RATTE 2

Regensburger Analysetool für Texte

Dokumentation

22.04.2021

Johannes Wild Johannes.Wild@ur.de
Markus Pissarek Markus.Pissarek@uni-passau.de

Inhalt

Hinweis: **Die Version 2.0 von RATTE nutzt natural language processing und setzt dafür R und die zitierten R-Pakete ein.** Zuverlässigere Ergebnisse erhält man, wenn eine korrekte Interpunktion der analysierten Texte vorliegt. Das Programm ist online verfügbar unter: ratte.lesedidaktik.net

Wörter

Wörter zählen

Ausgangspunkt ist das syntaktische Wort: Es werden die Wörter in der genauen Form wie sie im Text erscheinen, d.h. alle Wortformen, bei der Wörterzählung berücksichtigt. Für die Analyse werden die R-Packages openNLP 0.2.7 und die Bibliothek openNLPmodels.de genutzt.

Hinweis: Zur Hervorhebung seltener Wörter wird der childLex-Korpus genutzt (Schröder, Würzner, Heister, Geyken & Kliegl, 2015). Die dort ausgewiesenen Altersgruppen werden in RATTE gemäß der Jahrgangsstufe ausgewählt (Jgst. 1-7[1]) und die 25% seltensten Wörter werden im Text markiert. (Basis: lemma.norm = normalized lemma frequency per million i.e. *absolute frequency / (sum(absolute frequency)/1000000)*). Bei lexikalischen Dubletten (type) wird der Mittelwert gebildet. Wörter, die nicht im Korpus vorkommen, werden ebenfalls als selten markiert.

Wortlänge: Buchstaben

Die Wortlänge errechnet sich aus der durchschnittlichen Zahl der Buchstaben in einem Wort (s.o.), d.h. keine Grapheme.

$$\text{Wortlänge} = \frac{\text{Summe Buchstaben aller Wörter}}{\text{Zahl der Wörter im Text}}$$

Wortlänge: Silben

Die Erfassung der durchschnittlichen Wortlänge in Silben erfolgt mit Hilfe der R-Packages quanteda 2.1.2 und quanteda.textstats 0.92. Die Silbenzahl einzelner Wörter wird durch quanteda.textstats erfasst.

[1] In childLEx 6-8 Jahre (1.-2. Klasse), 9-10 Jahre (3.-4. Klasse) und 11-12 Jahre (5.-6. Klasse).

Wortarten

Für die Analyse werden die R-Packages openNLP 0.2.7 und die Bibliothek openNLPmodels.de genutzt. Das Modell wurde mit dem STTS-Tagger (Stuttgart-Tübingen-TagSet) trainiert.

ADJA = Attributives Adjektiv,
ADJD = Adverbiales oder prädikatives Adjektiv,
ADV = Adverb,
APPR = Präposition; Zirkumposition links,
APPRART = Präposition mit Artikel,
APPO = Postposition,
APZR = Zirkumposition rechts,
ART = Bestimmter oder unbestimmter Artikel,
CARD = Kardinalzahl,
FM = Fremdsprachlichles Material,
ITJ = Interjektion,
KOUI = unterordnende Konjunktion mit zu und Infinitiv,
KOUS = unterordnende Konjunktion mit Satz,
KON = nebenordnende Konjunktion,
KOKOM = Vergleichskonjunktion,
NN = normales Nomen,
NE = Eigennamen,
PDS = substituierendes Demonstrativpronomen,
PDAT = attribuierendes Demonstrativpronomen,
PIS = substituierendes Indefinitpronomen,
PIAT = attribuierendes Indefinitpronomen ohne Determiner,
PIDAT = attribuierendes Indefinitpronomen mit Determiner,
PPER = irreflexives Personalpronomen,
PPOSS = substituierendes Possessivpronomen,
PPOSAT = attribuierendes Possessivpronomen,
PRELS = substituierendes Relativpronomen,
PRELAT = attribuierendes Relativpronomen,
PRF = reflexives Personalpronomen,
PROAV = Pronominaladverb,
PWS = substituierendes Interrogativpronomen,
PWAT = attribuierendes Interrogativpronomen,
PWAV = adverbiales Interrogativ- oder Relativpronomen,
PAV = Pronominaladverb,
PTKZU = zu vor Infinitiv,
PTKNEG = Negationspartikel,
PTKVZ = abgetrennter Verbzusatz,
PTKANT = Antwortpartikel,
PTKA = Partikel bei Adjektiv oder Adverb,
TRUNC = Kompositions-Erstglied,
VVFIN = finites Verb, voll,
VVIMP = Imperativ, voll,
VVINF = Infinitiv,
VVIZU = Infinitiv mit zu,
VVPP = Partizip Perfekt,
VAFIN = finites Verb, aux,
VAIMP = Imperativ, aux,
VAINF = Infinitiv, aux,
VAPP = Partizip Perfekt,
VMFIN = finites Verb, modal,
VMINF = Infinitiv, modal,
VMPP = Partizip Perfekt, modal,
XY = Nichtwort, Sonderzeichen,
UNDEFINED = Nicht definiert, z. B. Satzzeichen (gefiltert)

