Universität Basel

Die Autorschaft der umstrittenen Paulusbriefe

untersucht mit Machine Learning

Seminararbeit Frühjahrssemester 2020
Seminar 56906-01: Einführung in Machine Learning

Der R-Code und begleitende Materialien sind verfügbar auf
https://github.com/jnussbaum/authorship-attribution

Verfasser:
Johannes Nussbaum
j.nussbaum@unibas.ch
13-701-974

Betreuer:
Prof. Dr. Gerhard Lauer

Digital Humanities Lab
Universität Basel
Bernoullistrasse 32
4056 Basel

Basel, den 19.11.2020

Inhalt

English Abstract

Which of the Paulinian letters were actually written by the Apostle Paul? This Authorship Attribution study tackles this question by the aid of the General Imposters Framework as implemented in the R-package stylo. The assumptions are that Rom, 1-2 Cor, Gal, Phil, 1 Thess and Phlm are authentic, and that the non-Paulinian texts in the NT form a good corpus of distractor authors. This study uses as text representations {1,2,3}-grams of Greek words, {1,2,3}-grams of Greek letters, {1,2,3}-grams of Strong-numbers, and some variations of Part-of-Speech-tags with morphological information. These representations were combined with the distance measures Cosine, Entropy, and Canberra. These combinations were tested for attributive success, and only those with a very small area of uncertainty were selected to run the analysis, so that on theoretical grounds, the results are expected to be highly significant. The results of the imposters' method show clear authorial signs of Paul for all seven contested Paulinian letters. Further research is needed to corroborate these results, but they could potentially gather big interest in the field of theology. Please note that this research is neither published nor peer-reviewed. Please have a look at the code and give feedback at https://github.com/jnussbaum/authorship-attribution.

1 Einleitung

Kann man herausfinden, von wem ein anonymer Text geschrieben wurde? Jeder Text enthält gewisse Merkmale, die nicht selbstverständlich sind, sondern die von jedem Schreibenden anders eingesetzt werden. Beispiele für solche Merkmale sind die Komplexität der Satzgefüge, oder die Häufigkeit von Personalpronomen wie «ich». Neben den klassischen literaturwissenschaftlichen Zugängen zu diesem Thema kann die Frage auch bearbeitet werden mit den Methoden der Statistik oder des maschinellen Lernens (Machine Learning, ML). Diese Methoden führten zu einem eigenen Forschungsfeld, welches Stilometrie oder Authorship Attribution (AA) genannt wird. Schon früh wurde auch das Neue Testament mit stilometrischen Methoden untersucht, weil darin Schriften enthalten sind, deren Autor nicht zweifelsfrei feststeht. Es liegt hier allerdings noch viel Potential brach: Einer der Gründe dafür ist die rasante Entwicklung, in welcher sich die AA immer noch befindet, sodass vorhandene Studien rasch veralten. Ausserdem stellt das Neue Testament die AA vor besondere Herausforderungen, sei es wegen der geringen Textmenge, dem umstrittenen Wortlaut des Textes, und der Unklarheit über das Studiendesign, z.B. welche Passagen überhaupt miteinander verglichen werden sollen. Ausserdem funktioniert AA mit griechischen Texten allgemein weniger gut als z.B. mit englischen.[1] Weitere

[1] Entsprechende Benchmarking-Vergleichsstudien wurden nur für Neugriechisch vorgenommen (Juola et al. 2019), aber es ist davon auszugehen, dass Altgriechisch noch schlechter abschneidet. Die Gründe dafür sind ungeklärt, könnten aber in der komplexeren Morphologie liegen, was dazu führt, dass es mehr unterschiedliche Wörter gibt, die seltener vorkommen.

Herausforderungen sind der Methodentransfer von den exakten Wissenschaften in die Geisteswissenschaften, und die Deutung der Antworten, welche von Algorithmen gegeben werden.

Als Fallbeispiel versucht die vorliegende Arbeit die Frage zu beantworten, welche von den umstrittenen Paulusbriefen im Neuen Testament tatsächlich vom Apostel Paulus stammen. In *2 Theoretische Grundlagen* werden exemplarisch die Schritte vorgestellt, welche zur Beantwortung dieser Frage ausgeführt werden müssen: Zuerst muss eine Textform des griechischen Grundtextes ausgewählt und in digitaler Form beschafft werden. Anschliessend müssen Textmerkmale ausgewählt werden, anhand derer die Texte repräsentiert werden, sodass diese Repräsentationen miteinander verglichen werden können. Anschliessend steht die Entscheidung an, mit welchen statistischen Verfahren diese Repräsentationen sinnvollerweise in Autorenklassen eingeteilt werden können. Zu diesem Zweck ist es nötig, die Grundlagen der Statistik zu kennen, z.B. Verfahren zur Validierung von Resultaten. Nicht zuletzt ist es unerlässlich, die verschiedenen Möglichkeiten zu kennen, wie eine Studie aufgebaut oder «designt» werden kann. Basierend auf diesen theoretischen Erkenntnissen wird das konkrete Vorgehen in *3 Design der vorliegenden Studie* beschrieben: Mit den Werkzeugen des R-Package `stylo` (Eder et al. 2016) werden die umstrittenen Paulusbriefe gemäss dem General Imposters Framework untersucht. In *4 Resultate* werden die Ergebnisse präsentiert, und anschliessend in *5 Diskussion* besprochen, bevor ein Fazit gezogen wird.

2 Theoretische Grundlagen

Die AA-Community besteht zu einem Grossteil aus Geisteswissenschaftlern ohne Ausbildung in den exakten Wissenschaften. Das Feld ist deshalb von einer pragmatischen Denkweise geprägt, welche zwar brauchbare Resultate hervorbringt, aber die theoretischen Grundlagen zum Teil missversteht. Die folgende Ermahnung aus einem ML-Einführungswerk aus den exakten Wissenschaften gilt deshalb nicht nur für Informatiker, sondern umso mehr für unser Feld: «(…) machine learning experiments need to be designed better. (…) The point is that testing should not be a separate step done after all runs are completed (…); the whole process of experimentation should be designed beforehand, relevant factors defined, proper experimentation procedure decided upon, and then, and only then, the runs should be done and the results analyzed.» (Alpaydın 2010, S. xxxvi). Ein grosser Teil der vorliegenden Arbeit beschäftigt sich deshalb damit, die wichtigsten Konzepte wie Textmerkmal, Textrepräsentation, Distanzmass, Studiendesign und Machine Learning klar und sauber zu definieren und voneinander abzugrenzen.

2.1 Der griechische Text des Neuen Testaments

Das Neue Testament ist eine Sammlung von 27 griechischen Schriften, die im 1. oder frühen 2. Jh. n.Chr. in der Osthälfte des Römischen Reiches entstanden sind. Sie stammen von verschiedenen

Autoren und gehören unterschiedlichen Textgattungen an. Einige nennen ausdrücklich ihren Autor, so z.B. die Paulusbriefe, andere sind anonym, wie z.B. der Hebräerbrief. Von den meisten Schriften entwickelte sich schon rasch ein Konsens, wer ihr Verfasser sei. Diese traditionellen Zuschreibungen sind heute teilweise umstritten. Die Sache verkompliziert sich weiter, weil die Paulusbriefe gemäss eigener Auskunft gar nicht von Paulus selbst geschrieben, sondern diktiert wurden. So heisst es im letzten Abschnitt des Galaterbriefs: «Seht, mit wie großen Buchstaben ich euch schreibe mit eigener Hand!» (Gal 6,11). Eine ähnliche Formulierung findet sich in 1. Petrus 5,12: «Durch Silvanus […] habe ich euch wenige Worte geschrieben…». Offensichtlich schrieb Paulus nur die letzten paar Zeilen seiner Briefe selbst. Zu welchem Grad der Schreiber in die Formulierung eingriff, ist unklar.

Weil die Schriften des Neuen Testaments schon früh eine grundlegende Rolle spielten im frühen Christentum, wurden sie immer wieder abgeschrieben und weiterverbreitet, sodass innert Kürze im gesamten Römischen Reich und darüber hinaus tausende von Kopien im Umlauf waren. Die originalen Schriftstücke der biblischen Autoren selbst, die sogenannten Autographen, sind wohl schon nach einigen Jahrzehnten verrottet, da Papyrus nicht gut konservierbar ist im mediterranen Klima. Schon früh tauchte das Problem auf, dass sich Fehler und Varianten im Text einschlichen, sei es durch Unachtsamkeit beim Kopieren oder absichtlich, um den Text im Sinne des Abschreibers zu «verbessern». Dem versuchte man entgegenzuwirken durch möglichst sorgfältiges Abschreiben, und durch den Abgleich mit möglichst authentischen Handschriften, sodass man abschätzen konnte, welche Variante wohl die ursprüngliche gewesen war. Dieses Vorgehen kannte man in der griechisch-römischen Welt bereits seit hellenistischer Zeit (3.-1. Jh. v.Chr.), wo u.a. in der Bibliothek von Alexandria die wissenschaftliche Disziplin der Textkritik betrieben wurde, also der Rekonstruktion des ursprünglichen Wortlauts von literarischen Werken (insbesondere von Homers Ilias und Odyssee), durch sorgfältigen Vergleich verschiedener Handschriften. Durch solche Massnahmen gelang es dem frühen Christentum, die neutestamentlichen Texte erstaunlich authentisch zu erhalten.

Die moderne Textkritik macht im Grunde dasselbe wie schon in der Antike: Sämtliche verfügbaren Handschriften, antiken Übersetzungen und Zitate in den Werken der Kirchenväter werden gesammelt, verglichen, und das Resultat in einer textkritischen Edition veröffentlicht. Der Herausgeber druckt denjenigen Wortlaut im Haupttext ab, der ihm am wahrscheinlichsten erscheint, und listet die Abweichungen davon im sogenannten textkritischen Apparat auf. Immer mit angegeben ist, welche Handschriften welche Varianten (genannt Lesarten) haben. Der Leser kann somit selbst entscheiden, ob er an einer bestimmten Stelle dem Haupttext folgen will, oder ob er eine abweichende Lesart für authentischer hält. Aufgrund der unüberschaubaren Fülle an Handschriften und Lesarten wird nie Einigkeit herrschen, welche der verschiedenen Rekonstruktionen «die korrekte» ist, oder ob ein bestimmter Satzteil oder gar Abschnitt wirklich ursprünglich ist. Für die computergestützte Autoridentifikation ist

dies problematisch: Vielleicht verbirgt sich gerade in einigen Lesarten des Apparats der Fingerabdruck des Autors, ohne dass wir wissen, ob dies nun der wahre Fingerabdruck ist, oder der Fingerabdruck von Kopisten. Diese Herausforderung ist aber auch ein möglicher Anwendungsfall für die Stilometrie: Von umstrittenen Passagen wie dem langen Markusschluss (Mk 16,9-20), der Ehebrecherin-Perikope (Joh 7,53-8,11) und dem langen Römerschluss (Röm 16,25-27) könnte untersucht werden, ob sie stilometrisch zu ihrem Werk passen.[2] Zunächst besteht die beste Lösung aber wohl darin, einen bibelwissenschaftlich breit abgestützten Standardtext zu wählen, wie Westcott-Hort, Nestle-Aland, oder das SBL Greek New Testament, und diejenigen Wörter daraus zu entfernen, welche als wahrscheinliche Interpolationen gekennzeichnet sind.

2.2 Digitale Versionen des Texts

Die digitale Verfügbarkeit der Texte ist natürlich von zentraler Bedeutung. Die Perseus Digital Library[3] bietet von zahlreichen Werken der griechischen Literatur alte Editionen zum Download an, deren Urheberrecht abgelaufen ist. Formatiert sind die Texte in TEI/XML, wobei die griechischen Zeichen mittels Beta-Code in UTF-8 codierte ASCII-Zeichen umgewandelt wurden, unter Beibehaltung von Akzenten und Gross-Kleinschreibung.[4] Das Neue Testament ist in der Perseus Digital Library in der Form der Edition von Westcott-Hort 1885 verfügbar. Eine andere Möglichkeit ist das SBL Greek New Testament[5], welches als OSIS/XML, XML, oder TXT Plain Text angeboten wird, wobei die griechischen Zeichen jeweils in UTF-8 Unicode, mit Akzenten und Gross-Kleinschreibung dargestellt sind. Die grösste Auswahl an Editionen bietet getbible.net[6]: Den Textus Receptus gemäss Stephanus 1550 mit den

[2] Allerdings sind diese Passagen eigentlich viel zu kurz, um verlässliche Resultate zu erwarten. Es kann nicht davon ausgegangen werden, dass jede kurze Passage repräsentativ für ihren Autor ist. Das sogenannte Gesetz von Zipf besagt, dass Textmerkmale weder unabhängig voneinander auftreten, noch gleichmässig verteilt sind.

[3] Leider werden die XML-Downloads auf der neuen Oberfläche Scaife-Viewer nicht angeboten, sondern nur im alten Perseus Hopper: http://www.perseus.tufts.edu/hopper/collection?collection=Perseus:collection:Greco-Roman.

[4] Beta-Code ist eine Codierung von polytonischem Griechisch mit lateinischen Buchstaben. Beta-Code kann mittels eines Converters in Unicode umgewandelt werden. Beispiel: Der Beta-Code *o(/te o(fi/loj a)pe/qanen entspricht dem Unicode Ὅτε ὁ φίλος ἀπέθανεν.

[5] http://sblgnt.com/download/.

