Zum Verständnis des LDA Topic Modeling: eine Evaluation aus Sicht der Digital Humanities

Inaugural-Dissertation zur Erlangung der Doktorwürde der
Philosophischen Fakultät
der
Julius-Maximilians-Universität Würzburg

vorgelegt von

Keli Du
Aus
VR China

2022

Erstgutachter/-in: Professor Dr. Fotis Jannidis
Zweitgutachter/-in: Professor Dr. Christof Schöch
Tag des Kolloquiums:27.03.2024

Inhalt

1. Einleitung

Topic Modeling ist eine weit verbreitete Methode der quantitativen Textanalyse. Eine der wichtigen Eigenschaft von Topic Modeling ist, dass hier keine annotierten Daten benötigt werden, vorliegen muss lediglich eine Sammlung der Dokumente in Wörter. Diese Methode kann daher schnell auf große Datenmengen angewendet werden und die latente thematische Struktur der Textsammlung aufdecken. Auf dieser Basis ist die Klassifizierung der Texte anhand der gefundenen Strukturen möglich. Die durch Topic Modeling erstellten Topics bestehen aus einer Reihe von Wörtern, die semantisch oder thematisch kohärent sind. Im Vergleich zu den einzelnen Schlagwörtern bieten die Topics offenbar mehr Informationen; ohne die einzelnen Texte durchzulesen, kann ein detaillierteres Verständnis ihres Inhalts gewonnen werden. Die vorhandenen Texte lassen sich schneller filtern und suchen, für den jeweiligen Zweck brauchbare Texte können leichter gefunden werden. Topic Modeling ist deswegen für die Bereiche wie z. B. Information Retrieval ein wertvolles Hilfsmittel.

Aufgrund der Vorteile des Topic Modeling hat sich diese Forschungsmethode in den vergangenen Jahren im Bereich Digital Humanities, also auf dem Feld der „digitalen Geisteswissenschaften“, verbreitet. Durch Topic Modeling wird das Bedürfnis erfüllt, Texte mithilfe von Computern vertiefend zu explorieren, zu verstehen, zu interpretieren und zu analysieren. Es existierten jedoch bereits vor dem Topic Modeling computergestützte Textanalysen. Tools bzw. Methoden wie Key Word In Context (KWIC) -Indexing, N-gram Viewer und Co-occurrence Analysis wurden eingesetzt, um Texte unter verschiedenen Aspekten zu untersuchen. Wörter werden durch die oben genannten Methoden, Tools und Disziplinen in ihren konkreten Kontext gestellt und analysiert. Im Vergleich dazu bezieht sich das Topic Modeling auf eine andere Ebene. Dazu sagt Jockers:

> „If our goal is to understand the narrative subjects and the recurrent themes and motifs that operate in the literary ecosystem, then we must go beyond the study of individual n-grams, beyond the words, beyond the KWIC lists, and beyond even the collocates in order to capture what is at once more general and also more specific. Cultural memes and literary themes are not expressed in single words or even in single bigrams or trigrams. Themes are formed of bigger units and operate on a higher plane … In short, simple word-to-word collocations and KWIC lists do not provide enough information to rise to the level of theme.“ (Jockers, 2013)

Durch die Analyse bestimmter Wörter, die in einem Satz zusammenstehen, lässt sich das Topic des Satzes ermitteln. Eventuell ist es auch möglich, das Topic des zugehörigen Absatzes oder den Inhalt eines ganzen Romans zu erkennen. Wenn der Forschungsgegenstand z. B. aus zahlreichen Romanen besteht, ist die wörterorientierte Methode allerdings nicht mehr geeignet. In diesen Fällen kann das Topic Modeling z. B. für die Literaturwissenschaft ein wertvolles Hilfsmittel dahingehend darstellen, als dass sich nicht nur der Inhalt des Textes analysieren lässt, sondern auch andere textanalytische Aufgaben (wie z. B. die Klassifikation des Textes) lösbar sind.

Allerdings ist das Topic Modeling selbst noch aufgrund seiner Komplexität in Bezug auf z. B. die verwendeten Algorithmen oder Parametereinstellungen noch nicht abschließend erforscht. Diese Situation könnte dazu führen, dass diese Methode nicht effektiv genutzt wird. Als Beispiel stelle man sich vor, es gelte das jeweils richtige Verkehrsmittel für die folgenden drei Reisen auszuwählen: zu einem Supermarkt in der Nähe des Wohnorts, zu einer Konferenz in einem anderen Land und zum Mond. Für die erste Reise würden die meisten Menschen wohl zu Fuß gehen, mit dem Rad fahren oder das Auto nehmen. Für die zweite Reise hätte wohl eine Mehrheit einen Zug oder ein Flugzeug präferiert. Für die dritte Mission dürfte wohl jeder eine Rakete oder ein Space-Shuttle wählen. Der Grund, aus dem es einen Grundkonsens über das richtige Verkehrsmittel gibt, liegt darin, dass ein gemeinsames klares Verständnis der folgenden zwei Aspekte existiert: Entfernung und Eigenschaft des Verkehrsmittels wie z. B. Transportkapazität oder Geschwindigkeit. Im Vergleich dazu haben wir jedoch kein klares Verständnis für die Auswirkungen der möglichen Parameter oder Faktoren auf das Topic Modeling oder das trainierte Topic-Modell. Daher werden häufig die gleichen, in den Topic-Modeling-Tools vorgegebenen Parametereinstellungen beim Training der Topic-Modelle verwendet, obwohl der Untersuchungskorpus sich in Bezug auf z. B. Sprache, Epoche unterscheidet. Das ist so, als würden Menschen mit demselben Verkehrsmittel zum Einkaufen, zu einer Konferenz und zu einer Weltraumreise aufbrechen.

