



Masterarbeit

im Studiengang Computerlinguistik
an der Ludwig-Maximilians-Universität München
Fakultät für Sprach- und Literaturwissenschaften

Sprachmodelle und Konstruktionsgrammatiken zur Textgenerierung und Konstruktionserkennung

vorgelegt von

Tim Sockel, B. Sc.

Matrikelnummer 12396122

tim.sockel@campus.lmu.de

Betreuerin: Leonie Weißweiler, M. Sc.

Prüfer: Prof. Dr. Hinrich Schütze

Bearbeitungszeitraum: 19. September 2023 – 06. Februar 2024

Sebstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt und alle Zitate als solche kenntlich gemacht habe sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

Für die Erstellung der Quellcodes zur Versuchsdurchführung und -visualisierung wurden zum Teil Sprachmodelle (ChatGPT und GitHub Copilot) unterstützend herangezogen.

München, 06. Februar 2023

Tim Sockel

Kurzfassung

Konstruktionsgrammatiken bilden Sprache durch bewusste Vereinigung verschiedener Ebenen der Linguistik ab (Syntax, Semantik, Lexik etc.) und haben somit eine holistische sprachwissenschaftliche Analyse zum Motiv. Große Sprachmodelle (LLMs) haben ebenfalls das Ziel, Sprache in ihrer Gänze zu erfassen, und nutzen dafür große Datenmengen und hohe Rechenleistung aus. Die vorliegende Arbeit untersucht, ob die Verknüpfung beider Ansätze zu einer Verbesserung der Leistung im Bereich des Sprachverständnisses (NLU) führen kann. Dabei werden zwei Hauptaufgaben formuliert: (a) die Erkennung von Sprachphänomenen in Form von Konstruktionen als Beispiele in Korpora; (b) die Generierung neuer Beispiele für definierte Konstruktionsbeschreibungen. Hierfür werden Sprachmodelle wie Llama 2, BERT und mBART verglichen sowie Erweiterungen durch bspw. Pseudowort-Verfahren und Few-Shot-Ansätze analysiert, implementiert sowie maschinell und manuell evaluiert.

Abstract

Construction Grammar aims to combine linguistic features such as syntax, semantics, and lexis, thereby analysing language holistically. Large language models (also known as LLMs) pursue a similar approach. They claim to grasp language features by using massive amounts of data and computational processing power. This thesis investigates joint approaches, targeting an optimization regarding the quality of natural language understanding (NLU). Hence, two main tasks are postulated: (a) the detection of language phenomena by means of finding instances for predefined constructions in corpora and (b) generating novel examples based on construction definition. In order to do so, LLMs like Llama 2, BERT, and mBART are compared and extended. To these models, concepts of few-shot learning and pseudoword embeddings are applied. The proposed ideas are analysed, then implemented, and finally evaluated both automatically and manually.

Danksagung

An dieser Stelle seien einige Personen und Institutionen benannt, welche die Entstehung dieser Arbeit besonders geprägt haben.

Zunächst danke ich selbstverständlich meiner Betreuerin Leonie Weißweiler (M. Sc.) für ihre Unterstützung und ihren Ansporn. Außerdem meinem Prüfer Prof. Dr. Hinrich Schütze und dem Centrum für Informations- und Sprachverarbeitung (CIS) insgesamt, dessen Lehrpersonal mir im Laufe meines Masterstudiums an der Ludwig-Maximilians-Universität München viele Eindrücke aus der Computerlinguistik liefern konnte.

Weiterhin danke ich der Rechnerbetriebsgruppe (RBG) des Instituts für Informatik an der LMU München sowie dem Forschungsinstitut CODE der Universität der Bundeswehr München, hier insbesondere der Professur für Data Science unter Prof.'in Dr. Michaela Geierhos. Ohne die zur Verfügung gestellten Ressourcen wie hochleistungsfähige GPUs und Remote-PCs wären die meisten Experimente im Rahmen der vorliegenden Arbeit nicht möglich gewesen.

Ebenfalls mit Blick auf das FI CODE danke ich meinen Kollegen, insbesondere Moritz Hennen (B. Sc.), der mir mit seinem praktischen Hintergrundwissen zu BART und Transformermodellen im Allgemeinen unerlässliche Hinweise und Erfahrungswerte mit auf den Weg gegeben hat. Weiterhin danke ich meinem Teamleiter Florian Babl (M. Sc.) für die interessanten Gespräche und neuen Ideen während der Entstehung dieser Arbeit.

Vom Institut für Germanistik an der Heinrich-Heine-Universität Düsseldorf möchte ich mich ganz besonders bei Nina Böbel (M. A.) bedanken. Sie hat in zahlreichen Beratungsgesprächen gemeinsam mit Leonie Weißweiler unermüdlich spannende Einblicke in die Arbeit am Konstruktikon sowie Informationen bzgl. der zugrundeliegenden Daten liefern können. Über den fachübergreifenden Austausch habe ich mich hierbei besonders gefreut!

Ganz besonders dankbar bin ich außerdem meinen beiden guten Freundinnen und Kommilitoninnen Paulina „Paula“ Belberov (M. Sc.) und Dr. Sarah A. Uffelmann, nicht nur für ihre stets ehrlichen Meinungen und Rückmeldungen zum Stand meiner Arbeit sondern auch für die nicht zuletzt mentale und moralische Stütze. Gleichermanßen Dank verdienen natürlich auch die anderen Mitglieder der Fachschaft Computerlinguistik, die mich auf meinem gesamten Studienweg unterstützt haben und die mir sehr ans Herz gewachsen sind.

Abschließend möchte ich ganz besonders bei meiner lieben Familie danken, die mich immer unterstützt, selbst wenn ich gerade weder geistig noch körperlich in der Nähe bin! Vielen lieben Dank euch!

München, im Februar 2024

»Speakers are at once impressively creative and impressively repetitive.«

ADELE E. GOLDBERG (2013, S. 26)

Inhaltsverzeichnis

1 Einführung	1
1.1 Motivation und Überblick	1
1.2 Konstruktionen als Abstraktion natürlicher Sprache	2
1.2.1 Von generativen zu konstruktionsbasierten Grammatiken	2
1.2.2 Definition und Bedeutung im Kontext von Sprachmodellen	4
1.3 Sprachmodelle zur Produktion natürlicher Sprachen	5
1.3.1 Definition und Bedeutung von Sprachmodellen	5
1.3.2 Transformer-Architektur, BERT und BERT-ähnliche Modelle	6
1.3.3 Fortschritte und Kritik	7
2 Vorangegangene Arbeiten	9
2.1 Induktion von Konstruktionsgrammatiken	9
2.1.1 Manuelle Induktion	9
2.1.2 Maschinelle Induktion	10
2.2 Erkennung von Mustern und Konstruktionskandidaten in Textdaten	12
2.3 Generierung neuer Beispiele für gegebene Konstruktionen	13
2.4 Spezialisierung und Erweiterung von Sprachmodellen hinsichtlich NLU	14
3 Erkennung von Konstruktionen in gegebenen Texten	17
3.1 Problemanalyse bei der Konstruktionsdetektion	17
3.2 Konzeption der Untersuchung hinsichtlich Konstruktionserkennung	18
3.2.1 Vorverarbeitung mit <i>Universal Dependencies</i>	18
3.2.2 Erkennung von Konstruktionen mit vortrainierten Sprachmodellen	21
3.2.3 Ansätze zur Erweiterung der vortrainierten Sprachmodelle durch Pseudowort-Embeddings	25
3.2.4 Vergleich von allgemeinen kontextuellen Embeddings mit Pseudowort-Embeddings	28
3.3 Details zu Implementierung und Hardware	35
3.3.1 Auswahl der konkreten Sprachmodelle	35
3.3.2 Anmerkungen zur Implementierung des Pseudowort-Embedding-Trainings mit BERT	35
3.3.3 Hardwareanforderungen bzgl. Llama 2 und BERT	36
4 Generierung von Textbeispielen zu Konstruktionen	37
4.1 Anforderungen an generierte Beispiele für Konstruktionen	37
4.2 Konzeption der Untersuchung hinsichtlich der Generierung von Beispielsätzen	37
4.2.1 Einsatz vortrainierter Sprachmodelle als Generatoren für Konstruktionsbeispiele	38
4.2.2 Zero-Shot-Ansatz mit Llama 2	38
4.2.3 Few-Shot-Ansatz mit Llama 2	41
4.2.4 Generierung von Beispielen ab einem KE-lex	43
4.3 Verwendung von Pseudowörtern zur Generierung	45
4.3.1 Pseudoword-Embeddings für Konstruktionen mit BERT	45
4.3.2 Erweiterung von mBART mit Pseudowörtern	47

Inhaltsverzeichnis

4.3.3	Details der Implementierung und Hardware beim Training von Pseudowort-Embeddings für mBART	51
4.4	Ergänzung von Prompting-Ansätze durch Pseudowort-Überprüfung	51
5	Zusammenfassung	55
5.1	Alternative Ansätze	56
5.2	Ausblick	58
A	Ergebnisse und Rohdaten	59
A.1	Daten zur Erkennung von Konstruktionen in gegebenen Texten	59
A.2	Daten zur Generierung von Textbeispielen zu Konstruktionen	73
B	Verzeichnis der untersuchten Konstruktionen	97
	Literaturverzeichnis	103

Abbildungsverzeichnis

1.1	Autoregressive Encoder-Decoder-Generierung mit Transformer-Modellen	6
1.2	Pre-Training von BERT	8
1.3	Verarbeitung von Eingabe und Generierung der Ausgabe über das Kontextfenster von GPT-3	8
2.1	Illustration der MaPP-Methodik zum Training von Pseudowort-Embeddings	15
3.1	Anzahl gefundener Sätze als potenzielle Beispiele bei der Vorfilterung mittels <i>Universal Dependencies</i> pro Konstruktion	19
3.2	Rangfolge von Konstruktionen gemäß durchschnittlicher Zugehörigkeit m der gefundenen potenziellen Beispiele	19
3.3	Konzept der Konstruktionserkennung mit Llama 2	22
3.4	Anteil korrekt erkannter Beispiele durch Llama 2 in Abhängigkeit zur Anzahl gegebener korrekter und inkorrektener Beispiele	23
3.5	Konzept der Konstruktionserkennung mit BERT und NSP	23
3.6	Anteil korrekt erkannter Beispiele durch BERT in Abhängigkeit zur Anzahl gegebener korrekter und inkorrektener Beispiele	24
3.7	Anteil korrekt erkannter Beispiele durch BERT bei vorgegebener Anzahl von Beispielen, die durch BERT als korrekt zu bewerten sind	24
3.8	Training eines Pseudowort-Embeddings für BERT auf Basis eines Beispielsatzes des Konstruktions mit zwei KE-Tokens	27
3.9	Anteil korrekt erkannter Beispiele durch BERT; KE-lex wurden durch entsprechende Pseudowort-Embeddings ersetzt	29
3.10	Anteil korrekt erkannter Beispiele durch BERT; KE-lex wurden durch entsprechende Pseudowort-Embeddings ersetzt; vorgegebene Anzahl von korrekten Beispielen	29
3.11	Ergebnisse des Signifikanztests (gepaarter t -Test) beim Vergleich von BERT vor und nach dem Ersetzen der KE-lex durch Pseudowort-Embeddings	30
3.12	Ergebnisse des Signifikanztests (gepaarter t -Test) bei vorgegebener Anzahl von korrekten Beispielen	30
3.13	Abschätzung der Zugehörigkeit gefilterter Beispiele zu Konstruktionen durch Ähnlichkeits- und Abstandsmetriken Δ über KE-lex-Embeddings	31
3.14	Anteile der Sätze des HDT-UD-Korpus, für die das jeweilige Pseudowort-Embedding näher am Satztoken-Embedding ist; Vergleich mit durchschnittlichem Embedding bei Instanziierung als Ein-Wort-Satz	32
3.15	Anteile der Sätze des HDT-UD-Korpus, für die das jeweilige Pseudowort-Embedding näher am Satztoken-Embedding ist; Vergleich mit durchschnittlichem Embedding über alle Konstruktions-Beispiele	32
3.16	Verhältnis der Häufigkeit einer Übereinstimmung pro Konstruktion; Ein-Wort-Sätze als Vergleichsembeddings	33
3.17	Verhältnis der Häufigkeit einer Übereinstimmung pro Konstruktion; Vergleichsembeddings gemittelt über die Konstruktions-Beispiele	33

Abbildungsverzeichnis

4.1	Konzept der Zero- bzw. Few-Shot-Ansätze bei der Generierung von Beispielsätzen für eine gegebene Konstruktion mit Llama 2	38
4.2	Anzahl verschiedener generierter Sätze pro Konstruktion durch Llama 2 mittels Zero-Shot-Ansatz (Ausschnitt der besten 20)	39
4.3	Anteil an Überlappungen zur gegebenen Definition in den durch Llama 2 generierten Sätzen mittels Zero-Shot-Ansatz pro Konstruktion (Ausschnitt der schlechtesten 20)	40
4.4	Anzahl verschiedener generierter Sätze pro Konstruktion durch Llama 2 mittels Few-Shot-Ansätzen (Ausschnitt der besten 20)	41
4.5	Anteil an Überlappungen zur gegebenen Definition in den durch Llama 2 generierten Sätzen mittels Few-Shot-Ansätzen pro Konstruktion (Ausschnitt der schlechtesten 20)	42
4.6	Generierung von Beispielsätzen durch Kürzen und Fortsetzen des Eingabesatzes ab dem KE-lex-Token	43
4.7	Training eines Pseudowort-Embeddings für mBART auf Basis eines Beispielsatzes des Konstruktikons	48
4.8	Konstruktionsübergreifende gemittelte Entwicklung des Loss-Werts für das Training von Pseudowort-Embeddings für mBART	49
4.9	Kombination von Verfahren zur Generierung mittels Llama 2 und Erkennung durch BERT mit Pseudowort-Embeddings	53
5.1	Idee eines intervallbasierten Trainings auf Konstruktionen	57
A.1	Anzahl verschiedener generierter Sätze und Anteil an Überlappungen durch Zero-Shot-Ansatz mit Llama 2	95
A.2	Anzahl verschiedener generierter Sätze und Anteil an Überlappungen durch Few-Shot-Ansatz mit Llama 2	96

Tabellenverzeichnis

3.1	Evaluationsergebnisse über die gefundenen Beispiele im HDT-UD-Korpus	34
4.1	Grammatikalität mittels Llama 2 generierter Sätze und deren Korrektheit bzgl. der jeweils gegebenen Konstruktion nach menschlicher Evaluation	42
4.2	Ausschnitt aus den <mask>-Vorhersagen durch mBART-50	44
4.3	Ausschnitt aus den [MASK]-Vorhersagen durch BERT unter Anwendung von Pseu- dowörtern	46
4.4	Ausschnitt aus den <mask>-Vorhersagen durch mBART-50 unter Anwendung von Pseudowörtern	50
A.1	Konstruktionen und ihr dazugehöriger <i>größter gemeinsamer Schnitt</i>	59
A.2	Gefilterte Sätze aus dem HDT-UD-Korpus mit m , q und menschlicher Evaluation; Vergleich der KE-lex mit Ein-Wort-Sätzen	65
A.3	Gefilterte Sätze aus dem HDT-UD-Korpus mit m , q und menschlicher Evaluation; Vergleich der KE-lex mit durchschnittlichen Embeddings über Konstruktikon- Beispiele	69
A.4	Grammatikalität und Korrektheit generierter Sätze mit Llama 2 und Zero-Shot- Ansatz	73
A.5	Grammatikalität und Korrektheit generierter Sätze mit Llama 2 und 3-Shot-Ansatz	82
B.1	Berücksichtigte Konstruktionen aus dem <i>FrameNet-Konstruktikon des Deutschen</i>	97

1 Einführung

1.1 Motivation und Überblick

Zur Erfassung natürlicher Sprache existieren unterschiedlichste Ansätze. Konstruktionsgrammatiken bilden hierbei ein prominentes Beispiel und sehen vor, linguistisches Wissen ausschließlich aus sogenannten *Konstruktionen* zu beziehen. Im Vergleich zu streng modularen Theorien können auf diese Weise insbesondere idiomatische Ausdrücke mit variabler Lexikalität korrekter und repräsentativer erfasst werden, da Bedingungen aus dem Bereich der Grammatik als auch aus der Lexik vereinigt werden müssen. Eine Sammlung solcher Konstruktionen kann entsprechend als *Konstruktikon* bezeichnet werden. (HILPERT, 2014, S. 22)

Die Notwendigkeit solcher Konstruktika wird demnach durch das Ideal motiviert, Sprache in ihrer Komplexität und Gänze holistisch zu beschreiben. Ein verwandtes Ziel verfolgt auch die Computerlinguistik, im Speziellen das häufig synonym genannte Natural Language Processing (NLP). In den letzten Jahren wurde insbesondere durch die Entwicklung der sogenannten *Large Language Models* (LLMs) bzw. (großen) Sprachmodelle auf Grundlage von künstlichen neuronalen Netzen (NNs) ein großer Fortschritt erzielt. Sprachmodelle sind in der Lage, im Vergleich zu bis dahin üblichen regelbasierten Methoden Sprache in überzeugenderer Weise zu verstehen und zu produzieren.

Die vorliegende Arbeit versucht sich in einer Vereinigung der Vorteile beider bisher weitgehend unabhängig voneinander entwickelten Bereiche der linguistischen Forschung. Die zentralen Fragestellungen beschäftigen sich damit, ob Sprachmodelle zur Detektion von Textnachweisen für gegebene Konstruktionen anwendbar sind und wie sie dazu befähigt werden können, neue Beispiele zu erzeugen. Hierbei ist die Struktur der Kapitel durchweg zweigeteilt: Zunächst wird untersucht, inwieweit Sprachmodelle spezielle, durch Konstruktika definierte Phänomene in Texten erkennen können; danach wird die Fähigkeit zur Generierung neuer Beispiele zu gegebenen Konstruktionen betrachtet. Um die aktuelle Ausgangslage sowohl der Konstruktionsgrammatiken als auch der Sprachmodelle sowie mögliche Problematiken darzustellen, wird in diesem Kapitel zunächst ein historischer Abriss beider Untersuchungsgegenstände vorgenommen. Darin wird auf den aktuellen Stand der jeweiligen Gebiete hingeführt, bevor in Kapitel 2 auf aktuelle Forschungsthemen und verwandte Beiträge im Bereich der wechselseitigen Nutzung von Konstruktionsgrammatiken und Sprachmodellen eingegangen wird. Hierzu werden – nach einer kurzen Einführung zum verwandten Thema der maschinellen und manuellen Konstruktionsinduktion – bisherige Ansätze diskutiert, die sich mit der Erkennung von Sprachphänomenen einerseits und der Generierung neuer Beispiele andererseits beschäftigen. Außerdem wird ein Konzept zur spezialisierten Erweiterung von Sprachmodellen mittels Pseudowörter diskutiert. Dieses soll danach auch in dieser Arbeit exemplarisch eingesetzt werden, indem es auf die Struktur von Konstruktionsgrammatiken angepasst wird.

Im Hauptteil dieser Arbeit werden entsprechend in Kapitel 3 ebenfalls zunächst Möglichkeiten dargelegt, mit denen die Fähigkeiten von Sprachmodellen zur Erkennung von Konstruktionsbeispielen in Korpora und anderen Texten ausgenutzt werden können. Nach einer schematischen Problemanalyse werden Konzepte und Ergebnisse diskutiert, die sich durch verschiedene Vorfilterungs- und Erkennungstechniken ergeben. Zentral ist die Beschreibung des Gebrauchs von Pseudowörtern, welcher mögliche Zero- und Few-Shot-Ansätze von generativen Modellen ge-

genübergestellt werden.

Im darauffolgenden Kapitel 4 wird im Gegenzug die Generierung neuer Beispiele untersucht. Es werden zunächst Anforderungen hieran definiert, Konzepte für den Einsatz von Sprachmodellen dargelegt und anschließend qualitativ ausgewertet. Außerdem wird eine Möglichkeit vorgestellt, mit derer die Vorteile der Erkennungs- und Generierungskonzepte potenziell verbunden werden können.

Abschließend fasst Kapitel 5 die Erkenntnisse der vorliegenden Arbeit zusammen, benennt einige alternative Konzepte zu den hier dargelegten und gibt einen Ausblick auf zukünftige Entwicklungen im Bereich der Konstruktionsgrammatiken in Verbindung mit Sprachmodellen. Im Anhang finden sich Rohdaten und ausführlichere Grafiken zu den dargestellten Themen.

1.2 Konstruktionen als Abstraktion natürlicher Sprache

1.2.1 Von generativen zu konstruktionsbasierten Grammatiken

In Kapitel 1.1 wurde bereits der Begriff der Konstruktionsgrammatik sowie des Konstruktionsgrammatikons motiviert. Eine allgemein anerkannte Definition von Konstruktionen und Konstruktionsgrammatiken lässt sich dennoch nicht ohne weiteres treffen. Verschiedene Ansichten und Interpretationen des Konstruktionsbegriffs sowie dessen Anwendung sind historisch und aktuell verbreitet. Um im Kontext von Sprachmodellen eine geeignete Variante ausfindig machen zu können, sollen einige bedeutende Schritte in der Geschichte der Konstruktionsgrammatiken diskutiert werden.

Traditionelle Ansätze in den linguistischen Sprachwissenschaften, etwa durch CHOMSKY in den frühen 1950-er Jahren, legen nahe, bei der Betrachtung von Sprachphänomenen zwischen Bereichen wie der Lexik, der Syntax oder der Semantik der jeweiligen Sprache streng zu unterscheiden. CHOMSKY als prominenter Vertreter der sogenannten generativen Grammatik ist bereits in frühen Arbeiten davon überzeugt, dass „der Begriff ‘grammatisch’ nicht mit ‘bedeutungstragend’ oder ‘sinnvoll’ in irgendeinem semantischen Sinn identifiziert werden“ (CHOMSKY, 2011, S. 17) dürfe und „daß jede Suche nach einer semantisch begründeten Definition für ‘Grammatikalität’ vergeblich sein wird“ (ebd., S. 18). Weiterhin merkt er an:

Der [sic!] Theorie der Sprach-Struktur muß eindeutig von einer Anleitung hilfreicher Verfahren zur Entdeckung von Grammatiken unterschieden werden [...] Wenn man diesen Standpunkt einnimmt, gibt es wenig Grund für den Einwand, die Ebenen zu vermischen, für die Annahme, daß Elemente der höheren Ebene buchstäblich aus Elementen der tieferen Ebene konstruiert sind, oder für die Meinung, syntaktische Arbeit sei verfrüht, bevor nicht alle Probleme der Phonemik oder Morphologie gelöst sind. (ebd., S. 125)

Während CHOMSKYS Ausführungen zur *generativen Syntax* insbesondere für die theoretische Informatik im Sinne der formalen Sprachen bedeutende Anstöße leisten konnte, stellten sich die Theorien zur Transformationsgrammatik (ebd.) und CHOMSKY-Hierarchie in der Linguistik als größtenteils unzureichend heraus. Gegenentwürfe lieferten unter anderem die Diskussion um die sogenannte *Tiefenstruktur* im Sinne einer *generativen Semantik*, welche die sogenannten *Linguistic Wars* zur Folge hatte (HARRIS, 2021). Insbesondere aber widersprechen auch *Konstruktionsgrammatiken* einer starren Trennung linguistischer Teildisziplinen, obwohl selbst CHOMSKY anmerkt: „Wechselbeziehungen zwischen der syntaktischen Struktur und der Bedeutung [...] könnten einen Teil des Gegenstandes einer allgemeineren Sprach-Theorie bilden“ (CHOMSKY, 2011, S. 127).

Als Ursprung der Theorien um Konstruktionsgrammatiken können die späten 1950-er Jahren betrachtet werden. GLEASON als amerikanischer Strukturalist beschreibt eine Konstruktion als

beliebige bedeutungstragende (*significant*) Gruppe von Formen oder Morphemen, unabhängig von Umfang bzw. Anzahl der enthaltenen Wörter (vgl. GLEASON, 1961, S. 132). Dies widerspricht Ansichten von Gegnern der strukturalistischen Strömung wie CHOMSKY, welcher ein generatives Verständnis des Grammatikbegriffs vertritt und anmahnt, linguistische Strukturen „abstrakt [...], also ohne besonderen Bezug zu einzelnen Sprachen“ (CHOMSKY, 2011, S. 13) zu untersuchen. Allerdings betonen seinerzeit auch andere Strömungen der zeitgenössischen Forschung die Bedeutung von lexikogrammatischen Mustern, etwa durch Begriffe wie *Kollokation* (FIRTH, 1951) – als gemeinsames Auftreten sprachlicher Elemente – und dem *Idiomprinzip* nach HUNSTON und FRANCIS (2000), welche in einer Definition der *Pattern Grammar* durch ebendiesen zusammengefasst werden. Das Idiomprinzip besagt hierbei, dass gewisse Muster beim Benutzen einer Sprache in unveränderlicher Weise verwendet werden, obwohl sie zunächst weiter segmentierbar erscheinen (vgl. SINCLAIR, 1991, S. 110). Somit besteht eine Pattern Grammar aus unterschiedlich komplexen idiomatischen Mustern.

Anhand der Analyse der Verbform *trained* (vgl. im Folgenden HUNSTON und FRANCIS, 2000, S. 71–72) machen sich jedoch gleichzeitig Schwachstellen dieses Ansatzes bemerkbar:

- (1) I have trained in counselling, hypnotherapy and ...
- (2) Ningali has trained in dance but never as an actor ...
- (3) Armani was never formally trained in design, though he says he must ...
- (4) ... and they are specially trained in underground rescue.

Hier lässt sich ein Muster der Form ‚V in n‘ annehmen, dieses ist also in den gegebenen Beispielen vermeintlich abhängig vom spezifischen Verb *trained* (im Sinne von *ausgebildet in einer Fähigkeit*), dem ein *in* und ein Substantiv (bzw. Nominalkompositum) folgt. Allerdings ist diese Definition problematisch für andere Vorkommnisse:

- (6) ... the group trained in New York and have performed in ...
- (7) ... his father Franics, who was trained in Paris at the famous Larue ...

Diese Vorkommnisse von *trained* besitzen eine grundsätzlich andere (lokative) Bedeutung (im Sinne von *ausgebildet an einem Ort*) und stellen die Annahme in Frage, dass diesem Muster ein bestimmtes Verb einer bestimmten Semantik zugrunde liegt. Stattdessen wäre es in einem solchen Fall hilfreicher, die möglichen Beschränkungen für die Elemente der Muster zu erweitern; etwa im gegebenen Beispiel um allgemeinere Notationen für Lokalbestimmungen (z. B. ‚V in LOC‘ für „trained in New York“) bzw. ‚Fähigkeit‘ (z. B. ‚V in SKILL‘ für „trained in design“). Allerdings sind solche Angaben nicht Teil der Pattern-Grammar-Ontologie, sodass diese sich als nicht aussagekräftig genug herausstellt.

Weiterführende und alternative Interpretationen, die solche Probleme zu umgehen versuchen, umfassen hierbei etwa die *Case Grammar* durch FILLMORE (1968). Diese heben vor allem die Bedeutung des Verbs und möglicher Fälle von Dependenz (cases) in einem Ausdruck hervor. Die diskutierten Probleme bei einer getrennten Betrachtung von Syntax und Semantik münden in der allgemeinen Annahme von Prinzipien, die den Theorien von Konstruktionsgrammatiken zugrunde liegen.

Die beschriebenen früheren Ideen spiegeln sich demnach heute in verschiedenen Strömungen des sogenannten *Konstruktionismus* wider, etwa der *Berkeley Construction Grammar*, zu deren Vertretern FILLMORE zählt, aber auch die *Cognitive Grammar* (u. a. LANGACKER, 2008) oder die *Radical Construction Grammar* (CROFT, 2009). Einen Versuch, diese Theorien auf eine gemeinsame Basis zu stellen, unternimmt u. a. GOLDBERG (2006, 2013). Die Autorin identifiziert hierbei fünf zentrale Ansichten, welche durch die meisten genannten Varianten des Konstruktionismus akzeptiert werden und damit eine Abgrenzung zu einer sog. generativen Grammatik bilden (vgl. GOLDBERG, 2013, S. 15–16):

1 Einführung

1. Grammatische Konstruktionen sind erlernte Paare aus Form und Funktion;
2. Semantik bezieht sich stets direkt auf die oberflächliche Struktur (und wird nicht über transformative oder ableitende Regeln gebildet);
3. Konstruktionen (inkl. Wörter und Morpheme) sind relational zueinander in einem Netzwerk angeordnet;
4. sprachenübergreifende Variabilität und Generalisierung entstehen v. a. durch bereichsübergreifende kognitive Prozesse;
5. Sprachelemente und -generalisierungen werden gebrauchsorientiert (*usage-based*) erlernt, sodass Konstruktionen unterschiedlich spezifisch definiert sein können.

GOLDBERG merkt an, dass insbesondere letzterer Punkt nicht von der gesamten linguistischen Gemeinschaft mitgetragen werde, argumentiert aber, dass dieser insbesondere für die Eingliederung des Konstruktionismus in aktuelle Betrachtungen zu Spracherwerb, -verarbeitung und -wandel bedeutsam sei. Punkt 3 bezieht sich auf die im Weiteren wichtige Annahme, dass Konstruktionen in einem Konstruktikon als Netzwerk strukturiert angeordnet werden können.

Die Sicht des Konstruktionismus erlaubt es aus der Perspektive der natürlichen Sprachverarbeitung (NLP) und insbesondere des *Natural Language Understanding* (NLU), eine akzeptable Abstraktion zu finden, um Sprachstrukturen einerseits für maschinelle Verarbeitung zu verallgemeinern und andererseits Modellierungen zu vermeiden, die einer natürlichen Sprache nicht hinreichend gerecht werden, etwa generative Grammatiken nach CHOMSKY (1956, 2011).

1.2.2 Definition und Bedeutung im Kontext von Sprachmodellen

Die vielfältigen Ansichten und teils gegensätzlichen Interpretationen erschweren eine Festlegung auf einen bestimmten Konstruktionsbegriff, da jede Variante bestimmte Aspekte stärker oder weniger stark beleuchtet, hinzufügt oder verneint. Auch die Ansichten des Konstruktionismus selbst werden nicht durch die gesamte linguistische Forschung gänzlich akzeptiert. Dennoch soll eine Definition versucht werden, auf die im Rahmen dieser Arbeit Bezug genommen wird.

Im weiteren Verlauf wird das *FrameNet-Konstruktikon des Deutschen* (HHU, 2024) für Datenakquise und -verarbeitung verwendet. Infolgedessen soll auch diejenige Definition angewendet werden, welche ebendiesem Projekt zu Grunde gelegt wird. Sie orientiert sich an der Ansicht von GOLDBERG (2006, 2013), entspricht deren fünf zentralen Punkten zum Konstruktionismus (siehe hierzu Kapitel 1.2.1) und fügt sich somit in aktuell gängige Interpretationsweisen ein.

Eine Konstruktion wird demnach definiert als „[k]onventionelles Form-Bedeutungs-Paar, das nicht vollständig kompositionell erschließbar oder durch eine hohe Auftretensfrequenz verfestigt ist [...]“ [GOLDBERG, 2006]. Konstruktionen variieren hinsichtlich des Grades an Spezifizierung und Abstraktheit.“ Weiterhin wird ein Konstruktikon im Ganzen als „[s]trukturiertes Netzwerk von Konstruktionen einer Sprache, das (meist in digitaler Form eines Online-Lexikons) als Repositorium bzw. strukturiertes Nachschlagewerk aufbereitet ist“ (HHU, 2024) festgelegt. Diese digitale Aufbereitung und die damit verbundene einheitliche Annotation erleichtern eine Arbeit mit Methoden der maschinellen Sprachverarbeitung bzw. ermöglichen diese überhaupt erst effektiv. Die spezifischen Einzelheiten und deren Anwendbarkeit auf die Nutzung mit Sprachmodellen werden in späteren Kapiteln (u. a. Kapitel 2.1.1 und 3.2) ausführlicher erläutert.

Ein Problem für die Linguistik in ihrer klassischen Arbeitsweise ist es zum einen, vermeintlich ‚interessante‘ Konstruktionen für ein Konstruktikon ausfindig zu machen und somit ein selbiges zu induzieren. Hierzu ist die Annotation und manuelle Analyse von einer großen Anzahl textueller Daten notwendig. Diese kann zwar gestützt durch maschinelle Werkzeuge stattfinden, erfordert allerdings weiterhin hohe personelle und zeitliche Kosten. Des Weiteren stellt auch die

Detektion einer bestimmten Konstruktion in einem Datensatz eine Herausforderung dar. Dabei müssen Textsequenzen gefunden werden, die mit der Annotation einer gegebenen Konstruktion in einem Konstruktikon übereinstimmen. Direkt verbunden damit ist im Umkehrschluss das Generieren zusätzlicher Beispiele bzw. Belege einer Konstruktion. Zwar erhebt die linguistische Forschung im Allgemeinen den Anspruch, ausschließlich real existierende Belegstellen zu einer gegebenen Konstruktion zu akzeptieren, dennoch wäre es gerade für die Untersuchung seltener Phänomene notwendig, zusätzliche Texte zu generieren, nicht zuletzt, um die Definition einer Konstruktion in ihrer Granularität und Komplexität kritisch zu hinterfragen.

In dieser Arbeit wird auf die letzteren beiden Problemstellungen Bezug genommen und – dadurch motiviert – untersucht, inwiefern Methoden der natürlichen Sprachverarbeitung und insbesondere Sprachmodelle allgemein anwendbare Möglichkeiten liefern können, Lösungen hierfür zu finden. Bisherige Ansätze finden sich in Kapitel 2 und werden dort eingehender diskutiert und bewertet, bevor in den Folgekapiteln eigene Versuche durchgeführt werden.

1.3 Sprachmodelle zur Produktion natürlicher Sprachen

Maschinelles Verständnis natürlicher Sprachen (NLU) war für lange Zeit ein Problem, das traditionelle Verfahren nur unzureichend lösen konnten. Regelbasierte Verfahren scheinen zunächst kostengünstig implementierbar zu sein, benötigen jedoch entweder eine große Anzahl an Definitionen, um möglichst viele Sonderfälle abzudecken, oder aber es müssen Qualitätseinbußen bei der Nutzung in Kauf genommen werden. Die damit einhergehende Diskussionskultur erstreckt sich nicht zuletzt auf die sogenannten *Linguistic Wars* (HARRIS, 2021).

Spätere Entwicklungen auf Basis erster künstlicher neuronaler Netzwerke (NNs) brachten eindeutige Verbesserungen mit sich – einen Überblick liefern JURAFSKY und MARTIN (2023). Dennoch konnten diese lange Zeit weiterhin nicht die nötige Effektivität im Verständnis von natürlicher Sprache erreichen. Ein größerer Durchbruch stellte vor wenigen Jahren die Einführung des Konzepts der Transformer-Architektur auf Basis der Encoder-Decoder-Modelle und des Attention-Mechanismus (VASWANI et al., 2017) dar.

1.3.1 Definition und Bedeutung von Sprachmodellen

Da die Idee der Transformer-Architektur im NLP-Bereich eng mit Sprachmodellen zusammenhängt, soll zunächst letzterer Begriff festgehalten werden. Ein Sprachmodell sei im Rahmen der vorliegenden Arbeit als ein probabilistisches Modell definiert, welches zu einer gegebenen Sequenz von Sprachtokens (z. B. durch Leerzeichen getrennte Wörter) das jeweils nächste Token vorhersagt. Ein Spezialfall stellen hierbei die sogenannten *Large Language Models* (LLMs) dar, welche sich durch die hohe Anzahl an verwendeten Trainingsdaten und die Anwendung von (Teilen der) Transformer-Architektur von anderen Sprachmodellen unterscheiden. Gleichzeitig zeichnen sie sich durch eine im direkten Vergleich zu anderen Sprachmodellen deutlich höhere Qualität bei der Generierung und dem Verständnis von Sprache aus. Im weiteren Verlauf dieser Arbeit wird unter dem Begriff ‚Sprachmodell‘ synonym ein LLM verstanden.

Insbesondere der NLU-Aspekt bei der Qualität der Ausgaben von Sprachmodellen ist interessant bei einer potenziellen Erfassung von Konstruktionsgrammatiken. Da LLMs Sprachdaten in einem großen Umfang umfassen, ist zu untersuchen, ob auch die Informationen, die linguistisch in Konstruktika annotiert sind, hiermit automatisiert verstanden werden können.

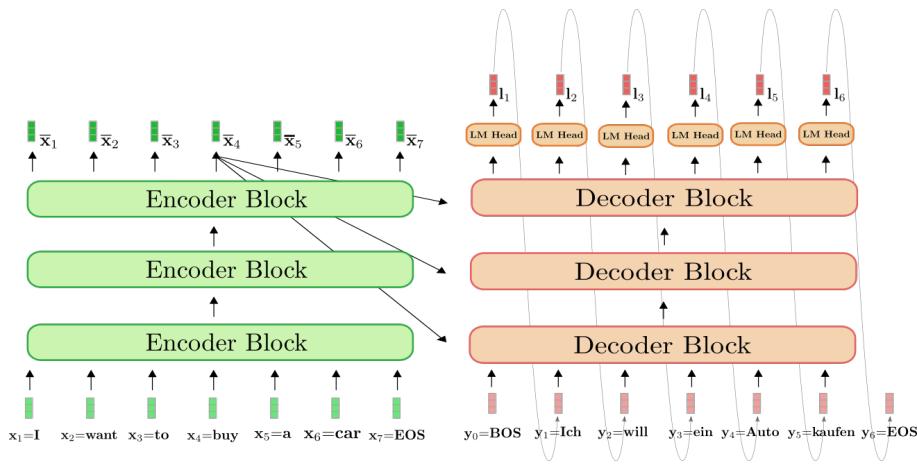


Abb. 1.1: Autoregressiven Encoder-Decoder-Generierung mit Transformer-Modellen; entnommen aus von PLATEN (2020)

1.3.2 Transformer-Architektur, BERT und BERT-ähnliche Modelle

Das Konzept von Self-Attention und der damit verbundenen Idee der Transformer-Modelle (VASWANI et al., 2017) legt die Grundlage für hochleistungsfähige Sprachmodelle. Die Architektur schlägt die Einführung von Encoder- und Decoder-Bausteinen (siehe Abbildung 1.1) in Verbindung mit sogenannten Attention-Mechanismen vor. Im Gegensatz zu beispielsweise (naiven Varianten von) klassischen NN-Implementierungen ermöglicht es dies, komplexere Abhängigkeiten innerhalb von gegebenen Textketten (Wortgruppen oder Sätzen) zu erfassen. Dabei müssen diese Attribute und Dependenzien nicht mehr gleichzeitig durch einen einzelnen Vektor inkludiert werden, sondern können unabhängig davon den Satztokens zugeordnet werden.

So ist es Modellen wie BERT (DEVLIN et al., 2019) möglich, Texte bidirektional zu verarbeiten, auf vorhergehende wie zukünftige Tokens zuzugreifen und somit den Kontext von Textdaten besser zu erfassen. BERT nutzt hierbei die Encoder-Architektur der Transformer-Theorie. Während des Pre-Trainings wird eine Maskierung jeweils einzelner Tokens vorgenommen (*Masked Language Modelling Objective*; MLM; siehe Abbildung 1.2a). Auf Grundlage der Vorhersage des maskierten Tokens werden das Modell und die Embeddings zu den Tokens trainiert. Weiterhin wird das Modell trainiert, indem die Wahrscheinlichkeit bestimmt wird, dass zwei Sätze aufeinander folgen. Das Modell sagt in einer *Next Sentence Prediction* (NSP; siehe Abbildung 1.2b) somit voraus, ob ein Satz zufällig gewählt wurde oder ob es sich um jeweils den nachfolgenden Satz handelt.

Die Veröffentlichung von BERT löste die Entwicklung einer Reihe von verwandten Sprachmodellen aus. Zum einen wurde das Wesen von BERT eingehender untersucht (*BERTology*), aber es entstanden auch Nachfolgemodelle, u. a. RoBERTa (YINHAN LIU et al., 2019), ALBERT (LAN et al., 2020), oder DistilBERT (SANH et al., 2020), welche das Konzept von BERT weiter verfeinern und etwa NSP als vermeintlich unnötigen Bestandteil aus dem Konzept tilgen. Fast zeitgleich hierzu entstand die GPT-Reihe von OpenAI (RADFORD et al., 2018), welche auf der Transformer-Decoder-Architektur basieren und aus welcher unter anderem GPT-3 (BROWN et al., 2020) und darauf basierend GPT-3.5 Turbo hervorgingen – dieses erlangte bekanntlich im Rahmen der öffentlichen Internetanwendung ChatGPT breite Popularität und sorgt bis heute für Diskussionsstoff, auch in ethischen Fragestellungen (vgl. u. a. BENDER et al., 2021; MAO et al., 2023). Vergleichbar sind die Modelle der Llama-Reihe von Meta AI, welche die GPT-Modelle in einzelnen Aspekten sogar übertreffen (TOUVRON et al., 2023a,b). Interessant ist hier jedoch vor allem die kostenfreie Veröffentlichung der Implementierungsdetails und der Modelle selbst für eine breite Öffentlich-

keit, sodass letzteres für weitere Untersuchungen besser geeignet ist. Die aktuellste Entwicklung bildet das Modell GEMINI von Google DeepMind (GEMINI TEAM & GOOGLE, 2023), welches hinsichtlich der Leistung mit GPT-4 (OPENAI, 2023) gleichgestellt wird (vgl. AKTER et al., 2023).

Spezielle Unterkategorien von Sprachmodellen, etwa sogenannte Sequence-to-Sequence-Modelle (Seq2Seq) wie BART (LEWIS et al., 2020), bieten alternative Möglichkeiten der Sprachverarbeitung. Indem Sequenzen von Texten auf andere Sequenzen abgebildet werden, wird etwa üblicherweise das Problem der maschinellen Übersetzung effektiver – im Vergleich zu früheren Verfahren – gelöst. Aber auch im Bereich des NLU lassen sich Seq2Seq-Modelle sinnvoll einsetzen, u. a. für Textparaphrasierung oder -klassifikation. BART setzt im Pre-Training Transformer-Techniken sowohl von BERT als auch von GPT ein und funktioniert bidirektional und autoregressiv (sodass eine Eingabe jeweils von der vorhergehend direkt abhängig ist). Zunächst wird ein MLM encodiert. Hierbei wird eine gesamte Untersequenz maskiert und vorhergesagt; im Gegensatz zu BERT, wo lediglich ein einzelnes Token maskiert wird. Es folgt das Durchlaufen einer Decoder-Architektur, wie sie beispielsweise von GPT verwendet wird (siehe Abbildung 1.3).

1.3.3 Fortschritte und Kritik

Sprachmodelle, insbesondere in Form von Transformer-Modellen, haben im NLP-Bereich zweifelsohne einen Durchbruch hinsichtlich Qualität der Sprachgenerierung erreicht. Ausgaben der Modelle sind teilweise nicht mehr von menschengemachten Texten zu unterscheiden – weder manuell noch in hinreichender Weise automatisch (vgl. KIRCHNER et al., 2023) – und bestehen insbesondere bei Versuchen mit Laien den TURING-Test (vgl. TURING, 1950; SRIVASTAVA et al., 2023). Dennoch liefern Sprachmodelle allein konzeptionell keine deterministischen Ausgaben. Zum einen besteht das Problem von sogenannten Halluzinationen. Versteht ein Modell eine Eingabe nicht bzw. besitzt es nicht ausreichende Datenbasis für eine korrekte Antwort, wird eine möglichst wahrscheinliche Ausgabe produziert. Diese kann nur von einer inkorrechten Antwort unterschieden werden, wenn bei der Nutzung des Systems durch einen Mensch entsprechende Fachkenntnisse vorhanden sind (für diese und weitere Risiken vgl. u. a. BENDER et al., 2021). Weiterhin ist für die meisten Modelle nur eine beschränkte Dokumentation zu formalen Vorgaben bei der Eingabe gegeben. Sogenannte ‚Prompt Engineers‘ bilden seit kurzem sogar einen eigenen Berufszweig, welcher sich mit der Untersuchung von Sprachmodellen hinsichtlich optimaler Eingaben beschäftigt, um möglichst erwartbare und korrekte Ausgaben zu erhalten. Fehlende Vorkenntnisse zum Wesen von Sprachmodellen tragen dazu bei, dass eine ‚Determinisierung‘ in der Praxis häufig aussichtslos bleibt (vgl. ZAMFIRESCU-PEREIRA et al., 2023).

Diese Probleme erstrecken sich auch auf das Thema der vorliegenden Arbeit. Wie können Sprachmodelle effektiv genutzt werden, um das Konzept von Konstruktionen zu erfassen und diese in Texten zu erkennen oder entsprechende Beispielsätze zu liefern? In den folgenden Kapiteln werden verschiedene Ansätze diskutiert und evaluiert, um die Möglichkeiten hierzu abzustecken und potenzielle Weiterentwicklungen im NLU-Bereich anzuregen, nicht zuletzt, um auch die Entwicklung von Sprachmodellen (potenziell durch einen Einfluss von Konstruktionsgrammatiken) gezielter zu optimieren.

1 Einführung

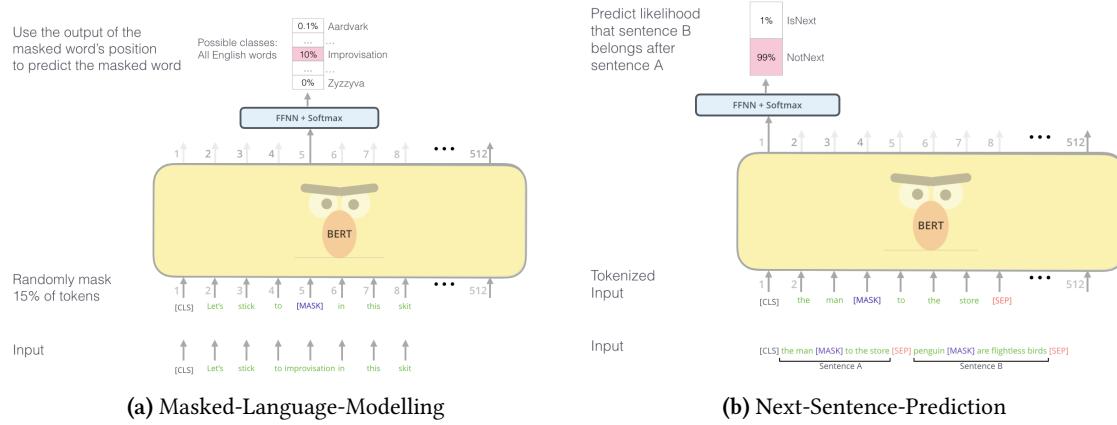


Abb. 1.2: Pre-Training von BERT; entnommen aus ALAMMAR (2018)

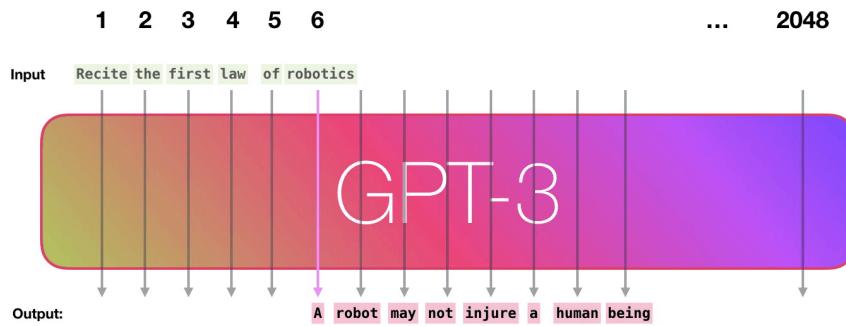


Abb. 1.3: Verarbeitung von Eingabe und Generierung der Ausgabe über das Kontextfenster von GPT-3; entnommen aus ALAMMAR (2020)

2 Vorangegangene Arbeiten

2.1 Induktion von Konstruktionsgrammatiken

Bei der Analyse von Textkorpora hinsichtlich Konstruktionen ist zwischen einer Erkennung von neuen Konstruktionen und einer Erkennung von bekannten, vordefinierten Konstruktionen zu unterscheiden. Ersteres wird im Folgenden als *Konstruktionsinduktion*, letzteres als *Erkennung von Konstruktionen* bezeichnet.

2.1.1 Manuelle Induktion

In Kapitel 1.2.2 wurden die Probleme der linguistischen Forschung bei der Induktion von Konstruktionsgrammatiken bereits kurz beleuchtet. Die klassische Herangehensweise bei der Erstellung eines Konstruktionskontraktions ist demnach die manuelle Arbeitsweise, die zwar computergestützt ausgeführt werden kann, aber nichtsdestotrotz zu einem großen Teil von menschlichen Expertinnen und Experten vorgenommen wird. Ein Beispiel hierfür stellt das *FrameNet-Konstruktikon des Deutschen* (HHU, 2024) dar, welches durch die Forschungsgruppe um ZIEM erstellt, ergänzt und gepflegt wird.

ZIEM stellt drei Voraussetzungen fest, die ein Konstruktikon trotz im Detail unterschiedlicher Auffassungen in der Literatur erfüllen sollte. Zum einen müsse ein Konstruktikon „für jede Konstruktion Bedingungen [...] definieren, die Konstrukte, also nicht-konventionalisierte sprachliche Einheiten, von Konstruktionen unterscheiden“, weiterhin „die jeweils gültigen Vererbungsbeziehungen“ festhalten und schließlich ein Maß finden, „bis zu welchem Grad Konstruktionen produktiv sind“ (ZIEM, 2014a, S. 24). Vorlage für das Konstruktikon des Deutschen sei das englischsprachige *FrameNet*-Projekt (FILLMORE et al., 2012), das auf festgelegten Relationen beruht, welche nach ZIEM eine Sprache nicht allein vollumfänglich erfassen können, aber eine erste Grundlage liefern (vgl. ZIEM, 2014b, S. 283–284). Darauf aufbauend definiert ZIEM verschiedene in Konstruktionsbeispielen zu annotierende Elemente: „Konstrukte“ (als übergeordnete Kategorie), genauer ein „Konstruktion evozierende[s] Element“ sowie „Konstruktionselemente“ (ebd., S. 285). Das einzelne konstruktionsevozierende Element wird in aktuelleren Veröffentlichungen zum Konstruktikon des Deutschen auch als KE-lex (lexikalisch festes Konstruktionselement) bezeichnet. Es löst eine Konstruktion aus, bedingt aber das Vorhandensein bestimmter Konstruktionselemente (KE); insbesondere sind sogenannte Kern-KE unerlässlich beim Zustandekommen einer Konstruktion. Für weitere Definitionen sei auf das Glossar des Konstruktikons des Deutschen verwiesen (vgl. HHU, 2024). Wie die genannten Elemente in der Praxis verwendet werden können, wird zudem in den Versuchsentwürfen der Kapitel 3.2.3 und 4.3 im Rahmen der Implementierung von Pseudowörtern (siehe Kapitel 3.2.3) näher erläutert.

BARTELD und ZIEM stellen drei Vorgehensweisen bei der Induktion eines Konstruktionskontraktions vor. Hierbei werden Konstruktionskandidaten jeweils wie folgt festgehalten (vgl. BARTELD & ZIEM, 2020, S. 6–7):

- analytisch auf Grundlage von Beispielen anderer Datensammlungen (z. B. ein anderes Konstruktikon oder Grammatikabhandlung);
- durch Herauslösen aus vollständig annotierten Textkorpora;

2 Vorangegangene Arbeiten

- mittels *Construction Mining* durch statistische Bewertung von extrahierten Mustern aus Textkorpora.

Während erstere beiden Ansätze für die genaue Arbeit mit annotierten Belegen geeignet seien, sei das Construction Mining die einzige Lösung für eine umfassende Betrachtungsweise, welchem nach BARTELD und ZIEM auch eine unterstützende Rolle bei der manuellen Annotation zukommen könne (vgl. BARTELD & ZIEM, 2020, S. 7, 13–14), wird aber momentan auch noch nicht vollumfänglich für das Konstruktikon des Deutschen angewendet.

Obwohl insgesamt die manuelle Induktion einer Konstruktionsgrammatik bzw. eines Konstruktions zeitaufwändig ist, kann zum einen sichergestellt werden, dass intuitiv interessante Sprachphänomene durch Konstruktionsdefinitionen repräsentiert werden können. Zum anderen kann durch mehrere personelle Instanzen die Korrektheit der Annotationen gewahrt werden. Zudem fehlen bei einer rein maschinellen Induktion die Festlegungen, wie komplexe Voraussetzungen einer Konstruktion zu interpretieren bzw. zu berücksichtigen sind. Außerdem gestaltet sich auch die maschinelle Induktion als eine aus Komplexitätstheoretischer Sicht schwierige Herausforderung, wie im nachfolgenden Kapitel 2.1.2 erörtert wird.

2.1.2 Maschinelle Induktion

Neben der klassischen manuellen Induktion durch menschliche Annotation gibt es auch vereinzelte Versuche einer rein maschinellen Erzeugung von Konstruktionsgrammatiken. Insbesondere DUNN beschäftigt sich in einer Reihe von Veröffentlichungen mit der Definition einer *Computational Construction Grammar* (C2xG), deren Konzept darauf beruht, mittels Verfahren wie Beam-Search große einsprachige Korpusdatensätze iterativ zu durchlaufen Muster zu erkennen, zu bewerten und in Form einer konstruktionsähnlichen Annotation zu definieren (vgl. DUNN, 2017a,b, 2018a,b,c, 2019, 2023a,b).

Um maschinell induzierte Konstruktionsgrammatiken auszuwerten, evaluiert DUNN (2018b) in seiner Arbeit grundsätzlich mithilfe zweier Werte, die im folgenden kurz erläutert werden: zum einen die Berechnungskomplexität von Konstruktionsgrammatiken, welche möglichst minimiert werden sollte; zum anderen die Beschreibungsadäquatheit von Konstruktionsgrammatiken in Bezug auf eingegebene Corpora, welche nicht annotiert sind – diese soll entsprechend maximiert werden. Um diese Werte miteinander zu verknüpfen, kommt hierbei die Methode der *Minimum Description Length* (MDL) zum Einsatz:

$$\text{MDL} = \min_G \{L_1(G) + L_2(D|G)\}. \quad (2.1)$$

Hierbei steht $L_1(G)$ für die Größe¹ der kodierten Grammatik G (und somit die Komplexität der Grammatik), $L_2(D|G)$ hingegen für die Größe des Datensatzes D , welche von einer bestimmten Grammatik G kodiert wird (die Beschreibungsadäquatheit). Zusätzlich nutzt DUNN (ebd.) ein $L_3(G)$, die Größe der Kodierung, welche benötigt wird, um mit dem universellen Code die Größe der Grammatik G zu bestimmen. Er beschreibt nicht, wie und an welcher Stelle dieser Parameter verwendet wird, aber sieht diesen Wert als notwendig an, wenn die Qualität einer Grammatik gegen die Ausgangswerte einer unkodierten Testmenge abgewogen werden soll.

Um zwei Grammatiken in ihrer Konfidenz zu vergleichen, wird der Betrag der Differenz aus beiden MDL-Werten gebildet. Als Konfidenz ist in diesem Zusammenhang die geschätzte Zuverlässigkeit bzgl. der Aussagekraft einer Grammatik zu verstehen. Soll hingegen ein Vergleich zu

¹Die Größe wird in Form eines negativen natürlichen Logarithmus betrachtet.

einem nicht-kodierten Datensatz gezogen werden, wird die Formel

$$1 - \frac{\text{MDL}(G)}{\text{MDL}(U)} \quad (2.2)$$

verwendet, sodass eine höhere Kompression durch G vorliegt, wenn der Wert näher an 1 liegt. Mittels dieser Methode solle verhindert werden, dass sowohl MDL als auch Signifikanz von der Größe einer bestimmten kodierten Testmenge abhängig sind.

U entspricht der nicht kodierten MDL, also der Datengröße ohne Kodierung ($U = \text{unencoded}$). Die Formel wird verwendet, um die Kompression durch eine Grammatik G im Vergleich zur unkomprimierten bzw. nicht kodierten Repräsentation zu bestimmen. Je näher der Wert an 1 liegt, desto stärker die Kompression der Sprache durch die Grammatik G .

Eine Differenz von 0 bedeutet, dass die Grammatiken ungefähr gleich gut komprimiert vorliegen; bei einer größeren Differenz stellt eine der beiden Grammatiken eine Verbesserung gegenüber der anderen Grammatik dar, was ihre Größe anbelangt („[...] to what degree is G_A better than G_B ?“ (ebd., S. 83)). Die Konfidenz ist hier gewissermaßen als Anreiz zu verstehen, die Grammatiken gegenüber dem nicht kodierten Korpus nach und nach zu verbessern.

Im weiteren untersucht DUNN (ebd.) die Stabilität von Grammatiken einer Sprache über unterschiedliche Neustarts seines Algorithmus. Er diskutiert, wie eine Konstruktionsgrammatik entsprechend optimiert werden kann und vergleicht Evaluationen über verschiedene europäische Sprachen hinweg.

