

Poesie als Fehler

Ein ‘Tool Misuse’-Experiment zur Prozessierung von Lyrik

Sluyter-Gäthje, Henny

sluytergaeth@uni-potsdam.de
Universität Potsdam, Germany

Trilcke, Peer

trilcke@uni-potsdam.de
Universität Potsdam, Germany

Problemhorizont und Fragestellung

Analysen der Computational Literary Studies (CLS) vorverarbeiten ihre Untersuchungsgegenstände typischerweise mit Tools des Natural Language Processing (NLP). Dabei weichen literarische Texte aufgrund ihrer historischen und/oder ästhetischen Eigenart teils eklatant von den Daten ab, auf deren Grundlage die *Models* der NLP-Tools erstellt wurden. Entsprechend sinkt die *Accuracy* der Tools etwa bei der Tokenisierung, der Lemmatisierung oder dem POS-Tagging von literarischen Texten (Scheible et al. 2011; für POS-Tagger: Rayson et al. 2007; Herrmann 2018; Bamman 2020), wobei die besondere ‘Schwierigkeit’ literarischer Texte alle gängigen Tools zu betreffen scheint (für Lemmatisierung vgl. Ortman, Roussel, Dipper 2019: 219).

Der *Performance Drop* der NLP-Tools bei Literatur ist ein computerlinguistisches Problem der Domänenadaption. Für die CLS könnte die ‘Fehlerhaftigkeit’ der Tools im Sinne devianzpoetischer Positionen (Fricke 1981) zudem die Möglichkeit bieten, ein computationelles Verständnis vom spezifischen Abweichungscharakter literarischer Texte auszubilden: Wenn die Tools für standardsprachliche Texte entwickelt wurden, dann könnten die Fehler, die diese Tools auf nicht-standardsprachlichen Texten wie der Literatur produzieren, etwas über deren Charakteristika aus computationeller Sicht verraten.

Das folgende Experiment geht von dieser basalen Überlegung aus. Gegenstand des Experiments ist die Lyrik, der regelmäßig eine “Tendenz zu erhöhter Devianz” (Müller-Zettelman 2000: 100) attestiert wird, die sich in Form gattungsspezifischer “Störungen” (Zymner 2019: 29f.) ausdrückt. Wir entwickeln eine Pipeline, die gezielt ‘Fehler’ von NLP-Tools provoziert und diese ‘Fehler’ regelbasiert typologisiert. Damit möchten wir für die CLS auch exemplarisch den Ansatz des *Tool Misuse* profilieren, bei dem die Erzeugung von ‘fehlerhaftem’ Output computationeller Tools Grundlage für Erkenntnisse über Literatur wird.

Operationalisierung und Korpora

Fehler von Tools wären idealerweise über Daten mit Gold-Standard-Annotationen zu ermitteln. Solche Daten liegen für unser Szenario nicht vor. Deshalb implementieren wir als Workaround eine Pipeline, die folgende Idee umsetzt: Die Verarbeitung einer Zeichenkette durch ein NLP-Tool (Tokenisierung, Lemmatisierung, POS-Tagging) sollte eine Zeichenkette ergeben, die in

einem Wörterbuch zu finden ist: Die Ausgabe, die aus der Eingabe “geht” resultiert, sollte sich als Lemma “gehen” der Wortart “Verb” in einem Wörterbuch finden. Da wir nicht die spezifischen Sprachverarbeitungsprobleme eines individuellen NLP-Tools in den Blick nehmen wollen, lassen wir unser Lyrikkorpus von mehreren Tools prozessieren und betrachten jene Types als potenzielle Fehler, deren von den Tools ausgegebene Lemmata (inkl. POS-Tag) wir nicht in einem Wörterbuch finden.

Prozessiert wird ein deutsches kanonbasiertes Korpus mit ‘prototypischer’ Lyrik, das Texte von 12 Autor:innen umfasst, die in der statistisch begründeten Anthologie von Braam (2019) am häufigsten mit Gedichten vertreten sind. Ins Korpus aufgenommen wurden sämtliche im TextGridRepository¹ verfügbaren Gedichte der 12 Autor:innen (Goethe, Schiller, Hölderlin, Eichendorff, Heine, Droste-Hülshoff, Claudius, Mörike, Storm, Rilke, Trakl, Tucholsky).² Zur eklatanten Unterrepräsentation von Autor:innen im Lyrikkanon siehe (Bers 2020). Vergleichend verwenden wir ein Prosa-korpus mit 100 deutschsprachigen Romanen aus dem 19. Jahrhundert, zusammengestellt aus dem TextGridRepository und Projekt Gutenberg³; Grundlage für die Romanauswahl ist die Erwähnung in Beutin et al. (2019).

