Identifikation von Vorkommensformen der Lemmata in Quellenzitaten frühneuhochdeutscher Lexikoneinträge

Stefanie Dipper
Sprachwissenschaftliches Institut
Fakultät für Philologie
Ruhr-Universität Bochum
stefanie.dipper@rub.de

Jan Christian Schaffert
Georg-August-Universität Göttingen &
Akademie der Wissenschaften zu Göttingen
jan.schaffert@phil.uni-goettingen.de

Abstract

In dieser Arbeit werden zwei Ansätze vorgestellt, die die Quellenzitate innerhalb eines Wörterbucheintrags im Frühneuhochdeutschen Wörterbuch analysieren und darin die Vorkommensform identifizieren, d. h. die Wortform, die dem Lemma dieses Eintrags entspricht und als historische Schreibform in verschiedenen Schreibvarianten vorliegt. Die Evaluation zeigt, dass schon auf Basis kleiner Trainingsdaten brauchbare Ergebnisse erzielt werden können.

1 Einleitung

Wörterbücher erschließen Sprachen, deren Dialekte, Sprachstufen und Fachwortschätze über die strukturierte Präsentation sprachbezogener Informationen. Die Ausdifferenzierung kann dabei sehr unterschiedlich sein und zeigt sich schon in den Namen der Wörterbücher (Deutsches Rechtswörterbuch, Wörterbuch der schweizerdeutschen Sprache, Wörterbuch der deutschen Pflanzennamen). Eine besondere Position nehmen die allgemeinen Wörterbücher ein, die den Gesamtwortschatz einer Sprache diachron oder synchron erfassen. Wird eine historische Sprachstufe bearbeitet, kommt den Werken zudem eine kulturpädagogische Funktion zu, da sie die diachronen Unterschiede zu der jeweiligen Standardsprache herausarbeiten müssen, um entsprechende sprachbezogene Informationen adäquat zu vermitteln (Reichmann, 1986).

Natürlich wurden während des Digital Turns neben den historischen Quellen auch die Wörterbücher digitalisiert, sodass deren Informationsangebot nun überall abrufbar, durchsuchbar und vielfältig auswertbar ist. Eine Verknüpfung der Quellen mit den Wörterbüchern fand jedoch nicht statt. Die historischen Quellen bieten daher aktuell nur rudimentäre Möglichkeiten der Nachnutzbarkeit (Klaffki et al., 2018). So mangelt es ihnen an jener semantischen Erschließungstiefe, die über ein passendes Wörterbuch erreicht werden könnte und maßgeblich zum Verständnis beitragen würde.

Im nachfolgenden Beitrag stellen wir auf Basis des Frühneuhochdeutschen Wörterbuches (im Folgenden FWB) zwei Ansätze vor, die dies möglich machen sollen, indem sie durch die Lemmatisierung frühneuhochdeutscher Wörter die Basis einer Semantisierung schaffen.

Der eine Ansatz wendet ein existierendes System zur Normalisierung historischer Schreibungen an, der andere nutzt ein künstliches neuronales Netzwerk.[1] Ausgangspunkt ist die Identifikation der Lemmata und deren Wortbildungen in den Quellenzitaten des FWBs. Langfristiges Ziel ist die möglichst umfassende automatische Lemmatisierung digitaler frühneuhochdeutscher Texte.

Der Artikel ist wie folgt aufgebaut: Zunächst stellen wir die Daten des FWBs vor (Kap. 2). Kap. 3 erklärt, wie unser genereller Ansatz aussieht. In Kap. 4 und 5 beschreiben wir die beiden Systeme zur Identifikation der Vorkommensform. Kap. 6 enthält die Resultate, gefolgt von einem Ausblick in Kap. 7.

2 Das Frühneuhochdeutsche Wörterbuch (FWB)

Das FWB ist ein semantisches Bedeutungswörterbuch mit kulturwissenschaftlichem Schwerpunkt, dessen Ziel es ist, den Gesamtwortschatz des Frühneuhochdeutschen synchron in seiner Heterogenität zu präsentieren (Reichmann, 1986). Um dessen Varietätenspektrum bestmöglich zu erfassen, bildet das FWB die drei wichtigsten frühneuhochdeutschen Heterogenitätsdimensionen Zeit

[1] Unser Dank gilt insbesondere Herrn Dr. Matthias Schütze, der das FWB seit vielen Jahren technisch begleitet und in diesem Zusammenhang auch das künstliche neuronale Netz entwickelt hat.

(1350 bis 1650), Raum (Thüringisch, Elsässisch, Alemannisch, etc.) und Textsorte (erbauliche, literarische, rechtsgeschichtliche, etc. Texte) möglichst ungewichtet in seinen Quellen und sprachbezogenen Informationen ab. Pro Lemma, bei polysemen Lemmata pro Einzelsemantik, bietet das FWB eine Vielzahl qualitativ hochwertiger, heuristisch kompetent überprüfter, semantischer und pragmatischer Informationen und belegt diese mit Zitaten.

Da das Frühneuhochdeutsche weder eine normativ geregelte Orthographie noch eine überdachende Leitvarietät aufweist, belegt das FWB pro Lemma zudem eine z. T. erhebliche Anzahl von Vorkommensformen (kurz: VKF) pro Lemma. Da diese seit 2017 manuell ausgezeichnet werden, ergibt sich ein unschätzbares Potenzial: Aktuell werden 13.178 VKF eindeutig 4.674 Lemmata zugeordnet. Dieses Verhältnis deutet die Problematiken der Lemmatisierungsansätze jener VKF an, die nicht ausgezeichnet sind.

Exemplarisch ist das Lemma *abenteuer*, das in den Quellenzitaten u. a. in folgenden VKF belegt ist: *aventevre*, *auffentür*, *abenteür*, *aventúre*, *abentewr*, *abentur*, *ofentúre*, *obentewer*, *aubentúr*, *aubenteur*, usw. (s. Abbildung 8 im Appendix mit einem Ausschnitt des FWB-Eintrags zu diesem Lemma). Wie kann in allen diesen Vorkommensformen (insgesamt 36 unterscheidbare) automatisch und möglichst eindeutig das Lemma erkannt werden?

Die in diesem Beitrag genutzten Daten des FWB stehen online z. T. frei zur Verfügung oder werden in den kommenden Jahren freigeschaltet.[2]

3 Identifikation von Vorkommensformen durch Lemmatisierung

In diesem Beitrag soll es also noch nicht um die Lemmatisierung beliebiger Texte des Frühneuhochdeutschen gehen, sondern zunächst um eine einfachere Aufgabe: Gegeben ein Lemma wie *abenteuer*, identifiziere die zugehörige Vorkommensform (VKF) innerhalb der Quellenzitate. (1) zeigt ein Beispiel für ein Quellenzitat für dieses Lemma aus einem nordoberdeutschen Text. Das Lemma ist in standardisierter Form vorgegeben, während die VKF eine flektierte Wortform sein kann, die zudem in der historischen Originalschreibung vorliegt. Ziel ist es also, in (1) die VKF *obentewern* zu identifizieren.