[6] Aufgrund der unklaren Herkunft der Texte erfüllt getbible.net zwar ein wichtiges akademisches Kriterium nicht. Aber die Recherchen und Inspektion der Texte durch den Verfasser stimmen optimistisch. Die griechischen Texte stammen aus zwei verschiedenen Quellen: Version 1 von getbible.net hat die griechischen Texte des Unbound-Projekts der kalifornischen Biola University (https://unbound.biola.edu). Ein Zugriff ist möglich via die API von getbible.net, welche den Text als JSON in UTF-8 Unicode anbietet, oder man kann die TXT-Dateien herunterladen auf https://getbible.net/Bibles oder https://github.com/getbible/Unbound-Biola/. Gemäss den Informationen im Github Repository enthalten diese Texte aber unkorrigierte Fehler, weshalb man die Version 2 benutzen soll. Aber die Version 2 gibt als Quelle ihrer Texte die anonyme Seite https://sites.google.com/a/wmail.fi/greeknt/home/greeknt an. Dort stehen diverse Editionen in diversen Codierungen zur Verfügung, allerdings teilweise in kryptischen, unlesbaren Formaten. Ein alternativer Zugriff auf diese Dateien könnte via JSON-API von getbible.net erfolgen (z.B. https://getbible.net/v2/westcotthort.json). Das Github Repository (https://github.com/getbible/v2) bietet keinen Download mehr an. Für die vorliegende Arbeit wurden die TXT-Dateien der Version 1 von getbible.net gewählt, also diejenigen von https://github.com/getbible/Unbound-Biola/.

Varianten von Scrivener 1894, die byzantinische Textform gemäss der Edition von Robinson/Pierpont 2000, Tischendorfs 8. Auflage 1869-1872, und schliesslich den Westcott-Hort 1881 mit den Varianten von Nestle-Aland 27. Auflage = United Bible Societies 4. Auflage. Für die vorliegende Untersuchung fällt die Wahl auf die letztgenannte Edition. Verfügbar ist sie als TXT-Datei in UTF-8 Unicode, alles in Kleinbuchstaben und ohne Akzente. Wenn alle mit VAR1 gekennzeichneten Passagen daraus entfernt werden, resultiert NA27 daraus. Zwei Variationen dieser Datei stehen zum Download bereit: Nur der griechische Text, oder eine geparste Form, in welcher jedes griechische Wort gefolgt wird von seiner Strong-Nummer und einem POS-Tag (Part of Speech, Erklärung siehe unten).

Für die vorliegende Untersuchung wurde die Version mit Strong-Nummer und POS-Tag gewählt. Bei der Strong-Nummer handelt es sich um eine eindeutige Identifikation jedes griechischen Wortstammes, der im Neuen Testament vorkommt. Da das System aber einige Mängel aufweist, wurde es vom Neutestamentler Maurice Robinson weiterentwickelt und um POS-Tags erweitert. POS-Tags sind eine Codierung für die grammatikalische Form eines Wortes, also für Informationen wie Wortart, Kasus, oder Tempus. Mit diesem System ist es also möglich, Wörter derselben Grundform als zusammengehörig zu identifizieren, auch wenn sie durch die Flexion sehr unterschiedlich lauten. Gleichzeitig geht aber die syntaktische Information nicht verloren. Das Annotationssystem ist in einem unpublizierten Artikel von Robinson beschrieben (Robinson undatiert, S. 1–7). Als Beispiel sei der Vers Matthäus 3,15 abgedruckt, so wie er in der Datei https://github.com/getbible/Unbound-Biola/blob/master/Greek__NT_Westcott_Hort_UBS4_variants_Parsed__westcotthort__LTR.txt vorzufinden ist:

```
40N||3||15||αποκριθεις G611 G5679 V-AOP-NSM δε G1161 CONJ ο G3588 T-NSM ιησους
G2424 N-NSM ειπεν G2036 G5627 V-2AAI-3S  {VAR1: αυτω G846 P-DSM } {VAR2: προς
G4314 PREP αυτον G846 P-ASM } αφες G863 G5628 V-2AAM-2S αρτι G737 ADV ουτως G3779
ADV γαρ G1063 CONJ πρεπον G4241 G5901 V-PQP-NSN εστιν G2076 G5748 V-PXI-3S ημιν
G2254 P-1DP πληρωσαι G4137 G5658 V-AAN πασαν G3956 A-ASF δικαιοσυνην G1343 N-ASF
τοτε G5119 ADV αφιησιν G863 G5719 V-PAI-3S αυτον G846 P-ASM
```

Jeder Vers steht auf einer separaten Zeile. Als erstes steht ein Code zur Identifikation des Bibelbuchs, welcher auf der Homepage des Unbound-Projektes aufgeschlüsselt werden kann.[7] Anschliessend folgen die Kapitel- und Versnummern. Gleich beim ersten Wort wird eine Schwäche der Strong-Nummern ersichtlich: Es gibt mehrere mögliche Strong-Nummern für dasselbe Wort. Für die vorliegende Untersuchung wurde immer nur die erste Nummer berücksichtigt. Auf der zweiten Zeile sieht man eine abweichende Lesart: Westcott-Hort hat αυτω, wo United Bible Societies 4. Auflage (im Haupttext identisch mit Nestle-Aland, 27. Auflage) προς αυτον hat. Da αυτω und αυτον von demselben

[7] https://unbound.biola.edu/index.cfm?method=unbound.showFAQ&faq_name=Bible_Index_Codes.

Wortstamm sind, haben beide dieselbe Strong-Nummer. Ihre unterschiedliche Form führt aber zu zwei verschiedenen POS-Tags.

Für die vorliegende Untersuchung wurden vier verschiedene Formen des Textes extrahiert: der griechische Text, eine Folge von allen Strong-Nummern, eine Folge von allen POS-Tags, und schliesslich noch eine Folge von allen ersten Elementen der POS-Tags. Dies ist nötig, weil die grammatikalischen Informationen der Wörter derart genau codiert werden in Robinsons POS-Tag-System, dass die Häufigkeit der einzelnen Tags sehr klein ist. Nimmt man hingegen nur die ersten Elemente der Tags, steigt die Häufigkeit des einzelnen Tags, sodass eine bessere stilometrische Analyse der Syntax möglich wird.

Der abgedruckte Beispielsvers Matthäus 3,15 führt also zu folgenden vier Textformen:

Griechischer Text:	`αποκριθεις δε ο ιησους ειπεν προς αυτον αφες αρτι ουτως γαρ πρεπον εστιν ημιν πληρωσαι πασαν δικαιοσυνην τοτε αφιησιν αυτον.`
Strong-Nummern:	`G611 G1161 G3588 G2424 G2036 G4314 G846 G863 G737 G3779 G1063 G4241 G5901 G2076 G2254 G4137 G3956 G1343 G5119 G863 G846`
POS-Tags:	`V-AOP-NSM T-NSM N-NSM V-2AAI-3S PREP P-ASM V-2AAM-2S ADV ADV CONJ V-PQP-NSN V-PXI-3S P-1DP V-AAN A-ASF N-ASF ADV V-PAI-3S P-ASM`
POS-Tags (nur erster Teil):	`V T N V PREP P V ADV ADV CONJ V V P V A N ADV V P`

Nun stellt sich natürlich die Frage, wie gut sich diese vier Textformen eignen, um daraus Textmerkmale (features) zu gewinnen für die AA. Sie sind sicher nicht schlecht für den Anfang, aber es gibt noch Verbesserungsbedarf, insbesondere bei den Strong-Nummern. Eine befriedigendere Lösung wäre, mithilfe des R-Package `UDPipe` die Lemmata und POS-Tags zu extrahieren (Straka und Straková 2017). Für Altgriechisch stehen in `UDPipe` zwei Modelle zur Verfügung (`ancient_greek-perseus` und `ancient_greek-proiel`). Dies hätte den Vorteil, dass die Lemmata einheitlicher wären, und dass die neutestamentlichen Texte auch mit nichtbiblischen Texten verglichen werden könnten wie z.B. Josephus.

2.3 Textmerkmale

Bevor irgendeine Methode zur Autorbestimmung angewendet werden kann, müssen Textmerkmale (engl. features) ausgewählt werden. Ein erster Schritt, der immer ausgeführt wird, ist das Preprocessing: Die Texte (Dokumente genannt) werden von allen Metatextelementen wie Seitenzahlen, Inhaltsverzeichnis oder Überschriften befreit, und anschliessend in einzelne Wörter zerlegt. Oft werden auch

alle Grossbuchstaben in Kleinbuchstaben umgewandelt, und je nach Sprache müssen Akzente und diakritische Zeichen entfernt werden. Der Grund dafür ist, dass man wissen möchte, wie oft ein Worttyp (engl. type/term) vorkommt, also muss man sicherstellen, dass jedes Einzelwort (engl. token) eine einheitliche Form hat. Worttypen sind nicht zu verwechseln mit Lemmata: Die beiden Einzelwörter (token) αὐτόν und αὐτός gehören zum selben Lemma (Wörterbuch-Grundform αὐτός), aber nicht zum selben Worttyp (type). Hingegen gehören αὐτόν und αὐτὸν zum selben Worttyp. Durch das Entfernen der Akzente wird sichergestellt, dass die beiden Einzelwörter αὐτόν und αὐτὸν als derselbe Worttyp erkannt werden. Die griechische Sprache stellt uns hier aber vor ein verzwicktes Problem: Wie eben dargelegt, tragen gleiche Silben je nach Position unterschiedliche Akzente, wodurch die Einzelwörter nicht mehr als demselben Worttyp zugehörig erkannt werden. Andererseits werden durch Akzentuierung auch Wörter völlig unterschiedlicher Bedeutung voneinander unterschieden, die aus derselben Buchstabenfolge bestehen, so z.B. οὐ («nicht») versus οὗ («wo»), oder ἡ (best.Art.f.Sg.) versus ἦ («bestimmt») versus ἤ («oder», «als» (Komparativ)). Ein Ausweg kann hier nur gefunden werden durch Lemmatisierung kombiniert mit POS-Annotation. Die vier Textrepräsentationen der vorliegenden Untersuchung sind also suboptimal, wie bereits oben dargelegt.

Das Resultat dieses ersten Schritts ist ein String aus Einzelwörtern. Die früher üblichen univariaten Verfahren haben dann ein einziges Textmerkmal daraus extrahiert, wie beispielsweise das Verhältnis von Anzahl Einzelwörtern zu Anzahl Worttypen (type-token ratio), Anzahl von nur einmal vorkommenden Wörtern (hapax legomena), Yules K-Mass, Sichels S-Mass, oder Honores R-Mass (Koppel et al. 2009, S. 11). Die neueren multivariaten Ansätze hingegen arbeiten meistens mit einer Häufigkeitstabelle, in welcher von sämtlichen vorkommenden Wörtern die Häufigkeit erfasst wird. Die einfachste Art, Textmerkmale aus den Dokumenten zu gewinnen, ist eine Häufigkeitstabelle der Worttypen, wo für jeden Worttyp angegeben wird, wie oft er in jedem Dokument vorkommt. Dabei gehen aber die Informationen zum Kontext, in welchem die Wörter stehen, verloren. Deshalb wird dieses Vorgehen «bag of words» genannt, weil die Wörter losgelöst voneinander betrachtet werden. Um diese Kontextinformationen nicht zu verlieren, können die Dokumente in n-Gramme statt Einzelwörter aufgeteilt werden, also in Gruppen von n aufeinanderfolgenden Einzelwörtern. Der schon besprochene Vers Matthäus 3,15 könnte folgendermassen in n-Gramme aufgeteilt werden:

n = 1 (bag of words): `αποκριθεις, δε, ο, ιησους, ειπεν, προς, αυτον,…`

n = 2 (Bigramme): `αποκριθεις δε, δε ο, ο ιησους, ιησους ειπεν, ειπεν προς,…`

n = 3 (Trigramme): `αποκριθεις δε ο, δε ο ιησους, ο ιησους ειπεν, ιησους ειπεν προς,…`

Die Häufigkeitstabelle wird dann erstellt auf Basis der n-Gramme statt der Einzelwörter. n kann beliebig hohe Werte annehmen, bloss nimmt die Häufigkeit jedes n-Gramms ab, je höher n ist, wodurch die Aussagekraft der Häufigkeitstabelle abnimmt.

Eine weitere Möglichkeit ist, die Häufigkeitstabellen basierend auf Buchstaben-n-Grammen zu erstellen. Für Matthäus 3,15 wäre das:

n = 1: α, ο, κ, ρ, ι, θ, ε, ι, σ, δ, ε, ο, ι, η, ς, …

n = 2 (Bigramme): απ, πο, οκ, κρ, ρι, ιθ, θε, ει, ισ, σδ, δε, εο, οι, ιη, ης, …

n = 3 (Trigramme): απο, ποκ, οκρ, κρι, ριθ, ιθε, θει, εισ, ισδ, σδε, δεο, εοι, οιη, …

Dieses Vorgehen ist kontraintuitiv, weil diese Textmerkmale keine linguistische Relevanz mehr haben. Dass der Autor aufgrund von Wörtern oder Wortpaaren bestimmt wird, ist einleuchtend, hingegen kann die Bestimmung anhand von Merkmalen, die für uns Menschen keine Relevanz haben, Skepsis auslösen. Tatsächlich wurde aber nachgewiesen, dass linguistisch irrelevante, zufällig gewonnene Textmerkmale manchmal sogar eine bessere Performanz aufweisen. So hat Eder aus Versehen eine falsche Konvertierung des Beta-Codes seiner griechischen Texte vorgenommen, was zu falschen Leerschlägen innerhalb von Wörtern führte. In seinen Experimenten erreichte diese Textform dann sogar eine bessere Performanz als der korrekt konvertierte Text (Eder 2011, S. 105). Buchstaben-n-Gramme haben selbst dann noch eine hohe Performanz, wenn andere Textmerkmale nicht mehr funktionieren, beispielsweise wenn das Korpus verunreinigt ist durch viele falsche Buchstaben, wie das bei schlechter OCR (Optical Character Recognition) passieren kann (Eder et al. 2018, S. 12). Vor diesem Hintergrund wäre es für die vorliegende Arbeit nicht einmal abwegig, die Rohdatei zu verwenden, ohne die drei Elemente Text, Strong-Nummer und POS-Tag voneinander zu trennen. Dies hätte auch den positiven Effekt, dass die Gesamt-Textmenge grösser wäre.