Unsere ‘Fehler-Pipeline’ präferiert *Recall* vor *Precision*. Sie produziert also zunächst Indizien für potenzielle Fehler. Eine größere Menge an *False Positives* ist zu erwarten, etwa weil wir *Out-of-Vocabulary*-Wörter der verwendeten Wörterbücher prozessieren.

Pipeline

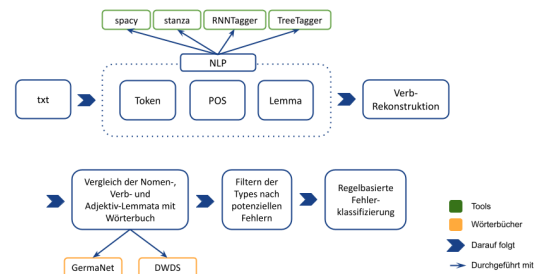


Abb. 1: Pipeline zur Fehlertypisierung der Korpora

Die in die Pipeline eingespeisten Textdateien (txt-Format) werden tokenisiert, POS-tagged und lemmatisiert (Abb. 1). Um potenzielle Fehler als taggerspezifische Fehler einzuordnen, verwenden wir vier NLP-Tools. Diese sind frei verfügbar, stellen *Models* für das Deutsche bereit und führen alle NLP-Schritte durch. Den RNNTagger (Schmid 2019) (im Folgenden “RNN”) und stanza (Qi et al. 2020) haben wir aufgrund der guten Performance nach Ortman et al. (2019) ausgewählt. Die *Models* beider Tools sind neuronal. Zusätzlich verwenden wir spacy (Hon-nibal et al. 2020), das für das Deutsche ein *Transformer Model* (de_dep_news_trf) zur Verfügung stellt. Um auch einen nicht-neuronalen Tagger aufzunehmen, verwenden wir den TreeTagger (Schmid 1994; Schmid 1995) (im Folgenden “Tree”), der auf Grundlage von *Decision Trees* annotiert.

Bei der Verb-Rekonstruktion werden abgetrennte Verbpartikel regelbasiert (mithilfe von POS-Tag und Abstandsmaßen) an das dazugehörige Lemma des Verbs angefügt. Die darauffolgende

Fehlerbetrachtung ist auf dem Vergleich der Lemmata mit Wörterbüchern gestützt. Da wir erwarten, dass Fehler, die ihren Ursprung in der Domänenspezifität haben, sich hauptsächlich auf Inhaltswörter, d.i. Nomen (NOUN), Verben (VERB) und Adjektive (ADJ), beschränken, werden der Wörterbuchvergleich und die darauffolgenden Schritte nur für diese Wortarten durchgeführt.

Die Lemmata werden unter Berücksichtigung der Wortart in der lexikalisch-semanticen Ressource GermaNet⁴ (Hamp & Feldweg 1997; Henrich & Hinrichs 2010), die 146.787 Lemmata (ADJ, NOUN, VERB) umfasst, sowie im Digitalen Wörterbuch der deutschen Sprache (DWDS)⁵ (Klein & Geyken 2010) nachgeschlagen, das über 230.000 Wörter umfasst.

Für jeden Type erzeugen wir Listen von Lemmata pro Tool. Für die 5.144 Gedichttexte ergibt das eine Typezahl von 70.422 (Tab. 1, Spalte "all"), die je nach Tagger aufgrund verschiedener Tokenisierung und POS-Tagging zu unterschiedlichen Tokenzahlen führt. Um zu verhindern, dass Fehler toolspezifisch sind, werten wir einen Type nur dann als potenziellen Fehler, wenn mindestens zwei Tools den Type lemmatisiert haben und kein Lemma aus den Listen im Wörterbuch gefunden wurde (Tab. 01, Spalte "pFail"). Diese potenziellen Fehler werden regelbasiert in Fehlertypen eingeteilt.