(1) *das die frembden in [. . .] wirtshewser geen mit iren obentewern.*

Wir fassen die Aufgabe als Lemmatisierungsaufgabe auf: Gegeben ein Kandidat für eine VKF, lässt sich dieser Kandidat auf das vorgegebene Lemma lemmatisieren? Dabei gehen beide Ansätze so vor, dass sie sämtliche historischen Worterformen w_i innerhalb eines Belegs mit dem vorgegebenen Lemma l paaren: $< w_i, l >$ und für jedes Paar überprüfen, ob l das Lemma von w_i sein könnte. Im Beispiel (1) wären das also die Paare *<das, abenteuer>*, *<die, abenteuer>*, *<frembden, abenteuer>* etc.

Ein möglicher Ansatz wäre es, für diese Aufgabe einen vorhandenen Lemmatisierer zu nutzen. Das ist allerdings aus verschiedenen Gründen nicht ohne Weiteres möglich:

Viele der Ziel-Lemmata aus dem FWB haben keine moderne Entsprechung, z. B. lauten die ersten zehn Lemmata einer Zufallsauswahl *sünde, *quatembergeld, *entspanen, erzeigen, *erbholde, *abtilgen, abschlagen, *äfern, streuen, abtun*[3] – für fünf davon (mit Stern markiert) gibt es keinen Eintrag in einem Standardwörterbuch wie dem Duden[4]. Damit lassen sich moderne Lemmatisierer nicht ohne Weiteres sinnvoll auf diese Daten anwenden, da die Zahl der ungesehenen Lemmata ungewöhnlich hoch ist. Auch lassen sich vorhandene Korpora wie z. B. das Anselm-Korpus[5], das RIDGES-Korpus[6] oder das Referenz-Korpus Frühneuhochdeutsch[7] nicht als Trainingsdaten verwenden, da diese moderne Lemmata verwenden.

Außerdem basieren viele moderne Lemmatisierer auf Wortart-Information (und integrieren gegebenenfalls einen entsprechenden Tagger), so z. B. Liebeck and Conrad (2015); Konrad (2019). Für unsere Daten liegen aber keine entsprechenden Wortart-Annotationen vor.

Ein weiteres Problem ist, dass die VKF-Kandidaten nicht in einer standardisierten Form vorliegen, sondern stark variieren können. Daher lässt sich beispielsweise der Ansatz von Wartena

[2] https://fwb-online.de/ (letzter Zugriff: 5.5.2021)

[3] Die Daten stammen aus dem Mittleren Ostoberdeutsch (moobd.), vgl. Abschnitt 4.

[4] https://www.duden.de/

[5] https://www.linguistics.rub.de/comphist/projects/anselm/

[6] https://www.linguistik.hu-berlin.de/de/institut/professuren/korpuslinguistik/forschung/ridges-projekt

[7] https://www.linguistics.ruhr-uni-bochum.de/ref/

(2019) nicht umsetzen, der von markierten Morphemgrenzen innerhalb der Trainingsdaten ausgeht.

Schließlich liegen in unserem Szenario nur wenig Trainingsdaten vor: Im Ansatz mit Norma (s. Abschnitt 4) stehen jeweils nur 500 Wörter als Trainingsdaten zur Verfügung – eine der Stärken von Norma ist es, mit solch geringen Datenmengen bereits gute Ergebnisse zu liefern. Dem künstlichen neuronalen Netz (s. Abschnitt 5) stehen mit ca. 170.000 Wörtern zwar mehr, aber ebenfalls vergleichsweise wenige Trainingsdaten zur Verfügung. Zum Vergleich: das neuronale Modell von Schmid (2019) nutzt zwei Millionen Wörter zum Trainieren.

4 Identifikation durch Norma

Im ersten Ansatz verwenden wir ein existierendes System, das frei verfügbar ist: Norma[8] (Bollmann 2012). Norma wurde entwickelt für eine automatische Normalisierung von (historischen) Schreibvarianten und wird auf Trainingspaaren der Form <original, normalisiert> trainiert. Norma integriert drei verschiedene Arten von Lernkomponenten (für Details, s. Bollmann (2012)):

1. Mapper: Diese Komponente stellt eine Liste der Paare <original, normalisiert> bereit, die in den Trainingsdaten gesehen wurden. Mapper kann also nur bekannte Schreibungen per Lexikon Lookup normalisieren.

2. RuleBased: Diese Komponente lernt kontextsensitive Ersetzungsregeln, die einen Buchstaben bzw. eine Buchstabensequenz durch eine andere ersetzen. Die Regeln sind nach Frequenz ihrer Anwendung geordnet.

3. WLD (weighted Levenshtein distance): Diese Komponente wendet gewichtete LD an, um Buchstaben-Ngramme aufeinander abzubilden.[9]

RuleBased und WLD benötigen außerdem ein Lexikon mit normalisierten Schreibungen, gegen das die generierten Kandidaten abgeglichen werden. Bei der Normalisierung wendet Norma die drei Komponenten der Reihe nach an. Sobald eine Komponente eine normalisierte Form generiert, wird abgebrochen. Norma generiert allerdings nur Kandidaten, die sich nicht mehr als eine gewisse Distanz vom Ausgangswort unterscheiden. Gibt es keinen solchen Kandidaten, bleibt der Output leer.

Norma wurde für die Normalisierung flektierter Wortformen entwickelt. In einer ersten Evaluation testeten wir daher, ob sich Norma prinzipiell auch für die Lemmatisierung eignet. Eine Crossvalidierung ergab Durchschnittswerte zwischen 56,8-69,8% Genauigkeit pro Teilkorpus, was Norma für die (wesentlich leichtere) Aufgabe der VKF-Identifikation als mögliches Tool erscheinen lässt. (Details zu dieser Evaluation im Appendix.)

Norma als Tool für die VKF-Identifikation Für die VKF-Identifikation lemmatisiert Norma zunächst jeden VKF-Kandidaten aus einem Beleg. Anschließend werden die generierten Lemmata mit dem vorgegebenen Lemma abgeglichen und die VKF wird ausgewählt, deren Lemma mit dem vorgegebenen übereinstimmt. Gegebenenfalls kann auch kein oder mehrere Kandidaten zum vorgegebenen Lemma lemmatisiert werden.