Die so erstellte Häufigkeitstabelle, egal auf welchen Textmerkmalen sie beruht, umfasst das gesamte Häufigkeitsspektrum. Traditionell hielt man den oberen Bereich am aussagekräftigsten für die Autorbestimmung, also die häufigsten Merkmale, welche oft als Most Frequent Words (MFW) bezeichnet werden, ob es sich nun um Wörter oder um etwas anderes handelt. Im Fall von Wörtern ist dies einfach nachvollziehbar, denn die häufigsten Wörter sind Funktionswörter wie Pronomen, Artikel, oder Partikeln, welche unabhängig von Gattung und Thema sind (Koppel et al. 2009, S. 11), und kaum bewusst verfälscht werden können durch den Autor. Im Fall von Wort-n-Grammen ist es auch einfach nachvollziehbar, weil so der kontextabhängige Gebrauch von Wörtern untersucht wird. Bei anderen Textmerkmalen ist die Funktionsweise zwar weniger intuitiv, aber die Performance spricht für sich.

Bald wurde aber klar, dass nicht nur der oberste Bereich der Häufigkeitstabelle relevant ist: Man kann irgendeine Anzahl MFW für die Auswertung auswählen, beginnend an irgendeinem Ort der Häufigkeitstabelle (z.B. vom ersten bis zum hundertsten MFW, oder vom fünfzigsten bis zum vierhundertsten MFW). Diese «Fenster» performen unterschiedlich gut, je nach Länge, Startort, Sprache, Gattung und gewähltem Distanzmass. *Abb. 1* zeigt die Resultate eines Experiments, in welchem die Bestimmungszuverlässigkeit von verschiedenen MFW-Fenstern anhand von lateinischen Prosatexten bekannter Autorschaft getestet wurde.

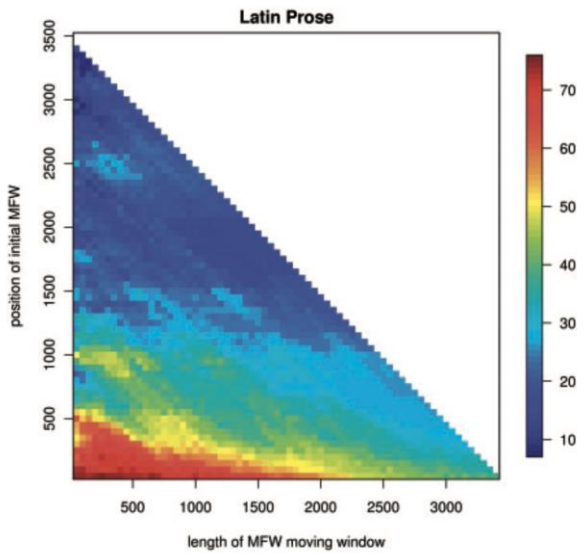

Abb. 1: Zuweisungserfolg für 94 lateinische Prosatexte, in Prozent korrekte Autorbestimmungen (Rybicki und Eder 2011, S. 318).

Es ist ersichtlich, dass bloss mit den MFW 1-50, 1-100, …, 1-700, und ausserdem mit den MFW 51-100, 51-150 und einigen anderen akzeptable Resultate erreicht wurden. Es ist zu beachten, dass ein anderes lateinisches Prosakorpus oder ein anderes Distanzmass bereits wieder zu anderen Resultaten führen kann. Leider ist bis heute unklar, was die genauen Faktoren hinter diesen Unterschieden sind. Um das optimale MFW-Fenster für einen konkreten Anwendungsfall zu finden, wie z.B. die neutestamentlichen Texte der vorliegenden Arbeit, muss also zuerst anhand von Texten bekannter Autorschaft in vielen mühsamen Iterationen das beste MFW-Fenster ermittelt werden.

Eine etwas suboptimale Alternative ist das Bootstrapping, dass also die Autorbestimmung mit einer grossen Anzahl MFW-Fenstern durchgerechnet wird, wodurch man viele richtige und viele falsche Autorbestimmungen erhält. Per Mehrheitsentscheid wird dann aus allen Bestimmungen diejenige ausgewählt, die am häufigsten vorkommt. Eine etwas bessere Alternative, zumindest für binäre Klassifizierungsprobleme, wo nur zwei Autoren zur Auswahl stehen, ist das Zeta-Verfahren. Das Zeta-Verfahren basiert auf einem Artikel von Burrows (Burrows 2007), und wurde dann verschiedentlich weiterentwickelt. Es ist ein Verfahren, um aus zwei Texten diejenigen Wörter zu extrahieren, anhand derer sich diese beiden Texte am klarsten unterscheiden lassen. Das Resultat sind zwei Listen, die eine mit Wörtern, die im ersten Text öfter vorkommen als im zweiten, und die andere Liste mit Wörtern, die der zweite Text öfter verwendet als der erste (Schöch et al. undatiert; Eder et al. 2018, S. 26–27).

Die bisher besprochenen Textmerkmale waren ausschliesslich lexikalisch, also auf den Wörtern basierend. Doch in den letzten Jahren sind viele kreative Ideen hervorgebracht worden, wie Textmerkmale aus einem Text gewonnen werden können. Es ist z.B. möglich, jedes Wort zu ersetzen mit der Anzahl seiner Buchstaben, sodass man eine Zahlenfolge erhält, die Aufschluss gibt darüber, mit wie langen Wörtern an welcher Position ein Autor seine Texte schreibt (Zheng et al. 2006, S. 385). Für stark flektierende Sprachen wie Griechisch ist es auch sinnvoll, morphologische Informationen zu benützen (Stamatatos et al. 2001). Eine weitere Möglichkeit sind syntaktische Informationen, wie sie z.B. in den POS-Tags enthalten sind, welche für die vorliegende Arbeit verwendet werden. Eine etwas elaboriertere Version davon ist, für jeden Satz einen Strukturbaum zu erstellen, welcher die Wörter hierarchisch darstellt, gemäss ihren grammatikalischen Abhängigkeiten. Aus diesem Strukturbaum können dann Textmerkmale gewonnen werden (Tschuggnall und Specht 2016). Schlussendlich gibt es natürlich noch die Möglichkeit, die Textmerkmale nicht durch menschliche Überlegung zu bestimmen, sondern mittels Feature Learning zu gewinnen, einer Gruppe von Machine Learning-Verfahren (Bengio et al. 2013).

2.4 Textrepräsentationen

Um die mathematischen Vorgänge der AA zu verstehen, muss zuerst geklärt werden, welche Vorgehensweisen es gibt, um den Autor eines Textes zu bestimmen. Bis zur ersten Hälfte des 20. Jh. suchte man nach einer Invarianten, also nach einem einzigen Textmerkmal, welches bei allen Texten desselben Autors konstant bleibt, sich aber von Autor zu Autor unterscheidet. Weil man dabei nur eine einzige Variable untersuchte, handelte es sich um ein univariates Verfahren, also mit einer eindimensionalen Messgrösse. Als Matrix dargestellt ist dies eine Tabelle, wobei die Zeilen den Beobachtungen entsprechen (den verschiedenen Dokumenten), und die (einzige) Spalte der (einzigen) Variable. Geometrisch dargestellt wird eine eindimensionale Beobachtung als Punkt auf einer metrisch skalierten Achse, oder als Vektor vom Ursprung (Nullpunkt) zu diesem Punkt hin. Einen Vektor kann man sich vorstellen als Wegbeschreibung zu einem Punkt, vom Ursprung (Nullpunkt) aus. Jeder Bereich auf dieser Achse kann einem Autor zugewiesen werden, weil davon ausgegangen wird, dass das definierte Textmerkmal invariant ist, also beim selben Autor immer in diesem Bereich zu liegen kommt. Ein Beispiel dafür ist die von Yule 1944 vorgeschlagene durchschnittliche Satzlänge:

		Variable(n)
		∅Satzlänge [Wörter/Satz]
Beobachtungen	Dokument 1	3
	Dokument 2	4
	Dokument 3	6
	Dokument 4	9

Tab. 1: Matrixdarstellung der univariaten Analyse einer einzigen Variablen.

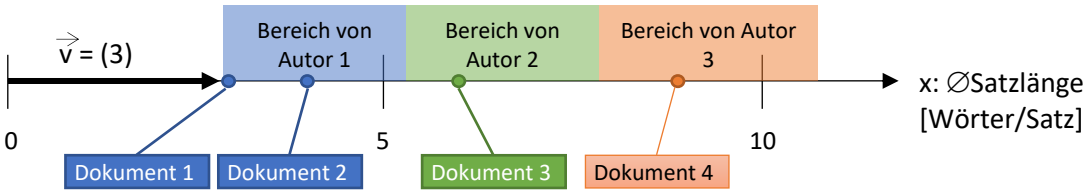

Abb. 2: Geometrische Darstellung der univariaten Analyse einer einzigen Variablen (Darstellung: J. Nussbaum).

Die Suche nach einem einzigen Merkmal erwies sich aber nicht als erfolgreich, weshalb Mosteller und Wallace 1964 erstmals einen multivariaten Ansatz wählten. Ihre Messgrösse war mehrdimensional, weil jede ihrer Beobachtungen (z.B. Text eines Autors) mehrere Variablen (Textmerkmale) erhob. Wenn nun n Variablen/Textmerkmale erhoben werden, kann man jeden untersuchten Text (Dokument, Beobachtung) als Punkt oder Vektor im n-dimensionalen Raum auffassen. Der Vektor ist dann eine Folge von n Zahlen, wobei n der Anzahl Dimensionen entspricht. Anders ausgedrückt werden die Eigenschaften der Texte numerisch parametrisiert und zu einem Merkmalsvektor zusammengefasst. Eine Möglichkeit, wie sich diese Texte dann vergleichen lassen, ist dass man die Distanz dieser Punkte miteinander vergleicht.

Diese Überlegung kann illustriert werden mit einem einfachen Beispiel von zwei Dokumenten und zwei untersuchten Wörtern. Zuerst wird eine Häufigkeitstabelle erstellt, wie in *2.3 Textmerkmale* beschrieben. Weil die beiden Beobachtungen (Dokumente A und B) bloss anhand von zwei Wörtern (Variablen) repräsentiert sind, lassen sich die Dokumente als Punkte in einem zweidimensionalen Koordinatensystem darstellen. Die Ähnlichkeit dieser beiden Dokumente lässt sich als grössere oder kleinere Distanz zwischen diesen beiden Punkten auffassen.

Korpus
Text A: “the the and and and and and”
Text B: “the the the the and and”

Häufigkeitstabelle der Wort-1-Gramme	
and	7
the	6

Textrepräsentationen		
	x-Variable: «the»	y-Variable: «and»
Text A	2	5
Text B	4	2

Tab. *2, 3, 4: Vom Korpus über die Häufigkeitstabelle zu den Textrepräsentationen.*

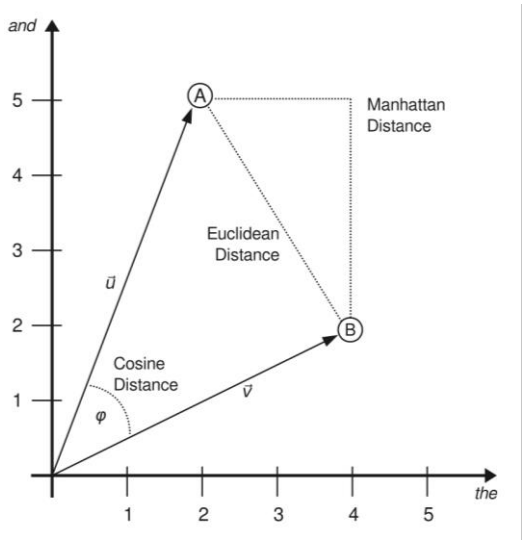

Abb. 3: Verschiedene Distanzmasse zwischen zwei Dokumenten A und B, vermessen mit den zwei Wörtern «and» und «the», resultierend in einem zweidimensionalen Koordinatensystem (Darstellung: Evert et al. 2017, ii7).

Diese Basis-Variante des Vektorraummodells wird auch term-frequency-Modell (tf) genannt, weil die Vektoren Darstellungen der Häufigkeit von Termen (Wörtern, n-Grammen, …) sind. Optional kann man diese Vektoren auf unterschiedliche Arten transformieren. Ein Beispiel dafür ist das term frequency – inverse document frequency Modell (tf-idf). Dabei werden die Häufigkeiten der Terme gewichtet gemäss ihrer Seltenheit im Korpus, wodurch solche Termen stärker gewichtet werden, welche nur in wenigen Dokumenten vorkommen. Ein anderes Modell ist das std-Modell, bei dem die Term-Häufigkeiten gemäss ihrer Standardabweichung (z-Score) über das gesamte Korpus hinweg gewichtet werden (Kestemont et al. 2016, S. 89).

Komplexere AA-Probleme haben viele verschiedene Punkte in einem hochdimensionalen Raum. Zur Beantwortung einer Autorschaftsfrage kommen nun zwei Möglichkeiten infrage: Entweder Machine Learning-Methoden, die Cluster entdecken können in den Punkten, oder die etwas einfachere Delta-Methode, bei welcher der Forscher aufgrund der blossen Distanzen zwischen den Texten selbst eine Zuordnung vornimmt.