Grundsätzlich unterscheidet DUNN (ebd.) beim Lernprozess drei Stufen von Konstruktionsgrammatiken: CxG_{LEX} (nur lexikalische Repräsentationen), CxG_{SYN} (nur syntaktische Repräsentationen) und CxG_{FULL} (Repräsentationen auf lexikalischer, syntaktischer und semantischer Ebene), wobei Konstruktionen aus den jeweils vorangegangenen Durchläufen im nächsten Durchlauf als atomare Einheiten betrachtet werden. Somit soll verglichen werden, ob es sich lohnt, eine jeweils höhere Ebene von Konstrukten zu betrachten.

Auch nach dem Zusammentragen von Konstruktionsdaten stellt sich die Frage nach der Wichtigkeit verschiedener Muster bzw. Konstruktionskandidaten. Zu kritisieren sind bei DUNN zum einen die teils willkürlich erscheinenden Festlegungen bzw. Vereinfachungen, die nötig sind, um auf diese Weise eine Abstraktion der Textdaten zu versuchen. Einerseits versucht der Autor, durch die Definition und Nutzung verschiedener Assoziationsmetriken auf Grundlage der so genannten ΔP -Metrik (vgl. DUNN, 2017b) und der *Minimum Description Length* bzw. MDL (vgl. DUNN, 2019) eine Grundlage für eine sinnvolle Sortierung der Konstruktionskandidaten festzulegen, zum anderen herrscht in der Literatur Uneinigkeit über eine aussagekräftige Art, Konstruktionsgrammatiken zu bewerten. Im Speziellen die unzureichende Evaluierung von semantischen Konstrukten wird jedoch kritisiert (vgl. RAMBELL et al., 2019).

Andere Forschungsgruppen bieten hier Varianten an, die auf anderen Theorien fußen. In der Literatur werden dafür verschiedene Metriken vorgeschlagen, anhand derer interessante² Konstruktionen in einer möglichen Reihenfolge sortiert werden können. Erwähnt seien an dieser Stelle frühe Konzepte zur Anwendung der hypergeometrischen Verteilung als Wahrscheinlichkeit, welche die Häufigkeit einer Sequenz im Korpus auf Basis ihrer enthaltenen Formen abschätzt (vgl. LAFON, 1984, S. 54–64) und der *C Formula* (GEFFROY et al., 1973), welche die Nähe von Kollokationen zueinander metrisch erfasst. Spätere Vorschläge umfassen die *Cost Criteria* (KITA et al., 1994) als eine näherungsweise Modellierung des menschlichen Aufwands beim Sprachenlernen und schließlich *UIF-PMI* (FORSBERG et al., 2014; BARTELD & ZIEM, 2020), als Erweiterung

²Was genau mit interessant bzw. „interesting“ (u. a. KITA et al., 1994, S. 24) gemeint ist, wird in der Literatur kaum geklärt und führt somit zu stark differierenden Auffassungen über eine vermeintlich korrekte Bewertung von Konstruktionsgrammatiken.

2 Vorangegangene Arbeiten

der *Pointwise Mutual Information*. Alle Gruppen untersuchen somit ihrerseits ebenfalls Möglichkeiten, Kollokationen und Konstruktionskandidaten in eine Rangfolge zu bringen und somit relevante Konstruktionen in einer maschinell induzierten Konstruktionsgrammatik zu erfassen. Allerdings lässt sich feststellen, dass keine der Vorschläge den Anspruch erhebt, eine allgemeingültige Aussage über die Relevanz eines bestimmten Konstruktionskandidaten zu treffen. Hier ist es vermutlich nötig, in Zukunft eine Kombination verschiedener Metriken zu erstellen, sodass aus einer großen Anzahl verschiedener Konstruktionskandidaten sinnvoll ausgewählt und somit ein Konstruktikon festgelegt werden kann.

Unabhängig von Kritik an den Verfahren nach bspw. DUNN im Einzelnen merken andere Forschungsgruppen an, dass im Allgemeinen bei einer ausführlichen und möglichst holistischen Analyse von Korpusdaten die algorithmische NP-Komplexität zu einer nicht zu unterschätzenden Problematik bezüglich der Laufzeit führe, sodass heuristische Herangehensweise angebracht seien (vgl. BARTELD & ZIEM, 2020; XU et al., 2023). Aktuelle Konzepte untersuchen bereits Sprachmodelle wie GPT-4 hinsichtlich ihrer potentiellen Fähigkeit, aus gegebenen Texten Definitionen für Konstruktionen bereitzustellen (TORRENT et al., 2023), wobei jedoch ebenfalls die probabilistische Unschärfe und der Nichtdeterminismus von Sprachmodellen beachtet werden muss.

Obwohl ein maschinelles Konstruktikon sicherlich direkter durch weitere NLP-Verfahren verarbeitet werden könnte, wird in dieser Arbeit auf eine manuell induzierte Sammlung von Konstruktionen – speziell das *FrameNet-Konstruktikon des Deutschen* – zurückgegriffen. Zwar lässt sich hierbei eine gewisse Subjektivität bei der Auswahl nicht vermeiden, die Qualität der Annotation und Auswahl der Konstruktionen ist allerdings bisweilen höher als bei bisherigen maschinellen Verfahren.

2.2 Erkennung von Mustern und Konstruktionskandidaten in Textdaten

Zur Untersuchung von Konstruktionen mittels Techniken der maschinellen natürlichen Sprachverarbeitung (NLP) wurden bereits einige Vorleistungen erbracht, welche sich insbesondere mit einzelnen sprachlichen Phänomenen beschäftigen.

Der Bedarf in der linguistischen Forschung besteht, spezielle Sprachphänomene in großen Textkorpora mittels NLP-Methoden und Sprachmodellen ausfindig zu machen. Zunächst beschränkte sich die Filterung von Korpora auf die Nutzung von klassisch-regelbasierten Verfahren. SHLAIN et al. stellen das *SPIKE*-System vor, mithilfe dessen syntaktische Muster, welche in einer beliebigen Textsequenz gefunden wurde, in anderen Teilen eines Dokuments oder Korpus wieder auffindbar sind. Durch eine spezielle Anfragesprache soll dieses Programm etwa bei der menschlichen Annotation unterstützend agieren. Hierbei wird angemerkt, dass sich SPIKE auch auf nicht-syntaktische Annotationen (etwa Semantik) erweitern ließe, es werde dazu allerdings ein akkurrater Parser benötigt (vgl. SHLAIN et al., 2020). Somit trennt dieser Ansatz Syntax und Semantik voneinander, was den inzwischen weitgehend widersprochenen generativen Ansichten von CHOMSKY (siehe Kapitel 1.2.1) gleichkommt. Dennoch wird auch im Rahmen dieser Arbeit eine Vorfilterung von Textdaten mit einem ähnlichen Grundgedanken vorgenommen, da diese Ansätze vor allem sehr ressourcenschonend auszuführen sind.

Andere Beiträge untersuchen bereits den Nutzen von Sprachmodellen für die Verarbeitung von linguistischen Konstruktionen. Für die englische Sprache betrachten unter anderem WEISSWEILER et al. die *Comparative Correlative*-Konstruktion (CC) daraufhin, ob Sprachmodelle diese einerseits syntaktisch erkennen, andererseits semantisch verstehen können (WEISSWEILER et al., 2022). Die syntaktische Untersuchung erfolgt mittels Analyse von Minimalpaaren aus Positiv- und Negativbeispielen, wobei Negativbeispiele sowohl synthetisch durch Wortpermutationen

aus Positivbeispielen erzeugt als auch korpusbasiert mithilfe von spaCy-POS-Tagging (HONNIBAL et al., 2020) in einem Korpus gefunden werden. Es stellt sich heraus, dass die untersuchten Sprachmodelle (u. a. BERT) kaum Probleme bei der korrekten Erkennung der CC-Konstruktion haben, allerdings tendenziell auf synthetischen Minimalpaaren bessere Leistungen erbringen. Dennoch zeigt sich bei der Untersuchung des tatsächlichen semantischen Verständnisses der Konstruktion über das MLM, dass BERT-basierten Modellen größere Probleme bei der Lösung dieser Aufgabe nachzuweisen sei.

Weitere Analysen in der Literatur umfassen etwa eine Konstruktion, in welcher ihrerseits ein unbestimmter Artikel, ein Adjektiv, ein Numeral und ein Substantiv aufeinander folgen (MISRA, 2023). Das Fine-Tuning auf Basis von BERT ist hierbei inspiriert durch das von POTTS diskutierte *Preposing in PP (PiPP)* (POTTS, 2024). Der Autor durchsucht dabei Korpora nach einem Algorithmus, der auf einem relativ allgemein gehaltenen regulären Ausdruck beruht (vgl. ebd., S. 16). Dadurch sollen mögliche Beispiele im Korpus weiträumig abgesteckt werden, was in Kapitel 3.2.1 in ähnlicher Weise durchgeführt wird. Allerdings erfolgt die genauere Analyse durch den Autor manuell und es wird versucht, daraus eine Aussage für die Häufigkeit der Konstruktion für die gesamte Sprache zu treffen. Dies dürfte mit einer starken Unschärfe einhergehen, obwohl POTTS versichert, die tatsächliche Häufigkeit durch ein 95-prozentiges Konfidenzintervall einzuschränken. Als Alternativmethode schlägt der Autor neben Untersuchungen bzgl. verschiedener GPT-3-Unterversionen³ ein Fine-Tuning von BERT auf Basis des MLM vor, wobei jeweils die festen Lexeme *though* bzw. *as* der PiPP-Konstruktion mittels MLM maskiert und vorhergesagt werden. Auf diese Weise kann POTTS durch ein entsprechendes Klassifizierungsmodell das Vorhandensein einer PiPP-Konstruktion bewerten. In den Kapiteln 3.2.3 und 4.3 wird dieser Gedanke aufgegriffen, auf die lexikalisch festen Elemente (KE-lex) anderer Konstruktionen erweitert und durch Pseudowort-Embeddings umgesetzt.

Darüber hinaus wurden in gewissem Rahmen auch spezielle Konstruktionen für andere Sprachen hinsichtlich ihrer automatisierten Verarbeitung untersucht, etwa die Fähigkeiten von Übersetzungsmodellen zum Verständnis der *Caused-Motion*-Konstruktion im Deutschen, Italienischen und Türkischen. Nicht nur stellen sich die Ausgaben als semantisch fehlerhaft heraus, was auf geringes Wissen der entsprechenden Modelle bzgl. Konstruktionen vor allem nicht-englischer Sprachen zurückzuführen ist. Es ist auch zu erkennen, dass sich die Literatur vorwiegend auf das Englische zu fokussieren scheint,⁴ sodass für andere Sprachen selbst mit großer Popularität die Evidenz bzgl. Konstruktionen und Sprachmodellen noch sehr gering ist. Dies ist einer der Gründe, warum sich die vorliegende Arbeit explizit dem Deutschen als Ein- und Ausgabesprache widmet, wenngleich diese Sprache noch relativ häufig in aktueller Literatur behandelt wird. Momentane Entwicklungen lassen darauf hoffen, dass auch Sprachfamilien, die seltener in der Literatur vertreten sind und eine große Basis an Sprecherinnen und Sprechern besitzen (z. B. austroasiatische, afroasiatische oder Niger-Kongo-Sprachen) – aber auch kleinere Sprachen – stärker in den Fokus gerückt werden. Allerdings besteht hier häufig das Problem, dass annotierte Daten (etwa ein Konstruktikon) seltener in umfangreicher Menge vorhanden sind. Dies führt zurück zu dem Bedarf an einer maschinellen Konstruktionsinduktion, wie sie in Kapitel 2.1.2 erörtert wird.

2.3 Generierung neuer Beispiele für gegebene Konstruktionen

In Kapitel 2.1.1 wird eine mögliche manuelle Herangehensweise diskutiert, mit der Korpora auf Vorkommnisse bestimmter Konstruktionen untersucht werden können. Hierbei werden im Allgemeinen Textsequenzen aus dem zeitgenössischen, schriftlichen Sprachgebrauch als Datengrundlage verwendet, beispielsweise Zeitungs- und Zeitschriftenartikel, Internetdokumente usw.

³Hierbei kritisiert POTTS die fehlende Veröffentlichung interner Architekturbeschreibungen.

⁴Für eine Übersicht bisheriger Untersuchungen vgl. WEISSWEILER et al., 2023.

2 Vorangegangene Arbeiten

Die vorliegende Arbeit beschäftigt sich unter anderem damit, wie auch neue Beispiele generiert werden können, um etwa ungewöhnliche oder weniger häufige Konstruktionen linguistisch zu untersuchen. Diese Aufgabenstellung wird in der Literatur bisher nur selten besprochen.

TORRENT et al. untersuchen GPT-4 und *OpenAssistant* auf ihre Fähigkeit, solche Beispiele in Form einer Chat-Antwort zurückzugeben. Betrachtet werden ausschließlich englische Konstruktionen, ihrerseits die *un-VERB*-Konstruktion (*unravel, unzip*), sowie das Phänomen der *Caused-motion frames* sowie die *XYZ*-Konstruktion (*X is the Y of Z*). Hierfür nutzen sie eine Definition der jeweiligen Konstruktion als Eingabe in das Sprachmodell. (TORRENT et al., 2023) TORRENT et al. halten jedoch fest, dass es bei ihren Ausführungen nicht um „understanding of language“ (ebd., S. 1) gehe. Gerade ein Modell, das wie GPT-4 mit großen Datenmengen trainiert wurde, zeigt teils nur geringe Kreativität bei den generierten Beispielen (vgl. ebd., S. 11, 14–15), wie ein Abgleich mit einer hohen Anzahl an Treffern bei der Google-Suche für die Ausgaben nahelegt. Somit ist davon auszugehen, dass hierunter viele Wiederholungen aus den Trainingsdaten sind. Außerdem stellt die Gruppe einen deutlichen Qualitätsverlust fest, sobald eine andere Sprache als Englisch betrachtet wird.⁵

2.4 Spezialisierung und Erweiterung von Sprachmodellen hinsichtlich NLU

Wie Sprachmodelle intern Sprachverständnis organisieren, ist eine Fragestellung, die aus verschiedenen Richtungen der aktuellen Forschung betrachtet wird. Sie hat zum Ziel, die Struktur der Modelle gezielt auszunutzen, um letztere weiter zu optimieren und auch komplexere oder ungewöhnliche Muster (oder, im weiteren Sinne, Konstruktionen) zuverlässig verarbeiten zu können.

BARONI stellt bereits vor dem Aufsehen um ChatGPT und ähnliche Modelle fest, dass Sprachmodelle beeindruckende Sprachkompetenz beweisen, wobei sie nicht kompositionell zu sein scheinen und ihre Fertigkeiten über einfache Mustererkennung hinausgehe. Allerdings hätten die untersuchten Modelle schon bei kleinen token-internen Fehlern (z. B. Tippfehler) Schwierigkeiten beim Verständnis der Satzstruktur. Er schlägt vor, das Training von Sprachmodellen mit kompositionellen Regeln zu ergänzen, mahnt aber auch an, mögliche alternative Konzepte zu untersuchen, die von Sprachmodellen intrinsisch erlernt werden. (BARONI, 2020)

Diese Ansicht wird durch Wu et al. unterstützt. Sie untersuchen, inwiefern Sprachmodelle tatsächlich kein rein oberflächliches, sondern ein ontologisches und, in der Folge, generalisiertes semantisches Verständnis besitzen. Dabei wird festgestellt, dass gewisse ontologische Konzepte durch Modelle wie BERT nur zu einem gewissen Grad verstanden werden. Dabei wenden sie unter anderem das Konzept der *Pseudowörter* an. Diese neu trainierten Pseudowörter ergänzen das BERT-Modell um Tokens, die bestimmte ontologische Eigenschaften kodieren sollen. Somit kann überprüft werden, wie gut das Modell Sprachstrukturen tatsächlich erfasst, während oberflächliche Inferenzen auf Grundlage der Trainingsdaten vermieden werden. (WU et al., 2023)

Pseudowörter stellen hierbei eine Semantik dar, die von real existierenden Wörtern bzw. Tokens unabhängig repräsentiert werden. Anhand von Homonymen wie beispielsweise *Bank* als Sitzgelegenheit und *Bank* als Geldinstitut könnte je ein Pseudowort konstruiert werden, das ausschließlich die jeweilige Bedeutungsvariante verkörpert. In manchen Anwendungsfällen können Pseudowörter aber auch mehrere Bedeutungen einzelner Wörter umfassen. Denkbar wäre ein Pseudowort, das ein Wort einer anderen Sprache modelliert; z. B. *indeed*, dessen komplexe Bedeutung im Deutschen nur durch Begriffe wie *freilich, allerdings* und *zwar* angenähert werden kann. Hier könnte ein Pseudowort kreiert werden, welches alle Nuancen von *indeed* für ein deut-

⁵TORRENT et al. untersuchen Portugiesisch in der brasilianischen Varietät.



Abb. 2.1: Illustration der MaPP-Methodik zum Training von Pseudowort-Embeddings; entnommen aus KARIDI et al. (2021)

sches Modell durch Verschmelzung der Bedeutungen möglicher Übersetzungen verkörpert (vgl. SCHÜTZE, 1998, S. 107). Im Rahmen dieser Arbeit wird auf diese verallgemeinernde Variante von Pseudowörtern jedoch nicht weiter eingegangen.

Als Werkzeug des NLU sind Pseudowörter allerdings nicht neu – sie wurden und werden in verschiedenen Bereichen eingesetzt. Frühere Untersuchungen nutzen diese unter anderem, um Algorithmen zur Disambiguierung von Wortbedeutungen zu evaluieren (ebd.). KARIDI et al. (2021) untersuchen hingegen, wie mittels Pseudowörter BERT-Tokens trainiert werden können, die einzelne semantische Bedeutungen anderer, mehrdeutiger BERT-Tokens innehaben (siehe Abbildung 2.1). Sie betrachten hierbei vor allem Präpositionen und ausgewählte Verbformen und erreichen schon mit wenigen Trainingsdaten (ca. fünf Sätze pro Pseudowort) eine nachweisbare Disambiguierung der jeweiligen Teilbedeutungen. Grundkonzept ist es hierbei, jeweils ausschließlich Beispielsätze zu verwenden, welche die entsprechende Bedeutungsnuance des zu disambiguierenden Tokens eindeutig beschreibt. Es werden dann vorübergehend Pseudowort-Embeddings für jeden Beispielsatz trainiert, deren Durchschnitt anschließend berechnet wird. Gemäß KARIDI et al. wird der aggregierte Loss am Beispiel von BERT dann wie folgt definiert:

$$\mathbf{z}_\eta^* = \arg \min_{\mathbf{z} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \left\| BERT(\mathbf{z}) - \mathbf{x}_t^{(i)} \right\|^2, \quad (2.3)$$

wobei \mathbf{z}_η^* ein Pseudowort-Embedding \mathbf{z}^* mit einer Bedeutung η darstellt, berechnet durch das arithmetische Mittel der Differenz aus dem ursprünglichen BERT-Embedding \mathbf{z} und verschiedenen Fokustokens $\mathbf{x}_t^{(i)}$, welche jeweils durch ein bestimmtes anderes Token j im Satz disambiguier werden. Das Token j wird durch das MLM von BERT maskiert, wodurch \mathbf{z}^* trainiert wird. Die Gruppe strebt das Finden eines Pseudowort-Embeddings auf Basis aller t an, für das *specialization* (d. h. es wird ein spezifischer Sinn von t ausgedrückt) und *generalization* (d. h. kann dieser spezielle Sinn von t auch in anderen Kontexten erkannt/konstruiert werden) gelten. Die beschriebene Technik wird von KARIDI et al. (ebd.) als „*MaPP (Masked Pseudoword Probing)*“ benannt.

Als Beispiel sei die Disambiguierung der englischen Präposition *in* erläutert. Diese kann wahlweise eine temporale (z. B. *in October*) oder lokale Bedeutung (z. B. *in London*) innehaben. In-

2 Vorangegangene Arbeiten

dem wahlweise die Zeitangabe oder der Ort in entsprechenden Beispielsätzen maskiert werden, kann für beide semantischen Nuancen je ein Pseudowort-Embedding trainiert werden. Bei der Evaluierung sagt das Modell für eine Maskierung dann ebenfalls bevorzugt eine Zeit oder eine Lokalität an, wenn das originale BERT-Embeddings durch das Pseudowort-Embedding ersetzt wird.

Diese auf Sprachmodelle wie BERT zugeschnittene Interpretation von Pseudowörtern wird auch im weiteren Verlauf der vorliegenden Arbeit angewendet, indem sie für die Konstruktionen aus dem Konstruktikon des Deutschen (beschrieben in Kapitel 2.1.1) abgewandelt wird. Somit können mögliche Ergänzungen von Sprachmodellen hinsichtlich gegebener Konstruktionen auf ihre Effektivität untersucht werden. Das Konzept muss allerdings in einigen Punkten angepasst werden, da einzelne Disambiguierungen der Pseudowörter beispielsweise nicht nur Einzeltokens bzw. -wörtern, sondern häufig ganzen Satzsequenzen entsprechen können. Diese Problematiken und potenzielle Lösungsmöglichkeiten werden in Kapitel 3.2.3 erörtert.

3 Erkennung von Konstruktionen in gegebenen Texten

3.1 Problemanalyse bei der Konstruktionsdetektion

In Kapitel 2.2 wurden bereits Arbeiten vorgestellt, die verschiedene Vorleistungen bei der algorithmischen Konzeption zur Erkennung von Konstruktionsbeispielen in Textkorpora erbringen. Hierbei wird jedoch meist eine Fokussierung auf nur eine spezielle oder wenige Konstruktionen vorgenommen. Die vorliegende Arbeit verfolgt unter anderem den Zweck zu analysieren, welche Konzepte zu einer allgemeingültigeren Erkennung verschiedenster Konstruktionen zielführend sind. Dabei werden folgende Rahmenbedingungen und Ziele definiert:

- Eine Menge von Konstruktionen (bzw. ein Konstruktikon) mit Definition und Beispielsätzen für jedes Element sei als gegeben angenommen.
- Weiterhin steht ein Korpus von Texten zur Verfügung, welcher eine möglichst alltägliche Sprache (und nicht etwa reine Fachtexte) beinhaltet.
- Sind sowohl ein Dokument des Korpus als auch eine Konstruktion des Konstruktikons gegeben, soll nun herausgefunden werden, welche Sequenzen des Textes der gegebenen Konstruktion entsprechen. Diese Sequenzen werden als Ergebnis je Dokument und Konstruktion zurückgegeben.

Datenbasis für die Untersuchung stellt zum Einen das in Kapitel 2.1.1 vorgestellte *FrameNet-Konstruktikon des Deutschen* (HHU, 2024) dar, welches neben Definitionen zu jeder Konstruktion auch einige Beispiele enthält. Die Konstruktionen sind im Anhang in Tabelle B.1 nachzuvollziehen. Zum anderen wird das HDT-UD-Korpus (BORGES VÖLKER et al., 2019) als allgemeine Textbasis genutzt.

Das aktuell bekannteste Sprachmodell stellt wohl, wie in vorigen Kapiteln bereits erwähnt, ChatGPT dar. OpenAIs GPT-3.5 Turbo und GPT-4 (BROWN et al., 2020; OPENAI, 2023) stehen allerdings momentan (Stand Januar 2024) per Programmierschnittstelle nur im Abonnements zur Verfügung, was die Arbeit mit diesem Modell im Rahmen dieser Arbeit stark einschränkt. Eine Alternative ist das relativ aktuelle Modell Llama 2 (TOUVRON et al., 2023b), das durch Meta AI herausgegeben wird und nicht unter den genannten Problemen leidet. Die in dieser Arbeit betrachteten Sprachmodelle seien dementsprechend BERT (cased) und Llama 2 jeweils in einer Variante, welche ein Fine-Tuning auf das Deutsche beinhaltet (DBMDZ, 2023; JOBBINS, 2023); zudem mBART-50 (TANG et al., 2021), also BART in einer multilingualen Version, welches bereits in hinreichender Weise mit Daten verschiedener Sprachen trainiert wurde. Die hier gewählte Auswahl hat den Hintergrund, einen Vergleich verschiedener Modelle in Bezug zu ihren Arbeitsweisen und Architekturen ziehen zu wollen, wobei eine Zugänglichkeit bzw. ein kostengünstiges Training auf Basis der gewählten Modelle vorausgesetzt wird. Gleichzeitig sollen verschiedene Modelle mit unterschiedlichen Fähigkeiten berücksichtigt werden.

In den folgenden Abschnitten des vorliegenden Kapitels werden verschiedene Möglichkeiten diskutiert, die beschriebenen Zielsetzungen zu erreichen. Bevor allerdings differenzierte Verfahren diskutiert werden, soll zunächst die Leistung bereits bestehender Sprachmodelle hinsichtlich der Unterscheidung von positiven und negativen Beispielen bzgl. einer Konstruktion untersucht werden. Auf diese Weise kann eine Grundlage festgestellt werden, anhand derer andere Ansätze evaluiert werden sollen.

3.2 Konzeption der Untersuchung hinsichtlich Konstruktionserkennung

Zunächst wird das Konzept vorgestellt, das für die Analyse verschiedener Sprachmodelle auf die Fähigkeit der Konstruktionserkennung hin implementiert werden soll. Dabei wird zunächst auf eine Vorverarbeitung mithilfe von Dependenzparsing eingegangen, die für alle in der Folge betrachteten Modelle gleichermaßen gültig ist. Danach werden zunächst bestehende Sprachmodelle u. a. mit Prompt-Engineering-Verfahren behandelt. Schließlich werden diese Varianten durch Ansätze basierend auf Pseudowörtern ergänzt.

3.2.1 Vorverarbeitung mit *Universal Dependencies*

Unabhängig vom verwendeten Korpus ist davon auszugehen, dass eine genaue Analyse mittels verschiedener Methoden zu umfangreich für die üblichen Textmengen wäre. Aus diesem Grund sollen zunächst regelbasiert die gegebenen Sätze vorgefiltert werden. Als Textgrundlage wird hier das HDT-UD-Korpus (BORGES VÖLKER et al., 2019) herangezogen. Dieses Korpus ist eine Sammlung von Sätzen in deutscher Sprache, welche vorwiegend aus Artikeln aus technischen Kontexten stammen.¹ Diese werden durch Annotationen gemäß *Universal Dependencies* – kurz: UD – (DE MARNEFFE et al., 2021) angereichert, wobei letztere ein Framework zur morphosyntaktischen Annotation von Sätzen bilden. Dabei werden Dependenzrelationen jeweils als Tripel aus Kopf, Relation und Dependent annotiert. Genauer wird jedem Token im Satz der Index des zugehörigen Kopfs vermerkt, der im Sinne einer ebenfalls beschriebenen Relation zum Dependent in Beziehung steht.

Während also beispielsweise POTTS (2024) zum Vorfiltern einen regulären Ausdruck verwendet, mit welchem vor allem die sogenannten lexikalisch festen Kern-Konstruktionselemente (KE-lex) abgeglichen werden (siehe Kapitel 2.2), kann aufgrund der Nutzung von UD hier die gegebene Annotation aus dem Konstruktikon des Deutschen verwendet werden, welche ebenfalls UD-Tags umfasst. Hierin sind insbesondere die folgenden Daten für die Worttokens der Beispielsätze annotiert:

- (U)POS-Tags (Beschreibung der syntaktischen Rolle im Satz),
- XPOS-Tags (erweiterte und spezifischere POS-Tags),
- Dependenzrelationen (auf welche Art zwei Tokens morphosyntaktisch verknüpft sind),
- Dependenz-Kopf (*HEAD*; zeigt das Ziel der Dependenz an).

Gegebenenfalls können fehlende oder unvollständige Annotationen mithilfe von spaCy (HONNIBAL et al., 2020) ergänzt werden. In der praktischen Umsetzung werden aus Gründen der einfacheren Implementierung alle Einträge durch spaCy annotiert; in zukünftigen Weiterentwicklungen sollte jedoch für eine bessere Genauigkeit, sofern möglich, die menschliche Annotation im Konstruktikon bevorzugt und entsprechend konvertiert werden.

¹Dies widerspricht zum Teil der Vorgabe aus Kapitel 3.1, allerdings sind die Texte trotzdem für ein breites Publikum verfasst.

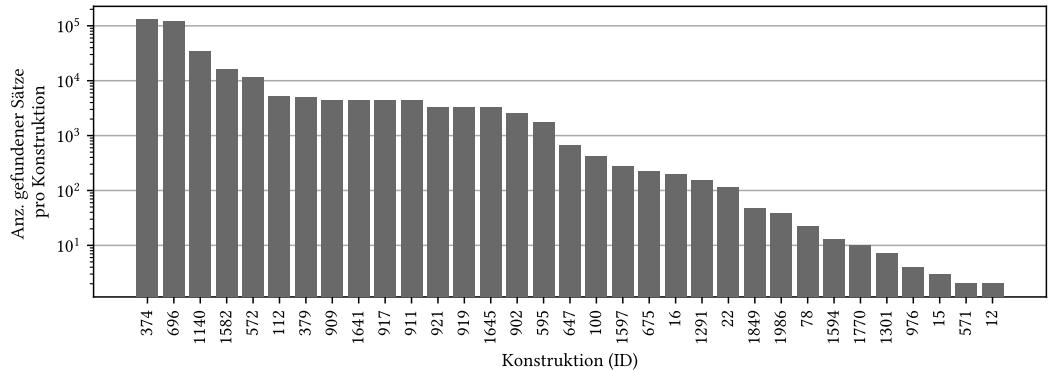


Abb. 3.1: Anzahl gefundener Sätze als potenzielle Beispiele bei der Vorfilterung mittels *Universal Dependencies* (logarithmische Skala) pro Konstruktion (Index gemäß Konstruktikon)

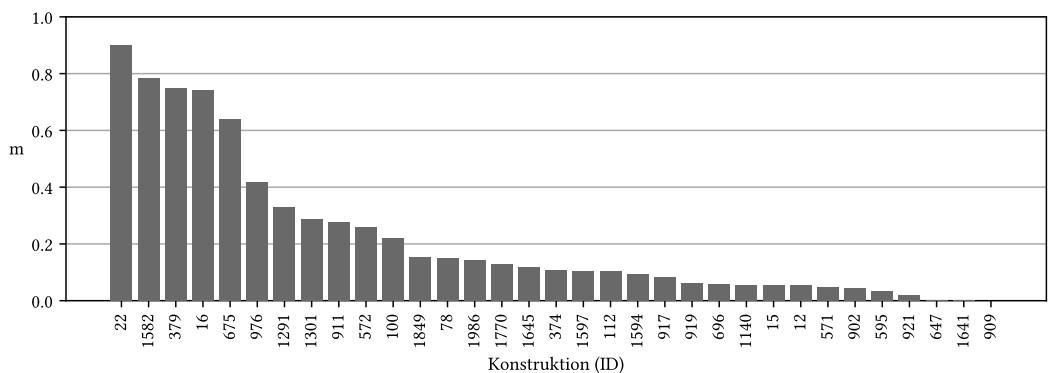


Abb. 3.2: Rangfolge von Konstruktionen gemäß durchschnittlicher Zugehörigkeit m der gefundenen potentiellen Beispiele

Nun werden für jede Konstruktion die gegebenen Beispielsätze des Konstruktikon durchlaufen. Dabei werden die annotierten Sätze auf gemeinsame Dependenz-Tripel, bestehend aus zwei Worttokens und ihrer Relation zueinander, miteinander verglichen. Die Worttokens werden hierbei alternierend durch die lexikalische Form, POS-Tag und XPOS-Tag repräsentiert. Es wird dann zwischen den Beispielsätzen überprüft, ob gemeinsame Tripel vorkommen. Diese werden gesammelt und somit eine Schnittmenge der Dependenztripel erstellt, die in allen Beispielsätzen einer gegebenen Konstruktion gleichermaßen vorkommen. Ziel ist, zumindest eine grobe, rein syntaktische Annäherung an die Voraussetzungen einer Konstruktion zu erreichen.

Bei der Betrachtung des allgemeinen Textkorpus (hier HDT-UD) werden dann für jede Konstruktion zunächst alle Sätze aus dem Korpus herausgelöst, welche eines der im Konstruktikon definierten KE-lex enthalten. Da ein Vorhandensein dieser eine notwendige Bedingung zum Zustandekommen einer Konstruktion darstellt und außerdem die meisten Konstruktionen ein solches KE-lex besitzen, ist dies bereits eine einfache und schnelle erste Abgrenzungsmöglichkeit plausibler Sätze. Die zuvor definierten Schnittmengen von Dependenztripeln können dann auf diese Sätze angewendet werden, wobei die HDT-UD-Sätze ebenfalls bereits eine Annotation mit den oben beschriebenen Merkmalen anbieten.

An zwei Stellen entstehen hierbei jedoch Probleme. Zum einen kommt es vor, dass sich zu Konstruktionen tatsächlich keine gemeinsamen Dependenztripel aus den Konstruktikon-Beispielsätzen extrahieren lassen. Aber auch bei einer erfolgreichen Extraktion werden teilweise keine oder nur selten Sätze im untersuchten Textkorpus gefunden, da diese in der Praxis möglicherweise zu

3 Erkennung von Konstruktionen in gegebenen Texten

spezifisch sind und durch Sprachvariation zu selten mit realen Sätzen kongruieren. Annotationsfehler im Konstruktikon sowie im Textkorpus können ebenfalls problematisch werden und eine Übereinstimmung erschweren.

Aus diesem Grund wird eine unscharfe Menge von Tags konzeptionell erlaubt, aber entsprechend in der Ausgabe vermerkt. Bei der Generierung der Dependenztripel aus den Konstruktikon-Beispielsätzen wird hierbei immer wieder zuerst ein Satz alternierend ausgelassen, dann zwei Sätze und so weiter, bis zum ersten Mal eine Übereinstimmung erfolgt. Der Gedanke ist, dass spezielle Sonderfälle oder Sätze mit Annotationsfehler ausgeschlossen werden und somit keinen Einfluss auf die Schnittbildung für die gemeinsame Menge der Dependenztripel mehr haben.

Für die Analyse der HDT-UD-Sätze wird hingegen zunächst überprüft, ob ein Satz alle Tripel beinhaltet. Ist dies nicht der Fall, so wird nach und nach je ein anderes (bzw. danach zwei, drei etc.) Tripel bei der Überprüfung weggelassen. Sobald dann die Annotierung des Satzes übereinstimmt, wird er in der Ausgabe als potenzielles Beispiel für die aktuell betrachtete Konstruktion festgehalten.

Abbildung 3.1 zeigt, dass die Anzahl auf diese Weise gefundener Beispielsätze pro Konstruktion zu sehr unterschiedlich vielen Ergebnissen führt. Während die Tripel für bspw. Konstruktion 374 („Negation_Schlussfolgerung:nur_weil_X_NEG_Y“) weit über 100.000 mal im HDT-UD-Korpus eine zumindest teilweise Übereinstimmung geliefert haben, sind es etwa für die Konstruktion 571 („Konjunktiv_Präteritum:V.conj.-II“) gerade einmal zwei. Auffällig ist außerdem, dass die Zahl der gefundenen Konstruktionen für das HDT-UD-Korpus im Verhältnis zur Gesamtzahl durch das Vorfiltern stark dezimiert wird, sodass für die meisten Konstruktionen kein einziger potenzieller Beispielsatz gefunden wird.

Ein positiver Nebeneffekt dieser unscharfen Korpusanalyse ist, dass mit der Angabe der Unschärfe gleichzeitig mögliche Werte für eine Rangordnung/Plausibilität der Beispiele erfolgen kann. Indem die beiden Unschärfewerte (während der Tripelbildung und während der eigentlichen Analyse) miteinander multipliziert werden, ergibt sich eine Metrik, welche angibt, wie sicher ein gefundener Beispielsatz zu einer gegebenen Konstruktion gehört. Der Zugehörigkeitswert m der Unschärfe kann demnach wie folgt festgelegt werden:

$$p_{ggS(k)} = \frac{ggS(k)}{\left| \bigcup_{b \in Bsp(k)} Dep(b) \right|} = \frac{\left| \bigcup_{b \in Bsp(k)} Dep(b) \right| - n_{excl}}{\left| \bigcup_{b \in Bsp(k)} Dep(b) \right|} \quad (3.1)$$

$$p_{Dep(s) \cap ggS(k)} = \frac{|Dep(s) \cap ggS(k)|}{|Dep(s)|} \quad (3.2)$$

$$m = p_{ggS(k)} \cdot p_{Dep(s) \cap ggS(k)}, \quad (3.3)$$

wobei $ggS(k)$ stellvertretend für den „größten gemeinsamen Schnitt“ einer Konstruktion k steht. Er beschreibt die Mächtigkeit der größtmöglichen Untermenge an UD-Tripeln, welche nach dem sukzessiven Ausschluss von Konstruktikon-Beispielsätzen gebildet werden kann.² $Bsp(k)$ bildet die Menge aller Konstruktikon-Beispielsätze einer Konstruktion k ab; $Dep(s)$ beschreibt die Menge der UD-Tripel eines Satzes s ; n_{excl} gibt die Anzahl der ausgeschlossenen Beispiele an. Der Wert m ist im Allgemeinen im Intervall $[0, 1]$ enthalten, kann aber durch inkonsistente Tokenisierung oder andere Annotationsfehler ggf. kleiner als 0 werden. In diesem Fall wird der eigentliche Wert als 0 angenommen.

Hierbei zeigt Abbildung 3.2 ein deutlich anderes Bild im Vergleich zu Abbildung 3.1. Zuvor nur selten gefundene Konstruktionen haben in einer Rangfolge nach m -Wert eine teilweise sehr hohe Bewertung. Andere besitzen nun eine überraschend niedrige Position: Etwa die vorher häufigste

²Die berechneten Werte für alle Konstruktionen finden sich in Tabelle A.1 im Anhang.

Konstruktion 374, welche laut Konstruktikon kein definiertes KE-lex besitzt, sodass hier mehr potenzielle Beispielsätze übernommen werden, obwohl diese nur wenig mit den durchschnittlichen Dependenzrelationen der Konstruktikon-Beispielsätze gemein haben. Hingegen lässt sich bei einer qualitativen Überprüfung der konkreten Ergebnisse feststellen, dass Konstruktionsbeispiele mit durchschnittlich höherem m -Wert auch häufiger sinnvolle Beispiele zu ebendieser Konstruktion darstellen. Die m -Metrik liefert also zumindest einen Anhaltspunkt dafür, ob es sich bei einem gefundenen Satz um einen tatsächlichen Repräsentanten einer Konstruktion handelt.

3.2.2 Erkennung von Konstruktionen mit vortrainierten Sprachmodellen

Vortrainierte Sprachmodelle (*Pre-trained (Large) Language Models; PLMs*) stellen insbesondere seit der öffentlichkeitswirksamen Freigabe von OpenAIs GPT-3.5 Turbo, auch bekannt als ChatGPT, eine erstaunliche Hilfe bei unterschiedlichsten Aufgaben dar, insbesondere (jedoch nicht ausschließlich), wenn sie sich auf Aufgaben zu natürlicher Sprache beziehen. Es liegt daher nahe, auch für die Untersuchung von Konstruktionsgrammatiken Versuche mit bereits bestehenden Sprachmodellen zu unternehmen.

Da es sich bei aktuellen Sprachmodellen häufig um generative Modelle handelt, ist eine Erkennung von bestimmten Konstruktionen in einem gegebenen Text nicht unmittelbar realisierbar. In der Praxis müssen bei einer manuellen Annotation passende Beispiele aus großen Textmengen herausgefiltert werden. Da allerdings Sprachmodelle nur eine begrenzte Anzahl an Tokens gleichzeitig und in einem gemeinsamen Kontext verarbeiten können, muss ein entsprechender Versuch anderweitig gestaltet werden.

Bevor elaborierte Methoden getestet werden, sollen zunächst die Fähigkeiten bestehender Modelle evaluiert werden. Hierzu werden zweierlei Versuche durchgeführt. In Anlehnung an WEISSWEILER et al. (2022) wird zunächst die Fähigkeit des jeweiligen Sprachmodells untersucht Positiv- von Negativ-Beispielen bzgl. einer gegebenen Konstruktion zu erkennen. Im Gegensatz zu WEISSWEILER et al. wird jedoch das Verfahren abgewandelt. Negativ-Beispiele werden nicht künstlich bzw. durch Korpussuche als im Sinne von Minimalpaaren erstellt. Stattdessen wird jeweils eine bestimmte Anzahl an Beispielen aus dem Konstruktikon zu der gegebenen Konstruktion zufällig ausgewählt und dann durch eine variierende Anzahl von Negativ-Beispielen ergänzt (insgesamt werden so bis zu zehn Sätze verglichen). Diese Gegenbeispiele sind zufällige Sätze aus dem Konstruktikon, die aus beliebigen anderen Konstruktionen stammen. Somit ist sichergestellt, dass die Sätze plausibel in einem Text vorkommen können, aber gleichzeitig eindeutig nicht durch Zufall der tatsächlich zu untersuchenden Konstruktion entsprechen könnten. Auf diese Weise soll eine Situation modelliert werden, in der aus einem umfangreichen Dokument derjenige Satz herausgefunden werden soll, der einer zu untersuchenden Konstruktion entspricht. Die Anzahl dieser Negativbeispiele wird sukzessive erhöht, um beobachten zu können, wie sich die Leistung des Modells verändert, wenn in komplexen Dokumenten und Korpora eine gesuchte Konstruktion verortet werden soll. Es ist zu vermuten, dass die Leistung des Modells abnimmt, je mehr Sätze unterschieden und analysiert werden müssen.

Erkennung von Konstruktionen mit Llama 2

Zunächst wird das vergleichsweise aktuelle Sprachmodell Llama 2 untersucht. Hierfür soll das Modell mittels geschicktem Prompting dazu gebracht werden, ausschließlich den Index des korrekten Beispiels oder der korrekten Beispiele zu einer Konstruktion aus einer Liste von gegebenen Sätzen auszuwählen. Die grundlegende Idee ist hierbei in Abbildung 3.3 illustriert: Dem Modell wird in einem Prompt zunächst die Definition der jeweiligen Konstruktion übergeben. Die Sätze werden vor der Eingabe in das Modell randomisiert und anschließend numerisch im

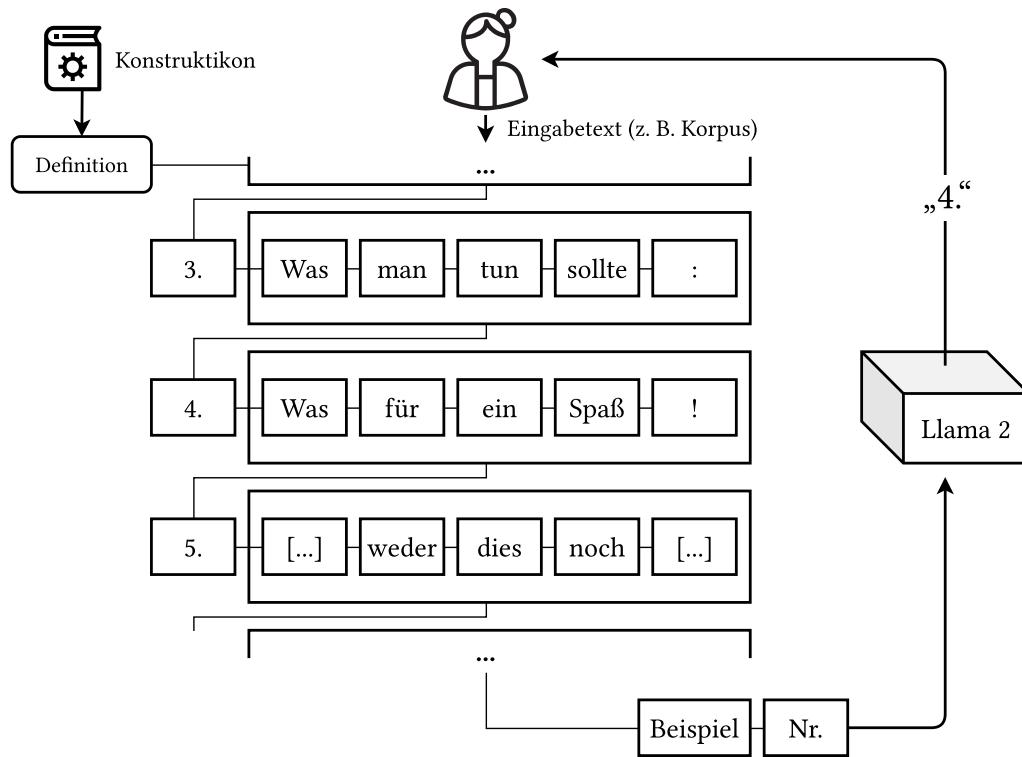


Abb. 3.3: Konzept der Konstruktionserkennung mit Llama 2

Prompt aufgelistet. Ergänzt wird ein Hinweis, dass das Modell ausschließlich die zu den gesuchten Sätzen gehörigen korrekten Indizes zurückgeben soll. Das Experiment wird für verschiedene Anzahlen an korrekten und inkorrekten Beispielen wiederholt, sodass mögliche Tendenzen oder Zufallsergebnisse besser erkannt werden können.

Da in der Realität bei einer Korpusanalyse häufig Gruppen von Sätzen betrachtet werden, in denen kein Beispiel für die gesuchte Konstruktion vorhanden ist, muss auch überprüft werden, wie sich Llama 2 in dieser Situation verhält.

Abbildung 3.4 zeigt die Ergebnisse der Teilversuche, aufgeschlüsselt nach Accuracy- und F1-Werten. Im Allgemeinen ist zu beobachten, dass es unter Llama 2 nur unzureichend möglich zu sein scheint, Konstruktionen anhand ihrer Definition aus Texten zu extrahieren. Zwar erreicht zumindest die Auswahl von wenigen korrekten Beispielen aus mehreren inkorrekten Sätzen eine Accuracy von bis zu 68 Prozent, allerdings bewegen sich die Ergebnisse für die meisten Experimente bei deutlich geringeren Werten. Zudem deutet der geringe F1-Score darauf hin, dass verhältnismäßig wenige bis keine Ergebnisse überhaupt zurückgeliefert wurden. Dies ist insbesondere für die Fälle erkennbar, in denen wenige korrekte Beispiele vorhanden sind.

Es bleibt zu prüfen, ob das Hinzufügen eines Beispiels im Sinne eines Few-Shot-Learnings das Ergebnis verbessern kann. Die Experimente mit Llama 2 selbst in einer quantisierten Version und mit weniger Parametern bedingen jedoch bereits einen hohen Ressourcenaufwand an Rechenleistung und wurden im Rahmen dieser Arbeit daher nicht tiefgehender durchgeführt.

Erkennung von Konstruktionen mit BERT

Um BERT für den Fall der Konstruktionserkennung anwenden zu können, kann es in einer Variante genutzt werden, in der jeweils die Plausibilität zwei aufeinanderfolgender Sätze durch das Modell bewertet wird (NSP). Hierbei werden, ähnlich wie zuvor unter Llama 2, in einem Prompt

3.2 Konzeption der Untersuchung hinsichtlich Konstruktionserkennung

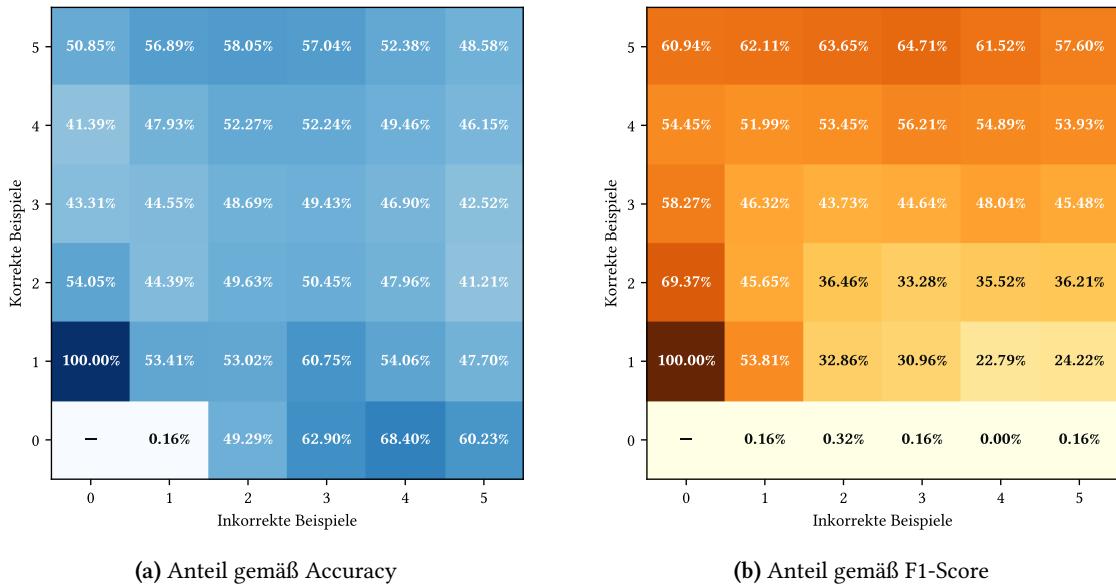


Abb. 3.4: Anteil korrekt erkannter Beispiele durch Llama 2 in Abhängigkeit zur Anzahl gegebener korrekter (Ordinate) und inkorrechter Beispiele (Abszisse)

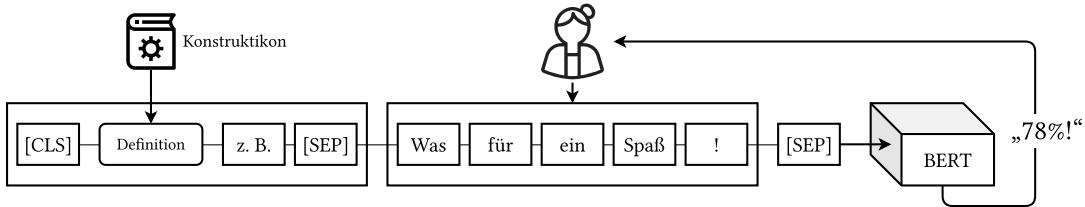


Abb. 3.5: Konzept der Konstruktionserkennung mit BERT und NSP

die Definition übergeben, ergänzt durch einen kurzen Hinweis, dass auf die Definition ein Beispiel folgen wird.

NSP findet in gegenwärtigen Untersuchungen nur noch wenig Beachtung und spielt nur noch selten eine tragende Rolle. Auch in aktuellen Sprachmodellen wird NSP nicht weiter verfolgt, weil sich diese im Rahmen des Pre-Trainings für BERT-ähnliche Modelle als nicht zuträglich erwiesen hat. (vgl. u. a. YINHAN LIU et al., 2019; SANH et al., 2020) Dennoch eignet sich das Konzept für das im Speziellen vorliegende Problem, da hiermit ein Versuchsaufbau gestaltet werden kann, bei dem aus einer Auswahl verschiedener Beispiele zu einer gegebenen Definition das passendste durch BERT ausgewählt werden muss. Eine mögliche Umsetzung ist in Abbildung 3.5 dargestellt. Hierbei wird BERT eine Definition mit dem Zusatz „Zum Beispiel:“ übergeben. Als mögliche nächste Sätze werden dem Modell nun nacheinander die verschiedenen Beispiele vorgeschlagen. BERT bewertet mithilfe der NSP in Form von Logit-Werten die Plausibilität dafür, dass die Definition und das potenzielle Beispiel aufeinander folgen. Erreicht ein Beispiel einen höheren Wert für die Bewertung „nicht zufällig“ als für die Bewertung „zufällig“, so wird dieser Satz als korrekt klassifiziert.

Abbildung 3.6 zeigt, dass dieser Ansatz durchaus vielversprechend erscheint. Für eine höhere Anzahl von Positiv- im Vergleich zu Negativ-Beispielen zeigt das Modell größere Schwierigkeiten. Allerdings führen die Fälle, in denen wenig korrekte und viele inkorrekte Beispiele eingegeben werden – und welche somit den Realbedingungen bei bspw. Korpusanalysen eher entsprechen – zu deutlich besseren Bewertungen für die Accuracy. Auch die entsprechenden F1-Scores sind mindestens annehmbar und befinden sich stets deutlich über 50 Prozent.

3 Erkennung von Konstruktionen in gegebenen Texten

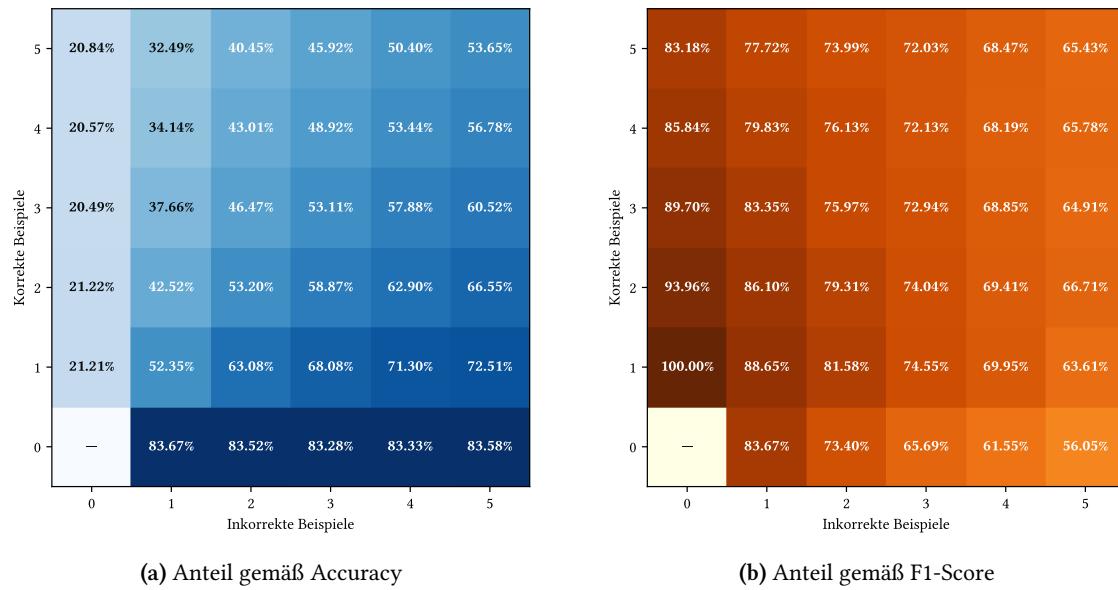


Abb. 3.6: Anteil korrekt erkannter Beispiele durch BERT in Abhängigkeit zur Anzahl gegebener korrekter (Ordinate) und inkorrekt erkannter Beispiele (Abszisse)

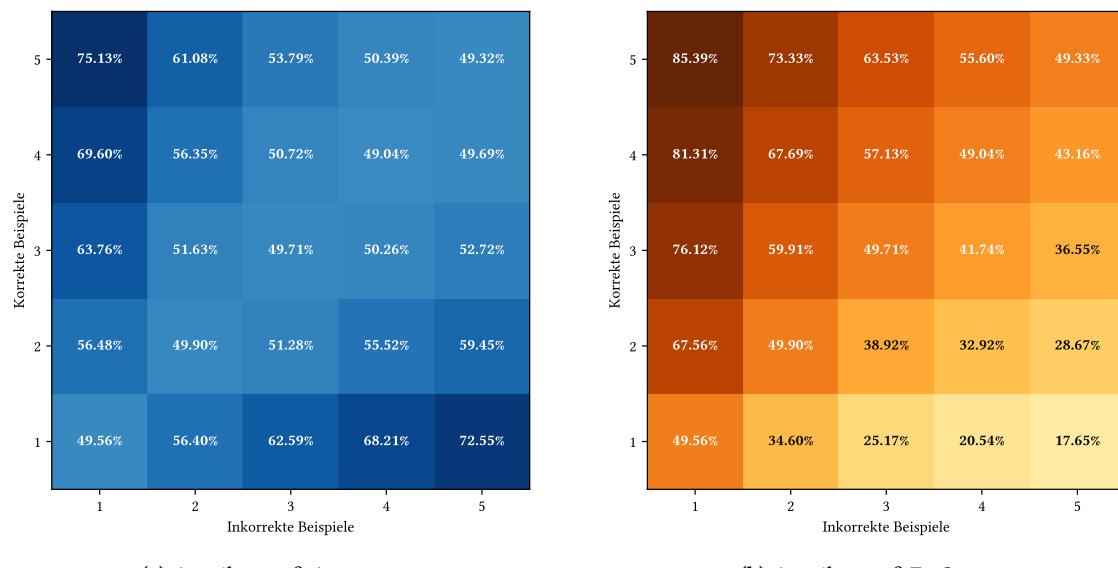


Abb. 3.7: Anteil korrekt erkannter Beispiele durch BERT in Abhängigkeit zur Anzahl gegebener korrekter (Ordinate) und inkorrekt erkannter Beispiele (Abszisse); bei vorgegebener Anzahl von Beispielen, die durch BERT als korrekt zu bewerten sind

Die Ergebnisse sind im Vergleich zu Llama 2 auf den ersten Blick erstaunlich. Obwohl für Llama 2 als das aktuellere Modell die besseren Ergebnisse zu erwarten wären, ist die Genauigkeit, mit der BERT das jeweilige Beispiel zu einer gegebenen Konstruktion korrekt vorhersagt, im Vergleich deutlich höher. Hierbei ist jedoch zu beachten, dass sich auch der Versuchsaufbau unterscheidet. Während Llama 2 aus allen gleichzeitig gegebenen Beispielen das korrekte ausgeben muss, produziert BERT nacheinander Logit-Werte zu jedem möglichen Beispielsatz, unabhängig von der Bewertung der anderen Beispielsätze. Somit können auch aus einer Menge von insgesamt eher unwahrscheinlichen Beispielsätzen einfacher die zumindest wahrscheinlicheren herausgefiltert werden.