Für diese Anwendung trainieren wir Norma auf Paaren der Form <historische Wortform, FWB-Lemma>.[10] Entsprechend besteht das Lexikon zum Abgleich aus Lemmata. Wir nutzen zwei unterschiedliche Lexika für den Abgleich:

1. Norma-full: das Lexikon besteht aus einer Liste von rund 78.000 Lemmata des FWB (“full lexicon”)

2. Norma-small: das Lexikon besteht nur aus dem vorgegebenen Lemma (“small lexicon”)

Das Szenario Norma-full entspricht dem üblichen Vorgehen und könnte beispielsweise bei der Lemmatisierung von Freitext (ohne vorgegebenes Lemma) Anwendung finden. Das Szenario Norma-small ist auf die aktuelle Aufgabenstellung zugeschnitten: Da das Ziel-Lemma schon bekannt ist, kann Normas Hypothesenraum extrem auf genau diese Form eingeschränkt werden. Das hat folgende Konsequenzen:

Norma-full generiert die Kandidaten sehr viel unrestriktiver als Norma-small. Daher kommt es

[8] https://github.com/comphist/norma (letzter Zugriff: 5.5.2021)

[9] Norma markiert nur bei der Komponente RuleBased die Wortgrenzen explizit (mit “#”). Die Wortgrenzen stellen eine wichtige Information für die Lemmatisierung dar: Am Wortende müssen andere Ersetzungen gelernt werden als in der Wortmitte. Daher fügen wir “#” für WLD am Wortanfang und -ende an.

[10] Sonderzeichen in den Wortformen innerhalb der Belege wie Satzzeichen (! ? , etc.) oder Anführungszeichen und Klammern werden gelöscht.

hier öfters vor, dass Norma-full bei keiner der Input-Formen die vorgegebene Lemma-Form generiert. D. h. Norma-full hat eine geringere Abdeckung als Norma-small.

Im Fall von sehr kurzen Wortformen und Lemmata kann Norma-small (zu) viele der Input-Wortformen auf das vorgegebene Lemma abbilden, da alle innerhalb der Abbruch-Schwelle liegen. (2) zeigt ein solches Beispiel. Das vorgegebene Lemma ist *öl* und der dazugehörige Beleg enthält viele sehr kurze Wortformen. (3) zeigt die Liste der Wortformen aus (2), die Norma-small auf das Lemma *öl* abbilden konnte. Die Liste ist nach einem Score geordnet, den Norma ausgibt. *öl* (der erste VKF-Kandidat) ist demnach der "beste" Kandidat, den Norma generiert (was hier auch die korrekte Form ist). In der Evaluation (Kap. 6) wird jeweils nur die erste Form berücksichtigt.

(2) *chümpt dann ain gast mit öl vnd wil zemarcht damit sten vnd gibet es von hant hin, als oft er ain lagel öls auf tuet, so geit er ain pfunt öls, als oft er die verchauffet.*

(3) *öl, als, als, wil, oft, oft, lagel, von, sten, es, er, er, er, hin, ain, ain, ain, so, die, geit, mit, vnd, vnd, auf, pfunt, hant, gast, dann, tuet*

Wir führen eine sechsfache Crossvalidierung durch und trainieren Norma auf jeweils 500 Paaren aus drei verschiedenen Sprachräumen (Nordoberdeutsch/nobd, Mittleres Ostoberdeutsch/moobd, Elsässisch/els) und evaluieren auf jeweils 100 Paaren.

5 Identifikation durch ein künstliches neuronales Netz

Im zweiten Ansatz verwenden wir ein künstliches neuronales Netz, um den Herausforderungen, die aus den FWB-Daten erwachsen können, zu begegnen. Neben ihrem geringem Umfang sind die Daten auch unvollständig und sehr spezifisch: derzeit liegen Trainingsdaten nur für die e-, q-, r- und st-Strecken vor. In unserer Evaluation zeigte es sich allerdings, dass hieraus keine größeren Nachteile entstehen: Das über die r-Strecke trainierte Netz generalisiert gut und ergibt für die anderen Strecken F-Scores, die mit denen der Trainingsdaten vergleichbar oder sogar besser sind (vgl. Tabelle 4). Dies ist von besonderer Bedeutung, da das FWB aktuell erst zu ca. 75% abgeschlossen ist und das Netz in Zukunft beliebige Texte lemmatisieren soll.

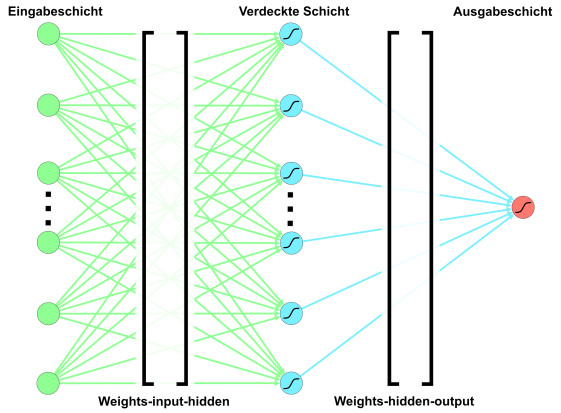

Abbildung 1: Schematische Darstellung der Topologie des Netzes

Da über das FWB nur eingeschränkte Trainingsdaten zu Verfügung stehen, wurden einige manuell erstellte Normalisierungsregeln in das Netz aufgenommen, um spezielle Fälle, die für einen relativ großen Prozentsatz von Fehlern verantwortlich sind, schnell zu entschärfen (Ernst-Gerlach and Fuhr, 2006; Pilz et al., 2007). Dieses Vorgehen erwies sich als erfolgreich, da schon wenige manuelle Regeln (vgl. Regeln 12–15 in Tabelle 1) den F-Score des Netzes signifikant anheben. Aktuell handelt es sich nur um Normalisierungsregeln im Sinne der Lemmazeichengestalt (Reichmann, 1986), in Zukunft sollen auch sprachraumspezifische Regeln implementiert werden.[11]

Die Topologie ist basal und empirisch unterstützt. Das Netz entspricht einem typischen dreischichtigen feedforward-Netz mit einfacher verborgener Schicht, angewendeter Sigmoid-Funktion und zwei Gewichtsmatrizes (Goodfellow et al., 2018), vgl. Abb. 1. Das Netz selbst ist in Python programmiert, orientiert sich in seinem Framework an Rashid (2017) sowie Steinwendner and Schwaiger (2019) und lernt gemäß der Aufgabenstellung überwacht und parametrisch. Die Matrizenmultiplikation wird mit NumPy[12] realisiert.

Die Eingabe- und verdeckte Schicht sind gleich mächtig, da die Verringerung von verdeckten Neu-

[11] Ein vergleichbares Vorgehen wurde für des Referenzkorpus Althochdeutsch genutzt, für dessen Lemmatisierer über 700 Regeln manuell aufgestellt wurden, die pro Zeit und Raum gewissen Lautständen mehr oder weniger statistische Bedeutsamkeit zuweisen und somit die Metadaten eines Textes produktiv in die Analyse einfließen lassen (Mittmann, 2016). Da das Frühneuhochdeutsche jedoch wesentlich umfassender und somit zwangsläufig diverser überliefert ist, gestaltet sich ein vergleichbarer Ansatz als unlösbar komplexe Aufgabe. In Zukunft sollen entsprechende Regeln daher erlernt werden.

[12] https://numpy.org/ (letzter Zugriff: 02.06.2021)

ronen zu erheblichen Schwankungen der Fehlerquote während des Trainings geführt hat. Die Ausgabeschicht besteht nur aus einem Neuron, da das Netz anhand des Scores nur eine Voraussage darüber trifft, ob ein Wort als Lemma erkannt wird oder nicht. Je höher der Score, desto sicherer wird das betreffende Lemma erkannt. Hierbei wird analog zu Norma mit Paaren der Form $< w_i, l >$ gearbeitet.