2.5 Die Delta-Methode

Betrachten wir zunächst die Delta-Methode. Um die folgenden Darlegungen besser verstehen zu können, wird jeder Schritt in der begleitenden Excel-Tabelle vorgeführt, welche auf https://github.com/jnussbaum/authorship-attribution/ verfügbar ist. Das Vorgehen ist vom Konzept her simpel: Zuerst wird eine Liste der Textmerkmale erstellt (genauer besprochen unter *2.3*

Textmerkmale. Entspricht Schritt 3 in der Excel-Tabelle). Wenn diese Liste n Einträge hat, lässt sich jedes Dokument darstellen durch eine Folge von n Zahlen, wobei jede Zahl die Häufigkeit ist, mit welcher das n. Textmerkmal in diesem Dokument vorkommt (Schritte 4-5 in der Excel-Tabelle). Diese Zahlenfolgen lassen sich als Vektor im n-dimensionalen Raum auffassen. An dieser Stelle spaltet sich der Pfad, weil es nun verschiedene Möglichkeiten gibt, diesen Vektor zu standardisieren. Das in der AA-Community oft verwendete Burrows' Delta (erstmals vorgestellt von Burrows 2002) geht hier den Weg des z-Score (Schritt 6.1 in der Excel-Tabelle). Wie sich später herausgestellt hat, war diese Entscheidung nicht optimal (siehe dazu Argamon 2008, dem ein wichtiger Verdienst zukommt bei der mathematischen Klärung und Präzisierung von Burrows' Delta). Der nächste Schritt ist, ein Distanzmass auf die Dokumente anzuwenden. Es gibt in der Mathematik tausende von Möglichkeiten, eine Distanz zwischen zwei Punkten zu berechnen. Die Wahl von Burrows fiel auf die Manhattan-Distanz (Schritt 7.1.1 in der Excel-Tabelle; zum Verständnis siehe *Abb. 3*).

Das Resultat ist eine Distanzmatrix, in welcher die Distanz von jedem Dokument zu jedem anderen dargestellt ist. Wichtig an diesem Punkt ist, dass es zwei Parameter gibt, welchen entscheidenden Einfluss ausüben auf die Performanz der AA, nämlich die Standardisierung und das Distanzmass. Evert et al. 2017 haben gezeigt, dass die Standardisierung auf Vektorlänge 1 (Schritt 6.2 in der Excel-Tabelle) zentral ist, um gute Resultate zu erreichen, und haben auch die mathematischen Grundlagen dazu erarbeitet.[8] Burrows ursprünglicher Weg der Standardisierung funktioniert zwar auch, aber performt deutlich schlechter in vergleichenden Experimenten. Die Wahl des Distanzmasses ist nicht mehr so zentral, kann aber auch einen Unterschied machen. Inzwischen gibt es diverse Anpassungen des Delta, beispielsweise Eders Delta.

Nun gibt es verschiedene Möglichkeiten, wie diese Distanzmatrix gedeutet wird. Im simpelsten Fall zieht der Forscher direkt aus dieser Matrix seine Schlüsse, indem er schaut, welches die nächsten Nachbarn des untersuchten Textes sind. Mathematisch gesehen gibt die Distanzmatrix ihm eine probabilistische Rangordnung der Kandidaten: Der wahrscheinlichste Autor ist derjenige des nächsten Dokuments, der zweitwahrscheinlichste Autor ist derjenige des zweitnächsten Dokuments, usw. So betrachtet kann das Delta als Klassifikationsmethode aufgefasst werden (Koppel et al. 2009, S. 10; Argamon 2008, S. 137). Eine elaboriertere Möglichkeit besteht darin, diese Distanzmatrix durch Clusteranalyse auszuwerten.

[8] Die Standardisierung auf Vektorlänge 1 führt zum selben Effekt, wie wenn die Cosinusdistanz ohne vorherige Standardisierung angewendet würde: Der Winkel zwischen den Vektoren wird zum entscheidenden Kriterium. Viele Benchmarking-Studien kamen zum Schluss, dass dies einen positiven Einfluss auf die Performanz hat, und hielten die Cosinusdistanz deshalb für besser als andere Distanzmasse. Doch es ist durchaus möglich, dass andere Distanzmasse besser performen als die Cosinusdistanz, so beispielsweise gezeigt von Kocher und Savoy 2019. Dies liegt daran, dass durch Standardisierung auf Vektorlänge 1 derselbe positive Effekt erzielt wird.

2.6 Machine Learning

Treten wir nun einen Schritt zurück, und betrachten nicht mehr die Abstandsmasse zwischen den Texten, sondern die ursprünglichen Vektoren/Punkte im n-dimensionalen Raum. Die Autoridentifikation ist ein Klassifizierungsproblem, wobei die Texte der Input sind, und der Autor des Textes der Output. Statistisch ausgedrückt sind die Texte die unabhängige Variable, und der Autor die abhängige Variable. Zwischen Input und Output steht der Klassifikator, also eine mathematische Funktion, welche die Texte einem Autor zuordnet. Wenn dieser Vorgang auf einem Computer geschehen soll, braucht es einen Algorithmus, also eine Folge von genauen Anweisungen, wie der Input in einen Output transformiert werden soll. Im Fall der Autoridentifikation könnte ein Algorithmus z.B. auf dem beschriebenen Delta-Verfahren beruhen. Die Statistik/ML hat aber noch andere solche Funktionen entwickelt, wie z.B. kNN, SVM, oder Naïve Bayes. Die Gemeinsamkeit dieser Verfahren besteht darin, dass es nicht möglich ist, eine genaue Anweisungsfolge niederzuschreiben, die für jeden Spezialfall anwendbar ist. Wir haben zwar den Input (Texte) und im Fall von Supervised ML auch den gewünschten Output (der bekannte Autor), aber es gibt keine allgemein gültige Regel, wie die Zuweisung (Abbildung/Transformation) geschehen soll. ML löst dieses Problem dadurch, dass nur das Grundgerüst der Funktion bereitgestellt wird, das noch viele Leerstellen hat. Mathematisch gesehen handelt es sich bei den Leerstellen um Parameter, also Variablen, für die gute Werte eingesetzt werden müssen. Der Lern-Schritt, oder das Training des ML besteht darin, dass die Parameter so gesetzt werden, dass die Funktion die Trainingsdaten möglichst gut beschreibt. ML kann als Erweiterung der induktiven Statistik verstanden werden. Die induktive Statistik versucht, die Merkmalsausprägungen innerhalb einer Stichprobe mit einem statistischen Modell zu beschreiben, um mit diesem Modell auf die Merkmalsausprägungen der Grundgesamtheit zu schliessen (Inferenz). Dieses statistische Modell ist im Grunde genommen eine mathematische Funktion mit vielen Parametern. ML gebraucht nun die Theorie der induktiven Statistik, um basierend auf den Trainingsdaten die Parameter der Funktion zu optimieren. Auf die Autoridentifikation bezogen ist diese Funktion ein Algorithmus (Klassifikator), welcher eine Zuordnung von Texten zu einem Autor vornimmt. Anders ausgedrückt findet ML die Grenzen zwischen den Klassen (Autoren), indem eine Klassifikations-Verlustfunktion minimiert wird. Die Art dieser Grenzen hängt von der angewendeten Lernmethode ab, aber jedenfalls sind sie mächtiger als blosse Distanzmasse wie Delta (Alpaydın 2010, S. 1–4; Bzdok et al. 2018; Koppel et al. 2009, S. 11).

ML-Verfahren können nach verschiedenen Gesichtspunkten systematisiert werden: Eine geläufige Unterscheidung ist Unsupervised vs. Supervised ML. Das Unsupervised ML erhält den Input ungelabelt, also ohne Informationen darüber, welches der gewünschte Output ist. Unsupervised ML versucht von sich aus, eine Struktur zu finden im Input. Dies kann auf zwei Arten geschehen: Entweder durch Dimensionalitätsreduktion, also indem die Anzahl Variablen im Datensatz reduziert wird, ohne die darin

enthaltene Information wesentlich zu reduzieren, oder durch Clusteranalyse, also indem homogene Untergruppen gefunden werden mittels der Distanzen zwischen den Datenpunkten. Clusteranalyse kann auf Basis einer Distanzmatrix geschehen, die man durch ein Distanzmass wie etwa Burrows' Delta errechnet hat. Viele Implementierungen von Clusteranalyse sind aber auch in der Lage, Merkmalsvektoren zu verarbeiten, ohne dass die Distanzen dazwischen bereits ausgerechnet sind. Angewandt auf Authorship Attribution könnte Unsupervised ML also die Texte des Neuen Testaments in Gruppen einteilen, die wahrscheinlich vom selben Autor geschrieben wurden (Gatto 2020, §4).

Beim Supervised ML hingegen ist der Input gelabelt mit der Information, was der gewünschte Output ist. Anhand dieser Trainingsdaten lernt das ML dann, neuen (ungesehenen) Input einem Output zuzuordnen. Die Klassifikation gehört zu Supervised ML, weil durch die Trainingsdaten die Klassen bereits bekannt sind, und neue Daten nun der korrekten Klasse zugeordnet werden sollen. Bei Klassifikatoren kann man unterscheiden zwischen binären, welche nur eine Einteilung in genau zwei Klassen vornehmen können, und solchen, die mehrere Klassen finden können. Eine weitere Unterscheidung liegt bei der Natur des Outputs: Bei der probabilistischen Klassifikation wird für jeden Input bestimmt, mit welcher Wahrscheinlichkeit er zu welcher Klasse gehört. Hingegen gibt eine Maximum-Likelihood-Methode nur diejenige Klasse zurück, welche die wahrscheinlichste ist. (Gatto 2020, §5)

Manchmal wird ML auch anders definiert als in der vorliegenden Arbeit. Insbesondere wird unter ML manchmal nur Deep Learning verstanden, was ein anderer Begriff für neuronale Netze ist. Dies geht aber über den Rahmen der vorliegenden Arbeit hinaus. Mehr Verwirrung kann gestiftet werden durch die Definition der Entwickler des R-package `stylo`: Sie verstehen unter ML nur das Supervised ML, und rechnen das Unsupervised ML zu den strukturentdeckenden statistischen Methoden. Dafür behandeln sie das Delta als Supervised ML. Diese Definition schlägt sich nieder in ihren beiden Funktionen `stylo()` für strukturentdeckende Verfahren, und `classify()` für Supervised ML und Delta (Eder et al. 2018, 7,21). Der Grund für diese Begriffsverwirrung liegt einerseits begründet in einer Vermischung der beiden Konzepte Distanz und Studiendesign, und andererseits in einer Verwirrung über das Konzept von Machine Learning. Wie in *2.4 Textrepräsentationen* dargelegt, ist Delta ein Distanzmass. Wenn man die Distanz von jedem Text zu jedem anderen berechnet, erhält man eine Distanzmatrix. Je nach Studiendesign stellt der Forscher dann andere Fragen an diese Distanzmatrix. Dies kann illustriert werden an der begleitenden Excel-Tabelle auf https://github.com/jnussbaum/authorship-attribution. Angenommen, in der Distanzmatrix 7.1.1 sei das Dokument 1 von unbekannter Autorschaft, während die Dokumente 2-4 von den Autoren 2-4 verfasst wurden, und die Forschungsfrage sei, Dokument 1 einem der Autoren 2-4 zuzuordnen. Dann führt das Studiendesign zu einem Klassifizierungsproblem, und die Distanzmatrix ist ein möglicher Klassifikator dafür, denn sie gibt uns in der ersten Zeile eine Rangordnung der möglichen Kandidaten: Autor 2 ist auf dem ersten Rang,

Autor 4 auf dem zweiten, und Autor 3 auf dem dritten. Wenn man diese Distanzmatrix als Maximum-Likelihood-Klassifikator anwenden möchte, dann gibt sie uns die Antwort «Autor 2». Zu dieser Antwort ist sie gekommen, weil von den drei Datenpunkten/Dokumenten 2-4 die Klasse/Autor bekannt ist. Der Input ist also gelabelt, und wir erhalten durch Klassifizierung einen Output – dies ist wahrscheinlich der Grund, weshalb die `stylo`-Entwickler das Delta als Supervised ML bezeichnen. Wer das Distanzmass Delta als Lerner bezeichnet, geht implizit von einem Studiendesign aus, und bezeichnet sämtliche Klassifikationen als Machine Learning. Dies ist auch die Erklärung dafür, dass Verfahren wie Clusteranalyse und Principal Components Analysis von den `stylo`-Entwickler nicht als Machine Learning bezeichnet werden: Es findet keine Klassifikation statt, sondern es werden Klassengrenzen etabliert in den zuvor unstrukturierten Daten. Dass dies aber durchaus zu Machine Learning gehört, zeigt ein kurzer Blick in Einführungswerke aus den exakten Wissenschaften (Mohri et al. 2018, 3,347-348; Hastie et al. 2009, S. 501–550; Alpaydın 2010, S. 109–161).

2.7 Kreuzvalidierung

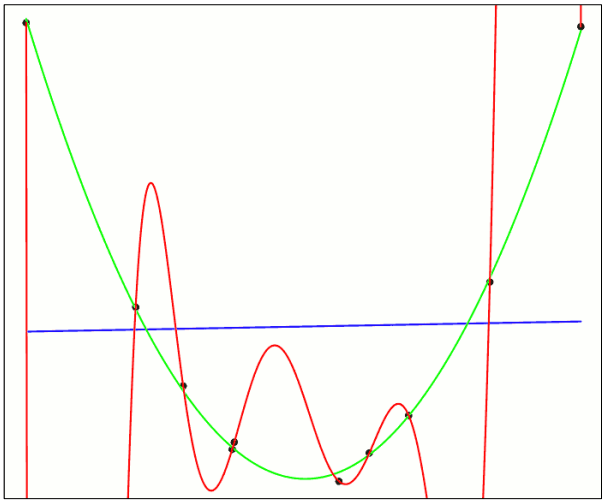

Abb. 4: Die durch die grüne Linie generierten Datenpunkte werden weder durch das blaue Modell adäquat beschrieben (Unteranpassung), noch durch das rote Modell (Überanpassung) (https://commons.wikimedia.org/wiki/File:Interpolation.png).