In einem alternativen Ansatz wurde weiterhin untersucht, ob sich die Ergebnisse noch verbessern lassen, indem BERT weitere Informationen mit an die Hand gegeben werden. In diesem Fall wurde das Experiment so angepasst, dass die durch BERT zurückgelieferten Logit-Bewertungen für „nicht zufällig“ in einer Liste sortiert werden. Danach wurde die reale Anzahl an korrekten Beispielen vom Kopf der Liste ausgewählt. Das Modell erhält also nun indirekt die Information darüber, wie viele Beispiele tatsächlich korrekt sind (im Experiment zuvor musste BERT wie zuvor schon Llama 2 eigenständig schließen, wie viele Sätze der Konstruktion entsprechen).

Wie in Abbildung 3.7 bereits angedeutet wird, wurde in diesem Experiment auf eine Bewertung der Fälle mit keinem korrekten bzw. inkorrektem Beispiel verzichtet. Da die Anzahl der Beispiele *a posteriori* bei der Auswahl angewendet wird, würde sich für diese Beispiele stets eine Accuracy von 100 Prozent ergeben. Alle anderen Kombinationen von Anzahlen werden weiterhin untersucht und zeigen eine deutliche Verbesserung insbesondere für die Fälle, in denen sich die Anzahl der korrekten von der Anzahl der inkorrekten Beispiele stark unterscheidet. Sind ungefähr gleich viele korrekte und inkorrekte Repräsentanten der Konstruktion gegeben, so liegt die Accuracy lediglich im Bereich um 50 Prozent. Für den F1-Score ist für diese Variante des Experiments zu erkennen, dass dieser Wert deutlich abnimmt, je mehr inkorrekte und je weniger korrekte Beispiele in den übergebenen Daten vorhanden sind. Dieses Ergebnis zeigt, dass es für realitätsnahe Bedingungen nur unter Einschränkungen von Vorteil sein kann, BERT die Information über die tatsächliche Anzahl von korrekten Sätzen zu übergeben – wobei dieser Fall nur in selten vorkommen dürfte.

3.2.3 Ansätze zur Erweiterung der vortrainierten Sprachmodelle durch Pseudowort-Embeddings

In Kapitel 2.4 wurde das Konzept der Pseudowörter theoretisch erläutert. Im Folgenden wird überprüft, ob Sprachmodelle durch eine diesbezügliche Erweiterung eine bessere Leistung bei der Erkennung von Konstruktionen in einem Korpus erbringen. Hierzu werden jeweils die sogenannten KE-lex in einem gegebenen Satz durch die vorher trainierten Pseudowörter ersetzt. Danach findet eine Analyse der Positiv- und Negativ-Beispiele statt, um die Qualität der Näherung durch Pseudowörter zu testen.

Implementierung von Pseudowörtern für Konstruktionsgrammatiken

In Kapitel 4.2 wird festgestellt werden, dass PLMs ohne Spezialisierung auf Konstruktionsgrammatiken nur unzureichend Konstruktionen erfassen und entsprechende Beispiele produzieren können. Folglich ist es plausibel, ein LLM um Informationen zu selbigen zu erweitern.

In Kapitel 2.4 wurde bereits eine Strategie vorgestellt, mittels derer durch zusätzliche Word-Embeddings spezifische semantische Nuancen eines Lemmas disambiguierter werden können (KARIDI et al., 2021). Als prototypisches Beispiel hierfür ist die Präposition *in* denkbar, welche sowohl im Englischen als auch im Deutschen jeweils eine temporäre („Wir sehen uns *in* Kürze.“) oder eine

lokative Bedeutung (‘Wir sehen uns *in* Leipzig.’) erhalten kann. Hierfür wird BERT (DEVLIN et al., 2019) als grundlegendes LLM verwendet. Durch einige Beispielsätze, in denen die jeweilige Semantik des Wortes dargestellt ist, werden neue, sogenannte Pseudoword-Embeddings trainiert. Das disambiguierende Wort (‘Kürze’ bzw. ‘Leipzig’), auch „cue token“ genannt, wird hierbei mit einer Maske versehen, wobei das Modell daraufhin versucht, das maskierte Element vorherzusagen. Das entstehende Pseudoword-Embedding repräsentiert danach im Idealfall nur noch eine bestimmte Variante von *in*, welches somit das „focus token“ darstellt.

Es ist nun möglich, diese Idee auch im Kontext von Konstruktionsgrammatiken weiterzuverfolgen. Um dies zu erreichen, können einige Eigenschaften der Konstruktionsdefinitionen im Konstruktikon ausgenutzt werden. Hierbei sind die *lexikalischen Konstruktionselemente* (KE-lex) und die (sonstigen) *Konstruktionselemente* von entscheidender Bedeutung. Ein KE-lex bildet das zentrale Element der jeweiligen Konstruktion, dessen Vorhandensein im Allgemeinen eine notwendige Bedingung zum Zustandekommen der Konstruktion darstellt. Allerdings ist nicht jeder Satz, der ein Wort enthält, das eine KE-lex entspricht, sofort ein Repräsentant einer bestimmten Konstruktion. Dies lässt sich erst durch die Anwesenheit und das Wesen der restlichen KEs bestimmen.

Als Beispiel sei die Konstruktion „Negation:NEG_X_geschweige_denn_Y“ gegeben.³ Sie enthält das KE-lex *geschweige denn*, welches notwendig ist, um die Konstruktion auszulösen: ‚Es wurden Änderungsanträge eingereicht, die *kaum* für einen Fachmann, *geschweige denn* für einen Nichtfachmann verständlich sind.‘ Das KE-lex *geschweige denn* bekräftigt nach einer negierten Aussage diese Negierung.

Es ließe sich auf den ersten Blick annehmen, dass allein die Anwesenheit dieses KE-lex ausreicht, um die Konstruktion nachweisen zu können: Die Anträge seien für eine geübte Person nicht verständlich *und erst recht nicht* für ungeübte. Ein Blick auf einen leicht modifizierten Satz zeigt jedoch ein Gegenbeispiel auf: ‚Es wurden Änderungsanträge eingereicht, die *schon* für einen Fachmann, *geschweige denn* für einen Nichtfachmann schwer verständlich sind.‘ In diesem Fall ist die Satzstruktur eine affirmative: Die Anträge sind für einen Fachmann unverständlich *und erst recht* unverständlich für Unerfahrene. Da es sich also um eine verstärkende und keine weiter negierende Semantik handelt, kann es sich in letzterem Beispiel nicht um die gegebene Konstruktion handeln. Formal lässt sich dies an dem fehlenden *Negator* erkennen, der in der „Negation:NEG_X_geschweige_denn_Y“-Konstruktion als Kern-KE definiert ist. Im ersten Beispiel stellt *kaum* diesen Negator dar. Im zweiten Beispiel nimmt *schon* die äquivalente Stelle im Satz ein, ist aber keinesfalls ein negierendes Element.

Es bleibt festzuhalten, dass ein KE-lex in einem Satz erst durch die restlichen KEs dahingehend disambiguiert wird, ob es im Rahmen einer bestimmten Konstruktion auftritt oder nicht. Hier lässt sich sofort die Parallele zu den Ausführungen durch KARIDI et al. (2021) bzgl. Pseudoword-Embeddings ziehen. Es kann also folgende Annahme getroffen werden: Das „focus token“ kann durch das KE-lex repräsentiert werden. Es ist essenziell für die jeweilige Konstruktion, allerdings muss stets untersucht werden, ob es auch tatsächlich die vermutete Konstruktion bedingt oder nicht. Dies geschieht durch abwechselndes Maskieren der restlichen KEs, welche den „cue tokens“ gleichgesetzt werden. Indem das neue Pseudoword-Token mit diesem MLM trainiert wird, kann das KE-lex auf die jeweilige Konstruktion spezialisiert und damit von anderen Vorkommen außerhalb der Konstruktion disambiguiert werden.

XU et al. (2023) geben jedoch bereits zu bedenken, dass Konstruktionsinformationen nur selten durch MLM-basierte Sprachmodelle erfasst würden und diese große Mengen an Trainingsdaten benötigten, da die Bedeutung einer Konstruktion gemäß HOFFMANN und TROUSDALE (2013, S. 2) nicht von Wörtern, sondern ganzen Sprachmustern abhängig sei. In der Praxis bedeutet dies, dass

³Im Englischen existiert mit der *Let_alone*-Konstruktion interesserweise eine analoge, vielfach in der Literatur diskutierte Variante.

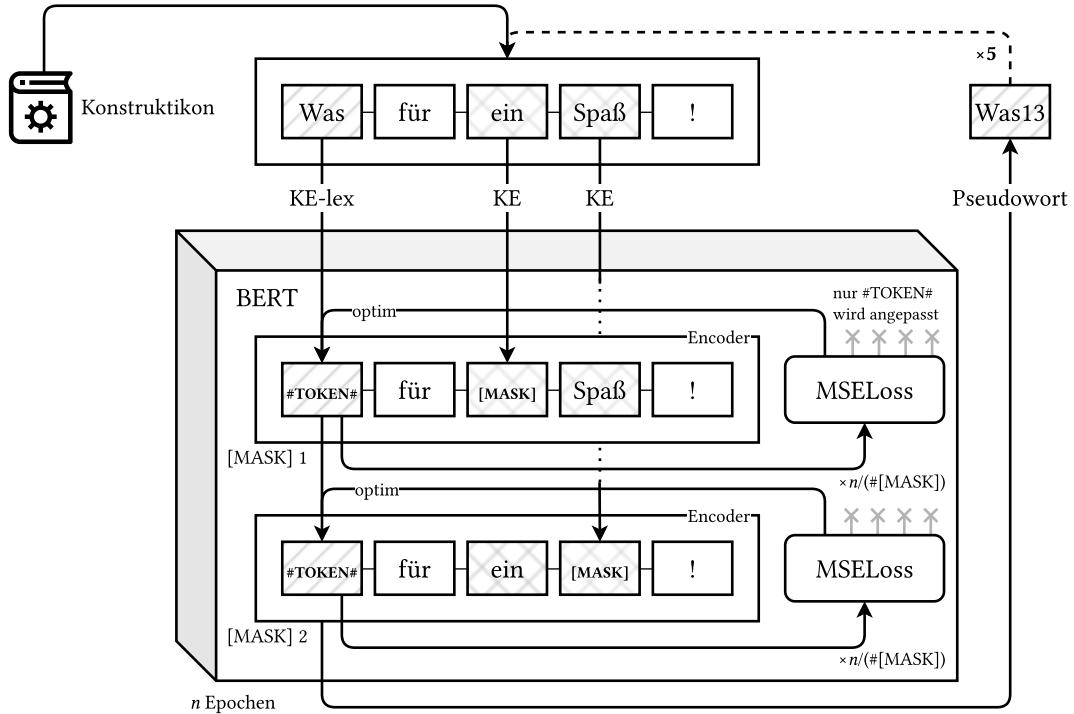


Abb. 3.8: Training eines Pseudowort-Embeddings für BERT auf Basis eines Beispielsatzes des Konstruktions mit zwei KE-Tokens

das durch KARIDI et al. vorgeschlagene Konzept der Pseudoword-Embeddings nicht direkt auf das BERT-Modell angewendet werden kann. Da BERT auf die Vorhersage von einzelnen Token im Rahmen des MLM-Objective vortrainiert ist, muss hier ein Weg gefunden werden, längere Sequenzen – und im Trainingsverlauf auch Sprachmuster – durch das LLM verarbeiten zu lassen.

Erweiterung von BERT mit Pseudowörtern

Pseudowörter im Rahmen von BERT wurden bereits durch KARIDI et al. (2021) und in dieser Arbeit in Kapitel 2.4 konzeptionell definiert. Im Folgenden soll zunächst das notwendige Training skizziiert werden und schließlich die Leistung eines durch Pseudowörter ergänztes BERT-Modell getestet werden. Beides wird in Abbildung 3.8 grafisch verdeutlicht.

Die zu disambiguierenden Elemente in einer Konstruktion des deutschen Konstruktions stellen die sogenannten lexikalisch festen Konstruktionselemente (KE-lex) dar. Diese Annahme kann getroffen werden, da diese zwar notwendige Bestandteile einer gegebenen Konstruktion sind (sofern sie definiert sind), aber ihr lexikalisches Vorhandensein das Bestehen einer Konstruktion noch nicht allein bedingt. Hinreichende Bedingung hierfür ist, dass das KE-lex mit sogenannten (Kern-)Konstruktionselementen gemeinsam auftritt.

Für eine Definition von Pseudowörtern bedeutet dies, dass jeweils ein neues Token für ein KE-lex trainiert werden muss, welches im Kontext der Konstruktion bzw. der restlichen Konstruktionselemente steht. Aus diesem Grund werden letztere mittels MLM während des Trainings maskiert und vorhergesagt, sodass das KE-lex-Token auf deren Anwesenheit nivelliert wird.

Die fertig trainierten neuen Embeddings können nun in ein neues BERT-Modell geladen werden. In einem weiteren Schritt werden die KE-lex-Tokens zunächst im korrekten Satz durch die Pseudowort-Embeddings repräsentiert. Auf diese Weise kann verglichen werden, ob die Leistung von BERT bei der Vorhersage des korrekten Beispielsatzes im Vergleich zur Nutzung der

3 Erkennung von Konstruktionen in gegebenen Texten

ursprünglichen, allgemeinen Embeddings gesteigert werden kann. Für die Unterscheidung zwischen korrekten und inkorrekten Sätzen wird erneut NSP angewendet.

Denkbar ist ein Einsatz der Pseudowort-Embeddings auch, indem in den anderen übergebenen Sätzen alle Vorkommnisse einer vermeintlichen KE-lex-Form durch das Pseudowort ersetzt werden. Hierbei würde bewusst ignoriert werden, dass die ersetzen Formen in den falsch-positiven Beispielen nicht tatsächlich ein KE-lex darstellen. Dadurch soll der Fall modelliert werden, in dem beispielsweise bei einer Annotation *a priori* nicht bekannt ist, welche Sätze eine Konstruktion enthalten. Die Idee ist es, durch die inkorrekten Ersetzungen zu erzielen, dass dem Modell ungewöhnliche Sätze übergeben werden, die potenziell seltener als korrekt klassifiziert werden. Dieser Versuch wurde in der vorliegenden Arbeit jedoch nicht weiter verfolgt, bleibt aber für zukünftige Betrachtungen interessant.

Die Abbildungen 3.9 und 3.10 stellen die Ergebnisse dar, die analog zu den Experimenten aus Kapitel 3.2.2 erzielt wurden. Zu erkennen ist eine deutliche Verbesserung für die Fälle, in denen keine oder wenige korrekte Beispiele zu einer gegebenen Konstruktion in der aktuellen Menge von Sätzen vorhanden sind. Auch die F1-Scores deuten auf eine höhere Rate bei der Anzahl der tatsächlich zurückgegebenen Ergebnisse hin. Die daraus folgende Annahme ist, dass die Erweiterung von BERT mit Pseudowort-Embeddings deutliche Qualitätszugewinne mit sich bringt.

Um sicherzustellen, dass die Unterschiede der Leistung vor und nach der Ergänzung von Pseudowörtern zum BERT-Modell statistisch signifikant sind, wird ein gepaarter *t*-Test durchgeführt (siehe hierzu die Abbildungen 3.11 und 3.12). Dieser kann Aufschluss darüber geben, ob die Differenzen tatsächlich plausibel sind oder nur zufälligen Schwankungen unterliegen. Die Nullhypothese umfasst hierbei die Annahme, dass die Erweiterung durch Pseudowort-Embeddings keine Verbesserung liefert. Im vorliegenden Fall vergleicht der *t*-Test die Ergebnisse für BERT vor und nach der Ersetzung der KE-lex-Tokens durch deren jeweiligen Pseudowörtern bei der Bewertung von potenziellen Beispielen. Zu beobachten ist, dass die Mehrheit der Ergebnisse bei einem Signifikanzniveau von $\alpha = 0,05$ tatsächlich statistisch signifikant ist. Die Nullhypothese kann in Bezug auf BERT demnach im Allgemeinen verworfen werden.

3.2.4 Vergleich von allgemeinen kontextuellen Embeddings mit Pseudowort-Embeddings

In den Kapiteln 3.2.2 und 3.2.3 wurde bereits analysiert, welche Leistungen von verschiedenen Ansätzen mit Llama 2 und BERT zu erwarten sind. Es wurde festgestellt, dass eine Erweiterung von BERT durch Pseudowort-Embeddings zu einer deutlich verbesserten Erfassung von Konstruktionen führt. Um nun die tatsächliche inhaltliche Nähe der spezialisierten Pseudowort-Embeddings zu den in einem realen Kontext stehenden Tokens auszuwerten, wird folgendermaßen vorgegangen: Zunächst wird ein Satz aus einem beliebigen Korpus betrachtet, für den vermutet wird, dass er eine bestimmte Konstruktion enthält. In Kapitel 3.2.1 wurde bereits eine entsprechende Liste an Beispiele aus dem HD-UDT-Korpus herausgelöst.

Zu jedem Satz wird nun mittels BERT ein Tensor von kontextuellen Embeddings erstellt sowie jeweils dasjenige Embedding betrachtet, das dem KE-lex entspricht. Die Idee ist es, jeweils von diesem Satz unabhängige Embeddings als Vergleichsobjekte heranzuziehen, die jeweils einmal durch allgemeine BERT- und einmal durch spezialisierte Pseudowort-Embeddings gebildet werden. Abbildung 3.13 skizziert, wie der Embeddingvektor zum KE-lex im gegebenen Satz mit diesen anderen beiden Vektoren verglichen werden kann. Hierbei müssen, für einen aus mathematischer Sicht sinnvollen Vergleich, alle Embeddingvektoren in einem gemeinsamen Vektorraum \mathcal{C} definiert sein. Dies wird erreicht, indem die Vergleichsvektoren analog zu den aktuell betrachteten Embeddings im Beispielsatz durch kontextuelle Embeddings repräsentiert werden.

3.2 Konzeption der Untersuchung hinsichtlich Konstruktionserkennung

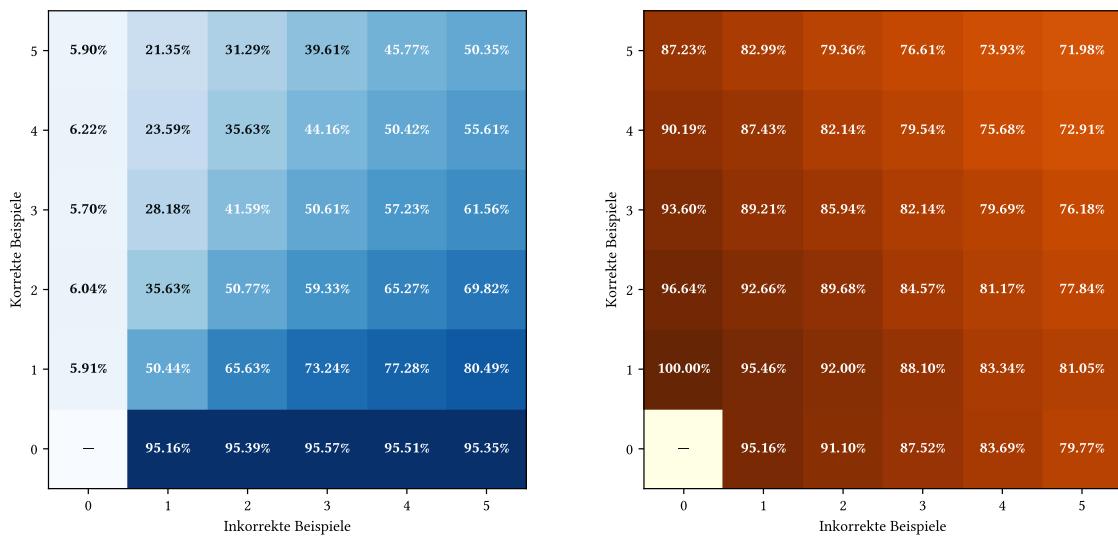


Abb. 3.9: Anteil korrekt erkannter Beispiele durch BERT in Abhängigkeit zur Anzahl gegebener korrekter (Ordinate) und inkorrechter Beispiele (Abszisse); KE-lex wurden durch entsprechende Pseudowort-Embeddings ersetzt

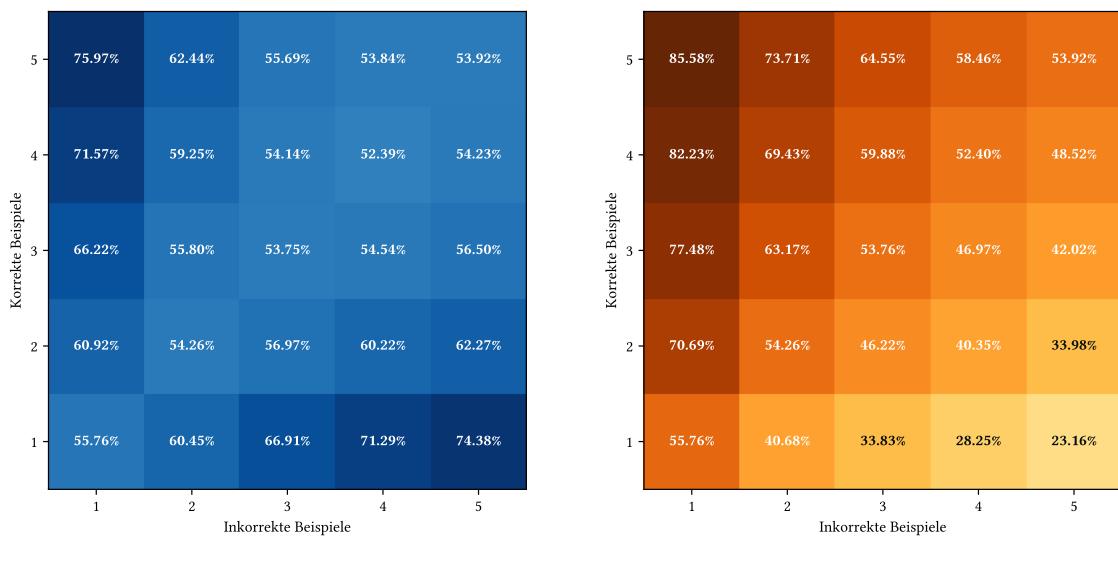


Abb. 3.10: Anteil korrekt erkannter Beispiele durch BERT in Abhängigkeit zur Anzahl gegebener korrekter (Ordinate) und inkorrechter Beispiele (Abszisse); KE-lex wurden durch entsprechende Pseudowort-Embeddings ersetzt; bei vorgegebener Anzahl von Beispielen, die durch BERT als korrekt zu bewerten sind

3 Erkennung von Konstruktionen in gegebenen Texten

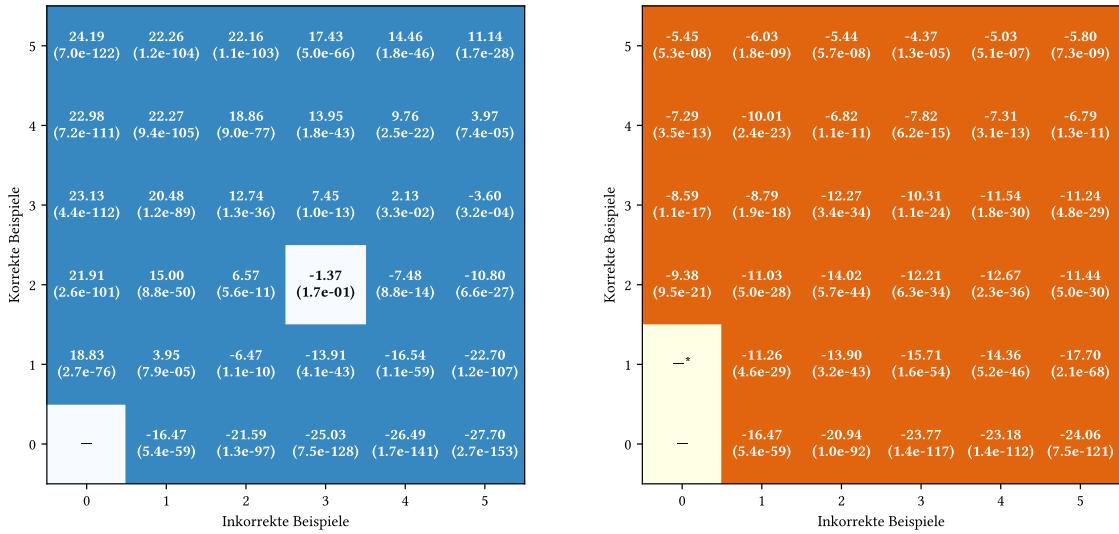


Abb. 3.11: Ergebnisse des Signifikanztests (gepaarter t -Test) beim Vergleich von BERT vor und nach dem Ersetzen der KE-lex durch Pseudowort-Embeddings
(* Anm.: die Vergleichswerte entsprechen hier jeweils ca. 1 und der Standardfehlerwert nähert sich 0 an; t und p werden aufgrund der Division durch 0 ungültig)

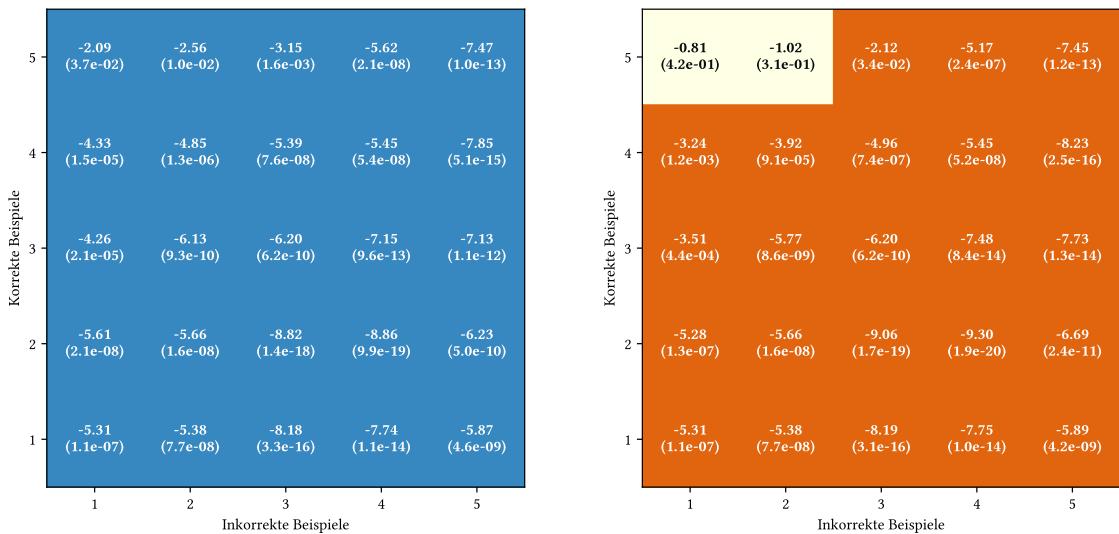


Abb. 3.12: Ergebnisse des Signifikanztests (gepaarter t -Test) beim Vergleich von BERT vor und nach dem Ersetzen der KE-lex durch Pseudowort-Embeddings; bei vorgegebener Anzahl von Beispielen, die durch BERT als korrekt zu bewerten sind

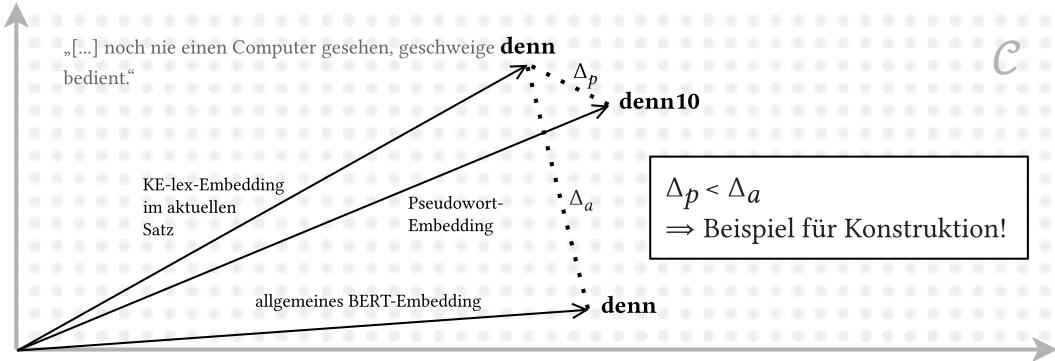


Abb. 3.13: Abschätzung der Zugehörigkeit gefilterter Beispiele zu Konstruktionen durch Ähnlichkeits- und Abstandsmetriken Δ über KE-lex-Embeddings (Embedding im untersuchten Satz (oben), Pseudowort-Embedding (mittig), allgemeines BERT-Embedding (unten)); alle Embeddings wurden durch BERT kontextualisiert und befinden sich somit in einem gemeinsamen Vektorraum \mathcal{C}

Konkret werden im Folgenden zwei mögliche Eingabevarianten für BERT ausprobiert: Einerseits können einfache ‚Ein-Wort-Sätze‘ genutzt werden, im Rahmen derer dem Modell jeweils nur das Pseudowort-Token bzw. das allgemeine KE-lex-Token übergeben wird. Hierbei fehlt der intuitiv notwendige Kontext allerdings, auch wenn es sich letztlich trotzdem um kontextuelle Embeddings handelt. Andererseits können erneut die vordefinierten Beispielsätze des Konstruktions verwendet und somit ein durchschnittlicher Vektor über die KE-lex-Tokens der verschiedenen Beispiele gebildet werden. Die KE-lex-Tokens werden für den Pseudowort-Embedding-Vektor durch die entsprechenden Pseudowort-Tokens ausgetauscht.

Somit existieren zu den einzelnen KE-lex-Tokens der vorgefilterten Beispielsätze jeweils zwei zusätzliche Vektoren, die sich im gleichen Vektorraum befinden wie die kontextuellen Embeddings des Satzes. Entsprechend können die Vektoren direkt mit den kontextuellen Embeddings des Beispielsatzes verglichen werden. Ziel ist es zunächst, das KE-lex des Beispielsatzes zu betrachten und dieses im Kontext des Satzes mit dem originalen BERT-Embedding der lexikalischen Form des KE-lex sowie mit dem spezialisierten Pseudowort-Embedding des KE-lex zu vergleichen, um mögliche Tendenzen hinsichtlich Distanz- und Ähnlichkeitsmetriken zu erkennen. Diese Metriken umfassen im Rahmen dieser Arbeit die Kosinus-Ähnlichkeit sowie Manhattan- und euklidische Distanz. Die Erwartung ist, dass für die KE-lex-Tokens die Pseudowort-Embeddings für tatsächliche Beispiele der jeweiligen Konstruktion näher am zugehörigen Embedding des Satzes zu verorten sein werden als die allgemeinen BERT-Embeddings.

An dieser Stelle seien einige Sonderfälle bei der Analyse und der Umgang mit ihnen erwähnt. Erstens kann ein Pseudowort-Embedding unter Umständen eine Entsprechung aus mehreren originalen Embeddings besitzen. Im Kapitel 3.3.2 wird etwa exemplarisch der Fall des KE-lex „Was“ noch einmal genauer diskutiert, welches in der Regel durch drei Tokens repräsentiert, als Pseudowort jedoch nur als einzelnes Token trainiert wird. Für den Fall, dass tatsächlich mehrere Embeddings durch ein gemeinsames Pseudowort-Embedding repräsentiert werden, gilt, dass die Anzahl der originalen BERT-Embeddings und die Anzahl der Embeddings für das Token im Beispielsatz übereinstimmen. Daher werden hier die Abstände und Distanzen der zueinander gehörigen Vektoren gebildet und gemittelt. Außerdem werden die Metriken des kontextuellen Embeddings des Pseudoworts im Verhältnis mit jedem der Embeddings des Tokens im Beispielsatz berechnet und ebenfalls das Mittel gebildet.

Zweitens kann es vorkommen, dass ein Beispielsatz im Korpus die zugelassene Anzahl an Wörtern bzw. Tokens überschreitet, die für das BERT-Modell 512 beträgt. Daraus folgende mögliche

3 Erkennung von Konstruktionen in gegebenen Texten

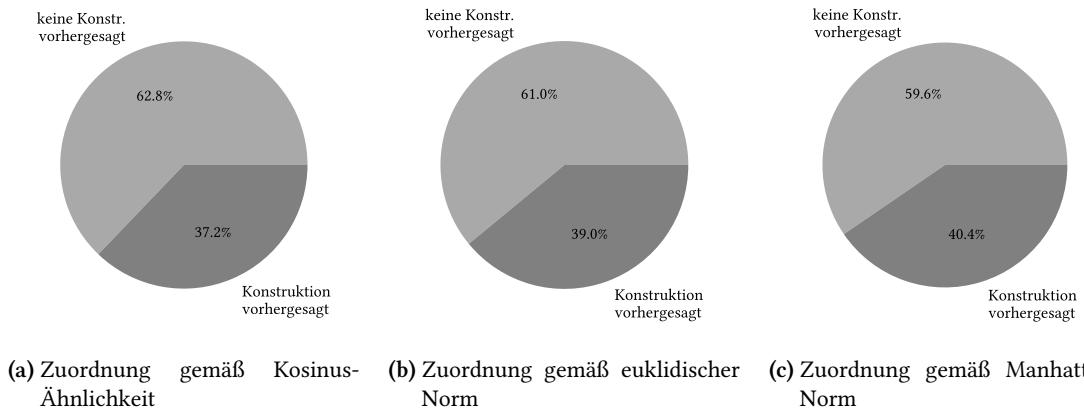


Abb. 3.14: Anteile der Sätze des HDT-UD-Korpus, für die das jeweilige Pseudowort-Embedding näher am Satztoken-Embedding ist; Vergleich mit durchschnittlichem Embedding bei Instanziierung als Ein-Wort-Satz

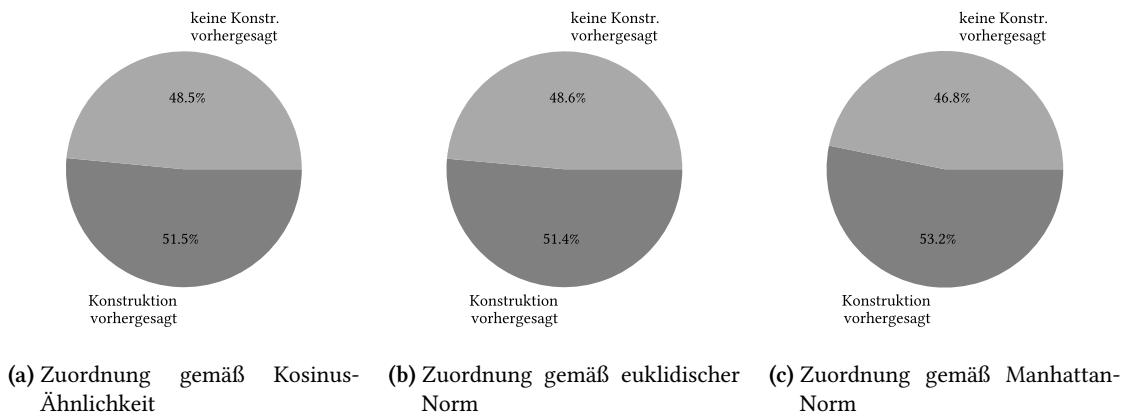


Abb. 3.15: Anteile der Sätze des HDT-UD-Korpus, für die das jeweilige Pseudowort-Embedding näher am Satztoken-Embedding ist; Vergleich mit durchschnittlichem Embedding über alle Konstruktionskon-Beispiele

Unstimmigkeiten der Anzahl der Vektoren bei der Berechnung der Distanz- und Ähnlichkeitsmetriken könnten grundsätzlich umgangen werden, indem ein sogenannter Sliding-Window-Ansatz implementiert wird. Hierbei wird der aktuelle Satz in mehrere Sequenzen der Länge 512 unterteilt, wobei diese stets um ein Token verschoben beginnen. Somit ist sichergestellt, dass mögliche Kontexte für das aktuell zu betrachtende Token bestmöglich modelliert werden. Danach werden zu jeder Repräsentation des Embeddings über die verschiedenen Sequenzen die Metriken berechnet und schließlich das arithmetische Mittel gebildet. Da dies in der Praxis jedoch zu einem hohen zusätzlichen Ressourcenaufwand mit der Gefahr bspw. eines Überlaufs des GPU-Speichers führt und zudem nur wenige Sätze tatsächlich diese Überlänge erreichen, wird hierauf verzichtet und die entsprechenden Sätze übersprungen.

Die Abbildungen 3.14 respektive 3.15 zeigen die Ergebnisse in Bezug auf die Variante mit Ein-Wort-Sätzen und einer Mittelwertbildung über die Konstruktionskon-Beispiele. Hierbei wird eine Konstruktion als vorhergesagt angenommen, wenn die jeweilige Metrik eine stärkere Ähnlichkeit bzw. geringere Distanz des Pseudowort-Embeddings zum Embedding des gegebenen Satzes angibt. Es ist zu beobachten, dass im ersten Fall tendenziell seltener eine Ähnlichkeit der Pseudowort-Embeddings festgestellt wird, während in der zweiten Variante mehr als die Hälfte aller vorgefilterten Beispiele als Repräsentanten der jeweiligen Konstruktion bewertet werden.

3.2 Konzeption der Untersuchung hinsichtlich Konstruktionserkennung

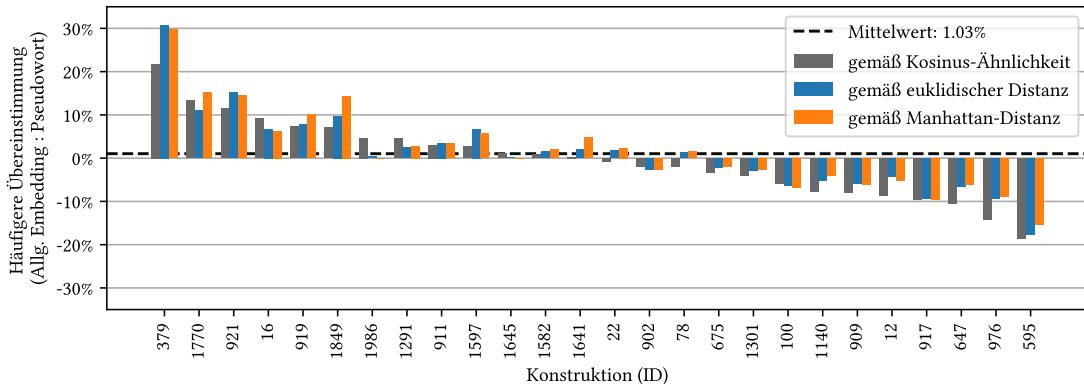


Abb. 3.16: Verhältnis der Häufigkeit einer Übereinstimmung pro Konstruktion (Indizes gemäß Konstruktion); positive Werte zeigen eine häufigere Übereinstimmung mit dem Pseudowort-Embedding, negative mit dem regulären Embedding des KE-lex (jeweils als Ein-Wort-Satz instanziert) an

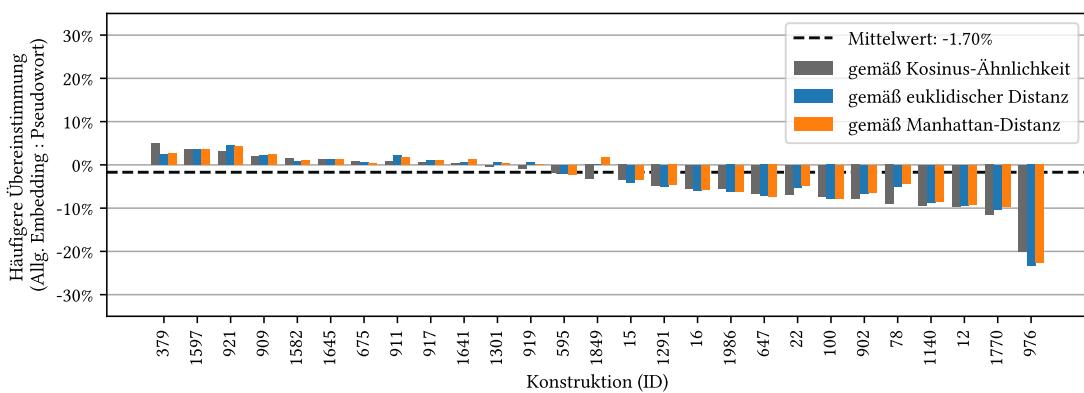


Abb. 3.17: Verhältnis der Häufigkeit einer Übereinstimmung pro Konstruktion (Indizes gemäß Konstruktion); positive Werte zeigen eine häufigere Übereinstimmung mit dem Pseudowort-Embedding, negative mit dem regulären Embedding des KE-lex (jeweils gemittelt über die Konstruktion-Beispiele) an

Mittelwerte	menschl. Eval. (micro)	q (micro)	menschl. Eval. (macro)	q (macro)
Vergleich mit Embedding aus Ein-Wort-Satz	21,77 %	18,22 %	23,27 %	17,70 %
Vergleich mit durchschnittl. Embedding aus Konstruktikon-Beispielen	19,30 %	28,77 %	22,71 %	28,07 %

Tab. 3.1: Evaluationsergebnisse über die gefundenen Beispiele im HDT-UD-Korpus

Die Abbildungen 3.16 und 3.17 schlüsseln dies weiter auf, indem die Metriken über die vorgefilterten Kandidaten pro Konstruktion gemittelt werden. Hier ist auf den ersten Blick erkennbar, dass sich die Vorhersagen zwischen den verschiedenen Konstruktionen im Durchschnitt stark unterscheiden. Insbesondere in der zweiten Variante ist zudem auffällig, dass vor allem dann das KE-lex-Embeddings eines Beispiels näher an dem entsprechenden Pseudowort-Embedding liegt, wenn es sich um eine Klammer-Konstruktion handelt (hier die Konstruktionen 379, 1597, 921 und 909).

Abschließend ist zu untersuchen, inwiefern die Bewertungen auf Grundlage der Ähnlichkeits- und Distanzmetriken mit der tatsächlichen Qualität der Beispiele in Hinblick auf die ihnen zugewiesene Konstruktion korrelieren. Hierzu wird in Abhängigkeit zur in Gleichung 3.3 definierten Zugehörigkeit m eines Satzes zu einer Konstruktion ein Qualitätswert q definiert:

$$q = m \cdot \frac{1}{2} (\Delta + 1) \quad (3.4)$$

$$= m \cdot \frac{1}{2} \left(\frac{\Delta_{\cos} + \Delta_2 + \Delta_1}{3} + 1 \right), \quad (3.5)$$

wobei $\Delta \in [-1, 1]$ die gemittelte Differenz der Ähnlichkeit angibt, in welcher das KE-lex-Embedding im untersuchten Satz zum Pseudowort-Embedding bzw. zum allgemeinen BERT-Embedding steht. Diese wird auf den Wertebereich $[0, 1]$ angepasst, sodass eine Multiplikation mit m zu einer erneuten Zugehörigkeitsmetrik $q \in [0, 1]$ führt. Diese Metrik q soll nun eine abschließende Bewertung darüber abgeben, ob es sich bei einem Beispiel tatsächlich um eines für eine bestimmte Konstruktion handelt.

Diese Hypothese wird mithilfe einer menschlichen Bewertung überprüft. Da aus Zeitgründen nicht alle gut 350.000 vorgefilterten Sätze überprüft werden können, werden pro Konstruktion jeweils höchstens die fünf bestbewerteten Kandidatensätze ausgewertet. Diese werden dabei binär mit „korrekt“ bzw. „inkorrekt“ annotiert. Die Auswertung zeigt, dass der Wert q mit dem tatsächlichen Anteil korrekter Beispielsätze zu einer gegebenen Konstruktion weitgehend korreliert, wie Tabelle 3.1 darstellt.⁴ Die menschliche Evaluation bewertet ein Beispiel als zutreffend zu einer Konstruktion mit einer Differenz von zwischen drei und neun Prozentpunkten im Vergleich zur q -Metrik (abhängig davon, wie die Vergleichseembeddings zuvor gewählt wurden). Gewichtet nach Konstruktion beträgt diese Differenz durchschnittlich ca. fünf Prozentpunkte und unterscheidet sich somit im Mittel kaum von der globalen Bewertung.

Das Ergebnis zeigt, dass zumindest ein beachtlicher Teil der letztlich übrig gebliebenen Beispielsätze des HDT-UD-Korpus als tatsächliche Repräsentanten der ihnen zugeordneten Konstruktionen bewertet werden konnten. Zwar gibt es weiterhin eine leichte Differenz zwischen der menschlichen und maschinellen Einschätzung, dafür lässt sich auch bei den als „inkorrekt“ titulierten Sätzen qualitativ erkennen, dass diese häufig eine gewisse Nähe zu der Konstruktions-

⁴Weitere Daten finden sich im Anhang in den Tabellen A.2 sowie A.3.

definition besitzen. Insgesamt lässt sich durchaus davon ausgehen, dass die q -Metrik nach einer größeren Vorfilterung ein gutes Maß dafür angibt, welche der übrigen Beispiele einer Konstruktion zuzuordnen sind.

3.3 Details zu Implementierung und Hardware

3.3.1 Auswahl der konkreten Sprachmodelle

Für die Verarbeitung deutscher Sprache eignen sich die ursprünglichen Modelle Llama 2 und BERT aus verschiedenen Gründen nicht. Zunächst sind beide Sprachmodelle vorwiegend auf das Englische vorgenutzt, wodurch eine Eingabe deutschsprachiger Texte zu ungenauen oder fehlerhaften Ausgaben führt. Im Fall von Llama 2 stellt zudem die große Anzahl von Modellparametern und die damit verbundenen Anforderungen an den GPU-Arbeitsspeicher ein Problem dar.

Für letzteres Sprachmodell wird aus diesem Grund auf das quantisierte Modell *Llama 2 13B German Assistant v4 - GPTQ* (JOBBINS, 2023) zurückgegriffen. Hierbei handelt es sich um eine reduzierte Version des 13B-Modells von Llama 2, das in der nötigen Rechenleistung durch sogenannte Quantisierungsverfahren reduziert wurde. Zudem basiert es auf einer bereits auf das Deutsche spezialisierten Variante des Sprachmodells.

Für ein deutschsprachiges BERT wurde das spezialisierte Modell *German BERT* in der *case-Variante* (DBMDZ, 2023) eingesetzt. Dieses hat den Vorteil, dass insbesondere Zeichensetzungs-token, welche für die Analyse verschiedener Konstruktionen unerlässlich sind, korrekt interpretiert und verarbeitet werden. Die Nutzung anderer verbreiteter BERT-Modelle für das Deutsche zeigen diesbezüglich Probleme bei der Unterscheidung solcher Tokens und zeigen diese beispielsweise als [unused_punctuation2] an. Dieses Verhalten ist nur spärlich dokumentiert und findet sich in der Literatur ebenfalls nur selten wieder; ein Beispiel hierfür liefert dennoch u. a. ANIKINA (2023).

3.3.2 Anmerkungen zur Implementierung des Pseudowort-Embedding-Trainings mit BERT

Das Problem mit Generierung einzelner Tokens mittels BERT besteht darin, dass die zu prognostizierenden Elemente in Konstruktionsbeispielen gemeinhin aus mehreren Tokens zusammengesetzt sind.

Ein erster Lösungsansatz besteht aus diesem Grund darin, die Maskierung zu alternieren. Wenn also ein KE aus mehreren Tokens besteht, wird der Datensatz so erweitert, dass zu jedem Token in einem KE eine Maskierung erstellt wird. Hat ein KE eines Beispielsatzes etwa fünf Tokens, so entstehen fünf Trainingseinheiten, da jedes Token des KEs nacheinander maskiert wird. Ein vorteilhafter Nebeneffekt ist, dass hierbei auch die Anzahl der Trainingsdaten erhöht wird, sodass das neue Token für das jeweilige KE-lex des Satzes mit mehr Eingaben trainiert wird. Außerdem sind BERT-Modelle im Allgemeinen kleiner und schneller zu verarbeiten als spätere, größere Modelle (etwa mBART, vgl. Kapitel 4.3.2). Für das vorliegende Experiment wird eine auf das Deutsche spezialisierte Variante von BERT eingesetzt.

Auch das KE-lex kann unter Umständen aus mehreren Tokens bestehen. Sind bereits im Korpus mehrere Wörter annotiert (vgl. „geschweige denn“), so werden jeweils einzelne Pseudoword-Embeddings dafür erstellt und trainiert. Durch Ungleichmäßigkeiten in der Annotation gibt es jedoch auch Fälle, in denen ein Token mutmaßlich ungewollt aus mehreren Tokens besteht. So existiert etwa ein Token „Was“. Dieses wird durch die meisten Tokenisierungsmethoden in drei

Tokens zerlegt. Solche Tokens werden allerdings weiterhin als ein einzelnes Token betrachtet. Beim Training würde für die Berechnung des Loss voraussichtlich „Was“ mit „Was“ verglichen, wodurch zwar keine perfekten, aber annehmbare Trainingsergebnisse entstehen, da die Generierung der annotierten Schreibweise weiterhin stark ähnelt.

Bei den Epochen orientiert sich das Training an den Angaben für die originale MaPP-Implementierung. Dort wurde für je fünf Beispiele insgesamt 100 Epochen trainiert, sodass insgesamt $5 \cdot 100 = 500$ Trainingsschritte durchgeführt werden. Das Training wird für fünf Anläufe wiederholt, sodass insgesamt 2500 Schritte pro Pseudowort durchgeführt werden. Da die Maskierung der verschiedenen KE-Tokens zur Disambiguierung des KE-lex-Pseudoworts alterniert wird, wird folgende Anzahl an Trainingsschritten festgelegt: fünf Anläufe pro Konfiguration der Maskierung, $10000/b$ Epochen bei b Beispielen. Die Gesamtzahl der Epochen wird hierbei mit der Begründung weiter angehoben, als dass Konstruktionen mehr Disambiguierungen umfassen als eine rein semantische, wie sie KARIDI et al. (2021) vornehmen.

3.3.3 Hardwareanforderungen bzgl. Llama 2 und BERT

Das Training der Pseudowort-Embeddings (siehe Kapitel 3.2.3) für BERT wurde parallel auf 15 GPUs des Typs *NVIDIA GeForce RTX 2060 SUPER* ausgeführt. Hierbei wurde das Konstruktikon als Eingabekorpus auf 15 gleiche Teile aufgeteilt, die unabhängig voneinander verarbeitet wurden, sodass eine erhebliche Reduktion der Trainingsdauer auf wenige Stunden erreicht werden konnte.

Die Untersuchungen hinsichtlich Llama 2 und BERT bzgl. der Unterscheidung von korrekten und inkorrekten Sätzen zu gegebenen Konstruktionen (siehe Kapitel 3.2.2) benötigten eine von *Google Colab* verwaltete Instanz einer GPU des Typs *NVIDIA A100-SXM4-40GB*.

4 Generierung von Textbeispielen zu Konstruktionen

4.1 Anforderungen an generierte Beispiele für Konstruktionen

In diesem Teil der vorliegenden Arbeit soll das Gegenstück zu der in Kapitel 3 betrachteten Problemstellung diskutiert werden. Anstatt bestehende Beispiele durch Sprachmodelle zu erkennen, werden im Folgenden mögliche Lösungswege analysiert, mithilfe derer zu einer gegebenen Konstruktion entsprechend neue Beispiele generiert werden sollen. Ein solcher generierter Beispielsatz soll hierbei im Idealfall einige Anforderungen erfüllen:

- Das Beispiel soll möglichst vollständig automatisch generiert werden und so wenig wie möglich manuelle Eingaben benötigen;
- der Beispielsatz soll neu sein und nicht der Sammlung an gegebenen Beispielen zu einer Konstruktion entstammen;
- das neue Beispiel soll möglichst kreativ sein, sodass etwa die linguistische Forschung interessante Phänomene beobachten kann, welche bisher potenziell unbeachtet geblieben sind.

Diese Kriterien sollen anschließend für die Evaluation der Methoden herangezogen werden, nachdem diese im Folgenden untersucht worden sind.

Die Datenbasis für die folgenden Experimente und Implementierungen stellt weiterhin das Konstruktikon des Deutschen dar, sowie erneut die deutschen bzw. multilingualen Versionen der Sprachmodelle Llama 2 und mBART. Letzteres Modell wird außerdem analog zum Training für BERT aus Kapitel 3.2.3 um Pseudowörter erweitert, sodass diese auf den Nutzen in dieser Aufgabenstellung überprüft werden können. Es folgt zunächst ein Entwurf der algorithmischen Vorgehensweise, bevor abschließend die Ergebnisse evaluiert sowie einige Details der Implementierung erörtert werden.

4.2 Konzeption der Untersuchung hinsichtlich der Generierung von Beispielsätzen

Analog zur Verfahrensweise in Kapitel 3 wird zunächst analysiert, inwieweit Sprachmodelle die beschriebenen Anforderungen ohne besondere Anpassungen erfüllen können und ob sie unter Umständen direkt für die Generierung neuer Beispiele geeignet sind. Danach werden die im Rahmen des Kapitels 3.2.3 trainierten Pseudowort-Embeddings erneut einbezogen. Es gilt hierbei herauszufinden, ob Pseudowörter für ein Sprachmodell hilfreiche Informationen beinhalten, welche bei der Erstellung passender Sätze von Nutzen sind. Abschließend soll ein Konzept beschrieben werden, nach welchem die beschriebenen Ansätze in einem kombinierten Verfahren mit Ideen zur Erkennung von Konstruktionen (siehe Kapitel 3.2.2) verknüpfbar sind.

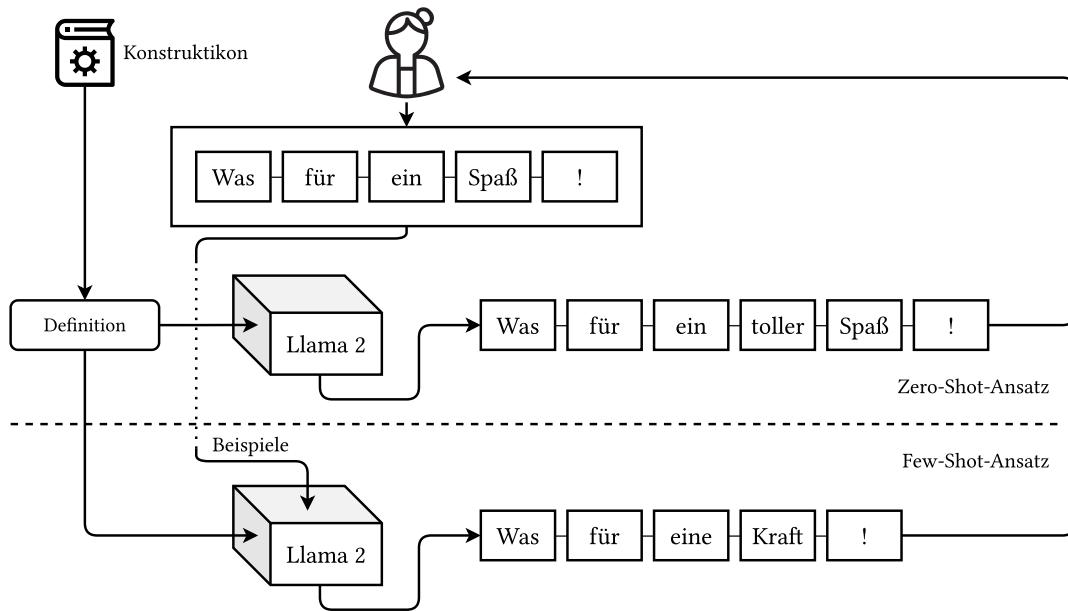


Abb. 4.1: Konzept der Zero- bzw. Few-Shot-Ansätze bei der Generierung von Beispielsätzen für eine gegebene Konstruktion mit Llama 2

4.2.1 Einsatz vortrainierter Sprachmodelle als Generatoren für Konstruktionsbeispiele

Die zu untersuchenden Modelle sind aufgrund ihrer individuellen Architekturen in unterschiedlichem Maße zu einer Generierung von Textbeispielen fähig. BERT beispielsweise ist vor allem darauf trainiert, zu einem maskierten Token in einem Satz das wahrscheinlichste zurückzugeben (MLM) oder die Wahrscheinlichkeit eines Folgesatzes zu ermitteln (NSP). Nicht durch BERT und ebenso wenig durch mBART wird die Verwendung von Prompts unterstützt, welche die vorliegende Aufgabe des Sprachmodells genauer erläutern würde. Da zumindest aber mBART die Fähigkeit besitzt, längere Textgenerierung vorzunehmen, wird dieses für eine Untersuchung berücksichtigt. Außerdem wird das Sprachmodell Llama 2 verwendet, welches ein auf die Aufgabenstellung angepasstes Prompting erlauben.

