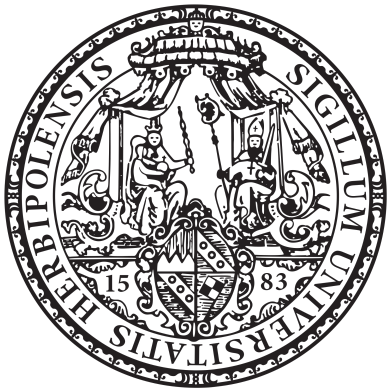
# Autorschaftsattribution mithilfe von Machine Learning Verfahren: Eine vergleichende Analyse

**Jan Paulus**




Seminararbeit

Institut für deutsche Philologie
Lehrstuhl für Computerphilologie und Neuere Deutsche Literaturgeschichte

Dozent: Thorsten Vitt

Würzburg, den 14.02.2020 (TODO)

# Inhaltsverzeichnis

# Abbildungsverzeichnis



# Tabellenverzeichnis

## Zusammenfassung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

# 1 Einleitung

Bevor Burrows 2001 sein Delta-Maß vorstellte, war es im Bereich der Autorschaftsattribution-Forschung üblich, eine begrenzte Anzahl an voraussichtlich ähnlichen Texten miteinander zu vergleichen (Evert u. a., 2017, S. ii5). Diese Vorgehensweise wird auch „closed games“ genannt (Burrows, 2002, S. 267). Bei „open games“ hingegen gibt es hinsichtlich eines unbekannten Textes zuvor keinen Anhaltspunkt, welchen Kandidaten-Texten der unbekannte Text ähnlich ist. Das Ziel von Burrows Verfahren war es deshalb, diese „open games“ in „closed games“ zu transformieren (Burrows, 2002, S. 268). Burrows verwendete dafür das Delta-Maß, welches die Ähnlichkeit über die Distanz von einem unbekannten Text zu einer Gruppe von Texten berechnet (Burrows, 2002). Dieses Maß hat die folgenden Jahren der Autorschaftsattributions-Forschung geprägt (Evert u. a., 2017, ii5f.). Neben Weiterentwicklungen des Maßes wurden in dieser Zeit jedoch auch vermehrt Machine Learning Verfahren wie K-Nearest Neighbors, Nearest Shrunken Neighbors oder Support Vector Machines für die Autorschaftsattribution verwendet (Jockers und Witten, 2010, S. 217).

In dieser Arbeit soll untersucht werden, wie gut Burrows Delta im Vergleich zu Machine Learning Klassifizierungsverfahren das „open games“-Problem der Autorschaftsattribution löst. Um einen bessere Vergleichbarkeit zwischen den Verfahren zu gewährleisten, wird Burrows Delta dabei als Nearest Neighbor Klassifizierungsverfahren aufgefasst. Die zu vergleichenden Klassifizierungsverfahren sollen die Verfahren Multinomial Naive Bayes, Support Vector Machines, Logistic Regression und Random Forests (TODO: wirklich) sein. Für den Vergleich von Burrows Delta und den anderen Klassifizierungsverfahren werden verschiedene Versuchsszenarien geschaffen. Es sollen zwei verschiedene Korpora die Grundlage für die Experimente sein, ein Reden-Korpus und ein Prosa-Korpus. Diese Korpora werden in Kapitel 3 genauer erläutert. Nach Problemen mit dem Prosa-Korpus bei den ersten Versuchen wurde sich dazu entschieden, Varianten des Prosa-Korpus zu erstellen und diese ebenfalls als Grundlage für die Experimente zu verwenden. Dies wird in Kapitel 4.1 näher beschrieben. Eine weitere Unterscheidung der verschiedenen Versuchsszenarien soll die Auswahl der Features sein. Dabei wird zum einen getestet, wie sich die Auswahl der maximalen Anzahl der häufigsten Features auf die Autorschaftsattribution auswirkt. Es ist Konsens in der

Autorschaftsattributions-Forschung, dass die Beschränkung der Eingabefeatures auf die häufigsten Wörter die zuverlässigsten Ergebnisse liefert (Jockers und Witten, 2010, S. 215). Neben der Anzahl der häufigsten Wörter soll auch untersucht werden, welche N-Gramm-Reichweite für die verschiedenen Klassifizierungen die besten Ergebnisse liefert. Die Auswahl der Features wird in Kapitel 4.2 genauer erläutert. In den verschiedenen Versuchsszenarien soll neben verschiedenen Korpora und Feature-Selektionen auch unterschiedliche Feature-Repräsentationen getestet werden. Dabei soll untersucht werden, wie sich verschiedene Vektorisierungsmethoden wie das Bag-of-Words-Modell, die TF-IDF-Gewichtung, die Normalisierung mit dem Z-Score oder die Kosinus-Ähnlichkeit auf die Klassifizierungen auswirken. Die Z-Score-Normalisierung bietet die Grundlage für die Darstellung von Burrows Delta als Klassifizierungsverfahren. Es soll deshalb auch überprüft werden, ob die Z-Score-Normalisierung für Burrows Delta überhaupt die beste Feature-Repräsentation ist oder ob andere Vektorisierungsmethoden in Kombination mit einem Nearest Neighbor Klassifizierungsverfahren besser Ergebnisse liefern. Dies wird in Kapitel 4.3 näher beschrieben.

Für die Evaluierung der Klassifizierungen soll ein F1-Score und ein Cross-Validation-F1-Score verwendet werden.[1] Diese sollen jedoch nicht alleine ausschlaggebend für die Güte eines Klassifizierungsverfahren sein. Weitere Faktoren zur Bewertung der Güte eines Klassifizierungsverfahren sollen auch die Komplexität der Implementierung und die Dauer des Klassifizierungsvorgang sein. Letzteres wird jedoch nur sehr lose als aussagekräftiges Kriterium verwendet, da die genaue Messung der Dauer auch von sehr vielen Faktoren außerhalb der Versuchsumgebung wie beispielsweise die Hardware-Komponenten des Computers, auf dem die Experimente durchgeführt worden sind, abhängig ist. Genauere Beschreibungen des Versuchsaufbau und der Evaluierung der Ergebnisse befinden sich in Kapitel 5. In dieser Arbeit wird auch verstärkt auf Probleme und ihre Lösungsansätze eingegangen, die sich im Verlauf der Experimente hervorgetan haben. Eines dieser Probleme ist die zu gute Klassifizierung von beinahe allen untersuchten Machine Learning Verfahren hinsichtlich des ursprünglichen Prosa-Korpus, welches Optimierungen aber auch detaillierte Vergleiche zwischen den Verfahren obsolet machte. Um diesem entgegenzuwirken, wurde eine reduzierte Variante des Prosa-Korpus für folgende Experimente erstellt.

[1] Im Verlauf dieser Arbeit werden die englischen Begriffe verwendet.

# 2 Stand der Forschung

Eine der meist benutzten und ältesten Methoden der Stilometrie ist die Autorschaftsattribution. Die ersten Herangehensweisen der Autorschaftsattributions-Probleme wurden bereits im späten 19. Jahrhundert entwickelt und werden „einheitliche invariante Ansätze“ (engl.: „Unitary Invariant Approach“) genannt (Argamon, Koppel und Schler, 2009, S. 10). 1964 nutzten Mosteller und Wallace multivariaten Analyse-Ansätze (engl.: „Multivariate Analysis Approach“) bei der Untersuchung der Federalist Papers (Argamon, Koppel und Schler, 2009, S. 10). Der grundlegende Gedanke hinter diesen Ansätzen ist es, dass durch eine Darstellung aller zu untersuchenden Dokumente in einem mehrdimensionalen Raum der Autor eines unbekannten Dokuments durch ein Distanzmaß ermittelt werden kann: Der Autor des Dokuments, welches am nächsten zum unbekannten Dokument in diesem mehrdimensionalen Raum liegt, ist wahrscheinlich auch der Autor des unbekannten Dokuments (Argamon, Koppel und Schler, 2009, S. 11). Eine der bekanntesten Methoden der Autorschaftsattributions-Forschung, die auf diesem Gedanken aufbaut, ist das Delta-Maß von John Burrows, welches nach seiner Einführung 2001 maßgeblich die Autorschaftsattributions-Forschung der folgenden Jahre prägte (Evert u. a., 2017, ii5f). Burrows Delta ist ein sehr einfaches, aber effektives Maß. Die Häufigkeiten der Wörter innerhalb des Korpus werden ermittelt und mit dem z-score standardisiert. Basierend auf den standardisierten Häufigkeiten wird dann das namensgebende Delta-Scoring berechnet, indem der Mittelwert der absoluten Unterschiede zwischen den z-scores für eine Menge von Wort-Variablen im gegebenen Textkorpus und den z-scores für dieselbe Menge von Wort-Variablen im zu untersuchenden Text ermittelt wird. Die Formel dafür lautet: $\Delta_B = \sum_{i=1}^{m} |z_i(D_1) - z_i(D_2)|$.

Burrows Delta funktionierte in der Praxis sehr gut, jedoch waren die genauen Funktionsweisen bis zur Aufklärung durch Shlomo Argamon ungeklärt. Dieser lieferte ein besseres Verständnis der grundlegenden Annahmen des Delta-Maßes, eine Einschränkung der Methoden sowie theoretisch fundierte Variationen und Erweiterungen des Maßes (Argamon, 2008). Eine dieser Variationen ist die Auffassung von Burrows Delta als achsengewichtetes Nearest Neighbor Klassifizierungsverfahren (Argamon, 2008, S. 132–135). Dies erleichterte es, den Delta-Ansatz mit anderen Textklassifikationsverfahren aus dem Bereich des Machine Learning zu vergleichen. Die multi-class Textklassifizierung kann nämlich als Autorschaftsattributionsproblem

aufgefasst werden (Stamatatos, 2009, S. 539). Auf dieser Annahme beruhend wurden vergleichende Versuche durchgeführt, die das Delta-Maß mit anderen Klassifikationsverfahren verglichen. Das Delta-Maß schien in diesen Versuchen schlechter abzuschneiden als andere Klassifizierungsverfahren (Eder, 2015, S. 169). Koppel oder Stamatatos zum Beispiel stellten fest, dass die Support Vector Machines in Kombinationen mit N-Grammen das Autorschaftsattributionsproblem besser als das Delta-Maß lösen konnten (Koppel, Schler und Argamon, 2009). In den Versuchen von Jockers und Witten hingegen konnte (AW) Burrows Delta das Autorschaftsattributionsproblem ähnlich gut bewältigen wie andere Klassifizierungsverfahren, unter anderem auch dem Support Vector Machine Verfahren (Jockers und Witten, 2010). Jockers und Witten benutzten zwei Varianten des Nearest Neighbor Verfahren: das K-Nearest Neighbor und das Nearest Centroid Verfahren (Jockers und Witten, 2010; Jockers, Witten und Criddle, 2008).