Wie stellt man fest, ob ein gelerntes Modell die Wirklichkeit adäquat beschreibt? Abb. 4 zeigt das Problem von Modellen auf: Angenommen, die Datenpunkte wurden durch den grünen Graphen generiert. Basierend auf diesen Datenpunkten als Trainingsdaten wurden dann zwei Modelle trainiert, nämlich der blaue und der rote Graph. Das blaue Modell ist zu simpel, es ist unterangepasst an die Daten. Weder vermag es die Trainingsdaten zu beschreiben, noch würde es neue Daten (beliebige Punkte auf der grünen Linie) adäquat voraussagen. Beim roten Modell ist das Gegenteil der Fall, nämlich Überanpassung an die Trainingsdaten. Die Trainingsdaten werden durch ein zu ausgeklügeltes Modell zwar perfekt erklärt, aber neue Daten würden trotzdem falsch vorausgesagt.

An dieser Stelle ist ein genauerer Blick auf die mathematischen Grundlagen nötig. Man kann im ML drei verschiedene Ebenen unterscheiden:

- Modell: Durch den Menschen ausgewählte mathematische Funktion, welche die Daten beschreibt, z.B. lineare Regression, oder polynomische Interpolation.

- Parameter: Die Parameter der mathematischen Funktionsgleichung, welche durch die Trainingsdaten optimiert werden.
- Hyperparameter: Je nach Modell gibt es durch den Menschen festzusetzende Werte, welche den Lernprozess kontrollieren, z.B. den Grad der polynomischen Interpolation.

In Abb. 4 stellt der blaue Graph das Modell «lineare Regression» dar. Die dazugehörige Funktionsgleichung

(1) $$y = m \cdot x + n$$

hat die beiden Parameter m und n. Das heisst, dass sich jede beliebige Linie im zweidimensionalen Raum ausdrücken lässt durch eine solche Gleichung, wobei die Werte von m und n die Richtung und Lage der Linie bestimmen. Wenn man nun mit den Trainings-Datenpunkten ein lineares Regressionsmodell trainiert, macht man nichts anderes, als die Werte für m und n derart zu bestimmen, dass der blaue Graph eine minimale Distanz zu den Datenpunkten hat. Der Grund für das schlechte Resultat ist, dass das Modell «lineare Regression» eine schlechte Wahl war.

Eine bessere Wahl wäre das Modell «polynomische Interpolation». Die dazugehörige Funktionsgleichung hat die Form

(2) $$y = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \cdots + a_2 \cdot x^2 + a_1 \cdot x + a_0$$

Dabei sind a_n … a_0 die Parameter, welche die Lage und den genauen Verlauf des Graphen bestimmen. Sie werden durch die Trainingsdaten optimiert. Die Anzahl der Parameter hingegen, also der Wert von n, ist ein Hyperparameter. Er bestimmt den Grad der polynomischen Interpolation, und muss durch einen Menschen festgelegt werden. In Abb. 4 ist der rote Graph ein Interpolationspolynom der Ordnung 9 ($n = 9$). Es vermag die Trainingsdaten sehr gut zu beschreiben, aber versagt beim Vorhersagen von neuen Daten. Diesmal liegt der Fehler weder beim Modell, noch beim Trainieren der Parameter, sondern bei der Wahl des Hyperparameters. Denn der grüne Graph ist ein quadratisches Interpolationspolynom mit $n = 2$.

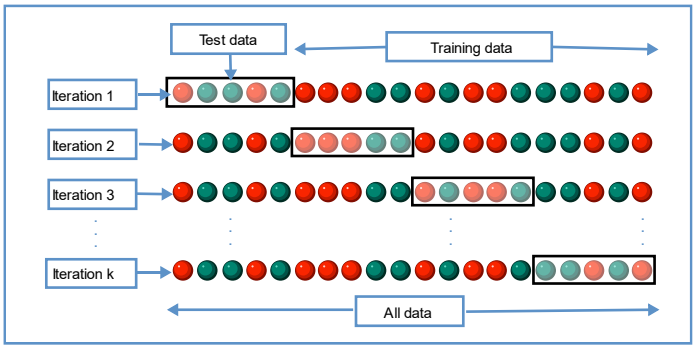

Abb. 5: K-fache Kreuzvalidierung mit k=4 (https://commons.wikimedia.org/wiki/File:K-fold_cross_validation_EN.svg).

Wie ist nun mit diesem Problem umzugehen? Nachdem ein Modell ausgewählt und die dazugehörigen Hyperparameter gesetzt wurden, und die Parameter mithilfe der Trainingsdaten optimiert wurde, könnte es immer noch sein, dass das Modell eine schlechte Performanz hat. Diesem Problem kann Abhilfe geschaffen werden durch die Kreuzvalidierung, mit welcher sich die Performanz einer bestimmten Modell-Hyperparameter-Kombination abschätzen lässt. Dabei werden die Trainingsdaten nicht ausschliesslich dazu verwendet, die Parameter der Modell-Hyperparameter-Kombination zu optimieren, sondern auch, um diese Kombination zu testen. Dies kann auf unterschiedliche Weise geschehen, wobei die 10-fache Kreuzvalidierung wohl die populärste Form darstellt. Dabei werden die zur Verfügung stehenden Daten in zehn gleich grosse Stücke aufgeteilt. In zehn Durchgängen (Iterationen) werden jeweils neun Teile als Trainingsdaten dazu verwendet, die Parameter einer spezifischen Modell-Hyperparameter-Konstellation zu optimieren. Anschliessend wird diese Konstellation am zehnten Teil getestet: Wenn die Vorhersage übereinstimmt mit den tatsächlichen Daten, wird dies als Erfolg gewertet, ansonsten als Misserfolg. Der Anteil von erfolgreichen Durchgängen erlaubt eine Einschätzung, ob diese Kombination von Modell und Hyperparametern robust ist, und mit unbekannten Daten gut funktionieren wird.

2.8 Studiendesign

Je nach Fragestellung lassen sich verschiedene AA Probleme unterscheiden. Wenn die Kandidatengruppe abgeschlossen ist, also wenn nur eine kleine Anzahl von Autoren infrage kommt, spricht man von Authorship Attribution. Wenn hingegen ein unbekannter Autor den Text geschrieben haben könnte, also wenn die Kandidatengruppe offen ist, handelt es sich um Authorship Verification, was anspruchsvoller ist. Bei den Paulusbriefen handelt es sich um eine Authorship Verification, weil die Frage ist, ob die umstrittenen Paulusbriefe eher von Paulus oder eher von einem anonymen Autor geschrieben wurden. Zu diesem Zweck wurde das Imposters-Studiendesign entworfen (General Imposters Framework), das wie folgt definiert wird: “[t]he general intuition behind the GI, is not to assess whether two documents are simply similar in writing style, given a static feature vocabulary, but rather, it aims to assess whether two documents are significantly more similar to one another than other documents, across a variety of stochastically impaired feature spaces, and compared to random selections of so-called distractor authors, also called ‘imposters’.” (Kestemont et al. 2016, S. 88)

Anders ausgedrückt heisst das, dass nicht nur die gesicherten Paulusbriefe mit den umstrittenen verglichen werden, sondern dass eine dritte Vergleichsgruppe dazu tritt, die sogenannten Hochstapler (imposters). Dies ist nötig, um eine belastbare Aussage darüber zu treffen, ob die umstrittenen Paulusbriefe den echten signifikant ähnlicher sind als z.B. den Evangelien. Diese Berechnung wird aber nicht ein einziges Mal durchgeführt, sondern mehrmals, mit je einer unterschiedlichen Anzahl von zufällig ausgewählten Textmerkmalen und Vergleichstexten. Auf den ersten Blick mag es verwunderlich erscheinen, warum nicht eine einzige Berechnung mit der Gesamtheit der Textmerkmale und Vergleichstexte durchgeführt wird. Doch diese einzige Berechnung würde daran kranken, dass es keine Garantie dafür gibt, dass die zur Verfügung stehenden Textmerkmale und Vergleichstexte repräsentativ für die Grundgesamtheit sind. In *2.3 Textmerkmale* auf S. 8f. wurde besprochen, dass unterschiedliche MFW-Fenster eine unterschiedlich gute Performanz aufweisen. Deshalb kann die Fehlerrate drastisch verkleinert werden, wenn in einem Bootstrapping-Verfahren verschiedene MFW-Bereiche zufällig ausgewählt werden, und am Schluss der Durchschnitt aus allen Berechnungen genommen wird. Dasselbe gilt für die Hochstapler. Eine saubere mathematische Formalisierung dieser Überlegung ist zu finden in Juola 2015, i107.

Das R-Package `stylo` implementiert das General Imposters Framework in der Methode `imposters()`, welche folgendermassen funktioniert: Zuerst werden die vier Hyperparameter Distanzmass, Anzahl Iterationen, prozentualer Anteil der Textmerkmale (features) und prozentualer Anteil der Imposter-Texte festgelegt. Die Wahl könnte beispielsweise auf folgende Werte fallen: Cosinus-Distanz, 100 Iterationen, 50% der Textmerkmale und 50% der Imposter-Texte. In der ersten Iteration werden dann in einem Zufallsverfahren 50% der Textmerkmale und 50% der Imposter-Texte ausgewählt. Dann werden anhand der ausgewählten 50% der Textmerkmale sämtliche Cosinus-Distanzen des zu untersuchenden Textes zu allen Texten des vermuteten Autors und zu den ausgewählten 50% der Imposter-Texte berechnet. Falls der nächste Nachbar des zu untersuchenden Textes ein Imposter ist, lautet das Resultat dieser Iteration «anderer Autor». Ist hingegen der nächste Nachbar ein Text des vermuteten Autors, so lautet das Resultat dieser Iteration «vermuteter Autor». Angenommen, dass 90 der 100 Iterationen «vermuteter Autor» liefern, dann liegt die Wahrscheinlichkeit, dass der vermutete Autor den zu untersuchenden Text geschrieben hat, bei 90% (Kestemont et al. 2016, S. 88).

Eine Variation dieser Methode besteht darin, den vermuteten Autor nicht von Beginn weg als solchen zu bezeichnen, sondern ihn unter die Hochstapler zu mischen. In diesem Fall nimmt reihum jeder Autor aus dem Set einmal die Rolle des vermuteten Autors ein, und wird gegen die anderen Autoren getestet, gemäss dem oben beschriebenen Verfahren. Daraus resultiert für jeden Autorkandidat eine Wahrscheinlichkeit. Falls einer der Kandidaten eine erheblich höhere Wahrscheinlichkeit erhält als alle anderen, handelt es sich um den tatsächlichen Autor. Interessanterweise beträgt bei diesem Vorgehen

die Summe der Wahrscheinlichkeiten aller Kandidaten nicht 100%. Der Grund dafür ist das stochastische Vorgehen: In jeder Iteration wird zufällig ein Teil der Textmerkmale und der Hochstapler ausgewählt. Deswegen kann es vorkommen, dass ein Hochstapler in «seinem» Durchlauf 20% der Stimmen erhält, aber der echte Autor in «seinem» Durchlauf 95%.

3 Design der vorliegenden Studie

Bei der Frage, ob Paulus der Autor der umstrittenen Paulusbriefe ist, handelt es sich um ein Authorship Verification-Problem, welches am besten mit dem General Imposters Framework behandelt werden kann. Momentan steht für das General Imposters Framework in `stylo` nur die Delta-Methode zur Verfügung, d.h. die `imposters()`-Methode kann nur mit Distanzmassen ausgeführt werden. Die Implementierung von ML-Methoden wie SVM, NSC, kNN oder Naive Bayes ist noch ausstehend. Oben wurde beschrieben, dass die Methode vier Hyperparameter benötigt. Doch wie kann man herausfinden, welche Werte diese Hyperparameter haben sollen? Um dies herauszufinden, stellt das `stylo`-Package die Funktion `imposters.optimize()` zur Verfügung. Wenn man diese Funktion mit einer bestimmten Textrepräsentation der Trainingstexte und einer bestimmten Kombination aus Hyperparametern aufruft, errechnet sie die beiden Werte p1 und p2. Diese Werte liegen zwischen 0 und 1 und begrenzen dasjenige Intervall, in dem die Resultate der `imposters()`-Methode unzuverlässig sein werden, wenn man sie mit genau derselben Kombination aus Textrepräsentation und Hyperparametern aufruft. Wenn beispielsweise die Wahrscheinlichkeit, dass Paulus den 1. Timotheusbrief geschrieben hat, p = 0.9 beträgt, aber das Unsicherheitsintervall [p1 = 0.3, p2 = 0.95] ist, so ist das vermeintlich klare Resultat wertlos.

