

Werkzeugunterstützung bei der Vermittlung der Grundlagen wissenschaftlichen Schreibens

Oliver Zscheyge, Karsten Weicker

Fakultät Informatik, Mathematik und Naturwissenschaften
Hochschule für Technik, Wirtschaft und Kultur Leipzig
Leipzig
Web: portal.imn.htwk-leipzig.de/fakultaet/personen/weicker
Email: karsten.weicker@htwk-leipzig.de

Zusammenfassung: Der Unterricht großer Studierenden-
gruppen im wissenschaftlichen Schreiben birgt vielfältige
organisatorische Herausforderungen und eine zeitintensive
Betreuung durch die Dozenten. Diese Arbeit stellt ein Lehr-
konzept mit Peer-Reviews vor, in dem das Feedback der Peers
durch eine automatisierte Analyse ergänzt wird. Die Software
Confopy liefert metrik- und strukturbasierte Hinweise für die
Verbesserung des wissenschaftlichen Schreibstils. Der Nutzen
von Confopy wird an 47 studentischen Arbeiten in Draft- und
Final-Version illustriert.

1 Motivation

Das Studium soll die Absolventen dazu befähigen, wissenschaftlich arbeiten zu können. Neben dem zwingend notwendigen fachlichen Wissen nimmt dabei die Fähigkeit, seine Ergebnisse adäquat und entsprechend der fachlichen Standards zu dokumentieren und zu publizieren, eine ebenso bedeutende Rolle ein. Wird dies im Studium erst im Rahmen von Studien-, Bachelor- oder Diplomarbeiten thematisiert, resultiert oft ein hoher individueller Betreuungsaufwand, wobei sich das Feedback vom einen zum nächsten Kandidaten gleicht. Daher wurde an der HTWK Leipzig ein Modul „Grundlagen wissenschaftlichen Arbeitens“ eingerichtet, in welchem zu einem frühen Zeitpunkt die Grundlagen und Gepflogenheiten präsentiert und die Nota-

tionen und Vorgehensweisen an kürzeren Texten geübt werden. Wie in allen Grundlagenveranstaltungen bedeutet dies einen hohen Betreuungsaufwand seitens der Lehrenden. Daher ist die Unterstützung durch den Einsatz von Werkzeugen sinnvoll.

Nach einem kurzen Abriss der Literatur in Abschnitt 2, behandelt Abschnitt 3 die Form der Lehrveranstaltung. Anschließend wird detaillierter die mögliche Unterstützung durch die Analysesoftware Confopy in Abschnitt 4 sowie konkrete Ergebnisse anhand der studentischen Texte der letzten beiden Jahre in Abschnitt 5 präsentiert.

2 Verwandte Arbeiten

In der Literatur gibt es zahlreiche Arbeiten, die sich mit der automatisierten Bewertung der sprachlichen und argumentativen Qualitäten von Schülern und Bachelorstudenten auseinandersetzen. [VNC03] liefern einen Überblick über die verschiedenen Ansätze der automatisierten Bewertung. [BB05] vergleicht in einer umfassenden Analyse die Techniken mit den Bewertungen durch Experten. Ein konkretes System, e-rater V.2, wird in [AB06] vorgestellt. Die Kriterien und Metriken dieser Arbeiten wurden weitestgehend im hier präsentierten System Confopy berücksichtigt, auch wenn sich die vorgenannten Arbeiten mehr auf allgemeine Essays als auf wissenschaftliche Schriften konzentrieren.

Unabhängig von der Automatisierbarkeit wird in [Tim11] ein allgemeiner Katalog an Kriterien erarbeitet, mit dem die Kompetenz des wissenschaftlichen Schreibens beurteilt werden kann.

Für eine tiefere inhaltliche Analyse, z. B. ob stichhaltig argumentiert wird, ist eine genauere semantische Analyse von Texten erforderlich. Anhaltspunkte können hier Forschungsergebnisse zur Textzusammenfassung und -extraktion liefern. Ein frühes Beispiel ist die Arbeit [TM02]. Diese Ansätze sind sehr aufwändig und werden aktuell in Confopy nicht benutzt.