Die zwei letztgenannten Modelle sollen unter vergleichbaren Gesichtspunkten zunächst hinsichtlich der Fähigkeit untersucht werden, ob sie Konstruktionen einer Konstruktionsgrammatik definitorisch erfassen können und entsprechende Beispiele generiert werden können. Hierzu werden zwei Varianten ausprobiert: Zunächst wird ein Zero-Shot-Ansatz versucht, in welchem nur die grundlegenden Anweisungen zur gewünschten Ausgabe sowie die Definition einer Konstruktion übergeben werden. Im zweiten Durchlauf werden in einem Few-Shot-Ansatz zusätzlich Beispielsätze übergeben. Damit erhalten die Modelle zusätzliche Informationen, analog dazu, wie sie im Kapitel 3.2.3 auch für das Training von zusätzlichen Pseudowort-Tokens für BERT bzw. mBART eingesetzt werden. Die Qualität der Ausgaben wird anschließend den in Kapitel 4.1 vorgeschlagenen Maßstäben entsprechend beurteilt.

4.2.2 Zero-Shot-Ansatz mit Llama 2

Das Konzept einer Generierung von Beispielsätzen mit Llama 2 basiert auf dem Gedanken, dem Sprachmodell Informationen in unstrukturierter Weise zu übergeben, da es auf dieser Art von Datenverarbeitung spezialisiert ist. Im ersten Versuch wird Llama 2 somit ausschließlich die Definition einer Konstruktion übergeben, die aus dem Konstruktikon entnommen werden kann.

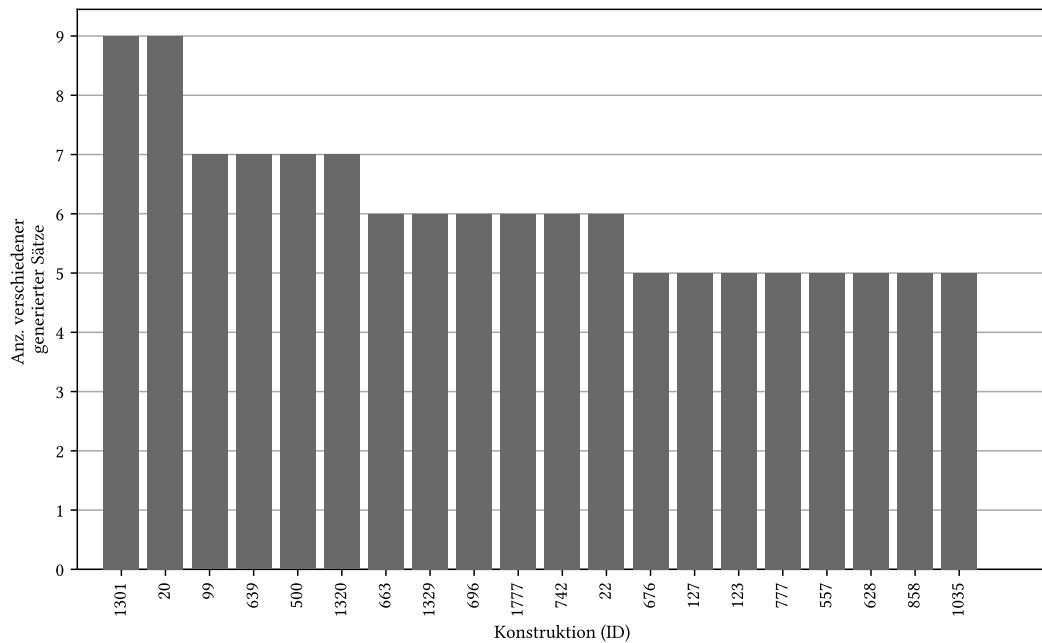


Abb. 4.2: Anzahl verschiedener generierter Sätze pro Konstruktion durch Llama 2 mittels Zero-Shot-Ansatz (Ausschnitt der besten 20)

Abbildung 4.1 stellt dar, dass eventuelle Beispiele in diesem Zero-Shot-Ansatz nicht hinzugefügt werden, sodass darauf vertraut werden muss, dass das Modell allein auf Grundlage der Informationen aus dem Konstruktikon fähig ist, Sätze anhand der Beschreibungen zu erstellen. Auf weitere Ergänzungen, etwa Begriffsklärungen aus dem Glossar des Konstruktikons (z. B. zu KE-lex oder KE) wird bewusst verzichtet. Da diese bereits in den meisten Definitionen der Konstruktionen noch einmal kurz erläutert werden und außerdem die maximale Eingabelänge für Sprachmodelle beschränkt ist, soll im Folgenden nur die Beschreibung der jeweiligen Konstruktion selbst eingegeben werden.

Die Ergebnisse nach einem Zero-Shot-Ansatz sind zumindest unter der Verwendung des Modells Llama 2 nicht konsistent und erscheinen hinsichtlich ihrer Qualität wechselhaft. Die fehlerhaften Ausgaben lassen sich jedoch kategorisieren und jeweils verschiedene Ursprünge vermuten. Zum einen bestehen einige Ausgaben zum Großteil aus sogenanntem *Echoing* oder *Parroting* (vgl. BENDER et al., 2021), wobei Wort- und Satzbestandteile aus der Eingabe wiederholt und teils ohne Sinnzusammenhang verwendet werden. So finden sich definitorische Begriffe wie „Konstruktion“ oder „äquativ“ genauso wieder wie Teile aus der Anweisung zum Ausgabeformat („kreativ“, „gewissenhaft“). In der praktischen Implementierung wird daher die Generierung von vornherein wiederholt, bis solche typischen Begriffe in der Ausgabe nicht mehr zu finden sind.

Abbildung 4.2 zeigt die 20 Konstruktionen, deren Definition als Eingabeprompt die jeweils größte absolute Anzahl an Beispielen erzeugen konnten.¹ Auffällig ist, dass es sich bei diesen Konstruktionen um solche handelt, deren definitorische Beschreibung im Konstruktikon eher knapp zu sein scheint. Allerdings ist zu beachten, dass diese Anzahl nur einen ersten Anhaltpunkt dafür liefert, wie gut die Leistung von Llama 2 beim Zero-Shot-Prompting auf Konstruktionsdefinitionen ist. Zwar wurden bei den angegebenen Ergebnissen bereits doppelte Nennungen getilgt, die Ergebnisse lassen sich aber noch differenzierter betrachten. Hierfür kann etwa überprüft werden, wie viele der zurückgegebenen Beispiele aus Elementen des Eingabeprompts zusammengesetzt sind. Auf diese Weise kann ein mögliches Parroting des Sprachmodells aufge-

¹Die vollständigen Ergebnisse finden sich im Anhang in den Abbildungen A.1 und A.2.

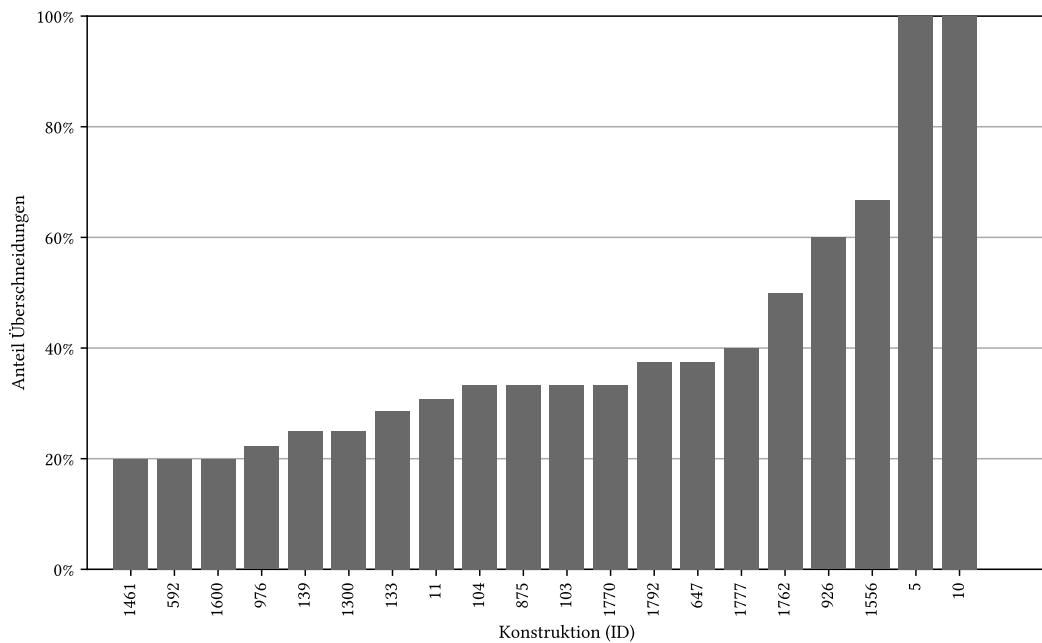


Abb. 4.3: Anteil an Überlappungen zur gegebenen Definition in den durch Llama 2 generierten Sätzen mittels Zero-Shot-Ansatz pro Konstruktion (Ausschnitt der schlechtesten 20)

schlüsselt werden. In Abbildung 4.3 sind die Konstruktionen dargestellt, die am stärksten von Wiederholungen aus dem Eingabeprompt betroffen sind. Anhand der Konstruktion 1777 („Korrelativer Äquativ:VP_in_dem_Maße_XP“) ist exemplarisch eine Überschneidung zwischen den beiden Metriken erkennbar. Während sie die dritthöchste Anzahl an Beispielen produziert, lassen sich 40 Prozent dieser als direkte Wiederholungen aus der Definition identifizieren.

Das beschriebene Verhalten von Llama 2 lässt darauf schließen, dass die reine Angabe einer fachlichen Definition zum einen zu komplex sein könnte und zum anderen zu wenig konkrete Beschränkungen zur Satzstruktur vorgibt. Deutlich ist auch der Unterschied zwischen den Modellen: Während Llama 2 eher selten tatsächlich korrekte Satzbeispiele liefert, entsprechen Ausgaben, die probeweise mit ChatGPT durchgeführt wurden, zumindest stärker der gewünschten Konstruktion. Dennoch haben diese noch keine zureichende Qualität, wie die folgende Ausgabe zeigt:

Eingabe:

„Hier ist eine Definition:

Die "Superlativ_Klimax:ADJ1_ADJ1-er_NP"-Konstruktion gehört zu der Familie [...].

Nenne passende Sätze zu dieser Definition im Format einer Python-Liste. Gib nur die Liste aus.
Gib nichts anderes aus.“

ChatGPT:

[

„Die Intelligenz dieses Schülers ist bemerkenswert, intelligenter als die meisten seiner Mitschüler und der intelligenteste in der Klasse.“,

„Die Geschwindigkeit dieses Autos ist beeindruckend, schneller als viele andere Modelle und das schnellste auf dem Markt.“,

„Die Schönheit dieses Gemäldes ist außergewöhnlich, schöner als viele andere Kunstwerke und das schönste in der Galerie.“

]

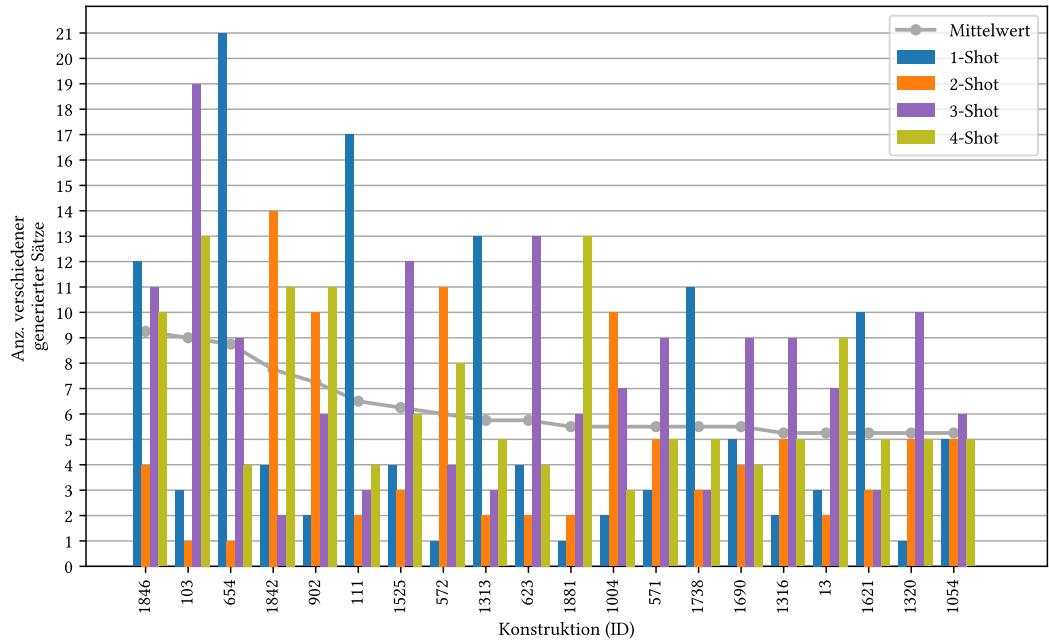


Abb. 4.4: Anzahl verschiedener generierter Sätze pro Konstruktion durch Llama 2 mittels Few-Shot-Ansätzen (Ausschnitt der besten 20)

Feinheiten einzelner Definitionen, wie die der „geschweige denn“-Konstruktion, werden nicht erfasst, sodass auch etwa Beispiele geliefert werden, die zwar die Phrase „geschweige denn“ enthalten, der restliche Satz aber etwa keinen Negator als notwendiges Kern-KE beinhaltet.

4.2.3 Few-Shot-Ansatz mit Llama 2

In einem weiteren Versuch wird der in Kapitel 4.2.2 verwendete Prompt um einige Beispiele aus den jeweiligen Konstruktikon-Einträgen erweitert. Der Anreiz ist, auf diese Weise dem Sprachmodell wichtige Anhaltspunkte zum Format der gewünschten Ausgabe zu liefern. Somit kann Llama 2 in der Generierung auf praktische und a priori als korrekt befundene Beispiele zurückgreifen. BROWN et al. (2020) illustrieren die Nützlichkeit eines Few-Shot-Learnings ihrerseits in Verbindung mit der Einführung von GPT-3, dessen Konzept in einer späteren Variante als ChatGPT bekannt wurde. Llama 2 konkurriert als Sprachmodell mit diesen, wobei auch hier indirekt und auf Grundlage der erzielten Leistungen ein Few-Shot-Ansatz durch TOUVRON et al. (2023b) nahegelegt wird.

Der im Folgenden beschriebene Versuch wird als Few-Shot-Prompting nacheinander mit einem, zwei, drei und vier zufällig gewählten Beispielsätzen durchgeführt. Hierbei wird weiterhin die Definition der gewünschten Konstruktion dem Prompt hinzugefügt, bevor die Beispiele ergänzt werden (siehe Abbildung 4.1 unten). Zu erwarten ist eine geringere Neigung zu Parroting durch das Sprachmodell, da im besten Fall die Konstruktion durch verschiedene Varianten für Llama 2 illustriert wird.

Beim direkten Vergleich der Konstruktionen mit den meisten generierten Beispielen zeigt Abbildung 4.4 kaum Übereinstimmungen mit dem Zero-Shot-Ansatz (siehe Abbildung 4.2). Dies erhärtet den Verdacht, dass es zwischen den verschiedenen Prompt-Konfigurationen einen deutlichen Unterschied bei der Verständnisfähigkeit und Produktivität des Sprachmodells gibt. Es ist zu erkennen, dass eine höhere Anzahl generierter Beispiele zu einem gewissen Grad mit der Anzahl der Beispiele im Prompt korreliert. Für den 3- und den 4-Shot-Ansatz ist zu sehen, dass diese

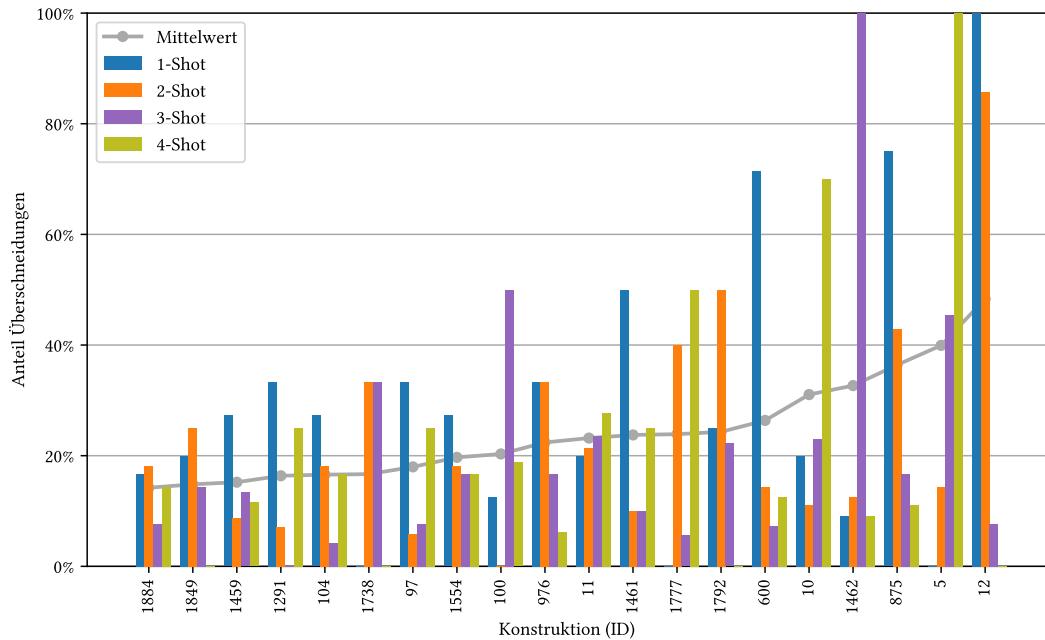


Abb. 4.5: Anteil an Überlappungen zur gegebenen Definition in den durch Llama 2 generierten Sätzen mittels Few-Shot-Ansätze pro Konstruktion (Ausschnitt der schlechtesten 20)

Mittelwerte	grammatisch (micro)	zutreffend (micro)	grammatisch (macro)	zutreffend (macro)
Zero-Shot	63,35 %	21,65 %	63,67 %	21,12 %
3-Shot	78,93 %	67,26 %	68,74 %	56,97 %

Tab. 4.1: Grammatikalität mittels Llama 2 generierter Sätze und deren Korrektheit bzgl. der jeweils gegebenen Konstruktion nach menschlicher Evaluation

meist überdurchschnittlich viele Ergebnisse pro Konstruktion erzeugen.

Bei der Betrachtung der Überlappungen der generierten Beispiele mit der Eingabe (siehe Abbildung 4.5) wird nun auch berücksichtigt, ob sich die im Rahmen des Few-Shot-Prompts übergebenen Beispiele in den Ausgaben widerspiegeln. Hierbei ist eine deutlichere, wenngleich keine vollständige Überschneidung mit den Ergebnissen des Zero-Shot-Ansatzes (Abbildung 4.3) zu beobachten. So befinden sich etwa die beiden Konstruktionen 5 und 10 mit jeweils vollständiger Überlappung beim Zero-Shot- ebenfalls unter den fünf dahingehend auffälligsten Konstruktionen im Few-Shot-Ansatz. Insgesamt lässt sich beobachten, dass es im letzteren Fall tendenziell seltener zu Parroting-Phänomenen kommt, während die Anzahl an insgesamt generierten Konstruktionen vergleichbar hoch bleibt.

Zwischen den verschiedenen Ausprägungen des Few-Shot-Promptings (d. h. in Bezug auf die Anzahl verwendeter Beispiele) lässt sich kein deutlicher Unterschied in der Qualität feststellen. Dies deckt sich mit den Ergebnissen von Llama 2 auf bekannten Benchmark-Tests wie SQuAD bzw. Weiterentwicklungen desselben (RAJPURKAR et al., 2018), welche ebenfalls das Textverständnis von Sprachmodellen evaluieren. Vergleicht man die Ergebnisse von Zero-Shot- zu Few-Shot-Ansätzen nach menschlicher Evaluation, wie in Tabelle 4.1 dargestellt, so zeigt sich ein deutlicher

4.2 Konzeption der Untersuchung hinsichtlich der Generierung von Beispielsätzen

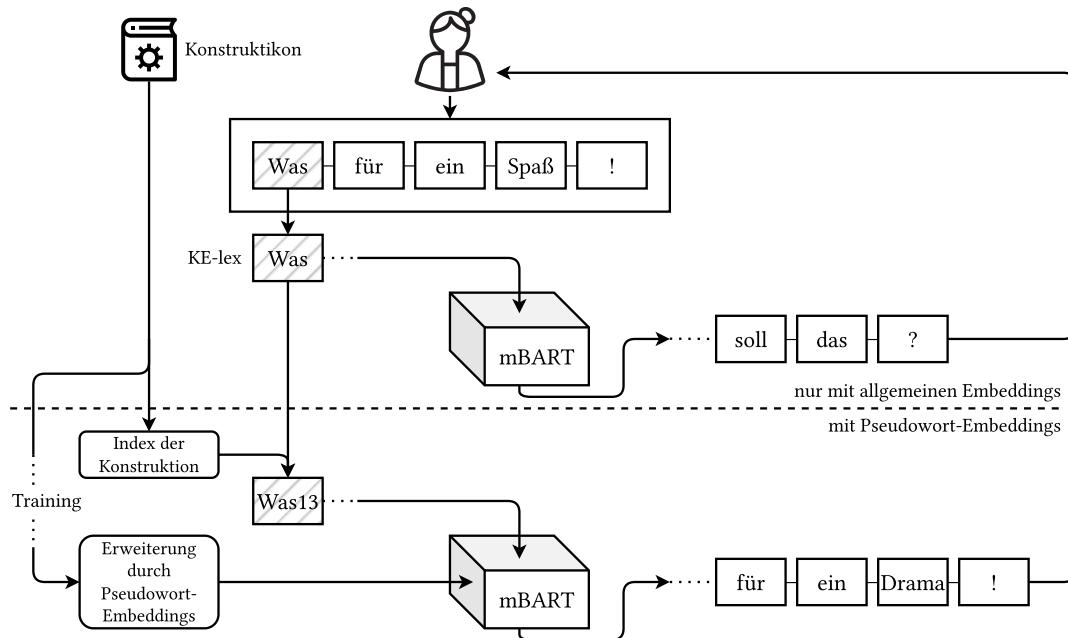


Abb. 4.6: Generierung von Beispielsätzen durch Kürzen und Fortsetzen des Eingabesatzes ab dem KE-lex-Token (ohne und mit Pseudowort-Embeddings); zu erwarten ist, dass eine Generierung von unspezialisierten Embedding ausgehend eher zu ungewünschten Ausgaben führt (oben)

Qualitätszugewinn beim Ergänzen der Definitions-Prompts durch Beispiele.² Zum einen ist zu erkennen, dass die sprachliche Korrektheit bzw. Grammatikalität der generierten Beispielkandidaten deutlich steigt, zum anderen bilden danach weit über die Hälfte der Sätze tatsächliche Repräsentationen für die jeweils gegebenen Konstruktionen. Hierbei wurden auch Sätze als zutreffend bewertet, die zwar nicht vollständig oder grammatisch unsauber sind, aber dennoch augenscheinlich das Wesen der jeweiligen Konstruktion erfassen. Somit soll möglichst die Aufgabe, grammatisch einwandfreie Sätze zu konstruieren von der Fähigkeit, Konstruktionsdefinitionen zu erfassen, getrennt werden. Beide Attributwerte können jedoch durch ein Few-Shot-Learning deutlich gesteigert werden.

4.2.4 Generierung von Beispielen ab einem KE-lex

Eine alternative Herangehensweise, welche jedoch auf eine vollautomatische Generierung von Beispielen verzichtet, ist es, die definierte Struktur von Konstruktionen im Konstruktikon des Deutschen auszunutzen. Hierin ist für den Großteil der nachgewiesenen Konstruktionen jeweils ein oder mehrere KE-lex annotiert, dessen Vorhandensein für das Zustandekommen einer Konstruktion notwendig ist.

Da generative Sprachmodelle wie Llama 2 oder auch mBART zu einer gegebenen Folge von Tokens eine Ergänzung generieren, die aus jeweils den nächstwahrscheinlichen Elementen besteht, kann diese Funktionsweise direkt ausgenutzt werden. Das Modell mBART wird neben Llama 2 im Rahmen dieser Arbeit bewusst ausgewählt, da es die Fähigkeit besitzt, zu gegebenen Maskierungen im Satz Tokens variabler (!) Anzahl zu generieren. Diese Abwandlung des MLM von BERT (siehe hierzu Abbildung 1.2a in Kapitel 1.3.2) hilft im späteren Verlauf in Kapitel 4.3.2 bei der Generierung von Pseudowort-Embeddings, die auf mBART zugeschnitten sind. Dort wird auch genauer auf die zugrundeliegende Architektur eingegangen.

²Die vollständigen Rohdaten finden sich im Anhang in den Tabellen A.4 und A.5

Konstr.	Satz mit vorhergesagtem <mask>-Token
5	Und schon gar nicht \emptyset Mehrwertsteuer.
	Und schon gar nicht <i>was man</i> macht.
:	:
10	Geschweige denn \emptyset kippten.
	""Ohne Europa sind viele Fragen nicht mehr seriös \emptyset ""
:	:

Tab. 4.2: Ausschnitt aus den <mask>-Vorhersagen (kursiv gedruckt) durch mBART-50; \emptyset stellt eine Nullinstanziierung dar, mBART-50 sagt hier kein Token vorher

Für die Untersuchung im vorliegenden Kapitel werden bestehende, valide Beispielsätze aus dem Konstruktikon genommen und derart gekürzt, dass diese hinter dem KE-lex gestutzt werden bzw. ihre hinteren (rechtsseitigen) Tokens entfernt werden. Die so entstehende Reihe von Tokens kann nun in ein generatives Sprachmodell eingelesen werden. Das Modell generiert die folgenden Wörter, welche im Idealfall den gegebenen Anfang eines Beispielsatzes sinnvoll und somit gemäß der gegebenen Konstruktion ergänzen sollten. Am Beispiel von mBART ist dieses Verfahren in Abbildung 4.6 (oben) dargestellt.³ Problematisch ist bei der Kürzung von Beispielsätzen, dass möglicherweise Informationen verloren gehen können, die für eine Konstruktion notwendigerweise essenziell sind. Dazu gehören insbesondere (Kern-)Konstruktionselemente (KE), welche in Kombination mit dem KE-lex implizit festlegen, ob es sich im vorliegenden Fall in hinreichender Weise um eine Konstruktion handelt. Befindet sich dieses rechts vom KE-lex, kann in dieser Konfiguration des Versuchs nicht sichergestellt werden, dass die Generierung der gewünschten Konstruktion entspricht.

Ein ähnliches Problem stellen Konstruktionen dar, deren KE-lex üblicherweise besonders weit links steht. In diesem Fall wird möglicherweise unabhängig vom Verbleib weiterer KEs zu viel (oder jegliche) Information entfernt, die in Anbetracht der Konstruktion zur Generierung sinnvoller Sätze unerlässlich ist. Diesem Problem kann zumindest teilweise Abhilfe geschaffen werden: Das Konstruktikon des Deutschen, welches sich ausschließlich auf real existierende Beispielsätzen aus der Literatur beruft, führt Buch über den jeweils vorangehenden und nachfolgenden Satz eines Beispiels, um den Kontext beim Verständnis einer Konstruktion besser zu erläutern.

Für Konstruktionsbeispiele, welche mit dem KE-lex beginnen, wird daher der vorhergehende Satz ergänzt, bevor das Modell diesen durch Generierung ergänzt. Da zudem mBART mit Mask-Language-Modelling arbeitet, werden zum Einfassen der Maskierung auch die letzten beiden Worttokens ergänzt – im Allgemeinen handelt es sich hierbei um ein Satzzeichen und das letzte Wort. Die stichprobenweise beobachteten Ergebnisse (siehe Tabelle 4.2) erhärten den Verdacht, dass dieser Ansatz zwar teilweise zu grundsätzlich grammatischen Sätzen führt, aber nur selten die Essenz einer gegebenen Konstruktion hinreichend widergespiegelt wird. Zu beobachten ist außerdem, dass es mBART vor Probleme stellt, sobald der Abstand zwischen KE-lex bzw. dem Beginn der Maskierung und dem Ende größer wird.

In Kapitel 4.3.1 wurden bereits positive Erfahrungen mit der Ergänzung des Modells durch Pseudowort-Embeddings gemacht, welche in den Sätzen anstelle der KE-lex-Tokens verwendet werden. Durch diese Erkenntnis motiviert soll auch ein solcher Versuch für mBART unternommen werden. Dementsprechend wird in Kapitel 4.3.2 ein vergleichbares Verfahren gewählt, bei

³Das Modell Llama 2 wird für diesen Versuch nicht weiter untersucht.

dem die KE-lex durch ihre entsprechenden Pseudowörter ersetzt werden, um das Problem des fehlenden Kontexts auszugleichen.

4.3 Verwendung von Pseudowörtern zur Generierung

In Kapitel 3.2.3 wurde bereits die Anwendung von Pseudowörtern auf das Modell BERT erfolgreich versucht. Im Folgenden werden diese Pseudowort-Embeddings im Rahmen von Experimenten in Hinblick auf die Beispielgenerierung angewendet. Dabei unterscheiden sich die Verfahren für BERT und mBART erheblich, da die Architektur und Funktionsweise des jeweiligen Modells sehr unterschiedlich sind. Die Untersuchungen bzgl. BERT sind hier allerdings eher als Machbarkeitsnachweis zu verstehen, um die grundsätzliche Fähigkeit von Sprachmodellen zu bestätigen, mithilfe von Pseudowörtern sinnvolle Ausgaben zu generieren. Anschließend wird dies auf das mBART-Modell ausgeweitet.

4.3.1 Pseudoword-Embeddings für Konstruktionen mit BERT

BERT ist üblicherweise darauf vorgenommen, zu einem Satz mit einem einzelnen maskierten Token eine Vorhersage für ebendieses Element zu tätigen, was zu nötigen Kompromissen bei der Implementierung und darauffolgend für das Training bedeuten, die in Kapitel 3.3.2 erklärt werden.

Um die Qualität der Pseudowort-Embeddings in Hinblick auf Generierungsaufgaben zu evaluieren bzw. herauszufinden, inwiefern die Embeddings die Informationen zu den jeweiligen KE-lex und somit Konstruktionen im Training verinnerlicht haben, werden alle gegebenen Beispielsätze nacheinander alternierend maskiert. Verwendet werden die bereits gemäß dem Training aus Kapitel 3.2.3 erstellten Pseudowort-Embeddings. Diese neuen Embeddings werden in ein unverändertes BERT-Modell als zusätzliche Embeddings geladen. Bei der Vorhersage der Maskierungen wird dann jeweils das erste Token akzeptiert, das

1. nicht zu den neu hinzugefügten Pseudowort-Tokens gehört;
2. nicht zu möglichen Platzhalter-Tokens des Modells gehört und
3. nicht exakt das gleiche Token wie im Originalsatz ist.

Diese Bedingungen entsprechen indirekt auch den Voraussetzungen, die in Kapitel 4.1 festgelegt wurden. Darauf aufbauend kann bewertet werden, ob die Generierungen für die jeweilige Konstruktion akzeptabel sind und somit neue Beispiele für eine gegebene Konstruktion darstellen.

Eine qualitative Überprüfung der Ausgaben bestätigt, dass die Generierungen weitgehend plausibel zu sein scheinen (siehe Tabelle 4.3). Allerdings ist hierbei kritisch zu hinterfragen, wie viele tatsächlich neue Daten für eine gegebene Konstruktion produziert werden. Während Llama 2 und mBART zum Großteil neue Sätze generieren, produziert BERT stets nur ein neues Token. Die tatsächliche Leistung des Sprachmodells im Verhältnis zu den manuell zu übergebenen Daten ist also vergleichsweise gering.⁴ BERT eignet sich deshalb nur sehr begrenzt für eine automatische Generierung von Beispielsätzen, bietet aber für diese Arbeit zumindest einen Anhaltspunkt, ob Pseudowörter grundsätzlich zu sinnvollen Ergebnissen in Bezug auf Konstruktionen führen bzw. sie zumindest nicht negativ beeinflussen. Diese Erkenntnis ist bedeutsam, da sie für das weitere Verfolgen des Pseudowort-Ansatzes für mBART die entsprechende Motivation liefert.

Weiterhin ist eine Erweiterung von BERT mit Pseudowörtern denkbar, um beispielsweise Sätze abzuwandeln, welche bewusst nicht als Originalzitat verwendet werden sollen oder dürfen. Hier sind jedoch ethische Standpunkte zu berücksichtigen. Auf diese Weise könnten (unbewusst)

⁴ Aus diesem Grund werden die Ergebnisse selbst in dieser Arbeit nicht näher dargestellt, da sie lediglich als *Proof of Concept* dienen und eine ausführliche menschliche Annotation der Ausgaben nötig wäre.

Konstr.	Satz mit vorhergesagtem [MASK]-Token
5	[CLS] Und5 schon gar nicht <i>bei</i> der Mehrwertsteuer . [SEP]
	[CLS] Und5 schon gar nicht mit <i>der</i> Mehrwertsteuer . [SEP]
	[CLS] Und5 schon gar nicht mit der <i>Polizei</i> . [SEP]
:	:
	[CLS] Und5 schon gar nicht , <i>weil</i> das neue System die Schüler schlauer macht . [SEP]
	[CLS] Und5 schon gar nicht , dass das neue <i>Konzept</i> die Schüler schlauer macht . [SEP]
:	:
10	[CLS] Geschweige10 denn , dass <i>sie</i> die Argumentation einfach kippten . [SEP]
	[CLS] Geschweige10 denn , dass sie <i>ihre</i> Argumentation einfach kippten . [SEP]
	[CLS] Geschweige10 denn , dass sie die <i>Autos</i> einfach kippten . [SEP]
:	:
	[CLS] » Ohne <i>sie</i> sind viele Fragen nicht mehr seriös in Angriff zu nehmen , geschweige denn10 zu lösen « , meint der grüne Bundesparteisekretär Lothar Lockl . [SEP]
	[CLS] » Ohne Europa <i>sind</i> viele Fragen nicht mehr seriös in Angriff zu nehmen , geschweige denn10 zu lösen « , meint der grüne Bundesparteisekretär Lothar Lockl . [SEP]
	[CLS] » Ohne Europa sind <i>diese</i> Fragen nicht mehr seriös in Angriff zu nehmen , geschweige denn10 zu lösen « , meint der grüne Bundesparteisekretär Lothar Lockl . [SEP]
:	:

Tab. 4.3: Ausschnitt aus den [MASK]-Vorhersagen (kursiv gedruckt) durch BERT unter Anwendung von Pseudowörtern

mögliche Urheberrechtsverstöße begangen werden, welche durch den Austausch einzelner Tokens weiter bestehen bleiben oder sogar verschärft werden. Die Chancen der möglichen Anwendung von BERT im Rahmen von Generierungen für Konstruktionsgrammatiken ist also tendenziell als Ergänzung zu einer manuellen Annotation zu verorten.

4.3.2 Erweiterung von mBART mit Pseudowörtern

Unter BERT ist es im Speziellen nicht sinnvoll (und ohne entsprechende Anpassungen kaum möglich), mehr als ein Token in einem gegebenen Kontext gleichzeitig vorherzusagen. Somit eignet sich BERT gemäß Kapitel 4.3.1 nur sehr begrenzt für Textgenerierungen, da dieses Sprachmodell darauf trainiert ist, nur einzelne Tokens vorherzusagen. Idealerweise könnte also auf ein Sprachmodell zurückgegriffen werden, welches Sequenzen beliebiger Länge maskieren und generieren kann.

Ein Modell, das diese Voraussetzungen erfüllt, stellt BART (LEWIS et al., 2020) dar, welches als ein sogenanntes Sequence-to-Sequence-Modell (Seq2Seq) hauptsächlich zur maschinellen Übersetzung verwendet wird. Das BART-Modell bzw. dessen multilinguale Version mBART unterscheiden sich konzeptionell in einigen Punkten von BERT. Die BART-Architektur ist sowohl aus einem Encoder- als auch ein Decoder-Bestandteil aufgebaut und bildet dahingehend das ursprüngliche Transformer-Konzept vollständig ab (vergleiche hierzu Abbildung 1.1 sowie Kapitel 1.3.2). Der Encoder entspricht hierbei zum Großteil dem BERT-Modell und verfolgt demnach ein MLM, während der Decoder autoregressiv von links nach rechts arbeitet und somit nach dem Vorbild von GPT (RADFORD et al., 2018) angelegt ist. Für die Anwendung des Pseudowort-Ansatzes auf Konstruktionsgrammatiken ist insbesondere relevant, dass BART neben dem BERT-ähnlichen *Token Masking* auch ein *Text Infilling* unterstützt, welches eine Maskierung von Sequenzen aufeinanderfolgender Tokens in unterschiedlicher Länge erlaubt (vgl. LEWIS et al., 2020).

Da das Konstruktikon, das im Rahmen dieser Arbeit betrachtet wird, die deutsche Sprache untersucht, wird das mit BART direkt verwandte mBART-50 (TANG et al., 2021) verwendet. Dieses Modell unterstützt 50 Sprachen inklusive Deutsch. Die weiteren Sprachen können zudem hilfreich beim korrekten Erkennen von beispielsweise Fremdwörtern oder Eigennamen sein.

Im direkten Vergleich zum Pseudoword-Ansatz von KARIDI et al. und dessen Anwendung auf Konstruktionserkennung mit BERT in Kapitel 3.2.3 müssen einige strukturelle Anpassungen vorgenommen werden, die in Abbildung 4.7 dargestellt sind. Die Grundidee ist es, zu jedem KE-lex der verschiedenen Konstruktionen des Konstruktikons jeweils ein neues Embedding zu trainieren. Als Trainingsdaten dienen hierfür die vordefinierten Beispiele aus dem Konstruktikon selbst, sodass die Beispiele nicht erneut manuell erstellt werden müssen. Die Maskierungen werden vorgenommen, indem alle zusammenhängenden, als KE annotierten Satzbestandteile durch das mBART-50-spezifische `<mask>`-Token ersetzt werden. Zusätzlich wird das KE-lex selbst temporär durch ein neues Token (#TOKEN#) ersetzt. Somit kann es durchaus vorkommen, dass ein KE-lex, welches aus mehreren aufeinanderfolgenden Wörtern besteht, in dieser Modellierung auf ein einzelnes neues Token abgebildet wird. So ist es etwa denkbar, dass das in einem vorherigen Beispiel genannte „geschweige“, welches durch mBART in mehrere Tokens unterteilt wird, durch ein neues, gemeinsames Token repräsentiert wird. Die spätere Benennung dieser neuen Tokens erfolgt durch die Konkatenation aller KE-lex-Tokens und der Nummer der zugehörigen Konstruktion, z. B. „geschweige10“. Dies ist nötig, da im Deutschen teilweise gleiche Wortformen für unterschiedliche Konstruktionen verwendet werden.

Der in Abbildung 4.7 dargestellte *MSELoss* entspricht weitgehend der Definition, wie sie in Gleichung 4.1 für BERT definiert ist. Erweitert wurde das Konzept lediglich um eine Multiplikation mit der Anzahl der durch das `<mask>`-Token ersetzen ursprünglichen Tokens. Dadurch ist

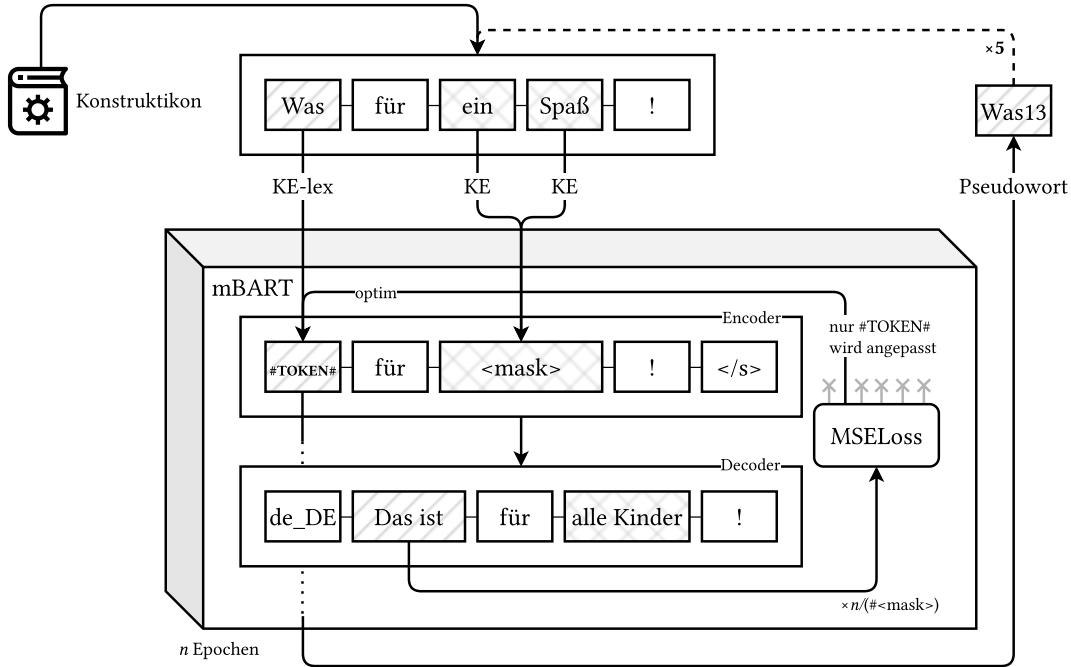


Abb. 4.7: Training eines Pseudowort-Embeddings für mBART auf Basis eines Beispielsatzes des Konstruktionskons

der Loss im Allgemeinen größer, je mehr Tokens maskiert wurden. Der Gedanke hierbei ist es, das mBART-Modell beim Training der Pseudowörter anzuregen, die Gewichte auch dann noch stark anzupassen, wenn die restlichen Elemente des Satzes bereits gute Ergebnisse liefern. Da der Loss bei einem schlechten Vergleichswert für das Pseudowort-Embedding entsprechend höher ist, werden auch die Gewichte in diesem Fall weiter stark angepasst. Die Gleichung für den aggregierten Loss im Kontext von mBART lautet demzufolge:

$$\mathbf{z}_\eta^* = \arg \min_{\mathbf{z} \in \mathbb{R}^d} \frac{1}{n} \left(\sum_{i=1}^n |\text{mBART}(\mathbf{z})| \cdot \|\text{mBART}(\mathbf{z}) - \mathbf{x}_t^{(i)}\|^2 \right). \quad (4.1)$$

Ausschlaggebend für diese Entscheidung war die Beobachtung, dass die Beispiele im Konstruktionskon vergleichsweise wortreich sind. Zugleich sind auch die nötigen Maskierungen, Anzahl an Tokens usw. komplexer als bei BERT, weshalb gerade bei längeren Sätzen der Einfluss des KE-lex im Satz aus Sicht des Modells nicht mehr groß genug ist, um noch ein starkes Training desselben zu bedingen.

Ein weiteres zu beobachtendes Problem beim Training des Modells ist, dass durch den direkt an die Ergebnisse des Encoders angeknüpften Decoder-Teil eine teils unvorhersehbare Verschiebung des ursprünglichen Eingabesatzes zustandekommt. Da es sich bei mBART zudem um ein sogenanntes Seq2Seq-Modell handelt, welches häufig für Übersetzungsaufgaben angewendet wird, kann schließlich nicht immer sichergestellt werden, dass der Ausgabesatz hinreichend mit dem Eingabesatz bezüglich dessen Tokens übereinstimmt. Es ist anzunehmen, dass das Modell gewisse Freiheiten bei der „Übersetzung“ nutzt, selbst wenn Eingabe- und Zielsprache identisch sind. Abbildung 4.7 zeigt dieses Problem anhand der Vorhersage im Trainingsschritt, bei dem „alle Kinder“ zwar mit gleich vielen Token vorhergesagt wird wie mit „ein Spaß“ auch die Eingabe der Maskierung besitzt. Allerdings können auch andere Tokens variable Längen und sogar Sortierungen für ihre Entsprechungen nach der Decoder-Verarbeitung erhalten; hier etwa „Was“, das durch „Das ist“ vorhergesagt wird. Diese Ungenauigkeit hat einen so massiven

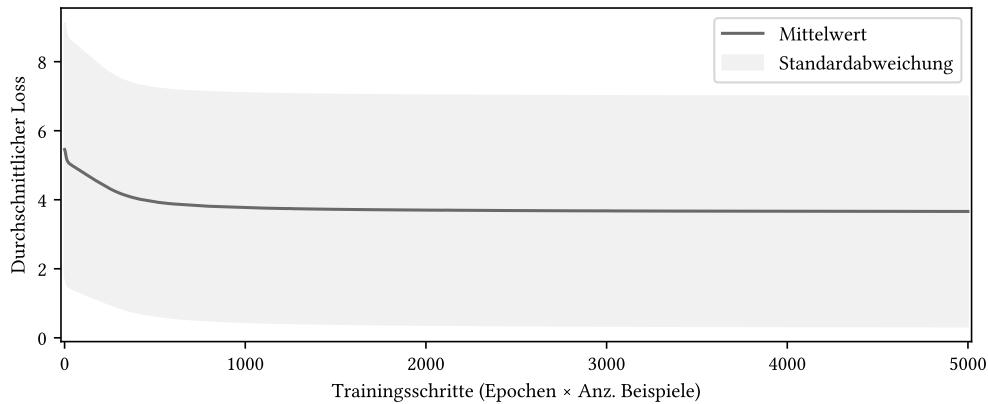


Abb. 4.8: Konstruktionsübergreifende gemittelte Entwicklung des Loss-Werts für das Training von Pseudowort-Embeddings für mBART

Einfluss auf das Training, dass für einige Konstruktionen nur unzureichende Trainingsergebnisse erzielt werden konnten. Der über alle Pseudowort-Trainingsanläufe gemittelte Verlauf des Losses in Abbildung 4.8 zeigt, dass der Loss mit der Zeit langsam aber stetig sinkt und sich für einige Beispiele sogar dem Wert 0 annähert. Andererseits bleibt er im Schnitt auf einem hohen Niveau, was sich nur zum Teil mit dem zusätzlichen Längen-Koeffizienten in der angepassten Loss-Funktion erklären lässt. Es findet zwar nachweislich ein Trainingseffekt statt, der aber sehr langsam und unpräzise erfolgt. Diese Vermutung bestätigt sich bei der Betrachtung der Ergebnisse beim Versuch, mit den mBART-Pseudowörtern korrekte Generierungen für Konstruktionen hervorzu bringen.

Die Auswertung des Ansatzes erfolgt, indem die trainierten neuen Embeddings an das Vokabular eines neu aufgesetzten mBART-50-Modells angefügt werden. Die ursprünglichen Entsprechungen der neuen Embeddings bleiben somit im Modell erhalten. Im Beispiel oben hat das Modell nach dem Anfügen der neuen Embeddings also jeweils Repräsentationen für „geschweige10“, aber weiterhin auch für die allgemeine Variante „geschweige“ bzw. deren untergeordnete Tokens. Außerdem werden die neuen Embeddings als Ausgaben für den Decoder nicht erlaubt, analog dem Training der Pseudowort-Embeddings. Ob die neue Repräsentation eines KE-lex die Eigenschaften der jeweiligen Konstruktion hinreichend gut erfasst, wird geprüft, indem ein gegebener Beispielsatz bis zum neuen Token dem Modell übergeben wird. Die daraufhin generierten Ausgaben sollten zusammen mit den gegebenen Wörtern einen Satz bilden.

Leider zeigen allein die qualitativ geführten Beobachtungen auch im Vergleich zu denen aus Kapitel 4.2.4 für ein nicht erweitertes mBART-Modell nur selten sinnvolle Ausgaben (siehe Tabelle 4.4). Dies lässt sich auf mehrere Gründe zurückführen. Zum einen besteht weiterhin das Problem der teilweise komplexen Maskierungen bei der Generierung, welche offensichtlich nicht mehr gut durch mBART erfasst werden. Weiterhin wirken sich aber auch die zum Teil schlechten Loss-Reduktionen während des Trainings direkt auf den Erfolg und die Qualität der Generierungen aus, da sogar einige Tokens gar nicht mehr korrekt dargestellt werden.

Die Ergebnisse zeigen, dass zwar an manchen Stellen und auch während des Trainings erkennbar Konstruktionseigenschaften durch die neuen Embeddings festgehalten werden, jedoch größtenteils fehlerhafte und unbrauchbare Ausgaben generiert werden.

Schon während des Trainings der Pseudowort-Embeddings mit mBART-50 zeigt sich ein deutlicher Nachteil der Architektur gegenüber beispielsweise BERT. Da es sich bei BART bzw. mBART nicht um ein Modell handelt, das rein auf dem MLM basiert, ist das einfache Ergänzen eines maskierten Satzes weniger deterministisch als bei BERT. Das zeigt sich teils sehr deutlich, indem bei

Konstr.	Satz mit vorhergesagtem <mask>-Token
5	Er glaubt nicht an mehr Förderung, Ø. Er glaubt nicht an mehr Förderung, nicht an kleinere Klassen, nicht an individua- lisierteren Unterricht. Ø
:	:
10	Die Chance dazu haben sie, und sie haben sie in anderen Verfahren schon mehrfach genutzt. Ø »Ohne Europa sind viele Fragen nicht mehr seriös in Angriff zu nehmen, geschwei- ge... Lockl.....
:	:

Tab. 4.4: Ausschnitt aus den <mask>-Vorhersagen (kursiv gedruckt) durch mBART-50 unter Anwendung von Pseudowörtern; Ø stellt eine Nullinstanziierung dar, mBART-50 sagt hier kein Token vorher

der Eingabe eines Satzes auch Tokens durch mBART-50 abgeändert werden, die eigentlich nicht maskiert sind und bei denen keine Änderung erwartet wird. Eine mögliche Erklärung für das Verhalten liefert der Fakt, dass BART und die darauf aufbauenden Modelle dem Seq2Seq-Konzept unterliegen. Das bedeutet, dass diese darauf vorgenommen sind, zu einem gegebenen Satz einen neuen Satz zu generieren, etwa im Rahmen einer Übersetzung von einer Sprache in eine andere. In letzterem Fall kann eine Variation des Textes sogar wünschenswert sein, was aber im gegebenen Anwendungsfall eher hinderlich ist. Ein Problem ist, dass sich selbst bei gleicher Ursprungs- und Zielsprache nicht immer zweifelsfrei vorhersagen lässt, wo sich die generierten Token im Satz befinden, die dem neuen Pseudowort-Token entsprechen sollen. Da der restliche Satz und somit auch die Position des Pseudoworts nicht mehr sicher mit dem Original übereinstimmen, führt dies wiederholt zu inkorrekteten Berechnungen des aktuellen Loss-Werts, welcher vorrangig auf der Qualität des aktuellen Pseudowort-Tokens beruhen soll. Dieses Phänomen lässt sich insbesondere bei ungewöhnlicheren Konstruktionen erkennen. Es ist zu vermuten, dass mBART-50 für diese häufiger auftretende Ausdrucksweisen bevorzugt, wodurch die Inkonsistenzen zustande kommen.

Eine denkbare Abwandlung des ursprünglichen Pseudowort-Konzepts bzgl. BERT, welches hier auf mBART-50 angewendet wurde, wäre es, den Loss stattdessen über die Qualität des gesamten generierten Satzes im Vergleich zum Referenzsatz zu berechnen, anstatt nur über das Pseudowort-Token. Dieser Ansatz wird jedoch nicht weiter verfolgt, da sich bei ersten Versuchen schnell herausstellt, dass der Loss-Wert schnell sehr gering wird und nicht mehr viele Anpassungen vorgenommen werden. Das liegt allerdings nicht daran, dass das Pseudowort bereits gut angenähert wurde. Stattdessen sind alle anderen Tokens im ausgegebenen Satz bereits gleich oder sehr ähnlich, sodass die geringe Qualität eines einzelnen Tokens (hier des Pseudoworts) in einer solchen Loss-Berechnung vernachlässigbar ist. Somit hätte ein solches Training auf sehr lange Sicht möglicherweise Erfolg, würde aber unverhältnismäßig hohe zeitliche und maschinelle Ressourcen in Anspruch nehmen. Im Rahmen dieser Arbeit wird also auf den ursprünglichen Ansatz zurückgegriffen, obwohl dabei Ungenauigkeiten in Kauf genommen werden müssen.

4.3.3 Details der Implementierung und Hardware beim Training von Pseudowort-Embeddings für mBART

Für die Implementierung des Trainingsalgorithmus wurde der Quellcode von KARIDI et al. (2021) wie auch schon für das Training mit BERT als Grundlage herangezogen. Allerdings konnte nur die grundsätzliche Struktur übernommen werden, da der gesamte restliche Code an die Anforderungen von mBART-50 angepasst werden musste.

Insbesondere geht die Implementierung von mBART-50 mit einigen Fallstricken einher, die insbesondere in Form von undokumentierten Eigenheiten des Modellcodes umgangen werden müssen. So wird etwa bei der Eingabe eines Satzes stets das erste Wort der Eingabe durch ein Starttoken `<S>` ersetzt, das gemäß Dokumentation effektiv nicht (mehr) durch das Modell genutzt werden soll. Dieses Token musste jedem Satz hinzugefügt werden, damit stets der vollständige Satz und somit auch KE-lex-Tokens, die am Anfang eines Satzes stehen, zielführend erlernt werden können. Weiterhin muss nach der Vorhersage der Maskierung durch den Encoder davon ausgegangen werden, dass die Ausgabe des Decoders nicht mehr in ihren Tokens deckungsgleich ist. Zwar werden in der Implementierung Versuche unternommen, die Indizes bei einer Verschiebung durch unterschiedliche Tokenlängen anzupassen, bei einer neuen Sortierung der Tokens durch das Modell ist dieser Ansatz jedoch nicht weiter zielführend. Weiterhin muss dafür gesorgt werden, dass die neuen Pseudowort-Embeddings ausschließlich Einfluss auf das Auflösen der Maskierung im Encoder Einfluss haben, da nur der MLM-Teil in Hinblick auf Pseudowörter genutzt wird. Möglicherweise wäre ein zusätzliches Konzept für ein Training des Decoders ebenfalls nötig, um in Zukunft zu qualitativ hochwertigeren Ergebnissen zu gelangen. Hierbei wäre es vielleicht möglich, die Unvorhersehbarkeit der Ausgaben des Decoders zu beschränken.

Außerdem war die Nutzung mehrerer leistungsfähiger GPUs unerlässlich. Für das Training der mBART-Pseudowort-Embeddings wurden parallel zwei Grafikkarten des Typs *NVIDIA GeForce RTX 3090* verwendet, die dauerhaft zur Verfügung standen. Diese eigneten sich aufgrund der Größe ihres Arbeitsspeicher von 24 GB. Eine Nutzung von GPUs über Google Colab wäre dahingehend auch möglich, aber aufgrund der Trainingsdauer von ungefähr einer Woche nicht praktikabel gewesen. Es ist zu empfehlen, für weitere Untersuchungen auf Grafikkarten höherer Leistung zurückzugreifen. Vor allem aber scheint für das Training der Pseudowort-Embeddings mit BERT aber ein hoher RAM in Hinblick auf lange Beispielsätze und Maskierungen relevant zu sein.

4.4 Ergänzung von Prompting-Ansätze durch Pseudowort-Überprüfung

Nachdem in dieser Arbeit bereits einige Ansätze jeweils zur Erkennung und Generierung von Konstruktionen und deren Beispielsätzen diskutiert wurden, soll nun ein Vorschlag dargelegt werden, wie einzelne Elemente dieser Verfahren miteinander kombinierbar sind.

In Kapitel 4.2 ließ sich feststellen, dass schon generative Sprachmodelle ohne zusätzliche Erweiterungen zum Teil in der Lage sind, überraschend gute Beispiele für gegebene Konstruktionsdefinitionen zurückzugeben. Allerdings schwanken die Ausgaben in der Qualität stark, sodass sich dieses Vorgehen allein nicht für alle Konstruktionen eignet. Umgekehrt lässt sich gemäß Kapitel 3.2.3 feststellen, dass Pseudowort-Embeddings das Wesen von KE-lex-Elementen vieler Konstruktionen zufriedenstellend verinnerlichen können, während für einige Konstruktionen die Anzahl oder Qualität der Beispiele nicht ausreichend ist, um entsprechende Tokens zu trainieren.

Eine mögliche Frage ist es deshalb, ob die beiden Ansätze – Generierung durch vortrainierte Sprachmodelle und Anwendung von Pseudoword-Embeddings zur Erkennung – in Kombination ihre Schwächen gegenseitig ausgleichen. Hierfür ist es jedoch notwendig, dass die *Generierung* von Beispielsätzen zu einer gegebenen Konstruktion konzipiert wird, was in Kapitel 4.2 diskutiert wird, aber zumindest in der spezialisierten Variante für mBART im Rahmen dieser Arbeit nicht erfolgversprechend zu sein scheint. Aus diesem Grund wird im Folgenden lediglich eine konzeptuelle Möglichkeit skizziert, die Stärken beider Ansätze zu vereinen. Folgendes Schema (siehe Abbildung 4.9) liegt dem Konzept zugrunde:

1. Zunächst werden wie gehabt Beispiele auf Grundlage von Definitionen zu einer gegebenen Konstruktion (ggf. mit exemplarischen Sätzen) generiert.
2. Als nächstes wird überprüft, ob die Generierungen jeweils eines der definierten KE-lex enthalten.
3. Ist dies der Fall, so wird das KE-lex durch das dazugehörigen Pseudowort ausgetauscht. Andernfalls wird das Beispiel fallen gelassen, da es nicht die notwendige Bedigung zum Zustandekommen der Konstruktion erfüllt.
4. Schließlich wird das kontextuelle Embedding an der Stelle des KE-lex gebildet. Dieses kann mit den kontextuellen Embeddings der gegebenen Beispielsätze aus dem Konstruktikon verglichen werden.
5. Anhand von Distanzmetriken ist zu bewerten, wie ähnlich das Pseudowort den KE-lex der tatsächlichen Beispielen aus dem Konstruktikon ist. Somit kann abgeschätzt werden, ob es sich bei dem generierten Beispiel um eine valide Repräsentation der jeweiligen Konstruktion handelt.