Tab. 1: Anzahl der Types und Token (ADJ, NOUN, VERB), differenziert nach Lyrik- und Prosa-korpus, jeweils für die Gesamtkorpora ("all") und für die Sets mit potenziellen Fehlern ("pFail"). Minimalwerte sind kursiv und Maximalwerte sind fett gedruckt.

	Lyrik	Lyrik	Prosa	Prosa
	all	pFail	all	pFail
Types	70.422	24.244	263.042	115.785 ⁶
spacy_Token	397.924	36.549	3.967.566	250.800
stanza_Token	390.605	34.493	3.958.315	254.049
RNN_Token	390.325	36.115	3.977.869	268.686
Tree_Token	414.395	39.243	4.276.486	280.850

Analyse

Typologie potenzieller Fehler

Auf dem pFail-Set führen wir eine regelbasierte Typologisierung durch. Die Typen postulieren wir ausgehend von manuellen Inspektionen des pFail-Set. Für jeden Typen wird eine Regel formuliert. Die Regeln werden daraufhin in einer spezifischen Reihenfolge auf das pFail-Set angewendet. Mehrfachtypisierungen sind nicht möglich; die Reihenfolge der Regelanwendung hat mithin Konsequenzen für die Menge an jeweils identifizierten Vorkommnissen. Die Reihenfolge lautet: CONTRACT, ELISION_APO, PUNC, SHORT, COMP_DASH, COMP, PART_ADJ, ELISION_SIMPLE, ORTH_UPPER, ORTH_SZ, PREFIXED, EPITHESES, ELISION_END. Die Typendefinitionen führt Tab. 2 auf.

Tab. 2: Typen potenzieller Fehler und Identifikationsregeln

Bezeichnung	Beschreibung	Regel
PUNC	Satzzeichen sind als ADJ, NOUN, VERB pos-getaggt oder Satzzeichen sind als Teil eines Wortes tokenisiert.	Wenn im Type ein Satzzeichen (Ausnahme: Apostrophe und Bindestriche, bei denen vor und nach dem Bindestrich mindestens ein Buchstabe steht) vorkommt, typisiere.
SHORT	Einzelne Buchstaben oder Ziffern sind als ADJ, NOUN, VERB pos-getaggt.	Wenn ein Type nur zwei Zeichen oder weniger aufweist, typisiere.
ORTH_SZ	Wörter verwenden die historische Schreibung mit "ß".	Ersetze "ß" im Lemma durch "ss" und prüfe im Wörterbuch; wenn im Wörterbuch zu finden, typisiere.
ORTH_UPPER	ADJ oder VERB wurden am Versanfang großgeschrieben.	Wenn ein Type am Anfang einer Zeile steht, ersetze initiale Großschreibung bei ADJ und VERB mit Kleinschreibung und prüfe im Wörterbuch; wenn im Wörterbuch zu finden, typisiere.
ELISION_APO	Vokale innerhalb eines Wortes wurden getilgt und durch Apostroph ersetzt.	Wenn innerhalb eines Type ein Apostroph vorkommt, typisiere.
ELISION_SIMPLE	Vokale wurden in der vorletzten oder letzten Silbe eines Wortes ohne Markierung durch Apostroph getilgt.	Wenn ein Type auf "nen", "ner", "ne", "n" endet und davor kein Vokal und kein "l" oder "r" steht, typisiere.
ELISION_END	Auslautende Vokale in NOUN wurden getilgt.	Ergänze am Ende eines NOUN ein "e" und prüfe im Wörterbuch; wenn im Wörterbuch zu finden, typisiere.
EPITHESES	An NOUN wurde ein auslautendes "e" angehängt.	Tilge bei NOUN, die auf "e" enden, das "e" und prüfe im Wörterbuch; wenn im Wörterbuch zu finden, typisiere.
CONTRACT	Ein nachfolgendes Pronomen "es" wurde in Form von "'s" an das voranstehende Wort kontrahiert.	Wenn ein Type auf "'s" endet, typisiere.
COMP_DASH	Mittels Bindestrich wurden mehrere Wörter zu einem Wort zusammengefügt, das nicht im Wörterbuch steht.	Wenn innerhalb eines Type ein Bindestrich vorkommt, typisiere.
COMP	Mehrere NOUN wurden zu einem Wort zusammengefügt, das nicht im Wörterbuch steht.	Wenn ein NOUN gleich viele oder mehr Zeichen aufweist als der Mittelwert der Zeichenanzahl aller Token im "all"-Set (8.8, gerundet 9), typisiere.
PART_ADJ	VERB wurde zu einem partizipialen ADJ abgeleitet, das nicht im Wörterbuch steht.	Wenn ein ADJ auf "end" endet, typisiere.
PREFIXED	Durch ein Präfix wurde ein Wort abgeleitet, das nicht im Wörterbuch steht.	Wenn ein Type mit einem Präfix aus einer vorgegeben Liste beginnt, entferne Präfix aus Lemma und prüfe im Wörterbuch; wenn im Wörterbuch zu finden, typisiere.