Die Paarung von historischen Wortformen mit den Lemmata des FWB $< w_i, l >$ ergeben die Daten, die als Bewertungsvektor formalisiert werden. Dieser Vektor bildet die Eingabeschicht, deren Werte gewichtet und auf der verdeckten Schicht propagiert werden. Nach einer weiteren Gewichtung gibt das Netz auf der Ausgabeschicht einen Score an, der determiniert, ob lemmatisiert wird oder nicht. Diese Lemmatisierung wird anschließend anhand der bereits ausgezeichneten Strecken evaluiert und das Netz so trainiert. Diese Aspekte werden im Folgenden genauer erläutert.

Bewertungsvektoren Da das FWB und die meisten maschinenlesbaren historischen Quellen nicht getaggt sind, kann nur auf jene Informationen zurückgegriffen werden, die sich aus dem Vergleich der VKF mit den Lemmata des FWB ergeben. Darüber hinaus werden Metadaten zum jeweiligen Sprachraum berücksichtigt, da diese genutzt werden, um sprachraumspezifische Normalisierungen in das Netz einfließen zu lassen.

Um z. B. die VKF *Refftrager* im Quellenzitat (4) als das Lemma *refträger* zu identifizieren, werden insgesamt neun Bewertungsvektoren für sämtliche Paarungen <*secht, refträger*>, [...], <*Refftrager, refträger*> erstellt. Der Bewertungsvektor für die VKF *Refftrager* entspricht der vierten Spalte von Tabelle 1.

(4) *secht recht wie ein Hundsschlager | Oder ein alter Refftrager*

Alle Informationen müssen für das Netz in numerische Werte transformiert werden, damit sie als Features der Eingabeneuronen dienen können. Die Features und deren Werte ergeben sich aus Tests-Trainings. Es wurden stets die Werte gewählt, für die sich die beste Entwicklung des Fehlerquotienten ergab (vgl. hierzu Abb. 2). Die Anhebung des F-Scores wurde erst ansatzweise durch die Implementierung der Regeln 12–15 angegangen.

Die Länge von Lemma und Wortform ergibt sich als Verhältnis zur maximalen Wortlänge von

	Merkmal	Erläuterung	Wert
1	LLemma	Länge Lemma	0,36
2	LOriginal	Länge Originalschreibung	0,4
3	Durchsn.L	Differenz Längen	0,05
4	Subst.	Substantiv	0,5
5	Adj.	Adjektiv/Adverb	0
6	Verb	Verb	0
7	Unbekannt	Unbekannt	0
8	JW	Jaro-Winkler-Distanz	0,93
9	phon	phonetische Distanz	1
10	JW-norm	JW-Distanz mit Normierung	0,984
11	phon-norm	ph-Distanz mit Normierung	1
12	JW-fwb-allg	JW-Distanz mit FWB-Norm.	0,93
13	JW-kw-qu	JW-Distanz kw-qu	0,93
14	JW-ai-ei	JW-Distanz ai-ei	0,93
15	JW-ich-ig	JW-Distanz auslautendes ich-ig	0,93
16	nrdnieders.	Niedersächsisch	0
...			
39	orfrk.	Ostfränkisch	1
...			
45	balt.	Baltisch	0

Tabelle 1: Bewertungsmatrix für die Wortform *Refftrager*

25 Buchstaben. Die Jaro-Winkler-Distanz wird entsprechend (Winkler, 1990), die phonetische Distanz gemäß der Kölner Phonetik (Postel, 1969) berechnet. Alle weiteren Distanzen ergeben sich aus den Normierungen, die Nichtbuchstaben aus dem Lemma entfernen, Diakritika und Ligaturen auflösen und die Originalschreibung gemäß der Richtlinien für die FWB-Lemmazeichengestalt normalisieren (Reichmann, 1986). Insofern folgt das Netz dem etablierten Ansatz, Vorkommensformen zu normalisieren, ermöglicht jedoch die Inklusion von Metadataden und händisch erstellten Regeln, die sich für vergleichbare Ansätze als hilfreich erwiesen haben.

Da für die Zukunft abzusehen ist, dass Informationen zu den Wortarten zwar hilfreich, aber nicht nutzbar sein werden, ist geplant, nach den sprachraumspezifischen Regelsätzen auch spezielle FLexions- und Deklinationsregeln zu implementieren, die generelle Prinzipien erfassen, jedoch nicht auf Informationen zur Wortart angewiesen sind. Der Defaultwert für die Wortarten ist 0, das Feature für die entsprechende Wortart (unter Sonstige subsummiert das FWB Artikel, Interjektionen, etc.) ist 0,5.

Insgesamt deckt das FWB vom Niederpreußischen bis zum Alemannischen 31 Sprachräume ab. Der Default-Wert der entsprechenden Neuronen ist wiederum 0. Je nachdem welchem Sprachraum das jeweilige Wort zugeordnet ist, müssen ggf. mehrere Neuronen aktiviert werden, da sowohl über-

als auch untergeordnete Sprachräume existieren. *Refftrager* ist z. B. in einer Nürnberger, d. h. einer oberfränkischen Quelle belegt (für Städte wird immer der Sprachraum gewählt, in dem sie liegen). Da sich der oberfränkische Sprachraum aus keinen untergeordneten zusammensetzt, wird nur dessen Neuron aktiviert und der Wert 1 eingetragen. Bei einer rheinfränkischen Quelle müssten hingegen mehrere Sprachräume berücksichtigt werden, weil dieser Sprachraum aus dem hessischen und pfälzischen besteht. Für solche Fälle wird der Wert nach der auf dem Kehrwert der Gebietszahl basierenden Formel Feature = 0,3 + 0,5/x für x = Anzahl aktivierte Sprachräume berechnet.

Lemmatisierung Das Netz identifiziert ein Wort als Lemma über den Score. Ist dieser größer als der aktuell noch willkürlich gewählte Wert von 0,58, wird lemmatisiert. Zentrales Element der Berechnung des Scores sind neben den Werten des Bewertungsvektors zwei Gewichtsmatrizes, die zwischen Eingabeschicht und verdeckter (Weights-input-hidden) sowie verdeckter und Ausgabeschicht (Weights-hidden-output) positioniert sind. Die Values der Eingabeschicht werden wie üblich per Matrixmultiplikation propagiert und auf der verdeckten Ebene mit einer Sigmoidfunktion auf das Intervall [0, 1] beschränkt und so der Rechenaufwand zu minimiert, ohne Präzision einzubüßen. Derart können auch verdeckte Neuronen deaktiviert werden und kann das Netz verschiedene Eingaben korrelieren und nichtlinear arbeiten.