Eine Alternativlösung zur Verwendung von mBART wäre der wiederholte Einsatz von BERT-Pseudowort-Embeddings, die gemäß Kapitel 3.2.3 erstaunliche Ergebnisse bei der Erkennung korrekter Konstruktionen in Textsammlungen erzielen. Hierbei stellt sich jedoch schnell das Problem des Hardwareaufwands ein, da sowohl für Llama 2 als auch für einen häufigen, wiederholten Einsatz von BERT entsprechende GPU-Ressourcen benötigt werden. Erste Versuche zeigen, dass solche Durchläufe sowohl GPUs mit hoher Leistung benötigen, als auch eine dauerhafte Verfügbarkeit selbiger gesichert sein muss. Bei einem Test mittels der in Kapitel 3.3 bereits genutzten Ressourcen für Llama 2 und BERT war besonders das Zeitlimit hinderlich, das (Stand Januar 2024) durch Google Colab-Instanzen auferlegt wird. Sind hingegen ausreichend leistungsstarke GPUs dauerhaft verfügbar, erscheint der oben beschriebene Versuch durchaus vielversprechend bei der Generierung neuer Beispielsätze, die so beispielsweise ein Konstruktikon durch künstlich erzeugte Texte auf zuverlässige Weise erweitern könnten.

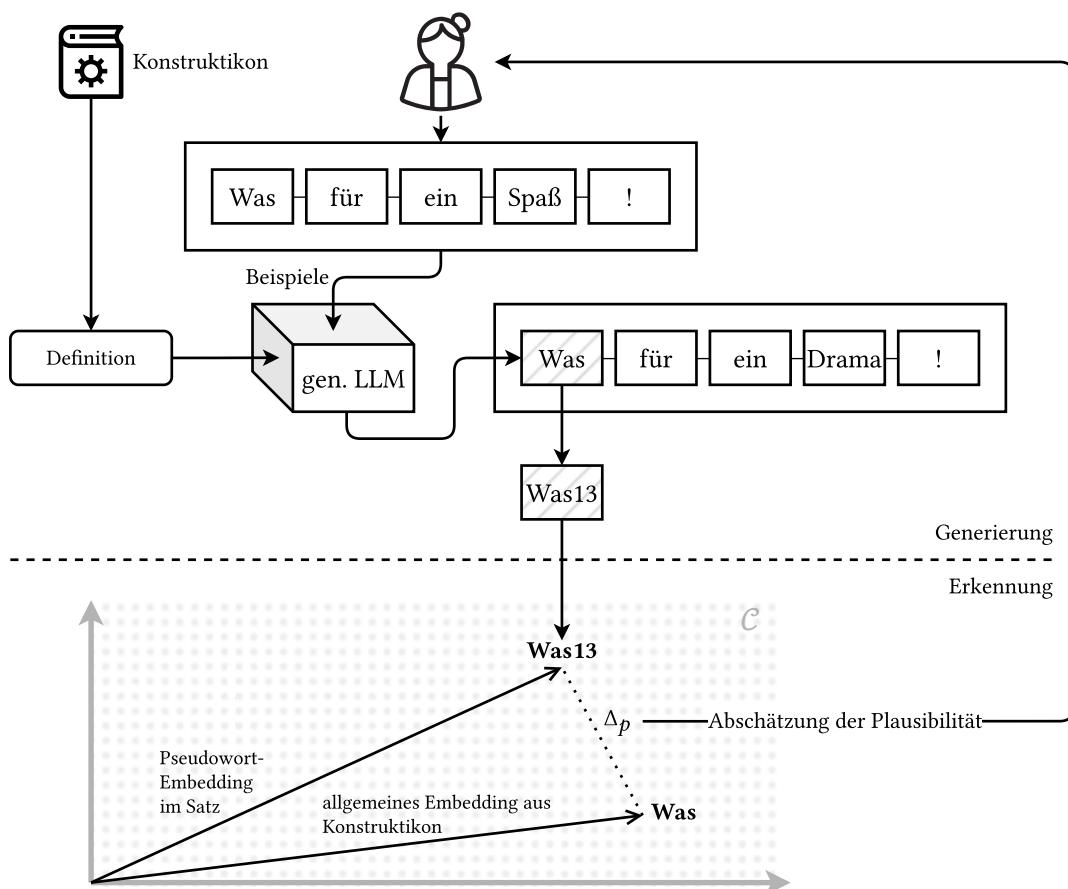


Abb. 4.9: Kombination von Verfahren zur Generierung mittels Llama 2 und Erkennung durch BERT mit Pseudowort-Embeddings

5 Zusammenfassung

Diese Arbeit hat sich der Untersuchung von zweierlei Fragestellungen angenommen: der Erkennung von Nachweisen gegebener Konstruktionen durch aktuelle Sprachmodelle auf der einen und der Generierung neuer Beispiele für gegebene Konstruktionen auf der anderen Seite.

Zunächst wurden Konstruktionsgrammatiken und Sprachmodelle im gemeinsamen Kontext vorgestellt. Dazu diente ein historischer Überblick über die Entstehung von Konstruktionsgrammatiken und deren Definition insbesondere in Hinblick auf die vormals untersuchten generativen Grammatiken (CHOMSKY, 1956) bis hin zu aktuellen Festlegungen durch Autorinnen und Autoren wie GOLDBERG (2013). Es wurde festgestellt, dass Konstruktionsgrammatiken vor allem die konzeptionelle Vereinigung verschiedener linguistischer Ebenen wie Syntax, Lexik oder Semantik anstreben, um das Verhalten von natürlicher Sprache vielseitiger und genauer beschreiben und verstehen zu können. Eine ähnliche Zielsetzung verfolgt die Informatik, im Speziellen der NLP-Bereich, bei der Entwicklung von Sprachmodellen. Aus diesem Grund wurden die Konzepte aktueller Transformer-Architekturen (VASWANI et al., 2017) aufgezeigt und kurz erläutert.

Somit ergibt sich eine Motivation für die weiteren Betrachtungen zur Erkennung und Generierung in Hinblick auf Konstruktionen und dazugehörige Textnachweise. Verschiedene Vorleistungen wurden zusammengefasst und als Inspiration für das weitere Vorgehen dargelegt. Hierbei war insbesondere das Thema der Vorfilterung, Multilingualität und Allgemeingültigkeit in Hinblick auf Konstruktionen ausschlaggebend für die weiteren Untersuchungen. Weiterhin wurde die Idee von Pseudowörtern (u. a. SCHÜTZE, 1998; KARIDI et al., 2021) vorgestellt und deren Chancen in Hinblick auf die Erweiterung von Sprachmodellen für *Natural Language Understanding* (NLU) im Allgemeinen und Konstruktionsgrammatiken im Einzelnen diskutiert. Die Anwendung entsprechend spezialisierter Embeddings stellte im Verlauf der Arbeit das exemplarisch betrachtete Vorgehen zur Erfassung von Konstruktionen dar.

Bei der Analyse der Anforderungen und dem Entwurf der entsprechenden Experimente wurde zunächst die Problemstellung der Konstruktionserkennung untersucht. Die Vorverarbeitung der Korpusdaten mithilfe von *Universal Dependencies* (DE MARNEFFE et al., 2021) stellte die Grundlage für die weiteren Implementierungen dar. Daraufhin wurde zuerst betrachtet, inwieweit vortrainierte Sprachmodelle in unveränderter Weise bereits mit der Aufgabe umgehen können, eine Konstruktion aus einer Menge von Texten zu extrahieren. Hierbei war zu erkennen, dass selbst aktuelle Modelle wie Llama 2 (TOUVRON et al., 2023b) durch Prompting nur unzureichende Ergebnisse liefern. Überraschend hingegen ist diesbezüglich die hohe Leistung von BERT-Modellen (DEVLIN et al., 2019; DBMDZ, 2023) unter Anwendung der Next-Sentence-Prediction, welche zumindest wenige zutreffende von vielen unzutreffenden potenziellen Beispielen für Konstruktionen mit einer Accuracy von teils über 70 Prozent unterscheiden konnten. Dieser Wert konnte um weitere zehn Prozentpunkte gesteigert werden, indem dem Modell Pseudowort-Embeddings hinzugefügt wurden. Es konnte gezeigt werden, dass die beobachtete Verbesserungen im Allgemeinen einem statistischen Signifikanztest standhalten. Mittels Pseudowörtern wurde weiterhin das vorgefilterte HDT-UD-Korpus (BORGES VÖLKER et al., 2019) betrachtet. Hierbei konnten zumindest einige Konstruktionen durch Ähnlichkeitsmetriken in den enthaltenen Sätzen nachgewiesen werden. Es war zu beobachten, dass sich die Ergebnisse, die mithilfe von Pseudowort-Embeddings erzielt werden können, von der jeweiligen Konstruktion selbst, aber nicht zuletzt vermutlich auch vom betrachteten Korpus abhängig sind.

5 Zusammenfassung

Analog wurde getestet, ob Sprachmodelle die Fähigkeit zur Generierung von sinnvollen und qualitativ hochwertigen Textbeispielen für gegebene Konstruktionen besitzen oder zumindest erlangen können. Dazu wurden zunächst entsprechende Anforderungen an die gewünschten Ausgaben definiert, was insbesondere die Grammatikalität und Korrektheit hinsichtlich der gegebenen Konstruktion betrifft. Daraufhin wurde mittels Zero- und Few-Shot-Learning (vgl. BROWN et al., 2020) beobachtet, wie sich Sprachmodelle wie Llama 2 bei der Generierung verhalten, je mehr Information in Form von Definition und Beispiele in den Eingabeprompts übergeben wird. Wie auch schon in den Kapiteln zur Erkennung von Beispielen war hier zu beobachten, dass sich die Leistung zwischen verschiedenen Konstruktionen immens unterscheidet – sowohl mit Blick auf maschinell messbare Metriken wie Anzahl der generierten Sätze als auch im Anteil der Wiederholungen dieser Ausgaben. Es lässt sich annehmen, dass Few-Shot-Ansätze ab drei Beispielen zu leichten Verbesserungen dahingehend führen.

Weniger Erfolg zeigte ein Versuchsaufbau, der von einer Vervollständigung von Sätzen ab dem jeweiligen lexikalisch festen Konstruktionselement (KE-lex) ausgeht. Da BART-Modelle (LEWIS et al., 2020) architektonische Bestandteile von BERT inkludieren, sollte analysiert werden, ob das Masked-Language-Modeling mit mBART-50 (TANG et al., 2021), welches die Vorhersage variabler Sequenzlängen erlaubt, zielführend ist. Dabei war zu erkennen, dass die Satzteile, die generiert werden müssen, offenbar bereits in einem unveränderten Modell zu lang sind, um diese sinnvoll vorherzusagen. Obwohl zudem gezeigt werden konnte, dass Ausgaben für BERT sinnvoll sind, wenn einzelne Konstrukte maskiert und mithilfe von ergänzten Pseudowort-Embeddings vorhergesagt werden, war hier auch eine Erweiterung von mBART nicht hilfreich. Hierbei wurde zunächst die Anpassung des Pseudowort-Konzepts auf die mBART-Architektur konzipiert und Schwierigkeiten benannt, die insbesondere mit den zusätzlichen Decoder in den BART-Modellen einhergehen. Die Probleme während des Trainings machten sich demzufolge auch in den Ausgaben für die Beispielgenerierung bemerkbar. Abschließend wurde ein Ansatz vorgestellt, mit dem die Methoden der Erkennung und Generierung zukünftig kombiniert werden könnten, sofern entsprechende Rechenressourcen zur Verfügung stehen.

5.1 Alternative Ansätze

Im zeitlichen Rahmen dieser Arbeit konnten einige Untersuchungen stichprobenweise vorgenommen werden, die jedoch keineswegs den gesamten Umfang an potenziellen Ideen zur automatischen Analyse und Generierung von und mit Konstruktionsgrammatiken ausschöpfen und diese mit den Fortschritten im Bereich aktueller Sprachmodelle zu verknüpfen. Selbstverständlich müssten die Untersuchungen aus dieser Arbeit mit weiteren (Arten von) Sprachmodellen durchgeführt werden, um ein repräsentativeres Bild von der Leistung verschiedener Systeme zu erhalten.

Insbesondere nicht untersucht wurde ein direktes Fine-Tuning von Modellen auf herkömmliche Weise, sodass eine Spezialisierung auf die Erkennung oder das Verständnis von Konstruktionen stattfindet. Hilfreich wäre dies insbesondere für die Generierung neuer Beispiele für Konstruktionen, bei denen die in dieser Arbeit untersuchten Ansätze keine zufriedenstellenden Ergebnisse lieferten. Hierbei sind einerseits der massive Ressourcenaufwand für aktuelle Sprachmodelle zu nennen, andererseits sind Daten nach Art des Konstruktions noch rar, wären für ein solches Fine-Tuning aber unerlässlich. Es ist zu testen, ob die Menge an Beispielen bspw. aus dem deutschen Konstruktikon dennoch ausreichend ist, um ein Sprachmodell in hinreichender Weise zu weiterzutrainieren.

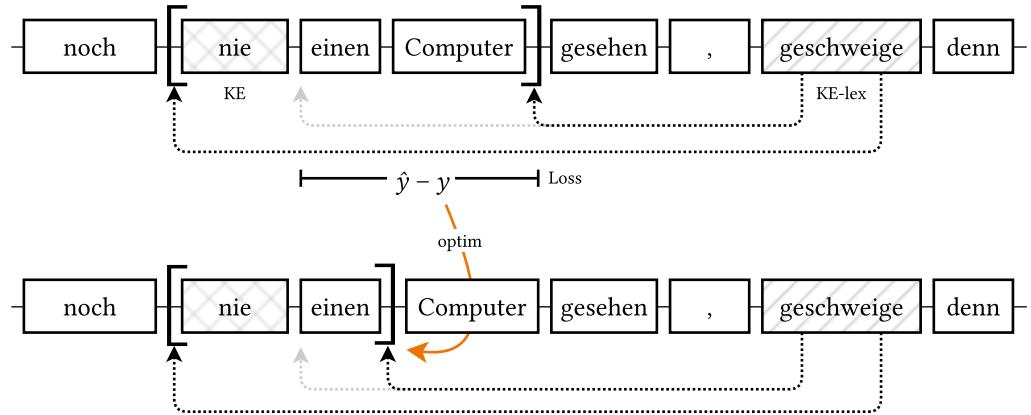


Abb. 5.1: Idee eines intervallbasierten Trainings auf Konstruktionen; der Optimizer passt beim Backtracking in Abhängigkeit des Loss-Wertes die vorhergesagten Intervallgrenzen um das Kern-KE an.

Eventuell könnte hierbei Transfer-Learning Abhilfe schaffen, indem Daten aus Konstruktika verschiedener anderer Sprachen herangezogen und fehlende Parallelen zwischen Korpora entsprechend ergänzt werden. Für solch ein sprachübergreifendes Transfer-Learning existieren bereits diverse Ansätze, die den verschiedenen Eigenheiten und unterschiedlich ausführlichen Dokumentationen verschiedener Sprache Rechnung tragen (vgl. u. a. AMMAR et al., 2016; LAMPLE et al., 2018; YIHONG LIU et al., 2023).

Für die Erweiterung von Sprachmodellen wurde in den vorliegenden Ausführungen vor allem die Implementierung von Ansätzen basierend auf Pseudowörtern beschrieben. Um hierzu nur eine Alternative exemplarisch zu erwähnen, sei die Möglichkeit der Nutzung von sogenannten Span-Klassifikationen erwähnt. In dieser Art der Klassifikation ist zu einem gegebenen Token oder einer Gruppe selbiger vorherzusagen, wo das Intervall des jeweils dazugehörigen Kontexts zu verorten ist. Dieser Ansatz wurde bereits im Kontext der *Named Entity Recognition* erfolgreich angewendet (vgl. u. a. LI et al., 2020). Auf diese Weise könnte etwa ein Token für ein Modell trainiert werden, indem die Loss-Funktion auf den Wahrscheinlichkeiten der linken und rechten Positionen der Intervallgrenzen basiert (siehe Abbildung 5.1). Dieses Prinzip ließe sich ebenfalls auf beispielsweise das Konstruktikon des Deutschen spezialisieren, indem ein derart trainiertes Modell zu gegebenen KE-lex-Tokens jeweils den Kontext bzw. die (Kern-)KEs in Form des Intervalls bestimmt wird. Dabei müsste theoretisch und praktisch überprüft werden, inwiefern sich der zurückgegebene Kontext mit den definierten KE-Elementen der Konstruktion deckt.

Einerseits könnte mit einem solchen Verfahren die Definition einer Konstruktion analog zur Nutzung von Pseudowörtern durch Sprachmodelle erfasst werden, andererseits stünden somit sogar neue Möglichkeiten zur maschinellen Konstruktionsinduktion offen: Indem ein Modell auf Grundlage aller KE-lex und deren Kontexten – ohne Trennung nach der jeweiligen Konstruktion – trainiert wird, könnte evaluiert werden, ob für *neue* potenzielle KE-lex-Kandidaten mit diesem Modell ebenfalls ein Kontext vorhergesagt werden kann. Falls sich herausstellt, dass die vorhergesagten Kontexte tatsächlich weitgehend den internen und externen KEs in einem Beispielsatz entsprechen, könnten somit automatisiert mögliche Konstruktionskandidaten vorannotiert werden. Hierfür müssten mögliche auffällige Wiederholungen in vorhergesagten Kontexten beobachtet werden, um interessante Muster in Textdaten zu erkennen. Menschliche oder maschinelle Evaluation müsste abschließend die Relevanz der potenziellen Konstruktionen bewerten.

5.2 Ausblick

Im gegebenen zeitlichen Rahmen konnte ein kleiner Teil der Möglichkeiten untersucht werden, welche Konstruktionsgrammatiken in Kombination mit den Fähigkeiten von Sprachmodellen bieten. Abschließend sollen weitere Forschungsthemen angesprochen werden, welche in Zukunft zu untersuchen sind.

Großes Optimierungspotenzial hat die Auswahl der Sprachmodelle. Während in dieser Arbeit nur kleinere und insbesondere kostenfreie und regionsunabhängige Varianten ausprobiert werden konnten (BERT, mBART, Llama 2), wäre es interessant, weitere und größere Modelle zu evaluieren. GEMINI etwa ist zum Zeitpunkt des Verfassens dieser Arbeit in der Europäischen Union nicht über die offizielle API verfügbar und somit nicht ohne Weiteres zugänglich. Ebenfalls wären aktuelle Modelle der GPT-Reihe (bspw. GPT-4) ein wichtiger Untersuchungsgegenstand, da diese auch in der breiten Öffentlichkeit häufig genutzt werden.

In Bezug auf die Erweiterung von Sprachmodellen um Pseudowort-Embeddings wäre ein Training denkbar, das die Vorteile von BERT und mBART vereint. BERT nutzt ein relativ zuverlässiges Masked-Language-Modelling, kann aber immer nur ein einzelnes Token vorhersagen. Zwar behebt mBART diesen Nachteil, arbeitet aber deutlich unzuverlässiger bezüglich des restlichen Satzes, da es sich hierbei nicht um ein reines MLM-basiertes Modell, sondern um ein Seq2Seq-Modell handelt. Außerdem ist auch die Flexibilität der Maskierungsvorhersage unter BART begrenzt. Denkbar wäre das grundlegende Aufsetzen eines neuartigen Abkömlings von BERT, welcher das Maskieren von Sequenzen variabler Länge unterstützt, während gleichermaßen der restliche Satz zuverlässig beibehalten wird. Ein solches Modell würde nicht nur für die Arbeit mit Pseudowörtern und Konstruktionsgrammatiken interessant sein, sondern das grundsätzlich interessante MLM durch die aktuelle Leistung anderer, generativer Sprachmodelle ergänzen.

Einen weiteren wichtigen Punkt stellen belastbare Fortschritte bei der maschinellen Konstruktionsinduktion dar. Die bisherigen Methoden und Ergebnisse hierzu sind momentan rar, würden bei größerem Erfolg allerdings das Problem der Datenbasis für das Training bzw. Fine-Tuning eines Sprachmodells entschärfen. Hierbei ist zum einen die Diskussion um passende Metriken zur Bewertung der aus Korpora herausgelösten Konstruktionskandidaten wichtig (siehe Kapitel 2.1.2), zum anderen ist stärker als bisher ein gemeinsames Verständnis der Elemente bzw. nötigen Konstrukte in den Konstruktionen einer maschinell erstellten Grammatik notwendig. Zudem müssen sinnvolle Heuristiken gefunden werden, welche die hohe Komplexität beim Durchsuchen von Korpora umgehen.

BENDER et al. (2021) kritisieren insgesamt eine tendenziell unverhältnismäßige Größe von Modellen bezüglich Datenbasis und Ressourcenaufwand. MAO et al. (2023) merken ebenfalls an, dass trotz hoher Investitionen eine menschenähnliche Leistung nur in wenigen Bereichen erzielt werden kann, die jedoch nur durch wenige Metriken nachgewiesen werden kann. Für eine effizientere Entwicklung von Sprachmodellen insgesamt müssen neue Ideen entwickelt werden, welche die Leistungen von LLMs gezielter optimieren. Konstruktionsgrammatiken und andere Konzepte aus der klassischen Linguistik können ein möglicher Ausblick darauf sein, wie Modellen in bestimmten Punkten spezialisiertes Wissen über die Funktionsweise natürlicher Sprache vermittelt werden kann, ohne lediglich die Datenbasis noch weiter zu erhöhen. Insbesondere für Sprachen, für die nur wenig digitale Daten verfügbar sind, ist dies in Zukunft unerlässlich.

A Ergebnisse und Rohdaten

Im Folgenden findet sich eine Auswahl der zentralen Datensätze, die für die Nutzung im Rahmen der vorliegenden Arbeit produziert worden sind.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Tab. A.1: Konstruktionen und ihr dazugehöriger *größter gemeinsamer Schnitt ggS* der Dependenztripel über die Beispiele des *FrameNet-Konstruktikon des Deutschen*

Konstr.	$1 - p_{ggS}(k)$	Größter gemeinsamer Schnitt ggS(k)
5	0.953125	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
10	0.4375	{(DET, NK, NN), (DET, NK, NOUN)}
11	0.8421052631578947	{(der, NK, NOUN), (ADP, MO, VVFIN), (NN, NK, APPR), (ADV, MO, VVFIN), (ADJA, NK, NN), (ART, NK, NN), (NOUN, NK, ADP), (ADJ, NK, NN), (DET, NK, NN), (die, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (ADV, MO, ADV), (der, NK, NN), (ADP, MO, VERB), (NOUN, NK, APPR), (DET, NK, NOUN), (die, NK, NOUN), (ADJ, NK, NOUN), (NN, NK, ADP), (ADV, MO, VERB)}
12	0.9375	{(DET, NK, NN), (DET, NK, NOUN), (ADJ, NK, NOUN), (ART, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (NOUN, SB, AUX), (., PUNCT, AUX), (\$, PUNCT, AUX), (PUNCT, PUNCT, AUX), (AUX, ROOT, AUX), (NN, SB, AUX), (ADJ, NK, NN), (ADJA, NK, NN)}
13	0.0	{(APPR, MNR, was), (für, MNR, PRON), (ADP, MNR, was), (für, MNR, was), (APPR, MNR, PRON), (ADP, MNR, PRON)}
14	0.7272727272727273	{(wie, MO, ADV), (wie, MO, ADJD), (PWAV, MO, ADV), (PWAV, MO, ADJD), (ADV, MO, ADV), (ADV, MO, ADJD)}
15	0.9333333333333333	{(\$, PUNCT, NN), (!, PUNCT, NOUN), (NOUN, SB, NN), (NOUN, SB, NOUN), (NOUN, ROOT, NN), (\$, PUNCT, NOUN), (NOUN, ROOT, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN), (NN, ROOT, NN), (welch, SB, NN), (welch, SB, NOUN), (NN, ROOT, NOUN), (!, PUNCT, NN)}
16	0.0	{(\$, PUNCT, VERB), (VERB, ROOT, VERB), (!, PUNCT, VERB), (PUNCT, PUNCT, VERB)}
19	1.0	\emptyset
20	1.0	\emptyset
21	1.0	\emptyset
22	0.0	{(ein, NK, NN), (!, PUNCT, NOUN), (PUNCT, PUNCT, NN), (NN, ROOT, NOUN), (ART, NK, NN), (ein, NK, NOUN), (ADV, MO, NN), (PUNCT, PUNCT, NOUN), (so, MO, NN), (DET, NK, NN), (\$, PUNCT, NN), (ART, NK, NOUN), (NOUN, ROOT, NN), (ADV, MO, NOUN), (so, MO, NOUN), (!, PUNCT, NN), (DET, NK, NOUN), (NOUN, ROOT, NOUN), (\$, PUNCT, NOUN), (NN, ROOT, NN)}
65	0.2142857142857143	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
69	1.0	\emptyset
74	0.5833333333333333	{(ADV, MO, ADJD), (so, MO, ADJD)}
78	0.8235294117647058	{(DET, NK, NN), (DET, NK, NOUN), (\$, PUNCT, NN), (!, PUNCT, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NOUN, ROOT, NN), (\$, PUNCT, NOUN), (NOUN, ROOT, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN), (NN, ROOT, NN), (NN, ROOT, NOUN), (!, PUNCT, NN)}
83	0.0	{(NOUN, NK, gegen), (NOUN, NK, APPR), (NN, NK, ADP), (NN, NK, gegen), (NN, NK, APPR), (NOUN, NK, ADP)}
85	0.0	{(NOUN, NK, APPR), (NN, NK, über), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (NOUN, NK, über)}
97	0.9801324503311258	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NOUN, OA, VERB), (NN, OA, VERB), (PUNCT, PUNCT, VAFIN)}
98	0.975	{(DET, NK, NN), (DET, NK, NOUN), (NN, CC, ADV), (die, NK, NOUN), (ADV, MO, ADV), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (wie, CM, NN), (so, MO, ADJD), (so, MO, ADV), (NOUN, NK, ADP), (KOKOM, CM, NN), (NN, CC, so), (ADV, MO, ADJD), (ADP, CM, NN)}
99	0.7857142857142857	{(ADJD, PD, AUX), (ADJD, PD, VAFIN), (NOUN, SB, AUX), (NOUN, SB, VAFIN), (PUNCT, PUNCT, AUX), (NN, SB, AUX), (NN, SB, VAFIN), (PUNCT, PUNCT, VAFIN), (ADV, PD, AUX), (ADV, PD, VAFIN)}

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	$1 - p_{ggS(k)}$	Größter gemeinsamer Schnitt $ggS(k)$
100	0.7272727272727273	{(DET, NK, NN), (DET, NK, NOUN), (PUNCT, PUNCT, VERB), (VERB, ROOT, VERB), (gleich, MO, ADJD), (gleich, MO, ADV), (ADV, MO, ADV), (ADV, MO, ADJD), (\$., PUNCT, VERB), (., PUNCT, VERB)}
101	0.0	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
103	0.0	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (wie, CM, NN), (wie, CM, NOUN), (KOKOM, CM, NN), (KOKOM, CM, NOUN), (ADP, CM, NN), (ADP, CM, NOUN)}
104	0.0	{(DET, NK, NN), (DET, NK, NOUN)}
111	0.9090909090909091	{(ADV, MO, ADJD), (umso, MO, ADV), (umso, MO, ADJD), (ADV, MO, ADV)}
112	0.875	{(., PUNCT, AUX), (!, PUNCT, VAFIN), (\$., PUNCT, AUX), (VAFIN, ROOT, AUX), (\$., PUNCT, VAFIN), (PUNCT, PUNCT, AUX), (VAFIN, ROOT, VAFIN), (AUX, ROOT, AUX), (AUX, ROOT, VAFIN), (PUNCT, PUNCT, VAFIN)}
122	0.9347826086956522	{(DET, NK, NN), (DET, NK, NOUN), (ADV, MO, VERB)}
123	0.41975308641975306	{(DET, NK, NN), (DET, NK, NOUN)}
125	0.0	{(ADP, CM, ADP), (PRON, SB, VAFIN), (PPER, SB, AUX), (es, SB, AUX), (wie, CM, ADP), (PPER, SB, VAFIN), (es, SB, VAFIN), (KOKOM, CM, ADP), (PRON, SB, AUX)}
127	0.0	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ein, NK, NN), (NOUN, NK, von), (ART, NK, NN), (ein, NK, NOUN), (NOUN, PD, VAFIN), (NOUN, PD, AUX), (ART, NK, NOUN), (NN, NK, APPR), (NN, NK, ADP), (NN, PD, AUX), (NN, NK, von), (NN, PD, VAFIN), (NOUN, NK, ADP)}
128	0.7857142857142857	{(ADJD, MO, ADJA), (DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ADV, MO, ADJ), (ADJ, NK, NOUN), (ART, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (NN, NK, ADP), (ADJD, MO, ADJ), (NN, NK, APPR), (NOUN, NK, ADP), (ADV, MO, ADJA), (ADJ, NK, NN), (ADJA, NK, NN)}
129	0.0	{(eine, NK, NN), (DET, NK, NN), (DET, NK, NOUN), (eine, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (DET, NK, art), (eine, NK, art), (ART, NK, art)}
130	0.0980392156862745	{(NN, NK, ADP), (NN, NK, APPR), (NN, NK, ohne)}
132	0.0	{(DET, NK, NN), (DET, NK, NOUN), (PUNCT, PUNCT, VERB), (ART, NK, NOUN), (ART, NK, NN), (PUNCT, PUNCT, VVFIN)}
133	0.8695652173913043	{(\$., PUNCT, VVFIN), (NN, SB, VERB), (\$., PUNCT, VERB), (PUNCT, PUNCT, VVFIN), (*, PUNCT, VVFIN), (NN, NK, APPR), (VERB, OC, AUX), (PRON, SB, VVFIN), (\$., PUNCT, VERB), (PRON, SB, AUX), (PPER, SB, VVFIN), (ADJA, NK, NN), (., PUNCT, VERB), (VVFIN, ROOT, VERB), (\$., PUNCT, VVFIN), (NN, SB, VVFIN), (VVPP, OC, VAFIN), (NOUN, NK, ADP), (VVFIN, ROOT, VVFIN), (ADJ, NK, NN), (\$., PUNCT, VVFIN), (DET, NK, NN), (., PUNCT, VVFIN), (ADJA, NK, NOUN), (VERB, ROOT, VERB), (VERB, OC, VAFIN), (NOUN, SB, VERB), (NOUN, NK, APPR), (DET, NK, NOUN), (PRON, SB, VAFIN), (ADJ, NK, NOUN), (PUNCT, PUNCT, VERB), (NN, NK, ADP), (*, PUNCT, VERB), (\$., PUNCT, VERB), (VVPP, OC, AUX), (VERB, ROOT, VVFIN), (PRON, SB, VERB), (PPER, SB, VERB), (NOUN, SB, VVFIN)}
135	0.4363636363636364	{(ADV, MO, ADJD), (so, MO, ADJD), (so, MO, ADV), (ADV, MO, ADV)}
136	0.0	{(PUNCT, PUNCT, AUX), (PRON, SB, AUX), (PRON, SB, VAFIN)}
139	0.16129032258064513	{(DET, NK, NN), (DET, NK, NOUN)}
317	0.7857142857142857	{(ADJ, NK, NOUN), (ADJA, NK, NOUN), (ADJ, NK, NN), (ADJA, NK, NN)}
318	0.0	{(DET, NK, NN), (DET, NK, NOUN)}
320	0.84	{(DET, NK, NN), (DET, NK, NOUN)}
349	0.0	{(NN, NK, ADP), (NOUN, NK, APPR), (NN, NK, APPR), (NOUN, NK, ADP)}
350	0.0	{(NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP)}
374	0.8636363636363636	{(\$., PUNCT, VERB), (PUNCT, PUNCT, VVFIN), (SCONJ, CP, AUX), (\$., PUNCT, VERB), (., PUNCT, VERB), (VVFIN, ROOT, VERB), (., PUNCT, VERB), (\$., PUNCT, VVFIN), (ADV, MO, AUX), (KOUS, CP, AUX), (AUX, OC, VERB), (VVFIN, ROOT, VVFIN), (\$., PUNCT, VVFIN), (., PUNCT, VVFIN), (., PUNCT, VVFIN), (VERB, ROOT, VERB), (nur, MO, AUX), (AUX, OC, VVFIN), (PUNCT, PUNCT, VERB), (weil, CP, AUX), (VERB, ROOT, VVFIN)}
379	0.0	{((., PUNCT, CARD), (\$., PUNCT, NUM), (PUNCT, PUNCT, NUM), (\$., PUNCT, CARD), (PUNCT, PUNCT, CARD), (CARD, NK, NOUN), (\$., PUNCT, NN), (PUNCT, PUNCT, NN), (NOUN, OA, VERB), (ADJA, NK, NN), (\$., PUNCT, NN), (\$., PUNCT, NOUN), (PUNCT, PUNCT, NOUN), (ADJ, NK, NN), (ADJA, NK, NOUN), (., PUNCT, NN), (NUM, NK, NN), (\$., PUNCT, NOUN), (NN, OA, VERB), (ADJ, NK, NOUN), (CARD, NK, NN), (NUM, NK, NOUN), (., PUNCT, NOUN), ((., PUNCT, NUM))}
381	0.8	{(NN, NK, ADP), (NOUN, NK, ADP)}
392	0.9302325581395349	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (NN, NK, ADP), (VAFIN, ROOT, AUX), (\$., PUNCT, AUX), (NN, NK, APPR), (\$., PUNCT, VAFIN), (VERB, OC, AUX), (VERB, OC, VAFIN), (PUNCT, PUNCT, AUX), (VAFIN, ROOT, VAFIN), (AUX, ROOT, AUX), (NOUN, NK, ADP), (AUX, ROOT, VAFIN), (PUNCT, PUNCT, VAFIN)}
488	1.0	\emptyset
492	0.9117647058823529	{(PTKA, PM, PIS), (PTKA, PM, PRON), (PRON, MO, VERB), (PIS, MO, VERB), (am, PM, PIS), (am, PM, PRON), (PART, PM, PIS), (PART, PM, PRON)}
498	0.0	{(DET, NK, NN), (DET, NK, NOUN), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (ADJ, NK, NN), (ADJA, NK, NN)}
500	0.2692307692307693	{(DET, NK, NN), (DET, NK, NOUN)}
557	0.0	{(ADJ, NK, NOUN), (ADJA, NK, NOUN), (NN, NK, ADP), (NOUN, NK, ADP), (ADJ, NK, NN), (ADJA, NK, NN)}

Tabelle wird auf der nächsten Seite fortgesetzt.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Konstr.	$1 - P_{ggS(k)}$	Größter gemeinsamer Schnitt ggS(k)
559	0.0	{(ADP, AC, ADP), (APPR, AC, vor), (ADP, AC, vor), (bis, AC, ADP), (bis, AC, vor), (APPR, AC, ADP)}
563	0.0	{(PUNCT, PUNCT, VVFIN), (PUNCT, PUNCT, VERB)}
571	0.9032258064516129	{(\$, PUNCT, AUX), (., PUNCT, AUX), (PUNCT, PUNCT, AUX), (., PUNCT, VAFIN), (\$, PUNCT, VAFIN), (PUNCT, PUNCT, VAFIN)}
572	0.0	{(PUNCT, PUNCT, NN), (., PUNCT, NN), (\$, PUNCT, NN)}
579	0.5833333333333333	{(DET, NK, NN), (DET, NK, NOUN)}
581	0.0	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN)}
584	1.0	\emptyset
590	0.0	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NOUN, NK, ADP)}
592	1.0	\emptyset
595	0.9473684210526316	{(PUNCT, PUNCT, PROPN), (NE, ROOT, NE), (DET, NK, NN), (DET, NK, NOUN), (NE, ROOT, PROPN), (PROPN, ROOT, PROPN), (., PUNCT, NE), (\$, PUNCT, NE), (PROPN, ROOT, NE), (., PUNCT, PROPN), (\$, PUNCT, PROPN), (PUNCT, PUNCT, NE)}
600	1.0	\emptyset
605	0.0	{(NN, OA, VERB), (NOUN, OA, VERB)}
618	0.5	{(DET, NK, NN), (DET, NK, NOUN)}
620	0.0	{(der, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (NN, AG, NN), (NOUN, AG, NN), (NOUN, AG, NOUN), (ART, NK, NN), (NN, AG, NOUN), (ART, NK, NOUN), (der, NK, NN)}
623	0.0	{(NN, NK, ADP), (NOUN, NK, APPR), (NN, NK, APPR), (NOUN, NK, ADP)}
626	0.0	{(ADJ, NK, NN), (ADJA, NK, NN), (ADJ, NK, NOUN), (ADJA, NK, NOUN)}
628	0.926829268292683	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ADV, MO, ADV), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (ADV, MO, ADJD), (ADJD, MO, ADV), (ADJD, MO, ADJD)}
631	0.8846153846153846	{(ADV, MO, ADJD), (PUNCT, PUNCT, AUX), (PUNCT, PUNCT, VAFIN), (ADV, MO, ADV)}
639	0.8235294117647058	{(DET, NK, NN), (ADJD, MO, VERB), (NOUN, NK, APPR), (DET, NK, NOUN), (ADJ, NK, NOUN), (ART, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (ADV, MO, VERB), (ADJ, NK, NN), (ADJA, NK, NN)}
647	0.7575757575757576	{(\$, PUNCT, AUX), (PUNCT, PUNCT, AUX), (., PUNCT, AUX)}
651	0.9230769230769231	{(APPR, MO, VERB), (NOUN, OA, VERB), (NN, OA, VERB), (mit, MO, VERB), (ADP, MO, VERB)}
653	0.9	{(ADV, MO, VVFIN), (DET, NK, NN), (DET, NK, NOUN), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (., PUNCT, NN), (ART, NK, NOUN), (PUNCT, PUNCT, VERB), (\$, PUNCT, NN), (., PUNCT, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN), (., PUNCT, NOUN), (ADV, MO, VERB), (ADJ, NK, NN), (ADJA, NK, NN)}
654	0.9230769230769231	{(DET, NK, NN), (DET, NK, NOUN), (PIAT, NK, NN), (PIAT, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, AUX), (PUNCT, PUNCT, NOUN), (PUNCT, PUNCT, VAFIN)}
663	1.0	\emptyset
671	0.0	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, AUX), (PUNCT, PUNCT, VAFIN)}
674	1.0	\emptyset
675	0.0	{(DET, NK, NN), (DET, NK, NOUN), (\$, PUNCT, VERB), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, VERB), (ADV, MO, VERB), (., PUNCT, VERB)}
676	0.0	{(ART, NK, NN)}
677	0.7446808510638299	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
681	0.6666666666666667	{(VERB, ROOT, VERB), (VERB, ROOT, VVFIN), (ADV, MO, ADV), (VVFIN, ROOT, VVFIN), (VVFIN, ROOT, VERB)}
682	0.7272727272727273	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN)}
683	0.0	{(NOUN, NK, APPR), (NN, NK, ADP), (NN, NK, mit), (NN, NK, APPR), (abstand, NK, ADP), (NOUN, NK, ADP), (abstand, NK, mit), (abstand, NK, APPR), (NOUN, NK, mit)}
696	0.9333333333333333	{(PUNCT, PUNCT, VVFIN), (ADP, MNR, NN), (NOUN, OA, VERB), (\$, PUNCT, VERB), (., PUNCT, VERB), (VVFIN, ROOT, VERB), (ART, NK, NN), (ADP, MNR, NOUN), (NOUN, NK, ADP), (NOUN, OA, VVFIN), (VVFIN, ROOT, VVFIN), (\$, PUNCT, VVFIN), (DET, NK, NN), (., PUNCT, VVFIN), (die, NK, NN), (ART, NK, NOUN), (VERB, ROOT, VERB), (NN, OA, VERB), (DET, NK, NOUN), (die, NK, NOUN), (PUNCT, PUNCT, VERB), (NN, NK, ADP), (VERB, ROOT, VVFIN), (NN, OA, VVFIN)}
697	0.0	{(KON, CD, ADV), (KON, CD, ganz), (und, CD, ADV), (CCONJ, CD, ADV), (und, CD, ganz), (CCONJ, CD, ganz)}
742	0.95	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (ADV, MO, ADV)}
756	0.0	{(DET, NK, NN), (DET, NK, NOUN), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (ADJ, NK, NN), (ADJA, NK, NN)}
757	0.0	{(NN, AG, NN), (DET, NK, NN), (DET, NK, NOUN), (NOUN, AG, NN), (NOUN, AG, NOUN), (ART, NK, NN), (NN, AG, NOUN), (ART, NK, NOUN)}

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	$1 - p_{ggS(k)}$	Größter gemeinsamer Schnitt $ggS(k)$
758	0.0	{(DET, NK, NN), (NOUN, NK, APPR), (PIAT, NK, NN), (PIAT, NK, NOUN), (DET, NK, NOUN), (NN, NK, ADP), (aller, NK, NN), (NN, NK, APPR), (aller, NK, NOUN), (NOUN, NK, ADP)}
762	0.8636363636363636	{(\$, PUNCT, VERB), (PUNCT, PUNCT, VERB), (PRON, SB, VERB), (PRON, SB, VVFIN), (., PUNCT, VERB)}
768	1.0	\emptyset
777	0.9090909090909091	{(NOUN, NK, APPR), (KON, CD, ADV), (KON, CD, ganz), (und, CD, ADV), (NN, NK, ADP), (CCONJ, CD, ADV), (NN, NK, APPR), (und, CD, ganz), (CCONJ, CD, ganz), (NOUN, NK, ADP)}
778	1.0	\emptyset
839	0.9423076923076923	{(ADV, MO, ADV)}
840	0.4444444444444444	{(DET, NK, NN), (DET, NK, NOUN), (ADV, MO, VERB)}
858	0.0	{(NOUN, NK, APPR), (NN, NK, in), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (NOUN, NK, in)}
875	0.8163265306122449	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
877	0.0	{(NUM, NK, NN), (CARD, NK, NOUN), (CARD, NK, NN), (NUM, NK, NOUN)}
882	0.0	{(PUNCT, PUNCT, AUX), (PUNCT, PUNCT, VAFIN)}
886	0.75	{(KOUS, CP, VVFIN), (DET, NK, NN), (DET, NK, NOUN), (KOUS, CP, VERB), (ADV, MO, ADV), (ART, NK, NN), (SCONJ, CP, VVFIN), (ADJA, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (ADJ, NK, NOUN), (SCONJ, CP, VERB), (NOUN, OA, VERB), (NN, OA, VERB), (ADV, MO, VERB), (ADJ, NK, NN), (ADV, MO, VVFIN)}
889	0.0	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
892	0.35	{(DET, NK, NN), (DET, NK, NOUN), (ADV, MO, VERB)}
900	0.0	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
902	0.9516129032258065	{(CCONJ, CD, NOUN), (PUNCT, PUNCT, VVFIN), (NOUN, OA, VERB), (KON, CD, NOUN), (\$, PUNCT, VERB), (VVFIN, ROOT, VERB), (NN, CJ, CCONJ), (NOUN, OA, VVFIN), (VVFIN, ROOT, VVFIN), (\$, PUNCT, VVFIN), (VERB, ROOT, VERB), (NN, CJ, sondern), (sondern, CD, NN), (CCONJ, CD, NN), (NN, OA, VERB), (PUNCT, PUNCT, VERB), (KON, CD, NN), (VERB, ROOT, VVFIN), (sondern, CD, NOUN), (NN, OA, VVFIN), (NN, CJ, KON)}
904	0.7692307692307692	{(DET, NK, NN), (DET, NK, NOUN), (KON, CD, ADJA), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (und, CD, ADJA), (CCONJ, CD, ADJA), (ADJ, NK, NN), (ADJA, NK, NN)}
905	0.94	{(DET, NK, NN), (ADP, MO, VVFIN), (NOUN, NK, APPR), (DET, NK, NOUN), (PROPN, PNC, NE), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (PROPN, PNC, PROPN), (NN, NK, APPR), (PROPN, NK, NN), (NE, PNC, NE), (APPR, MO, VVFIN), (NOUN, NK, ADP), (NE, NK, NN), (APPR, MO, VERB), (NE, PNC, PROPN), (ADP, MO, VERB)}
907	0.8095238095238095	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ADJ, NK, NOUN), (ART, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (ADJ, NK, NN), (ADJA, NK, NN)}
909	0.48	{(PUNCT, PUNCT, PROPN), (\$, PUNCT, NE), (PUNCT, PUNCT, NE), (\$, PUNCT, PROPN)}
911	0.0	{(PUNCT, PUNCT, PROPN), (), PUNCT, PROPN}, ((, PUNCT, PROPN), (NE, NK, ADP), (NE, NK, APPR), (PROPN, NK, APPR), ((, PUNCT, NE), (PROPN, NK, ADP), (\$, PUNCT, NE), (PUNCT, PUNCT, NE), (), PUNCT, NE), (\$, PUNCT, PROPN)}
917	0.8888888888888888	{(der, NK, NOUN), (), PUNCT, PROPN}, (NE, NK, NOUN), (NN, NK, APPR), ((, PUNCT, NE), (\$, PUNCT, NE), (PUNCT, PUNCT, NE), (ADJA, NK, NN), (PUNCT, PUNCT, PROPN), (\$, PUNCT, PROPN), (ART, NK, NN), (NOUN, NK, ADP), (ADJ, NK, NN), (DET, NK, NN), (PROPN, PNC, NE), (ADJA, NK, NOUN), (ART, NK, NOUN), (PROPN, PNC, PROPN), (PROPN, NK, NN), (NE, PNC, NE), (NE, NK, NN), (der, NK, NN), (NE, PNC, PROPN), (NOUN, NK, APPR), (DET, NK, NOUN), (ADJ, NK, NOUN), (PUNCT, PUNCT, VERB), (NN, NK, ADP), (PROPN, NK, NOUN), (ADV, MO, VERB), (), PUNCT, NE)}
919	0.8928571428571429	{((, PUNCT, NN), (NN, NK, APPR), (\$, PUNCT, NN), (PUNCT, PUNCT, NN), (), PUNCT, NOUN), (ADJA, NK, NN), ((, PUNCT, NOUN), (ART, NK, NN), (\$, PUNCT, NOUN), (PUNCT, PUNCT, NOUN), (NOUN, NK, ADP), (ADJ, NK, NN), (DET, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (NOUN, NK, APPR), (DET, NK, NOUN), (ADJ, NK, NOUN), (NN, NK, ADP), (, PUNCT, NN))}
921	0.8928571428571429	{(NN, NK, ADP), (\$, PUNCT, NN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN), (), PUNCT, NOUN}, (\$, PUNCT, NOUN), (), PUNCT, NN}, (NOUN, NK, ADP)}
923	0.0	{(NUM, NK, NN), (CARD, NK, NOUN), (CARD, NK, NN), (NUM, NK, NOUN)}
926	0.0	{(DET, NK, NN), (DET, NK, NOUN), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (ADJ, NK, NN), (ADJA, NK, NN)}
949	0.0	{(DET, NK, NN), (DET, NK, NOUN), (APPR, MNR, NN), (ADP, MNR, NN), (APPR, MNR, NOUN), (ADP, MNR, NOUN)}
973	0.9696969696969697	{(NN, NK, ADP), (NOUN, NK, ADP)}
976	0.0	{(\$, PUNCT, AUX), (., PUNCT, AUX), (PUNCT, PUNCT, AUX), (., PUNCT, VAFIN), (\$, PUNCT, VAFIN), (PUNCT, PUNCT, VAFIN)}
1004	1.0	\emptyset
1006	0.0	{(NOUN, NK, APPR), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (ADJ, NK, NN), (ADJA, NK, NN)}
1029	1.0	\emptyset
1033	0.9666666666666667	{(PRON, SB, VERB), (PRON, SB, VVFIN)}
1034	0.0	{(NN, NK, ADP), (NOUN, NK, APPR), (NN, NK, APPR), (NOUN, NK, ADP)}

Tabelle wird auf der nächsten Seite fortgesetzt.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Konstr.	$1 - P_{ggS(k)}$	Größter gemeinsamer Schnitt ggS(k)
1035	0.0	{(PUNCT, PUNCT, AUX)}
1043	0.4	{(NN, NK, ADP), (NOUN, NK, ADP)}
1051	0.0	{(DET, NK, NN), (DET, NK, NOUN)}
1054	0.0	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
1126	0.62	{(DET, NK, NN), (DET, NK, NOUN)}
1134	0.93181818181819	{(NOUN, SB, AUX), (\$., PUNCT, VAFIN), (PUNCT, PUNCT, AUX), (., PUNCT, VAFIN), (VAFIN, ROOT, VAFIN), (AUX, ROOT, VAFIN), (ART, NK, NN), (ADV, MO, AUX), (NN, SB, AUX), (DET, NK, NN), (die, NK, NN), (ART, NK, NOUN), (., PUNCT, AUX), (VAFIN, ROOT, AUX), (\$., PUNCT, VAFIN), (NOUN, SB, VAFIN), (AUX, ROOT, AUX), (PUNCT, PUNCT, VAFIN), (DET, NK, NOUN), (die, NK, NOUN), (NN, SB, VAFIN)}
1140	0.9387755102040817	{(CCONJ, CD, NOUN), (PUNCT, PUNCT, VVFIN), (NN, NK, APPR), (KON, CD, NOUN), (\$., PUNCT, VERB), (., PUNCT, VERB), (VVFIN, ROOT, VERB), (ADV, MO, VVFIN), (NOUN, CJ, und), (ART, NK, NN), (NN, CJ, CCONJ), (NOUN, NK, ADP), (VVFIN, ROOT, VVFIN), (\$., PUNCT, VVFIN), (DET, NK, NN), (NOUN, CJ, KON), (., PUNCT, VVFIN), (ART, NK, NOUN), (VERB, ROOT, VERB), (und, CD, NN), (CCONJ, CD, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (und, CD, NOUN), (PUNCT, PUNCT, VERB), (NN, NK, ADP), (NN, CJ, und), (NOUN, CJ, CCONJ), (VERB, ROOT, VVFIN), (KON, CD, NN), (ADV, MO, VERB), (NN, CJ, KON)}
1162	1.0	\emptyset
1219	1.0	\emptyset
1257	0.64	{(NN, NK, ADP), (NOUN, NK, ADP)}
1289	0.8125	{(\$., PUNCT, VERB), (VAINF, OC, VVINF), (VAINF, OC, lassen), (VERB, OC, AUX), (PRON, SB, AUX), (., PUNCT, VERB), (ART, NK, NN), (lassen, OC, AUX), (VVINF, OC, AUX), (sein, OC, VVINF), (sein, OC, lassen), (AUX, OC, VERB), (VERB, OC, VMFIN), (DET, NK, NN), (die, NK, NN), (ART, NK, NOUN), (PRON, SB, VMFIN), (NOUN, NK, NN), (VAINF, OC, VERB), (lassen, OC, VMFIN), (VVINF, OC, VMFIN), (DET, NK, NOUN), (die, NK, NOUN), (PUNCT, PUNCT, VERB), (AUX, OC, VVINF), (AUX, OC, lassen), (NOUN, NK, NOUN), (sein, OC, VERB)}
1291	0.6	{(wenn, CP, AUX), (\$., PUNCT, AUX), (KOUS, CP, AUX), (PUNCT, PUNCT, AUX), (AUX, ROOT, AUX), (SCONJ, CP, AUX)}
1300	0.7857142857142857	{(NOUN, SB, geht), (NN, SB, VERB), (geht, ROOT, geht), (geht, ROOT, VVFIN), (so, MO, VERB), (VVFIN, ROOT, VERB), (ADV, MO, VVFIN), (ADV, MO, geht), (NN, SB, VVFIN), (NN, SB, geht), (so, MO, geht), (so, MO, VVFIN), (VVFIN, ROOT, VVFIN), (VVFIN, ROOT, geht), (VERB, ROOT, VERB), (NOUN, SB, VERB), (geht, ROOT, VERB), (VERB, ROOT, geht), (VERB, ROOT, VVFIN), (ADV, MO, VERB), (NOUN, SB, VVFIN)}
1301	0.0	{(DET, NK, NN), (DET, NK, NOUN), (., PUNCT, NN), (\$., PUNCT, NN), (\$., PUNCT, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN), (., PUNCT, NOUN)}
1313	0.0	{(APPRART, MO, VAFIN), (PRON, SB, VAFIN), (NOUN, NK, zum), (ADP, MO, AUX), (zum, MO, AUX), (ADP, MO, VAFIN), (zum, MO, VAFIN), (NOUN, NK, ADP), (APPRART, MO, AUX), (PRON, SB, AUX), (NOUN, NK, APPRART)}
1315	0.8888888888888888	{(PRON, OA, VERB)}
1316	0.0	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (NN, NK, in), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (NOUN, NK, in)}
1320	0.0	{(DET, NK, NN), (DET, NK, NOUN)}
1322	1.0	\emptyset
1323	1.0	\emptyset
1324	0.7272727272727273	{(DET, NK, NN), (DET, NK, NOUN), (ADJ, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (ADJ, NK, NN)}
1329	0.0	{(jedermanns, AG, NOUN), (jedermanns, AG, NN), (PROPN, AG, NN), (PROPN, AG, NOUN)}
1337	1.0	\emptyset
1342	0.0	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
1346	0.0	{(NN, NK, im), (ADP, MO, VVFIN), (buche, NK, ADP), (wie, MO, VVFIN), (APPRART, MO, VERB), (PRON, SB, VVFIN), (PWAV, MO, VVFIN), (SCONJ, MO, VVFIN), (PPER, SB, VVFIN), (buche, NK, im), (ART, NK, NN), (NN, NK, APPRART), (APPRART, MO, VVFIN), (NOUN, NK, ADP), (PPER, SB, VERB), (im, MO, VERB), (DET, NK, NN), (ART, NK, NOUN), (buche, NK, APPRART), (NOUN, NK, im), (ADP, MO, VERB), (im, MO, VVFIN), (DET, NK, NOUN), (NN, NK, ADP), (wie, MO, VERB), (PRON, SB, VERB), (SCONJ, MO, VERB), (PWAV, MO, VERB), (NOUN, NK, APPRART)}
1347	0.0	{(NOUN, NK, APPR), (NOUN, NK, zu), (tode, NK, APPR), (NN, NK, ADP), (tode, NK, ADP), (NN, NK, APPR), (NN, NK, zu), (NOUN, NK, ADP), (tode, NK, zu)}
1351	1.0	\emptyset
1459	0.0	{(\$., PUNCT, PDS), (., PUNCT, PDS), (\$., PUNCT, das), (PRON, SB, VAFIN), (PUNCT, PUNCT, das), (PUNCT, PUNCT, PDS), (AUX, RC, PRON), (VAFIN, RC, PRON), (AUX, RC, das), (AUX, RC, PDS), (\$., PUNCT, PRON), (., PUNCT, PRON), (PUNCT, PUNCT, PRON), (VAFIN, RC, PDS), (PRON, SB, AUX), (VAFIN, RC, das), (., PUNCT, das)}
1461	0.0	{(DET, NK, NN), (DET, NK, NOUN), (PIAT, NK, NN), (ART, NK, NN), (ART, NK, NOUN), (NOUN, SB, AUX), (das, NK, leben), (NN, PD, AUX), (NN, PD, VAFIN), (NN, SB, AUX), (DET, NK, leben), (leben, SB, AUX), (das, NK, NN), (das, NK, NOUN), (ART, NK, leben)}
1462	0.0	{(DET, NK, NN), (DET, NK, NOUN), (PIAT, NK, NN), (PIAT, NK, NOUN), (kein, NK, NOUN), (NOUN, PD, AUX), (NOUN, PD, VAFIN), (ponyhof, PD, AUX), (ponyhof, PD, VAFIN), (NN, PD, AUX), (NN, PD, VAFIN), (DET, NK, ponyhof), (PLAT, NK, ponyhof), (kein, NK, ponyhof), (kein, NK, NN)}
1503	1.0	\emptyset