Neben der Auffassung von Burrows Delta als Klassifizierungsverfahren wurde nach der Einführung das Maß auch selber weiterentwickelt. Eine Weiterentwicklung war der erweiterte Anwendungsbereich, etwa die Anwendung des Delta-Maßes auf weitere Textgattungen (Hoover, 2004) oder auf Korpora in anderen Sprachen (Eder und Rybicki, 2013). Eine andere Weiterentwicklung war die Verwendung von Burrows Delta mit der Kosinus-Ähnlichkeit (Smith und Aldridge, 2011). TODO: Random Forest erklären.

# 3 Überblick über die Korpora

Für die Experimente in dieser Arbeit sollen zwei Korpora benutzt werden, um einen aussagekräftigen Vergleich von Burrows Delta zu den Machine Learning Klassifizierungsverfahren zu gewährleisten. Das erste Korpus besteht aus Reden von deutschen Politikern aus dem 21. Jahrhundert (Barbaresi, 2018). Für die Nutzung in dieser Arbeit wurde es bearbeitet, indem die Reden tokenisiert und die Namen der Redner aus den Reden entfernt wurden, da dies eventuell die Klassifizierungen verfälschen könnte. Weiterhin wurde das Korpus auf 13 Redner mit jeweils 10 Reden gekürzt. Um eine Einheitlichkeit hinsichtlich der anderen Korpora zu gewährleisten, werden die Redner innerhalb dieser Arbeit als „Autoren“ bezeichnet und ihre Reden als „Texte“.

Das zweite Korpus stellt eine Variante des „Corpus of German-Language Fiction“ von Frank Fischer und Jannik Strötgen dar (Fischer und Strötgen, 2017).

Ihr Korpus ist eine extrahierte und konvertierte Version von Werken aus dem „Projekt Gutenberg-DE“. Das Korpus ist zweigeteilt: Der Großteil besteht aus deutschen Werken von deutschsprachigen Autoren, der kleinere Teil aus Werken von nicht-deutschsprachigen Autoren, die ins Deutsche übersetzt wurden. Der zweite Teil wird in dieser Arbeit ignoriert, da er einige Probleme aufweist wie teilweise nicht übersetzte Texte und fehlende Erscheinungsjahre. Der Teil des Korpus mit Werken von deutschsprachigen Autoren besteht aus 2735 Prosa Werken von 549 verschiedenen Autoren. Die Erscheinungsjahre erstrecken sich dabei von 1510 bis 1940, wobei der größte Teil der Werke zwischen 1840 und 1930 erschienen ist.[2] Die Einteilung der Werke in die Gattung Prosa ist sehr vage, da die Prosagattungen sehr mannigfaltig sind (Bücher-Wiki, 2019). Anhand des Korpus ist nicht erkennbar, zu welcher Prosagattung die einzelnen Werke gehören. Einige der Werke geben ihre Gattung zu Beginn des Textes an. Leider ist dies bei nur sehr wenigen Werken der Fall, eine einheitliche Angabe der Textgattung ist beim ursprünglichen Korpus nicht enthalten. Laut Angaben der Ersteller enthält das Korpus „mainly novels and short stories“ (Fischer und Strötgen, 2017). Das Korpus von Fischer und Strötgen wird für die Untersuchungen in dieser Arbeit vorverarbeitet und reduziert, die genaue Vorgehensweise befindet sich in Appendix B.

# 4 Konzeption der Untersuchungen

## 4.1 Varianten der Korpora

Zu dem originalen Prosa-Korpus[3] sollte eine Variante erstellt werden, mit denen weitere Experimente durchgeführt werden sollten. Diese Überlegung kam nach den ersten Experimenten auf, nachdem sich herausstellte, dass die Klassifizierung des Prosa-Korpus zu gut funktionierte und viele Klassifizierungsverfahren Genauigkeiten über 95% erreichten, unabhängig der Vektorisierungsmethode.[4] Durch eine Variation des Prosa-Korpus sollte den Klassifizierungsverfahren die Klassifizierung erschwert werden, um damit einen Spielraum für Verbesserungen und stärkere Unterschiede zwischen den verschiedenen Experimenten zu schaffen. Die Idee der Erstellung einer Variante des Reden-Korpus wurde aus mehreren Gründen verworfen. Zum einen

[2] Dies wird auch durch Abbildung TODO im Appendix B bestätigt.
[3] Im Verlauf dieser Arbeit wird das Korpus so bezeichnet, um es von der Korpus-Variation abzugrenzen.
[4] Siehe Tabellen 2-5 TODO in Appendix C.1.

lieferten die ersten Experimente des Reden-Korpus nicht so gute Genauigkeiten wie das originale Prosa-Korpus, sodass eine Reduzierung oder eine Aufteilung des Reden-Korpus, um den Verfahren die Klassifikation zu erschweren, überflüssig gewesen wäre. Weiterhin wurde die Idee der Segmentierung des Reden-Korpus verworfen, da Tests zeigten, dass die Anzahl an Reden nur signifikant stieg, wenn die Segmente sehr wenige Wörter enthielten. Die Anzahl der Wörter war wiederum zu gering, um eine geeignete Feature-Selektion durchzuführen. Zuletzt zeigten Untersuchungen, dass das Korpus weder sehr seltene Wörter noch Wörter, die nur von einem Autor benutzt wurden, beinhaltete.

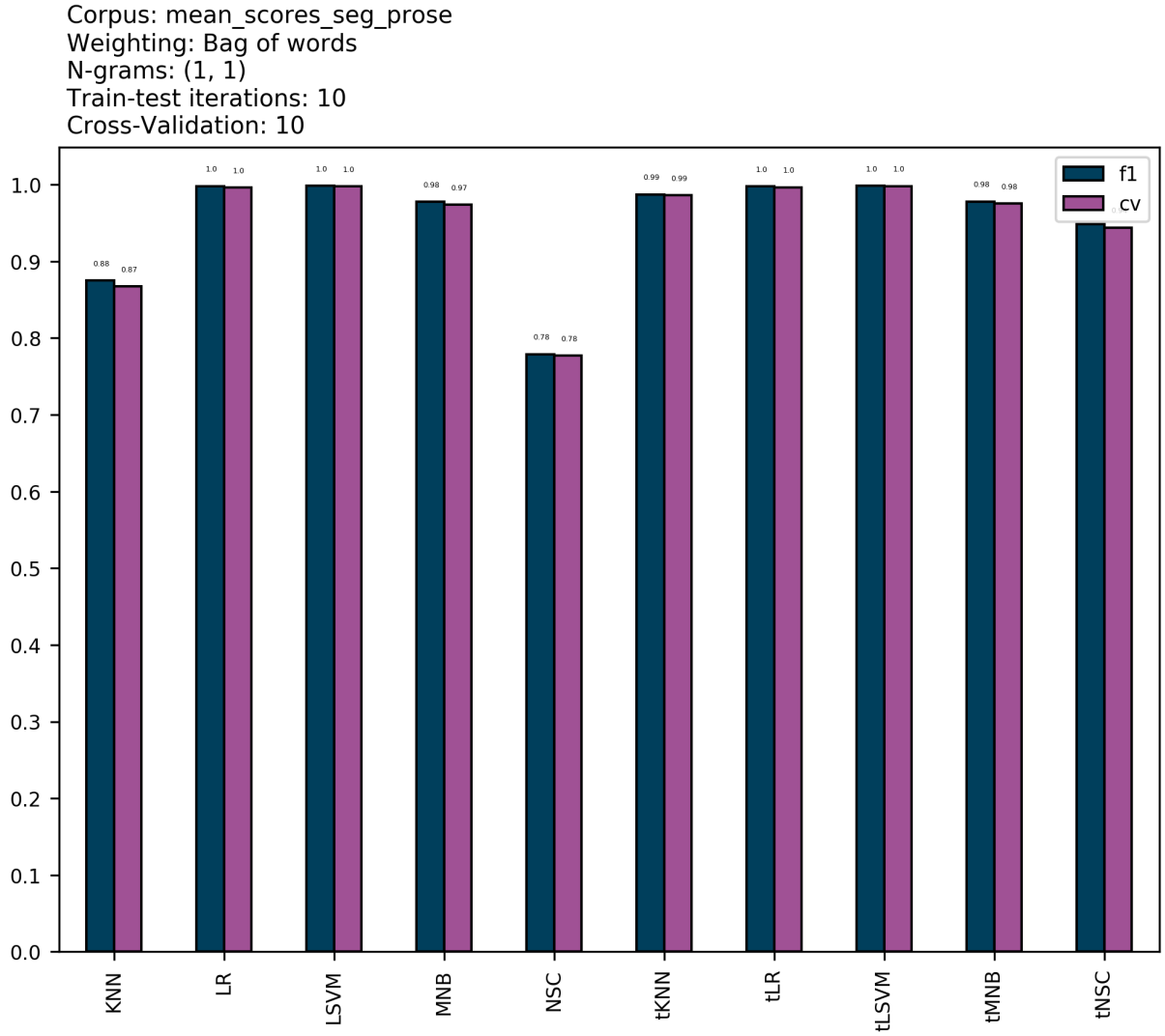
Der erste Ansatz, eine Variation des originalen Prosa-Korpus zu erstellen, war die Segmentierung der Texte. Das Prosa-Korpus wurde zwar vor den Experimenten vorverarbeitet, indem die Anzahl der Texte und Klassen verändert wurden, eine wirkliche Feature-Selektion fand jedoch nicht statt. Offensichtlich gab es beim originalen Prosa-Korpus noch zu viele Features, die eine zu gute Trennung der Klassen ermöglichten. Ähnlich wie die Vorverarbeitung bei Kocher und Savoy sollten nun Satzzeichen, seltene Wörter und Wörter, die nur von einem Autor benutzten wurden, entfernt werden (Kocher und Savoy, 2018, S. 428). Anders als bei Kocher und Savoy sollten jedoch alle Satzzeichen entfernt werden. Aufgrund der Einschränkung, nur die häufigsten $n$ Wörter als Features zu verwenden, wurden weder sehr seltene Wörter noch Wörter, die nur von einem Autor benutzt wurden, verwendet.