(vgl. https://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/germantagsets/ und https://www.ims.uni-stuttgart.de/documents/ressourcen/lexika/tagsets/stts-1999.pdf)

Sätze

Sätze zählen

Die Zählung von Sätzen folgt der pragmatischen Duden-Definition: „Ein Satz ist eine abgeschlossene Einheit, die nach den Regeln der Syntax gebildet worden ist“ (Gallmann 2005, 774). Erfasst werden somit sowohl einfache Sätze als auch Satzgefüge. Für die Analyse werden die R-Packages openNLP 0.2.7 und die Bibliothek openNLPmodels.de genutzt. Das Modell wurde mit dem SST-Tagger (Stanford Part-Of-Speech Tagger) trainiert. Aus den entsprechenden SST-Tags wird die Zahl der Sätze berechnet.

Satzlänge (Wörter)

Die durchschnittliche Satzlänge errechnet sich aus der Gesamtzahl der Wörter in einem Text im Verhältnis zur Zahl der Sätze im Text. Dafür werden die R-Packages quanteda 2.1.2 und quanteda.textstats 0.92 genutzt (s.o.).

Satzlänge (Silben)

Die Satzlänge errechnet sich als durchschnittliche Zahl der Silben in einem Wort.

$$\text{Satzlänge in Silben} = \frac{\text{Summe Silben aller Wörter}}{\text{Zahl der Sätze im Text}}$$

Nebensätze

Als „Sätze mit Nebensätzen" werden Satzgefüge bezeichnet. Das Programm berechnet die Summe aller Sätze, die Satzgefüge sind. (D.h. es zählt nicht die Zahl der Nebensätze in einem Text.) Analysiert werden Sätze in Hinblick auf das Vorliegen eingeleiteter Nebensätze in Vor- bzw. Nachstellung. Es kann sich hierbei um Objektsätze, Attributsätze und Adverbialsätze handeln (vgl. Hoffmann 2014, 70). Über das openNLP-Package (0.2.7) werden die SST-Tags "*KOUS*", "*KOUI"* und *„PWAV"* herangezogen (vgl. zuvor).

Pronominalisierungsindex (ProNIndex)

Berechnet das Verhältnis von Substantiven zu Proformen, um einen Indikator für den Vernetzungsgrad des Textes zu erhalten:

$$ProNIndex = \frac{Zahl\ der\ Pronomen\ im\ Text}{Zahl\ der\ Substantive\ im\ Text}$$

Mit Hilfe des openNLP-Package (0.2.7) werden für die Substantivklassifikation die SST-Tags "*NE*" und "*NN*", für die Proformklassifikation die Tags "*PDS*", "*PIS*", "*PPER*", "*PPOSS*", "*PRELS*", "*PRF*", "*PWS*" herangezogen (vgl. zuvor).

Type-Token

Die Type-Token-Ratio wird zur Messung der Beziehung zwischen der Anzahl der Types (= unterschiedliche Wörter) zu der Anzahl der Tokens (= Gesamtzahl der Wörter im Text) herangezogen. Grundsätzlich bestehen zwei Möglichkeiten, die Zahl der Types in einem Text zu bestimmen: (i) Die Betrachtung verschiedener Wortformen ohne Berücksichtigung des Konzeptes von Lexem/Lemma. Die Types geben hier eher die Formenvielfalt eines Textes an. (ii) Die Betrachtung verschiedener Lexeme. Hierzu ist der Abgleich mit Lexemlisten nötig, da Präfixe und Suffixe abgetrennt werden müssen. Ratte folgt dem zweiten Konzept. Die Berechnung von TTR und MATTR erfolgt mit den R-Packages quanteda 2.1.2 und quanteda.textstats 0.92.

