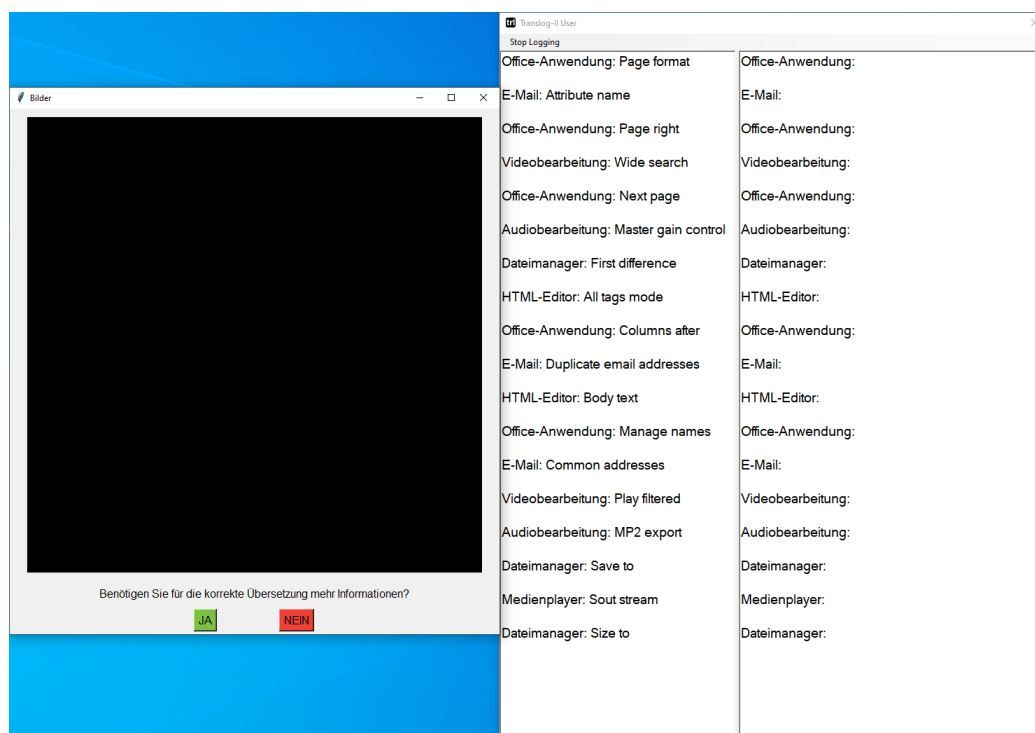


Abbildung 7.1: Projekt 1 in Translog II

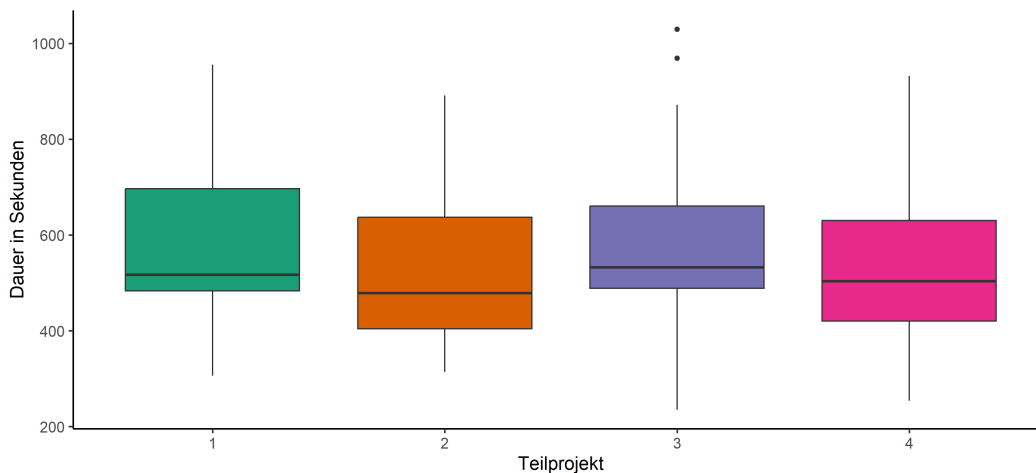


Abbildung 7.2: Bearbeitungsdauer nach Teilprojekten

auf vier Teilprojekte zu einer etwa vergleichbaren Bearbeitbarkeit der Teilprojekte führte, lässt sich die Bearbeitungsdauer der Teilprojekte betrachten (s. Abbildung 7.2), die auf Grundlage der Daten in den Dateien mit Sitzungsinformationen mit der Dateierweiterung „*.ss“ (Spalte „Dur“, vgl. Carl, Schaeffer und Bangalore 2016, S. 20) erstellt wurden. Diese wiederum wurden über die CRITT-TPR-Datenbank generiert (s. Abschnitt 7.6.1 und vgl. Carl, Schaeffer und Bangalore 2016). Mit „Dur“ wird dabei die Produktionsdauer eines finalen Zieltexts pro Sitzung angegeben, also die Zeit vom Beginn bis zum Ende einer Sitzung.⁵

Am Box-Whisker-Plot in Abbildung 7.2 ist zu sehen, dass die Bearbeitungsdauer der vier Teilprojekte für alle Probandinnen ähnlich lang war (Spalte „Dur“, vgl. Carl, Schaeffer und Bangalore 2016, S. 20). Der Median liegt für alle Teilprojekte insgesamt bei 512,2 Sekunden, der Interquartilsabstand beträgt 215,0 Sekunden, die Standardabweichung liegt bei 172,2 Sekunden. Für die Teilprojekte schwankt der Median zwischen 478,9 (Teilprojekt 2) und 532,0 Sekunden (Teilprojekt 3) und die Standardabweichung zwischen 150,9 (Teilprojekt 2) und 204,2 Sekunden (Teilprojekt 3).

An diesen Werten lässt sich insbesondere in Kombination mit der Betrachtung der Bearbeitungsdauer nach Probandinnen (s. Abbildung 7.7 sowie die weiteren Informationen dazu in Abschnitt 7.4) erkennen, dass die Probandinnen zwar sehr unterschiedlich lange Zeit für die Bearbeitung brauchten, dass sich das aber nur sehr eingeschränkt auf die Teilprojekte auswirkt: Ge-

⁵Vgl. dazu die Erläuterung dieses Messwerts in der Tabelle der Messwerte unter der Adresse <https://sites.google.com/site/centretranslationinnovation/tpr-db/features>; letzter Zugriff: 27.03.2025.

genüber dem Median von etwa 8,5 Minuten für alle Teilprojekte weicht die Bearbeitungszeit der Teilprojekte nur einen Bruchteil ab, nämlich etwa 30 Sekunden darunter bzw. 20 darüber. Daraus lässt sich schließen, dass die übersetzerischen Probleme weitgehend gleichmäßig auf die Teilprojekte verteilt waren.

7.3 Geplanter Ablauf und Zusatzmaterial

In Anhang C.2 findet sich die Anweisungsdatei, die den Probandinnen zu Beginn der Studie vorgelegt wurde. Darin ist unter „Ablauf der Studie“ der Ablauf zu finden, den die Probandinnen einhalten sollten. Sie sollten jeweils String für String von oben nach unten durcharbeiten, dabei zunächst ohne Recherchemöglichkeiten einen Übersetzungsvorschlag eingeben und danach im linken Fenster, das zu diesem Zeitpunkt keine Informationen enthielt (s. linkes Fenster in Abbildung 7.1), die Frage „Benötigen Sie für die korrekte Übersetzung mehr Informationen?“ mit „Ja“ oder „Nein“ beantworten. Für diese Schaltflächen wurden mit dem Programm AutoHotkey Version 2.0.10⁶ zwei Shortcuts, „Alt+J“ und „Alt-N“, definiert, damit die Probandinnen ohne Ablenkung durch Nutzung der Maus die Studie durcharbeiten konnten. Wenn sie diese Angabe gemacht hatten, wurde unabhängig von ihrer Antwort Zusatzmaterial in Form eines Screenshots, in dem der String zu sehen und mit einem roten Pfeil markiert war (beispielhaft für den String „First difference“ in Abbildung 7.3 zu sehen), oder in Form einer textlichen Beschreibung des Strings in einem Satz angezeigt (beispielhaft für den String „First difference“ in Abbildung 7.4 zu sehen). Der Screenshot stand dabei für die Informationen, die in einem Softwarelokalisierungstool oder in der lauffähigen Version der zu übersetzenden Software als situativer Kontext zur Verfügung stehen, und die textlichen Beschreibungen dafür, wie GUI-Strings im Rahmen der Dokumentation eines Softwareprodukts erläutert werden. Das Zusatzmaterial wurde unabhängig von der ersten Antwort der Probandinnen angezeigt, um eine gleichbleibende und damit auch ausreichend hohe Anzahl von Datenpunkten für alle Kontextkategorien und alle Varianten des Zusatzmaterials, also Screenshot und textliche Beschreibung, zu erhalten.

Wenn sie durch dieses Zusatzmaterial dazu veranlasst wurden, durften die Probandinnen nun ihren ersten Übersetzungsvorschlag überarbeiten. Danach sollten sie wieder mit den Shortcuts „Alt+J“ und „Alt-N“ die jetzt unter dem Material im linken Fenster angezeigte Frage, „Waren diese Informationen für die korrekte Übersetzung hilfreich?“, beantworten. Danach sollten sie mit dem nächsten String fortfahren.

⁶<https://www.autohotkey.com/>; letzter Zugriff: 22.08.2025

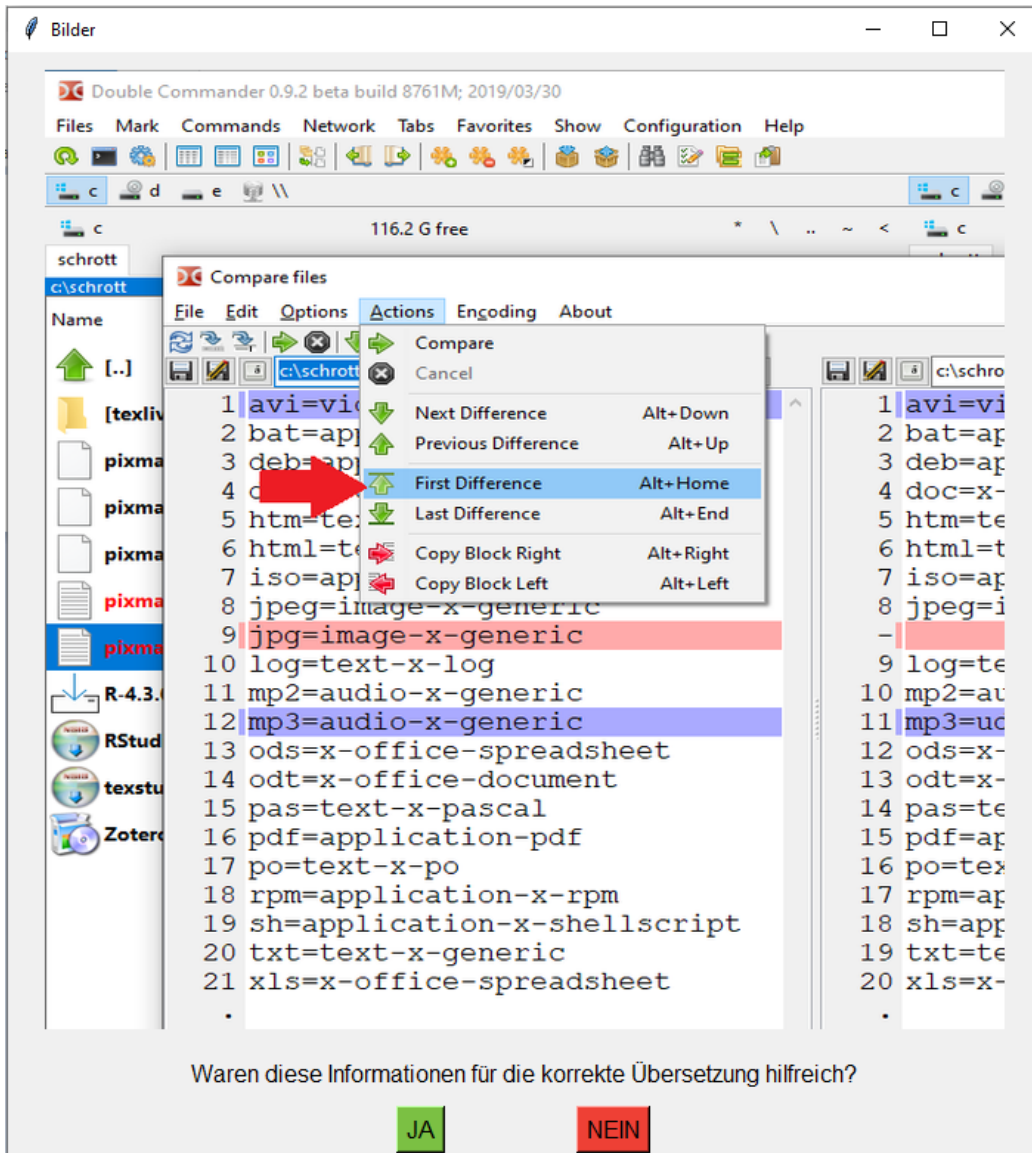


Abbildung 7.3: Screenshot des Strings „First difference“, Markierung mit rotem Pfeil

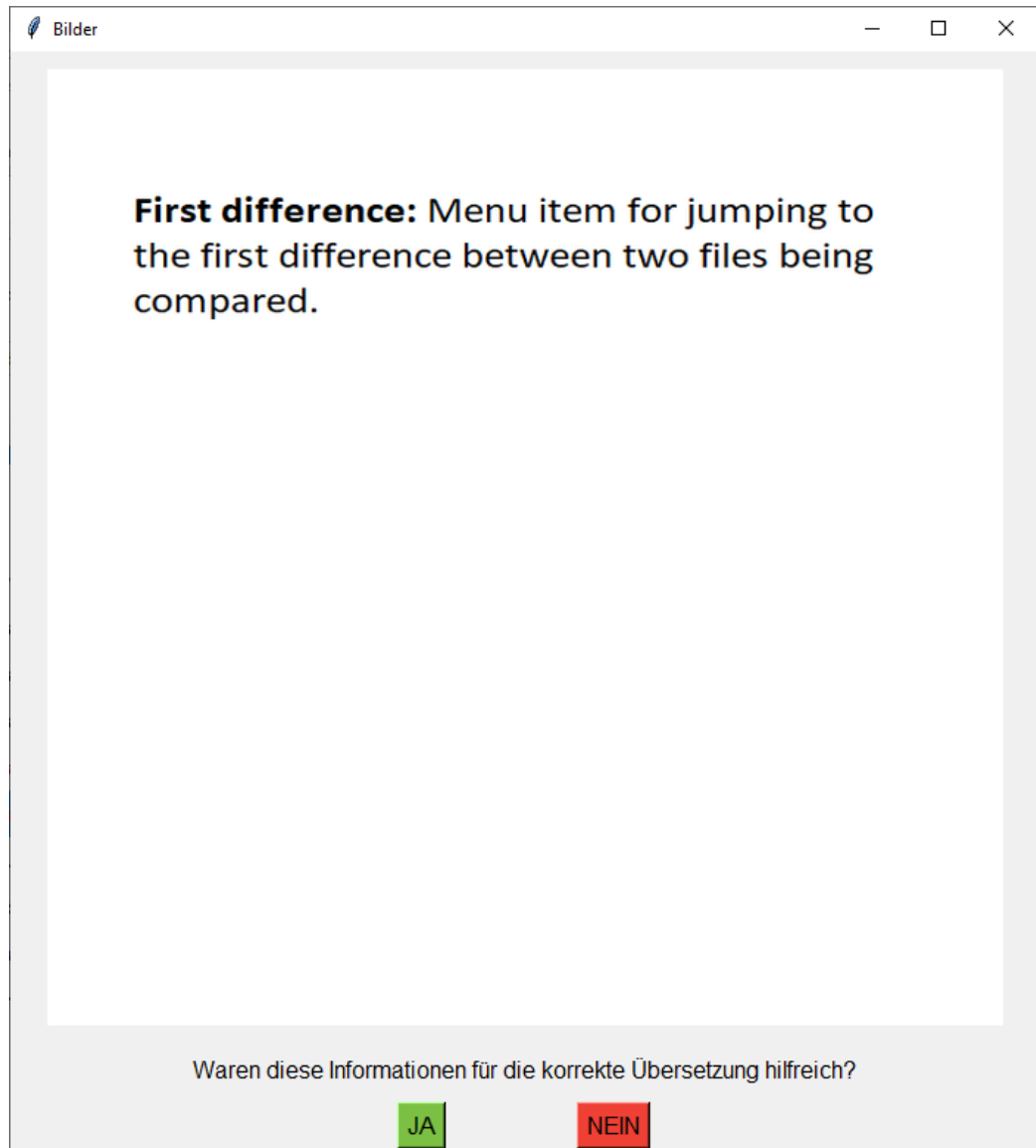


Abbildung 7.4: Textliche Beschreibung des Strings „First difference“

Die vier Teilprojekte in Translog II wurden von den Probandinnen in vier verschiedenen Reihenfolgen durchgearbeitet (Projekt 1, 2, 3, 4; Projekt 2, 3, 4, 1; Projekt 3, 4, 1, 2; Projekt 4, 1, 2, 3), wobei es jeweils zwei Teilgruppen (A1 und A2, B1 und B2, C1 und C2, D1 und D2) gab, von denen der einen Teilgruppe der Screenshot und der anderen die textliche Beschreibung eines Strings angezeigt wurde.

Die Anzeige des Zusatzmaterials erfolgte durch ein Python-Skript, das von Dr. Katharina Oster vom Fachbereich 06 der Johannes Gutenberg-Universität Mainz erstellt wurde. Im Skript wurden je nach Reihenfolge der Teilprojekte und Teilgruppe eine zugehörige Stimulusliste genutzt, um die vorgesehenen Materialien anzuzeigen. Zudem wurde der Stimulus mit den beiden von den Probandinnen auf die Frage gegebenen Antworten zusammen mit den Systemzeiten der Antworten in einer CSV-Datei gespeichert, die für die Auswertung genutzt werden konnte.

Die benötigten Screenshots konnten in vielen Fällen erstellt werden, indem die gewünschten Strings in der GUI des jeweiligen Programms gesucht wurden. Wo dies nicht möglich war, weil die Strings im Programm nicht gefunden werden konnten, wurden eine geeignete Stelle im Programm gesucht, an der der gewünschte String in einen Screenshot eingefügt werden konnte. Die textlichen Beschreibungen wurden in einigen Fällen auf Basis vorhandener Dokumentation der Softwareprodukte angepasst und vereinheitlicht, in anderen Fällen auf Basis der getesteten Funktion des Strings selbst erstellt.

7.4 Probandinnen

Für die Eye-Tracking-Studie wurden über verschiedene Kanäle professionelle Übersetzerinnen als Probandinnen gesucht, die deutsche Muttersprachlerinnen sind und über Erfahrung in der Softwarelokalisierung mit der Übersetzungsrichtung Englisch-Deutsch verfügen. Letztlich nahmen 26 Probandinnen an der Studie teil, die teils Freiberuflerinnen sind und zu einem anderen Teil festangestellt bei Softwareunternehmen oder Übersetzungsdienstleistern arbeiten.

Alle Probandinnen wurden gebeten, vor der eigentlichen Studie den Translation and Interpreting Competence Questionnaire (TICQ) auszufüllen, um eine Einschätzung der Übersetzungskompetenz der Probandinnen sowie die Vergleichbarkeit der Ergebnisse der vorliegenden Studie zu erreichen (vgl. Schaeffer, Huepe u. a. 2020). Mithilfe des TICQ sollen quantitative und qualitative Daten für mehrere Dimensionen der Übersetzungs- und Dolmetschkompetenz erfasst werden (vgl. Schaeffer, Huepe u. a. 2020, S. 92).

Für die Befragung im Rahmen der Eye-Tracking- und Keylogging-Studie

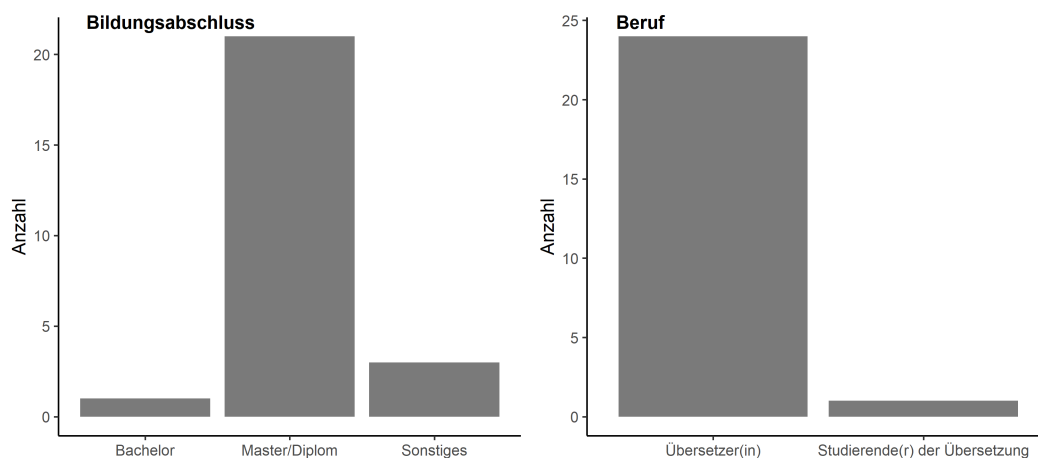


Abbildung 7.5: Was ist Ihr höchster Bildungsabschluss? Was sind Sie im Moment hauptberuflich?

in diesem Kapitel wurden die Abschnitte A für Spracherwerb und mehrsprachige Kompetenz und B für Übersetzungskompetenz verwendet (vgl. Schaeffer, Huepe u. a. 2020, S. 92). In Abschnitt B1 wurde dazu in Modul B2 („Translationskompetenz“) gegenüber der Standardversion folgende weitere Frage hinzugefügt: „Wie gut **lokalisieren** Sie von Ihrer ersten Fremdsprache in Ihre Muttersprache? Von 1 (sehr schlecht) bis 100 (sehr gut)“ (Hervorhebung im Original).

Der Fragebogen wurde über die Umfrageplattform (LimeSurvey) der Johannes Gutenberg-Universität Mainz bereitgestellt und von 25 Probandinnen vollständig ausgefüllt. Die 25 Probandinnen wiesen ein Durchschnittsalter von 44,4 Jahren auf (Standardabweichung: 9,8 Jahre). 20 gaben als Geschlecht „Weiblich“ an, drei „Männlich“ und zwei machten keine Angabe.

Dem linken Diagramm in Abbildung 7.5 können die Antworten auf die Frage „Was ist Ihr höchster Bildungsabschluss?“ entnommen werden: 21 Teilnehmende und damit 84 Prozent gaben „Master/Diplom“ als ihren höchsten Bildungsabschluss an, eine Person (4 Prozent) „Bachelor“ und drei (12 Prozent) „Sonstiges“. Auch die drei Probandinnen dieser letzten Gruppe gaben allerdings Ausbildungswege bzw. staatliche Prüfungen aus dem Bereich Übersetzen an. Das rechte Diagramm zeigt die Antworten auf die Frage „Was sind Sie im Moment hauptberuflich?“ 24 Teilnehmende und damit 96 Prozent gaben als Antwort „Übersetzer(in)“ an, eine Person (4 Prozent) „Studierende(r) der Übersetzung“.

Die Box-Whisker-Plots in Abbildung 7.6 zeigen die Selbsteinschätzung der Probandinnen hinsichtlich ihrer Übersetzungs- und Lokalisierungskom-

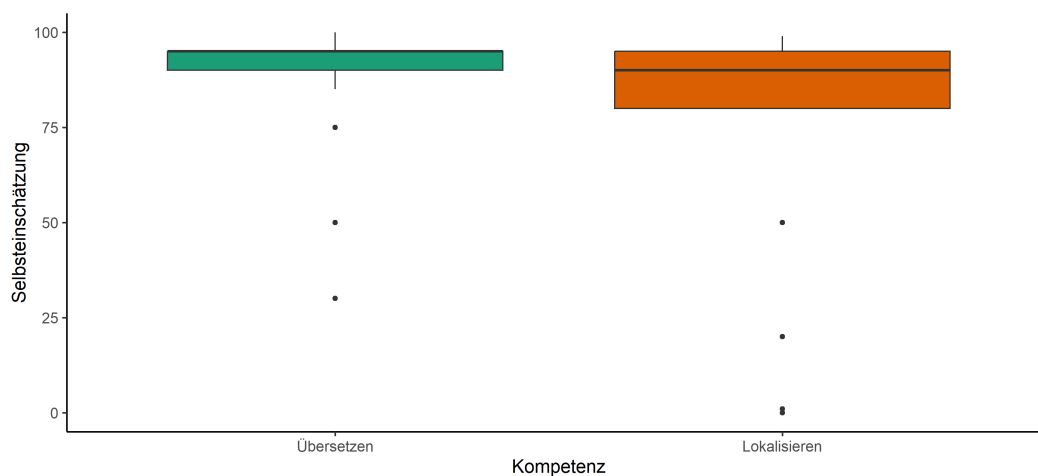


Abbildung 7.6: Wie gut übersetzen/lokalisieren Sie von Ihrer ersten Fremdsprache in Ihre Muttersprache? Von 1 (sehr schlecht) bis 100 (sehr gut)

petenz basierend auf folgenden Fragen: „Wie gut übersetzen Sie von Ihrer ersten Fremdsprache in Ihre Muttersprache? Von 1 (sehr schlecht) bis 100 (sehr gut)“ und „Wie gut lokalisieren Sie von Ihrer ersten Fremdsprache in Ihre Muttersprache? Von 1 (sehr schlecht) bis 100 (sehr gut)“.

Bei der Frage zur Übersetzungskompetenz (linker Box-Whisker-Plot) liegt der Median bei 95, der Interquartilsabstand beträgt 5 und die Standardabweichung liegt bei 16,0. Bei der Frage zur Lokalisierungskompetenz (rechter Box-Whisker-Plot) liegt der Median bei 90, der Interquartilsabstand beträgt 15 und die Standardabweichung liegt bei 29,1.

Die Outlier bei der Selbsteinschätzung zur Übersetzungskompetenz (die Werte 30 und 50) und zur Lokalisierungskompetenz (die Werte 0, 1, 20 und 50) lassen sich weitestgehend dadurch erklären, dass für die Studie die berufliche Spezialisierung im Bereich der Lokalisierung Englisch-Deutsch die Voraussetzung war, im Fragebogen an dieser Stelle aber explizit nach der Übersetzungs- bzw. Lokalisierungskompetenz aus der ersten Fremdsprache in die Muttersprache gefragt wurde. In allen bis auf einem dieser Fälle haben die Probandinnen als erste Fremdsprache Latein bzw. Französisch angegeben, sie haben also gar nicht die Frage nach der Übersetzungs- bzw. Lokalisierungskompetenz Englisch-Deutsch beantwortet, sondern beispielsweise die Frage nach der Übersetzungs- bzw. Lokalisierungskompetenz Latein-Deutsch, die erwartbar deutlich geringer ist als die im Haupttätigkeitsbereich der Lokalisierung Englisch-Deutsch. Nur die Angabe des Werts 0 in einem Fall lässt sich dadurch nicht erklären und könnte auf einem Fehler beruhen.

Auf Basis der TICQ-Werte können einerseits individuelle Fähigkeiten im

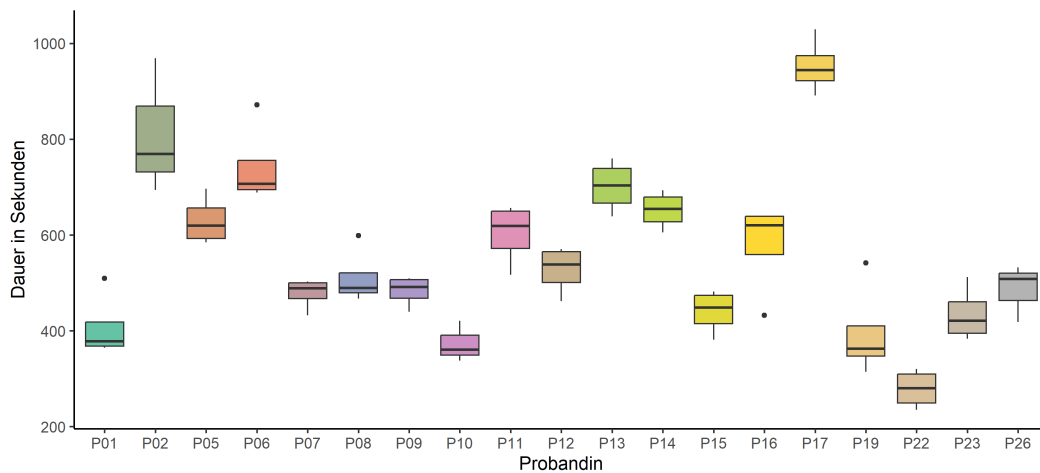


Abbildung 7.7: Bearbeitungsdauer nach Probandinnen

Rahmen des Übersetzungsprozesses (TICQ-F1z) und andererseits individuelle Fähigkeiten bei der Qualitätssicherung (TICQ-F2z) ermittelt werden (vgl. Jacob u. a. 2024, S. 548–549). Für TICQ-F1z ergeben die Werte der Probandinnen einen Median von -0,02 mit einer Standardabweichung von 1,0 und einem Interquartilsabstand von 1,1, für TICQ-F2z einen Median von -0,12 mit einer Standardabweichung von 1,0 und einem Interquartilsabstand von 0,84. Beide Messwerte zeigen sich damit für die Gruppe der Probandinnen sehr homogen, im Hinblick auf ihre Fähigkeiten bei Übersetzungsprozess und Qualitätssicherung sind die Probandinnen also gut vergleichbar. Dadurch erscheint die Gruppe insgesamt als sehr geeignet für die Eye-Tracking- und Key-Logging-Studie, für die eine homogene Gruppe explizit erwünscht war.

Die Probandinnen lassen sich noch in einem weiteren Punkt vergleichen, nämlich der Bearbeitungsdauer für einzelne Probandinnen (s. Abbildung 7.7), die wie auch die Bearbeitungsdauer nach Teilprojekten (s. Abschnitt 7.2) auf Grundlage der Daten in den Dateien mit Sitzungsinformationen mit der Dateierweiterung „*.ss“ (Spalte „Dur“, vgl. Carl, Schaeffer und Bangalore 2016, S. 20) erstellt wurden. Diese wurden wiederum über die CRITT-TPR-Datenbank generiert (s. Abschnitt 7.6.1 und vgl. Carl, Schaeffer und Bangalore 2016). Mit „Dur“ wird auch hier die Produktionsdauer eines finalen Zieltexts pro Sitzung angegeben, also die Zeit vom Beginn bis zum Ende einer Sitzung.