Eine weitere Änderung war die Segmentierung der Texte der einzelnen Autoren, wie sie auch beim Zeta-Maß verwendet wird (Schöch, 2018, S. 82). Da das Prosa-Korpus sehr lange Texte wie Romane beinhaltet, jedoch nur sehr wenige Texte einiger Autoren, konnte mit einer Segmentierung neben einer Reduktion sehr langer Texte auch eine Erhöhung des Cross-Validation-Wertes erreicht werden. Dieser wird bei jedem Experiment dynamisch an den jeweiligen Datensatz und die Anzahl der Texte bezüglich einer Klasse angepasst. Beim Prosa-Korpus wurden zuvor schon Autoren mit weniger als drei Texten entfernt, jedoch wäre das Korpus zu klein geworden, wäre die untere Grenze für die Anzahl von Texten pro Autor höher angesetzt worden. Dementsprechend wurde der Cross-Validation-Wert auch nur auf 2 gesetzt.[5] Als Trennwert für die Segmentierungen wurde 10000 verwendet. Segmente, die weniger als 10000 Tokens beinhalteten, wie etwa das Ende eines Romans, wurden nicht verwendet. Da dadurch das Korpus etwa 10 Mal so groß wurde, wurden Autoren mit 20 oder

[5] Siehe Abbildung TODO (ICH: abb. von experiment 1-4.

weniger Werken aus dem Korpus entfernt, da die Experimente ansonsten zu lange gedauert hätten. Die Anzahl der verschiedenen Autoren reduzierte sich von 87 auf 23. Die Anzahl der Texte erhöhte sich von 488 auf 696.

Nach einem ersten Experiment, bei der als Vektorisierungsmethode das Bag-of-Words-Modell und Monogramme verwendet wurden, zeigte sich jedoch, dass durch die Segmentierung die Klassifizierungsverfahren noch bessere Genauigkeiten erzielen konnten als beim originalen Prosa-Korpus. [6] Deshalb wurde eine weitere Variante des originalen Prosa-Korpus erstellt. Die segmentierte Variante wurde für keine weiteren Experimente mehr verwendet.




[6] Siehe Abbildung 1 TODO

**Abbildung 1:** Der linke Teil des Balkendiagramms stellt die Klassifizierungsergebnisse der nicht optimierten Verfahren dar und der rechte Teil die Ergebnisse optimierten Verfahren. Selbst die nicht optimierten Verfahren erreichen teilweise Genauigkeiten von 100%.

Basierend auf der Idee der Segmentierungs-Variante des originalen Prosa-Korpus ...
TODO
IDEEN:

- TODO: erwähnen, dass die Variation hier gegenteiligen Effekt erreicht hatte: viel zu gute Klassifizierung. weitere Variante erstellen ohne Segmentierung, bei der nur die ersten 10000 Zeichen verwendet werden; Rest des Romans weg.
- noch kleineren Jahreszeitraum? 1840-1900?
- BEGRÜNDUNG zu gute Klassifizierung:
- zu viele Daten?!
- zu großes Jahresspektrum, d.h. Daten ließen sich zu gut unterscheiden?
- anzahl von mindestanzahl für autoren von 3 auf 4 erhöht (nur noch 196 texte)
- TODO: Balkendiagramm für durchschnittliche Genauigkeit aller 6 max features zeigen
- TODO: nach Erklärung des einen Versuchs: noch weniger verschiedene Autoren und kleinerer zeitlicher Rahmen!

## 4.2 Häufigste Wörter und N-Gramme als Features

Für die Autorschaftsattribution ist es üblich, als Eingabefeatures die $n$ häufigsten Wörter zu verwenden (Jockers und Witten, 2010, S. 218). Die Größe von $n$ variiert in der Literatur je nach Experiment. Jockers und Witten verwendeten für die Autorschaftsattribution der Federalist Papers zwischen 298 und 2907 der häufigsten Wörter (Jockers und Witten, 2010, S. 218), Eder benutzte die 200 häufigsten Wörter für seine Experimente (Eder, 2015, S. 169) und Burrows und Grieve nutzten für die Tests bezüglich des Delta-Maßes und der chi-square Methode zwischen 40 und 1000 der häufigsten Wörter (Kocher und Savoy, 2018, S. 427). Ein wirklichen Konsens, welche

Anzahl von Features für die Autorschaftsattribution besonders gut funktioniert, gibt es nicht (Eder, 2017, S. 52). Da auch für die Experimente in dieser Arbeit die häufigsten Wörter als Eingabefeatures verwendet werden sollten, wurden mehrere Experimente mit variierenden Größen für $n$ durchgeführt. Bei den Grenzen von $n$ wurde sich dabei an den Grenzen von Jockers und Witten orientiert, da ihre Experimente denen in dieser Arbeit ähneln. Die zu untersuchten Werte für $n$ waren 200, 300, 500, 1000, 2000 und 3000.

- todo: ngrams

## 4.3 Vektorisierungsmethoden (TODO: oder Feature-Repräsentation?

Neben der Wahl der Feature-Selektion sollte auch mit verschiedenen Vektorisierungsmethoden experimentiert werden. Burrows Delta-Maß ist eigentlich ein Distanzmaß und kein Klassifizierungsverfahren, kann aber in Verbindung mit dem Nearest Neighbor Verfahren als Klassifikationsverfahren aufgefasst werden (Argamon, 2008, S. 132). Damit das Nearest Neighbor Verfahren jedoch das Delta-Maß repräsentiert, müssen die Häufigkeiten der Wörter mit dem z-score normalisiert werden. Für die Experimente in dieser Arbeit sollten die durch die z-scores normalisierten Häufigkeiten als eine mögliche Vektorisierungsmethode verwendet werden. Andere Vektorisierungsmethoden waren das Bag-of-Words-Modell, die Gewichtung mit dem TF-IDF-Maß und die Kosinus Ähnlichkeit. Letztere ist kein Distanzmaß im eigentlichen Sinne, kann jedoch in Kombination mit der euklidischen Distanz als Distanzmaß aufgefasst und als Vektorisierungsmethode verwendet werden.[7] In Kombination mit der Kosinus Distanz wird durch das Nearest Neighbor Verfahren die Variante von Burrows Delta von Smith und Aldridge implementiert (Smith und Aldridge, 2011). Ebenfalls ist anzumerken, dass das Nearest Centroid Klassifizierungsverfahren in Kombination mit dem TF-IDF-Maß auch als Rocchio Klassifizierungsverfahren bekannt ist.[8] Die in dieser Arbeit verwendeten Nearest Neighbor Klassifizierungsverfahren stellen also abhängig von ihrer Vektorisierungsmethode andere Verfahren dar. Es wurde deshalb auch untersucht, ob die Vektorisierungsmethode mit der z-score-Normalisierung oder mit

[7] Siehe die Erklärung eines Stack Overflow Nutzers `https://stackoverflow.com/questions/34144632/using-cosine-distance-with-scikit-learn-kneighborsclassifier` (abgerufen am 13.01.2020).

[8] Siehe `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid` (abgerufen am 13.01.2020).

dem Kosinus-Distanzmaß, mit der zwei Versionen von Burrows Delta implementiert werden, die besten Vektorisierungsmethoden für diese Klassifizierungsverfahren sind.

### 4.4 IDEEN 2

TODO

- baseline für Klassifizierung? 80% sagt JUOLA, The Rowling Case, S. i105 –> ICH: eher auf 90% setzen, da vor allem Prosa-Korpus das erreicht. Für Reden-Korpus muss noch getestet werden!

## 5 Aufbau der Experimente

Die Grundidee beim Entwurf der Experimente war es, dass alle Experimente ähnlich aufgebaut sein sollten, um sie gut miteinander vergleichen zu können. Weiterhin sollten Messungen der Klassifizierungsgenauigkeit öfters durchgeführt werden, um möglichst allgemeingültige Aussagen über die Genauigkeit der Klassifizierungsverfahren bezüglich der getesteten Korpora treffen zu können. Das zu untersuchende Korpus wurde bei jedem Experiment in einen Trainings- und Testdatensatz unterteilt. Die Trennung erfolgte nach dem Paretoprinzip, d.h. 80 Prozent des Korpus waren Teil des Trainingsdatensatzes und 20 Prozent Teil des Testdatensatzes. Für die Aufteilung wurde eine Stratifizierung bezüglich der Autoren benutzt. Damit wurde gewährleistet, dass der Trainings- und Testdatensatz ungefähr den gleichen Anteil an Texten vom gleichen Autor beinhalteten. Die Genauigkeit der Klassifizierung wurde mithilfe des F1-score ermittelt. Da es sich bei den Experimenten um eine Multiclass-Klassifizierung handelte, musste ein average-Parameter für den F1-score gesetzt werden. Dieser wurde auf „micro“ gesetzt, diese Wahl war willkürlich. Da die Klassen der Korpora verschieden groß sind, konnte kein allgemeiner Cross-Validation-Parameter gesetzt werden, da dieser von der Anzahl von Texten, die zu einer Klasse gehören, abhängig ist. Ein dynamischer Cross-Validation-Parameter wurde deshalb zu Beginn jedes Experiments ermittelt. Damit wurde gewährleistet, dass ein größtmöglicher Cross-Validation-Parameter genutzt werden konnte, welcher wiederum eine genauere Messung ermöglichte. Dieser Cross-Validation-Parameter hatte als obere Grenze den Wert 10. Um die Messungen der Genauigkeiten mit dem F1-score und der Cross-Validation noch zu verbessern, wurden zehn Trainings- und Test-Iterationen

ausgeführt, wobei der Trainingsdatensatz zuvor gemischt wurde. Eine Iteration bestand dabei aus der Berechnung des F1-scores und der Cross-Validation für jedes optimierte und nicht-optimierte Klassifikationsverfahren. Nach der Messung wurde von allen Werten jeder Iteration der Mittelwert gebildet. Diese finalen F1-scores und Cross-Validation Werte bildeten die Metriken für den Vergleich der Klassifizierungsverfahren.
Da auch untersucht werden sollte, wie lange die Klassifizierungsverfahren für die Klassifizierung brauchten, wurde ebenfalls die Dauer der Klassifizierungen gemessen. Da genaue Messungen der Ausführungen nicht geliefert werden konnten, da es zu viele Störfaktoren wie die aktuelle Auslastung der CPU oder eine unterschiedliche ausführliche Auswahl der Hyperparameter für die Hyperparameteroptimierung gibt, sollte die Dauer eher als ein grober Richtwert denn als ein aussagekräftiges Vergleichskriterium angesehen werden. Sie wurde in dieser Arbeit nur in Verbindung mit der Genauigkeit des Klassifizierungsverfahren interpretiert. Zudem wurde die Dauer als prozentualer Anteil von der Gesamtdauer einer Experimenten-Iteration aufgefasst. TODO: noch mehr?
Für den Vergleich von Burrows Delta mit den Machine Learning Klassifizierungsverfahren wurde es als Nearest Neighbor Klassifizierungsverfahren aufgefasst (Argamon, 2008, S. 132). Durch einen gleichen Aufbau konnte eine möglichst korrekte Evaluation von Burrows Delta und den einzelnen Verfahren gewährleistet werden. Alle hier vorgestellten Verfahren wurden mithilfe der Machine Learning Bibliothek Scikit-learn implementiert. Burrows Delta sollte mithilfe von drei verschiedenen Nearest Neighbors Klassifizierungsverfahren dargestellt werden: k-Nearest Neighbors (KNN) [9], Radius Neighbors (RN) und Nearest Shrunken Centroids (NSC). Diese sollten mit den Klassifizierungsverfahren Multinomial Naive Bayes (MNB), Logistic Regression (LR), linearen Support Vector Machines (LSVM), nicht-linearen Support Vector Machines (SVM) und Random Forests (RF) verglichen werden. Die Standard-Implementierung der Decision Trees von Scikit-learn wurden bei den Experimenten nicht berücksichtigt, da sie ein zeitaufwendiges Training benötigte und in ähnlichen Experimenten schlechtere Werte als andere, simplere Klassifizierungsverfahren lieferte (Zhao und Zobel, 2005). In Scikit-learn gibt es noch weitere, auf Entscheidungsregeln basierende Klassifizierungsverfahren, die