$$TTR = \frac{Gesamtzahl\ Types}{Gesamtzahl\ Token} * 100\%$$

MATTR nutzt ein „moving window“ und ist daher von der Textlänge unabhängiger: „We choose a window length (say 500 words) and then compute the TTR for words 1–500, then for words 2–501, then 3–502, and so on to the end of the text. The mean of all these TTRs is a measure of the lexical diversity of the entire text and is not affected by text length nor by any statistical assumptions. Further, the individual TTRs can be compared in order to detect changes within the text.” (Covington & McFall, 2010, 96). In RATTE wird ein Fenster von 50 Wörtern angesetzt, liegt die Gesamttextlänge darunter, wird dieses auf 1 gesetzt.

Lesbarkeitsindizes

Lesbarkeitsindizes gehen auf die Idee zurück, dass sprachliche Schwierigkeiten eines Textes auch Indikator für inhaltliche Schwierigkeiten sind. Die Berechnung dieser Indizes beruht auf sprachstatistischen Verfahren und umfasst „alle Merkmale eines Textes, die es einer bestimmten Gruppe von Lesern erleichtern, den Sinn zu verstehen." (Bamberger, 2006, 285.) Stilistische, semantische Kriterien oder strukturelle Aspekte eines Textes sowie des Layouts/Drucks werden i. d. R. nicht erfasst (vgl. Bamberger, 2006, 285). Demnach können auch inhaltlich anspruchsvolle Texte den Lesbarkeitsindex eines einfachen Textes erhalten (z.B. Kafka). Die Lesbarkeit wird erschwert durch Erhöhen der: Zahl der schwierigen Wörter, Variation in der Lexik, durchschnittlichen Satzlänge oder Präpositionalphrasen (vgl. Spiegel & Campbell, 1985, 4). Mit Ausnahme des LIX werden die Indizes mit den R-Packages quanteda 2.1.2 und quanteda.textstats 0.92 berechnet. Die Ampel richtet sich nach dem gSmog: Liegt der Wert unter der angegebenen Jahrgangsstufe wird diese grün, liegt der Wert in der Zone der nächsten Entwicklung (Jgst. + 1) ist die Ampel gelb. Ist der Wert größer, wird diese rot.

Simple Measure of Gobbledygook – german (gSmog)

Bamberger passte die ursprüngliche Formel von McLaughlin für den deutschsprachigen Raum an. Die Formel setzt die Zahl der mehrsilbigen (drei oder mehr, s.o.) Wörter ins Verhältnis zur Zahl der Sätze im ganzen Text. Da sich die ursprüngliche Formel auf eine Stichprobe von 30 Sätzen bezieht, muss sich diesbezüglich angepasst werden:

$$\mathrm{gSmog} = \sqrt{\frac{\text{Wörter mit drei oder mehr Silben} * 30}{\text{Zahl der Sätze}}} - 2$$

Das Resultat ergibt näherungsweise das Lesealter (in Schulstufen), für das der Text geeignet ist. (Vgl. Bamberger, 1984, 58f.)

Lesbarkeitsindex (LIX)

Der LIX (Lesbarkeitsindex) berechnet sich aus der durchschnittlichen Satzlänge und dem prozentualen Anteil langer Wörter. Er wurde ursprünglich von dem Schweden Björnsson entwickelt und nimmt Werte zwischen 15 und 80 an.

$$\mathrm{LIX} = \frac{\text{Zahl der Wörter}}{\text{Zahl der Sätze}} + 100 * \frac{\text{Zahl der Wörter mit mehr als sechs Buchstaben}}{\text{Zahl der Wörter}}$$

Interpretation der Werte:

	LIX	Jgst. nach Bamberger	Jgst.
sehr leicht	15+	1-2	2
leicht	30+	3-8	3-4
mittel	40+		5-7
schwer	50+		8-9
sehr schwer	60+		10-12

Vgl. Anderson, 1981, 13 und Bamberger, 2006, 286.