Bei diesem Vergleich ist zu erkennen, dass P22 ihre Teilprojekte in der kürzesten Zeit bearbeitet hat (Median: 280,0 Sekunden, Standardabweichung: 40,5 Sekunden), während P17 substantiell länger gebraucht hat (Median: 943,9 Sekunden, Standardabweichung: 58,0 Sekunden). Bei den übrigen

Probandinnen lag die Bearbeitungszeit zwischen diesen Extremwerten.

Wie bereits in Abschnitt 7.2 dargestellt, lässt sich an diesen Daten in der Zusammenschau mit der Bearbeitungsdauer nach Teilprojekten erkennen, dass die übersetzerischen Probleme weitgehend gleichmäßig auf die Teilprojekte verteilt waren, auch wenn die Bearbeitungsdauer der Teilprojekte durch die einzelnen Probandinnen teils sehr unterschiedlich war. Die Homogenität der Gruppe der Probandinnen in Bezug auf ihre Fähigkeiten bei Übersetzungsprozess und Qualitätssicherung führte also nicht dazu, dass alle ähnlich lange für die Bearbeitung der Strings in der Studie benötigten.

7.5 Durchführung der Studie

Ein Teil der Erhebungen wurde vor Ort im Eye-Tracking-Labor des Fachbereichs 06 der Johannes Gutenberg-Universität Mainz durchgeführt. Der übrige Teil wurde im Eye-Tracking-Labor der Technischen Hochschule Köln und in Räumlichkeiten verschiedener Unternehmen durchgeführt. Dabei wurde stets darauf geachtet, dass die Arbeitsbedingungen und Lichtverhältnisse so waren, dass möglichst optimale Eye-Tracking-Bedingungen vorlagen, und die Räume während der Datenerhebung für Dritte nicht zugänglich waren. In allen Fällen war die Studienleitung im selben Raum mit anwesend, um das Experiment zu überwachen.

Zur Datenerhebung wurde ein Laptop mit SMI RED 250 Mobile als Eye-Tracker genutzt, der an einem an den Laptop angeschlossenen externen Monitor befestigt war. Die Probandinnen arbeiteten mit einer ebenfalls an den Laptop angeschlossenen externen Tastatur ohne Maus. Die Eye-Tracking- und Key-Logging-Daten wurden mit Translog II Version 2.0.1.222 erhoben.

Vor Beginn des eigentlichen Tests wurden die Probandinnen über den Daten- und Persönlichkeitsschutz bei der durchzuführenden Studie aufgeklärt und unterzeichneten eine Einwilligungserklärung. Diejenigen Probandinnen, die nicht im Rahmen ihrer Arbeitszeit an der Studie teilnehmen konnten oder freiberuflich tätig waren, erhielten eine Aufwandsentschädigung von 50 Euro für die Teilnahme. Danach wurde mit den Probandinnen der Ablauf besprochen (s. Anweisungsdatei), den sie danach auch mithilfe von fünf Practice-Items testen konnten (s. erste Tabelle in Anhang C.1). Erst im Anschluss daran begann die eigentliche Studie und Datenerhebung. Ein paar der Practice-Items stammten dabei nicht aus dem Korpus, sondern waren andere Strings aus zwei oder drei Wörtern der verwendeten Programme, bei denen sich problemlos Screenshots und textliche Beschreibungen erstellen ließen. Vor Beginn der Übersetzung der Practice-Items und jeweils vor Beginn der Übersetzung der vier Teilprojekte wurde der Eye-Tracker kalibriert.

7.6 Auswertung der Studie

In diesem Kapitel werden die Ergebnisse der Eye-Tracking-Studie vorgestellt. Dazu wird zunächst die Vorbereitung der erhobenen Daten geschildert (s. Abschnitt 7.6.1). Im Anschluss daran folgen die Ergebnisse in Bezug auf die Nutzung der in der Studie angebotenen Zusatzmaterialien (s. Abschnitt 7.6.2) und zuletzt die Ergebnisse des Eye-Trackings und Key-Loggings (s. Abschnitt 7.6.3). Die Interpretation der Ergebnisse folgt dann in Abschnitt 7.7.

7.6.1 Vorbereitung der erhobenen Daten

Vor Beginn der eigentlichen Auswertung wurden die Eye-Tracking-Daten überprüft, indem in Translog II via Replays größere Anteile der Teilprojekte gesichtet wurden. Im Ergebnis wurden nur die Teilprojekte von Probandinnen in die Eye-Tracking-Auswertung (Abschnitt 7.6.3) aufgenommen, bei denen mindestens drei der vier Teilprojekte eine gute Zuordnung der Blickdaten zu den bearbeiteten Wörtern aufwiesen. Die Daten der anderen Probandinnen (P03, P04, P18, P20, P21, P24 und P25) wurden aus der Eye-Tracking-Auswertung herausgenommen, teils weil sie für manche Teilprojekte komplett unbrauchbar waren und teils weil es bei der Erfassung der Blickdaten bzw. bei der Kalibrierung Probleme gegeben hatte, die sich beispielsweise darin zeigten, dass die Blickdaten systematisch falschen Zeilen zugeordnet waren. Eine Nachbearbeitung der Daten in Translog II erfolgte nicht. Da die Daten für die Zusatzmaterialien unabhängig vom Eye-Tracking bzw. Key-Logging erhoben wurden, konnten sie dagegen für alle Probandinnen ausgewertet werden (s. Abschnitt 7.6.2).

Die 19 Datensätze aus dem Eye-Tracking und Key-Logging mit mindestens drei tauglichen Teilprojekten (16 vollständige Datensätze und drei mit nur drei Teilprojekten) wurden in die CRITT-Datenbank zur Translationsprozessforschung (vgl. Carl, Bangalore und Schaeffer 2016) hochgeladen und dabei via SimAlign (Parameter „SI“) automatisch aligniert. Weil in den Teilprojekten nur Zwei- und Dreiwortstrings pro Zeile enthalten waren, funktionierte die automatische Segmentierung sehr gut und nur in einzelnen Teilprojekten, bei denen von Probandinnen ein Punkt für Abkürzungen genutzt wurde, musste die Segmentierung korrigiert werden.

Die automatische Alignierung via SimAlign wurde manuell überprüft und bei Bedarf korrigiert. Dabei wurden für alle Daten Komposita in der Übersetzung, die von den Probandinnen zusammengeschrieben oder mit Bindestrichen verbunden wurden, jeweils als ein einziges Wort annotiert. Dadurch wurde sichergestellt, dass Komposita auch bei unterschiedlichen Schreibweisen

gleich behandelt wurden. Eine Ausnahme bildeten Komposita, die unverändert bzw. nur mit Änderung der Groß-/Kleinschreibung aus dem Englischen übernommen wurden, wie beispielsweise „Web developer“ mit der Übersetzung „Web Developer“ im Unterschied zu „Webentwickler“, „Web-Entwickler“ oder „Web-Developer“ in Teilprojekt 4. Bei solchen Übersetzungen wurden die einzelnen Wörter in der Übersetzung mit den zugehörigen Wörtern des AT aligniert. Bei den Angaben der Programmtypen zu Beginn der einzelnen Zeilen wurde dagegen die automatische Alignierung durch SimAlign übernommen: Die einzelnen Bestandteile von mit Bindestrichen geschriebenen Komposita wurden miteinander aligniert und die Bindestriche ebenso.

Wenn in der Übersetzung eine Präposition enthalten war, für die es im AT keine Entsprechung gab (Beispiel: „(to) access“ in Teilprojekt 3: „zugreifen auf“), wurden Präposition und Verb mit dem AT-Verb aligniert. Bei einzelnen freieren Übersetzungen, bei denen die eindeutige Zuordnung der AT-Wörter zu ZT-Wörtern nicht möglich war, wurde der gesamte String als ein einziges Item annotiert, auch wenn mehrere Wörter vorlagen. Beispiel: „On losing focus“ in Teilprojekt 3 und die Übersetzung „Nach Verwendung anzeigen“ (P06_T3⁷) oder „Proper case“ in Teilprojekt 3 und die Übersetzung „Erster Buchstabe groß“ (P16_T3). Ebenso wurde vorgegangen, wenn (scheinbar) Wörter im ZT fehlten, weil eine verkürzende Übersetzung gewählt wurde. Beispiel: „All tags mode“ aus Teilprojekt 1 und die Übersetzung „Alle Tags“ (P16_T1). Artikel wurden jeweils mit dem zugehörigen Substantiv aligniert. Beispiel: „Video Size“ aus Teilprojekt 3 und die Übersetzung „Größe des Videos“ (P06_T3), bei der „video“ mit „des Videos“ aligniert wurde.

In Teilprojekt P12_T2 erwies sich die Alignierung der letzten Strings ab „Fill single edit“ als nicht brauchbar. Als die Probandin für einen String eine sehr lange Übersetzung eingab, wurde am Zeilenende automatisch ein Zeilenumbruch eingefügt. Dadurch standen die ZT-Strings nicht mehr in denselben Zeilen wie die AT-Strings, weshalb die letzten vier Strings nicht mehr wie vorgesehen übersetzt und somit auch nicht aligniert werden konnten.

Nach der manuellen Überarbeitung der Alignierung wurden die Auswertungstabellen der CRITT-Datenbank erstellt und heruntergeladen und anschließend für die detaillierte Auswertung (s. Abschnitt 7.6.3, vgl. auch Carl, Bangalore und Schaeffer 2016) genutzt.

Bei den TXT-Dateien, in denen die Ergebnisse des Skripts gespeichert wurden (s. Abschnitt 7.6.2), wurden vor Beginn der Auswertung jeweils die ersten fünf Zeilen mit Informationen zu den Practice-Items manuell entfernt. Danach wurden die Inhalte aller TXT-Dateien in die Reihenfolge Teilprojekt

⁷In dieser Schreibweise findet sich zunächst die Probandinnennummer, hier P06, und nach dem Unterstrich das Teilprojekt, hier Teilprojekt 3.

1 bis 4 gebracht, wenn die Probandinnen die Teilprojekte in einer anderen Abfolge durchgearbeitet hatten (s. Abschnitt 7.3). Anschließend wurde eine Spalte mit Angaben zu den Kontextarten der einzelnen Strings eingefügt, um so die Auswertung anhand der verschiedenen Kontextarten (s. Abschnitt 7.6.2) zu ermöglichen. Diese Daten konnten dann mit den für die Auswertung der Eye-Tracking- und Key-Logging-Daten in der CRITT-Datenbank erstellten Dateien mit Segmentinformationen (Dateierweiterung „*.sg“) kombiniert werden, wodurch auch die Eye-Tracking- und Key-Logging-Daten anhand der Kontextarten ausgewertet werden konnten (s. Abschnitt 7.6.3). Für diesen letzten Schritt wurden allerdings die Teilprojekte entfernt, deren Eye-Tracking-Daten untauglich waren (s.o.).

7.6.2 Nutzen der Zusatzmaterialien

Im ersten Schritt der eigentlichen Auswertung der Studie werden die Daten der Zusatzmaterialien betrachtet (s. Abschnitt 7.3), also welche Antworten auf die beiden Fragen „Benötigen Sie für die korrekte Übersetzung mehr Informationen?“ und „Waren diese Informationen für die korrekte Übersetzung hilfreich?“ gegeben wurden. Dafür wurden die TXT-Dateien ausgewertet, in denen die Ergebnisse des Skripts gespeichert wurden.

Bei den folgenden Auswertungen wurden die Daten sämtlicher 26 Probandinnen und aller Teilprojekte genutzt, da sich die Probleme mit dem Erfassen der Blickdaten nicht auf diese Daten auswirkten und auf diese Weise ein größerer Datensatz genutzt werden konnte. Erst für die Auswertungen der Eye-Tracking- und Key-Logging-Daten, die im letzten Teil dieses Abschnitts beginnt, wurden dann die 19 Datensätze mit mindestens drei tauglichen Teilprojekten genutzt (s. Abschnitt 7.6.1).

Zunächst wurden die Antworten auf die Frage „Benötigen Sie für die korrekte Übersetzung mehr Informationen?“ ausgewertet. Die Ergebnisse sind in Abbildung 7.8 zu sehen. Diese Frage wurde von den Probandinnen im Anschluss an eine Erstübersetzung der einzelnen Strings, die ohne Zusatzmaterialien erfolgen musste, beantwortet (Informationen zur Interpretation der hier vorgestellten Ergebnisse s. Abschnitt 7.7). Bei 66,3 Prozent aller Strings wurde von den Probandinnen als Antwort auf diese Frage „Ja“ und bei 33,7 Prozent „Nein“ angegeben. Für die verschiedenen Kontextarten ergeben sich allerdings sehr unterschiedliche Anteile: Bei der Kategorie „Kein Kontext“ wurde diese Frage nur in 35,5 Prozent der Fälle mit „Ja“ beantwortet. Bei den anderen Kategorien lautete die Antwort dagegen jeweils in mehr als der Hälfte der Fälle „Ja“: Bei der Kategorie „Fachlicher Kontext“ in 60,5 Prozent, bei der Kategorie „Situativer Kontext“ in 78,8 Prozent und bei der Kategorie „Funktionaler Kontext“ in 90,6 Prozent der Fälle.

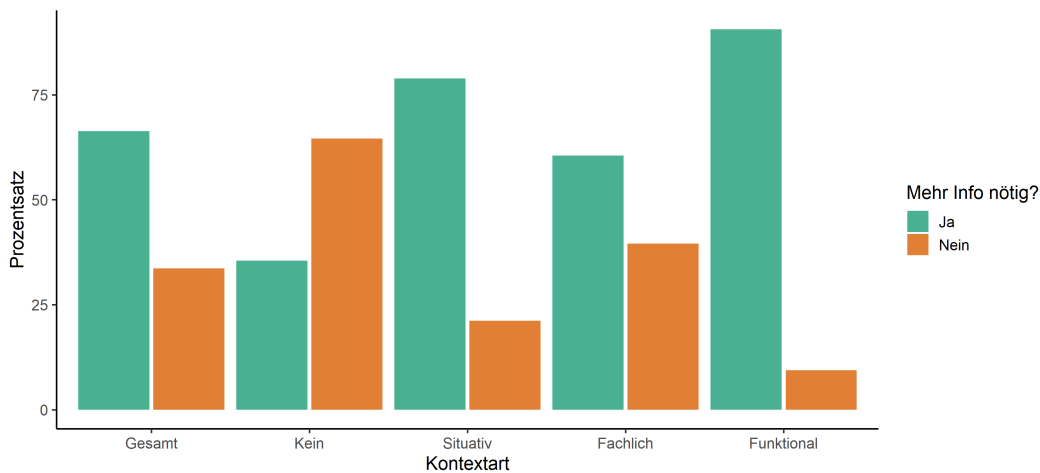


Abbildung 7.8: Benötigen Sie für die korrekte Übersetzung mehr Informationen?

Nach Beantwortung der ersten Frage wurde das Zusatzmaterial, entweder ein Screenshot mit markiertem zu übersetzendem String oder eine textliche Beschreibung des Strings, angezeigt. Die Probandinnen sollten sich dieses Material ansehen, bei Bedarf Änderungen an ihrer Übersetzung vornehmen und danach die zweite Frage, „Waren diese Informationen für die korrekte Übersetzung hilfreich?“, beantworten. Abbildung 7.9 zeigt in drei Diagrammen die Auswertung dieser Frage. Im ersten Diagramm, „Gesamt“, ist eine Übersicht unabhängig vom jeweiligen Material zu sehen: Insgesamt wurde diese Frage bei 56,7 Prozent der Strings mit „Ja“ beantwortet und in 43,3 Prozent mit „Nein“. Bei den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ wurde die Frage mit 41,2 Prozent bzw. 49,6 Prozent in weniger als der Hälfte der Fälle mit „Ja“ beantwortet, bei den Kategorien „Situativer Kontext“ und „Funktionaler Kontext“ mit 73,9 Prozent bzw. 62,2 Prozent in mehr als der Hälfte.

Das zweite Diagramm, „Beschreibungen“, enthält die Verteilung für die Fälle, in denen als Material eine textliche Beschreibung gezeigt wurde. Eine solche Beschreibung wurde insgesamt bei 58,7 Prozent der Strings als hilfreich eingeschätzt und bei 41,3 Prozent der Strings als nicht hilfreich. Auch hier wurde die Frage bei den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ mit 41,0 Prozent bzw. 48,7 Prozent in weniger als der Hälfte der Fälle mit „Ja“ beantwortet, bei den Kategorien „Situativer Kontext“ und „Funktionaler Kontext“ mit 74,8 Prozent bzw. 70,1 Prozent in mehr als der Hälfte.

Im dritten Diagramm, „Screenshots“, sind die Anteile für die Strings zu sehen, in denen ein Screenshot als Zusatzmaterial angeboten wurde. Mit 54,8

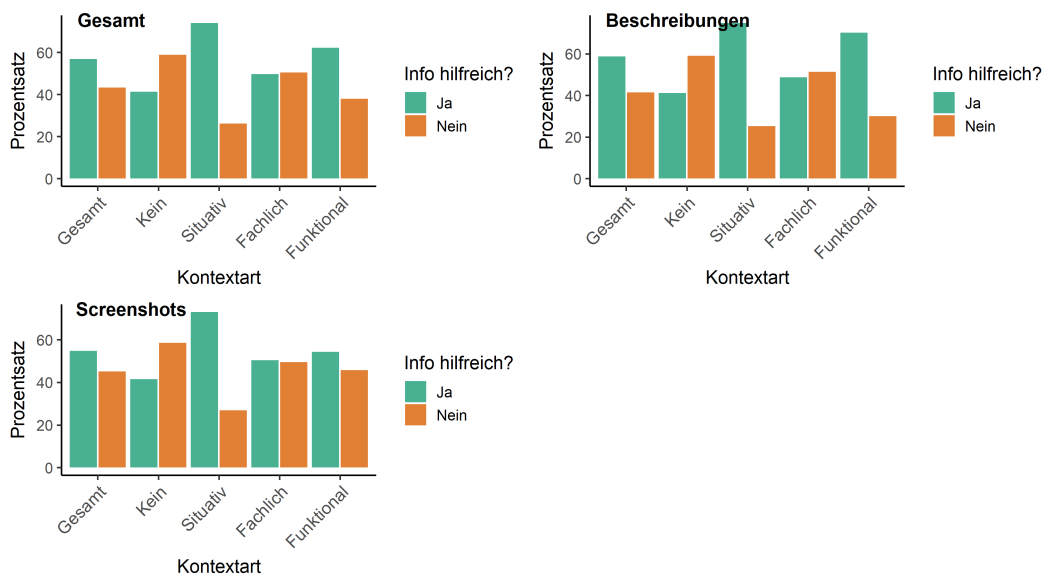


Abbildung 7.9: Waren diese Informationen für die korrekte Übersetzung hilfreich?

Prozent wurden auch diese insgesamt mehrheitlich als hilfreich eingeschätzt (in 45,2 Prozent der Fälle wurde diese Frage verneint). Nur die Kategorie „Kein Kontext“ lag in diesem Fall mit 41,5 Prozent Zustimmung unter der Hälfte. Bei den Kategorien „Fachlicher Kontext“ mit 50,4 Prozent, „Funktionaler Kontext“ mit 54,3 Prozent und „Situativer Kontext“ mit 73,1 Prozent wurde diese Frage bejaht.

Abbildung 7.10 kann entnommen werden, ob die Probandinnen die erste und die zweite Frage gleich (beide mit „Ja“ oder beide mit „Nein“) beantwortet (in den Diagrammen mit der Antwort „Ja“, nämlich beide Antworten sind identisch, angegeben) oder die Antwort von Ja“ in „Nein“ bzw. von „Nein“ in Ja“ geändert haben (in den Diagrammen mit der Antwort „Nein“, nämlich beide Antworten sind nicht identisch, angegeben). Auch hier gibt es zunächst eine Übersicht (Diagramm „Gesamt“), aus der sich entnehmen lässt, dass sowohl insgesamt bei 72,6 Prozent der Strings als auch für die einzelnen Kontextarten bei 66,0 Prozent („Fachlicher Kontext“), 67,3 Prozent („Funktionaler Kontext“), 76,7 Prozent („Kein Kontext“) und 80,6 Prozent („Situativer Kontext“) der Strings in mehr als der Hälfte der Fälle identische Antworten angegeben wurden.

Betrachtet man lediglich Beschreibungen als angebotenes Material (Diagramm „Beschreibungen“), zeigen sich ähnliche Ergebnisse: Insgesamt wurden bei 74,6 Prozent der Strings und auch für die einzelnen Kontextarten

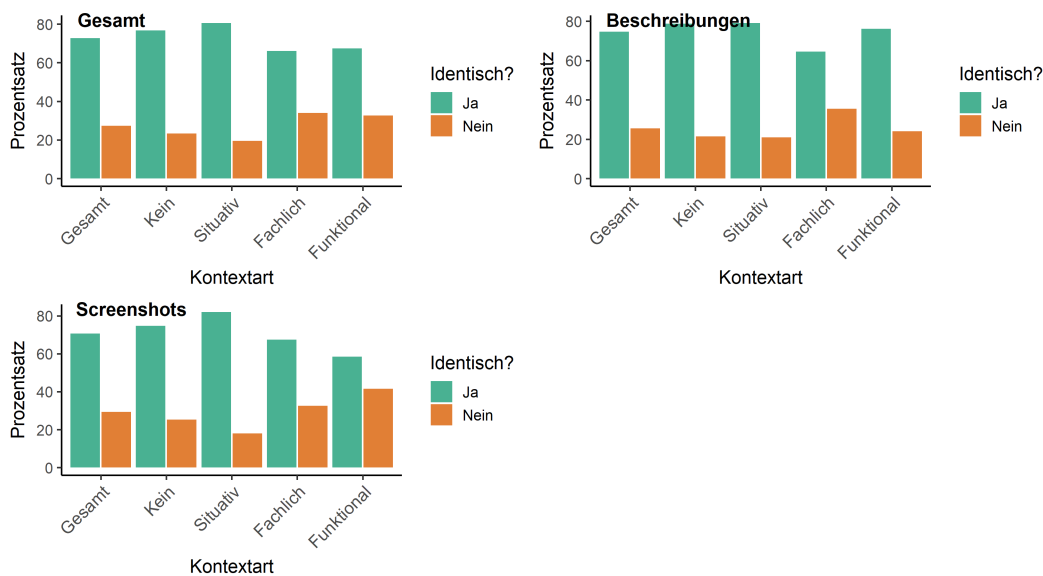


Abbildung 7.10: Sind die Antworten auf beide Fragen identisch?

bei 64,5 Prozent („Fachlicher Kontext“), 76,1 Prozent („Funktionaler Kontext“), 78,6 Prozent („Kein Kontext“) bzw. 79,1 Prozent („Situativer Kontext“) der Strings identische Antworten gegeben. Ebenso bei den Screenshots (Diagramm „Screenshots“): Hier blieb in 70,7 Prozent der Fälle gesamt und auch bei den einzelnen Kontextarten die Antwort unverändert. Allerdings gibt es ein paar Abweichungen bei den einzelnen Kontextarten: Mit 58,5 Prozent weist die Kategorie „Funktionaler Kontext“ den geringsten Wert für unveränderte Antworten auf, gefolgt von der Kategorie „Fachlicher Kontext“ mit 67,5 Prozent, der Kategorie „Kein Kontext“ mit 74,8 Prozent und der Kategorie „Situativer Kontext“ mit 82,1 Prozent.

Bei detaillierter Betrachtung der Antwortkombinationen ergibt sich die in Abbildung 7.11 zu sehende Verteilung: Insgesamt wurde in 47,9 Prozent der Fälle zwei Mal die Antwort „Ja“ gegeben, in 18,5 Prozent die Antwortenkombination erst „Ja“ und dann „Nein“, in 8,9 Prozent der Fälle die Kombination „Nein“ und dann „Ja“ und in 24,8 Prozent der Fälle zwei Mal „Nein“. Bei den verschiedenen Kontextarten zeigen sich variierende Ergebnisse: Bei der Kategorie „Kein Kontext“ wurde die Kombination NN mit 50,0 Prozent am häufigsten gegeben, gefolgt von JJ mit 26,7 Prozent. Bei der Kategorie „Situativer Kontext“ wurde die Kombination JJ mit 66,7 Prozent am häufigsten gegeben, die anderen drei Kombinationen liegen zwischen 7,3 (NJ) und 13,9 Prozent (NN). Bei der Kategorie „Fachlicher Kontext“ findet sich ebenfalls die Kombination JJ mit 38,0 Prozent am häufigsten, gefolgt von der Kombi-

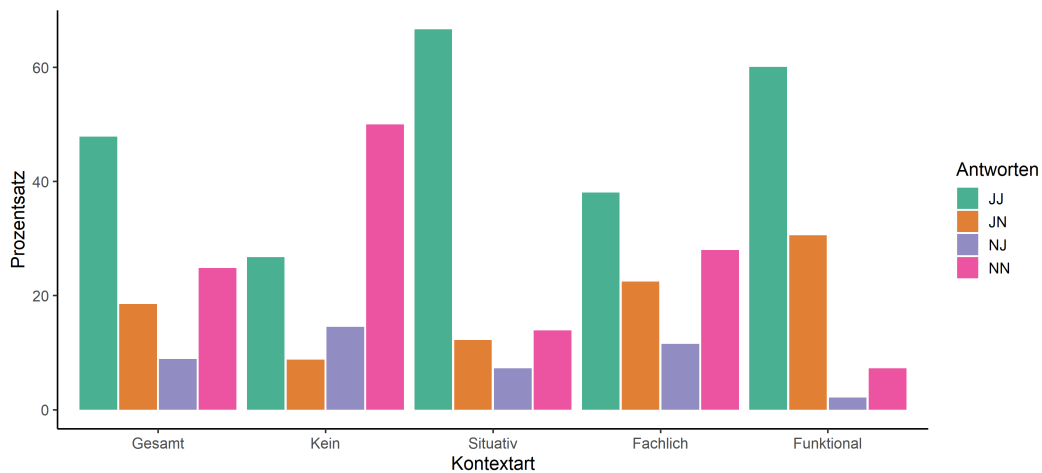


Abbildung 7.11: Welche Kombination von Antworten wurde bei den verschiedenen Kontextarten gegeben?

nation NN mit 28,0 Prozent. Auch bei der Kategorie „Funktionaler Kontext“ kommt die Kombination JJ mit 60,0 Prozent am häufigsten vor, die zweithäufigste Kombination ist in diesem Fall allerdings JN mit 30,6 Prozent.

Die in Abbildung 7.12 zu sehenden Diagramme zeigen die Details unter dem Blickwinkel, welche Antwortkombinationen bei Anzeige der verschiedenen Materialien gegeben wurden. Die Gesamtsicht in allen Diagrammen entspricht den Angaben in Abbildung 7.11, daran schließen sich die Anteile für die verschiedenen Materialien an. In allen Diagrammen ist dabei zu sehen, dass es nur geringe Schwankungen bei den Anteilen für die verschiedenen Materialien im Vergleich zur Gesamtbetrachtung gibt: Sowohl in der Gesamtbetrachtung als auch für die Kategorien „Kein Kontext“ und „Situativer Kontext“ gibt es für die beiden Materialien eine maximale Abweichung von 2 Prozent gegenüber den Anteilen insgesamt. Bei der Kategorie „Fachlicher Kontext“ ist dies ebenso, nur für die Kombination JJ liegt der Unterschied zwischen Gesamtanteil einerseits und Beschreibung bzw. Screenshot andererseits bei ca. 2,5 Prozent. Die größten Abweichungen gibt es bei der Kategorie „Funktionaler Kontext“: Hier liegt die Abweichung bei der Kombination JJ mit 68,4 Prozent bei Anzeige einer Beschreibung und 51,7 Prozent bei Anzeige eines Screenshots und bei der Kombination JN mit 22,2 Prozent bei Anzeige einer Beschreibung und 38,9 Prozent bei Anzeige eines Screenshots jeweils bei über 8 Prozent gegenüber dem Gesamtwert. Die Abweichungen für NJ und NN liegen dagegen jeweils bei max. 0,5 Prozent.

Abbildung 7.13 enthält einen Box-Whisker-Plot für den zeitlichen Abstand zwischen der Beantwortung der beiden Fragen und damit die Zeitdauer

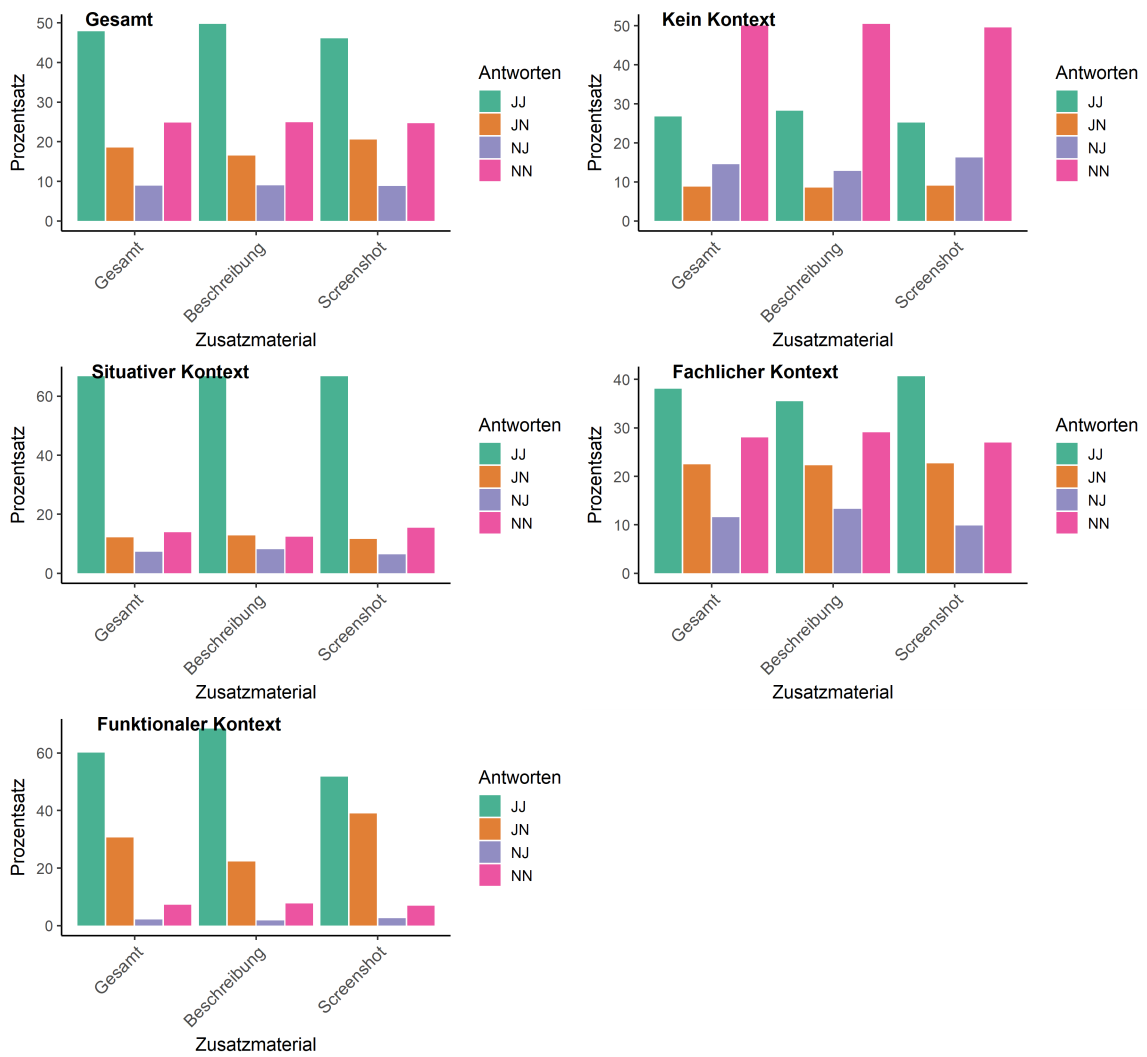


Abbildung 7.12: Welche Antwortkombinationen wurden bei den verschiedenen Kontextarten für die unterschiedlichen Materialien gegeben?