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	$1 - p_{ggS(k)}$	Größter gemeinsamer Schnitt ggS(k)
1509	0.0	{(der, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (der, NK, arzt), (DET, NK, arzt), (ART, NK, arzt), (der, NK, NN)}
1511	1.0	\emptyset
1514	0.8571428571428572	{(DET, NK, NN), (DET, NK, NOUN), (die, NK, NN), (die, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, VERB), (VERB, ROOT, VERB), (NN, OA, VERB), (\$, PUNCT, VERB), (., PUNCT, VERB)}
1517	1.0	\emptyset
1521	0.7037037037037037	{(NOUN, NK, APPR), (ADP, MO, VVFIN), (PUNCT, PUNCT, VERB), (NN, NK, ADP), (VERB, ROOT, VERB), (NN, NK, APPR), (NOUN, NK, ADP), (\$, PUNCT, VERB), (ADP, MO, VERB), (., PUNCT, VERB)}
1524	0.896551724137931	{(DET, NK, NN), (DET, NK, NOUN), (NN, NK, ADP), (NOUN, NK, ADP), (ADP, MO, VERB)}
1525	0.0	{(DET, NK, NN), (DET, NK, NOUN), (für, MNR, NOUN), (PIAT, NK, NOUN), (PIAT, NK, NN), (APPR, MNR, NN), (ADP, MNR, NN), (APPR, MNR, NOUN), (ADP, MNR, NOUN), (für, MNR, NN)}
1554	0.0	{(NN, PD, ist), (NOUN, PD, AUX), (NOUN, SB, AUX), (NOUN, PD, VAFIN), (NOUN, SB, VAFIN), (NN, PD, AUX), (NN, PD, VAFIN), (NN, SB, AUX), (NOUN, PD, ist), (NN, SB, VAFIN)}
1556	0.833333333333334	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP)}
1573	0.0	{(PRON, OA, VERB), (PRF, OA, VERB)}
1574	0.7857142857142857	{(DET, NK, NN), (NOUN, NK, APPR), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (ADV, MO, VERB), (NN, NK, APPR), (NOUN, NK, ADP), (APPR, MO, VERB), (VERB, OC, VERB), (ADP, MO, VERB)}
1582	0.0	{(DET, NK, NN), (DET, NK, NOUN), (PUNCT, PUNCT, VERB), (ART, NK, NOUN), (ART, NK, NN), (PUNCT, PUNCT, VVFIN), (VERB, ROOT, VERB), (\$, PUNCT, VERB), (., PUNCT, VERB)}
1593	0.6190476190476191	{(NE, SB, VMFIN), (PROPN, SB, AUX), (AUX, ROOT, VMFIN), (kann, ROOT, VMFIN), (VMFIN, ROOT, AUX), (PROPN, SB, kann), (NE, SB, AUX), (VMFIN, ROOT, kann), (VMFIN, ROOT, VMFIN), (kann, ROOT, AUX), (PROPN, SB, VMFIN), (AUX, ROOT, AUX), (NE, SB, kann), (AUX, ROOT, kann), (kann, ROOT, kann)}
1594	0.8695652173913043	{(DET, NK, NN), (ADP, MO, VVFIN), (ADP, MO, VERB), (DET, NK, NOUN), (ADJ, NK, NOUN), (ADJA, NK, NOUN), (., PUNCT, NN), (PUNCT, PUNCT, VERB), (NN, NK, ADP), (PUNCT, PUNCT, VVFIN), (\$, PUNCT, NN), (\$, PUNCT, NOUN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, NOUN), (., PUNCT, NOUN), (NOUN, NK, ADP), (ADJ, NK, NN), (ADJA, NK, NN)}
1597	0.88	{(NN, NK, APPR), (CARD, NK, NOUN), (\$, PUNCT, VAFIN), (PUNCT, PUNCT, AUX), (., PUNCT, VAFIN), (VAFIN, ROOT, VAFIN), (AUX, ROOT, VAFIN), (ADJA, NK, NN), (NOUN, CJ, und), (NN, CJ, CCONJ), (NOUN, NK, ADP), (ADJ, NK, NN), (NOUN, CJ, KON), (ADJA, NK, NOUN), (., PUNCT, AUX), (VAFIN, ROOT, AUX), (\$, PUNCT, AUX), (NUM, NK, NN), (AUX, ROOT, AUX), (PUNCT, PUNCT, VAFIN), (NOUN, NK, APPR), (ADJ, NK, NOUN), (CARD, NK, NN), (NN, NK, ADP), (NN, CJ, und), (NOUN, CJ, CCONJ), (NUM, NK, NOUN), (NN, CJ, KON)}
1600	0.7647058823529411	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (NOUN, NK, ADP)}
1602	0.9090909090909091	{(DET, NK, NN), (ADJD, MO, VERB), (NOUN, NK, APPR), (DET, NK, NOUN), (ADJ, NK, NOUN), (ART, NK, NN), (ADJA, NK, NOUN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPR), (NOUN, NK, ADP), (ADV, MO, VERB), (ADJ, NK, NN), (ADJA, NK, NN)}
1621	0.8360655737704918	{(DET, NK, NN)}
1624	1.0	\emptyset
1629	0.8421052631578947	{(PUNCT, ROOT, \$), (., ROOT, \$), (wegen, NK, ADP), (wegen, NK, von), (wegen, NK, APPR), (\$, ROOT, \$), (PUNCT, ROOT, PUNCT), (., ROOT, \$), (PUNCT, ROOT, \$), (\$, ROOT, \$), (\$, ROOT, PUNCT), (., ROOT, PUNCT)}
1630	0.0	{(DET, NK, NN), (trotz, NK, APPRART), (PIAT, NK, NN), (trotz, NK, zum), (PIAT, NK, NOUN), (DET, NK, NOUN), (NN, NK, ADP), (NN, NK, APPRART), (NN, NK, zum), (NOUN, NK, ADP), (trotz, NK, ADP)}
1631	0.0	{(jetzt, MO, heisst), (ADV, MO, heisst), (jetzt, MO, VERB), (jetzt, MO, VVFIN), (ADV, MO, VERB), (ADV, MO, VVFIN)}
1634	0.8636363636363636	{(der, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (PUNCT, PUNCT, AUX), (der, NK, NN), (PUNCT, PUNCT, VAFIN)}
1636	0.8125	{(DET, NK, NN), (DET, NK, NOUN)}
1637	0.9	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
1639	0.34615384615384615	{(NE, PNC, NE), (PROPN, PNC, PROPN), (NE, PNC, PROPN), (PROPN, PNC, NE)}
1641	0.40384615384615385	{(PUNCT, PUNCT, PROPN), (., PUNCT, PROPN), (., PUNCT, PROPN), (., PUNCT, NE), (\$, PUNCT, NE), (PUNCT, PUNCT, NE), (., PUNCT, NE), (\$, PUNCT, PROPN)}
1643	0.9558823529411765	{(der, NK, NOUN), (NN, NK, APPR), (ADP, MNR, NN), (APPR, MNR, NOUN), (ADJA, NK, NN), (ART, NK, NN), (ADP, MNR, NOUN), (NOUN, NK, ADP), (PROPN, NK, ADP), (ADJ, NK, NN), (DET, NK, NN), (NE, NK, ADP), (ADJA, NK, NOUN), (ART, NK, NOUN), (der, NK, NN), (PROPN, NK, APPR), (NOUN, NK, APPR), (DET, NK, NOUN), (ADJ, NK, NOUN), (NE, NK, APPR), (NN, NK, ADP), (APPR, MNR, NN)}
1645	0.7857142857142857	{(\$, PUNCT, NN), (NN, NK, APPR), (ADP, MNR, NN), (\$, PUNCT, NN), (PUNCT, PUNCT, NN), (., PUNCT, NOUN), (., PUNCT, NOUN), (ART, NK, NN), (\$, PUNCT, NOUN), (PUNCT, PUNCT, NOUN), (ADP, MNR, NOUN), (NOUN, NK, ADP), (DET, NK, NN), (ART, NK, NOUN), (ADP, MO, VERB), (NOUN, NK, APPR), (DET, NK, NOUN), (NN, NK, ADP), (., PUNCT, NN)}
1649	0.85	{(DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN), (ART, NK, NOUN), (NN, NK, ADP), (NN, NK, APPRART), (NOUN, NK, ADP), (NOUN, NK, APPRART)}
1660	0.4545454545454546	{(PUNCT, PUNCT, AUX), (PUNCT, PUNCT, VAFIN)}

Tabelle wird auf der nächsten Seite fortgesetzt.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Konstr.	$1 - P_{ggS(k)}$	Größter gemeinsamer Schnitt $ggS(k)$
1671	0.0	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
1681	0.0	{(NN, AG, NN), (NOUN, AG, mutter), (DET, NK, NN), (PIAT, NK, NN), (NOUN, AG, NN), (ART, NK, mutter), (PIAT, NK, NOUN), (ART, NK, NN), (NN, AG, NOUN), (NOUN, AG, NOUN), (ART, NK, NOUN), (DET, NK, NOUN), (aller, NK, NN), (aller, NK, NOUN), (NN, AG, mutter), (DET, NK, mutter)}
1684	0.88	{(ART, NK, NOUN), (DET, NK, NN), (DET, NK, NOUN), (ART, NK, NN)}
1690	0.7142857142857143	{(DET, NK, NN), (DET, NK, NOUN)}
1715	0.0	{(NN, PD, ist), (DET, NK, NN), (DET, NK, NOUN), (PIAT, NK, NN), (PIAT, NK, NOUN), (ART, NK, NN), (NOUN, PD, AUX), (auch, MO, NN), (NOUN, PD, VAFIN), (auch, MO, NOUN), (ADV, MO, NN), (ART, NK, NOUN), (ADV, MO, NOUN), (NN, PD, AUX), (NN, PD, VAFIN), (NOUN, PD, ist)}
1738	0.0	{(PUNCT, PUNCT, VAFIN)}
1756	0.8928571428571429	{(PRON, SB, VAFIN)}
1760	0.1351351351351351	{(so, MO, ADV), (ADV, MO, ADV)}
1762	1.0	\emptyset
1770	0.8421052631578947	{(der, NK, NOUN), (NN, NK, APPR), (ADP, MNR, NN), (PUNCT, PUNCT, NN), (PUNCT, PUNCT, AUX), (ADJA, NK, NN), (ART, NK, NN), (NN, CJ, CCONJ), (\$, PUNCT, NN), (PUNCT, PUNCT, NOUN), (ADP, MNR, NOUN), (NOUN, NK, ADP), (APPR, MO, VERB), (ADJ, NK, NN), (DET, NK, NN), (NOUN, CJ, KON), (die, NK, NN), (ADJA, NK, NOUN), (., PUNCT, NN), (ART, NK, NOUN), (\$, PUNCT, NOUN), (der, NK, NN), (PUNCT, PUNCT, VAFIN), (ADP, MO, VERB), (DET, NK, NOUN), (NOUN, NK, APPR), (die, NK, NOUN), (ADJ, NK, NOUN), (NN, NK, ADP), (NOUN, CJ, CCONJ), (., PUNCT, NOUN), (NN, CJ, KON)}
1772	0.9210526315789473	{(NOUN, NK, APPR), (NN, NK, ADP), (NN, NK, APPR), (PUNCT, PUNCT, ADP), (NOUN, NK, ADP), (PUNCT, PUNCT, APPR)}
1777	0.0	{(NN, NK, ADP), (NOUN, NK, ADP)}
1779	0.0	{(nicht, CJ, KON), (PART, CJ, KON), (PTKNEG, CJ, oder), (nicht, CJ, CCONJ), (PTKNEG, CJ, KON), (PART, CJ, CCONJ), (PTKNEG, CJ, CCONJ), (nicht, CJ, oder), (PART, CJ, oder)}
1792	0.10416666666666663	{(PRON, SB, VAFIN), (PIS, SB, AUX), (PRON, SB, ist), (PIS, SB, VAFIN), (PIS, SB, ist), (PRON, SB, AUX)}
1802	1.0	\emptyset
1831	0.6486486486486487	{(ADV, MO, ADJD), (so, MO, ADJD), (so, MO, ADV), (ADV, MO, ADV)}
1835	0.0	{(auch, MO, NN), (NOUN, PD, VAFIN), (ist, ROOT, ist), (VAFIN, ROOT, VAFIN), (NOUN, PD, ist), (AUX, ROOT, VAFIN), (VAFIN, ROOT, ist), (PIAT, NK, NN), (AUX, ROOT, ist), (ART, NK, NN), (auch, MO, NOUN), (ADV, MO, NN), (NN, PD, VAFIN), (ist, ROOT, AUX), (NN, PD, ist), (DET, NK, NN), (PIAT, NK, NOUN), (NOUN, PD, AUX), (ART, NK, NOUN), (VAFIN, ROOT, AUX), (ADV, MO, NOUN), (AUX, ROOT, AUX), (DET, NK, NOUN), (NN, PD, AUX), (ist, ROOT, VAFIN)}
1842	0.40625	{(DET, NK, NN), (DET, NK, NOUN)}
1846	0.0	{(?, PUNCT, AUX), (PWS, SB, AUX), (PRON, SB, VAFIN), (was, SB, AUX), (? , PUNCT, VAFIN), (\$, PUNCT, AUX), (PWS, SB, VAFIN), (\$, PUNCT, VAFIN), (PUNCT, PUNCT, AUX), (was, SB, VAFIN), (PUNCT, PUNCT, VAFIN), (PRON, SB, AUX)}
1849	0.833333333333334	{(\$, PUNCT, PPER), (und, CD, PRON), (PUNCT, PUNCT, PPER), (CCONJ, CD, PRON), (PRON, ROOT, PRON), (PPER, ROOT, PRON), (KON, CD, PPER), (? , PUNCT, PRON), (\$, PUNCT, PRON), (und, CD, PPER), (CCONJ, CD, PPER), (PUNCT, PUNCT, PRON), (PRON, ROOT, PPER), (PPER, ROOT, PPER), (? , PUNCT, PPER), (KON, CD, PRON)}
1881	0.0	{(DET, NK, NN), (DET, NK, NOUN), (PIAT, NK, NN), (PIAT, NK, NOUN), (kein, NK, NOUN), (kein, NK, NN)}
1884	0.0	{(NN, PD, ist), (NN, SB, ist), (NOUN, PD, AUX), (NOUN, PD, VAFIN), (NOUN, SB, AUX), (NOUN, SB, VAFIN), (NN, PD, AUX), (NN, PD, VAFIN), (NN, SB, AUX), (NOUN, PD, ist), (NN, SB, VAFIN), (NOUN, SB, ist)}
1986	0.8235294117647058	{(ADJD, PD, AUX), (NN, NK, APPR), (\$, PUNCT, VAFIN), (PUNCT, PUNCT, AUX), (., PUNCT, VAFIN), (VAFIN, ROOT, VAFIN), (AUX, ROOT, VAFIN), (ART, NK, NN), (NOUN, NK, ADP), (DET, NK, NN), (ART, NK, NOUN), (., PUNCT, AUX), (VAFIN, ROOT, AUX), (\$, PUNCT, AUX), (AUX, ROOT, AUX), (PUNCT, PUNCT, VAFIN), (ADV, PD, AUX), (NOUN, NK, APPR), (DET, NK, NOUN), (NN, NK, ADP)}
1987	0.7692307692307692	{(DET, NK, NN), (DET, NK, NOUN)}

Tab. A.2: Gefilterte Sätze aus dem HDT-UD-Korpus mit m - und q -Metrik sowie menschlicher Evaluation der Zugehörigkeit zur angegebenen Konstruktion; Vergleich der KE-lex mit Ein-Wort-Sätzen; dargestellt sind die bis zu fünf höchstbewerteten Sätze gemäß q

Konstr.	Satz	m	q	menschl. Eval.
12	Dieser hat allerdings weder die Möglichkeit , Personen zu Aussagen zu vorzuladen , noch hat er Zugang zu vertraulichen Dokumenten .	0.053571	0.025393	1
	Das gestrige Ereignis hatte weder verletzte Personen zur Folge , noch kam die Produktionsstätte von Texas Instruments durch das Feuer direkt zu Schaden .	0.053571	0.024945	1
16	Was benutzt der den für ein Handy , welche Freisprecheinrichtung hängt ihr um den Hals ?	0.750000	0.431609	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	Satz	<i>m</i>	<i>q</i>	menschl. Eval.
	Was die fertige Version der Software einmal kosten soll , verrät der Hersteller noch nicht .	0.750000	0.424061	0
	Was dafür verwendet wurde , geht aus dem Antrag nicht hervor .	0.750000	0.423668	0
	Was der Dienst danach kosten soll , weiß Viag Interkom noch nicht .	0.750000	0.423452	0
	Was Schily damit gemeint hat , bleibt freilich fraglich .	0.750000	0.423073	0
22	So sollen zum Beispiel Bestellformulare ausgefüllt werden können :	0.900000	0.470576	0
	So sollte Schülern und Studenten bei Prüfungen stets ein Internetzugang zur Verfügung stehen .	0.900000	0.469920	0
	So können die Netzbetreiber bei Fangschaltungen zur Abwehr belästigender Anrufe untereinander Daten austauschen .	0.900000	0.467985	0
	So wurden Sicherheitsmaßnahmen zum Schutz amerikanischer und israelischer Einrichtungen getroffen .	0.900000	0.467911	0
	So wird Linux längst nicht nur im Serverbereich eingesetzt .	0.900000	0.467851	0
78	Solche Finanzierungsgeschäfte , um die eigene Hardware an Carrier verkaufen zu können , sind allerdings keine Spezialität von Lucent .	0.151261	0.077641	0
	Solche Bilder sollen vom 1. Februar 2001 an der Vergangenheit angehören .	0.151261	0.077556	0
	Solche Versionen sind aus der Sicht von Microsoft Raubkopien , doch dem Software-Unternehmen fehlten bislang die rechtlichen Grundlagen , dagegen einzuschreiten , da der entsprechende Hersteller die Software legal eingekauft hat .	0.151261	0.077450	0
	Solche hätten allerdings durch den Brief erst ins Rollen gebracht werden können .	0.151261	0.077153	0
	Solche Verzögerungen habe es aber auch beim Start des digitalen Mobilfunks mit GSM-Standard vor zehn Jahren gegeben .	0.151261	0.077117	0
100	Die beiden etwa gleich großen Unternehmen versprechen sich von der Fusion die üblichen synergistischen Effekte :	0.245455	0.138126	1
	AMD produziert den Palomino wie schon den aktuellen Thunderbird in einem 0,18-Mikron-Verfahren , auch die Caches bleiben gleich groß :	0.245455	0.130163	1
	Auch die offiziellen OEM-Preise für 1000er-Stückzahlen richten sich nach diesem Rating , damit kostet der Athlon XP immer knapp weniger als der vermeintlich gleich schnelle Pentium 4 :	0.245455	0.127852	1
	Nicht alle Bewerbungen sind offensichtlich gleich ernst gemeint , wie auch der Versuch von Joerg Clausen zeigt :	0.245455	0.125576	1
	Der Verkauf US-amerikanischer US-DVDs in Deutschland verstößt gleich doppelt gegen deutsches Recht :	0.245455	0.122560	0
379	(Kabel-) Pech der Telekom	0.916667	0.666295	0
	Mode auf (Computer-) Abwegen	0.916667	0.663762	0
	Der Pauschaltarif (ohne Zeit- oder Volumenbeschränkung) sinkt ab 1. September von 49 auf 39 Mark .	0.916667	0.659016	0
	Shenwei-I - die (fast) unschlagbare Macht	0.916667	0.653832	0
	Deutsche Telekom mit (UMTS-) Schwierigkeiten	0.916667	0.649725	0
595	MSN vs. AOL :	0.035088	0.022486	0
	FPÖ vs. " nolife " :	0.035088	0.021946	0
	Ob SuSE , Red Hat , Corel oder wie sie auch immer heißen mögen :	0.035088	0.021870	0
	Rambus vs. Infineon :	0.035088	0.021834	0
	China vs. USA :	0.035088	0.021773	0
647	Ob sich Microsoft mit seinem nun vorgelegten Antrag gegenüber dem Gericht und in der Öffentlichkeit einen Gefallen getan hat , bleibt abzuwarten - aber Microsoft hat ja inzwischen eine eigene " Botschafterin " , um das Image der Firma aufzupolieren .	0.161616	0.078576	0
	Nicht nur , daß das vertraglich zugesicherte System 8 nun doch nicht " Rhapsody " wird , sondern lediglich ein aufgepepptes System 7 - vor allem ärgert man sich über die zukünftigen hohen Lizenzkosten (angeblich ein Mehrfaches einer Windows-Lizenz) , die Verzögerungen bei der MacOS-Entwicklung für CHRP-Rechner und Apples Hardware-Zertifizierung .	0.161616	0.077757	0
	Da mag Ella noch so lange behaupten " Your Daddy's rich and your Ma's good looking " - Recht hätte sie zwar , die gute Ella , wenn sie den Text von George trällert und das Spiegel-Kind meinte .	0.161616	0.077754	0
	Aber die Autoren bekommen ihr Geld erst 18 Monate später - wenn sie Glück haben " , kritisierte Crispin Evans von Universal Music Publishing .	0.161616	0.077441	0
	Schließlich sei das ein " großer Markt " - und glücklicherweise habe Alta Vista 38 Patente .	0.161616	0.077359	0
675	Diese Entwicklung habe für die Anwender gleich zwei Vorteile , meinte Gates :	0.750000	0.379592	0
	Kai-Uwe Ricke , Vorstandsvorsitzender von T-Mobile International , betonte in diesem Zusammenhang aber gleich , dass solche Dienste aus datenschutzrechtlichen Gründen nur nach Zustimmung des jeweiligen Kunden geliefert würden .	0.750000	0.379417	0
	Als Alternative empfiehlt Intel den USB , wofür den OEMs gleich vier Buchsen nahegelegt werden .	0.750000	0.379384	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Konstr.	Satz	m	q	menschl. Eval.
	Die war dann auch so wichtig , dass darüber gleich gechattet werden musste .	0.750000	0.376557	0
	Um einige " Kriminelle " ausfindig zu machen , überwachen staatliche Organisationen gleich pauschal alle Bürger , wie zuletzt beim Super Bowl .	0.750000	0.375655	0
902	Der Bundesgerichtshof hat die Registrierungspraxis der DENIC gut geheißen , die von uns nicht aus Eigeninteresse , sondern zum Nutzen der gesamten Internet-Gemeinschaft getützt wird , sagte Denic-Justiziar Stephan Welzel .	0.046083	0.023572	1
	Die Provider , so stellen die Verfasser des Entwurfs noch einmal klar , sollen zudem keine Technik nachrüsten , sondern der Polizei nur im Rahmen ihrer bereits vorhandenen Ausrüstung unter die Arme greifen müssen .	0.046083	0.023544	1
	Die Slot-Ausführung des neuen Prozessors soll daher auch nicht auf den freien Markt , sondern nur an OEMs ausgeliefert werden .	0.046083	0.023530	1
	In Notfällen ist nicht der Computer , sondern der Arzt des Vertrauens oder sogar der Notarzt gefragt .	0.046083	0.023460	1
	Das Internet wird den klassischen Handel ergänzen , aber nicht ersetzen können .	0.046083	0.023458	1
909	Bemerkenswert allerdings , dass Ex-Kumpan Paul Allen , mit dem Gates in prähistorischen Zeiten Microsoft gründete , zwischen dem 1. März und dem 5. Juni Microsoft-Aktien im Wert von 3 Milliarden US-Dollar verkaufte - und am 6. Juni laut den Unterlagen der amerikanischen Finanz- und Börsenaufsicht Securities and Exchange Commission (SEC) Papiere für weitere 138 Millionen .	0.260000	0.124632	0
	Das vom BDZV und vom Verband Bayerischer Zeitungsverleger (VBZV) ausgerichtete Forum Zeitung steht unter dem Thema " Zeitungen.com - Verlagsstrategien im Internet " .	0.260000	0.122274	0
	In neueren Publikationen und Standards , aufgelistet auf Intels Security-Page wird die PSN mit keinem Wort mehr erwähnt - etwa bei BIS (Boot Integrity Services , Abspeichern eines Public Key in einem geschützten Speicher) oder bei Intels kürzlich vorgestellter Protected Access Architecture (mit Chipkarte) .	0.260000	0.121561	0
	Wie jetzt bekannt wurde , will sich das Justizministerium (Department of Justice - DOJ) dabei möglicherweise auf die Analysen von Banken stützen .	0.260000	0.120270	0
	Hi (Name des Empfängers) !	0.260000	0.120251	0
911	HyperTransport (vormals Lightning Data Transport - LDT) ist als Punkt-zu-Punkt-Verbindung zwischen Chips konzipiert .	0.833333	0.443846	0
	Hi (Name des Empfängers) !	0.833333	0.440953	0
	Kernstück des ansonsten wenig intelligenten Roboters ist eine mit Bakterien gefüllte Brennstoffzelle (Microbial Fuel Cell - MFC) .	0.833333	0.439373	0
	Berlinweb bietet Internet-by-Call für 5 Pfennig die Minute an ; zudem hat der Provider eine Flatrate für 49 Mark monatlich (Bereitstellung 29 Mark) im Angebot - zu ihrer Nutzung muss man allerdings seinen Telefonanschluss bei der Berliner Gesellschaft Berlikomm haben .	0.833333	0.439140	0
	In neueren Publikationen und Standards , aufgelistet auf Intels Security-Page wird die PSN mit keinem Wort mehr erwähnt - etwa bei BIS (Boot Integrity Services , Abspeichern eines Public Key in einem geschützten Speicher) oder bei Intels kürzlich vorgestellter Protected Access Architecture (mit Chipkarte) .	0.833333	0.434385	0
917	Bemerkenswert allerdings , dass Ex-Kumpan Paul Allen , mit dem Gates in prähistorischen Zeiten Microsoft gründete , zwischen dem 1. März und dem 5. Juni Microsoft-Aktien im Wert von 3 Milliarden US-Dollar verkaufte - und am 6. Juni laut den Unterlagen der amerikanischen Finanz- und Börsenaufsicht Securities and Exchange Commission (SEC) Papiere für weitere 138 Millionen .	0.103943	0.051232	1
	Das vom BDZV und vom Verband Bayerischer Zeitungsverleger (VBZV) ausgerichtete Forum Zeitung steht unter dem Thema " Zeitungen.com - Verlagsstrategien im Internet " .	0.103943	0.048208	1
	Wie jetzt bekannt wurde , will sich das Justizministerium (Department of Justice - DOJ) dabei möglicherweise auf die Analysen von Banken stützen .	0.103943	0.047009	0
	Kernstück des ansonsten wenig intelligenten Roboters ist eine mit Bakterien gefüllte Brennstoffzelle (Microbial Fuel Cell - MFC) .	0.103943	0.046528	0
	Während der 6500/300 in Deutschland nur als 6000 DM teures Bundle mit einem üppigen Software-Paket und einer Avid-Cinema-Karte zu haben ist (vgl. auch Test in c't 6/97) , steht der Mac-kompatible StarMax auch einzeln in den (US-)Regalen - mit 16fach-CD-Laufwerk , ZIP-Drive , 5 PCI-Slots und 4,3-GByte-Harddisk für unter 2900 Dollar .	0.103943	0.045904	0
919	Lediglich im SGNT-Modus (Stop Grant , Auto Halt) begnügt er sich mit 0,8 Watt - doch diese Betriebsart erreicht er nur , falls das System nichts zu tun hat , also beispielsweise ohne Hintergrundaktivität auf Tastatureingaben wartet .	0.096429	0.055883	0
	Ozeanien - die neo-liberale Insel im Meer (http://www.heise.de/tp/te/1122/fhome.htm)	0.096429	0.055130	0
	Für die Windows- und die Mac-Variante der Software will MetaCreations in einigen Tagen einen kostenlosen Bugfix liefern - per Internet unter http://www.metacreations.com oder per kostenloser CD-ROM (muß angefordert werden) .	0.096429	0.054186	0
	Da haben wir voll ins Klo gegriffen " (siehe c't 12/98 , S. 32f) .	0.096429	0.054043	0
	Zum Aufpreis von 99 Dollar verwandelt sich dieser Drucker in einen 24-Bit-Scanner (360 dpi) - eine statt des Druckkopfes montierte Scaneinheit macht 's möglich .	0.096429	0.053919	0
921	Dies bestätigte die Firma Medion AG nun ausdrücklich schriftlich per Fax gegenüber c't - unter Bezugnahme auf den von ALDI vertriebenen Flachbildschirm Lifetec MD 9383 (Medion 9383) .	0.080357	0.048366	0
	Shenwei-I - die (fast) unschlagbare Macht	0.080357	0.048203	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	Satz	<i>m</i>	<i>q</i>	menschl. Eval.
	Bertelsmann ist in den USA über seine Beteiligung an Barnesandnoble.com hinter Weltmarktführer Amazon.com Zweitplazierter bei den Online-Medienshops - in Europa ist der Konzern mit Bertelsmann Online (BOL) in der gleichen Position .	0.080357	0.048102	0
	Er stellt früher getroffene Vereinbarungen auf eine vertragliche Grundlage und konkretisiert einige kritische Punkte - darunter der Zugang zum Verwaltungssystem und die Zahlungen an Network Solutions (NSI).	0.080357	0.048072	0
	Zum Aufpreis von 99 Dollar verwandelt sich dieser Drucker in einen 24-Bit-Scanner (360 dpi) - eine statt des Druckkopfes montierte Scaneinheit macht 's möglich .	0.080357	0.047966	0
976	Wo sind denn nun die deutschen Kandidaten ? , fragen sich inzwischen selbst deutsche Politikerinnen , aber vor allem auch diejenigen , die über die " I can ! " -Kampagne auf die Wahl aufmerksam geworden sind und eventuell sogar einen Kandidatenvorschlag gemacht haben .	0.666667	0.291200	0
	Wo Nachfrage ist , ist immer auch ein Angebot , sagte Schmidt laut dpa am heutigen Samstag .	0.333333	0.153045	1
	Die Traum-Maschine erschien auf Deutsch übrigens im Jahr 1985 ; welcher Schreihals , der auf der Existenz des PC vor dem IBM PC zu bestehen müssen meint , kann mit Überschriften wie " Warum ich mein Model 100 liebe " , " Wo sind die Vics vom vergangenen Jahr " , Erinnerungen eines Osborne " oder " Und noch die Kaypro-Geschichte " etwas anfangen ?	0.333333	0.151012	0
	Wo ist das Besondere , wenn alle ihre Homepage haben ?	0.333333	0.144803	0
1140	Die Software der Leitstelle ist nie ordnungsgemäß und nachvollziehbar dokumentiert worden ;	0.059311	0.030526	0
	Mach 's gut und danke für den Fisch !	0.059311	0.029885	0
	Ein c't-Leser hatte T-Online und heise online auf die immer noch bestehende Lücke aufmerksam gemacht :	0.059311	0.029851	0
	Diese ist anhand des auf der Rückseite eingeprägten Datums- und Herstellungscodes zu identifizieren :	0.059311	0.029850	0
	Celera , Hauptkonkurrent des HGP , hat sich denn auch beeilt , mit der Erklärung von Clinton und Blair voll und ganz übereinzustimmen :	0.059311	0.029846	1
1291	Wenn statt dessen die Persönlichkeit eines Menschen und der Respekt , der ihm entgegengebracht wird , gewertet würde , hätte die Liste ein anderes Aussehen .	0.333333	0.176293	1
	Wenn man AOL-Chef Steve Case , Telekom-Chef Ron Sommer und mich an einen Tisch setzen würde , hätte die Bundesregierung die Lösung in einer Stunde . "	0.333333	0.175171	1
	Wenn die beiden sich zusammentäten , hätten wir nichts anderes als ein Linux-Microsoft :	0.333333	0.174511	1
	Wenn wir einen MP3-Player verschenken sollen , dann wird der PC teurer und weniger interessant .	0.333333	0.174493	0
	Wenn der Bund nicht mit entsprechenden Maßnahmen in allen Hauptschulen mitziehe , dann hätten die Betroffenen später Nachteile bei der Lehrstellensuche .	0.333333	0.174406	0
1301	Der Liebesbrief-Generator auf CD-ROM mit den " schönsten Formulierungen für das schönste Gefühl der Welt " kostet gerade mal 19,95 Mark , das sind noch nicht einmal zwei Kinokarten oder über den Daumen gepeilt fünf Bier .	0.500000	0.240092	0
	Der Anführer des Protestes , Ex-Chefredakteur Jewgeni Kisseljow , befand sich nach Angaben seiner Mitarbeiter im entscheidenden Moment auf einer Dienstreise .	0.000000	0.000000	0
	Der Satellit , den Al Gore schon einmal vorneweg Triana taufte , soll ab dem symbolträchtigen Jahr 2000 " die Bewegungen der sich verändernden Wolken , das Wandern von Hurricanes , große Feuerbrünste auf Ölfeldern oder in Wäldern oder andere Ereignisse während dem Moment zeigen , an dem sie geschehen . "	0.000000	0.000000	0
1582	Dagegen hatte sich nun Libro gewehrt :	0.888889	0.484358	0
	Für den Letsbuyit-Chef Martin Coles jedenfalls hat sich das Börsenintermezzo gelohnt :	0.888889	0.481304	0
	Ob sich dieser Preis angesichts der Konkurrenten auf Dauer halten lässt , bleibt abzuwarten :	0.888889	0.479083	0
	Auch heute setzte sich der Aufwärtstrend fort :	0.888889	0.478257	1
	Auch auf den Straßen Berlins trug sich Merkwürdiges zu :	0.888889	0.478153	0
1597	Unter der Nummer 01805/383725 (24 Pfennig pro Minute) kann sich der interessierte Internet-Neuling ein Heftchen bestellen , das ihm Fragen der Art " Warum soll gerade ich ins Internet ? " oder " Wo kann ich das Internet ausprobieren ? " beantwortet .	0.115714	0.064218	0
	Die lange erwartete Linux-Version der Novell Directory Services (NDS) ist seit gestern zu haben :	0.115714	0.063796	0
	Mit 3D SeeThrough sollten in Ego-Shootern per Hotkey Wände transparent werden respektive die Spielerumgebung als Drahtmodell (Wireframe-Modus) erscheinen .	0.111429	0.063209	0
	Die umbenannten Produkte sind (in Klammern die alten Namen) :	0.111429	0.062610	0
	Möglich werden soll dies unter anderem durch eine neue Vertriebsstrategie im Geschäftskundenbereich und eine spätestens zum Jahresende verfügbare PC-Linie , deren Einstiegsmodell (166 MHz Pentium) nur 999 US-\$ kostet .	0.111429	0.062414	0
1641	HyperTransport (vormals Lightning Data Transport - LDT) ist als Punkt-zu-Punkt-Verbindung zwischen Chips konzipiert .	0.447115	0.242051	0
	Kernstück des ansonsten wenig intelligenten Roboters ist eine mit Bakterien gefüllte Brennstoffzelle (Microbial Fuel Cell - MFC) .	0.447115	0.241923	0
	Berlinweb bietet Internet-by-Call für 5 Pfennig die Minute an ; zudem hat der Provider eine Flatrate für 49 Mark monatlich (Bereitstellung 29 Mark) im Angebot - zu ihrer Nutzung muss man allerdings seinen Telefonanschluss bei der Berliner Gesellschaft Berlikomm haben .	0.447115	0.238893	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Konstr.	Satz	m	q	menschl. Eval.
	Hi (Name des Empfängers) !	0.447115	0.237929	0
	Während der 6500/300 in Deutschland nur als 6000 DM teures Bundle mit einem üppigen Software-Paket und einer Avid-Cinema-Karte zu haben ist (vgl. auch Test in c't 6/97), steht der Mac-kompatible StarMax auch einzeln in den (US-)Regalen - mit 16fach-CD-Laufwerk , ZIP-Drive , 5 PCI-Slots und 4,3-GByte-Harddisk für unter 2900 Dollar .	0.447115	0.237837	0
1645	Lediglich im SGNT-Modus (Stop Grant , Auto Halt) begnügt er sich mit 0,8 Watt - doch diese Betriebsart erreicht er nur , falls das System nichts zu tun hat , also beispielsweise ohne Hintergrundaktivität auf Tastatureingaben wartet .	0.191729	0.099213	0
	Fujis MX-700 für voraussichtlich 1798 DM bietet eine Auflösung von 1280x1024 Pixeln , ein 35-mm-Objektiv (KB-äquivalent) und speichert die aufgenommenen Bilder auf den briefmarkengroßen SmartMedia-Karten - wie auch der DS-7-Nachfolger DX-7 in der XVG-A-Klasse .	0.191729	0.097728	0
	Zum Aufpreis von 99 Dollar verwandelt sich dieser Drucker in einen 24-Bit-Scanner (360 dpi) - eine statt des Druckkopfes montierte Scaneinheit macht 's möglich .	0.191729	0.097694	0
	Mit einem Caffeine-Wert von 127 lag die JavaStation zwar deutlich über X-Terminals (zwischen knapp 30 und 80) - auch über dem Wert eines SPARC5-Clones (86) , aber ein 133er Pentium-PC war mit 1538 Marks dramatisch überlegen .	0.191729	0.097185	1
	Für die Windows- und die Mac-Variante der Software will MetaCreations in einigen Tagen einen kostenlosen Bugfix liefern - per Internet unter http://www.metacreations.com oder per kostenloser CD-ROM (muß angefordert werden) .	0.191729	0.096888	0
1770	Zurück , zurück , zu goldigen Zeiten , als die Tastaturen noch aus Gummi waren .	0.138158	0.083139	1
	Der übrige Vorstand sah die Strategie mit zunehmender Sorge und zog die Notbremse , als Nicklisch noch voll auf dem Expansionstrip war .	0.138158	0.079566	0
	Der verwickelte Rechtsstreit dreht sich vorwiegend um die Jahre 1991 und 1992 , als Infineon noch die Siemens-Halbleitersparte war .	0.138158	0.079465	0
	Ich war gerade einmal zwei Tage (Konzern)-Vorsitzender , als Flugzeuge mit meinen Triebwerken ein Gebäude trafen , das ich versichert habe , das ein Kommunikationsnetz hatte , was mir gehörte ; und unser Wachstum liegt für 2001 immer noch bei 11 Prozent , erzählte Jeffrey Immelt , der neue Vorsitzende von General Electric nach den Worten einer anderen Wochenschau .	0.138158	0.078847	0
	Der Umsatz war zwar mit 2,17 Milliarden Dollar um 15% niedriger als im Vorjahr , und mit 32 Millionen Dollar Verlust ist Apple noch weit entfernt von gesunden Verhältnissen , aber das Wall Street Journal wertete diese Zahlen als Anzeichen für eine Stabilisierung .	0.138158	0.077037	0
1849	Das alles in kleinen Häppchen , die schnell verwischen und in die nächste Szene überblenden .	0.156250	0.091240	0
	Wer über die Neuerscheinungen , insbesondere die deutschsprachigen , im Bilde bleiben will , der findet beim deutschen DVD-Club im Internet stets aktualisierte Listen und Rezensionen .	0.156250	0.090757	0
	Für Inhalte im Netz wollen die meisten Nutzer nichts bezahlen , und die Zeit der kostenlose Angebote geht mit den milliardenschweren Hoffnungen auf den elektronischen Markt zu Ende .	0.156250	0.090064	0
	In den Zeiten von BSE , Anabolika-Skandalen und MKS sind auch die Schweinerl nicht mehr das , was sie einmal waren .	0.156250	0.088385	0
	NetWare-Anwender , die über das LAN auch ins Internet wollen , werden das begrüßen - NetWare/IP war bislang eher eine Krücke , und zwei Protokolle auf dem Client sind auch nicht gerade der Weisheit letzter Schluß .	0.156250	0.088163	0
1986	Auswirkungen auf den deutschen Markt dürfte die EU-Entscheidung daher auch kaum haben :	0.167647	0.088038	1
	Sollte beispielsweise die Registrierung des ersten Peter Müller schiefgegangen sein , dürfte er wohl kaum mehr eine Chance haben , Peter.Müller@ePost.de zu ergattern .	0.158824	0.085383	1
	Wer jetzt erst bestellt , dürfte kaum noch eine Chance haben , vor Weihnachten eines der begehrten Stücke zu ergattern - es sei denn gegen Aufpreis .	0.158824	0.084625	1
	Wie Messungen der c't belegen , hatte die Abschaltung der Austauschpunkte kaum spürbare Auswirkungen :	0.158824	0.083132	1
	Das Aktionärstreffen ist deshalb kaum mehr als eine Pro-Forma-Veranstaltung :	0.158824	0.081143	1

Tab. A.3: Gefilterte Sätze aus dem HDT-UD-Korpus mit m- und q-Metrik sowie menschlicher Evaluation der Zugehörigkeit zur angegebenen Konstruktion; Vergleich der KE-lex mit durchschnittlichen Embeddings über Konstruktikon-Beispiele; dargestellt sind die bis zu fünf höchstbewerteten Sätze gemäß q

Konstr.	Satz	m	q	menschl. Eval.
12	Dieser hat allerdings weder die Möglichkeit , Personen zu Aussagen zu vorzuladen , noch hat er Zugang zu vertraulichen Dokumenten .	0.053571	0.025393	1
	Das gestrige Ereignis hatte weder verletzte Personen zur Folge , noch kam die Produktionsstätte von Texas Instruments durch das Feuer direkt zu Schaden .	0.053571	0.024945	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	Satz	<i>m</i>	<i>q</i>	menschl. Eval.
15	Welch ein Image-Gewinn ...	0.057143	0.398166	1
	Welch Wunder , dass die Macher nicht noch ein " a " darin platzierten , um den komischen Kringel auch noch mit aufnehmen zu können .	0.057143	0.397455	1
	Welch schönes Gefühl ist es , in Bobotemp-Shorts (Slogan : Creating desire since 1986) am Strand von Boboland zu liegen und sich nicht um Kleinigkeiten wie ausgefallene Webcams oder Stirbsereien vor eingeschalteten Webcams kümmern zu müssen .	0.047619	0.389999	1
16	Was Compaq nun ankündigt , hat eine andere Firma bereits realisiert :	0.750000	0.464550	0
	Was genau dies bedeute , werde auf der Pressekonferenz bekannt gegeben .	0.750000	0.456886	0
	Was es können sollte , wissen die Nokia-Entwickler :	0.750000	0.451272	0
	Was zusammen passt , sollte auch zusammen kommen , betonte er .	0.750000	0.431609	0
	Ohne den guten Athanasius gäbe es vielleicht nicht so beschauliche Shows voller Grooves wie den " Familientag Kryptografie " am Heinz Nixdorf Museums-Forum , das zum 4. Februar 2001 eigentlich in die Abteilung " Was wird " gehört .	0.750000	0.424061	0
22	So musste das Unternehmen im ersten Quartal diesen Jahres einen Gewinneinbruch von 41 Prozent in dieser Sparte melden .	0.900000	0.470576	0
	So können wir jungen oder unbekannten Bands zum Durchbruch verhelfen und unseren Usern musikalische Neuheiten und Spezialitäten bieten ."	0.900000	0.469920	0
	So soll der Anwender bis zu zehn verschiedene WAP-Gateways im Gerät abspeichern können , auch ermöglicht das Handy das Speichern von bis zu zehn WAP-Bookmarks .	0.900000	0.467985	0
	So wurden in Schweden und Norwegen 28 Prozent mehr Computer als im Vorjahr verkauft .	0.900000	0.467911	0
	So könnte ein Surfer namens Jan van Daalen , der am 17. März geboren und unter der Hausnummer 124 zu finden sei , etwa die Domains Janvandaalen.173.nl oder Vandaalen.124.nl registrieren .	0.900000	0.467851	0
78	Bei solch einer Weltregierung durch die Hintertür (so der Politologe Claus Leggewie in seinem Statement zur ICANN-Debatte) darf natürlich nicht der GBD fehlen .	0.151261	0.110385	0
	Solche Ergänzungen sind zum Beispiel die Programme " Rufident " für die Telefon-CD der Firma Klicktel oder " Reverz " für das Produkt D-Info .	0.151261	0.107076	0
	Wenn man es mit solch großen Distanzen zu tun hat , dann ist man normalerweise nur in der Lage , wenige Photonen einzufangen , die von einer Sendeantenne auf dem Mars gesendet werden .	0.151261	0.106989	0
	Solch triviale Funktionen wie etwa Surfen im Web und den Austausch von E-Mails beherrscht der Tablet PC natürlich auch .	0.151261	0.105929	0
	Hinter der am 16. Dezember veröffentlichten Spezifikation stehen solch bekannte Namen wie Microsoft , Intel und Compaq .	0.151261	0.102102	0
100	Denn gleich kommt der Vorstandsvorsitzende von Daimler-Chrysler .	0.218182	0.669068	0
	Wir wollten das Angebot erst mal mit unseren Kunden testen - und nicht gleich mehrere tausend User anbinden .	0.218182	0.661094	0
	Durch die Kooperation der drei Telekom-Konzerne wäre E-Plus in Deutschland jedoch gleich zweimal vertreten , da das Hutchison-Unternehmen Auditorium Investments Germany hier eine UMTS-Lizenz ersteigern soll .	0.218182	0.546745	0
	Vielelleicht findet der ein oder andere Surfer ja an einem der gedruckten Werke Gefallen und bestellt es gleich online und ohne viel Zeit zu verlieren auf der Website , auf der er sich gerade befindet - libri.de .	0.218182	0.534959	0
	Der Preis bleibt mit 75 Mark im Monat gleich .	0.218182	0.534417	1
379	Auf den nachfolgenden Plätzen rangieren Infineon , Mitsubishi (Japan) , die taiwanischen Firmen Mosel-Vitelic und Vanguard und schließlich der japanische Fujitsu-Konzern (in dieser Reihenfolge) .	0.666667	0.648410	0
	Mit Visual CASE stellt RogueWave (Halle 3 , Stand B10) ein UML-Werkzeug vor , das sich vollständig in die Oberfläche von Microsofts Visual C++ integriert .	0.666667	0.634156	0
	Kim Schmitz will den TÜV verklagen (Update)	0.833333	0.628236	0
	K-Vee & Crazee (20.00 Uhr)	0.833333	0.620987	0
	Unter Hinweis auf die millionenschwere Kampagne (und entsprechenden Schadensersatz) versuchte man , auch Revolution.com zu bekommen , mit der hübschen Argumentation , dass die Revolution auch nicht mehr das ist , was sie dureinst war .	0.666667	0.620782	0
595	Windows CE :	0.035088	0.045875	0
	Portal-Overkill :	0.035088	0.038523	0
	Uno-e/First-e :	0.035088	0.037518	0
	Boo.com pleite :	0.035088	0.035961	0
	Beim Internet-Auktionshaus Ricardo geht es tief her :	0.035088	0.035278	0
647	Wir haben weltweit Einsicht bei fast allen Computer-Herstellern und wissen , wie ihre Lagerbestände aussehen .	0.080808	0.373018	0
	Was mich aber wirklich ärgert , ist , dass es da eine ganze Menge von Firmen gab , die erst gestartet waren und nun darunter leiden , dass die Begeisterung für den Technologie-Sektor verflogen ist .	0.080808	0.371942	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.1 Daten zur Erkennung von Konstruktionen in gegebenen Texten

Konstr.	Satz	m	q	menschl. Eval.
	Deshalb muss das Haus auch immer mehr können " , meinte Bongers .	0.080808	0.366408	0
	Jürgen Rüttgers hat einfach kein Glück mit seinen Modernisierer-Auftritten - trotz Duumvirats mit Volker Rühe hatte er keine Chance gegen Angela Merkel , und nach missglückten Auftritten mit " Kindern statt Indern " und Forderungen nach einem Internet-Standort Nr. 1 für das Bundesland , das er bald regieren möchte , lässt er sich nun mit eben dieser Angela Merkel auf Plakaten für den laufenden Landtagswahlkampf zeigen .	0.161616	0.364880	0
	Wir haben noch einiges zu tun , meinte Fiorina in ihrem Ausblick auf die nächsten Monate , " aber wir haben auch alles , was notwendig ist . "	-0.080808	0.362163	0
675	Sollte die Klägerin in dem Prozeß Recht bekommen , warnt er , käme dies " einem Maulkorb fürs Internet " gleich .	0.500000	0.379592	1
	900.000 sollen es gleich sein , die sich als Autoren hergeben , um der FDP zum Bundeskanzlerposten zu verhelfen .	0.750000	0.379417	0
	Man sollte trotzdem nicht gleich " in Panik verfallen " , stellte der Geheimdienstler dann aber klar .	0.750000	0.379384	0
	Der Firmenzwanzenschluss käme gegebenenfalls einem Aufkauf von Inprise durch Corel gleich , bei dem die Inprise-Aktionäre mit Corel-Aktien ausbezahlt würden .	0.750000	0.376557	0
	Der Preis liegt nach wie vor bei 2499 Mark , ebenfalls gleich bleiben die anderen Ausstattungsmerkmale wie 64 MByte Hauptspeicher und CD-ROM-Laufwerk .	0.750000	0.375655	0
902	Das gilt nicht nur für die Telekom , sondern für die ganze Branche , sagte Dreyer der Welt am Sonntag .	0.043779	0.121561	1
	Nicht mit BellSouth , wie ursprünglich vermutet , sondern mit dem spanischen Telefonkonzern Telefonica verhandelt KPN über eine engere Zusammenarbeit .	0.043779	0.119650	1
	Google hilft aber nicht in seiner Funktion als Suchexperte , sondern liefert die Spracherkennungstechnik , wie BMW-Presse sprecher Andreas Klugescheid heise online bestätigte .	0.043779	0.081516	1
	Microsoft Deutschland tippt auf die routinemässige Signatur durch einen Proxyserver , kann aber nicht ausschließen , dass die Signatur auf das eingebaute sichere 128-bit Verschlüsselungsprogramm zurückzuführen ist .	0.043779	0.060144	0
	Die Settop-Boxen von Microsofts WebTV zeichnen , wie Telepolis gestern berichtete , nicht nur alle vom jeweiligen Kunden gesehenen TV-Sendungen , Filme , Websites und Online-Bestellungen auf , sondern versehen das Interessenprofil auch mit einer digitalen Signatur .	0.043779	0.059607	1
909	Nun fordert die Australische National Office for the Information Economy (NOIE) , dass sich dot cx einer Ver einbarung unterwirft , nach der NOIE den Registry-Betrieb kontrolliert und über eine eventuelle Neuvergabe entscheidet .	0.000000	0.439140	0
	Das brachte die niederländische Firma United Pan-European Communications (UPC) , Europas größten TV-Kabelnetzbetreiber , in die Bredouille :	0.000000	0.434385	0
	Wie die Deutsche Presseagentur (dpa) berichtet , stellte Prof. Gerd Binnig auf dem 9. Weltkongreß über Raster tunnelmikroskopie in Hamburg die neuesten Entwicklungen seiner Forschungsgruppe vor .	0.000000	0.433099	0
	Dies kann laut Sun durch Hinzufügen des Java Development Kit (JDK) geschehen oder durch Bereitstellung eines Patches , mit dem Microsoft das Nachrüsten des Java Native Interface (JNI) ermöglicht .	-0.520000	0.415722	0
	MSI (Microstar)	0.000000	0.368602	0
911	Bertelsmann Online (BOL) , die Medien-E-Commerce-Tochter der Bertelsmann AG (Gütersloh) , will künftig Musik im Internet zum Download anbieten .	-0.333333	0.443846	1
	Der Pentium-III in 0,13µm (Tualatin) soll im ersten Halbjahr 2002 noch 1,4 GHz bei 133 MHz Bustakt erreichen ... und dann aussterben .	0.333333	0.440953	0
	Die Zusammenführung des GMD-Forschungszentrums Informationstechnik mit der Fraunhofer-Gesellschaft (FhG) stößt auf heftigen Widerstand bei der GMD-Belegschaft .	0.333333	0.439373	0
	Bestandteil von " .NET " soll ein Common Language Subset (CLS) sein , der eine Zusammenarbeit von CLS-basierten Sprachen und Bibliotheken sicherstellt .	0.000000	0.432272	0
	Wesentliche Neuerung ist , dass die Bestimmungen jetzt ohne besondere Genehmigung der US-Exportbehörde Bureau of Export Administration (BXA) die Ausfuhr jeglicher Verschlüsselungs-Software auch in die Länder der Europäischen Union sowie nach Australien , Japan , Neuseeland , Tschechien , Ungarn , Norwegen , Polen und in die Schweiz erlauben .	0.333333	0.410617	0
917	Seagate Technologie liefert derzeit erste Testexemplare der Festplatte Elite-47 (ST 446452W) mit einer Kapazität von 47 GByte für 2995 Dollar an OEMs aus .	0.086022	0.054043	0
	Für die Contentprovider in China läuft nun eine Frist von 60 Tagen , innerhalb derer sie sich um eine Lizenz bei den Telekomregulierern , etwa dem Ministry of Information Industry (MII) , zu bemühen haben .	0.086022	0.053232	1
	Das hat die Boston Consulting Group (BCG) bei der Befragung von rund 10.000 Internet-Surfern in den USA und in Kanada herausgefunden .	0.086022	0.052961	1
	Wir wollen noch in diesem Jahr den neuen Staatsvertrag schließen und mit der einheitlichen Überwachung der Qualität beginnen , sagte der Leiter der bayerischen Staatskanzlei , Erwin Huber (CSU) , der Welt am Sonntag nach einem Treffen im Kanzleramt .	0.086022	0.052821	0
	Während der Käufer mit dem YP-MFN64 ein Gerät mit schickem Cyberdesign und Fernbedienung erhält , unterstützt der YP-MF das Format Solid State Floppy Disk Card (SSFDC) .	0.086022	0.052758	1
919	Veranstaltet wird die Tagung von dem an der Hochschule Leipzig angesiedelten Projekt KONUS (" KOoperative NUtzung von Datennetzen für die Bildung und berufliche Integration von blinden und Sehbehinderten Menschen ") .	0.042857	0.055883	1
	Wer bereit ist , für den gleichen Zugang 49,95 US-Dollar auszugeben , der erhält einen Pentium-III-Rechner (450 MHz) mit 10 GByte Festplatte , 128 MByte RAM , einem 17-Zöller und einem DVD-ROM-Laufwerk .	0.064286	0.055130	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstr.	Satz	<i>m</i>	<i>q</i>	menschl. Eval.
	Bei HBCI (Homebanking Computer Interface) gehöre Star Finanz zu den " Technologieführern in Deutschland " .	0.085714	0.054186	1
	(Das vollständige Urteil unter www.netlaw.de/urteile/lgd_23.htm)	0.064286	0.053919	0
	Dieses Tarifmodell soll keine Online-Zeit enthalten , sondern mit dem Tarif T-Online Eco (2,9 Pfennig/Minute) oder aber der 79-DM-Flatrate kombiniert werden können .	0.064286	0.053785	0
921	Unter dem Bundesdurchschnitt (55 Prozent) lag Sachsen bei der Ausstattung der Haushalte mit Personalcomputern .	0.026786	0.291200	0
	930.000 Haushalte (45 Prozent) besaßen im Oktober 1999 einen PC .	0.026786	0.153045	0
	Die Demonstration funktioniert nur , wenn ActiveX und JavaScript nicht abgeschaltet sind (Standardeinstellung).	0.026786	0.151012	0
	Damit nehme der Freistaat noch vor Baden-Württemberg (15,7 Prozent) Platz eins ein , sagte Schommer .	0.026786	0.144803	0
	Mit 25,40 Mark (12,99 Euro) liegt die Miete um 10 Pfennig unter dem von der Süddeutschen Zeitung vorab gemeldeten Betrag .	0.026786	0.048366	0
976	Wo sind denn nun die deutschen Kandidaten ? , fragen sich inzwischen selbst deutsche Politikerinnen , aber vor allem auch diejenigen , die über die " I can ! " -Kampagne auf die Wahl aufmerksam geworden sind und eventuell sogar einen Kandidatenvorschlag gemacht haben .	0.666667	0.026791	0
	Wo ist das Besondere , wenn alle ihre Homepage haben ?	0.333333	0.025830	0
	Die Traum-Maschine erschien auf Deutsch übrigens im Jahr 1985 ; welcher Schreihals , der auf der Existenz des PC vor dem IBM PC zu bestehen müssen meint , kann mit Überschriften wie " Warum ich mein Model 100 liebe " , " Wo sind die Vics vom vergangenen Jahr " , " Erinnerungen eines Osborne " oder " Und noch die Kaypro-Geschichte " etwas anfangen ?	0.333333	0.025232	0
	Wo Nachfrage ist , ist immer auch ein Angebot , sagte Schmidt laut dpa am heutigen Samstag .	0.333333	0.025167	1
1140	Damit pflegt RISE nach Cyrix und IDT als dritter Hersteller den unrühmlichen Brauch , statt des Core-Taktes ein irgendwie konstruiertes P-Rating anzugeben .	0.053571	0.481304	0
	Dabei betreibt die Mehrzahl der Betriebe über das Internet Informationsrecherche oder ruft Produkt- und Firmeninformationen anderer Betriebe ab .	0.053571	0.479083	0
	Unter www.bachdigital.org wird man auch die Brandenburgischen Konzerte , Sonaten und Kantaten hören und sich über Bachs Leben und Wirken informieren können .	0.057398	0.473350	0
	Offenbar strengt Adobe sich mächtig an , sein PDF-Format fit für kommende E-Books zu machen und dem kürzlich verabschiedeten Open-eBook-Standard (OEB) etwas entgegenzusetzen , hinter dem unter anderem Microsoft und der E-Book-Hersteller NuvoMedia stehen .	0.053571	0.468916	0
	Der Fehler sei durch die Bank zu vertreten und der Kunde werde deshalb keinerlei Nachteile haben .	0.057398	0.468371	0
1291	PCs sind zumindest bis 2003 gebührenbefreit , wenn über sie ARD oder ZDF empfangen werden .	0.333333	0.465935	0
	Jede Glühbirne könnte eine Adresse haben und zum Hersteller melden , wenn ihre Lebenszeit zu Ende geht .	0.333333	0.449384	0
	Wenn erst entsprechende Bandbreiten verfügbar sind und ein Musik-Streaming in guter Qualität auf das Handy möglich ist , könnten die Telekommunikationsunternehmen zu einer bedeutenden Macht in der Unterhaltungsindustrie werden .	0.333333	0.449215	0
	Louis Armstrong wäre an diesem Samstag , den 4. August , 100 Jahre alt geworden - auch wenn er sich strikt dagegen verwahrt hätte :	0.333333	0.448187	0
	Immerhin werden die 45 Mark nur dann fällig , wenn das Problem gelöst ist .	0.333333	0.446947	0
1301	Der Anführer des Protestes , Ex-Chefredakteur Jewgeni Kisseljow , befand sich nach Angaben seiner Mitarbeiter im entscheidenden Moment auf einer Dienstreise .	0.000000	0.399707	0
	Das befreiende Gefühl beim Sprung ist genau das , was wir auch mit Freenet vermitteln wollen , kommentierte Spöerr die Aktion .	0.500000	0.399612	0
	Studien haben ergeben , dass Ärzte erst nach intensivem Training das richtige Gefühl für diese immer noch neue Art des Operierens bekommen ; weniger geübte Ärzte müssen öfter die minimal invasive Methode während der OP verlassen und zur herkömmlichen offenen Operation zurückkehren , weil Komplikationen auftreten .	0.500000	0.396053	0
	Die Kriterien , die die Marktforscher dazu nennen , klingen wenig revolutionär : neuartige Angebote , ein starker Markenname und Downloads , die dem Kunden das Gefühl vermitteln , etwas in den Händen zu halten .	0.500000	0.391679	0
	Der Liebesbrief-Generator auf CD-ROM mit den " schönsten Formulierungen für das schönste Gefühl der Welt " kostet gerade mal 19,95 Mark , das sind noch nicht einmal zwei Kinokarten oder über den Daumen gepeilt fünf Bier .	0.500000	0.391443	0
1582	Bei C@llas-Call müssen sich die Kunden schriftlich anmelden , um mit der zusätzlichen Vorwahl 01098 Ferngespräche zu führen .	0.777778	0.484358	0
	Während der Server der Firma PGP (http://www.pgp.com) immerhin noch prüft , ob die Anfrage aus einer ausländischen Domain stammt (compuserve.com oder aol.com werden als US-Domains akzeptiert) , versendet das MIT , auf dessen Server (http://web.mit.edu/network/pgp.html) die Freeware-Version liegt , eine EMail mit einem Passwort , die man sich problemlos an einen Public-Mail-Server in den USA (z. B. http://www.rocketmail.com) zustellen lassen kann .	0.777778	0.478257	0
	Mehrere Unternehmen reagierten auch nur mit der Feststellung , dass sie sich längst an diese Vorschriften halten .	0.777778	0.478153	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstr.	Satz	m	q	menschl. Eval.
	Die viertägige Messe wird allerdings nicht mehr alleine von Firmen wie Sony und Panasonic beherrscht , die in den vergangenen Jahren mit CES-Premieren wie der MiniDisk (1993) oder der DVD (1996) das Interesse auf sich zogen .	0.777778	0.478046	0
	Das Gerät lässt sich über einen Compact Flash-Slot mit zusätzlicher Hardware ausrüsten .	0.777778	0.477620	0
1597	Auch Frauen hatten bei der rein männlichen Jury eine Chance : sie nominierten Meg Whitman , Carly Fiorina (HP) und Esther Dyson (Edventure).	0.094286	0.094762	0
	Bei einer ähnlichen Aktion an der Polytechnischen Hochschule in Lausanne (École Polytechnique Fédérale de Lausanne , EPFL) hatten die Interessenten mehr Glück : rund 250 Workstations plus TFT-Monitor wurden dort tatsächlich an Studenten und Mitarbeiter geliefert , und das zu den versprochenen günstigen Konditionen .	0.094286	0.063864	0
	Das Engagement (im Internet) bei RTL ist ungebrochen , sagte Hesse , dessen zu Bertelsmann gehörende Firma in fünf Jahren rund 500 Millionen Mark in das Netz investiert .	0.102857	0.000000	0
	Das Lesen wird nicht völlig überflüssig , aber mit der Vorfilterung durch die Sprachtechnologie werden wir weniger Irrelevantes lesen , sagte der wissenschaftliche Direktor des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI) .	0.102857	0.000000	0
	Das hat zur Folge , dass IT-Fachleute häufig schon mit Anfang 30 an Stress-Krankheiten wie Tinnitus (Hörsturz) leiden .	0.102857	0.000000	0
1641	In Großbritannien machen kostenlose Internet-Zugänge dem Online-Dienst America Online (AOL) zunehmend zu schaffen .	0.000000	0.146037	0
	Die Generaldirektion Wettbewerb (GD IV) der Kommission hat das US-Justizministerium und das US-Wirtschaftsministerium über die Eröffnung des Prüfverfahrens bereits informiert .	0.000000	0.145058	0
	Die Studien eins bis drei wurden mit Erlaubnis des Ausschusses in Telepolis veröffentlicht (siehe " Echelon Special ").	0.298077	0.143371	0
	Außerdem lassen sich andere Java-Standards leicht integrieren , etwa Enterprise Java Beans (EJB) oder JDBC-Datenbankanbindungen .	0.000000	0.137570	0
	Veröffentlichungen des Deutschen Patent- und Markenamts (DPMA) über Markenschutzrechte finden sich im Deutschen Elektronischen Marken-Suchsystem (DEMAS) .	0.000000	0.117271	0
1645	Beide Level sind voll assoziativ (L1-DTLB mit 32 und L2-DTLB mit 96 Einträgen) .	0.124060	0.092544	1
	Dem Sieger winkt ein Cash-Preis von 35.000 Schillingen (ca. 5000 DM) und die Wahl zwischen einem Gratisflug in die Heimat oder der Heirat mit einem/ei EU-Bürger/in via Online-Heiratsvermittlung .	0.124060	0.090243	1
	Unter dem Kürzel RWPPi (für " Rewritable Products Promotion Initiative ") haben sich zwölf internationale Unternehmen zusammengetan , um ein neues von Pioneer entwickeltes Format für wiederbeschreibbare Digital Versatile Discs (DVD) auf den Weg zu bringen .	0.124060	0.089072	0
	Unter der Nummer 01088/0191999 können sich Kunden mit Einzelplatz-Wählzugängen (Tarifmodelle knUUT , UUDial personal edition und UUDial Web edition) per Modem (V.90) oder ISDN einwählen .	0.124060	0.088857	1
	Es liest DVD-ROMs mit fünffacher DVD-Geschwindigkeit - das sind 6,75 MByte/Sekunde - und CD-ROMs mit 32X (4,8 MByte/Sekunde) .	0.078947	0.088805	1