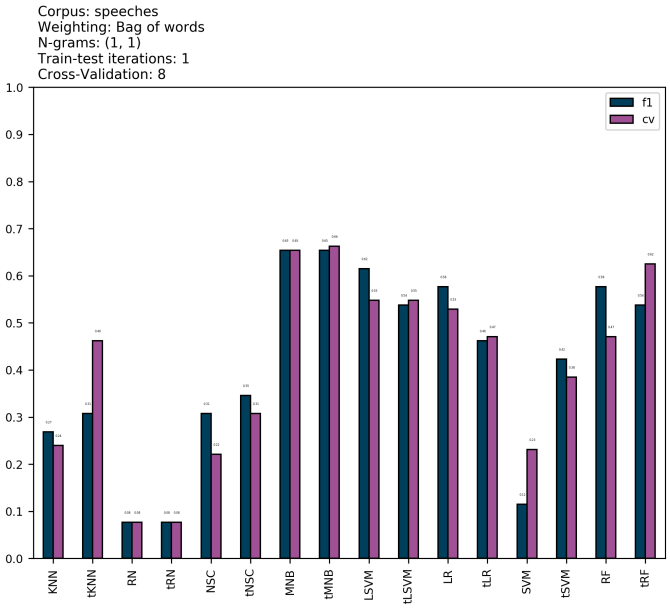
[9] Im weiteren Verlauf dieser Arbeit werden die Abkürzungen der Klassifizierungsverfahren innerhalb der Klammern benutzt. Diese Abkürzungen befinden sich auch in den Abbildungen.

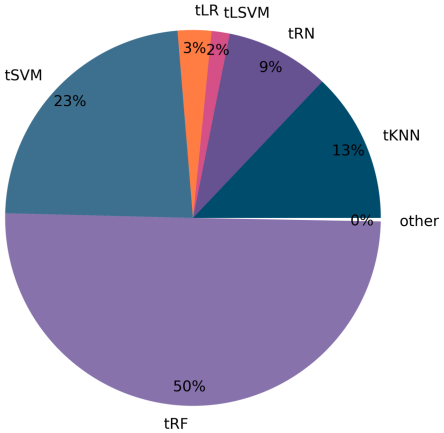
sich anhand ihrer Ensemble Learning Technik unterscheiden lassen. Die auf der Ensemble Learning Technik Boosting beruhenden Verfahren wie die Gradient Boosting Machines sowie die Varianten XGBoost, Catboost und LightGBM wurden für die Experimente nicht benutzt, da sie noch zeitaufwendiger und komplexer als die Decision Trees sind. Lediglich die auf der Ensemble Learning Technik Bagging basierenden Random Forest Verfahren (RF) sollten innerhalb der Experimente verwendet werden. Jedes der verwendeten Klassifizierungsverfahren sollte einmal mit seinen Standardparametern[10] und einmal mit optimierten Parametern verwendet werden. Diese Hyperparameteroptimierung wurde ebenfalls während eines Experiments durchgeführt. In den Abbildungen und weiteren Verlauf dieser Arbeit werden die optimierten Klassifizierungsverfahren mit einem kleingeschriebenen „t“ gekennzeichnet. Diese Hyperparameteroptimierung ist in keinem Fall vollständig und erhebt auch keinen Anspruch auf Vollständigkeit.
Alle Wörter der Korpora wurden für die Experimente in Kleinbuchstaben dargestellt. Dies ist für deutschsprachige Korpora nicht immer sinnvoll. Da hier jedoch nur die häufigsten Wörter und N-Gramme als Eingabedaten benutzt wurden, reduzierte die Darstellung aller Wörter in Kleinbuchstaben die Dopplung von Wörtern. Die Trainings- und Testdatensätze der Korpora wurden mit verschiedenen Methoden vektorisiert und standardisiert. Diese Methoden wurden in Kapitel 4.TODO bereits vorgestellt und sind die einfache Termhäufigkeit, repräsentiert durch das Bag-of-Words-Modell und die Gewichtung der Termhäufigkeiten mit dem TF-IDF Maß, der Kosinus-Ähnlichkeit und dem z-score. Dabei war das originale Delta-Maß von Burrows in Wirklichkeit nur gegeben, wenn die z-scores als Vektorisierungsmaß verwendet wurden. Wurde jedoch die Kosinus-Distanz verwendet, wurde die von Smith und Aldridge Variante des Delta-Maßes verwendet (Smith und Aldridge, 2011). Beide wurden in den Abbildungen mit einem „D“ vor den jeweiligen Nearest Neighbor Verfahren gekennzeichnet. Dies diente der Abgrenzung zu den Nearest Neighbor Verfahren mit anderen Vektorisierungsmethoden.
Bei einem vorangestellten Experiment wurde anhand des Reden- und des originalen Prosa-Korpus getestet, wie lange die einzelnen Klassifizierungsverfahren für einen einfachen Durchlauf prozentual benötigen würden und wie die Genauigkeit der Klassifizierung nach diesem Durchlauf für jedes Klassifizierungsverfahren etwa sein

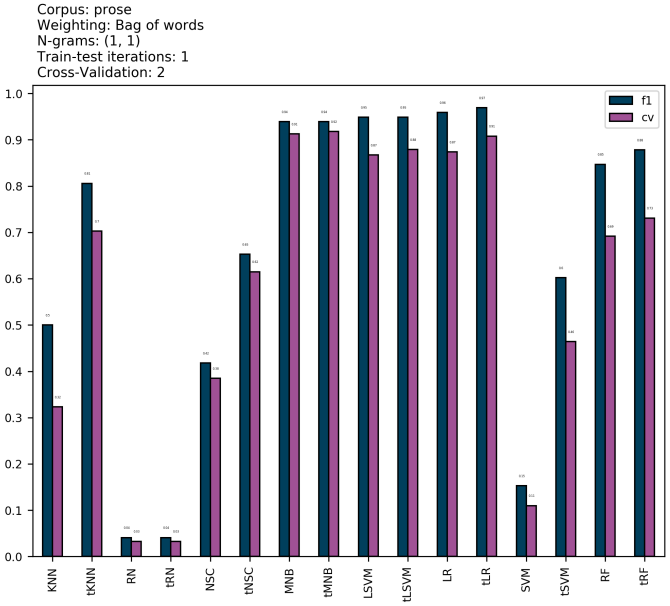
[10] Bei dem Logistic Regression Verfahren wurden die Parameter „multi_class“ und „solver“ bereits vorher angegeben.

würde [11]. Damit sollten sehr zeitaufwendige Verfahren und Verfahren mit einer schlechten Genauigkeit für die Experimente entfernt werden. Es stellte sich heraus, dass nicht-lineare Support Vector Machines schlechter als die linearen Support Vector Machines abschnitten. Sie wurden deshalb für weitere Experimente entfernt. Radius Neighbors lieferte ziemlich schlechte Werte, auch dieses Verfahren wurde für die Experimente in dieser Arbeit nicht mehr berücksichtigt. Auch das Random Forest Klassifizierungsverfahren wurde für die ersten Experimente ignoriert, da er vor allem beim Reden-Korpus vergleichsweise lange für die Klassifizierungen benötigte. Über die Genauigkeiten konnte jedoch mit diesem vorangestellten Experiment noch kein Urteil gebildet werden und andere Experimente ergaben gute Ergebnisse (Tabata, 2012). Deshalb sollte das Random Forest Klassifizierungsverfahren in späteren Experimenten noch einmal betrachtet werden (TODO: habe ich das gemacht?).

[11] Siehe Abbildung TODO


Corpus: speeches
Weighting: Bag of words
N-grams: (1, 1)
Train-test iterations: 1
Cross-Validation: 8
f1
cv
KNN
tKNN
RN
tRN
NSC
tNSC
MNB
tMNB
LSVM
tLSVM
LR
tLR
SVM
tSVM
RF
tRF


Duration of speeches corpus
Corpus: speeches
Weighting: Bag of words
N-grams: (1, 1)
Train-test iterations: 1
Cross-Validation: 8
tLR
tLSVM
tRN
3%
2%
9%
tSVM
23%
tKNN
13%
0%
other
50%
tRF


Corpus: prose
Weighting: Bag of words
N-grams: (1, 1)
Train-test iterations: 1
Cross-Validation: 2
f1
cv
KNN
tKNN
RN
tRN
NSC
tNSC
MNB
tMNB
LSVM
tLSVM
LR
tLR
SVM
tSVM
RF
tRF

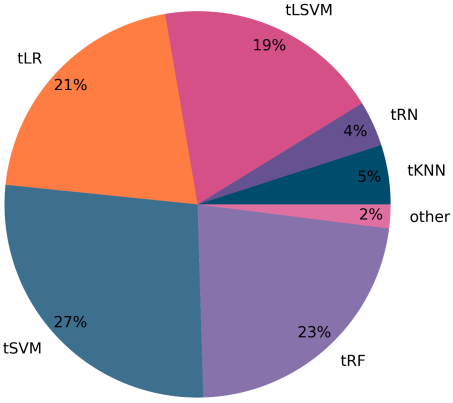

Duration of prose corpus
Corpus: prose
Weighting: Bag of words
N-grams: (1, 1)
Train-test iterations: 1
Cross-Validation: 2
tLSVM
19%
tLR
21%
tRN
4%
5%
tKNN
2%
other
27%
23%
tSVM
tRF

**Abbildung 2:** Für die Ermittlung der Genauigkeit und Dauer der Klassifizierung aller Klassifizierungsverfahren wurden das Bag-of-Words-Modell, Monogramme (N-Gramm-Reichweite = (1,1)) und eine einmalige Aufteilung der Korpora in Trainings- und Testdatensätze verwendet. Die Balkendiagramme stellen die Genauigkeit der Klassifizierungsverfahren mithilfe des F1-scores und der Cross-Validation dar und das Kreisdiagramm die prozentuale Dauer der Klassifizierung, gemessen an der Gesamtdauer. Die obere Reihe stellt die Messungen des Reden-Korpus dar, die untere Reihe die Messungen des originalen Prosa-Korpus. Bei beiden Versuchen schnitt das Radius-Neighbors Verfahren sehr schlecht ab, weshalb es für die weiteren Experimente in dieser Arbeit ignoriert wurde. Auch die nicht-linearen SVMs waren im Vergleich zur LSVM schlechter, zudem dauerte die Hyperparameteroptimierung aufgrund der verschiedenen getesteten Kernel-Arten natürlich länger. Da aber dadurch in beiden Fällen die LSVM nicht übertroffen werden konnte, wurden die nicht-linearen SVMs ebenfalls für weitere Experimente ignoriert. Die RF-Verfahren lieferten keine schlechten Genauigkeiten, die Hyperparameteroptimierungen dauerten jedoch sehr lange, obwohl die Anzahl der zu optimierenden Hyperparameter für diesen Versuch reduziert wurden. Die RF-Verfahren wurden deshalb für die ersten Experimente ignoriert, sollten aber in Kapitel ??? erneut aufgegriffen werden (TODO: welches kapitel? mach ich das überhaupt?).