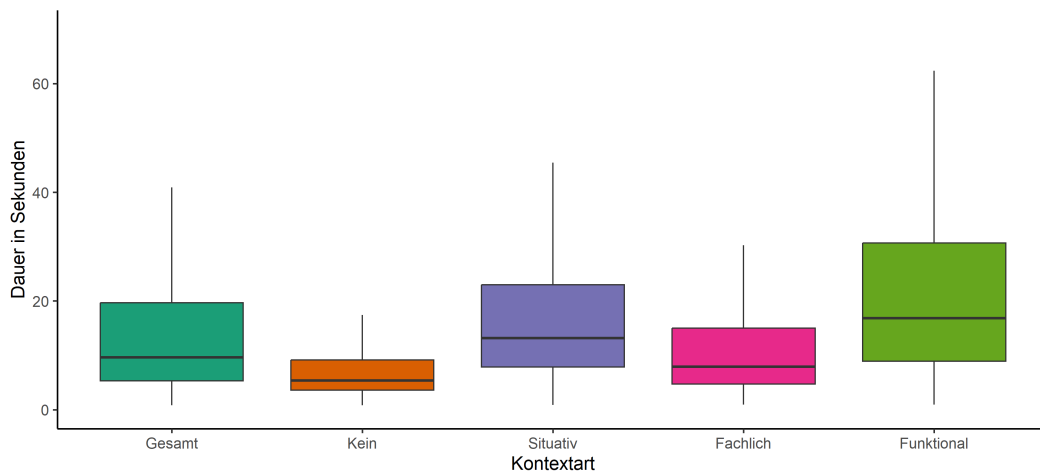


Abbildung 7.13: Dauer zwischen Beantwortung von Frage 1 und Frage 2

er, die für das Rezipieren des Materials und die ggf. nötige Überarbeitung der Übersetzung eines Strings nötig war. Für die Darstellung dieses Plots wurde der Wert von P03 beim String „Page format“ entfernt, weil dieser Wert mit mehr als 774 Sekunden weit außerhalb des Bereichs aller anderen Werte lag, was an einem Fehler bei der Datenerhebung lag. Zudem wurden die Outliers entfernt, um die Übersichtlichkeit zu verbessern.

Der Median für den zeitlichen Abstand zwischen der Beantwortung der beiden Fragen liegt für alle Strings insgesamt bei 9,6 Sekunden, der Interquartilsabstand beträgt 14,4 Sekunden, die Standardabweichung liegt bei 17,1 Sekunden. Für die Kategorie „Kein Kontext“ liegt der Median bei 5,3 Sekunden und der Interquartilsabstand beträgt 5,5 Sekunden, die Standardabweichung hat den Wert 8,3 Sekunden. Die Kategorie „Fachlicher Kontext“ weist einen Median von 7,9 Sekunden und einen Interquartilsabstand von 10,3 Sekunden auf. Die Standardabweichung beträgt 13,6 Sekunden. Die Kategorien „Situativer Kontext“ mit den Werten 13,2, 15,1 und 14,2 Sekunden und „Funktionaler Kontext“ mit 16,8, 21,8 und 23,8 Sekunden sind dagegen deutlich höher.

Die Ergebnisse zu Beantwortung der Fragen nach angezeigtem Zusatzmaterial und Bearbeitungsdauer lassen sich zu einem Streudiagramm (Abbildung 7.14) kombinieren, in dem die Dauer für die unterschiedliche Beantwortung der Fragen mit den verschiedenen Kontextarten in Verbindung gebracht und gleichzeitig die Nutzung der Zusatzmaterialien farblich unterschiedlich markiert wird.

Zusätzlich zu diesen Ergebnissen aus den TXT-Dateien, in denen die Daten aus dem Skript gespeichert wurden, finden sich in Abbildung 7.15 bereits

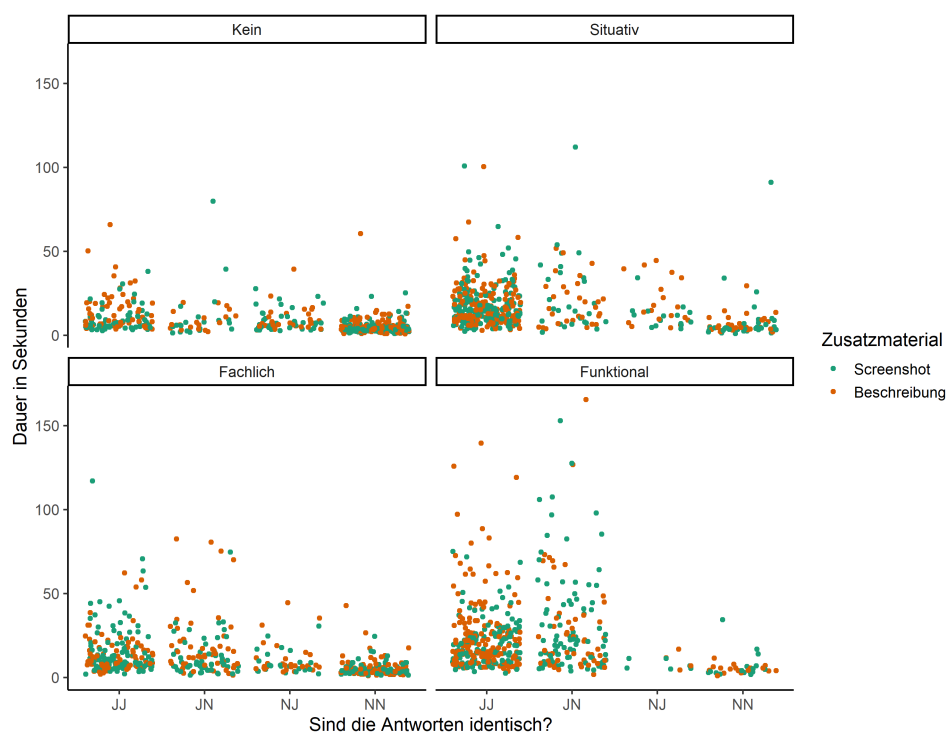


Abbildung 7.14: Dauer und Beantwortung der Fragen nach Kontextarten und Materialien

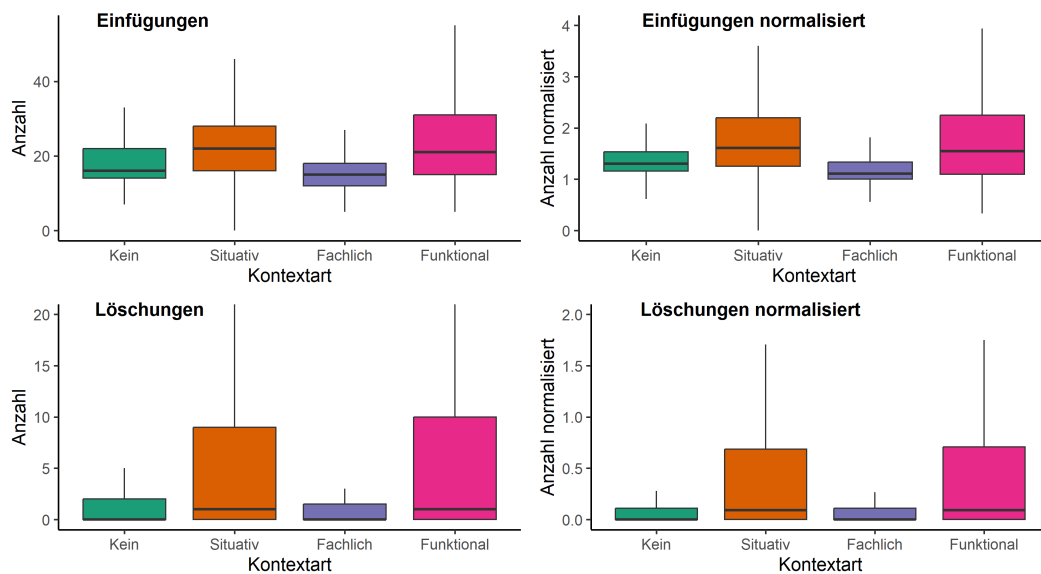


Abbildung 7.15: Anzahl der manuell eingefügten bzw. gelöschten Zeichen

erste Ergebnisse aus dem Key-Logging, die ebenfalls einen Einblick in den Nutzen der Zusatzmaterialien geben (die weiteren Ergebnisse dieser Auswertung finden sich in Abschnitt 7.6.3):⁸ Oben links ist die Anzahl der manuell eingefügten Zeichen (Spalte „Ins“ in den Dateien mit Sitzungsinformationen mit der Dateierweiterung „*.ss“) und unten links die Anzahl der manuell gelöschten Zeichen (Spalte „Del“) zu sehen (vgl. Carl, Schaeffer und Bangalore 2016, S. 21, in dieser Veröffentlichung allerdings noch als „Mins“ bzw. „Mdel“ bezeichnet), wofür jeweils die Outliers entfernt wurden. Rechts sind die Daten nach AT-Zeichen normalisiert angegeben.⁹ Verwendet wurden dafür die Daten, die über die CRITT-TPR-Datenbank generiert wurden (s. Abschnitt 7.6.1 und vgl. Carl, Schaeffer und Bangalore 2016). Da davon auszugehen ist, dass ein größerer Teil der Einfügungen und Löschungen auf Basis des rezipierten Zusatzmaterials erfolgte, ist deren Betrachtung im Zusammenhang mit der Frage des Nutzens der verschiedenen Zusatzmaterialien in diesem Abschnitt sinnvoll.

Tabelle 7.1 enthält Median, Interquartilsabstand (IQR) und Standardabweichung (SD) für diese Messwerte zur Anzahl der manuell eingefügten bzw. gelöschten Zeichen jeweils gesamt und aufgeteilt auf die vier Kontextarten,

⁸Für diese und die folgenden Berechnungen, die auf den Eye-Tracking- und Key-Logging-Daten basieren, wurden die 19 Datensätze mit mindestens drei tauglichen Teilprojekten herangezogen, wie in Abschnitt 7.6.1 erläutert.

⁹Informationen zur Normalisierung auf Basis der AT-Zeichen finden sich in Abschnitt 7.6.3

Tabelle 7.1: Anzahl der manuell eingefügten bzw. gelöschten Zeichen

	Median	IQR	SD
Anzahl manuell eingefügte Zeichen			
Gesamt	18,0	10,0	10,5
Kein Kontext	16,0	8,0	6,9
Situativer Kontext	22,0	12,0	10,4
Fachlicher Kontext	15,0	6,0	6,4
Funktionaler Kontext	21,0	16,0	13,7
Anzahl manuell eingefügte Zeichen normalisiert			
Gesamt	1,3	0,7	0,8
Kein Kontext	1,3	0,4	0,5
Situativer Kontext	1,6	1,0	0,8
Fachlicher Kontext	1,1	0,3	0,5
Funktionaler Kontext	1,6	1,2	1,0
Anzahl manuell gelöschte Zeichen			
Gesamt	0,0	4,0	7,5
Kein Kontext	0,0	2,0	4,7
Situativer Kontext	1,0	9,0	8,2
Fachlicher Kontext	0,0	1,5	5,0
Funktionaler Kontext	1,0	10,0	9,8
Anzahl manuell gelöschte Zeichen normalisiert			
Gesamt	0,0	0,3	0,6
Kein Kontext	0,0	0,1	0,4
Situativer Kontext	0,1	0,7	0,7
Fachlicher Kontext	0,0	0,1	0,4
Funktionaler Kontext	0,1	0,7	0,7

einschließlich der nach AT-Zeichen normalisierten Werte.

Im folgenden Abschnitt finden sich die weiteren Ergebnisse, die aus den Eye-Tracking- und Key-Logging-Daten hervorgehen.

7.6.3 Eye-Tracking und Key-Logging

Als erster Schritt bei der Auswertung der Eye-Tracking- und Key-Logging-Daten wurden die Daten der Dateien mit Segmentinformationen (Dateierweiterung „*.sg“) betrachtet, die über die CRITT-TPR-Datenbank erstellt wurden (s. Abschnitt 7.6.1 und vgl. Carl, Schaeffer und Bangalore 2016, S. 21). Da die einzelnen Segmente den Strings verschiedener Kontextarten entsprechen, ist dabei wie in Abschnitt 7.6.2 der Vergleich der verschiedenen

Kontextarten möglich.

Für die folgenden Auswertungen wird jeweils ein normalisierter Wert zum Vergleich angegeben. Analog zu Berechnungen von Carl und Schaeffer, die bei einer Auswertung nach Gesamtlesedauer pro Wort auf Basis der Zeichen normalisieren (vgl. Carl und Schaeffer 2017a, S. 95), wurde in der vorliegenden Studie ebenfalls auf Basis der Zeichen der AT-Strings normalisiert. Dabei wurden zwar die Leerzeichen zwischen den Wörtern in den Strings mitgezählt, aber die Zeichen abgezogen, die die Angabe der Programmtypen inkl. Doppelpunkt und anschließendem Leerzeichen umfassen, da diese sowohl in AT als auch in ZT enthalten waren und lediglich die eigentlichen AT-Strings übersetzt wurden. Dadurch wurden mögliche Verzerrungen reduziert, die durch die unterschiedliche Länge der Strings (zwischen 7 und 25 Zeichen) hervorgerufen wurden. Die Programmtypen wiesen darüber hinaus auch unterschiedliche Längen auf, die zwischen 8 Zeichen und 18 Zeichen, inkl. Doppelpunkt und Leerzeichen danach, lagen. In ähnlichen Studien, z. B. Koponen u. a. 2012 (Sekunden pro Wort normalisiert anhand der Anzahl der ZT-Tokens) oder Vieira 2016 (Fixationen pro Wort normalisiert anhand der Anzahl der AT-Wörter), wurden Werte ebenfalls normalisiert, allerdings anhand von Wörtern bzw. Tokens. Da aber bei der vorliegenden Studie im Unterschied zu diesen Studien nicht mit ganzen Sätzen, sondern mit Zwei- und Drei-Wort-Strings gearbeitet wurde, wurde die Normalisierung analog zu Carl und Schaeffer (vgl. Carl und Schaeffer 2017a) anhand der AT-Zeichenlänge als geeignetere Größe erachtet.

Zunächst wurde in Abbildung 7.16 die Bearbeitungszeit nach Kontextarten ohne (links) und mit (rechts) Normalisierung betrachtet, die ein ähnliches Bild wie der zeitliche Abstand zwischen der Beantwortung der beiden Fragen oben (s. Abbildung 7.13) bietet. Basis dafür war die Spalte „Dur“ aus der CRITT-TPR-Datenbank (vgl. Carl, Schaeffer und Bangalore 2016, S. 21). „Dur“ steht hierbei allerdings im Unterschied zur Gesamtbearbeitungsdauer der Projekte, die oben vorgestellt wurde und bei der die Dateien mit Sitzungsinformationen (Dateierweiterung „*.ss“) genutzt wurden, für die Zeit, die für die Produktion des Zieltexts für ein bestimmtes Segment benötigt wurde, also vom ersten bis zum letzten Tastaturanschlag bei dem Segment, einschließlich etwaiger Revisionen.¹⁰ Für die Abbildung wurden die Outliers entfernt, um die Darstellung übersichtlicher zu machen. Informationen zur Interpretation der in diesem Abschnitt vorgestellten Ergebnisse finden sich in Abschnitt 7.7.

¹⁰Vgl. dazu die Erläuterung dieses Messwerts in der Tabelle der Messwerte unter der Adresse <https://sites.google.com/site/centretranslationinnovation/tpr-db/features>; letzter Zugriff: 27.03.2025.

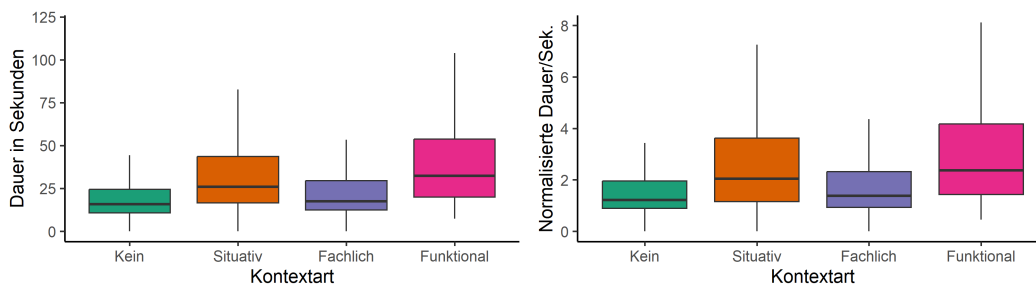


Abbildung 7.16: Bearbeitungsdauer der Segmente nach Kontextarten

Tabelle 7.2: Bearbeitungsdauer der Segmente nach Kontextarten

	Median	IQR	SD
Bearbeitungsdauer in Sekunden			
Gesamt	21,8	24,6	23,5
Kein Kontext	15,8	13,7	13,7
Situativer Kontext	25,9	27,0	22,1
Fachlicher Kontext	17,5	17,2	17,9
Funktionaler Kontext	32,4	33,9	30,6
Bearbeitungsdauer nach Zeichen normalisiert in Sekunden			
Gesamt	1,7	1,9	2,0
Kein Kontext	1,2	1,1	1,1
Situativer Kontext	2,0	2,5	2,0
Fachlicher Kontext	1,4	1,4	1,5
Funktionaler Kontext	2,4	2,7	2,7

In Tabelle 7.2 finden sich Median, Interquartilsabstand (IQR) und Standardabweichung (SD) für die Bearbeitungsdauer. Die nach AT-Zeichen normalisierten Werte werden separat aufgeführt.

Vergleicht man diese Auswertung mit der in Abbildung 7.13, zeigen sich in ersterer kürzere Bearbeitungszeiten, da darin die Dauer zwischen der Beantwortung der beiden Fragen dargestellt ist, was nur einen Teil der in Abbildung 7.16 dargestellten Gesamtsegmentbearbeitungsdauer abdeckt. In der Form ähneln sich die Box-Whiskers-Plots aber deutlich.

In Ergänzung zu den in Abschnitt 7.1.2 genannten Werten enthält Abbildung 7.17 in der linken Spalte oben die Anzahl der Fixationen auf die Tokens des Ausgangstexts (Spalte „FixS“) und darunter die Anzahl der Fixationen auf die Tokens des Zieltexts (Spalte „FixT“). Darunter findet sich die Gesamtlesedauer für den Ausgangstext (Spalte „TrtS“) und ganz unten

die Gesamtlesedauer für den Zieltext (Spalte „TrtT“, vgl. Carl, Schaeffer und Bangalore 2016, S. 21), alle jeweils ohne Outliers. In der rechten Spalte der Abbildung sind die zugehörigen Diagramme mit den wie oben beschrieben normalisierten Werten zu sehen.

Tabelle 7.3 können Median, Interquartilsabstand (IQR) und Standardabweichung (SD) für diese Messwerte zu Anzahl und Dauer der Fixationen jeweils gesamt und für die vier Kontextarten und auch normalisiert nach AT-Zeichen entnommen werden.

Tabelle 7.3: Anzahl der Fixationen und Gesamtlesedauer

	Median	IQR	SD
Anzahl Fixationen AT			
Gesamt	8,0	7,0	8,6
Kein Kontext	6,0	5,0	4,7
Situativer Kontext	8,0	8,0	8,0
Fachlicher Kontext	7,0	7,0	7,3
Funktionaler Kontext	11,0	10,5	11,4
Anzahl Fixationen AT normalisiert			
Gesamt	0,6	0,6	0,7
Kein Kontext	0,5	0,4	0,4
Situativer Kontext	0,6	0,7	0,7
Fachlicher Kontext	0,5	0,6	0,6
Funktionaler Kontext	0,8	0,7	0,8
Anzahl Fixationen ZT			
Gesamt	13,0	11,0	10,8
Kein Kontext	11,0	8,0	7,9
Situativer Kontext	15,5	10,8	9,6
Fachlicher Kontext	12,0	9,5	8,3
Funktionaler Kontext	17,0	12,5	14,8
Anzahl Fixationen ZT normalisiert			
Gesamt	1,1	0,9	0,9
Kein Kontext	0,9	0,6	0,6
Situativer Kontext	1,2	0,9	0,9
Fachlicher Kontext	0,9	0,7	0,7
Funktionaler Kontext	1,2	0,9	1,3
Gesamtlesedauer AT in ms			
Gesamt	4.588	6.270	7.176
Kein Kontext	3.094	4.005	3.845
Situativer Kontext	4.897	6.498	5.895

Tabelle 7.3: (Fortsetzung)

	Median	IQR	SD
Fachlicher Kontext	3.639	5.068	6.769
Funktionaler Kontext	7.587	9.720	9.541
Gesamtlesedauer AT in ms normalisiert			
Gesamt	352	499	614
Kein Kontext	261	309	312
Situativer Kontext	370	511	550
Fachlicher Kontext	286	433	625
Funktionaler Kontext	561	682	785
Gesamtlesedauer ZT in ms			
Gesamt	8.391	6.894	7.026
Kein Kontext	6.850	5.184	5.308
Situativer Kontext	9.228	7.815	6.578
Fachlicher Kontext	7.160	5.780	6.054
Funktionaler Kontext	10.363	8.420	8.786
Gesamtlesedauer ZT in ms normalisiert			
Gesamt	637	567	579
Kein Kontext	541	428	410
Situativer Kontext	746	651	557
Fachlicher Kontext	540	478	498
Funktionaler Kontext	740	644	745

Für die weiteren Auswertungen wurden die Daten aus den Dateien mit Informationen zu den AT-Tokens (Dateierweiterung „*.st“) betrachtet, die wie die anderen über die CRITT-TPR-Datenbank erstellt wurden (s. Abschnitt 7.6.1 und vgl. Carl, Schaeffer und Bangalore 2016, S. 23). Hierfür wurden aus den Dateien mit der Dateierweiterung „*.st“ sämtliche AT- und ZT-Tokens gelöscht, die aus Teilen der Programmtypen inkl. Bindestriche und Doppelpunkte bestanden. Auf diese Weise konnte die Auswertung auf die Tokens beschränkt werden, die tatsächlich übersetzt wurden.

In Abbildung 7.18 ist links die Dauer der ersten Fixation für AT-Tokens (Spalte „FFDur“, vgl. Carl, Schaeffer und Bangalore 2016, S. 29) zu sehen. Rechts befindet sich ein Diagramm mit der Verweildauer für AT-Tokens, also der Summe aller Fixationsdauern für ein AT-Token von der ersten Fixation bis zum Blick auf ein anderes Token (Spalte „FPDurS“, vgl. Carl, Schaeffer und Bangalore 2016, S. 29). Für diese Diagramme wurden die Outliers entfernt.

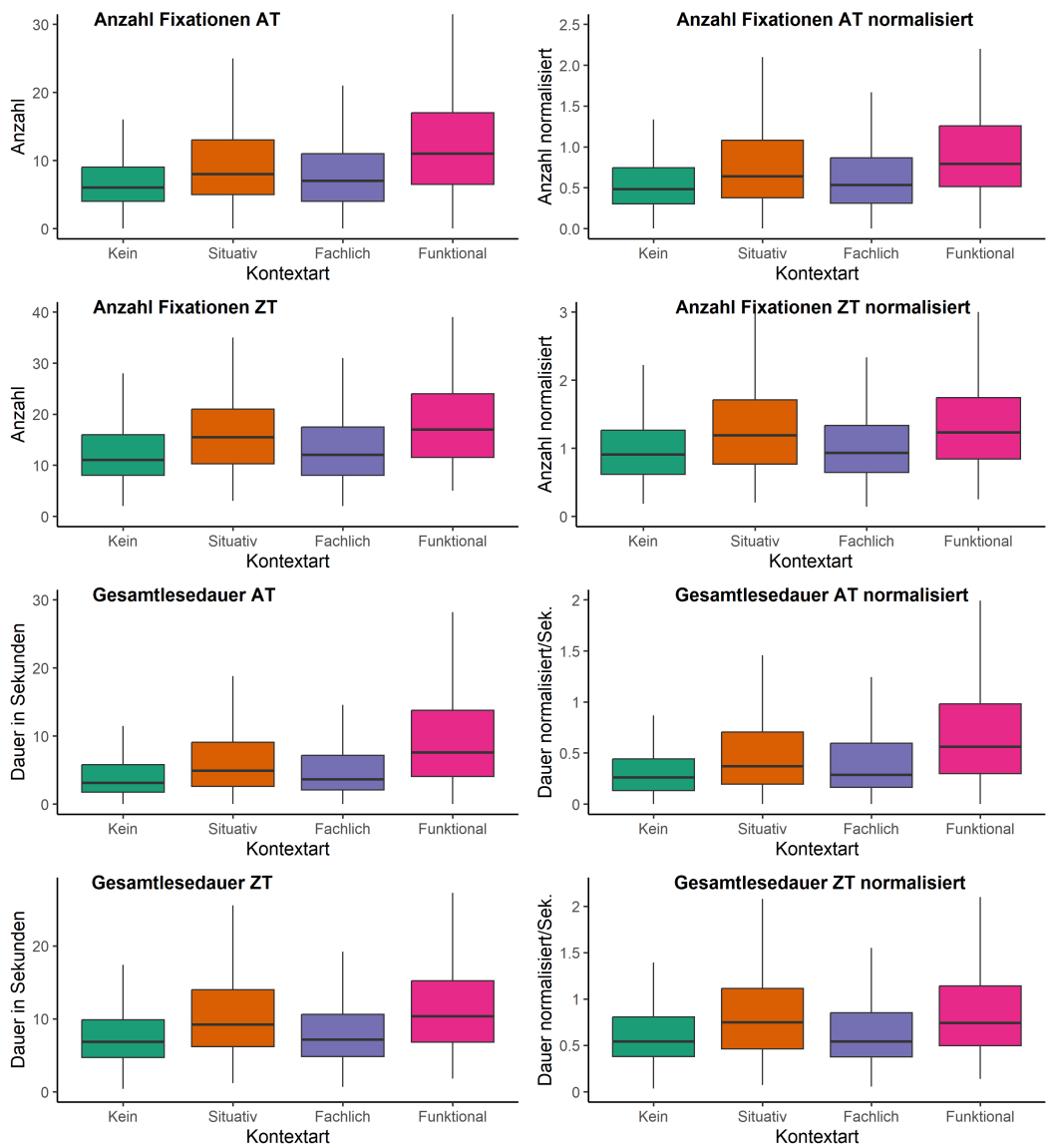


Abbildung 7.17: Anzahl der Fixationen und Gesamtlesedauer für AT- und ZT-Tokens

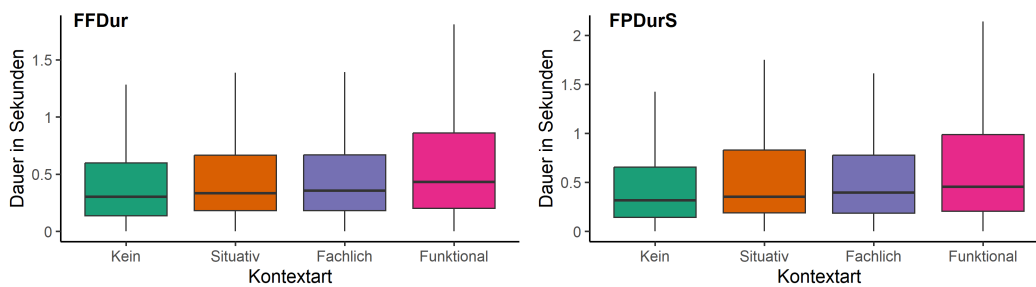


Abbildung 7.18: Dauer der ersten Fixation („FFDur“) und Verweildauer („FPDurS“) für AT-Tokens

Tabelle 7.4: Dauer der ersten Fixation („FFDur“) und Verweildauer („FPDurS“) für AT-Tokens

	Median	IQR	SD
FFDur in ms			
Gesamt	348	519	786
Kein Kontext	302	462	641
Situativer Kontext	332	484	681
Fachlicher Kontext	356	487	711
Funktionaler Kontext	432	659	1.017
FPDurS in ms			
Gesamt	380	619	912
Kein Kontext	316	515	809
Situativer Kontext	352	640	773
Fachlicher Kontext	396	592	753
Funktionaler Kontext	456	783	1.183

Der zugehörigen Tabelle 7.4 können Median, Interquartilsabstand (IQR) und Standardabweichung (SD) für die Dauer der ersten Fixation („FFDur“) und die Verweildauer („FPDurS“) für AT-Tokens jeweils gesamt und für die vier Kontextarten entnommen werden.

Zuletzt folgen in Abbildung 7.19 die Wertewerte für die Word Translation Entropy, die mit dem Messwert „HTra“ (s. gleichnamige Spalte „HTra“, vgl. Carl, Schaeffer und Bangalore 2016, S. 29–33) ermittelt wird: Links ist ein Diagramm mit den Werten für „HTra“ zu sehen, die errechnet werden, indem die in der Gruppe der Probandinnen festzustellenden Übersetzungsalternativen und damit die Wahrscheinlichkeit ermittelt wird, mit der eine dieser Alternativen gewählt wird: Bei einer hohen Zahl an Alternativen, die jeweils

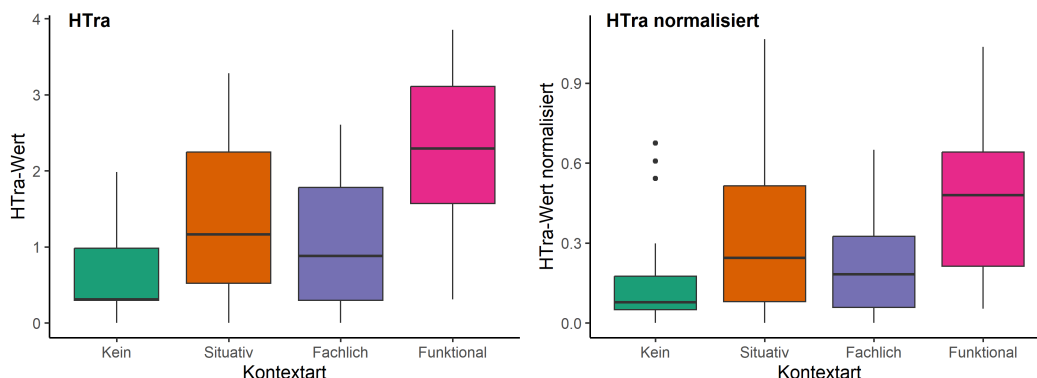


Abbildung 7.19: Word Translation Entropy: „HTra“ und „NHTra“

in wenigen Fällen genutzt werden, ist „HTra“ hoch, im umgekehrten Fall niedrig (vgl. Schaeffer, Dragsted u. a. 2016, S. 191). Im rechten Diagramm ist der auf dieser Basis errechnete Messwert „NHTra“ zu sehen, der den nach AT-Zeichen normalisierten Wert von „HTra“ darstellt.