Training Das Training des Netzes erfolgt auf ursprünglich randomisierten Gewichtsmatrizes entsprechend des hot cold learning. Ziel ist eine gleichbleibende, möglichst geringe Fehlerquote. Da unser Netz simpel aufgebaut ist, können wir mit einer sehr geringen Lernrate arbeiten und so das Minimum der Fehlerquote genau bestimmen, was zu einem robusten Netz führen sollte.

Die Kurve in Abb. 2 beschreibt die Entwicklung des Fehlerquotienten beim Training über der r-Strecke, die in 173.379 Wörtern 11.187 Vorkommensformen von 1.819 Lemmata enthält. Der lokale Anstieg des Fehlerquotieten weist auf eine zu hohe Lernrate hin, die den statistischen Gradientenabstieg in zu großen Schritten über das Minimum der Fehler-Gewichts-Kurven hinausschießen lässt. Es ist zu erkennen, dass noch ca. 2000 Fehler existieren.

Im Folgenden einige beispielhafte Analysen: Im

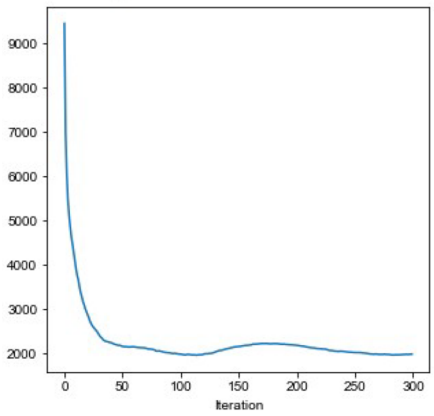

Abbildung 2: Lernkurve des Netzes

Quellenzitat (5) hat die Vorkommensform *abethewer* einen Score von 0,79, wird also korrekt lemmatisiert. Wir prüfen jedoch auch, ob zwei auf einander folgende Wörter einem getrennt geschriebenen Lemma entsprechen, damit erhalten wir für die aufeinander folgenden Wörter *aber einer* einen Score von 0,67 und somit einen false positive.

Im Quellenzitat (6) hingegen ermittelt das Netz für *avonture* einen Score, der kleiner als 0,58 ist. Die Vorkommensform wird also nicht erkannt, es liegt ein false negative vor.

(5) *sucht aber einer awßflucht, so sten einer seyn abethewer*

(6) *also uns die avonture und ouch daz buch noch seit*

6 Resultate

6.1 Ergebnisse von Norma

Als Baseline verwenden wir ein einfaches System, das jeweils die vorliegende Wortform als Lemma vorhersagt. Die Baseline entspricht damit dem Anteil der Stichworte, die formal mit dem Lemma übereinstimmen. In den drei Korpora liegt die Baseline zwischen 15,0–18,2% Genauigkeit.

Insgesamt ergeben sich durchschnittliche Genauigkeiten von 84,3% mit Norma-small und 60,8% mit Norma-full. Der Großteil der Fehlerrate von Norma-full ergibt sich aus den Fällen, bei denen Norma kein passendes Lemma generiert. Schaut man sich die Genauigkeit bei den generierten Lemmata allein an (d. h. die Precision), so ergibt sich bei Norma-small 88,6% und bei Norma-full 91,9%.

Von den Korpora ist nobd das schwierigste, mit Genauigkeiten von 82,5% (Norma-small) und

	–	Mapper	Rules	WLD
# Lemmata				
Norma-small	88	417	276	1.019
Norma-full	609	422	256	513
Precision				
Norma-small	0	81,3	97,8	89,1
Norma-full	0	80,3	98,0	98,2

Tabelle 2: Verteilung der erzeugten Lemmata über die Normalisierer sowie die jeweilige Genauigkeit (Durchschnitt in Prozent)

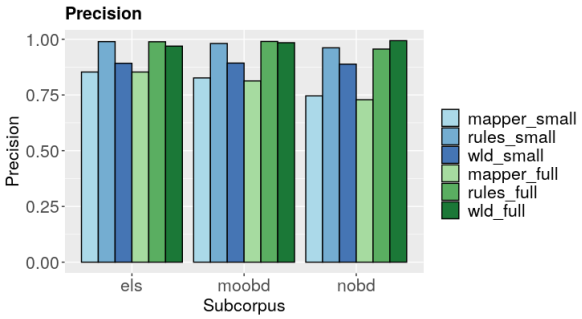

Abbildung 3: Precision der einzelnen Normalisierer

52,8% (Norma-full), gegenüber rund 85% bzw. 65% bei den beiden anderen Korpora.

Tabelle 2 zeigt, von welchen Normalisierern die erzeugten Lemmata in den beiden Szenarien stammen. "–" sind die Fälle, in denen Norma keinen Kandidaten generiert. Man sieht deutlich, dass ein Großteil der WLD-Lemmata, die im Szenario Norma-small dank des minimalen Ziellexikons erzeugt werden, im Szenario Norma-full nicht generiert werden und zu einer großen Anzahl von unanalysierten Fällen führen (33,8%) . Gleichzeitig zeigt es sich, dass die Precision von WLD bei Norma-small deutlich abfällt gegenüber Norma-full (89,1% vs. 98,2%). D. h. von den rund 500 Lemmata, die Norma-small zusätzlich generiert, sind nur rund 400 korrekt.

In Tabelle 2 fällt zudem auf, dass der Mapper in beiden Szenarien deutlich abfällt gegenüber den anderen Normalisierern. Das ist zunächst überraschend, da der Mapper nur bei bereits bekannten Paaren aktiv wird. Die Fehleranalyse unten zeigt, dass die schlechte Performanz zu großen Teilen auf Eigenschaften der Evaluationsdaten zurückgeführt werden kann.

Abb. 3 zeigt die Precision der einzelnen Normalisierer. Rules schneidet hier am besten ab (mit Werten von 95.6–99.0%). Im Szenario Norma-full liefert WLD vergleichbar gute Ergebnisse (96.9–99.3%).

Fehleranalyse Wie schon erwähnt, machen die fehlenden Lemmatisierungen einen wesentlichen Teil der Fehlerrate aus: bei Norma-small sind es 31,1%, bei Norma-full sogar 86,3%.

Kritischer sind allerdings die Fälle, in denen Norma eine VKF identifiziert, diese aber nicht die richtige ist (false positives). Das ist in 195 (Norma-small) bzw. 97 (Norma-Full) Fällen der Fall. Eine manuelle Analyse dieser Fälle ergab:

Bei Norma-Full sind nur zwei dieser Fälle tatsächlich echte (wenn auch nachvollziehbare) Fehler. Dabei identifizierte Norma als VKF des Lemmas *osterwind* einmal *westerwint* und einmal *oberwind* statt *oster(wint)* im Quellenzitat in (7). In allen anderen Fällen wurde als VKF entweder ein (korrekter) Bestandteil eines komplexen Wortes erkannt oder umgekehrt, oder es kamen im Quellenzitat mehrere mögliche VKFs vor und Norma wählte eine andere VKF als in den Testdaten vorgegeben. Einige Beispiele werden in Tabelle 3 gezeigt.