TODO allg

- JOCKERs text angucken, der beschreibt einige wichtige Maßnahmen, die stylom-Klassifizierung von normaler Dokument-Klassifizierung unterscheidet.
- HOOVER, s. 470 (und bei burrows delta irgendwo): sagt, dass Erhöhung der Features die Accuracy erhöhen: das beweisen!? (notiz auch bei nächstem kapitel)
  - hier aufbau erklären:
  - welche verfahren?
  - was waren die ideen?
  - was waren probleme?
  - „Problematik bei Klassifizierung: viele Klassen, wenig Beispiele. Dies macht die Klassifizierung schwierig, auch im Hinblick der Evaluation, da cv der Cross Validation nicht größer als 2 sein darf (s.o.)“
- Wurde erstmal ausgeklammert: Deshalb sollte auch untersucht werden (AF?), ob die z-scores als Maß bessere Werte im Zusammenhang mit den Nearest Neighbor Verfahren liefert als das Bag-of-Words-Modell, das TF-IDF Maß und die Kosinus-Ähnlichkeit. Die Kosinus-Ähnlichkeit wurde bereits erfolgreich in Verbindung mit Burrows Delta getestet (Smith und Aldridge, 2011). Auch

dies soll mit den Experimenten überprüft werden. Weitere zu untersuchende Parameter neben der Wahl der Vektorisierungsmaße sollen die Anzahl der häufigsten Wörter (engl.: „most frequent words“), die ausschließlich für eine Klassifizierung verwendet werden, und die Auswahl der N-Gramme sein. Diese zu untersuchenden Parameter sind typische Untersuchungsobjekte in der Stilometrie-Forschung (Jockers und Witten, 2010).

# 6 Experimente TODO

AUFBAU

- E1-4: Ausprobier Experimente (Prosa-Korpus nur noch in reduzierter Variante) TODO ICH: erzählen, dass auch eine Arte Ausprobier-Experiment war: wie sind Korpora z.b.! auch nur hier so ausführlich die tabellen
- E5-8 und 9-12: N-Gramme (2,2) und (3,3) (nur speeches und reduced prose)
- beste Ansätze nehmen und Random Forests benutzten und RICHTIGE Kosinus-Ähnlichkeit!

## 6.1 Monogramme

In den Experimenten 1 bis 4 wurden das Reden-Korpus, das originale Prosa-Korpus und das reduzierte Prosa-Korpus mit verschiedenen Vektorisierungsmethoden und Anzahl von Features getestet. Die N-Gramm-Reichweite war (1,1), d.h. es wurden nur Monogramme verwendet. In diesen ersten Experimenten sollte zunächst überprüft werden, welche Anzahl an Features für die jeweiligen Korpora in Kombination mit den Vektorisierungsmethoden Bag-of-Words, Z-Score, TF-IDF und Kosinus Ähnlichkeit am besten funktionierte. Die verschiedenen Parameter für die Featureanzahl waren 200, 300, 500, 1000, 2000 und 3000. Die Klassifizierungsergebnisse wurden in Tabellen gespeichert.[12] Die hier vorliegende Tabelle hebt die besten Cross-Validation-F1-Scores des besten Nearest Neighbor Verfahrens und des besten anderen Klassifizierungsverfahren für jedes Korpus hervor.

[12] Siehe Appendix C.1

Die Ergebnisse zeigen, dass die Nearest-Neighbor Verfahren bei jedem getesteten Korpus schlechter abschneiden als die anderen Klassifizierungsverfahren.[13] Dabei schwanken die Cross-Validation-F1-Scores je nach Korpus. Anders als angedacht erzielt das reduzierte Prosa-Korpus teils bessere Werte als das originale Prosa-Korpus. Am schlechtesten wurde das Reden-Korpus klassifiziert, dass im Vergleich zu den anderen beiden Korpora jedoch auch weniger Datensätze beinhaltet. Auffällig ist, dass die besten Werte mit einer großen Anzahl an maximalen Features (1000-3000) erreicht wurde. Dies steht zum einem im Kontrast zu anderen Autorschaftsattributions-Versuchen, die mit weniger Features bessere Ergebnisse erreichen konnten (Eder, 2015; Jockers und Witten, 2010; Kocher und Savoy, 2018), zum anderen unterstützt dies die These von Eder, dass es keinen wirklichen allgemeinen Konsens darüber gibt, welche maximale Feature-Anzahl besonders gut für die Autorschafts-Attribution funktioniert (Eder, 2017, S. 52). Eine weitere Auffälligkeit ist, dass das Delta-Verfahren in der Kombination mit einer Z-Score Matrix ähnlich gute Werte wie mit einer TF-IDF gewichteten Matrix erreichte. Welche Vektorisierungsmethode bessere Ergebnisse liefert, hängt hier vom jeweiligen Korpus ab. Insgesamt schnitt Bag-of-Words im Vergleich zum Z-Score und der TF-IDF-Gewichtung schlechter ab und die Kosinus-Ähnlichkeit lieferte konstant die schlechtesten Werte. Noch eine weitere Auffälligkeit ist, dass Logistic Regression in keiner Zelle vorkommt. Stattdessen dominiert die Lineare SVM, welche für jedes Korpus die besten Werte liefert. Multinomial Naive Bayes hingegen ist für bestimmte Vektorisierungsmethoden das beste Klassifizierungsverfahren.
In den nächsten Experimenten sollen nun verschiedene N-Gramm-Reichweiten ausprobiert werden, um zu sehen, ob dies eine Verbesserung der Klassifizierungsergebnisse mit sich bringt und ob eventuell dadurch andere Klassifizierungsverfahren die Lineare SVM übertreffen können.

TODO INTERPETATION
Erkenntnis original Prosa

- Kosinus insgesamt sehr schlecht, wird für weitere Experimente nicht mehr verwendet (wenn auch schlecht bei Speeches-Korpus)
- mit Ausnahme von TF-IDF KNN nicht bestes Verfahren: bei bow und z-score um

[13] Siehe Tabelle TODO

einiges schlechter als LR und LSVM

- TF-IDF Überraschung: KNN bestes Verfahren, leicht schlechter dahinter LSVM und MNB. LR hier sehr schlecht. NSC ist hier fast gleich gut wie KNN, wird immer besser, je mehr Features

- Insgesamt sehr hohe F1-scores, die im Fall von Z-score fast perfekt sind (LSVM bei 300 und 2000 Features 99%, sowohl bei F1-score als auch bei der Cross-Validation). Deshalb auch die Variante mit der Segmentierung, um es den Klassifizierungsverfahren schwieriger zu machen. CV-Wert hier auch nicht so aussagekräfitg deswegen, da nur auf CV = 2.

- Daten konnten zu gut auseinander gehalten werden = Stil der Autoren wahrscheinlich zu unterschiedlich. Reden-Korpus da interessanter!

| vectorizer | **prose** | | **speeches** | | **prose_reduced** | |
|---|---|---|---|---|---|---|
| | NN | others | NN | others | NN | others |
| **bow** | 0.738 | 0.902 | 0.454 | 0.69 | 0.867 | 0.899 |
| clf | tKNN | tLSVM | tKNN | tMNB | tKNN | tMNB |
| mf | 3000 | 3000 | 2000 | 2000 | 2000 | 2000 |
| **zscore** | 0.859 | **0.985** | 0.497 | **0.725** | **0.918** | **0.979** |
| clf | D-tKNN | **tLSVM** | D-tNSC | **tLVSM** | **D-tKNN** | **tLSVM** |
| mf | 3000 | **3000** | 3000 | **3000** | **2000** | **2000** |
| **tfidf** | **0.919** | 0.864 | **0.591** | 0.667 | 0.874 | 0.882 |
| clf | **tKNN** | tMNB | **tNSC** | tMNB | tKNN | tMNB |
| mf | **3000** | 3000 | **2000** | 2000 | 3000 | 3000 |
| **cos** | 0.34 | 0.363 | 0.432 | 0.644 | 0.351 | 0.6 |
| clf | tKNN | tMNB | tKNN | tLSVM | D-tKNN | tLSVM |
| mf | 1000 | 1000 | 3000 | 3000 | 2000 | 2000 |

**Tabelle 1:** Die Tabelle stellt die besten optimierten Nearest Neighbor Klassifizierungsverfahren und die besten anderen optimierten Klassifizierungsverfahren für jedes Korpus sowie für jede Vektorisierungsmethode dar. Die hervorgehobenen Zellen sind die besten Verfahren im Hinblick auf alle verwendete Vektorisierungsmethoden für jedes Korpus (AF?). Jede Zelle beinhaltet in der ersten Zeile den besten Cross-Validation-F1-Score, in der zweiten Zeile das beste Klassifizierungsverfahren der Nearest Neigbor oder anderen Klassifizierungsverfahren und in der letzten Zeile die beste getestete maximale Feature-Anzahl. Dabei wurden sich an der besten getesteten maximalen Feature-Anzahl des Nearest Neighbor Verfahren orientiert, die beste getestete maximale Feature-Anzahl der anderen Klassifizierungsverfahren ist deshalb immer identisch mit der Feature-Anzahl der Nearest Neighbor Klassifizierungsverfahren. Der beste Cross-Validation-F1-Score ist hier nicht unbedingt der höchste Cross-Validation-F1-Score, sondern auch ein Score, dessen Unterschied zum gemessenen F1-Score möglichst minimal sein sollte. Bei geringen Unterschieden zwischen den höchsten Cross-Validation-F1-Scores wurde als der Score mit einem geringen Unterschied zum F1-Score verwendet.

## 6.2 6.2. Bigramme und Trigramme

TODO

## 6.3 6.3. Noch mehr Features TODO

TODO. hier vllt auch random forests mit einbringen. bow und cos reauskicken. auch eventuell die andere richtige cosinus-delta verwenden!!!