Die Tabelle 7.5 enthält Median, Interquartilsabstand (IQR) und Standardabweichung (SD) für die Word Translation Entropy sowohl als Messwert „HTra“ als auch als nach AT-Zeichen normalisierter Messwert „NHTra“ jeweils gesamt und für die vier Kontextarten.

In den folgenden Abschnitten erfolgt die Interpretation der hier vorgestellten Ergebnisse zu Zusatzmaterialien und Recherche einerseits und nach Eye-Tracking und Key-Logging andererseits.

7.7 Interpretation der Ergebnisse der Eye-Tracking- und Key-Logging-Studie

In diesem Abschnitt findet sich die detaillierte Interpretation der in Abschnitt 7.6 vorgestellten Ergebnisse aus den Zusatzmaterialien mit Fokus auf Recherche (Abschnitt 7.7.1) sowie Eye-Tracking und Key-Logging mit Fokus auf kognitivem Aufwand (7.7.2).

Eine wichtige Basis für die Interpretation dieser Ergebnisse bilden die in Abschnitt 7.4 vorgestellten TICQ-Ergebnisse. Diese lassen sich so zusammenfassen, dass die Probandinnen weitestgehend die für die Studie avisierten Voraussetzungen erfüllt haben: Sie haben eine Ausbildung bzw. Prüfung im Bereich Übersetzen und arbeiten bis auf einen Fall als Übersetzerinnen. Darüber hinaus ist ihre Selbsteinschätzung in den Bereichen Übersetzen und Lokalisieren bis auf die besprochenen Ausnahmen sehr gut, etwas besser al-

Tabelle 7.5: Word Translation Entropy: „HTra“ und „NHTra“

	Median	IQR	SD
HTra			
Gesamt	1,16	1,96	1,08
Kein Kontext	0,31	0,68	0,56
Situativer Kontext	1,16	1,72	0,99
Fachlicher Kontext	0,88	1,48	0,87
Funktionaler Kontext	2,29	1,54	1,07
NHTra			
Gesamt	0,21	0,42	0,36
Kein Kontext	0,08	0,13	0,16
Situativer Kontext	0,24	0,43	0,34
Fachlicher Kontext	0,18	0,27	0,43
Funktionaler Kontext	0,48	0,43	0,34

lerdings für die Lokalisierung als für die Übersetzung. Darüber hinaus ist die Gruppe der Probandinnen auch im Hinblick auf ihre Fähigkeiten bei Übersetzungsprozess und Qualitätssicherung sehr homogen. Damit unterscheidet sich die an dieser Studie beteiligte Probandinnengruppe deutlich von der Probandinnengruppe beim vorangegangenen Übersetzungstest (s. Abschnitt 6.2), was für die Interpretation der Ergebnisse hilfreich sein wird und womit gleichzeitig die beim Übersetzungstest genannten Einschränkungen hinsichtlich der Probandinnengruppe, fehlende Erfahrung im Bereich Translation allgemein und fehlende professionelle Erfahrung im Besonderen mit der speziellen Textsorte GUI-Text (s. Abschnitt 6.6), vermieden werden.

7.7.1 Zusatzmaterialien und Recherche

Als Einstieg zur Interpretation der Daten in Bezug auf die angezeigten Zusatzmaterialien muss gesagt werden, dass viele Probandinnen bei den Erläuterungen zum Ablauf der Studie sowie zu den eingeblendeten Zusatzmaterialien und den Antwortmöglichkeiten darauf betonten, dass aus ihrer Sicht mehr Referenzmaterial immer sinnvoll sei. Dies könnte als Zeichen dafür interpretiert werden, dass bei der Lokalisierung von GUI-Strings in aller Regel mit zu wenig Referenzmaterial gearbeitet werden muss und dass daher mehr Referenzmaterial in jedem Fall als Arbeitserleichterung und zur Verbesserung der Qualität erwünscht ist. Um die Beantwortung der Fragen zum Zusatzmaterial im Rahmen der Studie dennoch möglichst einheitlich zu gestalten, wurde versucht, sämtliche Probandinnen bei der Einführung so zu instruieren,

dass sie im Sinne der Studie diese Fragen auf die einzelnen Strings und ihren konkreten diesbezüglichen Bedarf nach weiteren Informationen beziehen sollten. Dass sich bei der Auswertung Unterschiede zwischen den verschiedenen Kontextkategorien zeigten, kann als Hinweis darauf verstanden werden, dass diese Herangehensweise weitgehend erfolgreich war.

Sehr deutlich zeigten sich bereits die Unterschiede für die verschiedenen Kontextkategorien bei der ersten Frage, nämlich ob für die korrekte Übersetzung mehr Informationen benötigt würden (s. Abschnitt 7.6.2). Für die Kategorie „Kein Kontext“ wurde diese Frage in etwas mehr als einem Drittel der Fälle mit „Ja“ beantwortet, in den anderen in weitaus mehr Fällen, wobei sich auch zwischen den Kategorien deutliche Unterschiede ergaben. Dass von diesen verbleibenden Kontextarten die Kategorie „Fachlicher Kontext“ die war, in der mit drei Fünfteln in den wenigsten Fällen Recherche als nötig erachtet wurde, erscheint plausibel, da es hier um die Übersetzung der Fachtermini ging, die den Probandinnen je nach Vorkenntnissen sicherlich teils bekannt war. Eine ähnliche Einschätzung legten auch die Ergebnisse der Studie mit Studierenden nahe (s. Abschnitt 6.5.1). Die Probleme in den Kategorien „Situativer Kontext“ bzw. „Funktionaler Kontext“, in denen die Antwort „Ja“ in nochmals substantiell mehr Fällen gegeben wurde, ergeben sich im Unterschied dazu aus dem jeweiligen Produkt und der konkreten Verwendung eines Strings, sind also für alle Übersetzerinnen gleichermaßen bekannt bzw. unbekannt (vgl. beispielsweise die in Beste 2006, S. 73, und in Anhang 1 in Melby und Foster 2010, S. 1, für den situativen und beispielsweise in K.-D. Schmitz 2005b, S. 42, Drewer und Ziegler 2014, S. 178–179, und K. J. Dunne 2009, S. 202, für den funktionalen Kontext beschriebenen Probleme). In diesen Bereichen stellen sich eher die Fragen, ob der Übersetzerin erstens das vorliegende Problem bewusst ist und ob sie zweitens eine taugliche übersetzerische Lösung sieht, die ohne weitere Informationen auskommt, sofern sie die erste Frage bejahen kann. Dies scheint bei der Kategorie „Situativer Kontext“ eher der Fall zu sein als bei der Kategorie „Funktionaler Kontext“, was letztlich auch dem entspricht, dass die Probleme des situativen Kontexts in mehr Veröffentlichungen und auch konkreter beschrieben werden als die Probleme des funktionalen Kontexts, bei denen das in der Studie in der Mehrzahl der Strings der Kategorie „Funktionaler Kontext“ annotierte Feature `unklare_bedeutung` eher am Rande thematisiert wird (s. dazu die in Abschnitt 4.6 aufgeführten Quellen für die darin angegebenen Probleme beim Übersetzen von GUI-Text).

Die zweite Frage, ob die angezeigten Informationen für die korrekte Übersetzung hilfreich waren, sollten die Probandinnen unabhängig davon beantworten, wie sie die erste Frage beantwortet hatten. Hier bestätigte sich für die Kategorie „Kein Kontext“ der Trend aus der ersten Frage, indem in et-

was mehr als zwei Fünfteln aller Fälle weder Beschreibung noch Screenshot als nützlich erachtet wurden, wobei die Unterschiede zwischen Beschreibung und Screenshot minimal waren. Ebenso gering waren auch die Unterschiede zwischen Beschreibung und Screenshot bei den Kategorien „Fachlicher Kontext“ und „Situativer Kontext“. Bei der Kategorie „Fachlicher Kontext“ wurde allerdings lediglich in rund der Hälfte der Fälle das Material als nützlich eingeschätzt, bei der Kategorie „Situativer Kontext“ in knapp drei Viertel der Fälle.

Die Ursache dafür, dass in der Kategorie „Fachlicher Kontext“ nur in etwa der Hälfte der Fälle die angezeigten Informationen als hilfreich erachtet wurden, könnte ebenfalls im oben beschriebenen Sachverhalt liegen: Im Fall, dass den Probandinnen ein Fachterminus unbekannt war, hätten sie eher Zugriff auf zweisprachige Ergebnisse terminologischer Recherche benötigt, nämlich konkrete Übersetzungsvorschläge im passenden fachlichen Kontext (vgl. Ottmann 2005, S. 104, und Esselink 2000, S. 53, 401–402). Solche Informationen waren in den bereitgestellten Materialien aber nicht enthalten. Dennoch könnte das Material insofern als nützlich betrachtet worden sein, dass es Hinweise für eine konkretere Einschätzung dahingehend liefert, ob die Problemwörter tatsächlich der Fachterminologie angehören, was zu größerer Klarheit beim Übersetzungsversuch führen könnte (s. Abschnitt 4.5, in dem die in der Literatur genannten Referenzmaterialien verschiedenen Kontextarten zugeordnet werden).

Der hohe Nutzen der angezeigten Materialien in der Kategorie „Situativer Kontext“ leuchtet insofern ein, als dass sich die übersetzerischen Probleme des situativen Kontexts zwar einerseits gut in Form von Screenshots, aus dem der multimodale Kontext ersichtlich ist, darstellen und damit auch klären lassen, Beschreibungen der jeweiligen Funktionen hierfür aber andererseits ebenso hilfreiche Anhaltspunkte bieten, da sich der multimodale Kontext auch textlich beschreiben lässt. Beispielsweise weist der String „Duplicate email addresses“ in Teilprojekt 1 die Kategorie „Situativer Kontext“ auf, weil das Wort „Duplicate“ verschiedenen Wortarten angehören kann, nämlich Verb („duplizieren“, „kopieren“) oder Adjektiv („doppelt“). Die angezeigte Beschreibung dafür lautet „Title of a dialog containing a list of email addresses occurring more than once“ und im Screenshot ist zu sehen, dass der String der Titel eines Dialogfelds ist, in dem in einer Liste eine Adresse enthalten ist. Beide Informationstypen tragen so zur Lösung des vorliegenden Problems und damit zur Senkung der Entropie bei. Wei weist in Bezug auf die Übersetzung von Wörtern mit hohen HTra-Werten, also hoher Entropie, in herkömmlichen Texten einen ähnlichen Effekt nach: Diese sind in Activity Units mit Wörtern mit niedriger Entropie enthalten, wodurch sich die durchschnittliche Entropie der Activity Units verringert, was er so interpretiert, dass der sprachli-

che Kontext zum Sammeln zusätzlicher Informationen zur Disambiguierung dient, weshalb es Blickdaten auf andere Wörter im direkten linguistischen Kontext gibt (vgl. Wei 2021, S. 197). In Ermangelung eines solchen linguistischen Kontexts wirken die angezeigten Zusatzmaterialien auf ähnliche Weise entropiesenkend.

Die Ergebnisse in der Kategorie „Funktionaler Kontext“ unterscheiden sich von den anderen drei Kategorien dadurch, dass textliche Beschreibungen mit einer Zustimmung von mehr als zwei Dritteln als deutlich hilfreicher erachtet wurden als Screenshots mit einer Zustimmung von knapp über der Hälfte. Auch dieses Ergebnis lässt sich mit den speziellen übersetzerischen Problemen des funktionalen Kontexts erklären, bei denen sich die zugrunde liegende Funktion recht gut textlich beschreiben lässt, die Darstellung im Screenshot aber deutlich weniger Mehrwert bietet. Beispielsweise weist der String „All tags mode“ in Teilprojekt 1 die Kategorie „Funktionaler Kontext“ auf und hat eine unklare Bedeutung, weil der Zusammenhang der drei Wörter unklar ist. Die Beschreibung dazu enthält Folgendes: „Menu item for switching to a view mode where all tags are shown as icons“ und im Screenshot ist zu sehen, dass der String ein Menüpunkt im Menü „View“ ist. Ersteres ist zur Auflösung der Unklarheit und damit zur Senkung der Entropie analog zu Weis oben beschriebenen Erkenntnissen (vgl. Wei 2021, S. 197) deutlich hilfreicher als letzteres.

Damit zeigt sich schon, dass der Nutzwert von Screenshots und textlichen Beschreibungen für die verschiedenen Kontextarten als unterschiedlich betrachtet wird. Die lässt sich auch durch die weiteren Auswertungen noch bestätigen bzw. genauer eingrenzen. In allen Fällen wurde in einer deutlichen Mehrheit die Antwort nicht geändert, was einerseits so interpretiert werden kann, dass die Probandinnen in vielen Fällen bereits nach der Erstübersetzung ohne weitere Informationen sehr gut einschätzen konnten, ob sie weitere Informationen benötigten und welche das sein müssten, und andererseits, dass sie in der Mehrzahl der Fälle diese Informationen auch zu nutzen wussten, unabhängig davon, ob es sich dabei um Screenshots oder Beschreibungen handelte. Wenig erstaunlich ist nach den obigen Erklärungen, dass die Kategorie „Fachlicher Kontext“ im Vergleich aller Kontextarten die wenigsten unveränderten Antworten aufweist. Die diesbezüglichen Auswertungen zeigen, dass in einem relevanten Teil der Fälle Informationen benötigt wurden, die dann aber nicht in geeigneter Form bereitgestellt wurden. Bei der Auswertung in Bezug auf die Frage, ob nach Anzeigen des Zusatzmaterials die Antwort geändert wurde, ist daneben in erster Linie noch der Unterschied in der Kategorie „Funktionaler Kontext“ zwischen Beschreibung und Screenshot interessant, insbesondere vor dem Hintergrund, dass sich bei den anderen Kategorien nur kleine Unterschiede zeigten. In dieser Katego-

rie blieben bei Beschreibungen über drei Viertel der Antworten unverändert, bei Screenshots allerdings lediglich etwas weniger als 60 Prozent, was die obige Auswertung bestätigt und sich ebenso durch die obige Erläuterung erklären lässt. Beschreibungen sind in dieser Kategorie also entropiesenkender als Screenshots, was auch Auswirkungen auf den kognitiven Aufwand haben könnte.

Weitere Bestätigungen der eben angeführten Zusammenhänge lassen sich in der Auswertung nach den genauen Antworten für beide Fragen finden. In der Kategorie „Kein Kontext“ überwiegen wie zu erwarten die Antworten NN und JJ deutlich. In mehr als drei Vierteln aller Fälle wird eine dieser Antworten angegeben, in der Hälfte der Fälle die Antwort NN. Dies bestätigt die Erläuterungen dahingehend, dass in dieser Kategorie einerseits wenig Bedarf nach mehr Material bestand und andererseits bei Bedarf das angezeigte Material als nützlich erachtet wurde. Bei der Kategorie „Situativer Kontext“ ist mit zwei Dritteln ganz deutlich die Variante JJ vorherrschend: Die Probandinnen waren sich in der Mehrzahl der Fälle der übersetzerischen Probleme in dieser Kategorie bewusst und sahen daher weit überwiegend Informationsbedarf, der auch vom Material gedeckt wurde, unabhängig davon, ob es eine Beschreibung oder ein Screenshot war. Bei der Kategorie „Funktionaler Kontext“ wurde in etwa 90 Prozent der Fälle eine der beiden Kombinationen JJ oder JN angegeben, wobei erstere mit etwa 60 Prozent zwei Drittel dieser Werte ausmacht: Hier wurde also weit überwiegend Informationsbedarf gesehen, der in vielen Fällen über textliche Beschreibungen gedeckt wurde, in einem gewissen Teil aber nicht bzw. nicht so gut über Screenshots. Nur bei der Kategorie „Fachlicher Kontext“ ist die Verteilung nicht so deutlich wie bei den anderen Kategorien. JJ überwiegt mit knapp 40 Prozent, gefolgt von NN mit etwas weniger als 30 Prozent und JN mit etwas mehr als 20 Prozent: Es wurde Bedarf gesehen und ein gewisser Nutzen wurde dem Material eingeräumt, in vielen Fällen wurden die benötigten Informationen aber nicht bereitgestellt.

Die Einzelbetrachtung getrennt nach Beschreibung und Screenshot bestätigt die Unterschiede der verschiedenen Kategorien in Bezug auf den Nutzen der unterschiedlichen Materialien: Nur bei der Kategorie „Funktionaler Kontext“ zeigen sich diesbezüglich größere Unterschiede, was nochmals hervorhebt, dass im Unterschied zu den anderen Kontextarten, bei denen sich nur sehr geringe Abweichungen zwischen Beschreibung und Screenshot zeigen, speziell für diese Kategorie die textlichen Beschreibungen als deutlich hilfreicher erachtet wurden als Screenshots.

Vor diesem Hintergrund lassen sich auch die Unterschiede in der Zeit zwischen der Beantwortung der beiden Fragen für die einzelnen Kontextarten betrachten: In Fällen, in denen überwiegend kein Zusatzmaterial als nötig

erachtet wurde, wurde dieses vermutlich weniger lange betrachtet und gingen daraus auch weniger und weniger zeitintensive Änderungen hervor (vgl. dazu auch die Analyse im folgenden Abschnitt 7.7.2). Wo Material benötigt wurde, wurde Zeit darauf verwendet, dieses zu rezipieren und ggf. die Übersetzung daraufhin anzupassen, was zu einer längeren Bearbeitungsdauer führte. Gemäß den obigen Ergebnissen betrifft dies insbesondere die Kategorien „Situativer Kontext“ und „Funktionaler Kontext“, wobei für letztere noch mehr Zeit nötig war. Betrachtet man die oben genannten Beispiele für die Kategorie „Situativer Kontext“ („Duplicate email addresses“) und für die Kategorie „Funktionaler Kontext“ („All tags mode“) mit den angegebenen textlichen Beschreibungen, erscheint dies verständlich: Im ersten Fall, einer Wortartambiguität, muss anhand der Beschreibung lediglich ermittelt werden, welche Wortart im vorliegenden Fall genutzt wird (vgl. beispielsweise Beste 2006, S. 73), im zweiten muss bei einer unklaren Bedeutung dagegen ein inhaltliches Verständnis entwickelt werden, um einen tauglichen Übersetzungsvorschlag zu entwickeln (vgl. K. J. Dunne 2009, S. 202), was vermutlich auch ein Grund dafür ist, warum die Bearbeitungsdauer in der Kategorie „Funktionaler Kontext“ nochmal höher war als in der Kategorie „Situativer Kontext“.

Dass in den Kategorien „Situativer Kontext“ und „Funktionaler Kontext“, bei denen in den meisten Fällen Material für nötig erachtet wurde und auch die höchste Bearbeitungsdauer festzustellen war, häufiger Änderungen vorgenommen wurden, wird durch die Key-Logging-Auswertung der Einfügungen und Löschungen bestätigt. Bei diesen beiden Kategorien wurden sowohl deutlich mehr Zeichen eingefügt als auch gelöscht, wobei insgesamt nur sehr wenige Zeichen tatsächlich gelöscht wurden. Auch hier bestätigt sich die Zwischenstellung der Kategorie „Fachlicher Kontext“.

Zusammenfassend lässt sich also zunächst sagen, dass die Nullhypothese zurückgewiesen werden kann: In den Ergebnissen der Studie zeigt sich, dass der Nutzen von Screenshots und alternativ textlichen Beschreibungen für die verschiedenen Kontextarten unterschiedlich groß ist. Darüber hinaus ist festzustellen, dass auch der Bedarf an Referenzmaterial für die verschiedenen Kontextarten als unterschiedlich betrachtet wird. In vielen Fällen zeigten die Probandinnen bereits zu Beginn der Übersetzung eines Strings eine gute Einschätzung dahingehend, ob sie Material benötigten, und konnten das Material auch für die Beantwortung ihrer Fragen nutzen bzw. alternativ klar erkennen, ob und inwiefern das angezeigte Material von Nutzen für die Qualität ihrer Übersetzung war. Die Kategorie „Kein Kontext“ wurde mit wenig Materialbedarf in kurzer Zeit und mit wenig Änderungen bearbeitet. Bei den Kategorien „Situativer Kontext“ und „Funktionaler Kontext“ war der Materialbedarf deutlich höher, die Bearbeitung dauerte länger und umfasste mehr

Änderungen. Im ersteren Fall gab es keine Bevorzugung eines bestimmten Materialtyps, im letzteren wurde klar die Beschreibung als hilfreicher angesehen. Nur bei der Kategorie „Fachlicher Kontext“ waren die Ergebnisse weniger eindeutig, was sich einerseits darauf zurückführen lässt, dass die Probandinnen entweder mit der benötigten Terminologie vertraut waren oder nicht und damit nur in einem Teil der Fälle Bedarf für mehr Material gesehen wurde. Andererseits wurde dieser Bedarf durch die bereitgestellten Materialien nicht oder nur unzureichend gedeckt.¹¹

Auf Basis dieser Studie lässt sich also bestätigen, dass die in der Literatur genannten Referenzmaterialien bei der Lösung von Kontextproblemen hilfreich sind: Sowohl eine lauffähige Version der zu übersetzenden Software (vgl. Ottmann 2005, S. 102, und Esselink 2000, S. 53) oder Simulationen und Demos, falls dies nicht möglich ist (vgl. Ottmann 2005, S. 102), bzw. die Arbeit mit Softwarelokalisierungstools (vgl. z. B. Ottmann 2005, S. 102–103, Beste 2006, S. 93–94, Wahle 2000b, S. 37–38, 40, K.-D. Schmitz 2005a, S. 9, und Schildhauer 2005, S. 124) als auch die Dokumentation des Softwareprodukts in Form von Hilfeinformationen bzw. Onlinehilfe (vgl. Ottmann 2005, S. 104, Esselink 2000, S. 53, und Wahle 2000b, S. 45) oder auch Referenzhandbücher, die Beschreibungen der einzelnen Strings enthalten (vgl. Ottmann 2005, S. 104), können dazu beitragen, die Kontextprobleme bei der Übersetzung von GUI-Strings zu lösen. Ergänzt werden muss dies durch geeignete Glossare, wie etwa Terminologie für die relevanten Betriebssysteme, die Software oder das Fachgebiet (vgl. Ottmann 2005, S. 104, und Esselink 2000, S. 53, 401–402). Auf Basis dieser Studie lässt sich allerdings auch feststellen, dass textliche Beschreibungen in Form von Dokumentation den größten Nutzwert bei der Übersetzung von Kontextproblemen haben, da sie für alle Kontextprobleme hilfreich sind, wenn auch in unterschiedlichem Maße. Darstellungen des multimodalen Kontexts, hier beispielsweise als Screenshots aus der GUI der zu übersetzenden Software, zeigen sich dagegen als weniger hilfreich, weshalb es etwas verwunderlich erscheint, dass Softwarelokalisierungstools in der Literatur zur Softwarelokalisierung teils als Allheilmittel erscheinen. Da-

¹¹Studien aus der Translationsprozessforschung, die diese Ergebnisse bestätigen, konnten im Rahmen der Recherche zur vorliegenden Arbeit allerdings nicht identifiziert werden, weil es in diesem Bereich nur wenig Forschung mit Blick auf Hilfsmittel zu geben scheint. Eine Studie von Daems et al., die bei der Recherche zur vorliegenden Arbeit rezipiert wurde, befasst sich zwar mit Produktivität und Qualität bei Nutzung von Hilfsmitteln, dies wird aber im Vergleich von zwei Studierendengruppen und für Humanübersetzung einerseits und maschinelle Übersetzung andererseits erforscht (vgl. Daems u. a. 2016), lässt sich damit also nicht auf die vorliegende Arbeit übertragen. Dasselbe gilt für eine weitere Studie von Daems et al., bei der ebenfalls im Kontext von maschineller Übersetzung und Postediting u. a. die Recherche in externen Quellen betrachtet wird, aber erneut nur mit Studierenden als Probandinnen (vgl. Daems u. a. 2018).

bei zeigen sie sich nur für einen eingeschränkten Teil der in dieser Arbeit überprüften Probleme als nützlich, für die textliche Beschreibungen gleichermaßen tauglich sind, und kommen zudem primär bei nicht oder nicht ausreichend internationalisierter Software zum Einsatz (vgl. K. J. Dunne 2009, S. 191), was einer anderen Forderung aus der Literatur widerspricht, nämlich der einer umfassenden Internationalisierung als Vorbereitung auf die Lokalisierung, um letztere schnell und mit geringem Aufwand zu erstellen (vgl. beispielsweise K.-D. Schmitz 2005a). Hinzu kommt, dass es bei allen GUIs eine Gruppe von Strings wie Nachrichten gibt, die nicht in festen Bereichen der GUI angesiedelt sind und somit auch nicht visuell lokalisiert werden können (vgl. beispielsweise Esselink 2000, S. 59, und s. Abschnitt 2.1.4).

Vergleicht man diese Ergebnisse mit den Ergebnissen des Übersetzungstests (s. Abschnitt 6.4), in dem die Recherche bereits eine Rolle gespielt hat, fällt auf, dass die Studierenden insgesamt in deutlich weniger Fällen eine Recherche durchführten, also weitere Informationen benötigten.¹² Man könnte erwarten, dass eher der umgekehrte Fall vorliegt, dass nämlich die Studierenden mehr recherchieren, weil sie weniger eingearbeitet sind und sich weniger sicher fühlen.¹³ Es ist aber auch umgekehrt möglich, dass ihnen die Recherchekompetenz speziell im Bereich der Softwarelokalisierung fehlt und sie daher nicht wissen, in welchen Quellen sie in welchen Fällen auf welche Weise recherchieren sollen, wobei beim Übersetzungstest gleichzeitig auch die knappe Zeit dafür gesorgt haben kann, dass weniger recherchiert wurde, als für nötig erachtet wurde.

Die Unterschiede bei der Recherche zwischen beiden Studien zeigen sich besonders stark in den Kategorien „Kein Kontext“, „Situativer Kontext“ und „Funktionaler Kontext“: In der ersten dieser Kategorien haben die Studierenden lediglich in einem Fünftel aller Fälle eine Recherche durchgeführt,

¹²Einschränkend muss allerdings gesagt werden, dass bei der Eye-Tracking-Studie nur ein Subset der Strings aus dem Übersetzungstest verwendet wurde (s. Abschnitt 7.2). Die Daten können also nicht eins zu eins verglichen werden.

¹³Im Zusammenhang mit Postediting maschineller Übersetzung gehen Daems et al. davon aus, dass die maschinelle Übersetzung den Übersetzungsprozess erleichtert und beschleunigt und weisen in diesem Zusammenhang mit Verweis auf eine Studie von Garcia (vgl. Garcia 2011) darauf hin, dass professionelle Übersetzerinnen weniger vom Postediting profitieren als Übersetzungsstudierende, also ihr Produktivitätsgewinn kleiner ist als der von Übersetzungsstudierenden. Auf dieser Basis vermuten sie u. a., dass Unsicherheiten im Bezug auf die Qualität der MT-Ausgabe zu mehr konsultierten Ressourcen führt, was sich wiederum negativ auf die Produktivität auswirkt (vgl. Daems u. a. 2016, S. 113). Diese Aussagen könnten analog zu der hier geäußerten Vermutung gedeutet werden. Allerdings arbeiten sowohl Garcia als auch Daems et al. in den hier zitierten Studien ausschließlich mit Studierenden und bei Garcia spielen Hilfsmittel keine Rolle, während sie bei Daems et al. lediglich zur Demonstration einer neuen Methode und beim Vergleich zweier Studierendengruppen genutzt werden. Diese Aussagen scheinen also wenig stichhaltig zu sein.

während die professionellen Übersetzerinnen in etwas mehr als einem Drittel weiteren Informationsbedarf sahen. In der zweiten Kategorie lag der Recherchebedarf der Studierenden bei etwas weniger als der Hälfte der Fälle, der der professionellen Übersetzerinnen bei über drei Vierteln. In der Kategorie „Funktionaler Kontext“ war der Recherchebedarf der professionellen Übersetzerinnen um 15 Prozent höher als der der Studierenden. Nur in der Kategorie „Fachlicher Kontext“ gab es keine nennenswerten Unterschiede. Dies kann sich durch die Unterschiede in der translatorischen Kompetenz mit der speziellen Textsorte GUI-Text erklären lassen: Insbesondere in der Kategorie „Situativer Kontext“ ist ein Teil der Übersetzungsprobleme eher verdeckt und ist damit auch Erfahrung im Bereich Softwarelokalisierung nötig, um sie und damit weiteren Recherchebedarf zu erkennen. Ein Beispiel dafür sind die ambigen Wortarten (vgl. beispielsweise Beste 2006, S. 73, und s. Abschnitt 4.6). Die größere Erfahrung der professionellen Übersetzerinnen könnte also zu besserem Erkennen der speziellen übersetzerischen Probleme der Softwarelokalisierung und damit zu mehr Recherchebedarf führen. Diese Erläuterung könnte auch auf die Unterschiede bei der Kategorie „Funktionaler Kontext“ zutreffen. Die Unterschiede bei der Kategorie „Kein Kontext“ sind dagegen unklarer. Hier besteht die Möglichkeit, dass die professionellen Übersetzerinnen etwas übervorsichtig agieren, vielleicht aufgrund ihrer Erfahrungen mit den Problemen beim Übersetzen von Strings, evtl. auch in der speziellen Situation als Probandinnen einer Studie, wobei diese Faktoren auch zu den erhöhten Werten bei den anderen Kontextarten hätten beitragen können. Diese Diskrepanz bestätigt letztlich auch die oben erwähnte Aussage von vielen Probandinnen bei der Einführung in die Studie, dass sie mehr Informationen immer für sinnvoll hielten. Dass sich in der Kategorie „Fachlicher Kontext“ kaum Unterschiede zeigen, könnte an der Erläuterung oben liegen, dass den Probandinnen die Terminologie entweder bekannt oder unbekannt war. Dieselbe Erklärung kann wie beschrieben auch auf die Studierenden zutreffen, da diese im Unterricht und auch privat mit großer Wahrscheinlichkeit Kontakt mit verschiedenen Programmtypen und der zugehörigen Fachterminologie haben (s. Abschnitt 6.5.1). Es ist also durchaus denkbar, dass ihnen genauso ein Teil der benötigten Terminologie bekannt war wie den professionellen Übersetzerinnen.