Flesch (Kincaid)

Der Flesch-Index geht von dem Satz als kritische Einheit aus. Längere Sätze erfordern es, mehr Informationen im Gedächtnis zu halten, gleiches gilt für die Decodierung langer Wörter. (Vgl. Flesch, 2016.) Der Flesch-Kincaid-Index berechnet sich aus der durchschnittlichen Satz- und Wortlänge.

$$\text{Flesch} = 0{,}39 * \frac{\text{Zahl der Wörter}}{\text{Zahl der Sätze}} + 11{,}8 * \frac{\text{Silbenzahl im Text}}{\text{Zahl der Wörter}} - 15{,}59$$

Interpretation der Flesch-Werte:

Flesch	Jgst.	
90.0–100.0	5.	Very easy
80.0–90.0	6.	Easy
70.0–80.0	7.	Fairly easy
60.0–70.0	8.-9.	Plain English
50.0–60.0	10.-12.	Fairly difficult
30.0–50.0	Abiturienten	Difficult
0.0–30.0	Hochschule	Very difficult

Vgl. Flesch, 2016.

Vierte Wiener Sachtextformel (WSTF)

Die (vierte) Wiener Sachtextformel ist ein Index, der durch Regressionsanalysen von „einigen hundert Jugendbüchern" (Bamberger, 2006, 285) gewonnen wurde.

$$\text{WSTF} = 0{,}2656 * \frac{\text{Zahl der Wörter}}{\text{Zahl der Sätze}} + 0{,}2744 * \frac{\text{Zahl der Wörter mit mehr als drei Silben}}{\text{Zahl der Wörter}} * 100 - 1{,}693$$

Die Skala ist als Schulstufe zu interpretieren. Sie beginnt bei 4 und endet bei 15.

Lesedauer

Zu einer flüssigen Lektüre gehört das ausreichend schnelle, fehlerfreie Dekodieren von Wörtern. Erst dann kann einem Text Sinn entnommen werden. Die Anforderung an die Lesegeschwindigkeit liegt zwischen 150 Wörtern pro Minute (WpM) bei schwachen Lesern und kann bis zu 300-350 WpM bei geübten Lesern erreichen (vgl. Rosebrock & Nix, 2014, 38). Folgende Tabelle basiert auf einer US-amerikanischen Studie (Hasbrouck & Tindal, 2005) zur Lesegeschwindigkeit (Lautlesen) von über 15.000 Kindern und zeichnet ein differenziertes Bild.

Lesegeschwindigkeit in WpM für	**schwach (25%-Perzentil)**	**durchschnittlich (50%-Perzentil)**	**stark (75%-Perzentil)**
Ende**1. Klasse**	28	53	82
Beginn 2. Klasse	25	51	79
Beginn 3. Klasse	44	71	99
Beginn 4. Klasse	68	94	119
Beginn 5. Klasse	85	110	139
Beginn 6. Klasse	98	127	153
Beginn 7. Klasse	102	128	156
Beginn 8. Klasse	106	133	161
Ende**8. Klasse**	124	151	177

Vgl. Rosebrock, 2013, 131.

Im Schnitt nimmt die Lesegeschwindigkeit (ohne Berücksichtigung der Leseknicks, rote Markierung) pro Jahrgangsstufe um ca. 20 WpM zu, sodass die durchschnittliche Leseleistung näherungsweise aus der Jahrgangsstufe kalkuliert werden kann. Für das stille Lesen sind im Vergleich zu diesen Werten etwas höhere Werte anzunehmen. Im Umkehrschluss kann mit diesen Werten und der Zahl der Wörter eines Textes die benötigte Lesezeit berechnet werden:

$$\text{Lesezeit} = \frac{\text{Zahl der Wörter}}{47{,}612 * \ln(\text{Jahrgangsstufe}) + 45{,}011}$$

Hinweis: Die hier angezeigte Lesezeit impliziert nicht, dass die Kinder einen nicht-altersgemäßen Text lesen können.

Literatur

Anderson, N. (1981). *Analysing the Readability of English and Non-English Texts in the Classroom with Lix.* Darwin. http://files.eric.ed.gov/fulltext/ED207022.pdf

Bamberger, R. (2006). *Erfolgreiche Leseerziehung. Theorie und Praxis.* München: Domino.

Bamberger, R. & Vanecek, E. (1984). *Lesen-Verstehen-Lernen-Schreiben. Die Schwierigkeitsstufen von Texten in deutscher Sprache.* Wien: Jugend und Volk.

Covington, M.A. & McFall, J.D. (2010). Cutting the Gordian Knot: The Moving-Average Type-Token Ratio (MATTR) Journal of Quantitative Linguistics, 17(2), 94–100. doi: 10.1080/09296171003643098

Eisenberg, P. (2013a). *Grundriss der deutschen Grammatik (Band 1): Das Wort.* 4. Auflage. Stuttgart, Weimar: Metzler.