TODO

- HOOVER, s. 470 (und bei burrows delta irgendwo): sagt, dass Erhöhung der Features die Accuracy erhöhen: das beweisen!?
- EDER, 2015 (does size...), S. 169 rechts unten: SVM + ngrams angeblich bester AA-Ansatz (ICH: hier Literatur der beiden genannten angucken)
- ICH: siehe hier Notizen von hackmd für vers. Experimente
- ICH: viele Grafiken

UMFANG (TODO weg):

- min 9 Seiten
- einleitende worte mit theorie

# 7 Analyse der Experimente

TODO

- hier BURROWS/HOOVERS (S. 470) These, dass mehr features gleich bessere accuracy bei delta, analysieren
- ERKENNTNIS 1: umso größer Feature-Anzahl, umso länger die Dauer der Klassifizierungsverfahren und umso größer Dauer von SVM und LR –> sehr trivial, da ja auch mehr Daten verarbeitet werden müssen. Dauer von KNN bleibt interessanterweise aber ziemlich gleich (ICH: überprüfen, siehe clfduration csvs)
- EDER, does size, S. 170f.: weniger als 3000 Wörter schlechte Ergebnisse –> ICH: vllt als Erklärungsgrund für schlechte Klassifizierung des speeches korpus
- ICH: hier an evert sehr stark orientieren (Kapitel 3 und 4 angucken)

# 8 Schlussbetrachtung

TODO

- REMINDER: anfängliche Untersuchung / Fragestellungen
  - HAUPT: Burrows Delta besser als Machine Learning Verfahren? Im Hinblick auf Accuracy, Komplexität der Implementierung und Dauer
  - HAUPT: Mit welchen Feature Repräsentationen (Monogramme, Bigrammen, Trigrammen und Anzahl der Features) können die jeweiligen Klassifizierungsverfahren am besten arbeiten? Stimmen die Aussagen von Smith, Jockers oder Eder (siehe Kapitel 4.2)?
  - NEBEN: z-score mit KNN am besten? ohne evtl. besser? mit cos besser, stimmt das was SMITH2011 gesagt hat?
  - NEBEN: Was sind Probleme bei der Evaluation von Korpora? zu gute Klassifizierung, doch was bedeutet zu gute Klassifizierung? Bei Prosa-Korpus wahrscheinlich, dass game nicht closed genug, d.h. Stil lies sich zu einfach unterscheiden. Viele dieser Autoren könnten durch Expertise im Vorneherein ausgeschlossen werden

- Zu Forschungsfragen:
  - KOMPLEXITÄT DER IMPLEMENTIERUNG: Ähnlich komplex, da alle mit Sci-kit learn implementiert werden können
- Was sind Probleme bei der Evaluation von Korpora? zu gute Klassifizierung, doch was bedeutet zu gute Klassifizierung? Bei Prosa-Korpus wahrscheinlich, dass game nicht closed genug, d.h. Stil lies sich zu einfach unterscheiden. Viele dieser Autoren könnten durch Expertise im Vorneherein ausgeschlossen werden

# Literatur


Argamon, Shlomo (2008). „Interpreting Burrows's Delta: Geometric and Probabilistic Foundations“. In: *Literary and Linguistic Computing*, 23.2, S. 131–147.

Argamon, Shlomo, Moshe Koppel und Jonathan Schler (2009). „Computational Methods in Authorship Attribution“. In: *Journal of the American Society for Information Science and Technology* 60.1, S. 9–26.

Barbaresi, Adrien (2018). *A corpus of German political speeches from the 21st century*.

Bücher-Wiki (2019). *Prosa*. URL: `https://www.buecher-wiki.de/index.php/BuecherWiki/Prosa` (besucht am 03. 12. 2019).

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.

Eder, Maciej (2015). „Does size matter? Authorship attribution, small samples, big problem“. In: *Digital Scholarship in the Humanities* 30.2, S. 167–182.

— (2017). „Visualization in stylometry: Cluster analysis using networks“. In: *Digital Scholarship in the Humanities* 32.1, S. 50–64.

Eder, Maciej und Jan Rybicki (2013). „Do birds of a feather really flock together, or how to choose training samples for authorship attribution“. In: *Literary and Linguistic Computing* 28.2, S. 229–236.

Evert, Stefan u. a. (2017). „Understanding and explaining Delta measures for authorship attribution“. In: *Digital Scholarship in the Humanities* 32.2 (Supplement), S. ii4–ii16.

Fischer, Frank und Jannik Strötgen (6. Jan. 2017). *Corpus of German-Language Fiction*. URL: `https://figshare.com/articles/Corpus_of_German-Language_Fiction_txt_/4524680/1` (besucht am 03. 12. 2019).

Hoover, David (2004). „Testing Burrows' delta“. In: *Literary and Linguistic Computing* 19, S. 453–475.

Jockers, Matthew L. und Daniela M. Witten (2010). „A comparative study of machine learning methods for authorship attribution“. In: *Literary and Linguistic Computing* 25.2, S. 215–223.

Jockers, Matthew L., Daniela M. Witten und Craig Criddle (2008). „Reassessing authorship in the book of Mormon using nearest Shrunken centroid classification.“ In: *Literary Linguist Comput J Assoc Literary Linguist Comput* 23, S. 465–491.

Kocher, Mirco und Jacques Savoy (2018). „Distributed language representation for authorship attribution“. In: *Digital Scholarship in the Humanities* 33.2, S. 425–441.

Koppel, Moshe, Jonathan Schler und Shlomo Argamon (2009). „Computational methods in authorship attribution“. In: *Journal of the American Society for Information Science and Technology* 60.1, S. 9–26.

Schöch, Christof (2018). „Zeta für die kontrastive Analyse literarischer Texte. Theorie, Implementierung, Fallstudie“. In: *Quantitative Ansätze in den Literatur- und Geisteswissenschaften. Systematische und historische Perspektiven*. Hrsg. von Toni Bernhart u. a. Berlin: De Gruyter, S. 77–94.

Smith, Peter W.H. und W. Aldridge (2011). „Improving Authorship Attribution. Optimizing Burrows' Delta Method“. In: *Journal of Quantitative Linguistics* 18.1, S. 63–88.

Stamatatos, Efstathios (2009). „A Survey of Modern Authorship Attribution Methods“. In: *Journal of the American Society for Information Science and Technology* 60.3, S. 538–556.

Tabata, Tomoji (16. Juli 2012). *Approaching Dickens' Style through Random Forests*. URL: http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/approaching-dickens-style-through-random-forests.1.html (besucht am 03. 12. 2019).

Zhao, Ying und Justin Zobel (2005). „Effective and scalable authorship attribution using function words“. In: *Lecture Notes in Computer Science* 3689, S. 174–189.

# Appendix

## A Daten und Code

Im Ordner „app“ befindet sich die Python-Datei „utils.py“, welche Nachbearbeitungs- und Reduktionsfunktionen für die Korpora beinhaltet. Die Dateien „texts_to_csv.py“ und „classification.py“ sind CLI-Tools. Mithilfe von „texts_to_csv.py“ können csv-Dateien der Korpora erstellt werden. Mit dem Tool „classification.py“ wurden die verschiedenen Stilometrie-Experimente in dieser Arbeit durchgeführt. Mit „visualization.py“ werden die Abbildungen anhand der Klassifikations-Ergebnisse visualisiert. Zudem befinden sich im Ordner „app“ auch noch einige Jupyter-Notebooks, die zur Nachbearbeitung und Reduktion der Korpora genutzt und in der die Abbildungen, die sich in dieser Arbeit befinden, erzeugt wurden.

analysis.py noch erklären

in figures corpora analysis abbildungen für reduktion und so in figures results klassifizierungsergebnisse

TODO: erklären, wo tools, code –> github project TODO: erklären von speicherung der klassifizierungsergebnisse

# B Reduktion des Prosa Korpus

Das benutzte Korpus hat einige Probleme, weshalb es für die Untersuchungen in dieser Arbeit vorverarbeitet werden soll. Die Herangehensweise stützt sich dabei nicht auf eine literaturwissenschaftliche Wissensdomäne, sie soll eher auf schlichten Annahmen basieren. Die erste Veränderung soll eine Reduzierung des Korpus auf Werke sein, die zwischen 1840 und 1930 erschienen sind. Diese Aufteilung folgt der Aussage der Ersteller(AW?) des Korpus, dass der größte Teil der Werke des Korpus in diesem Zeitraum erschienen sei (Fischer und Strötgen, 2017). Dies wird durch Abbildung 1 (TODO) bestätigt, die eine Verteilung der Werke nach Erscheinungsjahren darstellt. Der Grund für die Reduzierung ist die Komplexität bei der Klassifizierung, welche durch ein kleineres Korpus verringert wird. Das reduzierte Korpus enthält nach der Reduzierung nur noch 2212 Werke, etwa zwanzig Prozent der Werke wurde entfernt. Die Anzahl der verschiedenen Autoren hat sich von 549 auf 439 verringert.
Die Ersteller(?) des Korpus merken einige bekannte Probleme mit dem Korpus an (Fischer und Strötgen, 2017). Fünf von neun dieser Probleme sind für eine Reduzierung uninteressant, da sie zum Beispiel außerhalb des Erscheinungszeitraums von 1840 und 1930 erschienen sind. Die anderen vier Probleme, bei denen es um Duplikate ging, wurden behoben, indem die problematischen Werke entfernt wurden. Nach der Entfernung bestand das Korpus noch aus 2208 Texten.
Die verschiedenen Prosagattungen der einzelnen Texte sind nicht bekannt. Dies ist problematisch, da die Menge an verschiedenen Textgattungen bei der Klassifikation einen Bias[14] erzeugen kann. Durch eine Reduzierung der Prosagattungen kann auch der Bias verringert werden. Da die Prosagattungen der einzelnen Texte jedoch unbekannt sind, werden hier die Textlängen als loser Indikator für die

[14] In dieser Arbeit wird der englische Begriff „Bias“ anstelle des deutschen Begriffs „systematischer Fehler“ benutzt.