Der Rechercheerfolg lässt sich weniger gut vergleichen, da in beiden Studien sehr unterschiedliche Recherchewege und -materialien bereitstanden. Dass die Recherche nach Ansicht der Studierenden insbesondere für die Kategorien „Kein Kontext“ und „Fachlicher Kontext“ mit um die drei Viertel der Fälle überwiegend erfolgreich war, passt zu den obigen Erläuterungen, ebenso wie die Probleme beim Recherchieren der Kategorien „Situativer Kontext“ und „Funktionaler Kontext“, für die diese Studie deutlich hilfreiche und maß-

geschneiderte Materialien bot. Der größere subjektive Rechercheerfolg der Studierenden bei der Kategorie „Fachlicher Kontext“ lässt sich ebenso durch die unterschiedlichen Recherchewege erläutern: Die Studierenden hatten über das Internet unter anderem Zugriff auf Terminologiedatenbanken, mit denen sich diese Probleme besser lösen ließen.

7.7.2 Eye-Tracking, Key-Logging und kognitiver Aufwand

Die Basis für die Auswertung der Eye-Tracking- und Key-Logging-Daten stellte die Nullhypothese dar, dass der kognitive Aufwand bei allen Kontextarten gleich hoch ist.¹⁴ Für diese Nullhypothese wurden analog zu Schaeffer et al. folgende Eye-Tracking-Werte aus der CRITT-Datenbank zur Translationsprozessforschung (vgl. Carl, Bangalore und Schaeffer 2016) überprüft: „TrtS“, „FFDur“, „FPDur“ und „HTra“ (vgl. Schaeffer, Dragsted u. a. 2016). Darüber hinaus wurde analog zu Carl und Schaeffer auch „Dur“ überprüft (vgl. Carl und Schaeffer 2017b, S. 51).

Bei der Bearbeitungsdauer nach Kontextarten, die über den Wert „Dur“ der Dateien mit Segmentinformationen (Dateierweiterung „*.sg“) ermittelt wurde, bestätigt sich zunächst das Bild, das sich bereits bei der Auswertung der Zeit zwischen der Beantwortung der beiden Fragen in Abschnitt 7.6.2 zeigte: Die Kontextarten weisen auch bei dieser Auswertung unterschiedliche Bearbeitungszeiten auf, wobei die Kategorie „Kein Kontext“ den kleinsten Medianwert für die Bearbeitungszeit zeigt, direkt gefolgt von der Kategorie „Fachlicher Kontext“. Die Kategorien „Situativer Kontext“ und „Funktionaler Kontext“ weisen dagegen deutlich höhere Werte auf.

Auch die Anzahl der Fixationen für AT- und ZT-Tokens („FixS“ und „FixT“) sowie die Gesamtlesedauer („TrtS“ und „TrtT“) entsprechen diesem Bild: Die Anzahl der Fixationen für AT und ZT unterscheiden sich, wobei für die Kategorien „Kein Kontext“ und „Fachlicher Kontext“ die niedrigsten Werte vorliegen, für die Kategorie „Situativer Kontext“ höhere und für die Kategorie „Funktionaler Kontext“ die höchsten. Allerdings liegen die nach Zeichen normalisierten Werte für die Anzahl an Fixationen eng beisammen und es gibt in drei Kategorien fast oder genau doppelt so viele Fixationen für die ZT-Tokens, nur in der Kategorie „Funktionaler Kontext“ lediglich 1,5 mal mehr. Die Gesamtlesedauer ist für die ZT-Tokens ebenso beinahe doppelt so

¹⁴Beziehungsweise folgende Alternativhypothese: Der kognitive Aufwand unterscheidet sich bei allen vier Kontextkategorien dahingehend, dass er bei der Kategorie „Kein Kontext“ am geringsten ist und über die Kategorien „Fachlicher Kontext“ und „Situativer Kontext“ bis hin zur Kategorie „Funktionaler Kontext“ ansteigt.

hoch wie für die AT-Tokens, allerdings mit einer deutlichen Ausnahme bei der Kategorie „Funktionaler Kontext“, bei der sie nicht mal 1,5 mal so hoch ist. Im Unterschied zur Anzahl der Fixationen ist die Gesamtlesedauer bei den Kategorien „Situativer Kontext“ und „Funktionaler Kontext“ deutlich höher als bei den anderen Kategorien. Der Fokus der Fixationen scheint also bei allen Probandinnen und allen Kategorien auf dem ZT zu liegen, wobei die Kategorie „Funktionaler Kontext“ in Relation zu den anderen einen geringeren Anstieg in Anzahl der Fixationen und Gesamtlesedauer zwischen AT und ZT aufweist.

Die Dauer der ersten Fixation für AT-Tokens („FFDur“) und die Verweildauer für AT-Tokens („FPDurS“) unterscheiden sich von diesen Werten dadurch, dass hier die Werte aller Kategorien mit Ausnahme der Kategorie „Funktionaler Kontext“ nahe beieinander liegen. Letztere ist normalisiert knapp 1,5 mal so hoch.

Hiervon weichen die Werte für Word Translation Entropy („HTra“) deutlich ab: Es gibt eine klare Steigerung der „HTra“-Werte von der Kategorie „Kein Kontext“ über die Kategorien „Fachlicher Kontext“ und „Situativer Kontext“ hin zur Kategorie „Funktionaler Kontext“, wobei sich die Werte von der Kategorie „Kein Kontext“ zur Kategorie „Fachlicher Kontext“ und dann zu Kategorie „Situativer Kontext“ jeweils in etwa verdoppeln und dann hin zur Kategorie „Funktionaler Kontext“ versechsfachen. Das heißt, die Kategorie „Funktionaler Kontext“ weist eine substanziell höhere Word Translation Entropy als alle anderen Kategorien auf.

In der Studie von Schaeffer et al. wurde bereits ein statistisch signifikanter Effekt der „HTra“-Werte auf die Gesamtlesedauer für AT-Tokens („TrtS“) festgestellt (vgl. Schaeffer, Dragsted u. a. 2016, S. 205). Rydning und Lachaud hatten ergänzend dazu bereits in einer früheren Studie kleinere „FFDur“-Werte bei professionellen Übersetzerinnen im Vergleich zu bilingualen Nicht-Übersetzerinnen beim Übersetzen von polysemen Wörtern ohne Kontext identifiziert (vgl. Rydning und Lachaud 2010) und Dragsted hatte festgestellt, dass bei Wörtern mit einer höheren Anzahl von Übersetzungsalternativen sowohl die Anzahl der Fixationen als auch die Blickdauer höher als bei Wörtern mit geringer Anzahl von Übersetzungsalternativen ist (vgl. Dragsted 2012). Bei „FPDurS“ stellen Schaeffer et al. allerdings keinen Effekt durch „HTra“ fest (vgl. Schaeffer, Dragsted u. a. 2016, S. 201). Nach Carl und Schaeffer wirkt sich der „HTra“-Wert beim Übersetzen allerdings auf die Produktionszeit der Übersetzung, also auf „Dur“, aus, wobei sie diese Werte für vollständige Sätze, nicht wie bei Eddington und Tokowicz für einzelne Wörter (vgl. Eddington und Tokowicz 2013), feststellen und betonen, dass dies im Unterschied zu den kontrollierteren Experimenten von Eddington und Tokowicz eine Auswertung darstellt, die für die professionelle Übersetzung

repräsentativer ist (vgl. Carl und Schaeffer 2017b, S. 51–52).

Zusammenfassend lässt sich also sagen, dass bei dieser Studie analog zum Übersetzungstest (s. Abschnitt 6.4.4), bei dem sich bei den Strings der verschiedenen Kategorien eine unterschiedliche Anzahl an Übersetzungsalternativen zeigte, die Strings der verschiedenen Kategorien unterschiedliche Entropiewerte aufweisen, wobei die Kategorie „Kein Kontext“ die geringste Entropie aufweist, worauf die Kategorien „Fachlicher Kontext“ und „Situativer Kontext“ folgen. Die Kategorie „Funktionaler Kontext“ zeigt die mit Abstand höchste Entropie. Dies spiegelt sich in den Bearbeitungsdauern sowie der Anzahl der Fixationen und der Gesamtlesedauer für AT und ZT, bei denen die Kategorien mit höheren Entropiewerten ebenfalls höhere Werte aufweisen, wobei die Unterschiede der Kontextarten in Bezug auf die Entropie sehr viel deutlicher ausfallen als die Unterschiede dieser Eye-Tracking-Werte. Hiervon unterscheiden sich allerdings die Werte für die Dauer der ersten Fixation, da sich bei diesem Messwert nicht dieselben parallelen Unterschiede zwischen den Kontextarten finden. Dies widerspricht den Erkenntnissen in den genannten Studien, insbesondere Schaeffer et al. (vgl. Schaeffer, Dragsted u. a. 2016, S. 199).

Eine mögliche Erklärung für diese Abweichung könnte darin liegen, dass in der Translationsprozessforschung Ambiguität und Schwierigkeit häufig gleichgesetzt werden, woraus dann der kognitive Aufwand abgeleitet wird, wie auch schon in Abschnitt 7.1.2 herausgearbeitet wurde (vgl. zum Beispiel Dragsted 2012, Mishra, Bhattacharyya und Carl 2013, Carl und Schaeffer 2017b). Größere Ambiguität und damit größere „HTra“-Werte geben größere Schwierigkeit beim Übersetzen und damit größeren kognitiven Aufwand an, wie auch Wei bestätigt: „[...] translation ambiguity as indicated by HTra, which tends to increase the difficulty for human translation“ (vgl. Wei 2021, S. 178). Aus der Feststellung von Schaeffer et al., dass „HTra“ und „FFDur“ positiv signifikant korreliert sind (vgl. Schaeffer, Dragsted u. a. 2016, S. 199) kann man also schließen, dass auch die „FFDur“-Werte mit zunehmender Schwierigkeit zunehmen: Sind Wörter weniger ambig, kann davon ausgegangen werden, dass es eine gemeinsame semantische Repräsentation in Ausgangs- und Zielsprache für dieses Wort gibt (vgl. Carl und Schaeffer 2017a, S. 85). In diesem Fall liegen frühe automatische Prozesse vor, die beispielsweise mit „FFDur“ sichtbar gemacht werden können (vgl. Schaeffer, Dragsted u. a. 2016, S. 185, 199), und damit einhergehend ein geringerer kognitiver Aufwand.

In der vorliegenden Studie ist dieser Zusammenhang zwischen Ambiguität und Schwierigkeit für die Kategorie „Funktionaler Kontext“ gegeben, was auch sämtliche Ergebnisse bestätigen. Betrachtet man Strings wie „Soutstream“, „All tags mode“ und „Master gain control“ aus Teilprojekt 1, lässt

sich diese Schwierigkeit erkennen, aus der die mit Abstand höchsten „HTra“-Werte sowie zu erwartende Werte für die Anzahl der Fixationen und die Gesamtlesedauer, die erste Fixation und die Bearbeitungsdauer hervorgehen. Bei der Kategorie „Situativer Kontext“ ist die Sachlage aber anders: Zu übersetzende Strings wie „Page right“, „Columns after“ oder „Duplicate email addresses“, wie sie in Teilprojekt 1 in dieser Kategorie zu finden sind, weisen nur einen sehr geringen Schwierigkeitsgrad beim Übersetzen in der Art auf, wie er möglicherweise in den genannten Studien betrachtet wird. Solche Strings können von den professionellen Übersetzerinnen weitestgehend automatisiert übersetzt werden, was durch die im Vergleich zu den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ ähnlich niedrigen „FFDur“-Werte gezeigt wird. Gleichzeitig gibt es aber eine höhere bzw. hohe Zahl an möglichen Übersetzungsalternativen und daher auch „HTra“-Werte, die im Vergleich zu den Kategorien „Kein Kontext“ und „Fachlicher Kontext“ höher liegen. Bei der vorliegenden Studie muss also wohl zu einem gewissen Grad eine Unterscheidung zwischen schwierigem und ambigem Übersetzen gemacht werden. Dies kann gleichzeitig auch eine weitere Erklärung dafür sein, warum es in der Kategorie „Situativer Kontext“ beim Übersetzungstest weniger Fragen der Studierenden gab als bei den professionellen Übersetzerinnen dieser Studie.

Auf Grundlage dieser Erkenntnisse kann die Nullhypothese, dass der kognitive Aufwand bei allen Kontextarten gleich hoch ist, zurückgewiesen werden, mit der Einschränkung, dass sich in der vorliegenden Studie im Fall der Kategorie „Situativer Kontext“ nicht für alle in der Literatur genannten Messwerte die mit höherer Entropie einhergehende Erhöhung zeigt. Ausnahmen sind die Dauer der ersten Fixation („FFDur“) und die Verweildauer („FPDurS“) für AT-Tokens. Die Abweichung bei der Dauer der ersten Fixation könnte sich dadurch erklären lassen, dass sich in der Kategorie „Situativer Kontext“ im Unterschied zu früheren Studien frühe automatische Prozesse zeigen, obwohl die Entropie erhöht ist. Diese erhöhte Entropie wäre hier also kein Hinweis auf einen erhöhten Schweregrad der Übersetzung. Dass keine Effekte bei der Verweildauer festgestellt wurden, entspricht hingegen den Ergebnissen von Schaeffer et al., die keinen signifikanten Effekt der Word Translation Entropy auf die Verweildauer feststellen (vgl. Schaeffer, Dragssted u. a. 2016, S. 201). Entsprechend ist die Alternativhypothese, dass sich der kognitive Aufwand bei allen vier Kontextkategorien dahingehend unterscheidet, dass er bei der Kategorie „Kein Kontext“ am geringsten ist und über die Kategorien „Fachlicher Kontext“ und „Situativer Kontext“ bis hin zur Kategorie „Funktionaler Kontext“ ansteigt, nicht für die Kategorie „Situativer Kontext“ zutreffend, sondern nur für die übrigen drei.

7.8 Einschränkungen und Probleme

Für diese Studie gilt dieselbe Einschränkung in Bezug auf die Auswahl von Strings aus zwei und drei Wörtern, die auch schon im letzten Kapitel (s. Abschnitt 6.6) genannt wurde: Eine Anzahl der in der Korpusstudie in Kapitel 5 im Korpus festgestellten Phänomene, insbesondere Übersetzungsprobleme im Zusammenhang mit der Übersetzung von Komposita, konnten nicht in die Studie mit einbezogen werden. Um alle im Korpus festgestellten Phänomene mit einzubeziehen, hätte die Studie umfangreicher und mit anderen Beschränkungen erstellt werden müssen.

Insbesondere bei den Screenshots fiel zudem auf, dass manche Probandinnen sich zur Prüfung vorbeugen mussten. Daraus lässt sich schließen, dass es besser gewesen wäre, die Screenshots in größerer Darstellung in die Studie mit einzubeziehen.

Des Weiteren wurden Effekte nicht berücksichtigt, die teils in früheren Studien beim Umgang mit Cognates im Unterschied zu Non-Cognates festgestellt wurden, weshalb Cognates in anderen Studien, wie beispielsweise bei Laxen und Lavaur (vgl. Laxén und Lavaur 2010), explizit vermieden wurden. Unter anderem wurde von Macizo und Bajo festgestellt, dass beim Lesen für das Übersetzen interlinguale Homographen, die lediglich in der Zielsprache ambig sind, zu längeren Reaktionszeiten führten, Cognates aber zu kürzeren (vgl. Macizo und Bajo 2006). Boada et al. fanden dagegen heraus, dass Bilinguale schneller waren, wenn sie die dominante von mehreren Übersetzungen einzelner Wörter erkennen sollten, wobei es keine Rolle spielte, ob Cognates oder Non-Cognates angegeben waren (vgl. Boada u. a. 2013). Eine etwaige Beeinflussung durch Cognates hätte durch explizites Ausschließen oder gleichmäßige Verteilung von Cognates auf die verschiedenen Kontextarten erreicht werden können.

Ein letzter Punkt ist die bereits in Abschnitt 6.6 genannte ökologische Validität: Auch diese Studie erfüllte nicht alle Aspekte der Forderung nach ökologischer Validität, nämlich der professionellen Realität von Probandinnen zu entsprechen (vgl. beispielsweise Saldanha und O'Brien 2013, S. 33), da beispielsweise nicht mit einem Softwarelokalisierungstool gearbeitet wurde und abwechselnd Strings verschiedener Programmtypen übersetzt werden mussten, anstatt kontinuierlich Strings aus einem einzigen Softwareprodukt wie typischerweise in der professionellen Praxis. Auch in diesem Fall ermöglichte die Konzeption als kontrolliertes Experiment und damit die manipulierten Items zuverlässigere Aussagen, weil die Anzahl der Datenpunkte deutlich höher war. Einige Aspekte des Experiments waren allerdings durchaus vergleichbar mit der professionellen Praxis im Bereich Softwarelokalisierung: AT- und ZT-Strings waren wie bei vielen TM-Programmen in zwei Fens-

tern nebeneinander angeordnet und als Referenzmaterial waren Screenshots und Beschreibungen verfügbar, die bei einem Softwarelokalisierungsprojekt in der Praxis vorhanden sein können, beispielsweise durch Nutzung eines Softwarelokalisierungstools oder Bereitstellung der Dokumentation, und den Probandinnen damit in der Mehrzahl der Fälle vertraut waren.

Kapitel 8

Fazit und Ausblick

Im Rahmen der vorliegenden Arbeit wurde zunächst versucht, die Besonderheiten der Fachtextsorte GUI-Text, also der Textelemente in grafischen Benutzeroberflächen von Softwareprodukten, im Hinblick auf die Frage des Kontexts zu konkretisieren. Dabei wurden die drei folgenden Hauptkontextarten beschrieben:

- **Der situative Kontext:** Kontext, der auf der Multimodalität von GUI-Text basiert und über den Informationen auf anderen semiotischen Ebenen als der Sprache als Kontext bedeutungsrelevant sind
- **Der fachliche Kontext:** Kontext, der auf der Eigenschaft von GUI-Text als Fachtext basiert und über den Informationen aus dem fachlichen Bereich des jeweiligen Softwareprodukts als Kontext bedeutungsrelevant sind
- **Der funktionale Kontext:** Kontext, der über die Verknüpfung von GUI-Textelementen mit Programmfunktionen entsteht und über den Informationen zu diesen Programmfunktionen als Kontext bedeutungsrelevant sind

Diesen Kontextarten konnten die typischen Referenzmaterialien zugeordnet werden, die im Rahmen von Softwarelokalisierungsprojekten benötigt und gefordert werden.

In einem nächsten Schritt wurden diesen Kontextarten die in der Literatur zur Softwarelokalisierung verschiedentlich beschriebenen typischen Probleme zugeordnet, die bei der Übersetzung von GUI-Text im Zusammenhang mit Kontext stehen. Dies bildete die Basis eines Annotationsschemas, in dem diese Probleme zunächst als Features operationalisiert und dann annotiert werden konnten. Die Annotation erfolgte in einem für diese Arbeit zusammengestellten Korpus mit Ausschnitten aus zehn verschiedenen Open-Source-

Softwareprodukten aus den Kategorien Anwendungs- und Unterstützungsprogramme. Bei diesen Schritten zeigte sich, dass die Operationalisierung über die identifizierten Kontextkategorien, die diesen zugeordneten Übersetzungsprobleme und deren Features möglich ist und für Annotationen zum Einsatz kommen kann.

Bei der Auswertung der Annotationen in diesem Korpus konnte festgestellt werden, dass darin Phänomene des fachlichen Kontexts am häufigsten waren, gefolgt von denen des situativen und des funktionalen Kontexts. Ebenso konnte die Relevanz der verschiedenen Features über deren Anteile an den Annotationen der drei annotierten Kontextkategorien ermittelt werden. Zudem konnte festgestellt werden, dass über die Hälfte der Strings des Korpus Kontextannotationen enthielt, wobei kürzere Strings mehr kontextbezogene Übersetzungsprobleme aufwiesen, insbesondere solche des situativen Kontexts.

Die nächsten Schritte, ein Übertetzungstest und eine Eye-Tracking- und Key-Logging-Studie, erbrachten insbesondere in den Bereichen Recherche und Referenzmaterial, Übersetzungsalternativen bzw. Entropie sowie kognitiver Aufwand weitere Erkenntnisse, die im Folgenden skizziert werden.

8.1 Ergebnisse zu Recherche, Übersetzungsalternativen/Entropie und kognitivem Aufwand

Recherche und Referenzmaterial: In beiden Studien wurden insgesamt drei typische Referenzmaterialien getestet und ihr unterschiedlicher Nutzen für verschiedene Kontextkategorien aufgezeigt: Materialien, die Aufschluss über Aspekte der Multimodalität geben, beispielsweise in Form einer lauffähigen Version der Software bzw. eines Demos oder wie in der zweiten Studie in Form von Screenshots, waren primär für Übersetzungsprobleme des situativen Kontexts hilfreich. Fachterminologie in Form von Glossaren oder Terminologiedatenbanken diente in erster Linie für Übersetzungsprobleme des fachlichen Kontexts als Hilfe. Textliche Beschreibungen, wie sie beispielsweise in Dokumentation und Hilfeinformationen zu finden sind, konnten für Übersetzungsprobleme des funktionalen Kontexts als Referenzmaterial dienen, aber auch für solche des situativen Kontexts, da darin einerseits Programmfunktionen, aber andererseits auch Aspekte der Multimodalität beschrieben werden können. Bei Strings des funktionalen Kontexts nannten die Probandinnen beider Studien den höchsten Recherchebedarf, gefolgt von denen des fachlichen Kontexts von den Studierenden und denen des situativen Kontexts von

den professionellen Übersetzerinnen, wobei beide Studien allerdings auch unterschiedlich konzipiert waren.

Übersetzungsalternativen/Entropie: Im Rahmen der beiden Studien zeigte sich, dass die verschiedenen Kontextarten unterschiedliche Anzahlen von Übersetzungsalternativen bzw. Entropiewerte aufwiesen. Am niedrigsten war die Anzahl der Übersetzungsalternativen/Entropie bei Strings, die keine Kontextannotation aufwiesen, etwas höher bei Strings mit Annotationen des fachlichen Kontexts, danach bei denen des situativen und zuletzt bei denen des funktionalen Kontexts. Die Studierenden konnten bei allen Kontextkategorien außer bei Annotationen des funktionalen Kontexts die Anzahl der Übersetzungsalternativen durch Recherche senken, obwohl sie lediglich auf Terminologiedatenbanken Zugriff hatten. Bei der Studie mit professionellen Übersetzerinnen konnte ein solcher Effekt nicht festgestellt werden, da in dieser Studie die Entropiewerte lediglich auf Basis der finalen Übersetzungen der Probandinnen ermittelt wurden.

Kognitiver Aufwand: In der Eye-Tracking- und Key-Logging-Studie zeigte sich, dass der kognitive Aufwand von der Kontextkategorie „Kein Kontext“ über die Kategorie „Fachlicher Kontext“ bis zur Kategorie „Funktionaler Kontext“ parallel zur Entropie anstieg. Bei der Kategorie „Situativer Kontext“ lag dagegen zwar eine verhältnismäßig hohe Entropie vor, diese führte aber nicht zu einer Erhöhung aller für die Ermittlung des kognitiven Aufwands überprüften Messwerte. Ausnahmen waren die erste Fixation und die Verweildauer für AT-Tokens und Grund dafür war, dass bei dieser Kontextkategorie nicht wie beispielsweise bei der Kategorie „Funktionaler Kontext“ die Ursache der Entropie eine erhöhte Übersetzungsschwierigkeit, sondern lediglich eine erhöhte Übersetzungsambiguität war.

8.2 Profile der Kontextkategorien

In den Probandinnenstudien konnten für Strings der vier Kontextkategorien mit zwei oder drei Wörtern unterschiedliche Profile identifiziert werden:

Kein Kontext: Strings ohne Annotation wurden schnell und automatisiert mit wenig und in der Regel erfolgreicher Recherche bzw. ohne Bedarf an speziellen Referenzmaterialien übersetzt. Sowohl Entropie als auch kognitiver Aufwand waren niedrig.

Situativer Kontext: Strings mit dieser Annotation wurden schnell und automatisiert übersetzt, es war aber Recherche erforderlich, bei der sowohl Screenshots als auch textliche Beschreibungen hilfreich waren. Die Entropie war aufgrund der Ambiguität dieser Strings erhöht, der kognitiver Aufwand aber eher niedrig.

Fachlicher Kontext: Die Übersetzung von Strings mit dieser Annotation war abhängig davon, ob den Probandinnen mögliche Übersetzungen der Fachterminologie bekannt waren. Wenn dies nicht der Fall war, war die Recherche in Terminologiedatenbanken hilfreich. Entropie und kognitiver Aufwand waren eher niedrig.

Funktionaler Kontext: Die Übersetzung von Strings mit dieser Annotation nahm am meisten Zeit in Anspruch und es war Recherche erforderlich, bei der sich textliche Beschreibungen im Vergleich zu Screenshots als hilfreicher erwiesen. Die Entropie war hoch und damit einhergehend auch der kognitiver Aufwand.

8.3 Forschungsdesiderata

Unter anderem lassen sich die folgenden Forschungsdesiderata dieser Arbeit nennen:

Sinnvoll wäre es, im Rahmen der weiteren Forschungsarbeit das Annotationsschema und dessen Features weiter zu optimieren, insbesondere im Hinblick auf die in der Arbeit genannten Defizite und die Ergebnisse beim Inter-Annotator-Agreement. Zudem könnte mit weiteren Untersuchungen in vergleichbaren Korpora festgestellt werden, ob sich die hier vorgestellten Ergebnisse replizieren lassen.

Der stetig wichtiger werdende Bereich der mobilen Betriebssysteme, wie Android oder iOS, wurde bei der Korpusstudie nicht berücksichtigt. Es ist allerdings anzunehmen, dass es bei mobilen Apps ebenfalls kontextbezogene Probleme gibt. Diese könnten sich aber aufgrund anderer Anforderungen wie vorhandener Platz auf dem Display durchaus in anderer Form zeigen. Eine ähnliche Korpusstudie wie die hier durchgeführte mit einem Spezialkorpus, in dem Strings aus mobilen Apps verschiedener mobiler Betriebssysteme enthalten sind, könnte hierzu interessante Ergebnisse erbringen.

In den Probandinnenstudien der vorliegenden Arbeit wurden längere Strings und damit einige Features, die primär in längeren Strings vorkommen, insbesondere problematische Aneinanderreihungen, nicht überprüft. Gleichmaßen waren Ein-Wort-Strings, aus denen diverse Beispiele in der Literatur zur Softwarelokalisierung stammen, nicht Teil der Studien. Es wäre sinnvoll, auch diese anderen Stringtypen zu untersuchen, um die Erkenntnisse der vorliegenden Arbeit zu ergänzen.

Im Rahmen der Korpusstudie wurde angesprochen, dass Annotationen der Kategorie „Funktionaler Kontext“ unter Umständen zumindest in Teilen als Indikator für die Qualität des Ausgangstexts dienen könnten. Sowohl für die Übersetzung als auch für die Qualitätssicherung des Ausgangstexts

könnte eine solche Einschätzung nutzbringend sein. Es müsste aber festgestellt werden, ob sich diese, vielleicht für einzelne Features wie die unklaren Aneinanderreihungen, automatisieren lässt.

Ein weiterer Fragenkomplex betrifft ebenfalls die professionelle Praxis: Es ist denkbar, dass eine Annotation wie die in dieser Arbeit vorgenommene dafür genutzt werden kann, den Schweregrad von zu übersetzendem GUI-Text zu ermitteln. Wenn dies und auf dieser Basis auch Prognosen im Hinblick auf die Dauer der Übersetzung möglich sind, könnte dies in der Praxis für die Planung von Lokalisierungsprojekten sehr hilfreich sein. Ebenso könnte eine automatische Einschätzung im Hinblick auf den Schweregrad einzelner Strings Übersetzerinnen dabei unterstützen, problematischen Strings mehr Aufmerksamkeit zu widmen. Eine solche Automatisierung der Erkennung gewisser Features könnte beispielsweise bei den ambigen Wortarten recht einfach umzusetzen sein. Insbesondere für die maschinelle Übersetzung und das Postediting könnte eine solche automatische Erkennung potenziell problematischer Strings eine optimale Ergänzung sein.

Ebenso könnte weiter untersucht werden, in welcher Form die Erkenntnisse aus der vorliegenden Arbeit zur Optimierung der universitären Ausbildung und der Einarbeitung in der Lokalisierungspraxis beitragen können.

Bibliographie

- Aarts, Bas (2011). *Oxford Modern English Grammar*. Oxford: Oxford University Press.
- Adamzik, Kirsten (2016). *Textlinguistik: Grundlagen, Kontroversen, Perspektiven*. 2. Auflage. Berlin, Boston: De Gruyter.
- Albrecht, Jörn (2013). *Übersetzung und Linguistik*. 2. überarbeitete Auflage. Tübingen: Narr.
- Arntz, Reiner, Heribert Picht und Klaus-Dirk Schmitz (2014). *Einführung in die Terminologearbeit*. 7., vollständig überarbeitete und aktualisierte Auflage. Hildesheim, Zürich, New York: Olms.
- Aschenberg, Heidi (1999). „Zum Kontextbegriff in der Übersetzungsforschung“. In: *Texte und Kontexte in Sprachen und Kulturen: Festschrift für Jörn Albrecht*. Hrsg. von Norbert Greiner u. a. Trier: WVT, S. 7–33.
- Astapenko, Anna (2008). *Lokalisierung komplexer Softwaresysteme: Technik, Sprache, Kultur*. Lübeck: Schmidt-Römhild.
- Austermühl, Frank (2001). *Electronic Tools for Translators*. Hoboken: Taylor and Francis.
- Baker, Mona (2006). „Contextualization in Translator- and Interpreter-Mediated Events“. In: *Journal of Pragmatics* 38.3, S. 321–337.
- Bangalore, Srinivas u. a. (2016). „Syntactic Variance and Priming Effects in Translation“. In: *New Directions in Empirical Translation Process Research*. Hrsg. von Michael Carl, Srinivas Bangalore und Moritz Schaeffer. Cham: Springer International Publishing, S. 211–238.
- Bateman, John A., Janina Wildfeuer und Tuomo Hiippala (2017). *Multimodality: Foundations, Research and Analysis - A Problem-Oriented Introduction*. Berlin, Boston: De Gruyter.
- Behrens, Alexander (2016). *Lokalisierbarkeit von User-Interface-Strings: übersetzerische Aspekte der Internationalisierung und Lokalisierung von Software, untersucht anhand der Übersetzungsrichtungen Englisch-Deutsch und Englisch-Russisch*. Frankfurt am Main, New York: Peter Lang.