(7) *Oster- und westerwint, den man ober und nider nent, wäen dick und oft und gegen denen pflegt man nit zu pauen; der oberwind pringt gern regen und ungewitter.*

Von Norma-Small wurden die ersten 100 Fälle manuell analysiert. Davon waren 75 eigentlich korrekt. Dabei handelte es sich z. T. um die gleichen Fälle wie bei Norma-Full. Zusätzlich kommt es hier zu echten Fehlern wie in der unteren Hälfte von Tabelle 3 illustriert. Z. B. wird für das Lemma *erbe* als VKF *brief* identifiziert. Der Grund dafür ist, dass der Mapper kein Ziellexikon nutzt und als erste Komponente die (eigentlich gesuchte) Wortform *erben* auf das Lemma *erben* lemmatisiert hat, so dass die weiteren Normalisierer gar nicht mehr auf diese Wortform angewendet wurden. Rules und WLD hätten sonst die VKF korrekt identifiziert. Dasselbe passiert im Fall von *straff*, das der Mapper auf *strafe* statt auf *strafen* lemmatisiert. Es wäre hier also zu überlegen, den Output des Mappers zusätzlich mit dem Ziellexikon abzugleichen.

6.2 Ergebnisse des Netzes

Schon jetzt ergibt das noch unfertige künstliche neuronale Netz vielversprechende Ergebnisse, s. Tabelle 4. Auf Basis des aktuellen Trainings erhalten wir einen durchschnittlichen F-Score von 0,931. Da die einzelnen F-Scores über die analysierten

Gold-Lemma	Gold-VKF	System-VKF	System
abschlagen	ab	schlueg	Nfl/Nsm
anheben	hueb	an	Nfl/Nsm
straus	straussen	strauß	Nfl/Nsm
entblössen	entblotzet	entblotzest	Nfl/Nsm
erbe	erben	brief	Nsm
strafen	straff	spricht	Nsm

Tabelle 3: False Positives von Norma-full (Nfl) und Norma-small (Nsm)

	r-Strecke	e-Str.	q-Str.	st-Str.
TN	153.687	10.876	71.132	67.953
TP	9.806	646	4.648	4.326
FN	715	52	310	285
FP	800	33	315	436
Precision %	92,5	95,1	93,7	90,08
Recall %	93,2	92,55	93,75	93,12
F-Score	0,928	0,938	0.937	0.923

Tabelle 4: Ergebnisse für das Netz: r-Strecken: Trainingsdaten; Rest: Testdaten. (TN: True Negatives, TP: True Positives, FN: False Negatives, FP: False Positives)

Strecken hinweg recht konstant sind, können wir davon ausgehen, dass das Netz weder über- noch unterangepasst ist. Um die Treffsicherheit des Netz zu verbessern, sollen in Zukunft gemischtere Trainingsdaten aus allen manuell getaggten Strecken erstellt werden. Zudem scheint es sinnvoll, den Score, über den lemmatisiert wird, nach oben zu korrigieren, das Netz so kritischer zu gestalten und false positives auszuschließen. Das damit einhergehende vermehrte Auftreten von false negatives ist zu verkraften, da diese, wenn das Netzwerk weiter trainiert wird, zurückgehen sollten.

Die folgenden Analysen basieren auf der a- und b-Strecke.

Hinsichtlich der Wortarten entfällt der Großteil der Fehler auf flektierte Verben, vgl. Abb. 4. Ein besonderes Problem stellen Partikelverben da, die getrennt geschrieben nur dann lemmatisiert werden können, wenn die betreffenden Teilstücke direkt aufeinander folgen. Adjektive/Adverbien und Sonstige werden durchschnittlich bzw. unterdurchschnittlich gut erkannt, fallen aufgrund ihres relativ geringen Anteils von ca. 12% Adjektive/Adverbien und nur ca. 1,7% Sonstige weniger ins Gewicht. Für die Verbesserung des Netzes ist daher zunächst sowohl die Implementierung von Flexionsregeln angedacht, die flektierte Formen zur Infinitivform hin normalisieren, als auch ein Mechanismus zum

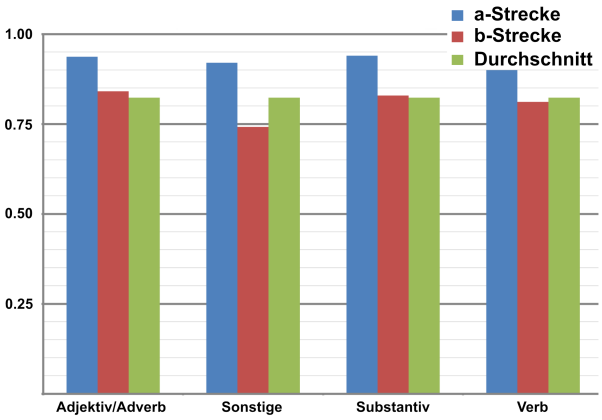

Abbildung 4: F-Scores nach Wortarten auf Basis der a- und b-Strecke

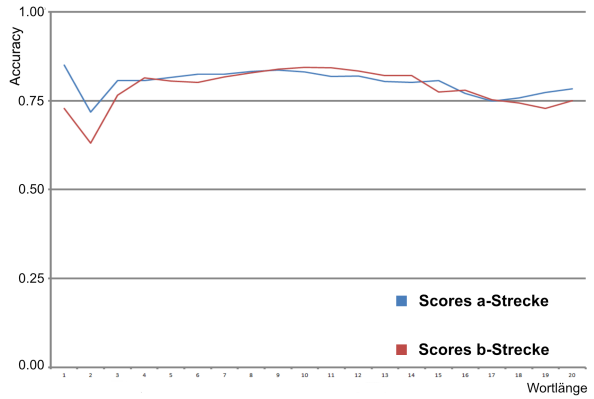

Abbildung 5: F-Scores in Abhängigkeit zur Wortlänge. Wörter mit mehr als 20 Buchstaben wurden ausgeklammert, da sie nur punktuell auftreten

Erfassen von nicht adjazent geschriebenen Partikelverben.

Hinsichtlich der Wortlänge werden VKF mit einer Länge von 5–12 Buchstaben überdurchschnittlich gut erkannt, vgl. Abb. 5. Dies ist erfreulich, da sie mit 62,5% (a-Strecke) und 72,2% (b-Strecke) den Großteil der zu lemmatisierenden VKF ausmachen.