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Tab. A.4: Grammatikalität mittels Llama 2 generierter Sätze und deren Korrektheit bzgl. der jeweils gegebenen Konstruktion nach menschlicher Evaluation; Zero-Shot-Ansatz

Konstruktion	generierter Satz	grammatisch	zutreffend
5	Rösler wollte nicht nach Berlin	1	0
	Rösler wollte nicht ins Gesundheitsministerium	1	0
10	Die meisten von ihnen haben zuvor noch nie einen Computer gesehen	1	0
	Die meisten von ihnen haben zuvor noch nie einen Computer bedient	1	0
11	Das Lied ist zu lang und wohl kaum je spielen werden.	0	1
	Das Auto fährt schlecht und wohl kaum von der Straße abweichen wird.	0	1
12	Es wird gesagt, dass jeder in den Bergen weilt	1	0
13	Was für ein großartiger Film war das?	1	1
14	Wow! Wie unglaublich war das!	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
	Oh wie schön war die Aussicht!	1	1
15	Welch artiges Geschrei machen Sie denn denn, wenn Sie in die Badewanne steigen?	1	0
16	Was für ein Stimmungswandel war das?	1	0
	Was für ein Verräter war das?	1	0
19	Dieser Stimmungsausdruck ist wirklich unverständlich!	1	0
	Dieser Lichtniveauesdruck ist wirklich verblüffend!	0	0
20	Allein	0	0
	wusste	0	0
	dass	0	0
	er	0	0
	nicht	0	0
	könnte	0	0
	den	0	0
	Hund	0	0
	erkennen	0	0
21	Ja, das stimmt!	1	0
	Oh nein, das stimmt nicht!	1	0
22	Ich habe so eine unerwartete Art von Stimulus erlebt.	1	0
	Das war so eine überraschende Art von Stimulus.	1	0
	So eine unerwartete Art von Stimulus hat mich überrascht.	1	0
	So eine unerwartete Art von Stimulus hat mich beeindruckt.	1	0
	Das war so eine erstaunliche Art von Stimulus.	1	0
	So eine unerwartete Art von Stimulus hat mich berührt.	1	0
74	Ich bin so ziemlich müde.	0	0
	Das Rampenlicht erzeugt so viel Licht.	1	1
	Sie sieht so alt aus.	1	1
78	Das war solch ein unglücklicher Tag	1	1
83	Ich bin ein schöner Mann	1	0
	Das ist ein hoher Turm	1	0
85	Eine große Menge an hohen Häusern	1	0
	Viele kleine Pflanzen in diesem Garten	1	0
	Drei große Tiere in diesem Zoo	1	0
97	Ein Mann, der höher ist als der Frau	0	0
	Ein Mann, der schneller ist als der Auto	0	0
	Ein Mann, der heller ist als die Sonne	1	0
98	Die 'äquativ' ist gleich 22	0	0
	Das Bild ist äquativ 15cm breit	0	0
99	Das Gemüse ist ungleichwertig	1	0
	Die Menschen sind ungleichartig	1	0
	Die Mobilität ist ungleichwertig	1	0
	Die Sprachen sind ungleichwertig	1	0
	Die Wetterbedingungen sind ungleichwertig	1	0
	Die Lebenserwartung ist ungleichwertig	1	0
	Die Arbeit ist ungleichwertig	1	0
100	Die beiden Pferde sind gleicher Größe.	1	1
	Die beiden Mannschaften sind gleicher Stärke.	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
101	Das Auto fährt wie eine Lokomotive	1	1
	Das Haus schaut wie ein Schloss aus	1	1
	Das Buch liest wie ein Handbuch	0	1
103	Auf dem Markt wie bei einem Verkäufer	1	0
	Ein Mann wie ein Bär	1	1
	Ein Elefant wie ein Kuhstier	0	1
104	Das Auto hat dasselbe Gewicht wie das Motorrad.	1	1
111	Die Temperatur in Deutschland steigt im Sommer im Vergleich zu den Monaten im Winter	1	0
	Die Prozentsätze der Abschläge im Sportunterricht sind höher bei den Schülern, die im Freitagskurs besuchen	0	0
	Der Vergleich zwischen den beiden aktuellen Wetter-Apps zeigt eine höhere Genauigkeit bei der Prognose von Regen und Schnee	1	0
	Die Anzahl der Ausgaben steigt im Vergleich zur Vorwoche, da die Wirtschaft wieder aufblüht.	1	0
122	Der rote Ball ist genauso groß wie der grüne Ball	1	1
	Das kleine Auto ist genauso schnell wie das große Auto	1	1
123	Die partielle Gleichheit zwischen den Rindern und den Pferden liegt darin, dass sie ähnlich schnell laufen können.	1	0
	Die partielle Gleichheit zwischen den Schmetterlingen und den Vögeln besteht darin, dass sie ähnlich geflügelt sind.	1	0
	Die partielle Gleichheit zwischen den Katzen und den Hunden besteht darin, dass sie ähnlich viele Haare haben.	1	0
	Die partielle Gleichheit zwischen den Menschen und den Affen besteht darin, dass sie ähnlich viele Finger haben.	1	0
	Die partielle Gleichheit zwischen den Elefanten und den Rindern besteht darin, dass sie ähnlich hohe Körpergrößen haben.	0	0
125	Es ist wie ein Hund, das heulend ist	0	0
	Es ist wie ein heulender Hund	1	0
	Es ist wie ein heultreibendes Hündchen	0	0
127	Ein Kind hat eine Geburtstagsparty.	0	0
	In der Woche fühlt sich ein Mann müde.	1	0
	Ein Student hat eine Prüfung.	1	0
	Eine Frau trägt ein Kleid	1	0
	In einer Stadt lebt eine Familie	1	0
128	Vergleich von 1, 2 und 3 auf einer Skala	1	0
	Vergleich von 1, 10 und 100 auf einer Skala	1	0
	Vergleich von 10, 20 und 30 auf einer Skala	1	0
129	Eine Art von Reis ist nach dem Kochen äquivalent zu Reis	1	0
	Eine Art von Schwarz-Weißen-Ziegeln ist äquivalent zu schwarzen Ziegeln	1	0
	Eine Art von Fischen ist äquivalent zu Fischen	1	0
	Eine Art von Reis ist äquivalent zu Reis	1	0
130	Ein Kind ohne Haus ist wie ein Mann ohne Fahrrad.	1	1
132	Die Blätter der Bäume gleichen den Haaren der Katze	1	1
	Die Zahl 8 gleicht der Zahl 9	1	1
133	Beziehung zwischen dem hypothetischen Vergleichsbereich und dem Vergleichswert	1	0
	Vergleich zwischen einem hypothetischen Bereich und einem Bezugswert	1	0
	Vergleich zwischen einem hypothetischen Bereich und einem Vergleichswert	1	0
	Vergleich zwischen einem hypothetischen Bereich und einem Bezugswert als ob	0	0
135	Immer mehr Menschen leben in Städten.	1	0
	Ein höherer Bildungsgrad erhöht die Einkommenschancen.	1	0
	Der Klimawandel führt zu extremeren Wetterbedingungen.	1	0
136	Eine Kuh weist eine lange Kreiß auf	0	0
139	Der Kühlschrank hatte eine äquative partielle Gleichheit ähnlich im Vergleich zu den anderen Kühlschränken im Haus.	0	0
320	Gegen die Wand zu schreiben	0	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
	statt zu schreiben	0	0
	woherum zu schreiben	0	0
349	Ich habe mich mit meiner besten Freundin nah beieintägt.	0	0
	Ich habe mich mit meiner besten Freundin beieintägt.	0	0
350	Ich bin sehr kreativ und gewissenhaft.	1	0
	Das Essen schmeckte köstlich.	1	0
379	Ich habe einen Apfel gegessen	1	0
	Das Bier ist lecker	1	0
	Der Film ist lustig	1	0
392	—	0	0
488	Er hat das meiste Gold	1	0
	Sie hat das wenigsten Zeit	0	0
	Das Auto hat das meiste Gewicht	1	0
	Der Hund hat das wenigsten Haare	1	0
492	Kinder schenken am meisten Zeit auf Musik ein.	0	0
	Am wenigsten Saft trinkt der Lehrer.	1	0
500	Die größte Stadt Europas ist Berlin.	1	0
	Die längste Flusslandschaft der Welt ist die Amazonas.	0	0
	Die schönste Käse im Weltall ist der Käse mit dem schönsten Geschmack und dem schönsten Zustande	0	0
	Das schlammste Kino der Welt ist das Kino mit den schlechtesten Filmen aller Zeiten.	1	0
	Die kreativste Person aller Zeiten war Leonardo Da Vinci.	1	0
	Die intelligenteste Person der Welt ist Stephen Hawking.	1	0
	Die größte Stadt der Welt ist Tokio.	1	0
557	The last hour of the day is approaching	0	0
	In the last hour of the day	0	0
	In the next two hours	0	0
	In the past three weeks	0	0
	In the next two weeks	0	0
559	Bis vor morgen	0	0
	bis vor 10 Minuten	1	1
	bis vor 15 Monaten	1	1
579	Das Auto fuhr auf der Straße	1	0
	Deine Schwester heißt Maria	1	0
581	Ich habe mich in ein Restaurant getaucht, das in einem Park am Stadtrand ist.	0	0
	Er hat eine Katze namens Kater über die Straße weggetragen.	0	0
584	Ich habe gestern ein neues Buch gekauft.	1	0
	Das Buch ist von einem bekannten Autor.	1	0
	Ich habe ein Geschenk für meine Schwester gekauft.	1	0
590	Ich bin ein echter Cyber-Security-Experte.	1	0
	Das Wasser ist wunderschön.	1	0
	Die Katze schläft	1	0
592	Klammern können verwendet werden, um auf die Art und Weise zu indizieren, wie ein Sprecher in einer Rede ihre Gedanken präsentiert	1	0
595	Ich bin ein Student	1	0
	Du bist ein Lehrer	1	0
	Mama kocht gerne	1	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
600	Ich bin überrascht ihre Entscheidung	1 0	0 0
605	Es gab eine Zeit, wo ich in einer Band sang.	1	1
618	The fastest car in the world The tallest building in the city	0 0	0 0
647	Wolfgang 'Ich-will-den-Erfolg-sofort' Amadeus Mozart Oliver 'Ich-will-den-Erfolg-sofort' Kahn Marie 'Ich-will-den-Erfolg-sofort' Curie	1 1 1	1 1 1
651	Elatur-Mord Intensiviert-Konstituente Mords-Intensiviert-Konstituente Intensivierter-Konstituente	0 0 0 0	0 0 0 0
654	Riesen-Nagel Riesen-Schrank	1 1	1 1
671	The apple is as delicious as the pear John's car is faster than Mary's The bird is more colorful than the flower	0 0 0	0 0 0
674	weil der Grund für die Trennung war weil er seine Arbeit verloren hatte weil die Küche nach dem Brand neu gemalt werden musste weil sie sich nicht einig waren, wann sie es tun sollen weil der Reis versalzen war	0 0 1 1 1	1 1 1 1 1
675	Zwei Äpfel sind äquivalent Dieser Apfel und der andere sind äquivalent	1 1	0 0
676	Die Farbe des Himmels ist äquivalent mit dem der Erde, da sie beide eine rote Farbe aufweisen. Der Blumenstand ist äquivalent mit dem des Baums, da sie beide grün sind. Das Auto hat die gleiche Leistung wie das Motorrad Das neue Handy hat die gleiche Größe wie das alte Handy Die Musikband hat die gleiche Größe wie die Fußballmannschaft	1 1 1 1 1	0 0 0 0 0
681	weilX weilY weilZ	0 0 0	0 0 0
683	Ich bin sehr hungrig. Das Auto fährt schnell. Das Haus ist sehr alt.	1 1 1	0 0 0
697	Ich bin ganz und gar froh Das Geräusch war ganz und gar leise Das Essen schmeckte ganz und gar gut	1 1 1	0 0 0
758	The intensification of all nominal phrases The intensification of some nominal phrases The intensification of all pronouns	0 0 0	0 0 0
762	—	0	0
777	Kreativität und Gewissen Ganz und Gar Gradierte Dimension KEE	0 0 0 0	0 0 0 0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
	Entität	0	0
858	KEE	0	0
	Quantifizierung	0	0
	Zeiteinheit	0	0
	Annäherung	0	0
	Intensivierung	0	0
875	Die blonde Frau trägt heute Grau	1	0
	Grau ist das neue Blond	1	1
	Sie hat ihr Haar neu gestrichen	0	0
	Die neue Farbe ist grau	1	0
882	Die rote Farbe des Balles gegenübergestellt mit der roten Farbe des Himmels	0	0
	Der schwarze Hund gegenübergestellt mit dem schwarzen Schokoladeneis	1	0
	Der große Fels gegenübergestellt mit der kleinen Blume	1	0
892	This text contains information in 'Klammer' that is not part of the main topic.	0	0
900	Ich bin Mitglied der CDU	1	0
	Er ist Mitglied der SPD	1	0
902	Die Kaffirspalte ist schön, NICHTSTENDIG	0	0
	Die Kaffirspalte ist schön	1	0
	Die Kaffirspalte ist unschön, NICHTSTENDIG	0	0
904	Es ist kalt und nicht heiß.	1	1
	Es ist hart und nicht weich.	1	1
	Es ist schnell und nicht langsam.	1	1
905	Johannes Brahms wurde am 3. Mai 1833 in Hamburg geboren und verstarb am 3. April 1897.	1	0
907	Ich bin ein Assistent	1	0
	Dieser Assistent ist sehr nützlich	1	0
909	The iPhone is a product from Apple.	0	0
	The LG TV is a high-quality video display.	0	0
	The Mercedes is a luxury vehicle from Germany.	0	0
917	Die Umgebung des Tors war am Nachmittag noch immer düster und unrühmlich.	1	0
919	The abbreviation 'NHL' stands for the National Hockey League.	0	0
	The abbreviation 'IRL' stands for the IndyCar Series.	0	0
	The abbreviation 'NBA' stands for the National Basketball Association.	0	0
921	Ich liebe Sie	1	0
	Du bist der beste Mann auf der Welt	1	0
949	Keine Fische ohne Kugeln	1	1
	Kein Schmetterling ohne Flügel	1	1
	Kein Computer ohne Fernseher	1	1
	Kein Kater ohne Flauschigkeit	1	1
973	Xoder	0	0
	Y	0	0
976	Wo das Licht ist, ist auch die Dunkelheit	1	1
	Wo der Regen ist, ist auch der Regenbogen	1	1
	Wo die Freude ist, ist auch der Schmerz	1	1
	Wo die Arbeit ist, ist auch der Erfolg	1	1
1004	Michael is the tallest	0	0
	John has the shortest	0	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
	This book is the thickest	0	0
1029	Mehr Artikel anzeigen	1	0
	Kaufen	1	0
	Buch kaufen	0	0
1033	Wir müssen das Auto reparieren	1	0
	Das Auto hat eine Reparatur benötigt	1	0
1034	Er erzählte seine Geschichten wiederholt.	1	0
	Ich habe den Weg mehrmals zurückgelegt.	1	0
	Sie haben sich mit diesem Problem bereits in der Vergangenheit auseinandergesetzt.	1	0
1035	Jane is a doctor	0	0
	He is a teacher	0	0
	She is a student	0	0
	They are both doctors	0	0
	He and she are both teachers	0	0
1126	Jugendliche Generation X	1	1
	Generation Y	1	1
	Generation Z	1	1
1134	Dieses Auto fährt sowohl schnell als auch sicher	1	1
1140	Wir sind hier	1	0
	Der Hund bellt	1	0
	Die Katze schläft	1	0
	Der Ball fällt in den Garten	1	0
1162	X1 oder nicht X1	0	0
	A oder nicht A	0	0
	B oder nicht B	0	0
1289	Ich bin ein Mann	1	0
	Dieser Tisch ist rot	1	0
1291	Ich möchte, dass sie etwas Kontrafaktisches tun	1	0
	Ich wünsche, dass sie etwas Kontrafaktisches machen	1	0
1300	Intensivierte Kategorie	0	0
	so geht NP	0	0
1301	Der Moment, als ich das erste Mal in der Wüste war	1	1
	Der Moment, als ich das erste Mal im Krankenhaus war	1	1
	Der Moment, als ich das erste Mal im Flugzeug war	1	1
	Der Moment, als ich das erste Mal im Meer war	1	1
	Der Moment, als ich das erste Mal in einer Stadt war	1	1
	Der Moment, als ich das erste Mal in einem Theater war	1	1
	Der Moment, als ich das erste Mal in einer Bar war	1	1
	Der Moment, als ich das erste Mal in einem Museum war	1	1
	Der Moment, als ich das erste Mal einen Schreibtisch hatte	1	1
1313	Es ist zum Putzieren Zeit!	0	0
1315	Ich bin sehr wütend.	1	0
	Das macht mich sehr frustriert.	0	0
1316	Das Auto fährt schnell durch den Nebelfilm	0	0
	Die Katze jagt den Vogel	1	0
	Der Mann trinkt den heißen Kaffee	1	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
1323	Ich habe mich intensiv am Schreiben der Arbeit gearbeitet	0	0
	Das Auto fährt sehr intensiv durch die Stadt	1	0
	Sie hat sich intensiv für das Kunstgeschäft interessiert	1	0
1324	intensivierte	0	0
	satt	0	0
1329	Jedermanns-Merkmal	0	0
	Jedermanns-Eigenschaft	0	0
	Jedermanns-Fähigkeit	0	0
	Jedermanns-Kenntnis	0	0
	Jedermanns-Erfahrung	0	0
	Jedermanns-Empfindung	0	0
1342	The prototypical excellence of the restaurant's service is a source of pride for the owner.	0	0
1346	Prototyp-Buchesteh	0	0
1347	Ich werde etwas intensivieren, um es zu einem Todesfaktor zu machen	1	0
	Das Verbrechen ist so intensiv geworden, dass es zum Tode geführt hat	1	0
	Die Diskussion wurde so intensiv, dass sie zu Tode führte	0	0
1351	Dieser Konstrukt ist identisch mit diesem Konstrukt.	0	0
1459	Ich muss meine Arbeit halten	1	0
	Ich habe eine neue Arbeit gefunden	1	0
	Ich habe eine neue Beziehung gefunden	1	0
	Ich habe eine neue Kürzung gemacht	1	0
1461	Das Leben ist kein Kohlsuß	0	1
	Das Leben ist kein Schmetterling	1	1
	Das Leben ist kein Reptil	1	1
	Das Leben ist kein Pferd	1	1
1462	Das Buch ist kein Ponyhof.	1	1
1503	Intensiviertes	0	0
	brutal	0	0
	Gerümpelt	0	0
	schwer	0	0
1509	Der Fußballtrainer intensiviert den Ball bis der Torhüter kommt.	0	0
	Die Musiklehrerin intensiviert das Lied bis der Schüler die Töne richtig richtet.	1	0
	Der Koch intensiviert das Gericht bis es genug isst.	1	0
	Der Leselehrer intensiviert das Lesen bis das Kind lesen kann.	1	0
1511	The cat is on the mat.	0	0
1525	Kein Essen gibt es für niemand.	1	1
	Kein Wasser trinken solltest du für niemand.	1	0
1554	Redes ist Silber Xist Gold	0	0
1573	Sie bewegt sich selbst durch das Dunkel.	1	1
1574	Der junge Mann schaut sich in seinem Spiegel über den Kauf eines neuen Autos nach.	0	1
	Der Mann klopft sich mit der Hand auf den Bauch.	1	1
	Sie beobachtet sich selbst im Spiegel.	1	1
	Sie schaut sich ihre neue Bluse ein weiteres Mal an.	1	1
1582	Ich gehe immer wieder den gleichen Weg zum Arbeitsplatz zurück	1	0
	Das Auto bewegte sich zufällig durch die Straßen	1	0
	Sie selbst erkannte, dass der Fehler ihr selbst war	0	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
	Das Kind streichelte sich selbst mit dem Ohrstäppchen	0	0
	Ich selbst habe mich soeben in meinem Bett verschlafen	0	0
1593	Ich kann einen Apfel essen.	1	0
1597	Johannes ist ein Mann.	1	0
1602	Der Geldbeutel war voller Gold	1	0
	Ein Mann namens Peter ging in die Stadt	1	0
1629	Es ist interessant zu erfahren, dass Sie eine Affinität zu den Farben Grau und Gelb haben.	1	0
	Von wegen, dass ich gerne Kaffirkräuter trinke.	1	1
	Von wegen, dass ich heute zum ersten Mal eine Gitarre spielen werde.	1	1
1630	Ich war gegenüber allen positiv, aber schaffte es trotzdem nicht, den Job zu bekommen.	1	0
1631	Bitte geh einfach und zufällig eine Zahl von 1 bis 10 aus.	0	0
	Nimm dir bitte das Glas mit der Wasserhydrate und trink es.	0	0
1637	Bei der Einordnung von "Katzen" in die Gattung "Säugetiere" wird "Katzen" als Gattung subsummiert.	0	0
	In der Einordnung von "Katzen" in die Gattung "Säugetiere" wird "Katzen" als Gattung subsummiert.	0	0
1641	The painting was created by Pablo Picasso.	0	0
	The poem was written by William Shakespeare.	0	0
1643	Die Wetterbedingungen sind entsprechend meiner Erwartung.	1	0
1649	Ein leckeres Essen	1	0
	Käse-Ice-Cream	0	0
	Leckeres Obstsalat	0	1
1660	Ich bin ein Mann	1	0
	Das Auto ist rot	1	0
	Es regnet	1	0
	Das Haus ist alt	1	0
1671	Ich habe das Konzept der modernen Stadt als Prototyp im Sinn.	1	0
1681	Das Schmetterling als Prototyp aller Arten	1	0
	Der Löwe als Prototyp aller Katzen	1	0
1690	Du bist nicht unkreativ	1	1
	Du bist kein gewissenhafter Mensch	1	0
1715	Kein N1 ist auch ein N1	0	0
1738	Es ist unglaublich wie du es gemacht hast.	1	0
1756	Ich habe gehabt	1	0
	Du hättest gehabt	1	1
	Er hatte gehabt	1	0
1760	So-so-so	0	0
	Groß-groß-klein	0	0
	Schwarz-weiß	1	0
	Hoch-hoch-niedrig	0	0
1762	Besser halten als gehen	1	1
	Besser spät als nie	1	1
	Besser zu früh als zu spät	1	1
1770	Auch als ob	0	0
	Es ist noch so	1	0
	Ging's noch	0	0
	Etwas mehr	1	0
	Sag's noch einmal	1	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
1772	Wir haben gestern einen schönen Tag gehabt	1	0
	Ich bin gestern leicht gekleidet gegangen	1	0
	Er hat gestern einen Job gehabt	1	0
1777	In dem Maß des Gleichen	0	0
	Im selben Maß	1	1
	In dem gleichen Maß	1	1
	Im gleichen Maß	1	1
	In einem gleichen Maß	1	1
	In gleichem Maß	1	1
1779	Du bist entweder jung oder alt	1	0
	Er ist entweder gut oder schlecht	1	0
	Es ist entweder kalt oder warm	1	0
1792	Ich bin niemand perfekt, aber ich bin immer noch wertvoll	0	0
	Keiner ist unersetbar, aber jeder hat seine Stärken	1	1
	Wir alle haben unsere Schwächen und Fehler	1	0
1831	So hörte ich nicht jemals zu.	0	0
	So gerne würde ich es tun.	1	0
	So wollte ich es niemals tun.	1	0
	So tief war ich vor der Show verzaubert.	1	0
1835	Jeder, der schwimmen kann, kann auch schwimmen gehen.	1	0
1846	Was ist mit deinem Computer?	1	1
	Was ist mit dem Regen?	1	1
	Was ist mit der Woche?	1	1
	Was ist mit deinem Job?	1	1
	Was ist mit deinem Auto?	1	1
1849	Und wie kann es nur möglich sein?	1	0
	Und da sagst du mir jetzt erst, dass du schwul bist?	1	0
1881	Die Axt ist kein Klinge	0	0
1884	Der Kaffee schmeckt intensiver als der Tee	1	0
	Die Börse war intensiver gestrühmt als der Theaterplatz	0	0
1986	Dimension	0	0
	kaum	0	0
	ADJ	0	0
1987	Das Auto fährt sehr schnell	1	0
	Die Musik ist sehr laut	1	0
	Das Essen schmeckt sehr gut	1	0

Tab. A.5: Grammatikalität mittels Llama 2 generierter Sätze und deren Korrektheit bzgl. der jeweils gegebenen Konstruktion nach menschlicher Evaluation; 3-Shot-Ansatz

Konstruktion	generierter Satz	grammatisch	zutreffend
5	Rösler wollte nicht nach Berlin ins Gesundheitsministerium	1	0
	Und schon gar nicht nach mehr davon	0	1
	Ich bin gerne ein Mann der Tat, aber nicht als solcher aufgewiesen	0	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
10	Die meisten von ihnen haben zuvor noch nie einen Computer bedient	1	0
	Dennoch hat Hamas es nicht geschafft, ihre militärischen Kapazitäten auszubauen, geschweige denn eine zivile Infrastruktur in ihrem Einflussgebiet zu etablieren	1	1
	Reicher, geschweige denn glücklicher, wird ein Volk dadurch nicht – ebenso wenig, wie es durch die Heirat des Geistlichen ärmer wird	1	1
11	Es ist wohl kaum wahrscheinlich , dass der Kaffee in diesem Glas heiß genug ist	1	0
	Es ist wohl kaum wahrscheinlich , dass der Film am Wochenende vor der Arbeit beginnt	1	0
	Es ist wohl kaum wahrscheinlich , dass sie in diesem Monat heiratet	1	0
	Es ist wohl kaum wahrscheinlich , dass er das Examen besteht	1	0
	Es ist wohl kaum wahrscheinlich , dass sie ihn verliebt hat	0	0
12	Das Kühlschrank hat weder Elektrizität noch Gas benötigt	1	1
	Der Kunde hat weder eine Waffe noch einen Schutzschild bei sich	1	1
	Die Regierung hat weder eine neue Steuersatz-Reform noch eine neue Gesundheitsreform verabschiedet	1	1
13	Was für ein unglaublicher Auftritt!	1	1
	Was für ein beeindruckender Leistung!	0	1
	Was für ein aufregender Erfahrung!	0	1
	Was für ein spannender Situation!	0	1
	Was für ein interessanter Idee!	0	1
	Was für ein großartiger Idee!	0	1
	Was für ein beeindruckender Idee!	0	1
14	Wie beeindruckend!	1	1
	Wie unglaublich!	1	1
	Wie erstaunlich!	1	1
15	Welch ein Reinfall !	1	1
	Welch ein Krach in der Oper	1	1
	Welch ein Blödsinn !	1	1
16	Was für eine Blume ist das?	1	0
	Was man ihr auch antworten würde	1	0
	Was für ein Tag es heute ist	1	1
19	Wow, diese Sicht auf den Himmel ist unglaublich schön!	1	0
	Dieser entsetzliche Geruch!	1	1
	Diese absurdnen Gesetze!	1	1
20	Allein diese Überraschung !	1	1
	Allein wie der junge Lehrer diesen Klassenraum aufgerissen hat	1	1
	Allein wie der neue Album vom Popstar in den Charts stand	0	1
21	Dass er die Ecke umgekehrt hat, obwohl er die falsche Richtung eingeschlagen hatte	0	0
	Dass sie schließlich eine gute Entscheidung getroffen hat	1	1
	Dass er plötzlich aufgehört hat zu lachen	1	1
	Dass sie so begeistert war, dass sie aufblätte	1	1
	Dass er wirklich so aufgeregzt war, dass er sich verplappte	1	1
22	So ein abgefahrener Konzept !	1	1
	So ein bequemer Platz im Wohnzimmer	1	1
	So ein großartiger Film	1	1
74	Das Essen schmeckte so schlecht, dass es kaum zu einem schnellen Übel wurde.	1	1
	Das Auto fuhr so schnell, dass es in der Kurve sehr nahe an den Baum geriet.	1	1
	Er wusste so gut, dass er jede Frage richtig beantworten konnte.	1	1
78	Solche Ungeschicklichkeit !	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
	Solche Unverschämtheit !	1	1
	Solche Verschwendung !	1	1
	Solche Unwahrheit !	1	1
	Solche Unsinn !	0	1
	Solche Langweile !	1	1
	Solche Einfaltspflege !	1	1
	Solche Dummheit !	1	1
	Solche Eitelkeit !	1	1
83	Weil der Wind gegen die Windsperrre anschreibt	0	0
	Liebe gegen Eifersucht	1	1
	Geld gegen Freiheit	1	1
85	Menschen im Internet werfen sich immer wieder Schattenwünsche	0	0
	Zwei Schiffe waren im Hafen von Nantucket festgemacht	1	0
97	Habt ihr eine Chance gehabt, in einem warmen Lager zu schlafen?	1	0
	Der Tisch ist heller als ein neuer Himmel.	1	0
	Der Wanderer ist frischer als ein Frühlingsgarten.	1	0
98	Ich habe zuvor nur so geschaltet	1	0
	Ich lese nicht mehr so viel wie früher	1	1
	Das Leben ist hier sozusagen wunderbar	1	0
	Der Anruf war so unerfreulich wie erwartet	1	0
99	Der Apple-iPhone ist im Vergleich zum Samsung-Galaxy-S-II schneller.	1	1
	Der neue Porsche-911-GT3-R-Sportwagen ist schneller als der Alfa-Romeo-Giulia-Quadrifoglio.	1	1
	Die Kosten für eine Dental-Behandlung sind im Vergleich zu einer Schönheits-Behandlung viel teurer.	1	0
100	Die Schülerzahlen im Jahrgang A und B sind gleich hoch	1	0
101	Er hat sich hart trainiert, um ein marathonfähiger Läufer zu werden, und ist jetzt wie ein bisschen Pferd auf dem Weg zum Ziel.	0	1
	Die Herstellung von Musikinstrumenten erfordert eine Menge Geduld und Technik, um Stimmung und Ton zu erzeugen, die präzise wie ein Klavier.	0	1
103	Ein Mann wie ein Bär	1	1
	Ein Mann wie ein Löwe	1	1
	Ein Mann wie ein Elefant	1	1
	Ein Mann wie ein Gorilla	1	1
	Ein Mann wie ein Kamel	1	1
	Ein Mann wie ein Tiger	1	1
	Ein Mann wie ein Panda	1	1
	Ein Mann wie ein Bison	1	1
	Ein Mann wie ein Hase	1	1
	Ein Mann wie ein Zwerg	1	1
	Ein Mann wie ein Pony	1	1
	Ein Mann wie ein Schaf	1	1
	Ein Mann wie ein Eichhörnchen	1	1
	Ein Mann wie ein Waschbär	1	1
	Ein Mann wie ein Känguru	1	1
	Ein Mann wie ein Schimpanse	1	1
	Ein Mann wie ein Delfin	1	1
	Ein Mann wie ein Albatros	1	1
	Ein Mann wie ein Seeskürpe	0	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
104	Das ist das gleiche wie mit der Atomkraft. Somit kam der in Las Vegas lebende Münchener auf das gleiche Ergebnis wie 2009. Das war dasselbe Zimmer, das Marschall Schukow betrat, ganz rot im Gesicht, und auf der anderen Seite der blasse deutsche General mit dem Stab des Heerführers in der Hand.	1 1 1	1 1 1
111	Die Art der Beschäftigung mit dem Thema entspricht der Sichtbarkeit, die das Thema hat in der Gesellschaft Je konkreter der Text ist, desto mehr Informationen enthält er Je häufiger wir diese Art von Erfahrung machen, desto enger werden wir uns mit ihr identifizieren	1 1 1	0 1 1
122	–	0	0
123	Ich fühle mich ähnlich unbäller als früher, als ich regelmäßig zum Fitnessstudio ging, um meinen Körper fit zu halten. Die Wirtin kocht gerne ähnlich scharfeckige Gerichte wie meine Tante.	0 1	0 1
125	In der Arbeit sei es wie in einem Marathon In der Politik sei es wie in einem Gladiatoren-Kampf In der Liebe sei es wie in einem unklaren Rätsel	1 1 1	1 1 1
127	Die Kekse sind genauso süß wie die Schokolade von einem Gebäckshop Das Gemüse ist genauso frisch wie das von einem lokalen Markt Das Haus ist genauso sauber wie das von einem Dienstboten Das Buch ist genauso interessant wie das von einem Autor mit viel Erfahrung	1 1 1 1	0 0 0 0
128	Das Verhältnis zwischen Arbeit und Freizeit ist wie die Beziehung zwischen dem Essensverzehr und dem Gewichtsverlust - artig ähnlich Der Konsum von Alkohol und Rauchen liegt in einem äquativ-kontrollierten Verhältnis - artig ähnlich Das Verhältnis zwischen Konsum und Sparen ist wie das Verhältnis zwischen dem Körpergewicht und der Wahrnehmung des eigenen Körperfildes - äquativ ähnlich	1 1 1	0 0 0
129	Russische Staatshacker nutzten Kasperskys Antiviren - Software wie eine Art Google , um die Computer von Kunden – darunter mehrere US - Behörden – nach sensiblen Informationen und Codenamen für Programme von US - Geheimdiensten zu durchsuchen Viele der neuesten Kryptowährungen , der Coins und Tokens , die auf den Markt kommen , funktionieren hingegen eher wie eine Art Börsengang ohne Börse. Das ist eine Art parlamentarische große Koalition aus Sozialdemokraten , linken Liberalen und Zentrum , also jenen Parteien , die seit der Reichstagswahl von 1912 die Mehrheit in Berlin stellen.	1 1	1 1
130	–	0	0
132	Der Faultier gleicht einem Schirm Das Buch gleicht einem Reiseführer Die Blume gleicht einem Regenbogen Die Sängerin gleicht einer Opernsängerin Der Film gleicht einer Maskerade Der Keks gleicht einem Rutschrutsch Der Spiegel gleicht einer Schattenwand	1 1 1 1 1 0 1	1 1 1 1 1 1 1
133	Als ob der Uhrzeitpunkt anders als der aktuelle Datumszeitpunkt war Als ob der Computer aus den 80er Jahren still war Als ob die Luft in einer Höhle ganz anders war als in einer Stadt	0 1 1	1 1 1
135	Die Farbe des Himmels ist so blau Die Königin hat so viele Zähne wie ein Zahnarzt Das Wetter ist so heiß Der Tisch ist so hart Die Statue ist so groß Das Auto ist so schnell	1 1 1 1 1 1	0 0 0 0 0 0
136	Es ist wie wenn man eine Schachtel mit Gemüestückchen isst, aber sie alle zusammenfügen kann Das ist wie wenn man in eine Schachthalle gehen möchte, aber sie mit Ketten aus Schwermetall verschlossen ist Es ist wie wenn man einen Korkenzieher in eine Pfütze werfen möchte, aber er an der Kante festgehalten wird	1 0 1	1 1 1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
	Das ist wie wenn man in einer Höhle gehen möchte, aber sie von Schatten umgeben ist	0	1
139	Köpfe sind ähnlich im Winter	1	0
	Die Temperaturen im Frühjahr und im Sommer waren ähnlich	1	0
	Diese Tiere sind ähnlich in ihrem Verhalten	1	1
320	Statt den Küchenabfall zu mischen , schüttet sie ihn direkt in die Müllabfüllung.	1	1
	Woherum es ihm schlecht geht, weiß er selbst am besten.	0	0
	Während er ein gebürteter Baum wartet, ist der kleine Baumknacker weggegangen.	0	1
349	—	0	0
350	—	0	0
379	Die Hauptstadt von Deutschland ist Berlin.	1	0
	Die Gesamtzahl der Corona-Verkaufseinnahmen betrug 1,2 Milliarden Euro.	1	0
392	Ich würde gerne in die Fußspuren des berühmtesten Komponisten der Welt treten.	1	1
	Wenn sie den Job nicht bekommen würden, würde sie ihren Job aufgeben.	1	1
	Er würde gerne ein erfolgreicher Geschäftsmann werden.	1	1
488	Da der Klimawandel für viele Regionen am meisten Auswirkungen hat.	1	1
492	Es ist eine sinnlose Aufgabe, zwischen einem Dessert und einem Obst zu wählen, was am wenigsten unangenehm ist	1	1
	Die Sache mit diesem Test ist, dass er am meisten auf die Kenntnisse des Testierten achtet und am wenigsten auf seine Fähigkeiten	1	1
	Ich habe mich am meisten sorgen gemacht darüber, ob ich meine Pässe rechtzeitig auf dem Tisch habe, als ob es etwas mit meinem Job zu tun hatte	0	1
	Das Beste am Wochenende ist, dass ich am meisten Zeit mit meinen Freunden verbringe, aber am wenigsten mit meiner Familie	1	1
	Ich habe mich am meisten darum gekümmert, dass ich genug Energie habe, um den ganzen Tag zu arbeiten und am wenigsten darum, dass ich genug Schlaf habe, um mich auszuruhen.	1	1
500	—	0	0
557	—	0	0
559	Der neue Film von Quentin Tarantino wurde erst vor ein paar Wochen veröffentlicht	1	0
	Bis vor heute Morgen habe ich immer noch nichts zu meinem Prüfungsfach gelernt	1	1
579	Die Angst vor der Unklarheit (Vagheit) der Zukunft ist eine der allgemeinsten Ängste der Menschen.	1	1
	Letzte Woche (Samstag) war ich auf dem Markt und sah einen Mann mit einem schwarzen Hund.	1	1
	Im Jahr 2020 beschloss der Rat der Europäischen Union (EU) eine klimapolitische Agenda zu erlassen.	1	1
581	Mit dem Sinn für Farben war er ganz allein	1	0
	Er hatte eine Vorliebe für Tiere	1	0
	Seine Leidenschaft war die Musik	1	0
	Sie sprach Deutsch auf Deutsch	0	0
	Er hat Deutsch als Fremdsprache gelernt	1	0
	Sie hat im Ausland studiert	1	0
584	Der Strom heult und knistert, als er die Wände des alten Hauses erfasst (siehe das Album der Band).	1	0
	Die Blätter flattern im Wind, der stärker ist als die vorhergesagte Wettervorhersage (siehe den aktuellen Bericht).	1	0
590	Der Flughafen von Montreal hat gestern (Montag) 1.35 Millionen Passagiere verzeichnet	1	1
	Im Dezember 2019 stieg der Preis für Brent-Öl um 27 Prozent auf 65 Dollar pro Barrel	1	0
	Die Sorte Tresca ist die beliebteste bei den Verbrauchern	1	0
592	Haben Sie sich denn schon einmal in einem Theater befunden, wenn der Schauspieler direkt mit der Zuschauerfront spricht?	1	0
	(nachdenklich)	0	1
	(ein leises "Nee")	0	1
595	Zwei Jahre Arbeit in der Pharmafirma	1	0
	Das war ein großartiger Erfolg	1	0
	Ich habe mich nach dem Abitur für Philosophie an der Universität beworben	1	0
600	Ich denke, dass die aktuelle Handlung des Spielers überraschend war (?)	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
	Ich hoffe, dass wir schnell eine Lösung finden, die diese unangenehmere Situation beseitigt (?)	1	1
	Der Kaffeekränzchen genießt seinen Morgenkaffee in einer erhöhten Stellung (?)	0	1
	Der Reiseführer behauptet, dass das Rauchen im Auto gesetzlich erlaubt ist (?)	1	1
	Es ist viel gerüchteweise behauptet, dass der König in der Tat ein Eichhörnchen sein könnte (?)	0	1
	Nachdem der Koch den Suppenkrücken versammelt hatte, befand er, dass die Mischung etwas zu schwefelig schmeckte (?)	0	1
	Der Politiker meinte, dass die letzten Wahlen mit der besten Wahlkampagnen gewonnen wurden (?)	0	1
	Der Nachtschwärmer sammelte nach einiger Zeit mehrere muffige Düte von Keksen (?)	0	1
605	Es gab eine Zeit, da der Bauernhof einen Kuhstall hatte.	1	1
	Es gab eine Zeit, da die Blätter auf den Bäumen noch grün waren.	1	1
	Es gab eine Zeit, da die Straßen noch schmutzig waren.	1	1
618	Mit ihren unermüdlichen Arbeiten im Hospital hatte ihre Kollegin die höchste Auszeichnung im Bereich der Notfallmedizin geerbt.	1	0
647	—	0	0
651	Nenn mich nicht Mords-Kater, ich bin ein Elativ-Gesell	1	1
	Er trägt immer Mords-Kleidung, um sich selbst zu unterstützen	1	1
	Ich habe ein Mords-Stellenangebot abgelehnt	1	1
654	Ich habe eine Riesenbemerkung gemacht	1	1
	Ich habe eine Riesenbelohnung erhalten	1	1
	Ich habe eine Riesenleistung gezeigt	1	1
	Ich habe eine Riesenbelastung getragen	1	1
	Ich habe eine Riesenfinanzierung erstellt	1	1
	Ich habe eine Riesenersparnis gemacht	1	1
	Ich habe eine Riesenstrecke zurückgelegt	1	1
	Ich habe eine Riesenbremse gebraucht	1	1
	Ich habe eine Riesenmasse von Arbeit zu erledigen gehabt	1	1
671	Der Supermarkt am Ort ist der beste unter den Supermärkten im Land.	1	1
	Die Universität ist die beste unter den Universitäten im Land.	1	1
	Das Restaurant in der Stadt ist das beste unter den Restaurants im Land.	1	1
674	Er sagte nicht, was er meinte	1	0
	Keine Erklärung für die Verzögerung	1	0
	Er würde sich nicht erinnern, warum er da war	1	0
675	—	0	0
676	Die neuen Arbeitsbedingungen sind äquivalente zu dem Schicksal	0	0
	Das neue Arbeitsmodell käme einem Kampf gegen das Unrecht äquivalente zu dem	0	0
	Die Aufgabe äquivalente zu dem Kampf gegen das Unrecht	0	0
	Das neue Arbeitsmodell äquivalente zu dem Kampf gegen das Unrecht	0	0
681	Das Auto hat einen Reifenplatten, weil es unterwegs einen Schlag bekommen hat	1	1
	Weil du dafür die Reifen pünktlich auf dem Fahrrad hastest	1	1
	Weitermachen, weil du das Gefühl hatte, als hättest du es nur auf dem Rad gehabt	1	1
683	Die mit Abstand aufregendste Filmkomödie des Jahres ist sicherlich 'Die Wunderkinder'	1	1
	Mit Abstand am häufigsten wird das Online-Banking-System von Commerzbank genutzt	1	1
	Im Vergleich zu den anderen Smartphones ist das iPhone mit Abstand das teuerste	1	1
697	—	0	0
758	Es ist nicht einfach, in aller Stille eine Karriere als Künstler zu etablieren.	1	1
	In aller Eile gibt man seine Arbeit nicht auf.	1	1
762	Ein bisschen frühstückskaffee trinken, bevor man auf die Arbeit fährt	0	1
	Ein bisschen Wärme in der Wohnung haben, bevor man sich auszieht	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
	Ein bisschen Geduld haben, wenn man im Verkehr steht	1	1
777	Der Streit zwischen den beiden Parteien wurde ganz und gar entschieden.	1	1
	Das neue Album war ganz und gar anders als das vorige.	1	1
	Ich habe mich ganz und gar auf das neue Unternehmen konzentriert.	1	1
858	Zwischen dem deiktischen Zentrum und dem zukünftigen Zeitpunkt wird eine Zeitspanne von 10 Jahren veranschlagt	1	0
	In neun Monaten nach dem Start seiner Behandlung verschwanden die Symptome der Patientin	1	0
	Die neueste Serie von Filmen wird im Sommer in den Kinos angeboten	1	0
	Das Restaurant wird in ein paar Wochen eröffnen	1	1
	In zwei Jahren wird das neue Computer-System eingeführt	1	0
875	Max hat das neue Karrierewechsel-Konzept präsentiert .	1	0
	Das neue Auto ist die aktuelle Nachricht im Auto-Verkauf .	1	1
	Der neue Trend sind langhalsige Kollar .	1	0
882	—	0	0
892	Als sie aber nach draußen auf die Straße ging, zog sie eine Burka an (die in SansibarNinja heißt, in Anlehnung an die japanischen Kämpfer, die man auch dort aus Filmen kennt)	1	0
900	Der 50-jährige 1.80-Meter-Stürmer wurde mit 16 Treffern mit dem Münsteraner Nicolai Müller (DM) Torschützenkönig.	1	1
	Solange die deutschen Wirtschaftsbeziehungen mit Russland nicht sicher genug sind , wird es auch das Erfordernis von Binnengrenzkontrollen geben	1	0
	Außenminister Sigmar Gabriel (SPD) , der nicht dem Verhandlungssteam seiner Partei angehört , hatte betont , dass es ohne die Bereitschaft für Europareformen im Sinne des französischen Präsidenten Emmanuel Macron keine Koalition mit der SPD geben werde	1	1
902	Nicht schnell, sondern langsam	1	0
	Nicht groß, sondern klein	1	0
	Nicht kalt, sondern heiß	1	0
	Nicht heiß, sondern kalt	1	0
	Nicht hart, sondern weich	1	0
	Nicht viel, sondern wenig	1	0
904	Das Auto war schnell und nicht erschwerlich	0	1
	Der Bierkrug war voll und nicht leer	1	1
	Das Haus war bequem und nicht ungemütlich	1	1
	Das Essen war lecker und nicht schlecht	1	1
	Der Garten war aufgerissen und nicht gepflegt	1	1
	Die Musik war lustig und nicht traurig	1	1
	Das Buch war interessant und nicht langweilig	1	1
	Das Film war begeisternd und nicht enttäuschend	1	1
	Das Projekt war erfolgreich und nicht fehlgeschlagen	1	1
905	Von den 100 Pilzarten, die im Zoo gezeigt werden, sind 77 Arten endemisch in Deutschland vorzufinden.	1	0
	Im Jahr 2000 wurde die Universität in der Nähe des Flughafens eröffnet, an der sich inzwischen rund 5000 Studierende immatrikuliert haben.	1	0
907	Am 6. April 1945 begann die Schlacht um Berlin (1945).	1	1
	Im Jahr 2023 (2023) wird die Weltpostkonferenz in Genf stattfinden (2023)	0	1
	In seinem Werk Über die Kunst im Werk (1978) beschreibt Gotthard Graubner die Ästhetik des Raums in der Malerei. (1978)	1	1
909	—	0	0
917	In der modernen Musikindustrie (MMI) haben die Musikverlage (MV) eine wichtige Rolle im Vertrieb von Musik (V) spielen.	0	1
	In der Arbeit (A) gibt es viele Abläufe (A).	1	1
919	Wow , das ist wahnsinnig !	1	0
	1 + 1 = 2	1	0
	Ich heiße Max	1	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
	Das Leben ist schwer	1	0
	Am Samstag werde ich Party machen	1	0
	Der Kater schläft am Nachmittag	1	0
	Wer ist der bessere Fußballer?, Cristiano Ronaldo oder Lionel Messi?	0	0
	Am Sonntag wird es regnen	1	0
	Das Leben ist eine Reise	1	0
	Ich habe gelesen	1	0
	Die Welt ist rund	1	0
	Ich kann nicht schwimmen	1	0
	Ich habe einen Job	1	0
	Ich habe ein Auto	1	0
921	—	0	0
949	Keine Zwiebel ohne Knobeln	1	1
	Kein Tuchus ohne Knobeln	0	1
	Kein Pferd ohne Pferdefüße	1	1
973	Nietzsche - oder der Wille zur Macht	1	1
	Theorie der Gerechtigkeit - oder die Rolle des Rechts	1	1
	Das Können - oder die Kunst des Selbstmanagement	1	1
976	—	0	0
1004	Besser	0	0
	Die besten	0	0
	Groß	0	0
	Größer	0	0
	Nie komplexer	0	0
	Nie selbsterfüllender	0	0
	Nie noch länger	0	0
1029	Was muss man beim Einkaufen im Supermarkt beachten?	1	0
	Warum ist Morgenstunden das beste Zeitpunkt für Sport?	0	0
	Wie kann man seine Energieeffizienz verbessern?	1	0
1033	—	0	0
1034	Das Gewinnstreben verlagerte sich zu Finanzinvestitionen, das Wirtschaftswachstum sank von Jahrzehnt zu Jahrzehnt, das Finanzierungsdefizit wanderte von den Unternehmen zum Staat, Arbeitslosigkeit und Staatsverschuldung stiegen.	1	1
	Des Öfteren steckten Polizisten Knöllchen an den Wagen, einmal kamen Diebe und stahlen ihn, und von Jahrzehnt zu Jahrzehnt fiel es den Ausstellungsmachern schwerer, ein weiteres Exemplar desselben Typs zur Anmietung aufzutreiben.	1	1
	Doch von Spiel zu Spiel wurde er besser und es gibt nicht wenige, die sein Understatement im Vorhinein als psychische Spielerei sehen.	1	1
1035	Werden Personen mit bestimmten gesellschaftlichen Roles als 'Entitäten' betrachtet?	1	0
1126	Die Generation Z	1	1
	Generation X	1	1
	Generation Alpha	1	1
	Generation Y	1	1
	Generation Millennial	1	1
	Generation Gen Z	0	1
1134	Die Nachfrage für den Umwelt-Rothschild-Papageien hat sowohl gestiegen als auch abgenommen, was auf eine zunehmende Beschäftigung mit Umweltthemen im Allgemeinen zurückzuführen wird.	0	1
	Die Behörden haben sowohl Maßnahmen zur Verhinderung von Quereinstiegshandlungen als auch Maßnahmen zur Förderung des Öffentlichen Personenkraftverkehrs ergriffen.	1	1
1140	—	0	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
1162	—	0	0
1289	—	0	0
1291	Wenn sie nur nicht so eifersüchtig wäre	1	1
	Wenn er sich nur hin und wieder beherrschen könnte	1	1
	Wenn sie nur nicht so viel Zeit hätte	1	1
1300	So muss Radfahren mit Motor gehen.	1	1
	So muss das Radfahren mit Motor gehen.	1	1
1301	Der Augenblick, wenn der Vogel singt	1	1
	Dieses Gefühl, wenn du siehst	1	1
	Der Moment, wenn du verstehst	1	1
	Das Gefühl, wenn du zuhörst	1	1
	Dieser Augenblick, wenn du lachen musst	1	1
	Das Gefühl, wenn du weinst	1	1
	Der Moment, wenn du siehst	1	1
	Der Augenblick, wenn du lachst	1	1
	Das Gefühl, wenn du lachst	1	1
	Dieser Augenblick, wenn du aufhörst zu lachen	1	1
1313	Es ist zum Verzweifeln	1	1
	Es ist zum Wahnsinnigen Tanzen	0	1
	Es ist zum Zittern	1	1
1315	Als ob es sich um eine Bewertung handelte	1	1
	Als ob ich der Meinung bin, dass die Luft auf dem Mond dick ist	1	1
	Als ob ich nicht bemerkte, dass sie schwanger war	1	1
1316	Aus dem Auto raus , rein in den Wald zu campieren.	1	1
	In ein neues Land ziehen , aus dem alten heraus raus , rein in die neue Wohnung	1	1
	Aus dem Büro raus , rein in die Natur zu wandern	1	1
	Die Arbeit loswerden , rein ins Freie , in die Natur zu gehen	0	0
	Aus dem Gewohnstem , rein in das Abenteuer	0	0
	In die Ferien raus , aus dem Alltag heraus	1	0
	In die Stadt zurückkehren , raus aus dem Land	1	0
	In eine andere Stadt ziehen , aus dem alten heraus raus	0	0
	In eine andere Stadt ziehen , raus aus dem Heim	1	0
1323	Praktikum pur auf dem Campus, was mich dazu bringt, dass ich meine Fähigkeiten im Bereich der Sozialen Arbeit verbessern möchte	1	1
	In der heutigen Woche ist das Arbeitsumfeld pur und ich möchte mich auf meine eigenen Fähigkeiten konzentrieren	0	0
	Der Kurs wird pur in deutscher Sprache gehalten, damit die Absolventen ihre Sprachkenntnisse verbessern können	1	0
1324	Schokolade satt, süß und verschmutzt	1	1
	Abendsonne satt und die Glühbirnen erhellen die Straßen	1	1
	Bier satt, kalt und köstlich	1	1
1329	Oder weil ein Essen nicht jedermanns Sache ist	1	1
	Und doch unterscheidet ihn wohl einiges von der pausenlos arbeitenden Büro-Mitarbeiter-Legende, die oft überfordert und geschwitzt wirkte und dessen Abläufe nicht jedermanns Verständnis trafen	1	1
	Ich will nicht jedermanns Sache, sondern nur von der, die bereit ist, im Rahmen unserer Gesetze im öffentlichen Park mit Sport zu spielen.	0	1
1342	Ein echter Sprachgenießer par excellence ist der Dialektologe Wolfgang Mieder.	1	1
	Der Nachrichtenreporter par excellence Gerald J. Schmitt hat eine große Auswahl an Informationen.	1	1
1346	Der Strippenzieher	1	0
	wie er im Buche steht	1	0