Aus diesem Grund wurden für die vorliegende Studie in einem aufwändigen Suchverfahren diejenigen Textrepräsentationen und Hyperparameter bestimmt, welche erfolgversprechend sind, weil sie ein kleines Unsicherheitsintervall haben. Dazu wurde eine Tabelle erstellt mit einigen Kombinationsmöglichkeiten der Textrepräsentationen mit möglichen Werten für die Hyperparameter. Dann wurden für alle diese Kombinationen das Unsicherheitsintervall bestimmt. Aufgrund von beschränkter Rechenkapazität wurde der Suchraum aufseiten der Textrepräsentationen beschränkt auf {1,2,3}-Gramme griechischer Wörter, {1,2,3}-Gramme griechischer Buchstaben, {1,2,3}-Gramme der Strong-Nummern, {1,2,3}-Gramme der ersten Elemente der POS-Tags, und 1-Gramme der gesamten POS-Tags. Diese 13 Textrepräsentationen wurden kombiniert mit den folgenden Hyperparametern: Distanzmasse Cosinus, Entropie und Canberra; Anteile der Imposter-Texte von 10%, 25%, 50%, 75% und 90%, und dieselben Anteile für die Auswahl der Textmerkmale, was 3 * 5 * 5 = 75 Kombinationen gibt. Es musste also je ein Unsicherheitsintervall errechnet werden für 13 * 75 = 975 Textrepräsentation-Hyperparameter-Kombinationen. Weil die Berechnung auf zufällig ausgewählten Texten und Repräsentationen beruht, liefern zwei nacheinander ausgeführte Berechnungen mit exakt derselben Kombination aber

kein identisches Resultat. Deshalb wurden alle Berechnungen sechs Mal durchgeführt, und anschliessend das arithmetische Mittel genommen, was die Anzahl Berechnungen auf 6 * 975 = 5850 erhöhte. Die Berechnung der Unsicherheitsintervalle in diesem Suchraum betrug mehrere Wochen auf einem (und zeitweise sogar zwei) handelsüblichen Laptops. Schlussendlich wurden aus allen 975 Textrepräsentation-Hyperparameter-Kombinationen nur diejenigen 46 berücksichtigt, welche ein Unsicherheitsintervall von weniger als 0.2 haben, mit einer Obergrenze von kleiner als 0.75. Die oben beschriebene `imposters()`-Methode wurde also 46 Mal ausgeführt, und entsprechend resultierte für jede Testtext-Autorkandidat-Kombination 46 Einschätzungen, mit welcher Wahrscheinlichkeit dieser Autorkandidat der tatsächliche Autor des Testtextes ist. Von diesen Einschätzungen wurden nur diejenigen behalten, welche ausserhalb ihres jeweiligen Unsicherheitsintervalls liegen.

Zur Beantwortung der vorliegenden Frage ist es nötig, drei Textgruppen zu definieren: Die sicheren Paulusbriefe P_S, die Vergleichstexte/Imposters I und die umstrittenen Paulusbriefe P_{US}. Dabei ist es unvermeidlich, eine Annahme zu treffen darüber, welche Paulusbriefe für echt gehalten werden. Trotz aller Uneinigkeit in der neutestamentlichen Forschung kann doch die Sieben-Briefe-Hypothese als eine relativ breit abgestützte Meinung angenommen werden. Es sollen also für die vorliegende Arbeit folgende Definitionen gelten:

Trainingsdaten		
Gruppe	**Dokument**	**Autor**
Sichere Paulusbriefe (P_S)	Römer (ohne 16,25-27)	Paulus
	1. Korinther	Paulus
	2. Korinther	Paulus
	Galater	Paulus
	Philipper	Paulus
	1. Thessalonicher	Paulus
	Philemon	Paulus
Impostors (I)	Matthäus 1-14	Matthäus
	Matthäus 15-28	Matthäus
	Markus 1-8	Markus
	Markus 9-16,8	Markus
	Lukas	Lukas
	Johannes (ohne 7,53–8,11)	Johannes
	Apostelgeschichte	Lukas
	1. Johannes	Johannes
	2. Johannes	Johannes
	3. Johannes	Johannes
	Offenbarung	Johannes

Tab. 5: Trainingsdaten, bestehend aus Texten mit gesicherter Autorschaft. Anhand dieser Texte «lernt» der Computer den Stil der fünf verschiedenen Autoren.

Testdaten	
Gruppe	**Dokument**
Umstrittene Paulusbriefe (P_{US})	Epheser
	Kolosser
	2. Thessalonicher
	1. Timotheus
	2. Timotheus
	Titus
	Hebräer
Textkritisch umstrittene Passagen (TUP)	Markus 16,9-20
	Römer 16,25-27
	Johannes 7,53–8,11

Tab. 6: Testdaten, bestehend aus Texten, deren Autorschaft zu bestimmen ist.

Diese Definitionen sind natürlich nicht perfekt, aber zumindest taugen sie für den vorliegenden Zweck. Dazu sind einige Bemerkungen nötig: Erstens sind die textkritisch umstrittenen Passagen nicht essentiell für die vorliegende Studie, sondern wurden aus blosser Neugierde mit eingeschlossen. Sie sind zu kurz, um signifikante Resultate zu erwarten. Zweitens mussten einige Texte in zwei Hälften gespaltet werden, weil einige Machine Learning-Verfahren darauf angewiesen sind, mindestens zwei Dokumente pro Autor zu haben. Aus diesem Grund wurden auch der Jakobus- und der Judasbrief ausgeschlossen, weil sie zu kurz sind, um aufgespalten zu werden. Die drei Petrusbriefe gingen durch Nachlässigkeit des Autors vergessen im R-Code, was leider erst bemerkt wurde, als sämtliche Analysen bereits fertig durchgeführt waren. Drittens mag es als unreflektiert erscheinen, dass die nicht-paulinischen Texte des Trainingskorpus ihren traditionellen Autoren zugeschrieben werden (z.B. die Offenbarung dem Johannes). Dies ist aber irrelevant für die Autorschaft der Paulusbriefe. Schwerer wiegt der Umstand, dass die Hochstapler grösstenteils einer anderen Textgattung angehören als die Paulusbriefe, es sind nämlich hauptsächlich die Evangelien und die Apostelgeschichte, deren Gattung vielleicht als literarische Biographie/Bericht bezeichnet werden könnte. Andererseits haben die Hochstapler auch viele Gemeinsamkeiten mit den Paulusbriefen: Beides sind christliche literarische Texte mit demselben Thema, die im 1. Jh. n.Chr. im oströmischen Reich entstanden sind, und zwar in einer genau definierten Untergruppe der hellenisierten Juden, nämlich der allerersten Christen. Deshalb sind signifikante Resultate zu erwarten, unabhängig von der Textgattung.

4 Resultate

Welche Art von Resultate liefert nun die `imposters()`-Methode, und wie sind sie zu deuten? Im Grunde geht es darum, für jeden Testtext eine Einschätzung vorzunehmen, mit welcher Wahrscheinlichkeit er von einem derjenigen Autoren geschrieben wurde, von welchen die Trainingstexte geschrieben wurden. Wie bereits beschrieben wurde die `imposters()`-Methode 46 Mal ausgeführt, jedes Mal mit einer anderen Kombination von Textrepräsentationen und Hyperparametern. Da die Trainingstexte von fünf Autorkandidaten stammen, und zehn Testtexte zu bestimmen sind, ergeben sich 50 Testtext-Autorkandidat-Kombinationen. Für jede dieser Kombinationen gibt es 46 Einschätzungen, mit welcher Wahrscheinlichkeit dieser Autorkandidat der tatsächliche Autor des Testtextes ist. Um diese Zahlenmenge verständlich aufzubereiten, wurde eine Visualisierung mit Boxplots gewählt. Für jeden der zehn Testtexte resultierten fünf Boxplots, welche die Streuung der Wahrscheinlichkeiten für jeden der fünf Autorkandidaten darstellen.

Lesehilfe: Die 46 Einschätzungen, mit welcher Wahrscheinlichkeit Johannes den 1. Timotheusbrief geschrieben hat, sind verteilt zwischen 0 und ca. 0.8. Der Median aller Einschätzungen liegt bei ca. 0.1, wobei die Hälfte aller Einschätzungen zwischen 0.05–0.15 liegen. Die kleinen Kreise sind Datenpunkte,

die besonders weit abweichen von den übrigen. Dort, wo es keine kleinen Kreise hat, liegen alle Datenpunkte innerhalb der gestrichelten Linien.

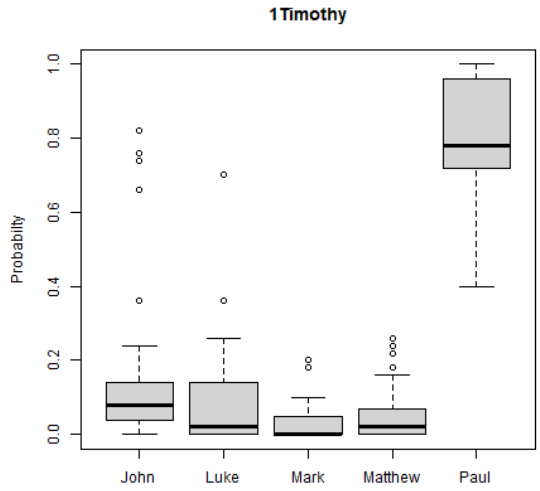

*Abb. 6: **1. Timotheus-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten.*

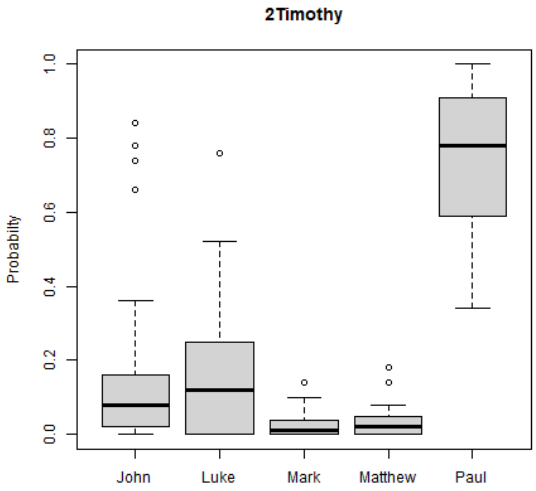

*Abb. 7: **2. Timotheus-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten.*

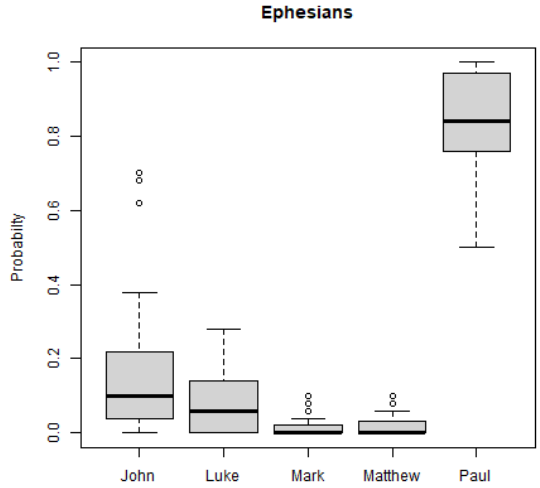

*Abb. 8: **Epheser-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten.*

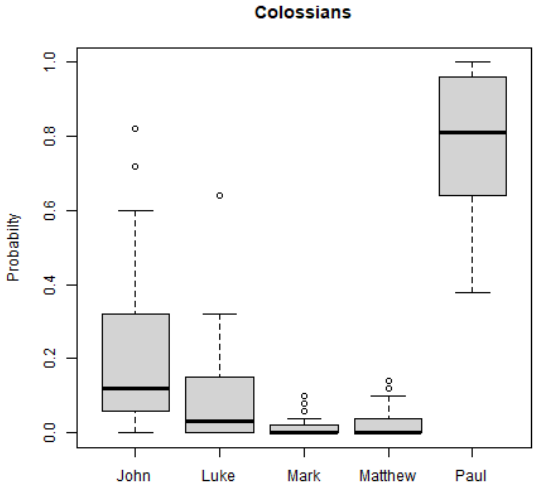

*Abb. 9: **Kolosser-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten.*

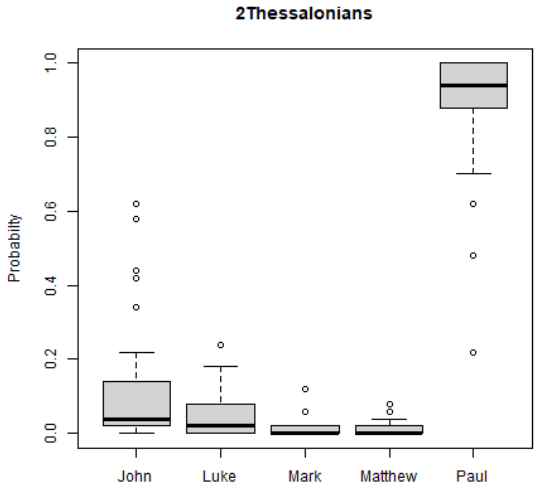

*Abb. 10: **2. Thessalonicher-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten.*

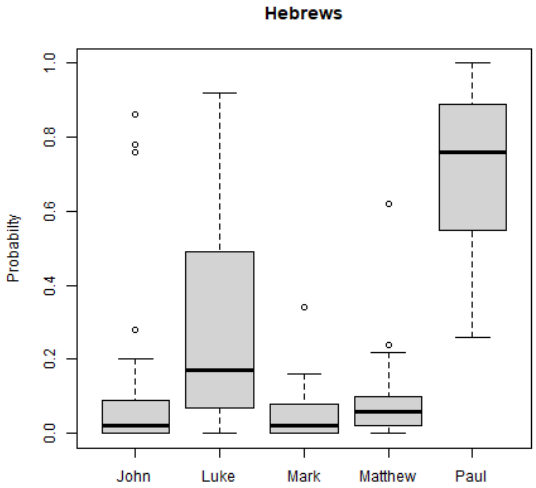

*Abb. 11: **Hebräer-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten.*

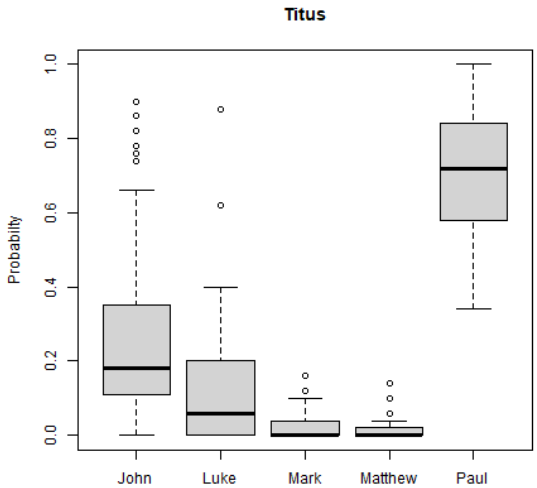

*Abb. 12: **Titus-Brief**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten*