Besonders interessant ist, dass Wörter, die nur aus einem Buchstaben bestehen, gut erkannt werden. Dies ist darauf zurückzuführen, dass es sich hierbei nur um Buchstabennamen handelt, die entsprechend gut zugeordnet werden können. Analog sollte dieser Mechanismus auch für besonders lange Wörter geltend gemacht werden, weswegen die Kurve nach dem zweiten lokalen Minimum nochmals ansteigt. Kurze Wörter werden erst dann problematisch, wenn sie, wie oben in (2) und (3) für Norma belegt, auch für das Netz zu false positives führen. Dies erklärt auch die vergleichsweise

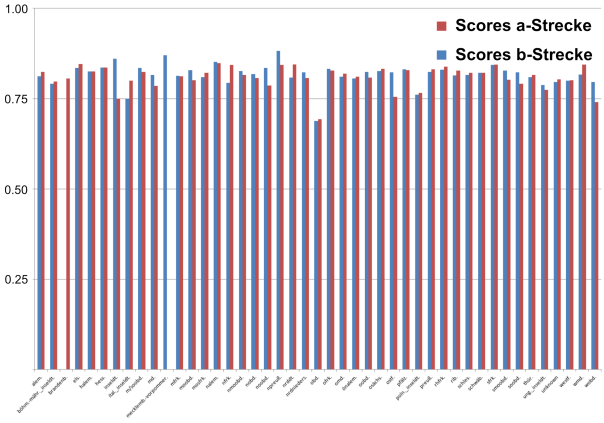

Abbildung 6: F-Scores nach Sprachraum

schlechten Scores der Sonstigen, da es sich hierbei generell um eher kurze Wörter handelt. Hier sollte die Korrektur des Scores nach oben false positives ausschließen. Eventuell ist zu überlegen, ob es unterschiedliche Limits für den Score geben könnte, die von der Wortlänge abhängig sind. Schlechter erkannte längere Wörter fallen aufgrund ihrer geringen Frequenz weniger ins Gewicht.

Über die Analyse der Sprachräume lassen sich einzelne herausarbeiten, die unterdurchschnittliche Werte aufweisen, wie z. B. das Oberdeutsche oder Preußische (s. Abb. 6).[13] Dies weist auf Sprachräume hin, deren VKF erheblich von der Gestalt des Lemmazeichens im FWB abweichen (Preußisch). Es könnte sich jedoch auch um Sprachräume handeln, die weniger belegt sind (Oberdeutsch mit nur durchschnittlich 0,1% aller VKF) und deren Systematiken daher nicht in genügendem Umfang vom Netz erlernt worden sind. Eine Lösung für beide Problematiken könnten Regelsets sein, die sprachraumspezifische Normalisierungen durchführen und über die entsprechenden Features der Sprachraum-Neuronen aktiviert werden. Solche Regelsets lassen sich mit Norma generieren und sollten für das Netz produktiv gemacht werden können. Daneben sollten die Trainingsdaten so gewählt werden, dass alle Sprachräume möglichst gleich stark vertreten sind.

7 Ausblick

Wir haben in diesem Beitrag zwei Ansätze beschrieben, die für die Identifikation von Vorkommensformen genutzt werden können. Beide erreichen noch keine perfekte Abdeckung. Norma erreicht mit extrem wenig Trainingsdaten bereits gute Ergebnisse: die Präzision liegt z. B. bei Norma-full bei nahezu 100%, bei einer Abdeckung von 66.2%. Das Netz wurde auf einer größeren Datenmenge trainiert, die allerdings weniger spezifisch waren. Es erreicht eine Abdeckung von 86,66% und hinsichtlich der Sprachräume wesentlich homogenere Ergebnisse.

Die hier umrissene Lemmatisierung stellt eine notwendige Grundlage für eine geplante Semantisierung frühneuhochdeutsche Texte dar. Ist ein genügend großer Anteil der entsprechenden Quellen lemmatisiert, kann, z. B. über Kollokationsanalysen und vektorbasierte Verfahren damit begonnen werden, die Lesarten der erkannten Lemmata zu disambiguieren. Eine solche Semantisierung würde z. B. Wortformen von *gnade* nicht mehr nur auf den entsprechenden Artikel verlinken, sondern auf eine der 20 verschiedene Lesarten, die im FWB notiert sind und von 1. “unverdiente, unerwartete, rettende, helfende Zuwendung des liebenden Gottes zum Menschen” über 10. “Gabe, die eine höhergestellte Person aufgrund einer wohlwollenden Gesinnung an einen in der Hierarchie Niedrigeren verteilt” bis hin zu 17. “Teil einer Begrüßungs- und Segensformel” reichen. Allein dies zeigt, welchen Mehrwert eine zukünftige Semantisierung frühneuhochdeutscher Texte haben könnte, der sich u. A. im Erkenntnisgewinn während der Lektüre niederschlagen würde oder tiefergehende semantische Analysen wie beispielsweise eine Methaphernanalyse unterstützen würde.

[13] In Ermangelung von geeigneten Sprachkürzeln gemäß ISO 639 werden die im FWB verwendeten Sprachkürzel verwendet.

Bibliographie

Marcel Bollmann. 2012. (Semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Workshop on Annotating Corpora for Research in the Humanities (ACRH-2)*, Lisbon.

Andrea Ernst-Gerlach and Norbert Fuhr. 2006. Generating search term variants for text collections with historic spellings. In *Proceedings of the 28th European Conference on Information Retrieval Research (ECIR 2006)*, München.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2018. *Deep Learning: Das umfassende Handbuch. Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze*. MIT Press.

Lisa Klaffki, Stefan Schmunk, and Thomas Stäcker. 2018. *Stand der Kulturgutdigitalisierung in Deutschland: Eine Analyse und Handlungsvorschläge des DARIAH-DE Stakeholdergremiums “Wissenschaftliche Sammlungen”*. DARIAH-DE working papers 26. Göttingen.

Markus Konrad. 2019. GermaLemma: A lemmatizer for German language text. https://github.com/WZBSocialScienceCenter/germalemma.

Matthias Liebeck and Stefan Conrad. 2015. IWNLP: Inverse Wiktionary for natural language processing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 414–418, Beijing, China.

Roland Mittmann. 2016. Automatisierter Abgleich des Lautstandes althochdeutscher Wortformen. *Journal for Language Technology and Computational Linguistics*, 31(2):17–24. Special issue on Corpora and Resources for (Historical) Low Resource Languages.

Thomas Pilz, Andrea Ernst-Gerlach, Sebastian Kempken, and Paul Rayson. 2007. The identification of spelling variants in English and German historical texts: Manual or automatic? *Literary and Linguistic Computing*, 23(1).

Hans Joachim Postel. 1969. *Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse*. IBM-Nachrichten, 19. Jahrgang. Stuttgart.

Tariq Rashid. 2017. *Neuronale Netze selbst programmieren. Ein verständlicher Einstieg mit Python*. O'Reilly, Heidelberg.

Oskar Reichmann. 1986. *Frühneuhochdeutsches Wörterbuch. Band 1: Einführung, a - äpfelkern*. Berlin, New York. Herausgeber: Robert R. Anderson [für Band 1], Ulrich Goebel, Anja Lobenstein-Reichmann [Einzelbände] & Oskar Reichmann [Bände 3 und 7 in Verbindung mit dem Institut für deutsche Sprache; ab Band 9, Lieferung 5 im Auftrag der Akademie der Wissenschaften zu Göttingen].

Helmut Schmid. 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 (DATeCH2019)*, page 133–137. Association for Computing Machinery.

Joachim Steinwendner and Roland Schwaiger. 2019. *Neuronale Netze programmieren mit Python*. Rheinwerk, Bonn.