Fehlerkommentierung

Tab. 3: Relative Häufigkeit für die Typen potenzieller Fehler für die beiden pFail-Sets

	Lyrik	Lyrik	Lyrik	Lyrik	Lyrik	Prosa	Prosa	Prosa	Prosa	Prosa
	merged_Types	spacy_Token	stanza_Token	RNN_Token	Tree_Token	merged_Types	spacy_Token	stanza_Token	RNN_Token	Tree_Token
PUNC	0,454	0,271	0,151	0,565	0,451	0,576	1,543	0,527	1,758	0,818
SHORT	0,223	0,865	1,087	0,623	6,116	0,124	0,355	0,524	0,251	0,776
ORTH_SZ	0,120	0,186	0,180	0,194	0,183	0,133	0,195	0,191	0,192	0,194
ORTH_UPPER	0,627	1,152	1,093	0,905	0,846	0,020	0,102	0,059	0,061	0,054
ELISION_APO	2,083	2,424	2,693	2,614	2,349	0,314	0,167	0,191	0,207	0,274
ELISION_SIMPLE	2,574	5,185	5,256	4,998	4,854	0,978	1,276	1,338	1,178	1,170
ELISION_END	1,081	2,129	1,279	2,093	1,233	0,246	0,555	0,401	0,582	0,335
EPITHESES	0,285	0,380	0,359	0,368	0,354	0,128	0,139	0,132	0,133	0,124
CONTRACT	0,639	0,350	0,406	0,144	0,892	0,271	0,127	0,197	0,068	0,864
COMP_DASH	1,926	1,280	0,217	1,252	1,269	4,957	2,354	0,323	2,309	2,497
COMP	37,783	29,506	30,400	29,608	28,160	46,432	33,805	35,196	33,173	32,839
PART_ADJ	3,234	5,190	5,294	5,242	4,852	2,060	4,898	4,844	4,580	4,392
PREFIXED	2,302	2,052	2,079	2,071	1,939	3,643	3,196	3,106	2,951	2,773

53,33 % der Types im pFail-Set für Lyrik und 59,88 % der Types im pFail-Set für Prosa werden identifiziert (vgl. Tab. 3). Die

identifizierten Typen können zu Gruppen zusammengefasst werden: PUNC und SHORT sind überwiegend unterhalb der Wortebene anzusiedelnde Zeichen, meist Rauschen, das bei Lyrik und Prosa in vergleichbarem Umfang auftaucht. ORTH_SZ dokumentiert den ebenfalls bei Lyrik und Prosa vergleichbar ausgeprägten Effekt der *Historischen Orthographie*, die in unseren Korpora durch Modernisierungen bereits weitgehend abgefangen ist. Ein weiterer Normalisierungsschritt etwa mit dem DTA::CAB-Web-services⁷ könnte hier Abhilfe schaffen.

Die 10 weiteren Typen lassen sich zu drei Gruppen zusammenführen. COMP_DASH, COMP, PART_ADJ, PREFIXED versammeln *Kreative Lexik*, d.i. Wortbildungsmechanismen (Komposition, Derivation); hier handelt es sich häufig um *Out-of-Vocabulary*-Wörter, also um Pipelinefehler, nicht um Toolfehler. Bei der Lyrik lassen sich 45,25 % des „pFail“-Sets dieser Gruppe zuweisen, bei der Prosa 57,09 %. Eine erwartungsgemäß höhere Fehlerrate für Lyrik (0,62 %) als für Prosa (0,02 %) produziert die Pipeline bei ORTH_UPPER, mit dem – in Form der versinitialem Großschreibung – eine Eigenart *Lyrischer Typographie* identifiziert wird. Ebenfalls höher ist die Gruppe *Prosodische Deformation*, bestehend aus ELISION_APO, ELISION_SIMPLE, ELISION_END, EPITHESES, CONTRACT, die bei der Lyrik 6,62 % des pFail-Sets, bei der Prosa 1,93 % des pFail-Sets beschreibt. Da im Prosa-korpus umfangreich auch direkte Rede enthalten ist, liegt die Annahme nahe, dass die Deformationen hier tatsächlich auf die metrisch-bedingte Hinzufügung bzw. Tilgung von Vokalen zurückzuführen ist.