- Bernal-Merino, Miguel Á. (2015). *Translation and Localisation in Video Games: Making Entertainment Software Global*. New York: Routledge.
- Beste, Kai (2006). *Softwarelokalisierung und Übersetzung*. Trier: WVT.
- Biber, Douglas (1993). „Representativeness in Corpus Design“. In: *Literary and Linguistic Computing* 8.4, S. 1–15.
- Biber, Douglas (1995). *Dimensions of register variation: a cross-linguistic comparison*. Cambridge, New York: Cambridge University Press.
- Boada, Roger u. a. (2013). „Effect of Multiple Translations and Cognate Status on Translation Recognition Performance of Balanced Bilinguals“. In: *Bilingualism: Language and Cognition* 16.1, S. 183–197.
- Bowker, Lynne und Jennifer Pearson (2002). *Working with Specialized Language: A Practical Guide to Using Corpora*. London, New York: Routledge.
- Brinker, Klaus, Steffen Pappert und Hermann Cölfen (2024). *Linguistische Textanalyse: eine Einführung in Grundbegriffe und Methoden*. 10., neu bearbeitete Auflage. Berlin: Erich Schmidt Verlag.
- Bühler, Karl (1999). *Sprachtheorie: die Darstellungsfunktion der Sprache*. 3. Auflage, ungekürzter Neudruck der Ausgabe von Fischer 1934. Stuttgart: Lucius und Lucius.
- Burkhardt, Livia (1995). *Unbekannte Wörter in fremdsprachigen Texten: eine Untersuchung zur Rolle des Kontextes in Bedeutungserschließungsprozessen am Beispiel des Französischen*. Bochum: Brockmeyer.
- Calderón Campos, Miguel und Gael Vaamonde (2024). „Introduction. Corpus Linguistics in the Era of Big and Rich Data: Methodological Perspectives on Spanish and Portuguese“. In: *Linguistic Corpora and Big Data in Spanish and Portuguese*. Hrsg. von Miguel Calderón Campos und Gael Vaamonde. Berlin, Boston: De Gruyter, S. 1–8.
- Carl, Michael, Hrsg. (2021). *Explorations in Empirical Translation Process Research*. Cham: Springer International Publishing.
- Carl, Michael, Srinivas Bangalore und Moritz Schaeffer, Hrsg. (2016). *New Directions in Empirical Translation Process Research: Exploring the CRITT TPR-DB*. Cham: Springer International Publishing.
- Carl, Michael und Moritz Schaeffer (2017a). „Measuring Translation Literality“. In: *Translation in Transition: Between Cognition, Computing and Technology*. Hrsg. von Arnt Lykke Jakobsen und Bartolomé Mesa-Lao. Amsterdam: John Benjamins Publishing Company, S. 81–105.
- Carl, Michael und Moritz Schaeffer (2017b). „Why Translation Is Difficult: A Corpus-Based Study of Non-Literality in Post-Editing and From-Scratch Translation“. In: *HERMES - Journal of Language and Communication in Business* 56, S. 43–57.

- Carl, Michael, Moritz Schaeffer und Srinivas Bangalore (2016). „The CRITT Translation Process Research Database“. In: *New Directions in Empirical Translation Process Research*. Hrsg. von Michael Carl, Srinivas Bangalore und Moritz Schaeffer. Cham: Springer International Publishing, S. 13–54.
- Catford, John C. (1978). *A Linguistic Theory of Translation: An Essay in Applied Linguistics*. 5. Auflage. Oxford: Oxford University Press.
- Chang, Youngick (2005). *Anglizismen in der deutschen Fachsprache der Computertechnik: eine korpuslinguistische Untersuchung zu Wortbildung und Bedeutungskonstitution fachsprachlicher Komposita*. Frankfurt am Main, New York: Peter Lang.
- Coseriu, Eugenio (2007). *Textlinguistik: eine Einführung*. Hrsg. von Jörn Albrecht. 4., unveränderte Auflage. Tübingen: Narr.
- Czulo, Oliver, Silvia Hansen-Schirra und Jean Nitzke (2017). „Contrasting Terminological Variation in Post-Editing and Human Translation of Texts from the Technical and Medical Domain“. In: *Empirical Translation Studies: New Methodological and Theoretical Traditions*. Hrsg. von Gert De Sutter, Marie-Aude Lefer und Isabelle Delaere. Berlin: De Gruyter Mouton, S. 183–206.
- Czulo, Oliver und Jean Nitzke (2016). „Patterns of Terminological Variation in Post-editing and of Cognate Use in Machine Translation in Contrast to Human Translation“. In: *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*. EAMT 2016, S. 106–114.
- Daems, Joke u. a. (2016). „The Effectiveness of Consulting External Resources During Translation and Post-editing of General Text Types“. In: *New Directions in Empirical Translation Process Research: Exploring the CRITT TPR-DB*. Hrsg. von Michael Carl, Srinivas Bangalore und Moritz Schaeffer. Cham: Springer International Publishing, S. 111–134.
- Daems, Joke u. a. (2018). „How Do Students Cope with Machine Translation Output of Multiword Units? An Exploratory Study“. In: *Multiword Units in Machine Translation and Translation Technology*. Hrsg. von Ruslan Mitkov u. a. Amsterdam: John Benjamins Publishing Company, S. 61–80.
- De Beaugrande, Robert-Alain und Wolfgang Ulrich Dressler (1981). *Einführung in Die Textlinguistik*. De Gruyter.
- Dietz, Frank (2007). „'How Difficult Can That Be?' – The Work of Computer and Video Game Localization“. In: *Revista Tradumática* 5, S. 1–6.
- Dragsted, Barbara (2012). „Indicators of Difficulty in Translation — Correlating Product and Process Data“. In: *Across Languages and Cultures* 13.1, S. 81–98.

- Drewer, Petra und Wolfgang Ziegler (2014). *Technische Dokumentation: eine Einführung in die übersetzungsgerechte Texterstellung und in das Content-Management*. 2., überarbeitete und aktualisierte Auflage. Würzburg: Vogel Buchverlag.
- Dunne, Keiran J., Hrsg. (2006a). *Perspectives on Localization*. Philadelphia: John Benjamins Publishing Company.
- Dunne, Keiran J. (2006b). „Putting the Cart behind the Horse - Rethinking Localization Quality Management“. In: *Perspectives on Localization*. Hrsg. von Keiran J. Dunne. Philadelphia: John Benjamins Publishing Company, S. 95–117.
- Dunne, Keiran J. (2009). „Assessing Software Localization - For a Valid Approach“. In: *Testing and Assessment in Translation and Interpreting Studies: A Call for Dialogue between Research and Practice*. Hrsg. von Claudia V. Angelelli und Holly E. Jacobson. Amsterdam: John Benjamins Publishing Company, S. 185–222.
- Dunne, Keiran J. und Elena S. Dunne, Hrsg. (2011). *Translation and Localization Project Management: The Art of the Possible*. Amsterdam, Philadelphia: John Benjamins Publishing Company.
- Eddington, Chelsea M. und Natasha Tokowicz (2013). „Examining English–German Translation Ambiguity Using Primed Translation Recognition“. In: *Bilingualism: Language and Cognition* 16.2, S. 442–457.
- Engberg, Jan (2024). „Wie man Fach-Kontext beim Übersetzen einbezieht – und was es für ihre Überprüfung braucht“. In: *Translation im Kontext*. Hrsg. von Sigmund Kvam u. a. Berlin: Frank & Timme, S. 245–264.
- Esselink, Bert (2000). *A Practical Guide to Localization*. Überarbeitete Auflage. Amsterdam, Philadelphia: John Benjamins Publishing Company.
- Esselink, Bert (2003). „Localization and Translation“. In: *Computers and Translation: A Translator's Guide*. Hrsg. von H. L. Somers. Amsterdam, Philadelphia: John Benjamins Publishing Company, S. 67–86.
- Finkbeiner, Rita (2015). *Einführung in die Pragmatik*. Darmstadt: WBG.
- Fischer, Sylvia (2011). „Verständlich Übersetzen“. In: *Fachzeitschrift 'technische kommunikation'* 1/2011, S. 38.
- Fix, Ulla (2001). „Zugänge zu Stil als semiotisch komplexer Einheit: Thesen, Erläuterungen und Beispiele“. In: *Perspektiven auf Stil*. Hrsg. von Eva-Maria Jakobs und Anneli Rothkegel. Tübingen: Max Niemeyer Verlag, S. 113–126.
- Fix, Ulla (2008). „Nichtsprachliches Als Textfaktor: Medialität, Materialität, Lokalität“. In: *Zeitschrift für germanistische Linguistik* 36.3, S. 343–354.
- Freigang, Karl-Heinz (1997). „Zum Stellenwert von Softwarelokalisierungsprojekten in der Übersetzer Ausbildung“. In: *Translationsdidaktik: Grund-*

- fragen der Übersetzungswissenschaft*. Hrsg. von Eberhard Fleischmann, Wladimir Kutz und Peter A. Schmitt. Tübingen: Narr, S. 122–132.
- Freigang, Karl-Heinz und Uwe Reinke (2005). „Translation-Memory-Systeme in der Softwarelokalisierung“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke und Klaus-Dirk Schmitz. Tübingen: gnv, Narr, S. 55–71.
- Gagola, Kamila (2017). *Bedeutung von Kontext bei der Softwarelokalisierung - Eine Untersuchung am Beispiel der Übersetzung von SAP Sports One*. Unveröffentlichte Masterarbeit, Johannes Gutenberg-Universität Mainz.
- Gambier, Yves (2021). „Context, Contextualisation and (Multimodal) Text“. In: *Studies about Languages* 39, S. 5–18.
- Gambier, Yves und Olli Philippe Lautenbacher (2024). „Text and Context Revisited within a Multimodal Framework“. In: *Babel. International Journal of Translation* 70.1–2, S. 1–16.
- Garcia, Ignacio (2011). „Translating by Post-Editing: Is It the Way Forward?“. In: *Machine Translation* 25.3, S. 217–237.
- Göpferich, Susanne (1995). *Textsortenkonventionen in den Fachsprachen der Naturwissenschaften und der Technik*. Tübingen: Narr.
- Göpferich, Susanne (2002). „Lokalisierung und Übersetzung: Abgrenzung - Zuständigkeiten - Ausbildung“. In: *Lokalisierung von technischer Dokumentation*. Hrsg. von Jörg Hennig und Marita Tjarks-Sobhani. Lübeck: Schmidt-Römhild, S. 27–41.
- Göpferich, Susanne (2006). „Text, Textsorte, Texttyp“. In: *Handbuch Translation*. Hrsg. von Mary Snell-Hornby u. a. Zweite, verbesserte Auflage, unveränderter Nachdruck der 2. Auflage 1999. Tübingen: Stauffenburg, S. 61–64.
- Göpferich, Susanne (2008). *Textproduktion im Zeitalter der Globalisierung: Entwicklung einer Didaktik des Wissenstransfers*. 3. Auflage. Tübingen: Stauffenburg.
- Gumm, Heinz-Peter und Manfred Sommer (2016). *Programmierung, Algorithmen und Datenstrukturen*. Berlin, Boston: De Gruyter Oldenbourg.
- Haapaniemi, Riku (2024). „Translation as Meaning-Construction under Co-Textual and Contextual Constraints: A Model for a Material Approach to Translation“. In: *Translation Studies* 17.1, S. 20–36.
- Halliday, Michael Alexander Kirkwood und Ruqaiya Hasan (1989). *Language, Context, and Text: Aspects of Language in a Social-Semiotic Perspective*. Oxford: Oxford University Press.
- Hansen-Schirra, Silvia, Sandra Hansen u. a. (2009). „Fachkommunikation, Popularisierung, Übersetzung: Empirische Vergleiche am Beispiel der Nominalphrase im Englischen und Deutschen“. In: *Linguistik Online* 39.3 (3), S. 109–118.

- Hansen-Schirra, Silvia und Jean Nitzke (2021). „Translation, the Process–Product Interface and Cognition“. In: *The Routledge Handbook of Translation and Cognition*. Hrsg. von Fábio Alves und Arnt Lykke Jakobsen. London, New York: Routledge, S. 589–614.
- Heilmann, Arndt u. a. (2019). „Shorter than a Text, Longer than a Sentence: Source Text Length for Ecologically Valid Translation Experiments“. In: *Target* 31.1, S. 98–124.
- Hildebrand, Simona (2020). *Übersetzungsverfahren für Softwareoberflächen - Versuch einer kontextuellen Kategorisierung von Übersetzungsproblemen bei der Softwarelokalisierung von Open-Source-Produkten aus dem Englischen ins Deutsche*. Unveröffentlichte Masterarbeit, Johannes Gutenberg-Universität Mainz.
- Hirschmann, Hagen (2019). *Korpuslinguistik: eine Einführung*. Stuttgart: J.B. Metzler Verlag.
- House, Juliane (2006). „Text and Context in Translation“. In: *Journal of Pragmatics* 38.3, S. 338–358.
- Irmeler, Ulrike und Doris Hartwig (2000). „Sprachliche Qualität im lokalisierten Softwareprodukt: Kriterien und Vorgehensweisen der Qualitätssicherung“. In: *Softwarelokalisierung*. Hrsg. von Klaus-Dirk Schmitz und Kirsten Wahle. Tübingen: Stauffenburg, S. 89–100.
- Jacob, Gunnar u. a. (2024). „The Psycholinguistics of Shining-through Effects in Translation: Cross-Linguistic Structural Priming or Serial Lexical Co-Activation?“ In: *Applied Psycholinguistics* 45.3, S. 542–566.
- Jakobsen, Arnt Lykke und Bartolomé Mesa-Lao, Hrsg. (2017). *Translation in Transition: Between Cognition, Computing and Technology*. Amsterdam, Philadelphia: John Benjamins Publishing Company.
- Jiménez-Crespo, Miguel A. (2013). *Translation and Web Localization*. London, New York: Routledge.
- Jiménez-Crespo, Miguel A. (2024). *Localization in Translation*. Abingdon, Oxon, New York: Routledge.
- Kalverkämper, Hartwig (1997). „Fach und Fachwissen“. In: *Fachsprachen: ein internationales Handbuch zur Fachsprachenforschung und Terminologiewissenschaft*. Hrsg. von Lothar Hoffmann u. a. Berlin, New York: De Gruyter, S. 1–24.
- Kemmann, Michael (2002). „Terminologearbeit in Software-Entwicklung und -Lokalisierung“. In: *Lokalisierung von technischer Dokumentation*. Hrsg. von Jörg Hennig und Marita Tjarks-Sobhani. Lübeck: Schmidt-Römhild, S. 88–100.
- Kilgarriff, Adam und Gregory Grefenstette (2003). „Introduction to the Special Issue on the Web as Corpus“. In: *Computational Linguistics* 29.3, S. 333–347.

- Kim, Changgun (2019). *Übersetzen von Videospieltexten: Nekrotexte lesen und übersetzen*. Berlin: Frank & Timme.
- Klos, Verena (2011). *Komposition und Kompositionalität: Möglichkeiten und Grenzen der semantischen Dekodierung von Substantivkomposita*. Berlin, New York: De Gruyter.
- Klug, Nina-Maria und Hartmut Stöckl (2016). *Handbuch Sprache im multimodalen Kontext*. Berlin: De Gruyter.
- Koponen, Maarit u. a. (2012). „Post-Editing Time as a Measure of Cognitive Effort“. In: *Proceedings of the AMTA 2012 Workshop on Post-editing Technology and Practice (WPTP 2012)*.
- Krings, Hans P. (2001). *Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes*. Kent, Ohio: Kent State University Press.
- Krüger, Ralph (2013). „A Cognitive Linguistic Perspective on Explicitation and Implicitation in Scientific and Technical Translation“. In: *trans-kom* 6.2, S. 285–314.
- Kußmaul, Paul (1989). „Kontext und einsprachiges Wörterbuch in der Übersetzer Ausbildung“. In: *Translation and lexicography: papers read at the EURALEX Colloquium held at Innsbruck 2-5 July 1987*. Hrsg. von Mary Snell-Hornby, Esther Pöhl und Benjamin Bannani. Philadelphia: John Benjamins Publishing Company, S. 107–119.
- Kußmaul, Paul (1995). *Training the Translator*. Amsterdam, Philadelphia: John Benjamins Publishing Company.
- Kußmaul, Paul (2015). *Verstehen und Übersetzen: ein Lehr- und Arbeitsbuch*. 3., überarbeitete und erweiterte Auflage. Tübingen: Narr.
- Kvam, Sigmund u. a. (2024). „Kontext, Kotext, Situation ...“. In: *Translation im Kontext*. Hrsg. von Sigmund Kvam u. a. Berlin: Frank & Timme, S. 11–20.
- Laxén, Jannika und Jean-Marc Lavaur (2010). „The Role of Semantics in Translation Recognition: Effects of Number of Translations, Dominance of Translations and Semantic Relatedness of Multiple Translations“. In: *Bilingualism: Language and Cognition* 13.2, S. 157–183.
- Lemmitzer, Lothar und Heike Zinsmeister (2015). *Korpuslinguistik: eine Einführung*. 3., überarbeitete und erweiterte Auflage. Tübingen: Narr.
- Liang, Yong (1985). *Untersuchungen zur deutschen Fachsprache der elektronischen Datenverarbeitung (EDV): eine morphematisch-semantische Analyse der Substantivbenennungen*. Stuttgart: Akademischer Verlag Heinz.
- Lytje, Inger (1996). „Computer Software as Text“. In: *AI & Society* 10.2, S. 142–163.

- Macizo, Pedro und M. Teresa Bajo (2006). „Reading for Repetition and Reading for Translation: Do They Involve the Same Processes?“ In: *Cognition* 99.1, S. 1–34.
- Marais, Kobus (2019). *A (Bio)Semiotic Theory of Translation: The Emergence of Social-Cultural Reality*. New York: Routledge.
- Mejías-Climent, Laura (2021). *Enhancing Video Game Localization Through Dubbing*. Cham: Springer International Publishing.
- Melby, Alan K. und Christopher Foster (2010). „Context in Translation: Definition, Access, and Teamwork“. In: *Translation & Interpreting* 2.2 (2), S. 1–15.
- Meloni, Ilaria (2024). „Zur Rolle und Funktion des Kontextes in der Translation multimodaler Textverbunde“. In: *Translation im Kontext*. Hrsg. von Sigmund Kvam u. a. Berlin: Frank & Timme, S. 161–180.
- Merriam-Webster (2025). URL: <https://www.merriam-webster.com/> (besucht am 22. 08. 2025).
- Microsoft Terminology (2025). URL: <https://learn.microsoft.com/en-us/globalization/reference/microsoft-terminology> (besucht am 22. 08. 2025).
- Mishra, Abhijit, Pushpak Bhattacharyya und Michael Carl (2013). „Automatically Predicting Sentence Translation Difficulty“. In: *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference Volume 2*, S. 346–351.
- Moreno García, Luis Damián (2024). „Recontextualizing Disassembled Texts: Exploring the Concept of the 'Web of Texts' in Mobile Game 'Blind' Localization from Chinese into Foreign Languages“. In: *Babel. International Journal of Translation* 70.1–2, S. 64–88.
- Munday, Jeremy (2022). *Introducing Translation Studies: Theories and Applications*. 5. Auflage. Milton: Taylor & Francis Group.
- Nord, Christiane (2009). *Textanalyse und Übersetzen: theoretische Grundlagen, Methode und didaktische Anwendung einer übersetzungsrelevanten Textanalyse*. 4., überarbeitete Auflage. Tübingen: Julius Groos Verlag.
- O'Hagan, Minako und Carmen Mangiron (2013). *Game Localization: Translating for the Global Digital Entertainment Industry*. Amsterdam, Philadelphia: John Benjamins Publishing Company.
- Ogawa, Haruka, Devin Gilbert und Samar Almazroei (2021). „RedBird: Rendering Entropy Data and ST-Based Information into a Rich Discourse on Translation: Investigating Relationships Between MT Output and Human Translation“. In: *Explorations in Empirical Translation Process Research*. Hrsg. von Michael Carl. Cham: Springer International Publishing, S. 141–163.

- Ottmann, Angelika (2002). „Software-Lokalisierung“. In: *Lokalisierung von technischer Dokumentation*. Hrsg. von Jörg Hennig und Marita Tjarks-Sobhani. Lübeck: Schmidt-Römhild, S. 146–163.
- Ottmann, Angelika (2005). „Lokalisierung von Softwareoberflächen“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke und Klaus-Dirk Schmitz. Tübingen: gnv, Narr, S. 101–115.
- Pérez González, Luis (2014). „Multimodality in Translation and Interpreting Studies: Theoretical and Methodological Perspectives“. In: *A Companion to Translation Studies*. Hrsg. von Catherine Porter und Sandra Bermann. Hoboken, New Jersey: Wiley-Blackwell, S. 119–131.
- Perkuhn, Rainer, Holger Keibel und Marc Kupietz (2012). *Korpuslinguistik*. Paderborn: Fink.
- Pettini, Silvia (2021). *The Translation of Realia and Irrealia in Game Localization: Culture-Specificity between Realism and Fictionality*. Abingdon, New York: Routledge.
- Prior, Anat u. a. (2011). „Translation Ambiguity in and out of Context“. In: *Applied Psycholinguistics* 32.1, S. 93–111.
- Raasch, Albert, Hrsg. (1984). *Arbeitsbuch Linguistik: Texte, Materialien, Begriffserläuterungen*. München: Hueber.
- Reineke, Detlef (2005). „Softwarelokalisierungswerkzeuge“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke und Klaus-Dirk Schmitz. Tübingen: gnv, Narr, S. 73–87.
- Reineke, Detlef und Klaus-Dirk Schmitz, Hrsg. (2005). *Einführung in die Softwarelokalisierung*. Tübingen: gnv, Narr.
- Reiß, Katharina (1986). *Möglichkeiten und Grenzen der Übersetzungskritik: Kategorien und Kriterien für eine sachgerechte Beurteilung von Übersetzungen*. 3. Auflage. München: Hueber.
- Reiß, Katharina und Hans J. Vermeer (1984). *Grundlegung einer allgemeinen Translationstheorie*. Tübingen: M. Niemeyer.
- Roelcke, Thorsten (2020). *Fachsprachen*. 4., neu bearbeitete und wesentlich erweiterte Auflage. Berlin: Erich Schmidt Verlag.
- Roturier, Johann (2015). *Localizing Apps. Translation Practices Explained*. London, New York: Routledge.
- Rydning, Antin Fougner und Christian Lachaud (2010). „The Reformulation Challenge in Translation: Context Reduces Polysemy during Comprehension, but Multiplies Creativity during Production“. In: *Translation and Cognition*. Hrsg. von Gregory M. Shreve und Erik Angelone. Amsterdam: John Benjamins Publishing Company, S. 85–108.
- Sachse, Florian (2005). „Lokalisierungsformate“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke und Klaus-Dirk Schmitz. Tübingen: gnv, Narr, S. 145–167.

- Saldanha, Gabriela und Sharon O'Brien (2013). *Research Methodologies in Translation Studies*. Manchester: St. Jerome Publishing.
- Sandig, Barbara (2000). „Text als prototypisches Konzept“. In: *Prototypentheorie in der Linguistik: Anwendungsbeispiele - Methodenreflexion - Perspektiven*. Hrsg. von Martina Mangasser-Wahl und Ulla Bohnes. Tübingen: Stauffenburg, S. 93–112.
- Sandrini, Peter und Marta García González (2015). „Translation and Openness: An Introduction“. In: *Translation and Openness*. Hrsg. von Peter Sandrini und Marta García González. Innsbruck: Innsbruck University Press, S. 7–17.
- Sanz López, Yvonne (2013). *Videospiele übersetzen - Probleme und Optimierung*. Berlin: Frank & Timme.
- Schaeffer, Moritz, Barbara Dragsted u. a. (2016). „Word Translation Entropy: Evidence of Early Target Language Activation During Reading for Translation“. In: *New Directions in Empirical Translation Process Research*. Hrsg. von Michael Carl, Srinivas Bangalore und Moritz Schaeffer. Cham: Springer International Publishing, S. 183–210.
- Schaeffer, Moritz, David Huepe u. a. (2020). „The Translation and Interpreting Competence Questionnaire: An Online Tool for Research on Translators and Interpreters“. In: *Perspectives* 28.1, S. 90–108.
- Schaeffer, Moritz, Jean Nitzke und Silvia Hansen-Schirra (2019). „Predictive Turn in Translation Studies: Review and Prospects“. In: *Handbook of the Changing World Language Map*. Hrsg. von Stanley D. Brunn und Roland Kehrein. Cham: Springer International Publishing, S. 1–23.
- Schäler, Reinhard (2008). „Linguistic Resources and Localisation“. In: *Topics in Language Resources for Translation and Localisation*. Hrsg. von Elia Yuste Rodrigo. Amsterdam: John Benjamins Publishing Company, S. 195–214.
- Scherer, Carmen (2014). *Korpuslinguistik*. 2., aktualisierte Auflage. Heidelberg: Winter.
- Schildhauer, Erich (2005). „Lokalisierung der Dokumentation und der Online-Hilfe“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke und Klaus-Dirk Schmitz. Tübingen: gnv, Narr, S. 117–125.
- Schmitt, Peter A. (2006). *Translation und Technik*. 2. Auflage (unveränd. Nachdr. der 1. Auflage). Tübingen: Stauffenburg.
- Schmitz, Klaus-Dirk (2002). „Lokalisierung: Konzepte und Aspekte“. In: *Lokalisierung von technischer Dokumentation*. Hrsg. von Jörg Hennig und Marita Tjarks-Sobhani. Lübeck: Schmidt-Römhild, S. 11–26.
- Schmitz, Klaus-Dirk (2005a). „Internationalisierung und Lokalisierung von Software“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke und Klaus-Dirk Schmitz. Tübingen: gnv, Narr, S. 1–18.

- Schmitz, Klaus-Dirk (2005b). „Terminologieverwaltung für die Softwarelokalisierung“. In: *Einführung in die Softwarelokalisierung*. Hrsg. von Detlef Reineke, Klaus-Dirk Schmitz und Kölner Tagung Softwarelokalisierung. Tübingen: gnv, Narr, S. 39–53.
- Schmitz, Klaus-Dirk (2007). „Indeterminacy of Terms and Icons in Software Localization“. In: *Indeterminacy in Terminology and LSP: Studies in Honour of Heribert Picht*. Hrsg. von Bassegy Edem Antia. Amsterdam, Philadelphia: John Benjamins Publishing Company, S. 49–58.
- Schmitz, Klaus-Dirk (2008). „Zur Begrifflichkeit von Terminologie in Softwareoberflächen“. In: *Sprachenvielfalt im Kontext von Fachkommunikation, Übersetzung und Fremdsprachenunterricht: für Reiner Arntz zum 65. Geburtstag*. Hrsg. von Felix Mayer und Hans P. Krings. Berlin: Frank & Timme, S. 267–275.
- Schmitz, Klaus-Dirk (2021). *Terminologearbeit in der Softwarelokalisierung*. URL: <https://docplayer.org/22810784-Prof-dr-klaus-dirk-schmitz-fh-koeln-fakultaet-03-iim-tekong-rg-nrw-fh-koeln-15-5-2008-lokalisierung-internationalisierung.html> (besucht am 21.04.2021).
- Schmitz, Klaus-Dirk und Kirsten Wahle, Hrsg. (2000). *Softwarelokalisierung*. Tübingen: Stauffenburg.
- Schmitz, Ulrich (2016). „Multimodale Texttypologie“. In: *Handbuch Sprache im multimodalen Kontext*. Hrsg. von Nina-Maria Klug und Hartmut Stöckl. Berlin: De Gruyter, S. 327–347.
- Schneider, Hans-Jochen, Hrsg. (1998). *Lexikon Informatik und Datenverarbeitung*. 4., aktualisierte und erweiterte Auflage. München: Oldenbourg.
- Seewald-Heeg, Uta (2016). „Terminologearbeit und Softwarelokalisierung“. In: *Terminologearbeit für Technische Dokumentation*. Hrsg. von Jörg Hennig und Marita Tjarks-Sobhani. 2., grundlegend überarbeitete Auflage. Stuttgart: tcworld, S. 154–162.
- Seewald-Heeg, Uta (2020). „Was mit Sprachen UND Technik“. In: *MDÜ* 5/2020, S. 10–16.
- Software* (2025). Wikipedia. URL: <https://de.wikipedia.org/wiki/Software> (besucht am 22.08.2025).
- Stallman, Richard M. (2002). *Free Software, Free Society*. Hrsg. von Joshua Gay. Boston (Mass.): Free Software Foundation.
- Steinseifer, Martin (2011). „Die Typologisierung multimodaler Kommunikationsangebote – Am Beispiel der visuellen Aspekte seitenbasierter Dokumente“. In: *Textsorten, Handlungsmuster, Oberflächen: linguistische Typologien der Kommunikation*. Hrsg. von Stephan Habscheid. Berlin: De Gruyter, S. 164–189.

- Stolze, Radegundis (1985). *Grundlagen der Textübersetzung*. 2. Auflage. Heidelberg: J. Groos.
- Stolze, Radegundis (2013). *Fachübersetzen - Ein Lehrbuch für Theorie und Praxis*. 45., unveränderte Auflage. Berlin: Frank & Timme.
- Tokowicz, Natasha und Judith F. Kroll (2007). „Number of Meanings and Concreteness: Consequences of Ambiguity within and across Languages“. In: *Language and Cognitive Processes* 22.5, S. 727–779.
- Vermeer, Hans J. (1972). *Allgemeine Sprachwissenschaft: eine Einführung*. Freiburg: Rombach.
- Vieira, Lucas Nunes (Juni 2016). „How Do Measures of Cognitive Effort Relate to Each Other? A Multivariate Analysis of Post-Editing Process Data“. In: *Machine Translation* 30.1–2, S. 41–62.
- Wahle, Kirsten (2000a). „Richtlinien zum Übersetzen von EDV-Texten“. In: *Softwarelokalisierung*. Hrsg. von Klaus-Dirk Schmitz und Kirsten Wahle. Tübingen: Stauffenburg, S. 101–134.
- Wahle, Kirsten (2000b). „Wie wird Software lokalisiert?“. In: *Softwarelokalisierung*. Hrsg. von Klaus-Dirk Schmitz und Kirsten Wahle. Tübingen: Stauffenburg, S. 31–47.
- Wei, Yuxiang (2021). „Entropy and Eye Movement: A Micro-analysis of Information Processing in Activity Units During the Translation Process“. In: *Explorations in Empirical Translation Process Research*. Hrsg. von Michael Carl. Cham: Springer International Publishing, S. 165–202.
- Wei, Yuxiang (2022). „Entropy as a Measurement of Cognitive Load in Translation“. In: *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Workshop 1: Empirical Translation Process Research)*. Association for Machine Translation in the Americas, S. 75–86.
- Wichter, Sigurd (1997). „Technische Fachsprachen im Bereich der Informatik“. In: *Fachsprachen: ein internationales Handbuch zur Fachsprachenforschung und Terminologiewissenschaft*. Hrsg. von Lothar Hoffmann u. a. Berlin, New York: De Gruyter, S. 1173–1182.
- Zerfaß, Angelika (2002). „Wie die Dokumentationserstellung die Übersetzbarkeit beeinflusst“. In: *Lokalisierung von technischer Dokumentation*. Hrsg. von Jörg Hennig und Marita Tjarks-Sobhani. Lübeck: Schmidt-Römhild, S. 208–223.
- Zittrain, Jonathan (2004). „Normative Principles for Evaluating Free and Proprietary Software“. In: *The University of Chicago Law Review* Vol. 71 (No. 1), S. 265–287.

Anhang A

Anhang zur Korpuserstellung

Dieser Anhang enthält die Details zu Vorgehensweise und Problemen bei der Aufnahme der Textproben in das Korpus, wie Versionen und URLs der verwendeten Softwareprodukte. Die Korpuserstellung erfolgte 2019, die Angaben zum Download der einzelnen Softwareprodukte wurden daher zu diesem Zeitpunkt protokolliert und sind unter Umständen nicht mehr aktuell (siehe dazu auch die Daten des letzten Zugriffs, die teils im Jahr 2019 liegen).