3 Modul „Grundlagen wissenschaftlichen Arbeitens“

Das Modul „Grundlagen wissenschaftlichen Arbeitens“ findet im Bachelorstudium Informatik im 2. Fachsemester und im Bachelorstudium Medieninformatik im 3. Fachsemester statt. Dabei orientiert sich der Ablauf des praktischen Anteils an dem einer wissenschaftlichen Tagung: Die Studieren-

den verfassen ein Schriftstück und reichen dieses ein. Es folgt eine Review-Phase, in der die Studierenden drei Arbeiten von Kommilitonen begutachten und die Gutachten entsprechend abgeben. Jeder Autor erhält das Feedback der Peers und überarbeitet seinen Text, den er dann in der endgültigen Fassung abgibt. Anschließend halten die Studierenden einen Vortrag zu ihrem Thema. Diese Vorgehensweise ist verbreitet [Gui01,Wei06]. Für die Einführung in das wissenschaftliche Arbeiten finden begleitend Vorlesungen zu verschiedenen Themen statt. Abb. 1 zeigt den zeitlichen Ablauf des Moduls.

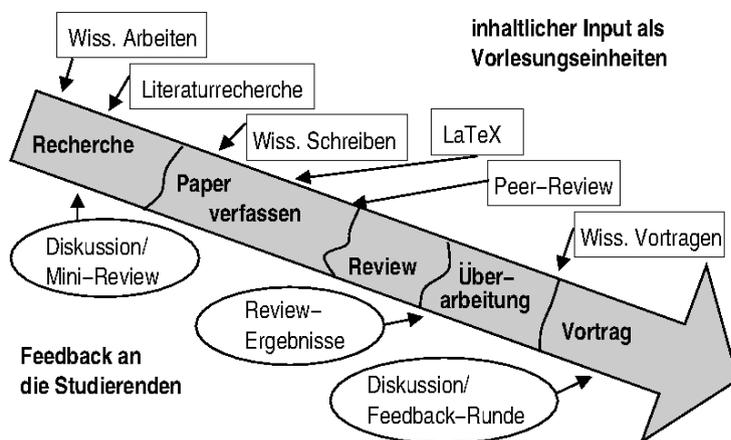


Abb. 1: Grundsätzlicher Ablauf der Lehrveranstaltung (nach [Wei06])

Die Themen umfassen einfache, für das erste Studienjahr geeignete Teilbereiche der Informatik, die zufällig verteilt werden. Die Benotung des Moduls erfolgt gemäß eines Punkteschemas, in welchem 60 Punkte erreichbar sind. Ab 24 Punkten ist der Kurs bestanden. Die Punkte werden für die folgenden Aspekte vergeben:

- Vortrag: 15 Punkte
- Ausarbeitung (Final-Version): 15 Punkte
- Überarbeitung der eigenen Ausarbeitung: 5 Punkte
- Gutachten zu den anderen Arbeiten: 15 Punkte
- aktive Beteiligung in den Seminaren: 10 Punkte

Die Lehrveranstaltung wird seit dem Sommersemester 2014 durchgeführt.

Die Studierenden bewerten insbesondere den praktischen Teil der Lehrveranstaltung als arbeitsintensiv. Durch das Konzept der Peer-Bewertung bleibt der Betreuungs- und Benotungsaufwand gering, da ein großer Teil des Feedbacks im Lernprozess zwischen den Studierenden stattfindet. Allerdings stellt dies auch eine große Schwachstelle dar, da von einigen studentischen Gutachtern wichtige Kriterien des wissenschaftlichen Schreibens ignoriert bzw. pauschale Bewertungen wie „Alles toll! Interessantes Thema!“ abgegeben werden.