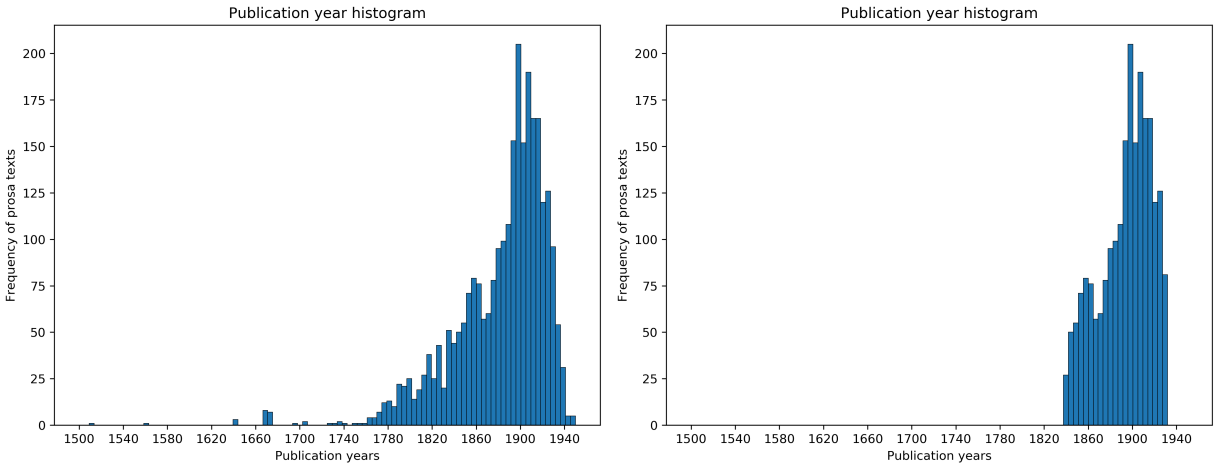



**Abbildung 3:** Das linke Histogramm zeigt die Verteilung der Häufigkeit der Erscheinungsjahre des unbearbeiteten Korpus. Wie von Strötgen und Fischer angemerkt, wurden die meisten Werke des Korpus im Zeitraum zwischen 1840 und 1930 erstellt. Das rechte Histogramm zeigt die Verteilung nach der Entfernung der Erscheinungsjahre außerhalb des Zeitraums 1840 bis 1930.

verschiedenen Prosagattungen verwendet. Die Annahme ist, dass kürzere Texte eine erhöhte Wahrscheinlichkeit haben, Prosagattungen wie „Kurzgeschichte“, „Essay“ oder „Brief“ anzugehören. Da es jedoch keine eindeutigen Grenzen für die Einteilung von Prosagattungen gibt, wurde hier der Mittelwert ($\overline{x} = 63870$) als Trennwert verwendet.[15] Dieser ist aufgrund der großen Menge an kurzen Texten verzerrt. Nachdem die kürzeren Texte mithilfe des Mittelwerts entfernt wurden, sollen auch sehr lange Texte aus dem Korpus entfernt werden. Dies hat vor allem Performance-Gründe. Zuletzt wurden Autoren aus dem Korpus entfernt, von denen im Korpus weniger als drei Texte vorhanden waren.

Neben der Reduktion des Korpus wurde noch einige Nachbearbeitungsmaßnahmen getroffen. Innerhalb der ursprünglichen Texte wurden zu Beginn Titel, Autor und Erscheinungsjahr des Textes genannt. Diese Informationen wurden aus den Texten entfernt. [16] Die Befürchtung war es, dass bei einer Beibehaltung der Meta-Informationen

[15] Siehe Abbildung 2. (TODO)

[16] Dies geschah in zwei Schritten: Zuerst wurde beim Einlesen der Texte ein Großteil der Meta-Informationen mithilfe eines regulären Ausdrucks entfernt. Dadurch konnten nicht alle Meta-Informationen beseitigt werden, da sich die Schreibweisen der Meta-Informationen von Titeln oder Autoren innerhalb des Textes teilweise geringfügig von den vorhandenen Meta-Informationen außerhalb des Textes unterschieden, wurde die Kosinus-Ähnlichkeit benutzt. Mit dieser wurde die Ähnlichkeit des Textanfangs jeweils mit Titel und Autor verglichen. Der Textanfang wurde dabei

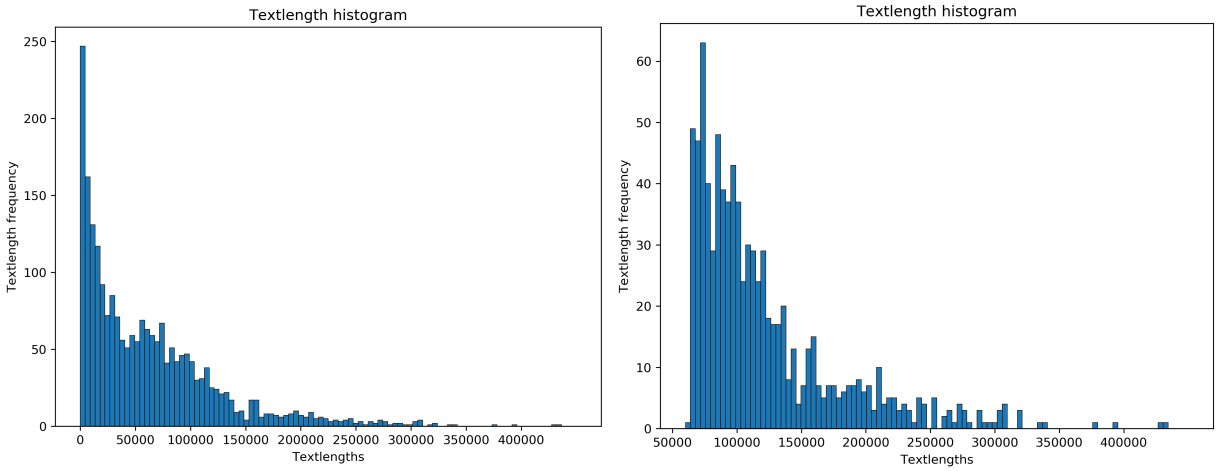



**Abbildung 4:** Die linke Grafik stellt die Häufigkeit von Texten anhand ihrer Textlänge dar. Sehr kurze Texte sind sehr häufig im Korpus und sehr lange Texte sind eher selten vertreten, die Verteilung ist also rechtsschief. Der Mittelwert der Gesamtheit aller Textlängen der Dokumente ist definiert als $\overline{x} = \frac{\sum L_D}{count(D_i)}$, wobei $L_D$ die Textlängen der einzelnen Texte und $count(D_i)$ die Gesamtzahl aller Texte repräsentiert. Der Mittelwert $\overline{x}$ wird als Trennwert verwendet und alle Texte, deren Länge kürzer als der Mittelwert sind, werden aus dem Korpus entfernt. Die Verteilung der Texte anhand ihrer Textlängen nach dieser Reduzierung wird in der rechten Grafik dargestellt.

im Text diese ein Einfluss auf die Ergebnisse der Klassifizierungen und stilometrischen Analysen haben und diese verfälschen könnten. Zuletzt wurden die Texte noch tokenisiert.

# C Ergebnisse der Experimente als Tabellen

Jede Seite jedes Unterkapitels stellt die Klassifizierungsergebnisse eines Korpus dar. Jede Tabelle auf jeder Seite steht für eine Vektorisierungsmethode. Um den Umfang der Tabellen zu reduzieren, wurden nur die Werte der optimierten Verfahren in den Tabellen angezeigt, für den Speicherort der gesamten Klassifizierungsergebnisse siehe Appendix A. Jede Tabellenzelle stellt den F1-Score dar, in Klammern dahinter

nach und nach vergrößert und iterativ mit dem Titel und Autor verglichen. Betrug die Ähnlichkeit gleich oder mehr als 60 Prozent, wurde der aktuelle Textanfang vom eigentlichen Text abgeschnitten und entfernt. Somit wurden auch einige andere Informationen als die Titel- und Autor-Informationen entfernt, die Menge ist jedoch im Vergleich der Textlänge so gering, dass dies keinen großen Einfluss auf die spätere stilometrische Analysen und Klassifizierungen haben sollte.

wird der Cross-Validation-F1-Score angezeigt. Das beste Klassifizierungsergebnis der Nearest Neighbor Verfahren und das beste Klassifizierungsergebnis der anderen Verfahren jeder Spalte wurden hervorgehoben. Dabei wurde sich primär an der Größe des F1-Scores orientiert. War der Unterschied zwischen dem F1-Score und dem Cross-Validation-F1-Score jedoch zu groß, wurde ein Verfahren gewählt, welches einen kleineren Unterschied aufwies. In den ersten drei Unterkapiteln werden die Ergebnisse der verschiedenen N-Gramm-Reichweiten dargestellt. TODO, WENN NOCH MEHR.

## C.1 Ergebnisse der Monogramm-Experimente

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **tKNN** | **0.731 (0.615)** | **0.74 (0.641)** | **0.767 (0.674)** | **0.809 (0.69)** | **0.833 (0.72)** | **0.829 (0.738)** |
| **tNSC** | 0.529 (0.499) | 0.581 (0.521) | 0.62 (0.56) | 0.672 (0.578) | 0.731 (0.621) | 0.739 (0.652) |
| **tMNB** | 0.899 (0.845) | 0.935 (0.878) | 0.941 (0.896) | 0.949 (0.913) | 0.966 (0.915) | 0.97 (0.926) |
| **tLSVM** | 0.95 (0.883) | 0.956 (0.888) | 0.963 (0.899) | **0.963 (0.899)** | **0.969 (0.895)** | **0.96 (0.902)** |
| **tLR** | **0.968 (0.945)** | **0.986 (0.955)** | **0.979 (0.953)** | 0.96 (0.9) | 0.938 (0.834) | 0.907 (0.793) |

**Tabelle 2:** Die Klassifizierungsergebnisse des originalen Prosa-Korpus, vektorisiert mit dem Bag-of-Words-Modell und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **D-tKNN** | **0.815 (0.727)** | **0.854 (0.768)** | **0.881 (0.798)** | **0.906 (0.829)** | **0.937 (0.844)** | **0.931 (0.859)** |
| **D-tNSC** | 0.738 (0.643) | 0.767 (0.697) | 0.797 (0.731) | 0.832 (0.779) | 0.895 (0.808) | 0.89 (0.817) |
| **tLSVM** | **0.989 (0.963)** | **0.99 (0.977)** | **0.993 (0.983)** | **0.995 (0.982)** | **0.999 (0.987)** | **0.996 (0.985)** |
| **tLR** | 0.972 (0.894) | 0.985 (0.918) | 0.985 (0.909) | 0.944 (0.827) | 0.912 (0.74) | 0.885 (0.696) |

**Tabelle 3:** Die Klassifizierungsergebnisse des originalen Prosa-Korpus, vektorisiert mit dem Z-Score und einer N-Gramm-Reichweite von (1,1). Multinomial Naive Bayes konnte nicht in der Kombination mit einer Z-Score Matrix verwendet werden.