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
1347	—	0	0
1351	Können wir mit euch - Stiefel hin , Stiefel her - das Problem besprechen?	1	1
	Da wir als Ingenieur - Modell hin , Modell her - in die Lage sind, umfangreiche Projekte zu realisieren.	1	1
1459	Morgen wird das, was heute Friseur war, das, was früher Kopfschneider war	—	0
	Heute ist das, was früher ein Auto war, das, was heute noch immer ein Auto ist	1	1
	Morgen wird das, was heute ein Rucksack war, das, was früher ein Fackelträger war	0	1
	Heute ist das, was früher ein Telefon war, das, was heute noch immer ein Telefon ist	1	1
	Morgen wird das, was heute ein Bildschirm war, das, was früher ein Fernseher war	0	1
1461	Das Leben ist kein Pferd und Du bist nicht auf dem Tier	1	1
	Das Leben ist kein Reiseplan und Du bist nicht in der Position, es umzumachen	0	1
	Das Leben ist kein Saft und Du bist nicht bereit, darüber zu sprechen	1	1
1462	Ponyhof ist kein Ponyhof	0	1
1503	Intensiviertes Essen brutal	1	1
	Intensiviertes Musik brutal	1	1
	Intensiviertes Schlaf brutal	1	1
1509	Intensivieren, bis der Arzt kommt	1	1
	Intensiviertes, bis der Arzt kommt	0	1
	Intensivieren, bis der Arzt eintrifft	1	1
	Intensivieren, bis der Arzt kommt und sich bedient	1	1
1511	Jugend und Arbeit hin , Leistung und Anpassungsfähigkeit her	1	1
	Liebe hin , Wahrheit her , so etwas darf nicht passieren	1	1
	Komfort hin , Freiheit her	1	1
	Freiheit hin , Schande her	1	1
	Hilfe hin , Selbstloskeit her	0	1
1525	Kein Essen für niemand.	1	1
	Keine Freizeit für niemand.	1	1
	Kein Schlaf für niemand.	1	1
	Kein Geld für niemand.	1	1
	Kein Job für niemand.	1	1
	Kein Glück für niemand.	1	1
	Keine Hilfe für niemand.	1	1
	Kein Zeit für niemand.	1	1
	Keine Energie für niemand.	1	1
	Keine Geduld für niemand.	1	1
	Kein Raum für niemand.	1	1
	Keine Liebe für niemand.	1	1
1554	Redend Silber , Wartenden Gold	0	0
	Intensiviertes Silber , Intensiviertes Gold	1	0
1573	Mit einer Bewegung bewegte sie sich im Raum, der ihr vertraut war, der Musik entsprach, die ihr immer gefallen hatte.	1	1
1574	Der Mann, der sich selbst als Herr der Welt bezeichnete, erkannte schließlich, dass er nur ein armer Sündenfall war.	1	1
	Die Idee, dass die Regierung der Vereinigten Staaten eine mächtige Institution darstellt, ist eine Illusion.	1	0
	Wer mit dir im Bett liegt, der ist schön und lieblich, aber auch unerträglich ständig.	0	0
1582	Das Kind hat sich eine Gasse gekämpft und zurückgelegt.	0	1
1593	Münster kann Katholikentag	1	1
	Stoiber kann Kampagne	1	1
	Die SPD kann Kanzler	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

Konstruktion	generierter Satz	grammatisch	zutreffend
1597	Die dem Menschen am nächsten stehenden Affen sind die so genannten Menschenaffen (Schimpansen, Bonobos, Gorillas, Urang-Utans), während Halbaffen (Spitzhörnchen, Lemuren, Koboldmakis) sozusagen auf der anderen Seite stehen, den Affen ähnlich, die andere Hälfte ist aber, etwa wegen der feuchten hundeähnlichen Schnauze, nicht den Menschen, sondern anderen, kleineren Säugetieren wie Hunden oder Mäusen ähnlich.	1	1
1602	—	0	0
1629	—	0	0
1630	Denn allen Vorwürfen des Plagiats zum Trotz war sie immer noch der bestbewertete Schüler in ihrer Klasse.	1	1
	Denn allen Bedenken und Zweifeln zum Trotz entschied sich die Jury für den Verteidiger.	1	1
	Denn allen Schreckensmeldungen und Krankheitswarnungen zum Trotz fuhr sie jeden Tag zur Arbeit.	1	1
1631	—	0	0
1637	Die Goldfische in der Haltung sind in einem großen Aquarium untergebracht, das durch ein Netz aus Fischen und Schalwasser vor Fischen geschützt wird. Das Aquarium wird von einem runden Becken mit einem Durchmesser von 1,6 Meter umgeben, das durch eine Aluminiumröhre konstruiert ist. Die Beckenwand ist mit Schale aus Kunststoff überzogen und mit Schleim abgedichtet.	0	0
	In der Regierungszeit von Mussolini wurden in Deutschland die Juden verfolgt und Opfer von Deportation und Deportation werden in ein Konzentrationslager wie Auschwitz-Birkenau transportiert. Die Deportation wurde durch den Nationalsozialismus durchgeführt, der den Holocaust erfüllte. Die Deportation führte zu massiven Verlusten an Menschenleben.	1	0
	Auf dem Vormarsch der Alliierten auf Berlin im Zweiten Weltkrieg, wurde die Stadt durch die Rote Armee belagert und belagert in den Wintermonaten. Die Stadt war zerstört worden, als die Armee schließlich in die Stadt eindrang. Die Rote Armee setzte dabei ihre gesamte Panzerabteilung ein.	0	0
1641	Von den 52.000 Teilnehmern des Marathons waren 19.500 Männer.	1	0
	Nachdem sie die Kalkulation durchgeführt hatte, fühlte sich die Frau beruhigt.	1	0
1643	—	0	0
1649	Lena sitzt auf ihrem Küchenbankstuhl und schwelgt in der köstlichen Leckerei ihres gebratenen Hähnchens.	0	0
	Das Mittagessen wird heute von der Familie Langer in der Gaststätte zum Leckermaul bejubelt.	0	0
1660	—	0	0
1671	Ein Zwerg	1	0
	Ein Riese	1	0
	Ein Hobbit	1	0
	Ein Kater	1	0
1681	Der Typ aller modernen Technologien	0	0
	Die Mutter aller moderner Technologien ist das Smartphone	1	1
	Die Mutter aller moderner Technologien ist die Kunst der Architektur	1	1
1690	Die Nachricht ist nicht unerwartet, sondern kaum unvorhergesehen.	0	1
	Kein unbedenklicher Entscheidungsgrund.	1	1
	Nicht unglücklich, sondern nur gedörrt.	1	1
	Kein unnötiger Aufwand, sondern nur eine ungereimte Darstellung.	1	1
	Nicht unerwünscht, sondern nur ungünstig.	1	1
	Kein unerwarteter Verlust, sondern nur ein unglücklicher Zufall.	1	1
	Nicht unerfindlich, sondern nur ein unnötiger Auslauf.	0	1
	Kein ungenügender Erfolg, sondern nur ein unbefriedigender Ergebnis.	0	1
	Nicht unfassbar, sondern nur ungereimt.	0	1
1715	Kein Hund ist auch ein Hund	1	1
	Kein Auge ist auch ein Auge	1	1
	Kein Handy ist auch ein Handy	1	1
	Kein Auto ist auch ein Auto	1	1
	Kein Haus ist auch ein Haus	1	1
	Kein Lkw ist auch ein Lkw	1	1
	Kein Laptop ist auch ein Laptop	1	1
	Kein Kühlschrank ist auch ein Kühlschrank	1	1
	Kein Kabel ist auch ein Kabel	1	1

Tabelle wird auf der nächsten Seite fortgesetzt.

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

Konstruktion	generierter Satz	grammatisch	zutreffend
	Kein Baum ist auch ein Baum	1	1
1738	Wie verdammtn genug ist das denn schon !	1	1
	Wie unglaublich ist das denn ?	1	1
	Wie unglaublich ist das denn !	1	1
1756	Ich hatte geglaubt, dass er schon zu Hause war.	1	0
	Es wäre schön, wenn sie bald zurückkehren würde.	1	1
	Wenn ich früher geübt hätte, wäre ich besser auf dem Spielplatz geworden.	1	1
1760	Doch so ungern er über seine Musik spricht , so sehr ist er eben ein Meister seiner sinnlichen Umsetzung .	1	1
	So lange nicht alle Zweifel ausgeräumt sind , so lange wird der Ratifizierungsprozess ausgesetzt , sagte Tusk .	1	1
	So wenig sie sich - trotz dessen Attacken gegen ihre Flüchtlingspolitik - in den US-Wahlkampf eingemischt hatte , so klar zeigte sie dem neuen mächtigsten Mann der Welt sofort nach dem Wahlsieg eine Grenze auf .	1	1
1762	Besser ein tadelloses Verhalten als ein schlechtes	1	1
	Besser ein kleiner Fehler als ein größerer	1	1
	Besser etwas zu wenig als etwas zu viel	1	1
1770	—	0	0
1772	Von wegen einfach nur das Höchste zu geben	1	1
	Von wegen nichts gibt sich einfacher als das Leben	0	1
1777	Der neue Wagen braucht nur 5 Liter Benzin im Monat, im selben Maße wie der alte Wagen	1	1
	In diesem Jahr habe ich im selben Maße wie im Vorjahr gespart	1	1
	Das neue Restaurant bietet im selben Maße wie das alte mehr verschiedene Kuchengerichte an	1	1
1779	Entweder man ist ein Mann oder nicht	1	1
	Entweder der Himmel ist blau oder nicht	1	1
	Entweder es ist Sommer oder nicht	1	1
	Entweder die Katze ist grün oder nicht	1	1
1792	Keiner ist unschlagbar	1	1
	Niemand ist perfekt	1	1
	Keine Verzahnung ist fehlerfrei	1	0
	Kein Gewicht ist unbedingt	1	0
	Kein Produkt ist komplett fehlerfrei	1	0
1831	So schnell sie auch retten , kann ich mich nicht daran gewöhnen , dass sie immer noch untreu bleiben .	1	1
	So fröhlich sie auch die ganze Woche lang lachen und lachen , denken sie an jedes einsame Wochenende zurück und fragen sich , was sie eigentlich von ihrem Leben machen müssen.	0	1
1835	Kein Frühling ist auch ein Sommer	1	1
	Kein Essen ist auch ein Trinken	1	1
	Kein Tag ist auch eine Nacht	1	1
1846	Was ist mit dem Computer?	1	1
	Was ist mit dem Auto?	1	1
	Was ist mit dem Tablet?	1	1
	Was ist mit dem Mobiltelefon?	1	1
	Was ist mit dem Fernsehen?	1	1
	Was ist mit dem Radio?	1	1
	Was ist mit dem Laptop?	1	1
	Was ist mit dem Handy?	1	1
	Was ist mit dem Smartwatch?	0	1
	Was ist mit dem Tablett?	1	1
	Was ist mit dem Fernseher?	1	1
1849	Ich und der neue Job ?	1	1

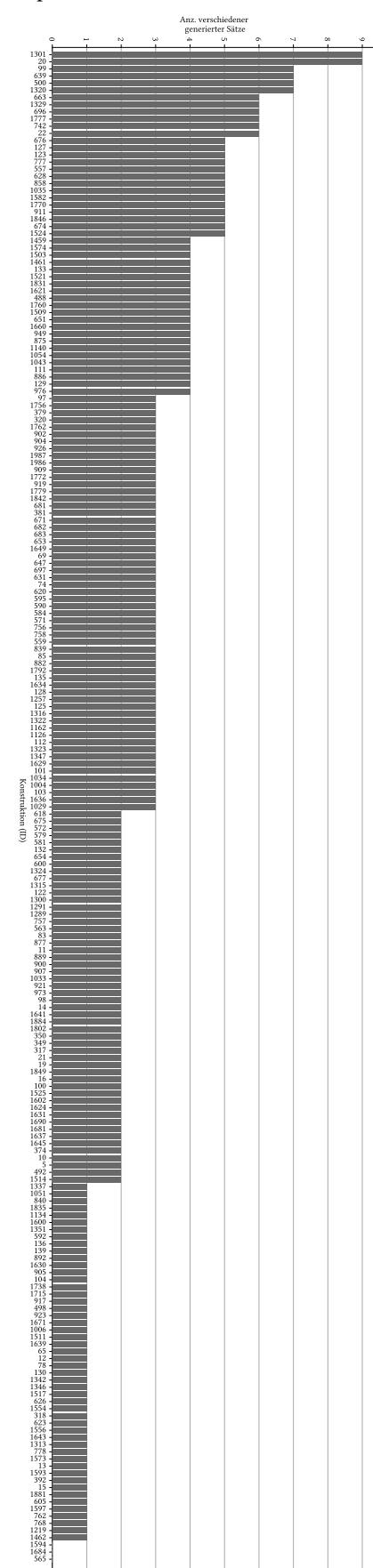
Tabelle wird auf der nächsten Seite fortgesetzt.

A Ergebnisse und Rohdaten

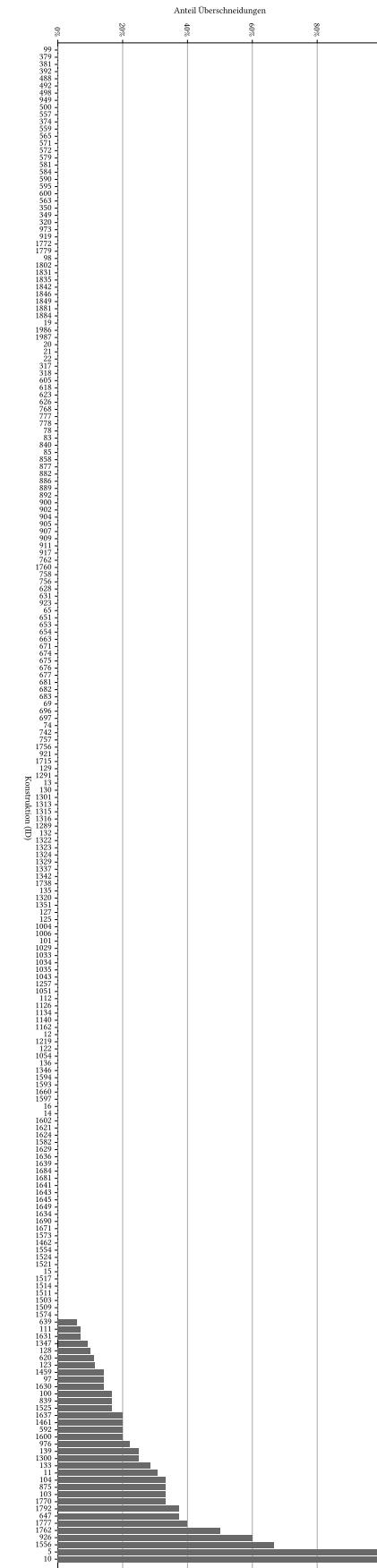
Konstruktion	generierter Satz	grammatisch	zutreffend
	Du und die neue Beziehung ?	1	1
	Sie und das neue Haus ?	1	1
	Er und die neue Stadt ?	1	1
	Wir und das neue Auto ?	1	1
1881	Ein Mann ist kein Mann	1	0
	Ein Hund ist kein Elefant	1	1
	Schokolade ist kein Brokkoli	1	1
	Ein Sommer ist kein Winter	1	1
	Der Mond ist kein Planeten	0	0
	Ein Haus ist kein Schiff	1	1
1884	—	0	0
1986	Die Vorhut vor 20.000 Jahren war für das Riesen-Bodenfaultier kaum existenzgefährdend.	1	1
	Je 500 Millionen Euro wurden in den vergangenen beiden Jahren in den Irak geschickt - eine große Summe, doch sie erscheint kaum ausreichend angesichts der enormen Schäden, die drei Jahre Herrschaft des IS im Land hinterlassen haben.	1	1
	Ohne diesen finanziellen Hintergrund wäre ihr die Pflege der Mutter kaum möglich gewesen.	1	1
1987	Irgendwann an diesem Abend muss Milanti* sehr intensiv staunen.	1	1

A.2 Daten zur Generierung von Textbeispielen zu Konstruktionen

(a) Anzahl verschiedener generierter Sätze pro Konstruktion durch Llama 2

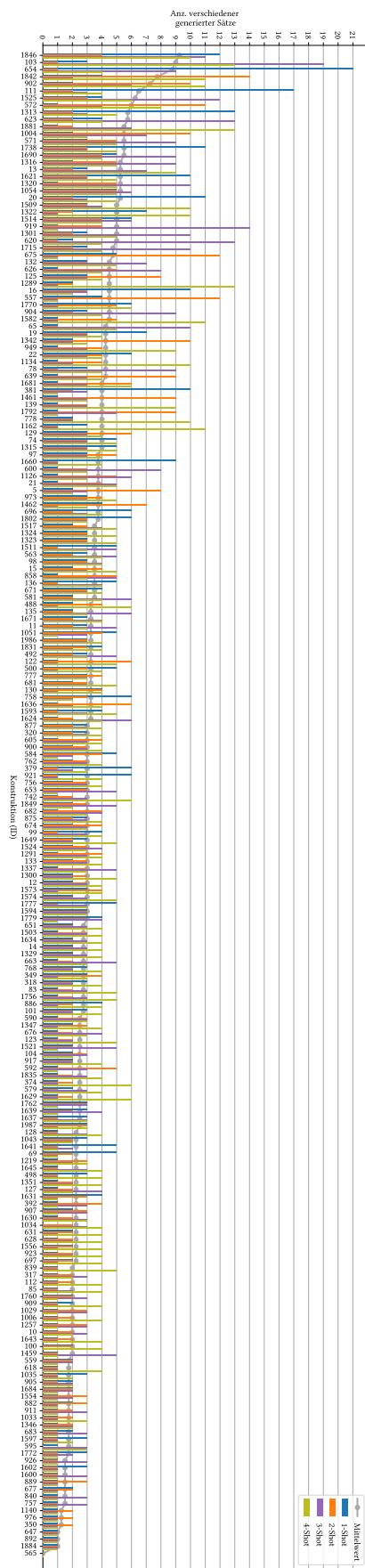


(b) Anteil an Überlappungen zur gegebenen Definition in den durch Llama 2 generierten Sätzen pro Konstruktion



A Ergebnisse und Rohdaten

(a) Anzahl verschiedener generierter Sätze pro Konstruktion durch Llama 2



B Verzeichnis der untersuchten Konstruktionen

Hier sind die Konstruktionen des *FrameNet-Konstruktikon des Deutschen* aufgelistet, die für die Untersuchung mit den Sprachmodellen verwendet wurden.

Tab. B.1: Berücksichtigte Konstruktionen aus dem *FrameNet-Konstruktikon des Deutschen* (HHU, 2024); sortiert nach Web-ID und zugeordneter Kategorie

ID	Kategorie	Bezeichnung
5	Negation	NEG_X_und_schon_gar_nicht_Y
10	Negation	NEG_X_geschweige_denn_Y
11	Negation	NEG_X_und_wohl_kraum_Y
12	Negation	weder_XP_noch_XP
13	Exklamativ	was_für_ART.indef_N
14	Exklamativ	wie_ADJ
15	Exklamativ	welch_(ART.indef_)N
16	Exklamativ	was_VL
19	Exklamativ	dieser_N
20	Exklamativ	Allein_STIMULUS
21	Exklamativ	dass_VL
22	Exklamativ	so_ART.indef_N
65	Superlativ	ART_ADJ-(e)st
69	Tautologie	X1_ist_X1
74	Intensivierung	so_ADJ
78	Exklamativ	solch_(ART.indef_)N
83	Reduplikation_Konkurrenz	N1_gegen_N1
85	Reduplikation_Quantifizierung	N1_über_N1
97	Komparativ	ADJ1-er_als_ADJ1
98	Äquativ	so_ADJ_wie_XP
99	Komparativ	ADJ-er_als_X
100	Äquativ_Plural	gleich_ADJ
101	Äquativ	ADJ_wie_NP
103	Äquativ	ein_N_wie_ein_N
104	Äquativ	dasselbe_NP
111	Korrelativer_Komparativ	je_ADJ-er_desto_ADJ-er
112	Exklamativ	V1
122	Äquativ_Gleichsetzung	genauso_NP_wie_NP
123	Äquativ_partielle_Gleichheit	ähnlich_ADJ_wie_NP
125	Äquativ	PRÄP_NP_ist_es_wie_PRÄP_NP
127	Äquativ	ein_N_von_einem_NP
128	Äquativ_Komposition	N-artig

Tabelle wird auf der nächsten Seite fortgesetzt.

B Verzeichnis der untersuchten Konstruktionen

ID	Kategorie	Bezeichnung
129	Äquativ	eine_Art_X
130	Äquativ_Gleichsetzung	Ein_NP1_ohne_XP1_ist_wie_ein_NP2_ohne_XP2
132	Äquativ	X_gleicht_Y
133	Äquativ_Schlussfolgerung_hypothetisch	VP_als_ob_VP
135	Intensivierung_Schlussfolgerung	so_ADJP_dass_XP
136	Äquativ	X_wie_wenn_Y
139	Äquativ_partielle_Gleichheit_Plural	ähnlich_ADJ
317	Kontrast	V1_V1
318	Spezifizierung_Sachverhalt	Satzwertige_Partizipphrasse
320	Kontrast	Hauptsatz_Adversativsatz
349	Reduplikation_Nähe	N1_an_N1
350	Reduplikation_Kontinuität	N1_PRÄP_N1
374	Negation_Schlussfolgerung	nur_weil_X_NEG_Y
379	Entsprechung	Klammern
381	Komparativ	ADJ-er_als_NP
392	Konjunktiv	würde_V.inf
488	Superlativ	am_meisten_ADJ
492	Superlativ	am_meisten_V
498	Superlativ	ART_meist-V.part.perf
500	Superlativ	am_ADJ-sten
557	Zeitdeixis	in_der_SPEZIFIZIERUNG_ZEITEINHEIT
559	Zeitdeixis_Quantifizierung	bis_vor_NUM_ZEITEINHEIT
563	Kontrast	V2_V2
571	Konjunktiv_Präteritum	V.conj-II
572	Phrasenkompositum	XP-Y
579	Einordnung	Klammern
581	Konkretisierung	Klammern
584	Verweis	Klammern
590	Attribution	Klammern
592	Direkte_Rede_Ebenenwechsel	Klammern
595	Direkte_Rede	SprecherIn_Doppelpunkt_Turn
600	Kommentar	Klammern
605	Kontrast	es_gab_eine_Zeit_XP
618	Superlativ	NP_ohnegleichen
620	Superlativ	ART_NP1_der_NP1s
623	Superlativ_Komposition	ADJ-st-EINHEIT
626	Superlativ	INTENSIVIERUNG_NP
628	Superlativ	INTENSIVIERUNG_ADJ
631	Superlativ	ADJ-st_ADJ
639	Superlativ_Wortbildung	INTENSIVIERUNG-ADJ
647	Personenreferenz_expressiv	Vorname_Zitat_Nachname
651	Intensivierung_Elativkompositum	Mords-N
653	Intensivierung_Elativkompositum	sau-X
654	Intensivierung_Elativkompositum	Riesen-N
663	Intensivierung_Elativkompositum	X-Y
671	Äquativ	ein_NP_unter_den_NP

Tabelle wird auf der nächsten Seite fortgesetzt.

ID	Kategorie	Bezeichnung
674	Kausaler_Konnektor	weil
675	Äquativ	einem_NP_gleich
676	Äquativ	einem_X_gleichkommen
677	Additiv_Koordinativkompositum	X-X
681	Kausaler_Konnektor	weil_X
682	Intensivierung	INTENSIVIERUNG_N
683	Intensivierung	mit_Abstand_ADJ.sup
696	Konjunktiv_Präsens	V.conj-I
697	Intensivierung	ganz_und_gar_ADJ
742	Intensivierung	INTENSIVIERUNG_ADJ
756	Intensivierung_Reduplikation	ADJ1_ADJ1_NP
757	Intensivierung_Hervorhebung	ART_NP1_ART_NP2
758	Intensivierung	PRÄP_aller_NP
762	Relativierung	ein_bisschen_ADJ
768	Intensivierung	INTENSIVIERUNG_V
777	Intensivierung	ganz_und_gar_NP
778	Kategorisierung_exklamativ	du_NP
839	Relativierung	RELATIVIERUNG_ADJ
840	Relativierung	RELATIVIERUNG_V
858	Zeitdeixis_Quantifizierung	in_NUM_ZEITEINHEIT
875	Kategorisierung_Trendwechsel	X_ist_das_neue_Y
877	Quantifizierung	CARD_NP
882	Kategorisierung_Kontrast	XP1_statt_XP2
886	Sekundäre_Information	Klammern
889	Parenthese	Klammern
892	Texteinschub	Klammern
900	Einordnung_Mitgliedschaft	Klammern
902	Negation_Kontrast	NEG_X_sondern_Y
904	Kategorisierung_Kontrast_Negation	XP1_und_nicht_XP2
905	Einordnung_Lebensdaten	Klammern
907	Einordnung_Ereigniszeit	Klammern
909	Einordnung_Markenangabe	Klammern
911	Einordnung_Ortsangabe	Klammern
917	Entsprechung_Abkürzungseinführung	Klammern
919	Entsprechung_Abkürzungserläuterung	Klammern
921	Entsprechung_Übersetzung	Klammern
923	Entsprechung_Äquivalenz	Klammern
926	Quantifizierung	ORD_NP
949	Korrelation_Negation	kein_X_ohne_Y
973	Disjunktion_Doppeltitel	X_oder_Y
976	Korrelation_Affirmation	Wo_X_ist_ist_Y
1004	Superlativ_Klimax	ADJ1_ADJ1-er_NP
1006	Superlativ	PRÄP_ADJ-ster_NP
1029	w-Überschrift	w-VL-Satz
1033	Exklamativ	XP_Hauptache_YP
1034	Reduplikation_Kontinuität	von_N1_zu_N1

Tabelle wird auf der nächsten Seite fortgesetzt.

B Verzeichnis der untersuchten Konstruktionen

ID	Kategorie	Bezeichnung
1035	Reduplikation_Gleichstellung	von_N1_zu_N1
1043	Spezifizierung_Entityt	Satzwertige_Partizipphrase
1051	Intensivierung	ART_EN_ART_NP
1054	Intensivierung	ART_HERRSCHERTITEL_ART_NP
1126	Referenz_Altersgruppe	Generation_XY
1134	Additiv_Koordination	sowohl_X_als_auch_Y
1140	Tautologie_Hendiadyoin	XP1_und_XP2
1162	Tautologie	X1_oder_nicht_X1
1219	Optativ_Konjunktiv	V.conj.-II
1257	Tautologie_Pleonasmus	X1_X2
1289	Tautologie_Reduplikation	X1_X1_sein_lassen
1291	Optativ	Wenn_V.conj.-II
1300	Prototyp	so_geht_NP
1301	Expressivitt	Der_Moment_wenn_S
1313	Kategorisierung_exklamativ	es_ist_zum_N
1315	Expressivitt	als_ob_S
1316	Kontrast_Situationswechsel	raus_aus_NP_rein_in_NP
1320	Diminutiv	N-lein
1322	Diminutiv	N-chen
1323	Intensivierung_Attribut_nachgestellt	NP_pur
1324	Intensivierung_Attribut_nachgestellt	NP_satt
1329	Generalisierung	jedermanns_N
1337	Diminutiv	X-i
1342	Prototyp_Attribut_nachgestellt	NP_par_excellence
1346	Prototyp_Attribut_nachgestellt	NP_wie_es_im_Buche_steh
1347	Intensivierung	zu_Tode_X
1351	Reduplikation_Irrelevanz	N1_hin_N1_her
1459	Kategorisierung_Trendwechsel	X_ist_das_was_frher_Y_war
1461	Kategorisierung	Das_Leben_ist_kein_NP
1462	Kategorisierung	NP_ist_kein_Ponyhof
1503	Intensivierung_Attribut_nachgestellt	NP_brutal
1509	Intensivierung	VP_bis_der_Arzt_kommt
1511	Irrelevanz	N1_hin_N2_her
1514	Diminutiv	X-eln
1517	Intensivierung_Attribut_nachgestellt	NP_INTENSIVIERUNG
1521	Diminutiv	ADJ-lich
1524	Diminutiv_Redundanz	X-x1-x2
1525	Expressivitt_Negation	kein_X_fr_niemand
1554	Intensivierung_Komparativ	Reden_ist_Silber_X_ist_Gold
1556	Prototyp_Attribut_nachgestellt	NP_schlechthin
1573	Reflexive_Bewegung	NP_VP_REFL_PP
1574	Reflexives_Partikelverb	NP_VP_REFL_PART
1582	Reflexiver_Weg	NP_VP_REFL_NP
1593	Zuordnung	EN_kann_N
1594	Prototyp_Kompositum	Muster-N
1597	Konkretisierung_exhaustiv	Klammern

Tabelle wird auf der nchsten Seite fortgesetzt.

ID	Kategorie	Bezeichnung
1600	Konkretisierung_exemplifizierend	Klammern
1602	Konkretisierung_restriktiv	Klammern
1621	Passe-partout-Kompositum	X-ding
1624	Koordination_morphologisch	Klammern
1629	Negation	von_wegen_X
1630	Konzessiv_adverbial	allen_X_zum_Trotz
1631	Direktiv	jetzt_heißt_es_X
1634	Reduplikation_Wortbildung_Ablautdoppelung	X-X
1636	Reduplikation_Wortbildung_Reimdoppelung	X-X
1637	Einordnung_Gattung	Klammern
1639	Einordnung_Nation	Klammern
1641	Einordnung_Urheberschaft	Klammern
1643	Entsprichung_numerisch	Klammern
1645	Zuordnung	Klammern
1649	Kategorisierung	lecker_N
1660	Passe-partout-Phrasem	X_Kopula_ART_Y-ding
1671	Prototyp	Inbegriff_NP.gen
1681	Prototyp	die_Mutter_aller_NP
1684	Prototyp_Kompositum	Standard-N
1690	Negation_Litotes	nicht_un-ADJ
1715	Kategorisierung	kein_N1_ist_auch_ein_N1
1738	Exklamativ	wie_ADJ_Kopula_PRON.dem
1756	Konjunktiv_Präteritumperfekt	V.conj-II_Partizip-II
1760	Korrelativer_Äquativ	so_ADV_VP_so_ADV_VP
1762	Komparativ_Präferenz	besser_XP_als_YP
1770	Tautologie_nostalgisch	als_NP1_noch_NP1_war
1772	Indirekte_Redewiedergabe	von_wegen_X
1777	Korrelativer_Äquativ	VP_in_dem_Maße_XP
1779	Tautologie	entweder_X_oder_nicht
1792	Kategorisierung_Negation	NegPRO_ist_ADJ
1802	Interrogativ_Vergewisserungssignal	XP_X
1831	Konzessiv	so_VP_VP
1835	Kategorisierung	kein_N1_ist_auch_ein_N2
1842	Personenreferenz_abwertend	x-ling
1846	Interrogativ	was_ist_mit_NP
1849	Frage_rhetorisch	NP_und_X
1881	Kategorisierung	X_ist_kein_Y
1884	Intensivierung_Komparativ	X_ist_Silber_Y_ist_Gold
1986	Relativierung	kaum_ADJ
1987	Intensivierung	sehr_V

Literaturverzeichnis

- AKTER, SYEDA NAHIDA, ZICHUN YU, AASHIQ MUHAMED, TIANYUE OU, ALEX BÄUERLE, ÁNGEL ALEXANDER CABRERA, KRISH DHOLAKIA, CHENYAN XIONG und GRAHAM NEUBIG (2023). *An In-Depth Look at Gemini's Language Abilities*. 24. Dez. 2023. URL: <http://arxiv.org/abs/2312.11444> (besucht am 18.01.2024).
- ALAMMAR, JAY (2018). *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*. 3. Dez. 2018. URL: <http://jalammar.github.io/illustrated-bert/> (besucht am 30.01.2024).
- (2020). *How GPT3 Works - Visualizations and Animations*. 27. Juli 2020. URL: <http://jalammar.github.io/how-gpt3-works-visualizations-animations/> (besucht am 30.01.2024).
- AMMAR, WALEED, GEORGE MULCAIRE, YULIA TSVETKOV, GUILLAUME LAMPLE, CHRIS DYER und NOAH A. SMITH (2016). *Massively Multilingual Word Embeddings*. 21. Mai 2016. URL: <http://arxiv.org/abs/1602.01925> (besucht am 03.02.2024).
- ANIKINA, TATIANA (2023). „Towards Efficient Dialogue Processing in the Emergency Response Domain“. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*. Toronto: Association for Computational Linguistics, 2023, S. 212–225. doi: [10.18653/v1/2023.acl-srw.31](https://doi.org/10.18653/v1/2023.acl-srw.31).
- BARONI, MARCO (2020). „Linguistic Generalization and Compositionality in Modern Artificial Neural Networks“. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 375.1791 (3. Feb. 2020). doi: [10.1098/rstb.2019.0307](https://doi.org/10.1098/rstb.2019.0307).
- BARTELD, FABIAN und ALEXANDER ZIEM (2020). „Construction Mining. Identifying Construction Candidates for the German Constructicon“. In: *Belgian Journal of Linguistics* 34 (31. Dez. 2020), S. 5–16. doi: [10.1075/bjnl.00030.bar](https://doi.org/10.1075/bjnl.00030.bar).
- BAYERISCHE STAATSBIBLIOTHEK (2023). *German BERT*. Hugging Face. 6. Sep. 2023. URL: <https://huggingface.co/dbmdz/bert-base-german-cased> (besucht am 31.01.2024).
- BENDER, EMILY M., TIMNIT GEBRU, ANGELINA McMILLAN-MAJOR und SHMARGARET SHMITCHELL (2021). „On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? “. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. New York, NY: Association for Computing Machinery, 2021, S. 610–623. doi: [10.1145/3442188.3445922](https://doi.org/10.1145/3442188.3445922).
- BORGES VÖLKER, EMANUEL, MAXIMILIAN WENDT, FELIX HENNIG und ARNE KÖHN (2019). „HDT-UD: A Very Large Universal Dependencies Treebank for German“. In: *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*. Paris: Association for Computational Linguistics, 2019, S. 46–57. doi: [10.18653/v1/W19-8006](https://doi.org/10.18653/v1/W19-8006).
- BROWN, TOM, BENJAMIN MANN, NICK RYDER et al. (2020). „Language Models are Few-Shot Learners“. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Hrsg. von H. LAROCHELLE, M. RANZATO, R. HADSELL, M. F. BALCAN und H. LIN. Bd. 33. Red Hook, NY: Curran Associates, 2020, S. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf (besucht am 05.02.2024).
- CHOMSKY, NOAM (1956). „Three Models for the Description of Language“. In: *IRE Transactions on Information Theory* 2.3 (Sep. 1956), S. 113–124. doi: [10.1109/TIT.1956.1056813](https://doi.org/10.1109/TIT.1956.1056813).

Literaturverzeichnis

- CHOMSKY, NOAM (2011). *Strukturen der Syntax*. Übers. von KLAUS P. LANGE. Reprint 2010. Janua Linguarum. Series Minor 182. Berlin: De Gruyter Mouton, 20. Apr. 2011. doi: 10.1515/9783110813623.
- CROFT, WILLIAM (2009). *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Reprint. Oxford: Oxford University Press, 2009. ISBN: 978-0-19-829954-7.
- DE MARNEFFE, MARIE-CATHERINE, CHRISTOPHER D. MANNING, JOAKIM NIVRE und DANIEL ZEMAN (2021). „Universal Dependencies“. In: *Computational Linguistics* 47.2 (20. Mai 2021), S. 255–308. doi: 10.1162/coli_a_00402.
- DEVLIN, JACOB, MING-WEI CHANG, KENTON LEE und KRISTINA TOUTANOVA (2019). „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“. In: *Proceedings of the 2019 Conference of the North*. Minneapolis, MN: Association for Computational Linguistics, 2019, S. 4171–4186. doi: 10.18653/v1/N19-1423.
- DUNN, JONATHAN (2017a). „Computational Learning of Construction Grammars“. In: *Language and Cognition* 9.2 (Juni 2017), S. 254–292. doi: 10.1017/langcog.2016.7.
- (2017b). „Learnability and falsifiability of Construction Grammars“. In: *Proceedings of the Linguistic Society of America* 2.1 (Juni 2017). doi: 10.3765/plsa.v2i0.4009.
- (2018a). „Finding variants for construction-based dialectometry: A corpus-based approach to regional CxGs“. In: *Cognitive Linguistics* 29.2 (25. Mai 2018), S. 275–311. doi: 10.1515/cog-2017-0029.
- (2018b). „Modeling the Complexity and Descriptive Adequacy of Construction Grammars“. In: (Jan. 2018). Publisher: University of Massachusetts Amherst. doi: 10.7275/R59P2ZTB.
- (2018c). „Multi-Unit Directional Measures of Association: Moving Beyond Pairs of Words“. In: *International Journal of Corpus Linguistics* 23.2 (5. Okt. 2018), S. 183–215. doi: 10.1075/ijcl.16098.dun.
- (2019). „Frequency vs. Association for Constraint Selection in Usage-Based Construction Grammar“. In: *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*. Hrsg. von EMMANUELE CHERSONI, CASSANDRA JACOBS, ALESSANDRO LENCI, TAL LINZEN, LAURENT PRÉVOT und ENRICO SANTUS. Minneapolis, MN: Association for Computational Linguistics, Juni 2019, S. 117–128. doi: 10.18653/v1/W19-2913.
- (2023a). „Computational Construction Grammar: A Usage-Based Approach“. Nov. 2023. URL: <https://jdunn.name/cxg/> (besucht am 30.01.2024).
- (2023b). „Exploring the Constructicon: Linguistic Analysis of a Computational CxG“. In: *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*. Hrsg. von CLAIRE BONIAL und HARISH TAYYAR MADABUSHI. Washington, D.C.: Association for Computational Linguistics, März 2023, S. 1–11. URL: <https://aclanthology.org/2023.cxgsnlp-1.1> (besucht am 26.09.2023).
- FILLMORE, CHARLES J. (1968). „The Case for Case“. In: *Universals in Linguistic Theory*. Hrsg. von EMMON BACH und ROBERT THOMAS HARMS. New York, NY; Chicago, IL; San Francisco, CA; Atlanta, GA; Dallas, TX; Montreal; Toronto; London: Holt, Rinehart, und Winston, 1968, S. 1–88. ISBN: 978-0-03-068935-2.
- FILLMORE, CHARLES J., RUSSELL R. LEE-GOLDMAN und RUSSELL RHOMIEUX (2012). „The FrameNet Constructicon“. In: *Sign-Based Construction Grammar*. Hrsg. von HANS C. BOAS und IVAN A. SAG. CSLI Lecture Notes 193. Stanford, CA: CSLI Publ., Center for the Study of Language und Information, 2012, S. 309–372. ISBN: 978-1-57586-629-1 978-1-57586-628-4.
- FIRTH, JOHN R. (1951). „Modes of Meaning“. In: Essays and Studies (1951). Hrsg. von THE ENGLISH ASSOCIATION.

- FORSBERG, MARKUS, RICHARD JOHANSSON, LINNÉA BÄCKSTRÖM, LARS BORIN, BENJAMIN LYNGFELT, JOEL OLOFSSON und JULIA PRENTICE (2014). „From Construction Candidates to Construction Entries: An Experiment Using Semi-automatic Methods for Identifying Constructions in Corpora“. In: *Constructions and Frames* 6.1 (2014), S. 114–135. ISSN: 1876-1933. DOI: 10.1075/cf.6.1.07for.
- GEFFROY, ANNIE, PIERRE LAFON, GILL SEIDEL und M. TOURNIER (1973). „Lexicometric Analysis of Co-Occurrences“. In: *Computer and Literary Studies*. Hrsg. von A. J.AITKEN, RICHARD W. BAILEY und N. HAMILTON-SMITH. Edinburgh: University Press, 1973, S. 113–133. ISBN: 0-85224-232-8.
- GEMINI TEAM und GOOGLE (2023). *Gemini: A Family of Highly Capable Multimodal Models*. Technical Report. Dez. 2023. URL: https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf (besucht am 18.01.2024).
- GLEASON JR., HENRY A. (1961). *An Introduction to Descriptive Linguistics*. Revised Edition. New York, NY; Chicago, IL; San Francisco, CA; Toronto; London: Holt, Rinehart und Winston, 1961.
- GOLDBERG, ADELE E. (2006). *Constructions at Work: The Nature of Generalization in Language*. Oxford Linguistics. Oxford; New York, NY: Oxford University Press, 2006. ISBN: 978-0-19-926852-8.
- (2013). „Constructionist Approaches“. In: *The Oxford Handbook of Construction Grammar*. Hrsg. von THOMAS HOFFMANN und GRAEME TROUSDALE. Oxford Handbooks in Linguistics. Oxford: Oxford University Press, 2013, S. 14–31. DOI: 10.1093/oxfordhb/9780195396683.013.0002.
- HARRIS, RANDY ALLEN (2021). *The Linguistics Wars. Chomsky, Lakoff, and the Battle over Deep Structure*. Second Edition. New York, NY: Oxford University Press, 11. Nov. 2021. DOI: 10.1093/oso/9780199740338.001.0001.
- HEINRICH-HEINE-UNIVERSITÄT DÜSSELDORF (2024). *FrameNet-Konstruktikon des Deutschen*. URL: <https://framenet-constructicon.hhu.de> (besucht am 18.01.2024).
- HILPERT, MARTIN (2014). *Construction Grammar and its Application to English*. Bearb. von HEINZ GIEGERICH. Edinburgh Textbooks on the English Language – Advanced. Edinburgh: University Press, 2014. ISBN: 978-0-7486-7586-9.
- HOFFMANN, THOMAS und GRAEME TROUSDALE (2013). „Construction Grammar: Introduction“. In: *The Oxford Handbook of Construction Grammar*. Hrsg. von THOMAS HOFFMANN und GRAEME TROUSDALE. Oxford Handbooks in Linguistics. Oxford: Oxford University Press, 2013, S. 1–12. DOI: 10.1093/oxfordhb/9780195396683.013.0001.
- HONNIBAL, MATTHEW, INES MONTANI, SOFIE VAN LANDEGHEM und ADRIANE BOYD (2020). *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.1212303.
- HUNSTON, SUSAN und GILL FRANCIS (2000). *Pattern Grammar. A Corpus-Driven Approach to the Lexical Grammar of English*. Bearb. von ELENA TOGNINI-BONELLI und WOLFGANG TEUBERT. Studies in Corpus Linguistics 4. Amsterdam; Philadelphia, PA: John Benjamins Publishing Company, 2000. ISBN: 978-1-55619-399-6.
- JOBBINS, TOM (2023). *Llama 2 13B German Assistant v4 - GPTQ*. Hugging Face. 27. Sep. 2023. URL: <https://huggingface.co/TheBloke/Llama-2-13B-German-Assistant-v4-GPTQ> (besucht am 31.01.2024).
- JURAFSKY, DANIEL und JAMES MARTIN (2023). „Speech and Language Processing“. 3rd Ed. Draft. 7. Jan. 2023. URL: <https://web.stanford.edu/~jurafsky/slp3/>.
- KARIDI, TAE LIN, YICHIU ZHOU, NATHAN SCHNEIDER, OMRI ABEND und VIVEK SRIKUMAR (2021). „Putting Words in BERT’s Mouth: Navigating Contextualized Vector Spaces with Pseudowords“. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural*

Literaturverzeichnis

- Language Processing*. Online; Punta Cana: Association for Computational Linguistics, 2021, S. 10300–10313. doi: 10.18653/v1/2021.emnlp-main.806.
- KIRCHNER, JAN HENDRIK, LAMA AHMAD, SCOTT AARONSON und JAN LEIKE (2023). *New AI Classifier for Indicating AI-Written Text*. 20. Juli 2023. URL: <https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text> (besucht am 19.01.2024).
- KITA, KENJI, YASUHIKO KATO, TAKASHI OMOTO und YONEO YANO (1994). „A Comparative Study of Automatic Extraction of Collocations from Corpora: Mutual Information vs. Cost Criteria“. In: *Journal of Natural Language Processing* 1.1 (1994), S. 21–33. doi: 10.5715/jnlp.1.21.
- LAFON, PIERRE (1984). *Dépouillements et Statistique en Lexicométrie*. Bd. 24. Travaux de Linguistique Quantitative. Genf, Paris: Slatkine-Champion, 1984. ISBN: 2-05-100613-X.
- LAMPLE, GUILLAUME, ALEXIS CONNEAU, MARC’AURELIO RANZATO, LUDOVIC DENOYER und HERVÉ JÉGOU (2018). „Word Translation Without Parallel Data“. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=H196sainb> (besucht am 05.02.2024).
- LAN, ZHENZHONG, MINGDA CHEN, SEBASTIAN GOODMAN, KEVIN GIMPEL, PIYUSH SHARMA und RADU SORICUT (2020). „ALBERT: A Lite BERT for Self-supervised Learning of Language Representations“. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=H1eA7AEtvS> (besucht am 02.02.2024).
- LANGACKER, RONALD W. (2008). *Cognitive Grammar: A Basic Introduction*. New York, NY: Oxford University Press, 2008. ISBN: 978-0-19-804419-2.
- LEWIS, MIKE, YINHAN LIU, NAMAN GOYAL, MARJAN GHAVVININEJAD, ABDELRAHMAN MOHAMED, OMER LEVY, VESELIN STOYANOV und LUKE ZETTLEMOYER (2020). „BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension“. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, S. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.
- LI, XIAOYA, JINGRONG FENG, YUXIAN MENG, QINGHONG HAN, FEI WU und JIWEI LI (2020). „A Unified MRC Framework for Named Entity Recognition“. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Hrsg. von DAN JURAFSKY, JOYCE CHAI, NATALIE SCHLUTER und JOEL TETREAULT. Online: Association for Computational Linguistics, Juli 2020, S. 5849–5859. doi: 10.18653/v1/2020.acl-main.519.
- LIU, YIHONG, HAOTIAN YE, LEONIE WEISSWEILER, RENHAO PEI und HINRICH SCHÜTZE (2023). „Crosslingual Transfer Learning for Low-Resource Languages Based on Multilingual Copelexification Graphs“. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Hrsg. von HOUDA BOUAMOR, JUAN PINO und KALIKA BALI. Singapur: Association for Computational Linguistics, Dez. 2023, S. 8376–8401. doi: 10.18653/v1/2023.findings-emnlp.562.
- LIU, YINHAN, MYLE OTT, NAMAN GOYAL, JINGFEI DU, MANDAR JOSHI, DANQI CHEN, OMER LEVY, MIKE LEWIS, LUKE ZETTLEMOYER und VESELIN STOYANOV (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 26. Juli 2019. URL: <http://arxiv.org/abs/1907.11692> (besucht am 02.02.2024).
- MAO, RUI, GUANYI CHEN, XULANG ZHANG, FRANK GUERIN und ERIK CAMBRIA (2023). *GPTEval: A Survey on Assessments of ChatGPT and GPT-4*. 23. Aug. 2023. URL: <http://arxiv.org/abs/2308.12488> (besucht am 03.02.2024).
- MISRA, KANISHKA (2023). *AANN-Detector*. Hugging Face. 12. Nov. 2023. URL: <https://huggingface.co/kanishka/aann-detector> (besucht am 09.01.2024).
- OPENAI (2023). *GPT-4 Technical Report*. 18. Dez. 2023. URL: <http://arxiv.org/abs/2303.08774> (besucht am 04.02.2024).

- POTTS, CHRISTOPHER (2024). *Characterizing English Preposing in PP Constructions*. lingbuzz/007495. LingBuzz. URL: <https://lingbuzz.net/lingbuzz/007495> (besucht am 09. 01. 2024).
- RADFORD, ALEC, KARTHIK NARASIMHAN, TIM SALIMANS und ILYA SUTSKEVER (2018). „Improving Language Understanding by Generative Pre-Training“. In: (2018). Hrsg. von OPENAI. Publisher: OpenAI. URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (besucht am 25. 11. 2023).
- RAJPURKAR, PRANAV, ROBIN JIA und PERCY LIANG (2018). „Know What You Don't Know: Unanswerable Questions for SQuAD“. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Hrsg. von IRYNA GUREVYCH und YUSUKE MIYAO. Melbourne: Association for Computational Linguistics, Juli 2018, S. 784–789. doi: [10.18653/v1/P18-2124](https://doi.org/10.18653/v1/P18-2124).
- RAMBELLI, GIULIA, EMMANUELE CHERONI, PHILIPPE BLACHE, CHU-REN HUANG und ALESSANDRO LENCI (2019). „Distributional Semantics Meets Construction Grammar. Towards a Unified Usage-Based Model of Grammar and Meaning“. In: *Proceedings of the First International Workshop on Designing Meaning Representations*. Hrsg. von NIANWEN XUE, WILLIAM CROFT, JAN HAJIC, CHU-REN HUANG, STEPHAN OEPEN, MARTHA PALMER und JAMES PUSTEJOVKSÝ. Florenz: Association for Computational Linguistics, Aug. 2019, S. 110–120. doi: [10.18653/v1/W19-3312](https://doi.org/10.18653/v1/W19-3312).
- SANH, VICTOR, LYSANDRE DEBUT, JULIEN CHAUMOND und THOMAS WOLF (2020). *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter*. 29. Feb. 2020. URL: <http://arxiv.org/abs/1910.01108> (besucht am 02. 02. 2024).
- SCHÜTZE, HINRICH (1998). „Automatic Word Sense Discrimination“. In: *Computational Linguistics* 24.1 (1998). Hrsg. von JULIA HIRSCHBERG, S. 97–123. URL: <https://aclanthology.org/J98-1004>.
- SHLAIN, MICAH, HILLEL TAUB-TABIB, SHOVAL SADDE und YOAV GOLDBERG (2020). „Syntactic Search by Example“. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, 2020, S. 17–23. doi: [10.18653/v1/2020.acl-demos.3](https://doi.org/10.18653/v1/2020.acl-demos.3).
- SINCLAIR, JOHN McHARDY (1991). *Corpus, Concordance, Collocation*. Bearb. von JOHN McHARDY SINCLAIR und RONALD CARTER. Describing English Language. Oxford: Oxford University Press, 1991. ISBN: 978-0-19-437144-5.
- SRIVASTAVA, AAROHI, ABHINAV RASTOGI, ABHISHEK RAO et al. (2023). *Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models*. 12. Juni 2023. URL: <http://arxiv.org/abs/2206.04615> (besucht am 26. 09. 2023).
- TANG, YUQING, CHAU TRAN, XIAN LI, PENG-JEN CHEN, NAMAN GOYAL, VISHRAV CHAUDHARY, JIATAO GU und ANGELA FAN (2021). „Multilingual Translation from Denoising Pre-Training“. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 2021, S. 3450–3466. doi: [10.18653/v1/2021.findings-acl.304](https://doi.org/10.18653/v1/2021.findings-acl.304).
- TORRENT, TIAGO TIMPONI, THOMAS HOFFMANN, ARTHUR LORENZI ALMEIDA und MARK TURNER (2023). *Copilots for Linguists: AI, Constructions, and Frames*. Elements in Construction Grammar. Cambridge University Press, 31. Dez. 2023. doi: [10.1017/9781009439190](https://doi.org/10.1017/9781009439190).
- TOUVRON, HUGO, THIBAUT LAVRIL, GAUTIER IZACARD et al. (2023a). *LLaMA: Open and Efficient Foundation Language Models*. 27. Feb. 2023. URL: <http://arxiv.org/abs/2302.13971> (besucht am 02. 02. 2024).
- TOUVRON, HUGO, LOUIS MARTIN, KEVIN STONE et al. (2023b). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 19. Juli 2023. URL: <http://arxiv.org/abs/2307.09288> (besucht am 18. 01. 2024).

Literaturverzeichnis

- TURING, ALAN M. (1950). „I.—Computing Machinery and Intelligence“. In: *Mind* LIX.236 (Okt. 1950). Hrsg. von GILBERT RYLE, S. 433–460. ISSN: 0026-4423. DOI: 10.1093/mind/LIX.236.433.
- VASWANI, ASHISH, NOAM SHAZEE, NIKI PARMAR, JAKOB USZKOREIT, LLION JONES, AIDAN N. GOMEZ, ŁUKASZ KAISER und ILLIA POLOSUKHIN (2017). „Attention is all you Need“. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY: Curran Associates, 2017, S. 6000–6010. ISBN: 978-1-5108-6096-4.
- VON PLATEN, PATRICK (2020). *Transformer-Based Encoder-Decoder Models*. Hugging Face – Blog. 10. Okt. 2020. URL: <https://huggingface.co/blog/encoder-decoder> (besucht am 30.01.2024).
- WEISSWEILER, LEONIE, TAIQI HE, NAOKI OTANI, DAVID R. MORTENSEN, LORI LEVIN und HINRICH SCHÜTZE (2023). „Construction Grammar Provides Unique Insight into Neural Language Models“. In: *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*. Hrsg. von CLAIRE BONIAL und HARISH TAYYAR MADABUSHI. Washington, D.C.: Association for Computational Linguistics, März 2023, S. 85–95. URL: <https://aclanthology.org/2023.cxgsnlp-1.10> (besucht am 02.02.2023).
- WEISSWEILER, LEONIE, VALENTIN HOFMANN, ABDULLATIF KÖKSAL und HINRICH SCHÜTZE (2022). „The Better Your Syntax, The Better Your Semantics? Probing Pretrained Language Models for the English Comparative Correlative“. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi: Association for Computational Linguistics, 2022, S. 10859–10882. DOI: 10.18653/v1/2022.emnlp-main.746.
- WU, WEIQI, CHENGYUE JIANG, YONG JIANG, PENGJUN XIE und KEWEI TU (2023). „Do PLMs Know and Understand Ontological Knowledge?“ In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto: Association for Computational Linguistics, 2023, S. 3080–3101. DOI: 10.18653/v1/2023.acl-long.173.
- XU, LVXIAOWEI, JIANWANG WU, JIAWEI PENG, ZHILIN GONG, MING CAI und TIANXIANG WANG (2023). „Enhancing Language Representation with Constructional Information for Natural Language Understanding“. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto: Association for Computational Linguistics, 2023, S. 4685–4705. DOI: 10.18653/v1/2023.acl-long.258.
- ZAMFIRESCU-PEREIRA, J.D., RICHMOND Y. WONG, BJOERN HARTMANN und QIAN YANG (2023). „Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts“. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. New York, NY: Association for Computing Machinery, 2023. DOI: 10.1145/3544548.3581388.
- ZIEM, ALEXANDER (2014a). „Konstruktionsgrammatische Konzepte eines Konstruktikons“. In: *Grammatik als Netzwerk von Konstruktionen: Sprachwissen im Fokus der Konstruktionsgrammatik*. Hrsg. von ALEXANDER LASCH und ALEXANDER ZIEM. Sprache und Wissen 15. Berlin: De Gruyter, 8. Mai 2014. DOI: 10.1515/9783110353693.15.
- (2014b). „Von der Kasusgrammatik zum FrameNet: Frames, Konstruktionen und die Idee eines Konstruktikons“. In: *Grammatik als Netzwerk von Konstruktionen: Sprachwissen im Fokus der Konstruktionsgrammatik*. Hrsg. von ALEXANDER LASCH und ALEXANDER ZIEM. Sprache und Wissen 15. Berlin: De Gruyter, 8. Mai 2014, S. 263–290. DOI: 10.1515/9783110353693.263.