4 Automatische Analyse wissenschaftlicher Texte

Um sicherzustellen, dass jeder Autor ein hilfreiches Feedback erhält, sollten die Gutachten um eine zusätzliche „unbestechliche“ Meinung ergänzt werden. Aufgrund der großen Menge der Arbeiten und der sehr kurzen Zeitspanne, die für die Begutachtung zur Verfügung steht, kann dies nicht durch die Lehrkräfte realisiert werden. Daher war die Idee entstanden, durch das neu entwickelte Werkzeug Confopy ein automatisierbares Feedback mit sinnvollen Hinweisen zur Wissenschaftlichkeit des Schreibstils zu liefern. Perspektivisch sollen Fachspezifika stärker berücksichtigt werden – insbesondere die mathematisch-formale Sprache der Informatik im Gegensatz zu Normen anderer Domänen wie Betriebswirtschaft oder Rechtswissenschaften. Der hier präsentierte Ansatz wurde erstmals in [Zsc15] vorgestellt.

4.1 Messbare Sprachmerkmale

Ein Text wird unter Zuhilfenahme des TIGER-Corpus [Bra04] als deutsche Sprachreferenz in Wort- und Satztoken zerlegt. Über das POS-Tagging (*Part-Of-Speech*) werden jedem Wort seine Art (Artikel, Substantiv, Verb etc.) sowie weitere Attribute zugeordnet. Auf dieser Basis können nun verschiedene Sprachmerkmale eines Textes bestimmt werden. Derzeit werden die folgenden allgemeinen Sprachmetriken erhoben:

- die *durchschnittliche Wortlänge* *wordlength*,
- die *Rechtschreibfehler* *spellcheck* als Anteil der Wörter im Text, die nicht als deutsche Wörter erkannt werden,
- der *relative Umfang des Wortschatzes* *lexicon* als das Verhältnis der im Text vorkommenden Wörter (in ihrer Grundform) zur Länge des Textes

- als Anzahl aller Wörter (z. B. würde „Buch der Bücher“ $\text{lexicon}=2/3$ aufweisen, da „Buch“ und „Bücher“ dieselbe Grundform besitzen) und
- die *Lesbarkeit* *ari* (*automated readability index*) nach [SS67] als 1:9-gewichtete Summe der durchschnittlichen Satz- und Wortlänge – $\text{ari}=50$ entspricht dem Niveau eines Viertklässlers, $\text{ari}=70$ ist Zeitungssprache.

Darüber hinaus werden die folgenden Metriken für Sprachmerkmale wissenschaftlicher Texte erhoben:

- drei Metriken bezüglich des *Schreibstils* als relative Häufigkeit verschiedener Wörter: persönlicher Stil („ich“, „wir“, „Sie“) *personalstyle*, unpersönlicher Stil („man“) *impersonalstyle* und Passiv-Konstrukte („wird“, „werden“) – der Wert sollte jeweils möglichst niedrig sein –,
- die *passende Zeitform* *simplepres* als Anteil der Verben, die im Präsens benutzt werden – der Wert sollte möglichst hoch sein –,
- *verstärkende bzw. unpräzise Adverbien* („leicht“, „sehr“, „viel“) *adverbmodifier* relativ zur Gesamtwortzahl – der Wert sollte möglichst niedrig sein –,
- *inaktive Verben* („gehören“, „liegen“, „beinhalten“) *deadverbs* relativ zur Anzahl der Sätze – je niedriger, desto besser –,
- *Füllwörter* („also“, „offenbar“, „bereits“) *fillers* relativ zur Gesamtwortzahl – je niedriger, desto besser –,
- *Beispielgehalt* *examplecount* als absolutes Vorkommen von Wörtern wie („Beispiel“, „beispielsweise“, „z. B.“) – je höher, desto besser –,
- die *Variation der Satzlänge* *sentlengthvar* als eine der wenigen Möglichkeiten, wissenschaftliche Texte abwechslungsreich zu gestalten (neben aktiven, anschaulichen Sätzen mit hohem Beispielgehalt), gemessen als durchschnittliche Differenz der Länge direkt aufeinander folgender Sätze im Text – je höher, desto besser.