Eisenberg, P. (2013b): *Grundriss der deutschen Grammatik (Band 2): Der Satz.* 4. Auflage. Stuttgart, Weimar: Metzler.

Eisenberg, P. (2005). Phonem und Graphem. In Dudenredaktion (Hrsg.), *Duden. Die Grammatik* (19-94). 7. Auflage. Mannheim, Leipzig, Wien, Zürich: Dudenverlag 2005.

Flesch, R. (o.J.): How to Write Plain English. [unpag.]. http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml

Gallmann, P. (2005). Was ist ein Wort? In Dudenredaktion (Hrsg.), *Duden. Die Grammatik* (129-132). 7. Auflage. Mannheim, Leipzig, Wien, Zürich: Dudenverlag 2005.

Gallmann, P. (2005). Der Satz. In Dudenredaktion (Hrsg.), *Duden. Die Grammatik* (773-1066). 7. Auflage. Mannheim, Leipzig, Wien, Zürich: Dudenverlag 2005.

Hasbrouk, J. & Tindal, G. (2005). Oral Reading Fluency: 90 Years of Measurement. *Behavioral research & testing*, [unpag.]. http://files.eric.ed.gov/fulltext/ED531458.pdf

Hoffmann, L. (2014). *Deutsche Grammatik. Grundlagen für Lehrerausbildung, Schule, Deutsch als Zweitsprache und Deutsch als Fremdsprache*. 2. Auflage. Berlin: Erich Schmidt.

Popescu, I. (2009). *Word Frequency Studies.* Berlin u.a.: de Gruyter. http://evrika-braila.ro/wp-content/uploads/2014/10/Book1-Word-Frequency-Studies-June-2009.pdf

Rosebrock, C. (2013). Leseförderung aus systematischer Sicht: Dimensionen von Lesekompetenz und adaptive Förderverfahren.In F. Hellmich & K. Siekmann (Hrsg.), *Sprechen, Lesen und Schreibenlernen* (112-134) Berlin: DGLS.

Rosebrock, C. & Nix, D. (2014). *Grundlagen der Lesedidaktik und der systematischen schulischen Leseförderung*. Baltmannsweiler: Schneider.

Schröder, S., Würzner, K., Heister, J., Geyken, A. & Kliegl, R. (2015). childLex – Eine lexikalische Datenbank zur Schriftsprache für Kinder im Deutschen. *Psychologische Rundschau*, 66(3), 155–165. https://doi.org/10.1026/0033-3042/a000275 [Korpus: childlex.de]

Spiegel, G. & Campbell, J. (1985). Measuring Readability with a Computer: What We Can Learn. Los Angeles.

<u>Packages</u>

Attali, D. (2020). *shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds. R package version 2.0.0.* https://CRAN.R-project.org/package=shinyjs

Benoit, K., Watanabe, K.; Wang, H.; Nulty, P.; Obeng, A.; Müller, S. & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*. 3(30), 774.

Chang, W. (2021). *shinythemes: Themes for Shiny. R package version 1.2.0.* https://CRAN.R-project.org/package=shinythemes

Chang, W.; Cheng, J.; Allaire, J.; Sievert, C.; Schloerke, B.; Xie, Y.; Allen, J.; McPherson, j.; Dipert, A. & Borges, B. (2021*). shiny: Web Application Framework for R. R package version 1.6.0.* https://CRAN.R-project.org/package=shiny

Hornik, K. (2019). *openNLP: Apache OpenNLP Tools Interface. R package version 0.2-7*. https://CRAN.R-project.org/package=openNLP

Hornik, K. (2020). *NLP: Natural Language Processing Infrastructure. R package version 0.2-1.* https://CRAN.R-project.org/package=NLP

Lincoln, M. (2020). *clipr: Read and Write from the System Clipboard. R package version 0.7.1.* https://CRAN.R-project.org/package=clipr

R Core Team (2020). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*. Vienna, Austria. https://www.R-project.org/.

Rinker, T. W. (2018). *textreadr: Read Text Documents into R version 0.9.1.* http://github.com/trinker/textreadr

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag: New York.

Wickham, H. (2019). *stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0.* https://CRAN.R-project.org/package=stringr

Wickham, H.; François, R.; Henry, L. & Müller, K. (2021). *dplyr: A Grammar of Data Manipulation. R package version 1.0.4.* https://CRAN.R-project.org/package=dplyr