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **tKNN** | **0.915 (0.849)** | **0.922 (0.863)** | **0.941 (0.889)** | **0.953 (0.905)** | **0.969 (0.914)** | **0.957 (0.919)** |
| **tNSC** | 0.884 (0.832) | 0.907 (0.848) | 0.922 (0.88) | 0.955 (0.897) | 0.972 (0.904) | 0.954 (0.904) |
| **tMNB** | 0.896 (0.847) | 0.915 (0.869) | **0.935 (0.896)** | **0.941 (0.877)** | **0.939 (0.872)** | **0.929 (0.864)** |
| **tLSVM** | **0.928 (0.819)** | **0.931 (0.842)** | 0.935 (0.854) | 0.935 (0.838) | 0.936 (0.824) | 0.917 (0.808) |
| **tLR** | 0.601 (0.354) | 0.598 (0.349) | 0.584 (0.361) | 0.591 (0.349) | 0.595 (0.358) | 0.565 (0.366) |

**Tabelle 4:** Die Klassifizierungsergebnisse des originalen Prosa-Korpus, vektorisiert mit TF-IDF und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **D-tKNN** | **0.39 (0.324)** | **0.361 (0.33)** | **0.389 (0.346)** | **0.4 (0.34)** | **0.39 (0.334)** | **0.384 (0.33)** |
| **tNSC** | 0.299 (0.275) | 0.267 (0.28) | 0.289 (0.286) | 0.286 (0.28) | 0.274 (0.272) | 0.293 (0.262) |
| **tMNB** | **0.385 (0.345)** | **0.363 (0.342)** | **0.372 (0.352)** | **0.369 (0.363)** | **0.388 (0.375)** | 0.411 (0.377) |
| **tLSVM** | 0.366 (0.283) | 0.321 (0.251) | 0.338 (0.243) | 0.367 (0.293) | 0.351 (0.308) | **0.419 (0.34)** |
| **tLR** | 0.384 (0.221) | 0.373 (0.23) | 0.389 (0.23) | 0.434 (0.233) | 0.468 (0.267) | 0.509 (0.273) |

**Tabelle 5:** Die Klassifizierungsergebnisse des originalen Prosa-Korpus, vektorisiert mit der Kosinus-Ähnlichkeit und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **tKNN** | **0.385 (0.391)** | **0.412 (0.401)** | **0.377 (0.418)** | **0.404 (0.441)** | **0.492 (0.454)** | **0.388 (0.44)** |
| **tNSC** | 0.265 (0.239) | 0.265 (0.268) | 0.319 (0.263) | 0.292 (0.288) | 0.281 (0.307) | 0.285 (0.303) |
| **tMNB** | **0.631 (0.576)** | **0.638 (0.629)** | **0.688 (0.638)** | **0.65 (0.677)** | **0.658 (0.69)** | **0.731 (0.685)** |
| **tLSVM** | 0.55 (0.5) | 0.558 (0.519) | 0.581 (0.554) | 0.6 (0.562) | 0.612 (0.572) | 0.612 (0.575) |
| **tLR** | 0.542 (0.509) | 0.527 (0.502) | 0.565 (0.542) | 0.542 (0.537) | 0.562 (0.554) | 0.535 (0.538) |

**Tabelle 6:** Die Klassifizierungsergebnisse des Reden-Korpus, vektorisiert mit dem Bag-of-Words-Modell und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **D-tKNN** | **0.458 (0.403)** | **0.392 (0.44)** | 0.408 (0.389) | 0.308 (0.351) | 0.235 (0.287) | 0.246 (0.238) |
| **D-tNSC** | 0.373 (0.332) | 0.369 (0.39) | **0.415 (0.439)** | **0.454 (0.468)** | **0.485 (0.495)** | **0.496 (0.497)** |
| **tLSVM** | **0.585 (0.512)** | **0.608 (0.615)** | **0.65 (0.663)** | **0.715 (0.719)** | **0.712 (0.723)** | **0.7 (0.725)** |
| **tLR** | 0.446 (0.406) | 0.435 (0.477) | 0.419 (0.441) | 0.454 (0.425) | 0.462 (0.422) | 0.512 (0.489) |

**Tabelle 7:** Die Klassifizierungsergebnisse des Reden-Korpus, vektorisiert mit dem Z-Score und einer N-Gramm-Reichweite von (1,1). Multinomial Naive Bayes konnte nicht in der Kombination mit einer Z-Score Matrix verwendet werden.

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **tKNN** | **0.504 (0.498)** | 0.585 (0.528) | 0.546 (0.562) | **0.592 (0.589)** | 0.577 (0.613) | **0.565 (0.576)** |
| **tNSC** | 0.504 (0.522) | **0.573 (0.541)** | **0.623 (0.568)** | 0.612 (0.563) | **0.573 (0.591)** | 0.558 (0.58) |
| **tMNB** | 0.523 (0.547) | **0.646 (0.573)** | **0.612 (0.612)** | **0.692 (0.639)** | **0.642 (0.667)** | **0.673 (0.674)** |
| **tLSVM** | **0.554 (0.557)** | 0.635 (0.578) | 0.627 (0.605) | 0.665 (0.626) | 0.65 (0.64) | 0.631 (0.663) |
| **tLR** | 0.45 (0.469) | 0.488 (0.451) | 0.488 (0.478) | 0.488 (0.419) | 0.423 (0.449) | 0.438 (0.432) |

**Tabelle 8:** Die Klassifizierungsergebnisse des Reden-Korpus, vektorisiert mit TF-IDF und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **D-tKNN** | **0.369 (0.393)** | **0.423 (0.407)** | **0.4 (0.435)** | **0.388 (0.444)** | **0.419 (0.435)** | **0.427 (0.432)** |
| **tNSC** | 0.342 (0.336) | 0.358 (0.317) | 0.319 (0.293) | 0.296 (0.345) | 0.369 (0.318) | 0.269 (0.319) |
| **tMNB** | 0.412 (0.429) | 0.454 (0.441) | 0.419 (0.469) | 0.454 (0.506) | 0.558 (0.519) | 0.508 (0.555) |
| **tLSVM** | **0.515 (0.493)** | **0.577 (0.538)** | **0.55 (0.586)** | **0.596 (0.611)** | **0.658 (0.611)** | **0.596 (0.644)** |
| **tLR** | 0.512 (0.482) | 0.531 (0.512) | 0.535 (0.549) | 0.55 (0.575) | 0.608 (0.571) | 0.581 (0.594) |

**Tabelle 9:** Die Klassifizierungsergebnisse des Reden-Korpus, vektorisiert mit der Kosinus-Ähnlichkeit und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **tKNN** | 0.808 (0.795) | 0.865 (0.805) | 0.822 (0.846) | **0.862 (0.858)** | **0.905 (0.867)** | **0.898 (0.876)** |
| **tNSC** | **0.802 (0.79)** | **0.822 (0.809)** | **0.84 (0.848)** | 0.855 (0.857) | 0.878 (0.853) | 0.868 (0.83) |
| **tMNB** | 0.842 (0.808) | 0.852 (0.823) | 0.878 (0.868) | 0.9 (0.883) | **0.95 (0.899)** | **0.928 (0.912)** |
| **tLSVM** | 0.882 (0.854) | 0.922 (0.847) | 0.9 (0.87) | **0.912 (0.878)** | 0.935 (0.88) | 0.92 (0.876) |
| **tLR** | **0.912 (0.865)** | **0.935 (0.862)** | 0.91 (0.867) | 0.928 (0.837) | 0.88 (0.786) | 0.852 (0.778) |

**Tabelle 10:** Die Klassifizierungsergebnisse des reduzierten Prosa-Korpus, vektorisiert mit dem Bag-of-Words-Modell und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **D-tKNN** | 0.852 (0.836) | 0.885 (0.861) | 0.905 (0.901) | **0.948 (0.903)** | **0.918 (0.918)** | **0.925 (0.893)** |
| **D-tNSC** | **0.888 (0.865)** | **0.898 (0.88)** | **0.918 (0.907)** | 0.933 (0.874) | 0.912 (0.811) | 0.86 (0.739) |
| **tLSVM** | **0.935 (0.931)** | **0.957 (0.935)** | **0.972 (0.962)** | **0.997 (0.978)** | **0.99 (0.979)** | **0.995 (0.983)** |
| **tLR** | 0.908 (0.85) | 0.903 (0.809) | 0.878 (0.797) | 0.852 (0.754) | 0.802 (0.679) | 0.718 (0.638) |

**Tabelle 11:** Die Klassifizierungsergebnisse des reduzierten Prosa-Korpus, vektorisiert mit dem Z-Score und einer N-Gramm-Reichweite von (1,1). Multinomial Naive Bayes konnte nicht in der Kombination mit einer Z-Score Matrix verwendet werden.

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **tKNN** | **0.828 (0.78)** | 0.85 (0.807) | 0.87 (0.828) | 0.867 (0.849) | **0.88 (0.871)** | **0.875 (0.874)** |
| **tNSC** | 0.803 (0.773) | **0.835 (0.81)** | **0.878 (0.857)** | **0.88 (0.866)** | 0.858 (0.817) | 0.855 (0.763) |
| **tMNB** | **0.82 (0.792)** | **0.855 (0.824)** | **0.885 (0.85)** | **0.908 (0.875)** | **0.89 (0.865)** | **0.92 (0.882)** |
| **tLSVM** | 0.862 (0.797) | 0.872 (0.809) | 0.898 (0.827) | 0.888 (0.841) | 0.87 (0.834) | 0.875 (0.808) |
| **tLR** | 0.617 (0.449) | 0.6 (0.459) | 0.57 (0.463) | 0.56 (0.431) | 0.48 (0.41) | 0.492 (0.395) |

**Tabelle 12:** Die Klassifizierungsergebnisse des reduzierten Prosa-Korpus , vektorisiert mit TF-IDF und einer N-Gramm-Reichweite von (1,1).

| | 200 | 300 | 500 | 1000 | 2000 | 3000 |
|---|---|---|---|---|---|---|
| **D-tKNN** | **0.318 (0.345)** | **0.35 (0.326)** | **0.315 (0.357)** | **0.375 (0.339)** | **0.348 (0.351)** | **0.338 (0.336)** |
| **tNSC** | 0.275 (0.302) | 0.322 (0.271) | 0.305 (0.287) | 0.293 (0.288) | 0.205 (0.242) | 0.198 (0.203) |
| **tMNB** | 0.345 (0.393) | 0.418 (0.366) | 0.415 (0.391) | 0.438 (0.412) | 0.42 (0.46) | 0.495 (0.456) |
| **tLSVM** | **0.49 (0.417)** | **0.48 (0.415)** | **0.515 (0.468)** | 0.63 (0.498) | **0.592 (0.6)** | **0.742 (0.637)** |
| **tLR** | 0.412 (0.304) | 0.458 (0.323) | 0.465 (0.335) | 0.535 (0.4) | 0.473 (0.463) | 0.628 (0.492) |

**Tabelle 13:** Die Klassifizierungsergebnisse des reduzierten Prosa-Korpus, vektorisiert mit der Kosinus-Ähnlichkeit und einer N-Gramm-Reichweite von (1,1).

# Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.
Ich erkläre weiterhin, dass die vorliegende Arbeit in gleicher oder ähnlicher Form noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Würzburg, den 2. Februar 2020 Autor