Methodenkritik und Ausblick

Zu resümieren, dass die spezifisch lyrische ‘Störung’ für die NLP-Tools insbesondere aus der *Prosodischen Deformation* des Wortmaterials und den Eigenarten der *Lyrischen Typographie* resultiert, wohingegen die *Kreative Lexik* (für die zudem präzisere Regeln notwendig wären) auch bei der Prozessierung literarischer Prosa erhebliche Schwierigkeiten bereitet, erweist sich nicht nur angesichts fehlender Signifikanztests als zu einfach. Denn darüber hinaus ist erstens unsere Pipeline noch zu grob gebaut: zu viele potenzielle Fehler sind, wie etwa bei der kreativen Lexik, faktisch keine Tool-Fehler, sondern Pipelinefehler. Zweitens vermag unsere regelbasierte Typologisierung mit 53,33 % nur etwa die Hälfte des pFail-Sets zu beschreiben.

Darin zeigen sich zwei Felder für Anschlussforschungen: Erstens wäre zu erproben, ob sich bessere Pipelines für die automatisierte NLP-Tool-Fehleridentifikation ohne Annotationsdaten konzipieren lassen, dafür wäre es hilfreich, die Pipeline auf einem kleinen Set an Gold-Standard-Annotationen zu evaluieren; zweitens könnte auf der Grundlage unserer Pipeline gegen die Baseline von 53,33 % das regelbasierte Typologisierungsverfahren optimiert werden. Die manuelle Annotation einer kleinen Sammlung von Gedichten mit Informationen zum Abweichungscharakter jedes einzelnen Wortes würde es ermöglichen, unsere Annahme, dass unser Verständnis der Devianz durch die Nutzbarmachung des Problems der Domänenadaptation von NLP-Tools operationalisiert werden kann, zu prüfen. So könnte sichergestellt werden, dass wir durch die Fehlertypisierung der NLP-Tools tatsächlich etwas über die Spezifik des Literarischen erfahren.

In jedem Fall haben wir mit dem vorliegenden *Tool Misuse*-Experiment noch nicht gut genug gelernt, die NLP-Tools ‘falsch’ zu verwenden.

Fußnoten

1. <https://textgridrep.org/>
2. Die Korpora sowie der Pipeline-Code sind hier zu finden: https://gitup.uni-potsdam.de/sluytergaeth/poetry_as_error
3. <https://www.projekt-gutenberg.org/>
4. <https://uni-tuebingen.de/en/faculties/faculty-of-humanities/departments/modern-languages/departments-of-linguistics/chairs/general-and-computational-linguistics/ressources/lexica/germanet/>
5. <https://www.dwds.de/>
6. Im Prosa-korpus ist die Quote der Types, die von “all” nach “pFail” übergeben werden, auffällig größer als im Lyrik-korpus, was u.a. an als NOUN getaggtten Eigennamen liegt. Eine abschließende Erklärung muss einer genaueren Analyse des Prosa-korpus vorbehalten bleiben.
7. <https://www.deutschestextarchiv.de/public/cab/>