Christian Wartena. 2019. A probabilistic morphology model for German lemmatization. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 40–49.

William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods*, Göttingen.

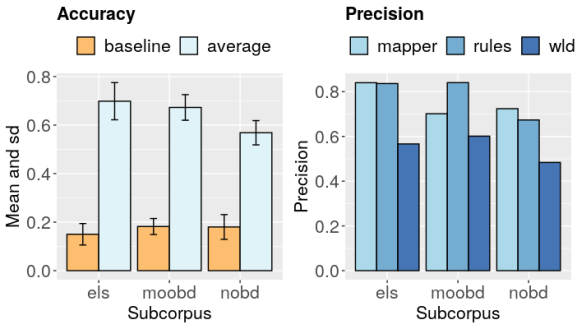

Abbildung 7: Genauigkeit (links) und Precision der einzelnen Normalisierer (rechts) in der ersten Evaluation von Norma

Appendix

Norma als Lemmatisierer Norma wurde für die Normalisierung flektierter Wortformen entwickelt. Für die VKF-Identifikation setzen wir Norma abweichend dafür ein, VKF-Kandidaten aus Belegen zu lemmatisieren. In einer ersten Evaluation untersuchten wir daher zunächst, wie gut Norma flektierte Originalschreibungen auf standardisierte Lemmata abbilden kann. Dazu führten wir eine sechsfache Crossvalidierung durch und trainierten Norma auf jeweils 500 Paaren aus drei verschiedenen Sprachräumen (Nordoberdeutsch/nobd, Mittleres Ostoberdeutsch/moobd, Elsässisch/els) und evaluierten auf jeweils 100 Paaren. Dieselben Splits wurden in Kap. 6 für die Evaluation der VKF-Identifikation durch Norma genutzt.

Als Baseline verwendeten wir ein einfaches System, das jeweils die vorliegende Wortform als Lemma vorhersagt.

Abb. 7 zeigt die Ergebnisse. Die Baseline der Genauigkeit liegt zwischen 15,0–18,2%, die Durchschnittswerte ("average") zwischen 56,8-69,8% pro Teilkorpus, was Norma für die (wesentlich leichtere) Aufgabe der VKF-Identifikation als mögliches Tool erscheinen lässt. Die Normalisierer Mapper und Rules erreichen gute Precision-Werte (Mapper: 70,1–84,0%, Rules: 67,3–84,0%). WLD schneidet am schlechtesten ab (48,3–60,1%), allerdings muss hier berücksichtigt werden, dass WLD als letzte Komponente die "schwierigen" Fälle übernimmt und insgesamt die meisten Wortformen lemmatisiert (gesamt: 1.800; WLD: 1.013; Mapper: 453; Rules: 333; ohne Analyse: 1). In allen Fällen sind die Werte für das Korpus nobd am niedrigsten.

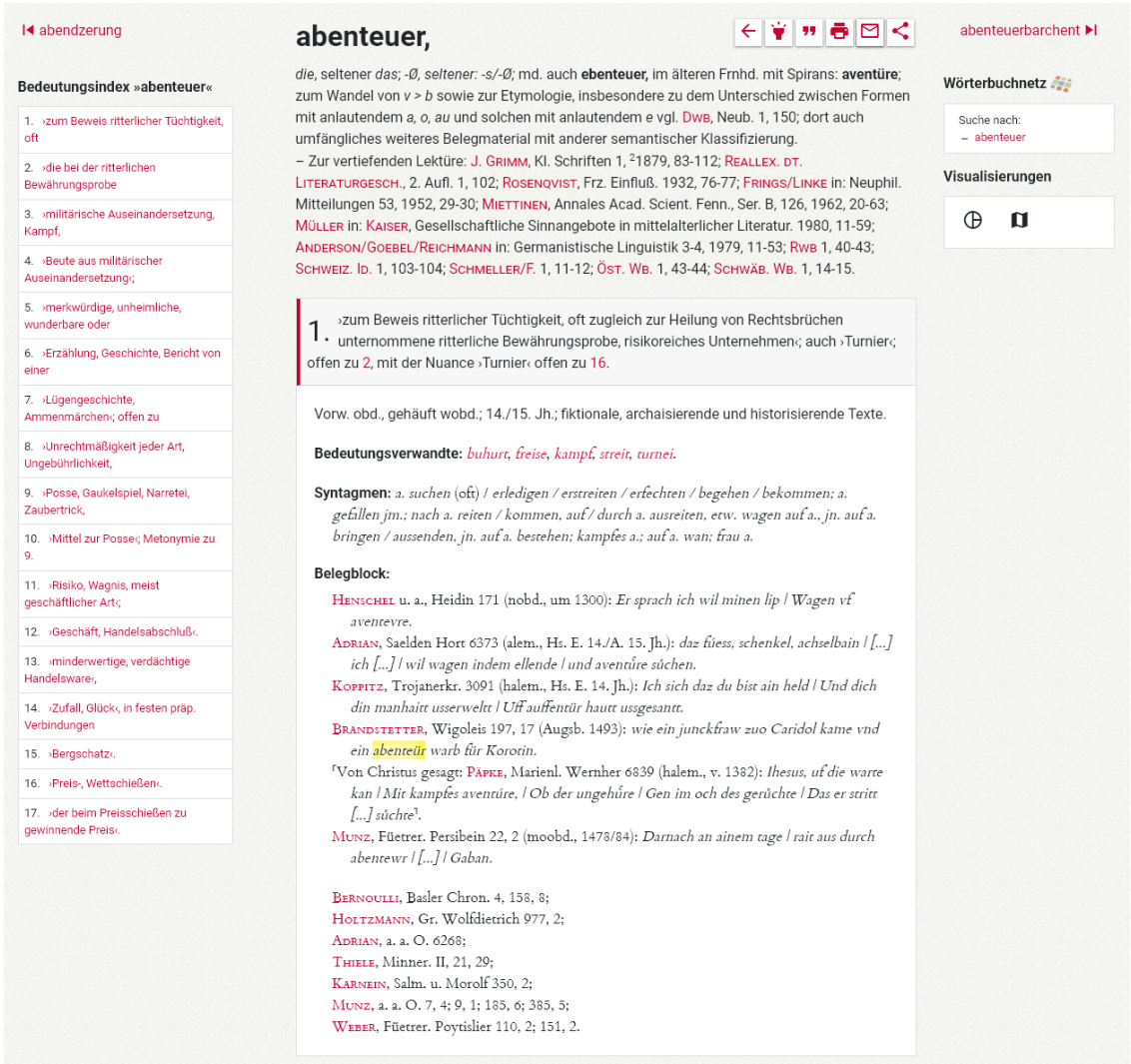

Abbildung 8: Beispielhafter Bedeutungsansatz 1 des Lemmas *abenteuer* in der Online-Ausgabe des FWB (http://fwb-online.de/go/abenteuer.s.1fn_1619637065, letzter Zugriff: 03.06.2021)