A.1 Audacity

Audacity ist ein Softwareprodukt zum Aufnehmen und Bearbeiten von Audiodateien.¹ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	Audacity Version 2.2.2
Dateiname	audacity-git-repository.zip
Downloaddatum	22.03.2019
Link	https://www.audacity.de/downloads/ ; letzter Zugriff: 22.03.2019
Bereich	Link Download Audacity Git Repository unter der Überschrift Quellcode

In dieser Zip-Datei enthält das Verzeichnis `audacity-master/locale` die PO-Dateien mit den Lokalisierungen. Sämtliche englischen und deutschen Strings sind dabei in der Datei `de.po` zu finden. In dieser Datei wurden die ersten Strings bis Zeile 1.108 (msgid "Edit Chains") einschließlich verwendet, was insgesamt ein Volumen von 1.001 englischen Wörtern ergibt.

¹<https://de.wikipedia.org/wiki/Audacity>; letzter Zugriff: 05.07.2025

Auszug aus der Datei de.po:²

```
#: src/BatchProcessDialog.cpp:109
msgid "Apply to Current &Project"
msgstr "Auf aktuelles &Projekt anwenden"

#: src/BatchProcessDialog.cpp:110
msgid "Apply to &Files..."
msgstr "&Auf Dateien anwenden ..."

#: src/BatchProcessDialog.cpp:111 src/BatchProcess [...]
msgid "&Cancel"
msgstr "&Abbrechen"
```

In das Korpus wurden die Strings aus den Zeilen mit der Zeichenfolge `msgid` am Anfang aufgenommen. Dazu wurde die PO-Datei zunächst in ApSIC Xbench³ geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die Strings in die Datei `audiobearbeitung.txt` eingefügt, die in das Korpus aufgenommen wurde.

Folgende Strings wurden nicht in das Korpus aufgenommen, weil dafür kein Zieltext vorhanden war:

```
msgid "Cle&ar\tCtrl+L"
msgid "&Matching Paren\tF8"
msgid "&Top S-expr\tF9"
msgid "&Higher S-expr\tF10"
msgid "&Previous S-expr\tF11"
msgid "&Next S-expr\tF12"
msgid "&Go\tF5"
msgid "Harvey Lubin (logo)"
msgid "Clear"
msgid "Clear selection"
msgid "Go to matching paren"
msgid "Top"
msgid "Go to top S-expr"
msgid "Go to higher S-expr"
```

²In diesem und einigen der folgenden Auszüge aus Dateien wurden zu lange Zeilen gekürzt, was mit der Angabe [...] deutlich gemacht wird.

³<https://www.xbench.net/>; letzter Zugriff: 05.07.2025

```
msgid "Go to previous S-expr"
msgid "Go to next S-expr"
msgid "Panel 1"
msgid "Cross-platform GUI library"
msgid "Commit Id:"
```

A.2 Avidemux

Avidemux ist ein Softwareprodukt zum Bearbeiten von Videodateien.⁴ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	Avidemux Version 2.7.3
Dateiname	avidemux_2.7.3.tar.gz
Downloaddatum	15.04.2019
Link	https://www.foosshub.com/Avidemux.html ; letzter Zugriff: 05.07.2025
Bereich	Link DOWNLOAD Avidemux Source File - Linux

In dieser GZ-Datei (Linux-Kompressionsformat) ist das Verzeichnis `avidemux_2.7.3\avidemux\qt4\i18n` mit TS-Dateien für verschiedene Lokalisierungen der GUI von Avidemux enthalten. Von diesen TS-Dateien wurde die Datei `avidemux_de.ts` für die vorliegende Untersuchung verwendet, die einen Teil der Strings von Avidemux enthält. In dieser Datei wurden die ersten Strings bis Zeile 2.254 (`<source>Error while cutting out.</source>`) einschließlich verwendet, was insgesamt ein Volumen von 1.002 englischen Wörtern ergibt.

Auszug aus der Datei `avidemux_de.ts`:

```
<message>
<location line="-36"/>
<source>Cannot find a demuxer for %s</source>
<translation>Kann Demuxer für %s nicht finden</translation>
</message>
<message>
<location line="+64"/>
<source>width</source>
<translation>Breite</translation>
</message>
```

⁴<https://de.wikipedia.org/wiki/Avidemux>; letzter Zugriff: 05.07.2025

```
<message>
<location line="+6"/>
<source>%sheight</source>
<translation>%sHöhe</translation>
</message>
```

In das Korpus wurden die Strings aus dem XML-Tagpaar `<source></source>` in dieser TS-Datei aufgenommen. Dazu wurde die TS-Datei zunächst in ApSIC Xbench geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die Strings in die Datei `videobearbeitung.txt` eingefügt, die in das Korpus aufgenommen wurde.

Folgende Strings wurden nicht in das Korpus aufgenommen, weil dafür kein Zieltext vorhanden war:

```
<source>Some timing information are incorrect.
It happens with some capture software.
If you re encode video we should drop these informations,
else it will cause dropped frame/jerky video.
If you just copy the video without reencoding,
you should keep them.
Drop timing informations ?</source>
<source>Time Left : Infinity</source>
<source>\# Images :</source>
<source>ProcessImages</source>
<source>Speed</source>
<source>0 fps</source>
<source>\_Profile::</source>
<source>\_Bitrate:</source>
<source>SBR enabled</source>
<source>After burner</source>
<source>FDK-AAC Configuration</source>
<source>Alt+Return</source>
<source>Codec 0\&ptions</source>
<source>XXXX</source>
<source>\&Navigation</source>
<source>Play</source>
<source>Forward one minute</source>
<source>Se\&lection</source>
<source>Selection: 000000</source>
```

<source>Vol\&ume</source>
<source>Audio \&metre</source>
<source>Ctrl+Alt+C</source>
<source>Ctrl+Alt+N</source>
<source>Ctrl+Alt+S</source>
<source>Ctrl+Alt+V</source>
<source>Ctrl+Alt+M</source>
<source>_Bitrate:</source>
<source>Shell</source>
<source>Clear</source>
<source>Evaluate</source>
<source>Ctrl+Return</source>
<source>VapourSynth Proxy</source>
<source>Select VS file</source>
<source>Port to use: </source>
<source>Run!</source>
<source>© 2001 - 2019 Mean / Eumagga0x2a</source>
<source>Avidemux </source>
<source>DXVA2 (best)</source>
<source>Zoom 1:4</source>
<source>Zoom 1:2</source>
<source>Zoom 1:1</source>
<source>Zoom 2:1</source>
<source>Fontconfig</source>
<source>Gettext</source>
<source>x86</source>
<source>x86-64</source>
<source>De_interlacing (ffmpegdeint)</source>
<source>It is impossible to cut out the entire video. [...]
<source>Deleting</source>
<source>It is impossible to delete the entire video. [...]
<source>The end point of the cut is not on a keyframe.
Video saved in copy mode will be corrupted at this point.
Proceed anyway?</source>
<source>The end point of the deletion is not on a keyframe.
Video saved in copy mode will be corrupted at this point.
Proceed anyway?</source>
<source>The tinypy plugin is missing.
Expect problems.</source>
<source>The saved video is incomplete. The error [...]
<source>The cut points of the pasted video are not on [...]

Video saved in copy mode will be corrupted at these points.
Proceed anyway?

A.3 BlueGriffon

BlueGriffon war ein WYSIWYG-HTML-Editor, dessen Entwicklung 2024 eingestellt wurde.⁵ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	BlueGriffon Version 3.0.1
Dateiname	bluegriffon-3.0.1.win-x86_64.zip
Downloaddatum	10.05.2019
Link	http://www.bluegriffon.org/#download ; letzter Zugriff: 10.05.2019
Bereich	Link (zip file) unter „Windows 7/8/10“

In dieser ZIP-Datei ist das Verzeichnis BlueGriffon\chrome mit JAR-Dateien enthalten. Die JAR-Datei BlueGriffon-en-US.jar wiederum enthält das Verzeichnis locale\en-US\bluegriffon mit den Dateien bluegriffon.dtd und bluegriffon.properties sowie das Verzeichnis locale\en-US\cssproperties mit der Datei cssproperties.dtd, die für die vorliegende Untersuchung verwendet wurden.

Die deutschen DTD- und Properties-Dateien fanden sich unter folgenden Links:

- <http://sources.disruptive-innovations.com/bluegriffon-110n/trunk/de/bluegriffon/base/locale/bluegriffon/>
- <http://sources.disruptive-innovations.com/bluegriffon-110n/trunk/de/bluegriffon/sidebars/cssproperties/>⁶

Informationen dazu: <http://bluegriffon.org/freshmeat/110n.pdf>; letzter Zugriff: 10.05.2019

Aus diesen Dateien wurden in Memsource⁷ via Alignment TMs in Form von XLSX-Dateien erstellt, in denen das Alignment dann manuell korrigiert wurde. Anschließend wurden aus diesen TMs die Access-Keys, die in den

⁵<https://de.wikipedia.org/wiki/BlueGriffon>; letzter Zugriff: 05.07.2025

⁶; Letzter Zugriff auf alle diese URLs: 10.05.2019

⁷Jetzt Phrase TMS (vgl. <https://en.wikipedia.org/wiki/Memsource>; letzter Zugriff: 05.07.2025)

DTD-Dateien als einzelne Buchstaben enthalten waren (s. Auszug aus der Datei unten) gelöscht, so dass nur die vollständigen Strings als Segmentpaare vorhanden waren. Die Strings der beiden Dateien `bluegriffon.dtd` und `bluegriffon.properties` wurden vollständig in das Korpus aufgenommen, die Strings der Datei `cssproperties.dtd` wurden bis String `<!ENTITY getNaturalSize.label "Get natural size">` in Zeile 159 einschließlich verwendet, was insgesamt ein Volumen von 1.001 englischen Wörtern ergibt. Aus den XLSX-Dateien wurden dann die Strings in die Datei `html-editor.txt` eingefügt, die in das Korpus aufgenommen wurde.

Auszug aus der Datei `bluegriffon.dtd`:

```
<!ENTITY fileMenu.label      "File">
<!ENTITY fileMenu.accesskey  "f">
<!ENTITY newMenu.label       "New">
<!ENTITY newMenu.accesskey   "n">

<!ENTITY newWizardMenu.label "New wizard">

<!ENTITY editMenu.label      "Edit">
<!ENTITY editMenu.accesskey  "e">

<!ENTITY viewMenu.label      "View">
<!ENTITY viewMenu.accesskey  "v">
<!ENTITY customizeToolbar.label "Customize">
<!ENTITY customizeToolbar.accesskey "c">

<!ENTITY insertMenu.label    "Insert">
<!ENTITY insertMenu.accesskey "i">
```

Auszug aus der Datei `bluegriffon.properties`:

```
DocumentTitle=Page Title
NeedDocTitle=Please enter a title for the current page.
DocTitleHelp=This identifies the page in the window [...]
ExportToText=Export to Text
SaveDocumentAs=Save Page As
XHTMLfiles=XHTML Files
untitled=untitled
SaveDocument=Save Page
SaveFileFailed=Saving file failed!
```

Folgendes war bei der Bearbeitung dieser Dateien auffällig:

- Die Access-Keys sind in den DTD-Dateien nicht in den eigentlichen Strings kodiert, sondern in eigens für die Access-Keys angelegten zusätzlichen Strings (s. Ausschnitt). Aus diesem Grund konnten die Access-Keys für diese Dateien nicht in das Korpus übernommen werden.
- Bei der automatischen Alignierung wurden einzelne Strings, in denen ein Satzendezeichen oder ein Ausrufezeichen vorkam, in zwei Segmente und damit in der XSLX-Datei in zwei Zeilen aufgeteilt. Solche Strings wurden bei der Überprüfung manuell in eine Zeile kopiert, damit sie im Korpus auch als ein String zu identifizieren waren.

A.4 Claws Mail

Claws Mail ist ein E-Mail-Programm.⁸ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	Claws Mail Version 3.17.3
Dateiname	<code>claws-mail-3.17.3.tar.gz</code>
Downloaddatum	09.05.2019
Link	https://www.claws-mail.org/releases.php ; letzter Zugriff: 05.07.2025
Bereich	Link claws-mail-3.17.3.tar.gz unter „Latest Claws Mail Releases -3.17.3 (22-December-2018) - Download“

In dieser GZ-Datei (Linux-Kompressionsformat) ist das Verzeichnis `claws-mail-3.17.3\po` und darin PO-Dateien für die verschiedenen Sprachen enthalten. Von diesen PO-Dateien wurde die Datei `de.po` für die vorliegende Untersuchung verwendet. In dieser Datei wurden die ersten Strings bis Zeile 1.130 (`msgid "%s:%d: unknown host.\n"`) einschließlich verwendet, was insgesamt ein Volumen von 1.001 englischen Wörtern ergibt.

Auszug aus der Datei `de.po`:

```
#: src/account.c:823
msgid "_Set as default account "
msgstr "_Als Standardkonto setzen"
```

```
#: src/account.c:915
msgid "Accounts with remote folders cannot be copied."
```

⁸https://de.wikipedia.org/wiki/Claws_Mail; letzter Zugriff: 05.07.2025

```

msgstr "Konten mit fernegelegenen Ordnern sind nicht [...]"

#: src/account.c:922
#, c-format
msgid "Copy of %s"
msgstr "Kopie von %s"

#: src/account.c:1119
#, c-format
msgid "Do you really want to delete the account '%s'?"
msgstr "Möchten Sie das Konto '%s' wirklich löschen?"

```

In das Korpus wurden die Strings aus den Zeilen mit der Zeichenfolge `msgid` am Anfang aufgenommen. Dazu wurde die PO-Datei zunächst in ApSIC Xbench geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die Strings in die Datei `email.txt` eingefügt, die in das Korpus aufgenommen wurde.

A.5 Double Commander

Double Commander ist ein Dateimanager.⁹ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	Double Commander Version 0.9.2
Dateiname	<code>doublecmd-0.9.2-src.tar.gz</code>
Downloaddatum	09.05.2019
Link	https://sourceforge.net/p/doublecmd/wiki/Download/ ; letzter Zugriff: 05.07.2025
Bereich	Link doublecmd-0.9.2-src.tar.gz unter „Download - Source - Double Commander“

In dieser GZ-Datei (Linux-Kompressionsformat) ist das Verzeichnis `doublecmd-0.9.2\language` mit PO-Dateien für die verschiedenen Sprachen enthalten. Von diesen PO-Dateien wurde die Datei `doublecmd.de.po` für die vorliegende Untersuchung verwendet. In dieser Datei wurden die ersten Strings bis Zeile 2.332 (`msgid "Sort left panel by &Date"`) einschließlich verwendet, was insgesamt ein Volumen von 1.004 englischen Wörtern ergibt.

Auszug aus der Datei `doublecmd.de.po`:

⁹https://en.wikipedia.org/wiki/Double_Commander; letzter Zugriff: 05.07.2025

```

#: fsyncdirsdlg.rscomparingpercent
msgid "Comparing... %d%% (ESC to cancel)"
msgstr "Vergleiche... %d%% (ESC zum Abbrechen)"

#: fsyncdirsdlg.rsdeleteright
msgid "Right: Delete %d file(s)"
msgstr "Lösche Datei(en) %d"

#: fsyncdirsdlg.rsfilesfound
msgid "Files found: %d (Identical: %d, Different: %d, [...])"
msgstr "Dateien gefunden: %d (Identisch: %d, [...])

#: fsyncdirsdlg.rslefttorightcopy
msgid "Left to Right: Copy %d files, total size: %d bytes"
msgstr "Links nach rechts: Kopiere %d Dateien, [...]"

```

In das Korpus wurden die Strings aus den Zeilen mit der Zeichenfolge `msgid` am Anfang aufgenommen. Dazu wurde die PO-Datei zunächst in ApSIC Xbench geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die Strings in die Datei `dateimanager.txt` eingefügt, die in das Korpus aufgenommen wurde.

Folgende Strings wurden nicht in das Korpus aufgenommen, weil dafür kein Zielttext vorhanden war:

```

msgid "&Verify"
msgid "Benchmark"
msgid "Benchmark data size: %d MB"
msgid "Hash"
msgid "Time (ms)"
msgid "Speed (MB/s)"
msgid "Select shortcut from list of remaining free [...]"
msgid "Reload"
msgid "&Cancel"
msgid "&OK"
msgid "Name"
msgid "Plugins"
msgid "Close"
msgid "Terminate"
msgid "Unlock"
msgid "Unlock All"

```

```

msgid "File Handle"
msgid "Process ID"
msgid "Executable Path"
msgid "Configuration of hot keys"
msgid "Cancel search, close and free from memory"
msgid "For all other "Find files", cancel, close and [...]"
msgid "New search (clear filters)"
msgid "Switch to Next Page"
msgid "Switch to &Previous Page"
msgid "Hexadecimal"
msgid "&Action"
msgid "For all others, cancel, close and free from memory"
msgid "&View"
msgid "Activate Tab By Index"
msgid "&Benchmark"
msgid "Configuration of Archivers"
msgid "Configuration of Plugins"
msgid "Save Position"
msgid "Configuration of searches"
msgid "Configuration of tooltips"
msgid "For all searches, cancel, close and free memory"
msgid "Swap focus"
msgid "Switch between left and right file list"
msgid "Focus on tree view"
msgid "Switch between current file list and tree [...]"

```

A.6 GIMP

GIMP ist ein Programm zum Bearbeiten von Grafikdateien.¹⁰ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	GIMP Version 2.10.8
Dateiname	<code>gimp-2.10.8.tar.bz2</code>
Downloaddatum	25.03.2019
Link	https://download.gimp.org/mirror/pub/gimp/v2.10/ ; letzter Zugriff: 05.07.2025
Bereich	Link <code>gimp-2.10.8.tar.bz2</code>

In dieser BZ2-Datei (Kompressionsformat) enthält das Verzeichnis `gimp-2.10.8\po` die PO-Dateien mit den Lokalisierungen. Sämtliche eng-

¹⁰<https://de.wikipedia.org/wiki/GIMP>; letzter Zugriff: 05.07.2025

lischen und deutschen Strings sind dabei in der Datei `de.po` zu finden. In dieser Datei wurden die ersten Strings bis Zeile 605 (`msgid "Open images as new"`) einschließlich verwendet, was insgesamt ein Volumen von 1.001 englischen Wörtern ergibt. Im Verzeichnis `gimp-2.10.8` sind noch andere Verzeichnisse mit PO-Dateien enthalten, wie `gimp-2.10.8\po-libgimp` und `gimp-2.10.8\po-plugin-ins`, in denen weitere Dateien mit dem Namen `de.po` enthalten sind, die aber Übersetzungen für Teilkomponenten von GIMP enthalten.

Auszug aus der Datei `gimp-2.10.8\po\de.po`:

```
#: ../app/main.c:162
msgid "Show license information and exit"
msgstr "Lizenzinformationen anzeigen und Programm beenden"

#: ../app/main.c:167
msgid "Be more verbose"
msgstr "Ausführlicher verhalten"

#: ../app/main.c:172
msgid "Start a new GIMP instance"
msgstr "Eine neue GIMP-Instanz starten"
```

In das Korpus wurden die Strings aus den Zeilen mit der Zeichenfolge `msgid` am Anfang aufgenommen. Dazu wurde die PO-Datei zunächst in `ApSIC Xbench` geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die Strings in die Datei `bildbearbeitung.txt` eingefügt, die in das Korpus aufgenommen wurde.

Folgende Strings wurden nicht in das Korpus aufgenommen, weil dafür kein Zieltext vorhanden war:

```
#: ../desktop/org.gimp.GIMP.appdata.xml.in.in.h:24
msgid ""
"Fonts Tagging with the same user interface as for [...]
"gradients"

#: ../desktop/org.gimp.GIMP.appdata.xml.in.in.h:48
msgid ""
"Plug-in debugging improved to output stack traces from [...]
"trace-mode command line option not only on receiving [...]
"warnings and critical errors when \"fatal-warnings\" [...]"
```

A.7 LibreOffice

LibreOffice ist ein Office-Paket mit Funktionen für Tabellen, Textverarbeitung, Grafikbearbeitung, Präsentationen, mathematischen Formeln und einer Datenbank.¹¹ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	LibreOffice Version 6.2.2.2
Dateiname	libreoffice-translations-6.2.2.2.tar.xz
Downloaddatum	05.04.2019
Link	https://de.libreoffice.org/download/download/ ; letzter Zugriff: 05.07.2025
Bereich	Link libreoffice-translations-6.2.2.2.tar.xz unter „SDK und Quellcode“ → „Quellcode herunterladen“

In dieser XZ-Datei (Linux-Kompressionsformat) ist die TAR-Datei `libreoffice-translations-6.2.2.2.tar` enthalten, die wiederum das Verzeichnis `libreoffice-6.2.2.2\translations\source\de` und darin zahlreiche Unterverzeichnisse mit PO-Dateien für die Programmkomponenten und die Hilfe von LibreOffice enthält. Von diesen PO-Dateien wurde die Datei `UI.po` im Verzeichnis `officecfg\registry\data\org\openoffice\Office\UI.po` für die vorliegende Untersuchung verwendet, in der die Strings der LibreOffice-Komponenten bereitgestellt werden. In dieser Datei wurden die ersten Strings bis Zeile 4.246 (`msgid "Detective"`) einschließlich verwendet, was insgesamt ein Volumen von 1.000 englischen Wörtern ergibt.

Auszug aus der Datei `UI.po`:

```
#: CalcCommands.xcu
msgctxt ""
"CalcCommands.xcu\n"
"..CalcCommands.UserInterface.Commands..uno:EditPrintArea\n"
"ContextLabel\n"
"value.text"
msgid "~Edit..."
msgstr "~Bearbeiten..."
```

```
#: CalcCommands.xcu
msgctxt ""
"CalcCommands.xcu\n"
```

¹¹<https://de.wikipedia.org/wiki/LibreOffice>; letzter Zugriff: 05.07.2025

```

"..CalcCommands.UserInterface.Commands..uno:EditPrintArea\n"
"TooltipLabel\n"
"value.text"
msgid "Edit Print Ranges"
msgstr "Druckbereiche bearbeiten"

```

In das Korpus wurden die Strings aus den Zeilen mit der Zeichenfolge `msgid` am Anfang aufgenommen. Dazu wurde die PO-Datei zunächst in ApSIC Xbench geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die benötigten Strings in die Datei `office-anwendung.txt` eingefügt, die in das Korpus aufgenommen wurde.

An dieser Datei war auffällig, dass bei einigen Strings (s. Ausschnitt oben) das Tildezeichen (`~`) als Access-Key verwendet wird.

A.8 Mozilla Firefox

Mozilla Firefox ist ein Webbrowser.¹² Die folgende Tabelle enthält Informationen zu Version und Download:

Version	Mozilla Firefox Version 66.0.3		
Dateiname	de-8c11cac98cfb.tar.gz für die ZT-Dateien und mozilla-central-6c9e7cba261d.tar.gz für die AT-Dateien		
Downloaddatum	15.04.2019		
Link	https://hg.mozilla.org/ ;	letzter	Zugriff:
		15.04.2019	
Bereich	Link gz für mozilla-central sowie Link l10-central und dort Link gz für de		

In der GZ-Datei (Linux-Kompressionsformat) mit den deutschen Übersetzungen ist das Verzeichnis `de-8c11cac98cfb\browser\chrome\browser` und darin die Datei `browser.dtd` enthalten. Die entsprechende AT-Datei `browser.dtd` findet sich im Verzeichnis `mozilla-central-6c9e7cba261d\browser\locales\en_US\chrome\browser` der AT-GZ-Datei `mozilla-central-6c9e7cba261d.tar.gz`. Aus diesen beiden Dateien wurde in Memsources via Alignment ein TM in Form einer XLSX-Datei erstellt, in der das Alignment dann manuell korrigiert

¹²https://de.wikipedia.org/wiki/Mozilla_Firefox; letzter Zugriff: 05.07.2025

wurde. Anschließend wurden aus diesem TM die Access-Keys, die als einzelne Buchstaben enthalten waren (s. Auszug aus der Datei unten) gelöscht, so dass nur die vollständigen Strings als Segmentpaare vorhanden waren. Die Strings der Datei `browser.dtd` wurden bis String `<!ENTITY identity.activeLoaded "You have disabled protection on this page.">` in Zeile 815 einschließlich verwendet, was insgesamt ein Volumen von 1.004 englischen Wörtern ergibt. Aus der XLSX-Datei wurden dann die Strings in die Datei `browser.txt` eingefügt, die in das Korpus aufgenommen wurde.

Auszug aus der Datei `browser.dtd`:

```
<!ENTITY unpinTab.label "Unpin Tab">
<!-- LOCALIZATION NOTE(pinTab.accesskey, pinSelectedTabs [...]
unpinTab.accesskey, unpinSelectedTabs.accesskey): These [...]
same accesskey but will never be visible at the same time. -->
<!ENTITY unpinTab.accesskey "p">
<!ENTITY sendPageToDevice.label "Send Page to Device">
<!ENTITY sendPageToDevice.accesskey "n">
<!ENTITY sendLinkToDevice.label "Send Link to Device">
<!ENTITY sendLinkToDevice.accesskey "n">
<!ENTITY undoCloseTab.label "Undo Close Tab">
<!ENTITY undoCloseTab.accesskey "U">
<!ENTITY hiddenTabs.label "Hidden Tabs">
```

Folgendes war an dieser Datei auffällig:

- Die Access-Keys sind in dieser DTD-Datei nicht in den eigentlichen Strings kodiert, sondern in eigens für die Access-Keys angelegten zusätzlichen Strings (s. Ausschnitt). Aus diesem Grund konnten die Access-Keys für diese Datei nicht in das Korpus übernommen werden.
- Bei der automatischen Alignierung wurden einzelne Strings, in denen ein Satzendezeichen oder ein Ausrufezeichen vorkam, in zwei Segmente und damit in der XSLX-Datei in zwei Zeilen aufgeteilt. Solche Strings wurden bei der Überprüfung manuell in eine Zeile kopiert, damit sie im Korpus auch als ein String zu identifizieren waren.

A.9 Notepad++

Notepad++ ist ein Texteditor.¹³ Die folgende Tabelle enthält Informationen zu Version und Download:

¹³<https://de.wikipedia.org/wiki/Notepad%2B%2B>; letzter Zugriff: 05.07.2025

Version Notepad++ Version 7.6.4 (64 Bit)
Dateiname npp.7.6.4.bin.x64.zip
Downloaddatum 18.03.2019
Link <https://notepad-plus-plus.org/download/v7.6.4.html>; letzter Zugriff: 18.03.2019
Bereich Link **Notepad++ zip package 64-bit x64** unter der Überschrift **Download 64-bit x64**

In dieser Zip-Datei enthält das Verzeichnis `localization` die XML-Dateien mit den Übersetzungen der Strings. Sämtliche englischen Strings sind dabei in der Datei `english.xml` zu finden, die deutschen in der Datei `german.xml`. In diesen Dateien wurden die ersten Strings bis Zeile 421 (`<Item id="1932" name="Treat each line as a separate string"/>`) einschließlich verwendet, was insgesamt ein Volumen von 1.005 englischen Wörtern ergibt.

Auszug aus der Datei `english.xml`:

```
<Main>
<!-- Main Menu Entries -->
<Entries>
<Item menuId="file" name="\&File"/>
<Item menuId="edit" name="\&Edit"/>
<Item menuId="search" name="\&Search"/>
<Item menuId="view" name="\&View"/>
```

Für das Korpus wurden die Strings aus dem XML-Code extrahiert, also hier die für das Attribut `name` angegebenen Strings wie `&File` oder `&Edit`. Im weiteren Verlauf war der Text auch noch in anderen Attributen wie `title` enthalten. Dazu wurden Die XML-Dateien zunächst in einem Editor geöffnet, dann wurden die XML-Elemente wenn möglich über die Funktion „Suchen und Ersetzen“ global und ansonsten einzeln manuell gelöscht. Zuletzt wurden die Strings in die Datei `editor.txt` eingefügt, die in das Korpus aufgenommen wurde.

Folgendes war an dieser Datei auffällig:

- In dieser Datei wird `&`, der HTML-Code für das Zeichen `&`, als Access-Key genutzt.
- Für den String `<Item id="48014" name="Open Plugins Folder..."/>` in Zeile 298 der AT-Datei gibt es im Zieltext keine Entsprechung, daher wurde er nicht in das Korpus aufgenommen.

- Im String `<Find title="" titleFind=[...]` in Zeile 348 der AT-Datei wurde für das Attribut `title` kein Wert angegeben, im ZT allerdings schon (**Suchen und ersetzen**), daher wurde dieses Attribut nicht in das Korpus aufgenommen.

A.10 VLC Media Player

VLC Media Player ist ein Medienplayer.¹⁴ Die folgende Tabelle enthält Informationen zu Version und Download:

Version	VLC Media Player Version 3.0.6
Dateiname	<code>vlc-3.0.6.tar.xz</code>
Downloaddatum	07.05.2019
Link	https://www.videolan.org/vlc/download-sources.html ; letzter Zugriff: 05.07.2025
Bereich	Link VLC source code tar.xz unter „Tarballs - Latest VLC source code tarball (3.0.6):“

In dieser XZ-Datei (Linux-Kompressionsformat) ist die TAR-Datei `vlc-3.0.6.tar` enthalten, die wiederum das Verzeichnis `vlc-3.0.6.\po` und darin PO-Dateien für die verschiedenen Sprachen enthält. Von diesen PO-Dateien wurde die Datei `de.po` für die vorliegende Untersuchung verwendet. In dieser Datei wurden die ersten Strings bis Zeile 882 ("`modules.`") einschließlich verwendet, was insgesamt ein Volumen von 1.009 englischen Wörtern ergibt.

Auszug aus der Datei `de.po`:

```
#: include/vlc_config_cat.h:43
msgid "Main interfaces"
msgstr "Hauptinterfaces"

#: include/vlc_config_cat.h:44
msgid "Settings for the main interface"
msgstr "Einstellungen für das VLC-Hauptinterface"

#: include/vlc_config_cat.h:46 src/libvlc-module.c:81
msgid "Control interfaces"
msgstr "Steuerinterfaces"
```

¹⁴https://de.wikipedia.org/wiki/VLC_media_player; letzter Zugriff: 05.07.2025

```
#: include/vlc_config_cat.h:47
msgid "Settings for VLC's control interfaces"
msgstr "Einstellungen für die VLC-Steuerinterfaces"
```

In das Korpus wurden die Strings aus den Zeilen mit der Zeichenfolge `msgid` am Anfang aufgenommen. Dazu wurde die PO-Datei zunächst in ApSIC Xbench geöffnet und via **Tools** → **Export Items** in eine XLSX-Datei exportiert, wobei das Kontrollkästchen **Remove repetitions in exported file** abgewählt wurde. Aus dieser XLSX-Datei wurden dann die Strings in die Datei `medienplayer.txt` eingefügt, die in das Korpus aufgenommen wurde.