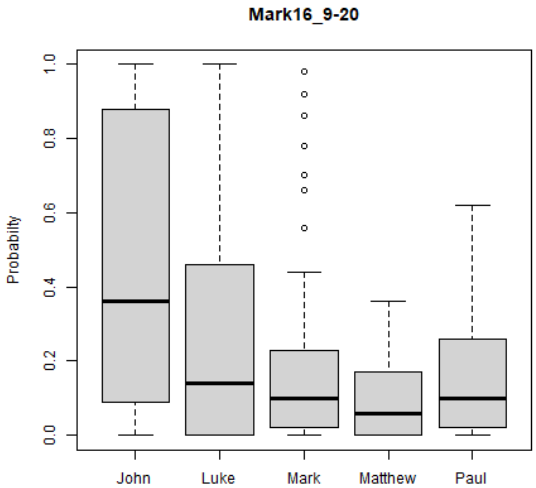

*Abb. 13: **Langer Markusschluss**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten*

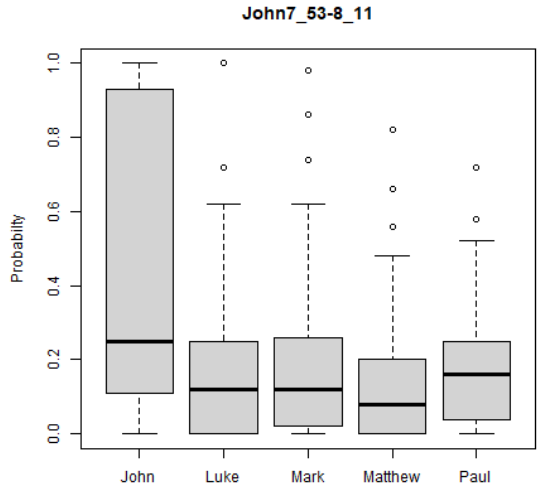

*Abb. 14: **Pericope adulterae**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten*

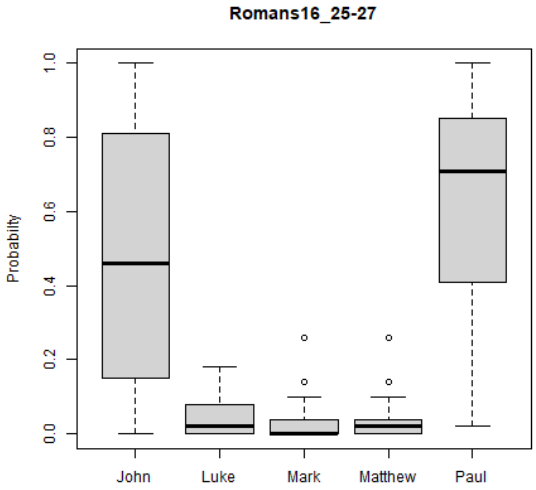

*Abb. 15: **Langer Römer-Schluss**, Überblick über die Resultate aller 46 Iterationen, die je eine Wahrscheinlichkeit pro Autorkandidat lieferten*

5 Diskussion

Die Resultate besagen mit überraschender Klarheit, dass die umstrittenen Paulusbriefe mit einer sehr viel höheren Wahrscheinlichkeit von Paulus verfasst wurden als von einem anderen Autor des Neuen Testaments. Ist daraus nun zu schliessen, dass man diese Briefe wirklich dem Apostel zuschreiben sollte? Dagegen gibt es mehrere Einwände:

Erstens könnte man geltend machen, dass niemand ernsthaft behauptet, dass einer der vier Evangelisten die umstrittenen Paulusbriefe geschrieben hat. Doch das ist gerade der springende Punkt des General Imposters Framework: Die Hochstapler sollen Texte sein, die dem zu untersuchenden Text zwar ähnlich sind, aber von anderen Autoren stammen. Es wäre sogar fatal, unwissentlich Texte vom tatsächlichen Autor unter den Hochstapler-Texten zu haben. Dies ist explizit so definiert in der Dokumentation und in der Literatur (Eder 2018; Kestemont et al. 2016, S. 88–89).

Ein zweiter Einwand könnte lauten, dass nicht die Autorschaft, sondern die Textgattung und das Thema zu diesem Resultat geführt haben. Es gibt aber viele Studien, auch sogenannte Benchmark-Studien, welche die existierenden AA-Methoden an Korpora testen, von denen der Autor bekannt ist. Es ist Konsens in der AA-Community, dass die Worthäufigkeiten von Funktionswörtern themen- und gattungsunabhängig sind, und auch nicht bewusst verändert werden können durch den Autor. Dieser Punkt bedarf aber weiterer Klärung. Eine Möglichkeit, diese Bedenken auszuräumen, wäre eine feinere Unterteilung der Autoren der Trainingstexte: Man könnte die Johannesbriefe, die Petrusbriefe, den Jakobus- und Judasbrief als je eigenständige Autorklassen definieren, um mehr Vergleichsmaterial aus der Gattung der Briefe zu haben. Dann wäre eine Aussage möglich, inwiefern sich die umstrittenen

Paulusbriefe von den anderen neutestamentlichen Briefen unterscheiden, anstatt sie nur mit sehr Evangelien-lastigen Autorenklassen vergleichen zu können. Eine andere Möglichkeit ist die von Stamatatos 2018 vorgeschlagene Methode, die Texte in eine thematisch neutrale Form zu transformieren, ohne dabei das Autorenprofil zu schädigen.

Schwerwiegender ist allerdings ein dritter Einwand, nämlich: Sind die Werte für Paulus genug hoch, um signifikant zu sein? Wie klar muss sich Paulus von den anderen Autorkandidaten abheben, um als Autor zu gelten? Es ist nämlich denkbar, dass die betroffenen Dokumente aus P_{US} von einem Autor X stammen, dessen Stil näher an Paulus' Stil liegt als an den Stilen der anderen neutestamentlichen Autoren. Juola 2015, i107-i109 hat spannende Ansätze dazu, aber am Schluss hat man das Problem, dass man nicht weiss, ob die Wahrscheinlichkeiten unabhängig voneinander sind, oder abhängig voneinander, und man weiss auch nicht, ob das eine Rolle spielt. Und man muss eine Erfolgsrate für den einzelnen Test schätzen (den man aber durch Kreuzvalidierung herausfinden könnte). Man könnte Juolas Ansatz auch noch verbessern, z.B. indem man nicht nur die Rangfolge berücksichtigt, sondern wie weit die Kandidaten auseinanderliegen (S. i111). Juola ortet hier Forschungsbedarf.

Ausserdem muss mit der Möglichkeit gerechnet werden, dass der Schreibstil des Paulus stark variiert, oder dass sein Schreibgehilfe im Diktierprozess einen starken Einfluss auf die Formulierung genommen hat. Dennoch hat die vorliegende Studie das Potential, das breite bibelwissenschaftliche Argumentarium um ein weiteres Einzelargument bereichern. Grundsätzlich ist also festzuhalten, dass AA nicht eine abschliessende Wahrheit finden kann, sondern nur neue Argumente in einer komplexen Diskussion.

6 Fazit und Ausblick

Die vorliegende Studie hat wertvolle Einsichten und Resultate zutage gefördert: Eine davon ist, dass eine gründliche Auseinandersetzung mit Konzepten und mathematischen Zusammenhängen unerlässlich ist. Dies zeigt sich beispielsweise an Burrows, einem Pionier der Authorship Attribution, dessen Verdienste hier nicht geschmälert werden sollen. Doch bei ihm zeigt sich exemplarisch, wobei die Gefahren bestehen, wenn Geisteswissenschaftler ohne formale Ausbildung in den exakten Wissenschaften sich auf das neue Terrain der Digital Humanities begeben. Burrows bezeichnete in seinen frühen Publikationen das Delta als das oberste Häufigkeitsspektrum von Textmerkmalen, aber gleichzeitig auch als Distanzmass und auch eine Möglichkeit, einen unbekannten Text einem Autor zuzuordnen (Burrows 2002, 2007). Damit vermischt er die drei Konzepte Textmerkmal, Distanzmass und Studiendesign. Zwar konnte er trotzdem valide Resultate produzieren. Aber ein mangelndes Verständnis der Vorgänge führt dazu, dass digitale Methoden als eine wundersame Blackbox gelten, deren Resultate nicht genügend reflektiert werden. Wie das Kapitel *5 Diskussion* aber gezeigt hat, ist es dringend

notwendig, genau erklären zu können, was die eigenen Resultate bedeuten, was sie können, und was sie nicht können. Abgesehen davon stiftet die unklare Verwendung von Begriffen eine nicht unerhebliche Verwirrung, wie z.B. die stillschweigende Gleichsetzung von Machine Learning mit Supervised Machine Learning im R-Package `stylo`.

Der Verfasser zählt sich aber selbst auch zur Gruppe von Geisteswissenschaftlern, die erst nochmals gründlich über die (Statistik-)Bücher müssen. Das Kapitel *2 Theoretische Grundlagen* ist noch unausgereift, so werden z.B. bei Machine Learning und Kreuzvalidierung nur die parametrischen Methoden erklärt. Die nicht-parametrischen Methoden sind aber fast noch wichtiger für die Authorship Attribution, denn bei der Delta-Methode handelt es sich im Grunde genommen um ein kNN-Verfahren. Eine weitere Wissenslücke des Verfassers hat sich in *5 Diskussion* gezeigt, nämlich die Unklarheit darüber, ob die Resultate überhaupt signifikant sind. Vielleicht liesse sich für jede Einschätzung ein p-Wert errechnen, also die Wahrscheinlichkeit, dass ein x-beliebiger Autorkandidat rein zufällig auf dasselbe Resultat wie Paulus kommt. Von diesem p-Wert kann dann klar gesagt werden, ob er sich über oder unter einem akzeptablen Signifikanzniveau befindet.

Trotzdem haben die Resultate zu den umstrittenen Paulusbriefen das Potential, ein innovativer Beitrag in der theologischen Fachwelt zu werden, denn die Resultate sind sowohl unerwartet als auch empirisch. Um einer soliden Überprüfung standzuhalten, sind allerdings noch einige Verbesserungen nötig. Dazu gehört sicher eine Auseinandersetzung mit dem Problem der zu kleinen Stichproben. Hierzu existieren glücklicherweise viele Forschungsarbeiten, weil ein juristisches und ökonomisches Interesse daran besteht, problematische Social Media-Posts einem Autor zuordnen zu können, auch wenn sie kurz sind. Einige Arbeiten dazu wären Rebora et al. 2019; Eder 2015, 2017; Mikros und Perifanos 2013; Rocha et al. 2017, aber besonders auch die konkreten Lösungsvorschläge von Hinneburg et al. 2007. Des Weiteren gibt es noch deutlichen Verbesserungsbedarf bei der Gewinnung von Textmerkmalen, z.B. automatische Gewinnung von morphologischen Daten mittels UDPipe (Straka und Straková 2017), Entropie (Juola 2005), Document Embeddings (Gómez-Adorno et al. 2018), pq-gram Indices von Parsing Trees (Tschuggnall und Specht 2016), oder die von Savoy 2013 und Savoy 2015 evaluierten feature selection-Methoden. Ebenfalls müsste – bei allen diesen Textmerkmalen – geprüft werden, ob gewisse Teilbereiche der Häufigkeitstabelle besser performen als die gesamte Tabelle. Aus pragmatischen Gründen arbeitete die vorliegende Studie nur mit der gesamten Tabelle. Des Weiteren wäre es prüfenswert, ob es nicht effizientere Möglichkeiten gibt, die optimalen Hyperparameter für die `imposters()`-Methode zu suchen. Aufgrund des immensen Rechenaufwands der Grid Search musste sich die vorliegende Studie auf einen relativ kleinen Suchraum beschränken, und viele Distanzmasse unberücksichtigt lassen. Ausserdem wäre es nur von Vorteil, das General Imposters Framework nicht nur mit einfachen Textdistanzen, sondern auch mit ML oder Deep Learning

(d.h. Neuronalen Netzwerken) durchzuführen (Schaalje et al. 2011), was momentan mit `stylo` aber noch nicht möglich ist. Eine Deep Learning Methode, für die bereits eine Implementation besteht, ist die Distributed Language Representation, welche die Zusammenhänge der Wörter und deren Kontexte berücksichtigt (Kocher und Savoy 2018; Posadas-Durán et al. 2017). Eine Benchmarking-Studie, die den Nutzen von verschiedenen Typen von Neuronalen Netzwerken für die AA untersucht, ist Schaetti und Savoy 2020.