Bibliographie

- Bamman, David** (2020): “LitBank: Born-Literary Natural Language Processing” [Preprint]. https://people.ischool.berkeley.edu/~dbamman/pubs/pdf/Bamman_DH_Debates_CompHum.pdf [letzter Zugriff 14. Juli 2021].
- Bers, Anna** (2020): “Nachwort” in: Anna Bers (eds.): *Frauen / Lyrik*. Gedichte in deutscher Sprache. Stuttgart: Reclam 793-851.
- Beutin, Wolfgang et al.** (2019): *Deutsche Literaturgeschichte*. Von den Anfängen bis zur Gegenwart. Berlin: 9. Aufl., Metzler.
- Braam, Hans** (2019): *Die berühmtesten deutschen Gedichte*. Auf der Grundlage von 300 Gedichtsammlungen. Stuttgart: 2. Aufl., Kröner.”
- Fricke, Harald** (1981): *Norm und Abweichung*. Eine Philosophie der Literatur. München: Beck.
- Hamp, Birgit / Feldweg, Helmut** (1997): “GermaNet - a Lexical-Semantic Net for German”, in: *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. Madrid: 9–15. <https://aclanthology.org/W97-0802.pdf> [letzter Zugriff 14. Juli 2021].
- Henrich, Verena / Hinrichs, Erhard** (2010): “GernEdiT - The GermaNet Editing Tool”, in: *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010)*, Valletta, Malta: 2228-2235. http://www.lrec-conf.org/proceedings/lrec2010/pdf/264_Paper.pdf [letzter Zugriff 14. Juli 2021].
- Herrmann, J. Berenike** (2018). “Praktische Tagger-Kritik. Zur Evaluation des PoS-Tagging des Deutschen Textarchivs”, in: *DHd2018: Kritik der digitalen Vernunft*. Book of Abstracts. Köln: 287-290. https://zenodo.org/record/3684897#.YO_x1W5CTOQ [letzter Zugriff 14. Juli 2021].
- Honnibal, Matthew / Montani, Ines / Van Landeghem, Sofie / Boyd Adriane** (2020): *spaCy: Industrial-strength Natural Language Processing in Python*. Zenodo. <https://doi.org/10.5281/zenodo.1212303> [letzter Zugriff 14. Juli 2021].
- Klein, Wolfgang / Geyken, Alexander** (2010): “Das ‘Digitale Wörterbuch der Deutschen Sprache DWDS’”, in: *Lexicographica* 26: 79–96.
- Müller-Zetzelmann, Eva** (2000): *Lyrik und Metalyrik*. Theorie einer Gattung und ihrer Selbstbespiegelung anhand von Beispielen aus der englisch- und deutschsprachigen Dichtkunst. Heidelberg: Winter.

Ortmann, Katrin / Roussel, Adam / Dipper, Stefanie (2019): “Evaluating Off-the-Shelf NLP Tools for German”, in: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*: 212-222. https://konvens.org/proceedings/2019/papers/KONVENS2019_paper_55.pdf [letzter Zugriff 14. Juli 2021].

Qi, Peng / Dozat, Timothy / Zhang, Yuhao / Manning, Christopher D. (2018): “Universal dependency parsing from scratch”, in: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brüssel: 160-170.

Rayson, Paul / Archer, Dawn / Baron, Alistair / Culpeper, Jonathan / Smith, Nicholas (2007): “Tagging the bard: Evaluating the accuracy of a modern POS tagger on Early Modern English corpora”, in: *Proceedings of Corpus Linguistics (CL2007)*. https://eprints.lancs.ac.uk/id/eprint/13011/1/192_Paper.pdf [letzter Zugriff 14. Juli 2021].

Schmid, Helmut (1994): “Probabilistic part-of speech Tagging using decision trees”, in: *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK: 154-162.

Schmid, Helmut (1995): “Improvements in Part-of-Speech Tagging with an Application to German”, in: *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland: 13-25. <https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf> [letzter Zugriff 14. Juli 2021].

Schmid, Helmut (2019): “Deep learning-based morphological taggers and lemmatizers for annotating historical texts”, in: *Proceedings of the 3rd international conference on digital access to textual cultural heritage*, Brüssel: 133-137.

Scheible, Silke / Whitt, Richard J. / Durrell, Martin / Bennett, Paul (2011): “A gold standard corpus of Early Modern German” in: *Proceedings of the 5th Linguistic Annotation Workshop*: 124–128. <https://dl.acm.org/doi/abs/10.5555/2018966.2018981> [letzter Zugriff 14. Juli 2021].

Zymner, Rüdiger (2019): “Begriffe der Lyrikologie” in: Hildebrandt, Claudia et al. (eds.) *Lyrisches Ich, Textsubjekt, Sprecher?* (= Grundfragen der Lyrikologie, Bd. 1). Berlin: De Gruyter 25-50.