Für all diese Merkmale wurden für Texte in der Größenordnung, wie wir sie in unserer Lehrveranstaltung zu erwarten haben, typische Wertebereiche ermittelt. Dies ermöglicht, für beliebige Texte die Metriken zu ermitteln und die Autoren auf Abweichungen hinzuweisen.

4.2 Strukturbezogene Regeln

Neben der rein metrikbasierten Auswertung eines Textes erlaubt die vorliegende Tokenisierung auch Aussagen über die Struktur eines Textes. In Confopy wurden zunächst beispielhaft drei Regeln formuliert:

- Kapitel sollen eine Einleitung haben und nicht direkt mit einem Unterkapitel starten.

$is_chapter(x) \rightarrow has_introduction(x)$

- Weist ein Abschnitt Unterabschnitte auf, so sollten dies mindestens zwei sein.

$(is_chapter(x) \text{ or } is_section(x)) \rightarrow |subsections(x)| <> 1$

- Gleitobjekte sind beschriftet und werden im Text vor der Abbildung referenziert.

$is_float(x) \rightarrow (was_referenced_before(x) \text{ and } has_caption(x))$

Hier sind noch weitere Regeln denkbar. Wenn man sich auf ein zahlbasiertes Zitierschema beschränkt, könnten auch Literaturquellen erkannt und deren Referenz im Text geprüft werden. Auch sind Aussagen über die in der Informatik oft unbeliebten Fußnoten denkbar – so könnten Fußnoten nur erlaubt werden, wenn sie eine URL enthalten, um beispielsweise auf Software-Werkzeuge hinzuweisen. In eher formalen Arbeiten können Definitionen erkannt und die Verwendung der definierten Begriffe geprüft werden. Auch Aussagen über die Verwendung von mathematischen Notationen und Formeln sind denkbar, bedürfen allerdings mehr Arbeit bzgl. der Tokenisierung über weitere Heuristiken.

4.3 Confopy

Die Software Confopy setzt alle in Abschnitt 4.2 aufgeführten Metriken und Regeln um. Sie kann einen Text in einem XML-Beschreibungsformat einlesen und analysieren. Es steht ebenfalls ein Prototyp zur Verfügung, der aus PDF-Dokumenten die XML-Beschreibung extrahiert – allerdings mit Einschränkungen, die dem Format PDF als Seitenlayoutbeschreibung und nicht als syntax- und semantikerhaltende Beschreibung geschuldet sind. Viele gängige insbesondere mit LaTeX erzeugte Dokumente werden im ein- und zweispaltigen Layout erkannt. Textblöcke wie Paragraphen, Überschriften, Fußnoten und Gleitobjektbeschriftungen sowie der Textfluss werden über Heuristiken ermittelt.

Confopy ist mit Python realisiert und es werden die Bibliotheken pdfminer, lxml, nltk (Natural Language Toolkit), pattern und PyEnchant benutzt (vgl. <https://pypi.python.org/pypi>).

Abb. 2 zeigt beispielhaft die Ausgabe von Confopy auf einer studentischen Arbeit, die dem Studierenden als Feedback neben Peer-Gutachten zur Verfügung gestellt werden kann.

```

## Metriken
* adverbmodifier 0.03 (erwartet: max. 0.04)
  OK!
* ari 62.66 (erwartet: max. 71.96)
  OK!
* deadverbs 0.09 (erwartet: max. 0.06)
  Versuche weniger tote Verben wie gehören, liegen, befinden, beinhalten, geben,
  bewirken etc. zu verwenden!
* examplecount 5.0 (erwartet: min. 1.00)
  OK!
* fillers 0.04 (erwartet: max. 0.03)
  Zu viele Füllwörter!
* impersonalstyle 0.15 (erwartet: max. 0.05)
  Zu viele Sätze mit ‚man‘.
* lexicon 0.44 (erwartet: zw. 0.46 und 0.56)
  Zu geringer Wortschatz
* passiveconstructs 0.3 (erwartet: max. 0.37)
  OK!
* personalstyle 0.09 (erwartet: max. 0.06)
  Zu persönlicher Schreibstil! Sätze mit ‚ich‘, ‚wir‘, ‚sie‘ umschreiben!
* sentlength 18.44 (erwartet: max. 17.38)
  Zu viele lange Sätze!
* sentlengthvar 11.19 (erwartet: min. 5.27)
  OK!
* simplepres 0.83 (erwartet: min. 0.74)
  OK!
* spellcheck 0.21 (erwartet: zw. 0.10 und 0.20)
  Entweder zu viele Rechtschreibfehler oder zu viele Fremdwörter!
* wordlength 4.41 (erwartet: max. 6.28)
  OK!

## Regeln
Kapitel "2. Aufbau der Programmiersprache" hat keine Einleitung!
Gleitobjekt "Tabelle 1: Die wichtigsten arithmetischen und logischen
Befehle in Forth" wird nicht im Text referenziert!
Gleitobjekt "Tabelle 1: Die wichtigsten arithmetischen und logischen
Befehle in Forth" wird nicht im vorstehenden Text referenziert!