Darüber hinaus gibt es aber auch noch andere Studiendesigns, für die eine Implementierung gefunden (oder selbst vorgenommen) werden könnte, z.B. das Unmasking, bei dem für jeden Autorkandidaten ein Modell gelernt wird, um den zu untersuchenden Text vom Autorkandidaten zu unterscheiden. Anschliessend werden schrittweise diejenigen Textmerkmale aus der Analyse ausgeschlossen, welche am ausschlaggebendsten sind für die Unterscheidung des zu untersuchenden Textes zum jeweiligen Autorkandidaten. Beim tatsächlichen Autor verschlechtert sich die Unterscheidungsfähigkeit rapide, sobald einige wenige ausschlaggebende Textmerkmale entfernt werden. Hingegen lassen sich die falschen Autorkandidaten noch lange sehr gut vom zu untersuchenden Text unterscheiden, auch wenn viele Textmerkmale ausgeschlossen wurden (Koppel et al. 2009, S. 18–21). Ein anderes Studiendesign, das durch seine Einfachheit besticht, ohne bei der Performanz Abstriche zu machen, ist das SPATIUM-L1. Dabei handelt es sich um eine Variation des Imposters Framework, bei dem die Manhattan-Distanz eines Textes zum vermuteten Autor im Verhältnis zu zufällig ausgewählten Hochstaplern betrachtet wird (Kocher und Savoy 2017).

Schlussendlich sollte nicht unterschätzt werden, dass ein Schlussresultat besser akzeptiert wird vom Endnutzer, wenn er zusammen mit dem Resultat eine Einschätzung erhält, wie zuverlässig das Resultat ist. Dazu gibt es verschiedenste Möglichkeiten, z.B. recall/precision-Raten (Koppel et al. 2009, S. 17–18) oder die von Savoy 2016 vorgeschlagene Methode. Nicht zuletzt ist eine genauere Würdigung der bisherigen AA-Arbeiten an biblischen Texten wünschenswert, wie z.B. Alviar 2008; Libby 2016; Royal 2012; Savoy 2019; Tschuggnall und Specht 2016.

7 Literaturverzeichnis

Alpaydın, Ethem (2010): Introduction to Machine Learning. 2. Aufl. Cambridge (MA): MIT Press.

Alviar, J. José (2008): Recent Advances in Computational Linguistics and their Application to Biblical Studies. In: *New Testament Studies* 54 (01), S. 139–159. DOI: 10.1017/S0028688508000088.

Argamon, Shlomo (2008): Interpreting Burrows's Delta. Geometric and Probabilistic Foundations. In: *Literary and Linguistic Computing* 23 (2), S. 131–147. DOI: 10.1093/llc/fqn003.

Bengio, Yoshua; Courville, Aaron; Vincent, Pascal (2013): Representation learning. A review and new perspectives. In: *IEEE transactions on pattern analysis and machine intelligence* 35 (8), S. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

Burrows, John (2002): Delta. A Measure of Stylistic Difference and a Guide to Likely Authorship. In: *Literary and Linguistic Computing* 17 (3), S. 267–287.

Burrows, John (2007): All the Way Through. Testing for Authorship in Different Frequency Strata. In: *Literary and Linguistic Computing* 22 (1), S. 27–47. DOI: 10.1093/llc/fqi067.

Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018): Statistics versus machine learning. In: *Nature methods* 15 (4), S. 233–234. DOI: 10.1038/nmeth.4642.

Eder, Maciej (2011): Style-Markers in Authorship Attribution. A Cross-Language Study of the Authorial Fingerprint. In: *Studies in Polish Linguistics* 6, S. 99–114.

Eder, Maciej (2015): Does size matter? Authorship attribution, small samples, big problem. In: *Digital Scholarship in the Humanities* 30 (2), S. 167–182. DOI: 10.1093/llc/fqt066.

Eder, Maciej (2017): Short samples in authorship attribution. A new approach. Online verfügbar unter https://dh2017.adho.org/abstracts/341/341.pdf.

Eder, Maciej (2018): Authorship verification with the package stylo. Online verfügbar unter https://computationalstylistics.github.io/docs/imposters, zuletzt geprüft am 11.11.2020.

Eder, Maciej; Rybicki, Jan; Kestemont, Mike (2016): Stylometry with R. A Package for Computational Text Analysis. In: *The R Journal* 8 (1), S. 107–121.

Eder, Maciej; Rybicki, Jan; Kestemont, Mike (2018): 'Stylo'. A package for stylometric analyses. Online verfügbar unter https://github.com/computationalstylistics/stylo_howto/blob/master/stylo_howto.pdf.

Evert, Stefan; Proisl, Thomas; Jannidis, Fotis; Reger, Isabella; Pielström, Steffen; Schöch, Christof; Vitt, Thorsten (2017): Understanding and explaining Delta measures for authorship attribution. In: *Digital Scholarship in the Humanities* 32 (suppl_2), ii4-ii16. DOI: 10.1093/llc/fqx023.

Gatto, L. (2020): An Introduction to Machine Learning with R. Abgerufen am 03.09.2020. Online verfügbar unter https://lgatto.github.io/IntroMachineLearningWithR.

Gómez-Adorno, Helena; Posadas-Durán, Juan-Pablo; Sidorov, Grigori; Pinto, David (2018): Document embeddings learned on various types of n-grams for cross-topic authorship attribution. In: *Computing* 100 (7), S. 741–756. DOI: 10.1007/s00607-018-0587-8.

Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009): The Elements of Statistical Learning. Data Mining, Inference, and Prediction. 2. Aufl. New York: Springer New York.

Hinneburg, A.; Mannila, H.; Kaislaniemi, S.; Nevalainen, T.; Raumolin-Brunberg, H. (2007): How to Handle Small Samples. Bootstrap and Bayesian Methods in the Analysis of Linguistic Change. In: *Literary and Linguistic Computing* 22 (2), S. 137–150. DOI: 10.1093/llc/fqm006.

Juola, Patrick (2005): A Controlled-corpus Experiment in Authorship Identification by Cross-entropy. In: *Literary and Linguistic Computing* 20 (Suppl 1), S. 59–67. DOI: 10.1093/llc/fqi024.

Juola, Patrick (2015): The Rowling Case. A Proposed Standard Analytic Protocol for Authorship Questions. In: *Digital Scholarship in the Humanities* 30 (Supplement 1), i100-i113. DOI: 10.1093/llc/fqv040.

Juola, Patrick; Mikros, George K.; Vinsick, Sean (2019): A comparative assessment of the difficulty of authorship attribution in Greek and in English. In: *Journal of the Association for Information Science and Technology* 70 (1), S. 61–70. DOI: 10.1002/asi.24073.

Kestemont, Mike; Stover, Justin; Koppel, Moshe; Karsdorp, Folgert; Daelemans, Walter (2016): Authenticating the writings of Julius Caesar. In: *Expert Systems with Applications* 63, S. 86–96. DOI: 10.1016/j.eswa.2016.06.029.

Kocher, Mirco; Savoy, Jacques (2017): A simple and efficient algorithm for authorship verification. In: *Journal of the Association for Information Science and Technology* 68 (1), S. 259–269. DOI: 10.1002/asi.23648.

Kocher, Mirco; Savoy, Jacques (2018): Distributed language representation for authorship attribution. In: *Digital Scholarship in the Humanities* 33 (2), S. 425–441. DOI: 10.1093/llc/fqx046.

Kocher, Mirco; Savoy, Jacques (2019): Evaluation of text representation schemes and distance measures for authorship linking. In: *Digital Scholarship in the Humanities* 34 (1), S. 189–207. DOI: 10.1093/llc/fqy013.

Koppel, Moshe; Schler, Jonathan; Argamon, Shlomo (2009): Computational methods in authorship attribution. In: *Journal of the American Society for Information Science and Technology* 60 (1), S. 9–26. DOI: 10.1002/asi.20961.

Libby, James A. (2016): The Pauline Canon Sung in a Linguistic Key. Visualizing New Testament Text Proximity by Linguistic Structure, System, and Strata. In: *BAGL* 5, S. 122–201. Online verfügbar unter 311923095_The_Pauline_Canon_Sung_in_a_Linguistic_Key_Visualizing_New_Testament_Text_Proximity_by_Linguistic_Structure_System_and_Strata.

Mikros, George K.; Perifanos, Kostas A. (2013): Authorship Attribution in Greek Tweets Using Author's Multilevel N-Gram Profiles. Analyzing Microtext: Papers from the 2013 AAAI Spring Symposium. Online verfügbar unter https://www.researchgate.net/publication/236583621_Authorship_attribution_in_Greek_tweets_using_author%27s_multilevel_N-gram_profiles.

Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2018): Foundations of machine learning. Second edition (Adaptive computation and machine learning).

Posadas-Durán, Juan-Pablo; Gómez-Adorno, Helena; Sidorov, Grigori; Batyrshin, Ildar; Pinto, David; Chanona-Hernández, Liliana (2017): Application of the distributed document representation in the authorship attribution task for small corpora. In: *Soft Computing* 21 (3), S. 627–639. DOI: 10.1007/s00500-016-2446-x.

Rebora, Simone; Herrmann, J. Berenike; Lauer, Gerhard; Salgaro, Massimo (2019): Robert Musil, a war journal, and stylometry. Tackling the issue of short texts in authorship attribution. In: *Digital Scholarship in the Humanities* 34 (3), S. 582–605. DOI: 10.1093/llc/fqy055.

Robinson, M. A. (undatiert): The Online Greek New Testament Parsings And Declensions For All Occurring Forms. Unpublizierter und undatierter Artikel, heruntergeladen am 17.08.2020. Online verfügbar unter https://0364f38b-a-3a69caf4-s-sites.googlegroups.com/a/wmail.fi/greeknt/home/greeknt/P-GK-ALL.PDF.

Rocha, Anderson; Scheirer, Walter J.; Forstall, Christopher W.; Cavalcante, Thiago; Theophilo, Antonio; Shen, Bingyu et al. (2017): Authorship Attribution for Social Media Forensics. In: *IEEE Transactions on Information Forensics and Security* 12 (1), S. 5–33. DOI: 10.1109/TIFS.2016.2603960.

Royal, Kenneth D. (2012): Using Stylometric Techniques to Evaluate New Testament Authorship. In: *Journal of Multidisciplinary Evaluation* 8 (19). Online verfügbar unter https://journals.sfu.ca/jmde/index.php/jmde_1/article/view/352/0.

Rybicki, Jan; Eder, Maciej (2011): Deeper Delta across genres and languages. Do we really need the most frequent words? In: *Literary and Linguistic Computing* 26 (3), S. 315–321. DOI: 10.1093/llc/fqr031.

Savoy, Jacques (2013): Feature selections for authorship attribution. In: Sung Y. Shin und José Carlos Maldonado (Hg.): Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13. the 28th Annual ACM Symposium. Coimbra, Portugal, 18.03.2013 - 22.03.2013. New York, New York, USA: ACM Press, S. 939.

Savoy, Jacques (2015): Comparative evaluation of term selection functions for authorship attribution. In: *Digital Scholarship in the Humanities* 30 (2), S. 246–261. DOI: 10.1093/llc/fqt047.

Savoy, Jacques (2016): Estimating the probability of an authorship attribution. In: *Journal of the Association for Information Science and Technology* 67 (6), S. 1462–1472. DOI: 10.1002/asi.23455.

Savoy, Jacques (2019): Authorship of Pauline epistles revisited. In: *Journal of the Association for Information Science and Technology* 70 (10), S. 1089–1097. DOI: 10.1002/asi.24176.

Schaalje, G. B.; Fields, P. J.; Roper, M.; Snow, G. L. (2011): Extended nearest shrunken centroid classification. A new method for open-set authorship attribution of texts of varying sizes. In: *Literary and Linguistic Computing* 26 (1), S. 71–88. DOI: 10.1093/llc/fqq029.

Schaetti, Nils; Savoy, Jacques (2020): Comparison of Visualisable Evidence-based Authorship Attribution Methods using Recurrent Neural Networks.

Schöch, Christof; Calvo, José; Zehe, Albin; Hotho, Andreas (undatiert): Burrows Zeta. Varianten und Evaluation. Online verfügbar unter http://www.dmir.uni-wuerzburg.de/projects/cligs/?tx_extbibsonomycsl_publicationlist%5BuserName%5D=dmir&tx_extbibsonomycsl_publicationlist%5BintraHash%5D=03d6baf7b43a7c8d4c2815c1f04c60d0&tx_extbibsonomycsl_publicationlist%5BfileName%5D=DHd2018_pyzeta.pdf&tx_extbibsonomycsl_publicationlist%5Baction%5D=download&tx_extbibsonomycsl_publicationlist%5Bcontroller%5D=Document&cHash=445fcd00f5e998175d79579553d8ca02.

Stamatatos, Efstathios (2018): Masking topic-related information to enhance authorship attribution. In: *Journal of the Association for Information Science and Technology* 69 (3), S. 461–473. DOI: 10.1002/asi.23968.

Stamatatos, Efstathios; Fakotakis, N.; Kokkinakis, G. (2001): Computer-Based Authorship Attribution without Lexical Measures. In: *Computers and the Humanities* 35 (2), S. 193–214. Online verfügbar unter https://www.jstor.org/stable/30204850.

Straka, Milan; Straková, Jana (2017): Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Jan Hajič und Dan Zeman (Hg.): Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Vancouver, Canada. Stroudsburg, PA, USA: Association for Computational Linguistics, S. 88–99.

Tschuggnall, Michael; Specht, Günther (2016): From Plagiarism Detection to Bible Analysis. The Potential of Machine Learning for Grammar-Based Text Analysis. In: Bettina Berendt, Björn Bringmann, Élisa Fromont, Gemma Garriga, Pauli Miettinen, Nikolaj Tatti und Volker Tresp (Hg.): Machine Learning and Knowledge Discovery in Databases. Cham: Springer International Publishing (9853), S. 245–248.

Zheng, Rong; Li, Jiexun; Chen, Hsinchun; Huang, Zan (2006): A framework for authorship identification of online messages. Writing-style features and classification techniques. In: *J. Am. Soc. Inf. Sci.* 57 (3), S. 378–393. DOI: 10.1002/asi.20316.