Die folgenden beiden Strings sind im Korpus unvollständig enthalten, weil die automatische Erstellung nicht fehlerfrei funktioniert hat:

```
#: include/vlc_config_cat.h:129
msgid ""
"Stream output settings are used when acting as a [...]
"saving incoming streams.\n"
"Streams are first muxed and then sent through an [...]
"that can either save the stream to a file, or stream [...]
"RTSP).\n"
"Sou[t streams modules allow advanced stream [...]
"duplicating...).]"
```

```
#: include/vlc_config_cat.h:141
msgid ""
"Muxers create the encapsulation formats that are [...]
"elementary streams (video, audio, ...) together. [...]
"always force a specific muxer. You should probably [...]
"You can also set default parameters for each [muxer.]"
```

Die Satzteile in eckigen Klammern in den letzten Zeilen dieser Strings sind in der Datei `de.po` vorhanden, nicht aber im Korpus.

Anhang B

Anhang zum Übersetzungstest

Dieser Anhang enthält Details zu verschiedenen Aspekten des Übersetzungstests.

B.1 Anweisungsdatei für den Übersetzungstest

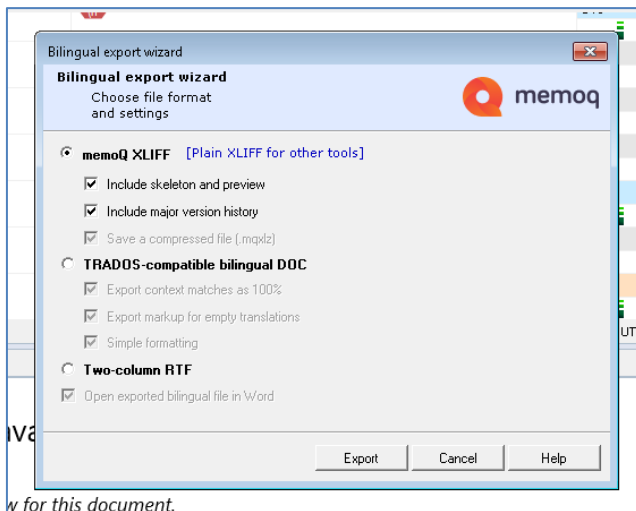
Nachfolgend findet sich die Anweisungsdatei für den Übersetzungstest.

Probandentest Promotion Marcus Wiedmann - Übersetzungsanweisungen

Hintergrund: Dieser Test wird im Rahmen meines Promotionsvorhabens durchgeführt und soll dazu dienen, spezielle Probleme bei der Übersetzung der Benutzeroberflächen von Computerprogrammen herauszuarbeiten, um so das für eine gute Übersetzung nötige Referenzmaterial besser identifizieren zu können. Daher würde ich mich freuen, wenn Sie ihn möglichst sorgfältig bearbeiten. Vielen Dank für Ihre Unterstützung! Die Datenauswertung erfolgt anonym, Sie müssen Ihre Testdatei für das Zurückschicken also nicht umbenennen.

Bearbeitungszeit: Nehmen Sie als Richtwert für die Bearbeitung des ganzen Tests **90 Minuten**. Der Test umfasst 100 Elemente mit insgesamt 234 Wörtern, pro Element haben Sie damit im Schnitt etwas weniger als eine Minute Zeit für die Übersetzung inkl. Recherche. Wenn Sie etwas länger als 90 Minuten brauchen, ist das allerdings kein Problem. Bitte versuchen Sie aber, den Test vollständig zu bearbeiten.

Erläuterungen zur Vorgehensweise: In der Datei „Probandentest_Wiedmann.xlsx“ finden Sie eine Tabelle mit sechs Spalten. **In der ersten Spalte** sind die zu übersetzenden Elemente zu finden. Dabei handelt es sich um Texte aus der Benutzeroberfläche verschiedener Programme, also beispielsweise einzelne Menüpunkte oder auch Informationen in Fenstern, die ausgewählt werden können. Hier ein Beispiel:



Die Elemente in diesem Fenster, wie der Fenstertitel („Bilingual export wizard“) sowie die Beschriftungen der Optionsfelder („memoQ XLIFF“ ...), der Kontrollkästchen („Include skeleton and preview“ ...) oder auch der Schaltflächen („Export“ ...), könnten solche zu übersetzenden Elemente in der Datei „Probandentest_Wiedmann.xlsx“ sein. Die Elemente stammen aus verschiedenen Programmen, daher ist in der Datei jeweils der Programmtyp in Fettdruck als Hilfe für die Übersetzung mit angegeben.

Bitte tragen Sie in **die zweite Spalte** Übersetzungsvorschläge ein, die Ihnen spontan, also **ohne irgendeine Recherche**, einfallen. Falls Ihnen spontan mehrere Vorschläge einfallen, tragen Sie sie in

der Reihenfolge ein, in der Sie Ihnen einfallen. Falls Ihnen nichts einfällt, kann diese Spalte auch leer bleiben.

Tragen Sie in **die dritte Spalte** Ihren **endgültigen Übersetzungsvorschlag** ein, also nach Abschluss Ihrer Überlegungen bzw. einer ggf. erforderlichen oder gewünschten Recherche. Falls sich gegenüber dem spontanen Übersetzungsvorschlag nichts geändert hat, tragen Sie diesen hier trotzdem bitte nochmals ein. Bitte geben Sie in dieser Spalte möglichst für alle zu übersetzenden Elemente einen Übersetzungsvorschlag an, auch wenn das ein Vorschlag ist, mit dem Sie nicht zufrieden oder bei dem Sie sich nicht sicher sind.

Bitte führen Sie nur dann eine Recherche durch, wenn Sie dies für erforderlich halten, und geben Sie entsprechend in den **Spalten 4, 5 und 6** an, **ob Sie eine Recherche durchgeführt haben, wo Sie recherchiert haben** (Mehrfachnennungen möglich) und **ob die Recherche erfolgreich war**. Beachten Sie dabei aber, dass einzelne Recherchevorgänge nicht zu viel Zeit in Anspruch nehmen, damit Sie den gesamten Test auch tatsächlich in etwa 90 Minuten abschließen können. Brechen Sie die Recherche also ggf. ab, wenn Sie nicht schnell genug fündig werden.

Für die Recherche können Sie alle gewünschten Quellen verwenden, auch Suchmaschinen. Bitte **geben Sie die zu übersetzenden Elemente aber nicht als Ganzes in eine Suchmaschine ein** und verwenden Sie auch **keine maschinelle Übersetzung**.

B.2 Excel-Tabelle mit dem eigentlichen Test

Nachfolgend findet sich der Übersetzungstest in einer PDF-Version.

Probandentest

Zu übersetzende Elemente	Erster spontaner Einfall für die Übersetzung sowie ggf. alle weiteren Ü-Vorschläge ohne Recherche (in der Reihenfolge des Einfallens)	Endgültiger Ü-Vorschlag	Recherche durchgeführt (ja/nein)	Wo haben Sie recherchiert? (Mehrfachnennungen möglich)	War die Recherche erfolgreich? (ja/nein/weiß nicht)
Audiobearbeitung					
File format support					
File type					
Master gain control					
MP2 export					
Report generated to					
Sample rate conversion					
Save log to					
Split horizontally					
Start script					
Bildbearbeitung					
Many usability improvements					
Browser					
Bookmarking tools					
Email image					
New window					
Save to pocket					
Translate this page					
Web developer					
Dateimanager					
Brief view					
Calculate checksum					
Clear log file					
Copy block left					
Copy permissions					
Feed to listbox					
First difference					
Keep scrolling					
NTFS attributes					
Paint background					
Regular expressions					
Save all					
Save to					
Size to					
To the directory					
E-Mail					
Attribute name					
Busy searching					
Common addresses					
Delete group					

Probandentest

Duplicate email addresses					
File not specified					
Import pine file					
LDAP servers					
New vCard					
Reset to default					
Editor					
Change search engine					
Collapse level					
EOL conversion					
Find in files					
Increase line indent					
Load session					
Next search result					
On losing focus					
Paste special					
Print now					
Proper case					
Tab to space					
Word wrap					
HTML-Editor					
A date picker					
All tags mode					
Body text					
CSS properties					
Fixed width					
Get natural size					
Line height					
List properties					
Markup cleaner					
More options					
Ordered list					
Shape and size					
Medienplayer					
Access output					
Add directory					
Add to playlist					
Audio filtering failed					
Audio resampler					
Create folder					
Repeat one					
Reverse stereo					
Sout stream					
Textverarbeitung					
Color scale					
Columns after					
Fill single edit					
Fill up					
Goal seek					
Manage names					
Manage track changes					
Mark dependents					
Next page					
Page format					
Page right					

Probandentest

Position in document					
Recalculate hard					
Split window					
To next sheet					
Decoding video					
Videobearbeitung					
File error					
Nb lines					
No change					
Play filtered					
Previous frame					
Save as JPEG					
Two pass					
Video size					
Wide search					

B.3 Kontextarten der Strings im Übersetzungstest

In diesem Abschnitt werden die Strings aus dem Übersetzungstest nach Kontextarten sortiert aufgeführt, jeweils mit zugehörigem Programmtyp. Zuerst folgt eine Liste mit Strings, die nicht ausgewertet wurden, weil sie bei der finalen Annotation mehreren Kontextarten angehörten oder andere Probleme vorlagen. Daran schließen sich die verwendeten Strings mit Kontextarten sowie eine Übersicht über die Untersysteme und Features an, denen die Strings im Korpus angehören. Dabei sind die Strings nach Kontextart sortiert.

Nicht ausgewertete Strings aus dem Übersetzungstest:

File format support	Audiobearbeitung
Sample rate conversion	Audiobearbeitung
Save log to	Audiobearbeitung
Split horizontally	Audiobearbeitung
Email image	Browser
Brief view	Dateimanager
Clear log file	Dateimanager
Feed to listbox	Dateimanager
Import pine file	E-Mail
Change search engine	Editor
Increase line indent	Editor
Word wrap	Editor
List properties	HTML-Editor
Markup cleaner	HTML-Editor
Audio filtering failed	Medienplayer
Reverse stereo	Medienplayer
Mark dependents	Office-Anwendung
To next sheet	Office-Anwendung
Previous frame	Videobearbeitung

**Ausgewertete Strings der Kategorie „Kein Kontext“ aus dem
Übersetzungstest:**

File type	Audiobearbeitung
Many usability improvements	Bildbearbeitung
New window	Browser
Translate this page	Browser
First difference	Dateimanager
Delete group	E-Mail
File not specified	E-Mail
Find in Files	Editor
Next search result	Editor
Print now	Editor
Shape and size	HTML-Editor
Add directory	Medienplayer
Add to playlist	Medienplayer
Create folder	Medienplayer
Color scale	Office-Anwendung
Manage names	Office-Anwendung
Next page	Office-Anwendung
Page format	Office-Anwendung
File error	Videobearbeitung
No change	Videobearbeitung
Save as JPEG	Videobearbeitung
Video size	Videobearbeitung

Ausgewertete Strings der Kategorie „Situativer Kontext“ aus dem Übersetzungstest:

Report generated to	Audiobearbeitung ⁶
Start script	Audiobearbeitung ¹
Bookmarking tools	Browser ⁵
Copy block left	Dateimanager ⁷
Copy permissions	Dateimanager ¹
Paint background	Dateimanager ¹
Save all	Dateimanager ⁸
Save to	Dateimanager ⁶
Size to	Dateimanager ⁶
To the directory	Dateimanager ⁶
Duplicate email addresses	E-Mail ³
Reset to default	E-Mail ⁴
Load Session	Editor ¹
Access output	Medienplayer ¹
Repeat one	Medienplayer ⁹
Columns after	Office-Anwendung ⁷
Fill up	Office-Anwendung ⁷
Page right	Office-Anwendung ⁷
Position in document	Office-Anwendung ¹
Split window	Office-Anwendung ²
Decoding video	Videobearbeitung ⁵
Play filtered	Videobearbeitung ¹⁰

Diese Strings gehören folgenden Untersystemen und Features an:

- Untersystem PROBLEM_WORTART mit folgenden Features:
 - ¹: verb.inf_oder_subst.sg
 - ²: verb.inf_oder_past.part
 - ³: verb.inf_oder_adj
 - ⁴: verb.inf_oder_past.part_oder_subst.sg
 - ⁵: gerund_oder_v.subst_1.teil.kompositum
- Untersystem PROBLEM_GUI-BEZUG mit folgenden Features:
 - ⁶: über_präposition
 - ⁷: über_richtung
- Untersystem PROBLEM_FLEXIONSENDUNG mit folgenden Features:
 - ⁸: flexion_adj
 - ⁹: flexion_pronomen
 - ¹⁰: flexion_partizip

Ausgewertete Strings der Kategorie „Fachlicher Kontext“ aus dem Übersetzungstest:

MP2 export	Audiobearbeitung ⁷
Web developer	Browser ⁵
Calculate checksum	Dateimanager ¹
NTFS attributes	Dateimanager ⁷
Regular expressions	Dateimanager ²
Attribute name	E-Mail ⁵
LDAP servers	E-Mail ⁷
New vCard	E-Mail ⁴
Collapse level	Editor ³
EOL conversion	Editor ⁶
A date picker	HTML-Editor ⁵
Body text	HTML-Editor ⁵
CSS properties	HTML-Editor ⁷
Fixed width	HTML-Editor ²
More options	HTML-Editor ¹
Ordered list	HTML-Editor ²
Audio resampler	Medienplayer ⁵
Goal seek	Office-Anwendung ⁵

Diese Strings gehören folgenden Untersystemen und Features an:

- ¹: substantivischer_fachterm
- ²: adjektiv-partizip_plus_substantiv-fachterm
- ³: verbaler_fachterm
- ⁴: eigennamen
- Untersystem FACHTERM_ANEINANDERREIHUNG mit folgenden Features:
 - ⁵: komp_zwei_substantive
 - ⁶: komp_mit_abkürzung
 - ⁷: komp_mit_eigennamen

Ausgewertete Strings der Kategorie „Funktionaler Kontext“ aus dem Übersetzungstest:

Master gain control	Audiobearbeitung ¹
Save to pocket	Browser ³
Keep scrolling	Dateimanager ⁴
Busy searching	E-Mail ⁴
Common addresses	E-Mail ⁴
On losing focus	Editor ⁴
Paste special	Editor ⁴
Proper case	Editor ⁴
Tab to space	Editor ²
All tags mode	HTML-Editor ⁴
Get natural size	HTML-Editor ⁴
Line height	HTML-Editor ²
Sout stream	Medienplayer ⁴
Fill single edit	Office-Anwendung ⁴
Manage track changes	Office-Anwendung ⁴
Recalculate hard	Office-Anwendung ⁴
Nb lines	Videobearbeitung ⁴
Two pass	Videobearbeitung ⁴
Wide search	Videobearbeitung ⁴

Diese Strings gehören folgenden Untersystemen und Features an:

- ¹: drei_substantive im Untersystem UNKLARE_ANEINANDERREIHUNG und homonym_it-term für das Wort „control“
- ²: homonym_it-term
- ³: metaphorik
- ⁴: unklare_bedeutung

B.4 Änderungen vor Beginn der Auswertung

Vor Beginn der Auswertung des Übersetzungstests wurden die ausgefüllten Übersetzungstests zu einer Excel-Tabelle zusammengefasst. In dieser Excel-Tabelle wurden verschiedene Vereinheitlichungen und Fehlerkorrekturen vorgenommen, damit sie anschließend maschinell ausgewertet werden konnte. Diese Änderungen werden in diesem Abschnitt nach Spalten sortiert beschrieben.

Spalten „Erster spontaner Einfall“ und „Endgültiger Ü-Vorschlag“

- In Zellen, in denen keine Angabe gemacht wurde, wurden eingefügte Schrägstriche, Bindestriche und Fragezeichen gelöscht.
- In der Spalte „Endgültiger Ü-Vorschlag“ enthaltene Fragezeichen oder Ausrufezeichen am Ende wurden gelöscht.
- Die Groß-/Kleinschreibung wurde in allen Zellen angepasst. Zum Beispiel wurde das erste Wort des Strings immer großgeschrieben und die Groß-/Kleinschreibung von Abkürzungen wurde vereinheitlicht. Beispiele für vorhandene Varianten: „pine“ vs. „,pine“ und „PINE“, „Sout“ vs. „,SOUT“.
- Tipp- und Rechtschreibfehler wurden korrigiert.
- Die Schreibung von Komposita wurde vereinheitlicht. Beispiele für vorhandene Varianten: „Dateiformatunterstützung“ vs. „Dateiformat-Unterstützung“, „MP2-Export“ vs. „MP2 Export“, „Weiter Scrollen“ vs. „Weiter scrollen“, „Neue vCard“ vs. „Neue v-Card“.
- Schreibung und Geschlecht bei Fremdwörtern wurde vereinheitlicht. Beispiele für vorhandene Varianten: „Skript“ vs. „Script“, „decodieren“ vs. „dekodieren“, „das Frame“ vs. „der Frame“.
- Drei Punkte bei ansonsten identischen Strings (z. B. bei Übersetzungsvorschlägen für „Busy searching“ und „Save to“) wurden gelöscht bei „Save to“, aber bei „Busy searching“ beibehalten, weil sie im letzteren Fall bedeutungsunterscheidend genutzt werden.
- Potenzielle kleinere Übersetzungsfehler, primär Singular vs. Plural („Bookmarking tools“: „Lesezeichen-Tool“ vs. „Lesezeichen-Tools“) und Grammatik („Copy block left“: „Linker Block kopieren“ vs. „Linken Block kopieren“; „Save to“: „Speicher auf“ -> „Speichern auf“), wurden korrigiert. Aber: „No change“ = „Keine Änderung“ und „Keine Änderungen“ sowie „Play filtered“ = „Gefiltertes abspielen“ und „Gefiltertes Abspielen“ wurden nicht korrigiert, weil solche Unterschiede verschiedene Übersetzungsalternativen darstellen.
- Übersetzungsvorschläge wie „Horizontal teilen/unterteilen/spalten“ wurden ausformuliert in die vollständigen Übersetzungsalternativen, wie hier „Horizontal teilen“, „Horizontal unterteilen“ und „Horizontal spalten“.
- Verschiedene Arten von Anführungszeichen wurden vereinheitlicht, z. B. bei Übersetzungen für „All tags mode“.
- Alternativen bei „Erster Spontaner Einfall“, die nur auf der Rechtschreibung beruhen („Dateiformatunterstützung“ vs. „Dateiformat-Unterstützung“) wurden ignoriert.
- Kleinere inhaltlich irrelevante Unterschiede, z. B. bei Artikeln oder Präpositionen, wurden vereinheitlicht. Beispiele: „A date picker“: Übersetzung mit und ohne „ein“. „Add to playlist“: mit „Zur ... hinzufügen“ vs.

„Zu ... hinzufügen“.

- Abweichungen bei der Nutzung von Fugen-s bei Komposita (Attributsname vs. Attributname) wurden vereinheitlicht.

Spalte „Recherche durchgeführt?“

- P17: Eintrag „ein“ wurde geändert in „nein“.
- Einzelne Probandinnen hatten nur jeweils „ja“ angegeben und statt „nein“ die betreffenden Zellen leer gelassen. In diesen Fällen wurde „nein“ eingefügt.
- P16: Im Feld „Wo haben Sie recherchiert?“ waren Angaben enthalten, im zugehörigen Feld „Recherche durchgeführt?“ nicht. In diesen Fällen wurde „ja“ ergänzt.
- P08: Die Spalte wurde teils leer gelassen, wenn für den entsprechenden String kein Übersetzungsvorschlag angegeben wurde. Dies wurde so übernommen, weil vermutlich nicht ausreichend Zeit für Recherche vorhanden war.
- P14: In der Spalte „Erster spontaner Einfall“ war ein Eintrag vorhanden, aber in der Spalte „Endgültiger Ü-Vorschlag“ nicht, in den weiteren Spalten fanden sich ebenfalls keine Angaben. In diesem Fall wurde folgende Ergänzung vorgenommen: In die Spalte „Recherche durchgeführt?“ wurde „nein“ eingefügt, in die Spalte „Endgültiger Ü-Vorschlag“ wurde der Eintrag aus der Spalte „Erster spontaner Einfall“ übernommen.

Spalte „War die Recherche erfolgreich?“

- P05: „War die Recherche erfolgreich?“ = „ja/weiß nicht“ geändert in „ja, weiß nicht“, „nein/weiß nicht“ geändert in „nein, weiß nicht“.
- P11: „War die Recherche erfolgreich?“ = „ja (kann, wie auch in allen anderen Fällen, nur unter Berücksichtigung des fehlenden Kontextes gesagt werden)“: Die Ergänzung in der Klammer wurde gelöscht.
- In Fällen, in denen kein endgültiger Übersetzungsvorschlag (und teils auch kein spontaner Einfall) angegeben war, aber gleichzeitig in der Spalte „Recherche durchgeführt?“ die Angabe „ja“ vorhanden war, war zum Teil im Feld „War die Recherche erfolgreich?“ keine Information enthalten. Für dieses Feld wurde in solchen Fällen der Wert „nein“ angegeben.
- P07: „Wo haben Sie recherchiert?“ = „weiß nicht“, aber gleichzeitig „Recherche durchgeführt?“ = „nein“: wurde gelöscht.

- In 8 Fällen wurde in der Spalte „Recherche durchgeführt?“ die Antwort „ja“ angegeben und in der Spalte „Wo haben Sie recherchiert?“ waren Informationen enthalten, aber in der Spalte „War die Recherche erfolgreich?“ fand sich keine Angabe. Diese Fälle wurden unverändert übernommen.
- P11: In der Spalte „War die Recherche erfolgreich?“ wurde in einem Fall „weiß nicht (abgebrochen)“ angegeben. Dies wurde in „weiß nicht, abgebrochen“ geändert. In einem weiteren Fall wurde „weiß nicht (abgebrochen) (hier ist, wie auch im Folgenden, das wahrscheinlichste Ergebnis der endgültige Übersetzungsvorschlag)“ angegeben. Hier wurde die Information in der Klammer gelöscht.
- P15: In der Spalte „War die Recherche erfolgreich?“ wurde in einem Fall „weiß nicht, kommt auf Spezialisierungsgrad des Programmes etc an“ angegeben. Die Information nach dem Komma wurde gelöscht.
- Wenn in der Spalte „War die Recherche erfolgreich?“ die Angabe „Nein, Typo?“ bzw. „Nein. Typo?“ enthalten war, wurde diese in „nein“ geändert. Die Werte „ja, weiß nicht“, „nein, weiß nicht“ und „Weiß nicht, abgebrochen“ wurden unverändert übernommen.

Anhang C

Anhang zur Eye-Tracking- und Key-Logging-Studie

Dieser Anhang enthält Details zu verschiedenen Aspekten der Eye-Tracking- und Key-Logging-Studie.

C.1 Items aus der Eye-Tracking- und Key-Logging-Studie

In den folgenden Tabellen sind die Items für die Eye-Tracking- und Key-Logging-Studie enthalten. In der linken Spalte wird jeweils das Item mit Programmtyp angegeben, in der rechten Spalte die Kontextart.¹ Zunächst werden die Practice-Items aufgeführt, für die allerdings keine Kontextarten angegeben sind. Danach folgen die jeweils 18 Items der vier Teilprojekte, die in der Studie in unterschiedlicher Reihenfolge durchlaufen wurden. Weitere Informationen zu den genauen Untersystemen und Features der Items finden sich in Anhang [B.3](#).

Practice-Items:

- Bildbearbeitung: By color
- Office-Anwendung: Color scale
- Bildbearbeitung: Configure input devices
- Audiobearbeitung: Save project
- Dateimanager: Save all

Teilprojekt 1:

¹Informationen zu den Features, die im Korpus annotiert wurden, finden sich in Anhang [B.3](#).

Office-Anwendung: Page format	Kein Kontext
E-Mail: Attribute name	Fachlicher Kontext
Office-Anwendung: Page right	Situativer Kontext
Videobearbeitung: Wide search	Funktionaler Kontext
Office-Anwendung: Next page	Kein Kontext
Audiobearbeitung: Master gain control	Funktionaler Kontext
Dateimanager: First difference	Kein Kontext
HTML-Editor: All tags mode	Funktionaler Kontext
Office-Anwendung: Columns after	Situativer Kontext
E-Mail: Duplicate email addresses	Situativer Kontext
HTML-Editor: Body text	Fachlicher Kontext
Office-Anwendung: Manage names	Kein Kontext
E-Mail: Common addresses	Funktionaler Kontext
Videobearbeitung: Play filtered	Situativer Kontext
Audiobearbeitung: MP2 export	Fachlicher Kontext
Dateimanager: Save to	Situativer Kontext
Medienplayer: Sout stream	Funktionaler Kontext
Dateimanager: Size to	Situativer Kontext

Teilprojekt 2:

HTML-Editor: Shape and size	Kein Kontext
Dateimanager: Copy block left	Situativer Kontext
E-Mail: File not specified	Kein Kontext
HTML-Editor: More options	Fachlicher Kontext
Dateimanager: Copy permissions	Situativer Kontext
Medienplayer: Create folder	Kein Kontext
Videobearbeitung: Two pass	Funktionaler Kontext
Editor: EOL conversion	Fachlicher Kontext
Dateimanager: To the directory	Situativer Kontext
Editor: Find in files	Kein Kontext
Medienplayer: Audio resampler	Fachlicher Kontext
E-Mail: Reset to default	Situativer Kontext
Office-Anwendung: Fill single edit	Funktionaler Kontext
E-Mail: LDAP servers	Fachlicher Kontext
Dateimanager: NTFS attributes	Fachlicher Kontext
Office-Anwendung: Position in document	Situativer Kontext
Editor: Print now	Kein Kontext
Browser: New window	Kein Kontext

Teilprojekt 3:

Medienplayer: Add directory	Kein Kontext
Videobearbeitung: Video size	Kein Kontext
Editor: Proper case	Funktionaler Kontext
Medienplayer: Repeat one	Situativer Kontext
Dateimanager: Keep scrolling	Funktionaler Kontext
Editor: Next search result	Kein Kontext
HTML-Editor: CSS properties	Fachlicher Kontext
Dateimanager: Regular expressions	Fachlicher Kontext
Office-Anwendung: Split window	Situativer Kontext
Dateimanager: Calculate checksum	Fachlicher Kontext
Editor: On losing focus	Funktionaler Kontext
E-Mail: Busy searching	Funktionaler Kontext
HTML-Editor: Line height	Funktionaler Kontext
Office-Anwendung: Recalculate hard	Funktionaler Kontext
Medienplayer: Access output	Situativer Kontext
E-Mail: Delete group	Kein Kontext
Editor: Collapse level	Fachlicher Kontext
HTML-Editor: A date picker	Fachlicher Kontext

Teilprojekt 4:

Office-Anwendung: Goal seek	Fachlicher Kontext
Browser: Web developer	Fachlicher Kontext
HTML-Editor: Ordered list	Fachlicher Kontext
Videobearbeitung: Save as JPEG	Kein Kontext
Audiobearbeitung: File type	Kein Kontext
Browser: Save to pocket	Funktionaler Kontext
Editor: Load session	Situativer Kontext
Dateimanager: Paint background	Situativer Kontext
Browser: Bookmarking tools	Situativer Kontext
Editor: Tab to space	Funktionaler Kontext
HTML-Editor: Get natural size	Funktionaler Kontext
Office-Anwendung: Fill up	Situativer Kontext
Browser: Translate this page	Kein Kontext
Medienplayer: Add to playlist	Kein Kontext
Editor: Paste special	Funktionaler Kontext
E-Mail: New vCard	Fachlicher Kontext
HTML-Editor: Fixed width	Fachlicher Kontext
Office-Anwendung: Manage track changes	Funktionaler Kontext

C.2 Anweisungsdatei für die Eye-Tracking- und Key-Logging-Studie

Nachfolgend findet sich die Anweisungsdatei für die Eye-Tracking- und Key-Logging-Studie.

Anweisungen

Bei dieser Studie müssen Sie Elemente aus grafischen Benutzeroberflächen verschiedener Programme übersetzen. Die Strings sind dabei nicht nach Programmen gruppiert, sondern werden Ihnen durchmischt angezeigt. Damit Sie dennoch wissen, aus welchem Programm ein String stammt, wird der Programmtyp jeweils vor dem zu übersetzenden String angegeben. Der String folgt dann nach einem Doppelpunkt. Sie müssen aber nur die eigentlichen Strings nach dem Doppelpunkt übersetzen, die angegebenen Programmtypen vor dem Doppelpunkt dienen lediglich zu Ihrer Information.

Folgende Programmtypen sind in der Studie enthalten:

- Audibearbeitung
- Bildbearbeitung
- Browser
- Dateimanager
- Editor
- E-Mail-Programm
- HTML-Editor
- Medienplayer
- Office-Anwendung
- Videobearbeitung

Beachten Sie beim Übersetzen Folgendes:

1. Bitte verändern Sie Ihre Sitzhaltung während des Eye-Trackings möglichst nicht.
2. Bitte verschieben Sie die angezeigten Fenster nicht.
3. Wenn Sie für einen Terminus verschiedene Übersetzungsmöglichkeiten kennen, orientieren Sie sich an der Microsoft-Terminologie, falls bekannt.
4. Die Schreibung von Fremdwörtern, Komposita usw. spielt keine Rolle.
5. Es ist keine Recherche möglich. Versuchen Sie dennoch, für jedes GUI-Element einen Übersetzungsvorschlag anzugeben, auch wenn dieser nur tentativ sein sollte und noch geprüft werden muss.

Ablauf der Studie:

1. In der rechten Hälfte des Bildschirms sind zwei Fenster zu sehen. Im linken dieser Fenster steht der Ausgangstext. Den Zieltext geben Sie in das rechte dieser Fenster ein.
2. Übersetzen Sie jeweils Zeile für Zeile/String für String und halten Sie dabei folgende Abfolge von Schritten ein:

- a. Geben Sie zunächst einen Übersetzungsvorschlag ein, der Ihnen möglichst korrekt erscheint.
- b. Links befindet sich ein Fenster mit der Information „Benötigen Sie für die korrekte Übersetzung mehr Informationen?“. **Beantworten Sie diese Frage**, indem Sie dort entweder **Ja** oder **Nein** auswählen, je nachdem, ob Sie der Ansicht sind, dass Ihr Übersetzungsvorschlag auch ohne weitere Informationen korrekt ist oder ob Sie für einen korrekten Vorschlag weitere Informationen benötigen. Benutzen Sie dafür die **Shortcuts Alt+J** für **Ja** bzw. **Alt+N** für **Nein**.
- c. Unabhängig davon, wie Sie diese Frage beantworten, wird jetzt weiteres Material angezeigt.
- d. **Prüfen Sie dieses Material in jedem Fall.** Falls sich dadurch Änderungen für Ihren Übersetzungsvorschlag ergeben, korrigieren Sie jetzt Ihren in Schritt 2a eingegebenen Übersetzungsvorschlag. Beantworten Sie dann die Frage „Waren diese Informationen für die korrekte Übersetzung hilfreich?“ mit **Ja (Alt+J)** oder mit **Nein (Alt+N)**.