```

Abb. 2: Beispiel eines durch Confopy erzeugten Berichts für den Autor

Confopy kann auch zwei Versionen derselben Arbeit vergleichen (siehe Abb. 3), z. B. im Rahmen der Notenfindung als Indikator für die Überarbeitung der Arbeit durch die Studentin.

PROGRESS: Vorher- --> Nachher-Wert.	
	(+) ... Erhöhung
	(-) ... Verringerung
	(=) ... gleichbleibend
METRIC	PROGRESS
-----+-----	
wordlength	04.41 --> 04.45 (+)
spellcheck	00.21 --> 00.16 (-)
lexicon	00.44 --> 00.49 (+)
sentlength	18.44 --> 17.88 (-)
ari	62.66 --> 62.74 (+)
personalstyle	00.09 --> 00.08 (-)
impersonalstyle	00.15 --> 00.06 (-)
passiveconstructs	00.30 --> 00.33 (+)
simplepres	00.83 --> 00.80 (-)
adverbmodifier	00.04 --> 00.04 (=)
deadverbs	00.09 --> 00.06 (-)
fillers	00.03 --> 00.04 (+)
examplecount	05.00 --> 06.00 (+)
sentlengthvar	11.19 --> 08.82 (-)

Abb. 3: Ausgabe von Confopy beim Vergleich der Metriken zweier Versionen derselben studentischen Arbeit

5 Ergebnisse

Dieser Abschnitt präsentiert die Ergebnisse von Confopy auf 47 studentischen Arbeiten, die jeweils in einer ersten eingereichten Version (Draft) und einer Version nach Begutachtung und Überarbeitung (Final) vorliegen. Da die Software in den vergangenen Lehrveranstaltungen noch nicht zur Verfügung stand, kann hier nicht der direkte Einsatz evaluiert werden. Stattdessen soll die Analyse der Ergebnisse zwei Thesen untermauern: (1) Confopy kann dem Autor sinnvolle Hinweise liefern und (2) in einer „Groß-

veranstaltung“ kann ein automatisiertes Feedback Peer-Reviews ergänzen und damit den Lernprozess gezielter beeinflussen.

Tab. 1 zeigt die zusammengefassten Messungen für die Draft- und Final-Versionen. Dies veranschaulicht einerseits, dass sich Texte aus Zeitungsartikeln (im TIGER-Corpus) von (studentischen) wissenschaftlichen Arbeiten messbar unterscheiden. An den markierten Werten liegt der Mittelwert aller Arbeiten in einem Bereich, in dem Hinweise an den Autor geliefert werden, was (1) unterstreicht. Die fehlende Entwicklung zwischen Draft- und Final-Version zeigt, dass das reine Feedback der Peers nicht ausreicht (2).

Metrik	Draft-Version		Final-Version		TIGER
	Mittelwert	Sdv	Mittelwert	Sdv	
wordlength	4,65	0,34	4,47	0,76	5,52
spellcheck	0,21	0,06	0,22	0,11	0,08
lexicon	0,45	0,09	0,44	0,11	0,11
sentlength	18,09	3,36	17,98	4,84	15,58
ari	64,78	3,66	63,08	10,45	70,51
personalstyle	0,06	0,07	0,05	0,06	0,1
impersonalstyle	0,07	0,09	0,05	0,07	0,02
passiveconstructs	0,07	0,09	0,3	0,15	0,1
simplepres	0,8	0,13	0,78	0,17	0,66
adverbmodifier	0,03	0,01	0,03	0,01	0,04
deadverbs	0,05	0,03	0,05	0,04	0,05
fillers	0,02	0,01	0,02	0,01	0,03
examplecount	2,96	2,55	2,81	2,61	293,0
sentlengthvar	9,85	3,58	10,17	3,8	9,09

Tab. 1: Durchschnittlich gemessene Metriken in den studentischen Arbeiten im Vergleich zum TIGER-Corpus. Werte, deren Mittelwert von der empirisch ermittelten Empfehlung abweicht, sind markiert.

Der Nutzen des Werkzeugs im Sinne von (1) wird auch in Tab. 2 deutlich, da fast jeder Autor durchschnittlich 3–4 konstruktive Hinweise erhalten hat, die aus Bereichsverletzungen abgeleitet wurden. Für etwa die Hälfte der Arbei-

ten konnten zusätzlich strukturelle Hinweise aufgrund von durchschnittlich 3–4 Regelverletzungen gegeben werden.

	Draft-Version		Final-Version	
	Anzahl	Verletzungen	Anzahl	Verletzungen
Arbeiten mit Bereichsverletzungen	46	3,5	47	3,51
Arbeiten mit Regelverletzungen	22	3,91	25	3,32

Tab. 2: Anzahl der Arbeiten mit Bereichs- und Regelverletzungen sowie die durchschnittliche Anzahl der Verletzungen pro Arbeit

Tab. 3 zeigt, wie sich die Metrikwerte in den einzelnen Arbeiten verändert haben. Dabei werden jeweils die Paper zusammengefasst, bei denen der Metrikwert größer bzw. kleiner wurde. Diese Tabelle demonstriert, dass die Metriken bei der Überarbeitung eines Papers eine Aussagekraft besitzen. Dort ist zu sehen, dass bei vielen Metriken insbesondere in der erwünschten Richtung im Durchschnitt erhebliche Veränderungen erreicht werden können. Beispielsweise bei der Lesbarkeit (ari) stellt eine durchschnittliche Verbesserung von 7,47 einen beachtlichen Qualitätsgewinn dar, wenn der obere akzeptierte Schwellwert bei 71,96 liegt – die Erwartungsbereiche können der Ausgabe in Abb. 2 entnommen werden. Auch bei der Verwendung der Passivkonstruktionen steht eine Verbesserung um 0,195 Punkte dem erlaubten Maximalwert von 0,37 gegenüber. In jeweils 12 Arbeiten wurden bei der Überarbeitung so durchschnittlich 1,833 Bereichsverletzungen und 3,5 Regelverletzungen entfernt.

6 Zusammenfassung und Ausblick

Die vorgestellte Software Confopy kann sinnvoll den Peer-Review im Rahmen der Lehre wissenschaftlichen Schreibens ergänzen, wie die präsentierten Messungen zeigen. Weitere Anpassungen für die Konventionen in Schriften der Informatik sind möglich und können in der näheren Zukunft umgesetzt werden. Eine genauere Analyse beispielsweise der Schlüssigkeit von Argumentation im Text kann über den präsentierten Ansatz nicht erfolgen.

Inwieweit das Feedback von den Studierenden als hilfreich empfunden wird und ob sich auf dieser Basis die Qualität der studentischen Arbeiten verbessern, kann nur über genauere empirische Studien in den nächsten Jahren ermittelt werden. In jedem Fall zeigen erste Rückmeldungen, dass die Metriken eine gewisse Selbstreflektion begünstigen.

Metrik	Erhöhung des Metrikwerts		Verringerung des Metrikwerts		Anzahl gleichbleibender Werte
	Anzahl	Delta	Anzahl	Delta	
wordlength	16	0,268	31	0,643	0
spellcheck	14	0,095	33	0,066	0
lexicon	15	0,12	32	0,08	0
sentlength	28	4,525	19	5,111	0
ari	19	4,02	28	7,47	0
personalstyle	26	0,084	20	0,028	1
impersonalstyle	30	0,118	12	0,056	5
passiveconstructs	22	0,0139	25	0,195	0
simplepres	28	0,061	19	0,172	0
adverbmodifier	32	0,013	15	0,014	0
deadverbs	26	0,038	18	0,035	3
fillers	27	0,013	20	0,011	0
examplecount	27	3,815	13	9,2462	7
sentlengthvar	28	4,157	19	3,947	0
Bereichsverletzungen	15	1,733	12	1,833	20
Regelverletzungen	15	2,600	12	3,500	20

Tab. 3: Veränderung der Metrikwerte und Regelverletzungen im paarweisen Vergleich der Draft- und Final-Versionen. Die Werte der erwünschten Richtung sind fett und kursiv hervorgehoben, „Anzahl“ entspricht der Anzahl der Arbeiten und „Delta“ der durchschnittlichen Veränderung in diesen Arbeiten.

Beim Einsatz in der Lehre muss berücksichtigt werden, dass es sich um keine Software zur Bewertung/Benotung handelt, sondern lediglich gewisse kon-

strukturelle Verbesserungsvorschläge generiert werden. Studierende müssen sich dessen bewusst sein, dass die Textqualität nach Abschluss des Moduls durch einen menschlichen Prüfer bewertet wird.

Die Software steht zum Download unter der folgenden URL bereit: <https://github.com/ooz/Confopy> .

Confopy zeigt für dieses Paper nur eine Bereichsverletzung an: die Sätze sind mit durchschnittlich 17,54 Worten zu lang. Die Lesbarkeit sollte mit $\text{ari}=66,03$ gegeben sein. Der Wortschatz bewegt sich mit 0,46 an der unteren Grenze.

Literatur

- [AB06] Y. Attali, J. Burstein: Automated Essay Scoring With e-rater V.2. *Journal of Technology, Learning, and Assessment*, 4:3, S. 191–205, 2006.
- [BB05] R. E. Bennett, A. Ben-Simon: Toward More Substantively Meaningful Automated Essay Scoring. *Journal of Technology, Learning, and Assessment*, 6:1, S. 1–47, 2007.
- [Bra04] S. Brants et. al.: TIGER: Linguistic Interpretation of a German Corpus. *Journal of Language and Computation*, 2, S. 597–620, 2004.
- [Gui01] W. H. Guiford: Teaching Peer Review and the Process of Scientific Writing. *Advances in Physiology Education*, 25:3, S. 167–175, 2001.
- [SS67] E. A. Smith, R. J. Senter: Automated Readability Index. Technischer Bericht AMRL-TR-66-220, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio, 1967.
- [TM02] S. Teufel, M. Moens: Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28:4, S. 409–445, 2002.
- [Tim11] B. E. C. Timmerman et. al.: Development of a ‚universal‘ rubric for assessing undergraduates’ scientific reasoning skills using scientific writing. *Assessment & Evaluation in Higher Education*, 36, S. 509–547, 2011.
- [VNC03] S. Valenti, F. Neri, A. Cucchiarelli: An Overview of Current Research on Automated Essay Grading. *Journal of Information Technology Education*, 2, S. 319–330, 2003.
- [Wei06] N. Weicker, B. Draskoczy, K. Weicker: Fachintegrierte Vermittlung von Schlüsselkompetenzen der Informatik. In: HDI 2006, Hrsg.: P. Forbrig, G. Siegel, M. Schneider, S. 51–62, Bonn, Gesellschaft für Informatik, 2006.
- [Zsc15] O. Zschebye: Automatische sprachliche und strukturelle Analyse wissenschaftlicher Texte. Masterarbeit, HTWK Leipzig, 2015.