Darüber hinaus ist beim Einsatz des Topic Modeling bekannt, dass diese Methode empfindlich auf die Einstellung der jeweiligen Parameter reagiert. Aus diesem Grund wird Topic Modeling oft kritisiert. In Da (2019) wird z. B. erläutert: „(Topic modeling) is extremely sensitive to parametricization, prone to overfitting, and is fairly unstable as an “aboutness” finder for sophisticated texts because you need only tweak small details to discover completely different topics.“ Um zu sehen, wie die Chunk-Length die gefundenen Topics beeinflussen kann, hat Da

zwei Robustheitstests durchgeführt. Ihre Beobachtung ist, dass selbst dann, wenn nur 1 % der Daten aus dem Korpus zufällig entfernt wird, sich alle Topics in dem trainierten Topic-Modell ändern, obwohl alle anderen Parameter unverändert bleiben. Nicht nur die Parameter des Topic-Modells (z. B. Hyperparameter Alpha und Beta), sondern auch die Vorverarbeitung der Textdaten (Chunking des Textes oder Entfernung der Stoppwörter etc.), können einen Einfluss auf das trainierte Modell haben. In Evert et al. (2017) wird durch systematische Untersuchungen gezeigt, dass Burrows' Delta leicht unterschiedlich funktioniert, wenn diese Methode für eine stilistische Analyse von englischen, französischen und deutschen Texten eingesetzt wird. Topic Modeling und der Delta-Test sind zwei quantitative Verfahren, die auf unterschiedlichen mathematischen Hintergründen basieren. In Bezug auf das Delta ist allerdings sinnvoll zu prüfen, ob die Sprache des Textes für das Topic Modeling ebenfalls eine Rolle spielt. Die vorgegebenen Parametereinstellungen von Topic-Modeling-Tools stammen in der Regel aus Untersuchungen auf Basis englischer Texte. Ob sie für das Topic Modeling auf deutschen Texten ebenfalls gut geeignet sind, ist unklar.

Abgesehen von der Festlegung der Parameter wurde das Verständnis von „Topic" häufig diskutiert. Es wird z. B. in Shadrova (2021) behauptet, dass Topic Modeling konzeptionell schwach ist, weil Topic Modeling keine sinnvolle Modellierung von Themen ist und die Validierung von Topic-Modell und Topics problematisch ist. Topic Modeling wurde ursprünglich für Information Retrieval Tasks oder die inhaltliche Exploration von Sachtexten wie z. B. Zeitungsartikeln oder Artikeln in wissenschaftlichen Zeitschriften entwickelt. Eigentlich wird in einer Fußnote des originalen LDA-Papers erläutert: „We refer to the latent multinomial variables in the LDA model as topics, so as to exploit text-oriented intuitions, but we make no epistemological claims regarding these latent variables beyond their utility in representing probability distributions on sets of words" (Blei et al., 2003). Es wird davon ausgegangen, dass die Topics als bestimmte Themen interpretiert werden können. Insgesamt ist ein Topic eine Wahrscheinlichkeitsverteilung von Wörtern und die Top-Wörter in einem Topic treten im Text häufig gemeinsam auf. Topics können das gemeinsame Auftreten von Begriffen aufdecken, die eventuell von Menschen beim Close Reading nicht in Betracht gezogen wurden.

Allerdings kann das gemeinsame Auftreten der Topic-Wörter nicht nur die semantische Relation der Wörter, sondern auch andere Verbindungen widerspiegeln. In der Praxis werden häufig Topics beobachtet, die nicht als Themen interpretierbar sind. Dies gilt insbesondere im

Bereich Digital Humanities, weil hier mit Textdaten aus unterschiedlichen Sprachen, Zeiträumen und Domänen mit unterschiedlichen Forschungszielen gearbeitet wird. Wie in Newman, Noh, et al. (2010) beschrieben wird, ist ein Topic von römische Zahlen „viii vii xii xiii xiv xvi xviii xix xvii ....“ für den thematischen explorativen Gebrauch wenig geeignet, obwohl die Wörter offensichtlich auch verwandt sind. In Tabelle 1.1 sind drei Beispieltopics dargestellt. Das erste Topic ist ein thematisches Topic, das als „Krankenhaus“ interpretiert werden kann. Im Vergleich zu diesem Topic sind andere zwei Topics keine thematischen Topics. Obwohl die Wörter in den zwei Topics nicht semantisch verwandt sind, sind sie für Menschen immer noch interpretierbar, nämlich als ein Fremdwörter-Topic und ein OCR-Fehler-Topic.

Topics	Topicwörter	Interpretation
Topic 1	kranken krankheit lager arztes fieber schmerzen bette leiden wunde genesung pflege kissen ärzte gesundheit wunden arzte sterbenden verwundeten puls	Krankenhaus
Topic 2	est vous mais tout mon qu petit pour französin on sur nous elle point enfin un grand chose enfant	Fremdwörter
Topic 3	deu uud nnd uicht eiu mau seiu anch siud werdeu mnß deutscheu eiue möglich eiuer lind nen laugen kaun wnrden	OCR-Fehler

Tabelle 1.1 Drei Beispieltopics und ihre möglichen Interpretationen

Ein anderes Beispiel ist das Topic 8 aus dem Topic-Modell „Klaus Mann“ [1]. Die ersten 20 Topic-Wörter sind „sagen, sehen, geben, lassen, bleiben, wissen, stehen, denken, scheinen, lachen, sprechen, fragen, finden, halten, nehmen, fast, kind, auge, schließlich, schön“. Offenbar handelt es sich hier nicht um ein thematisches Topic. In Abbildung 1.1 wird die Topic-Dokument-Verteilung des Modells visualisiert. Von oben nach unten sind die 30 Topics und von links nach rechts die Dokumente dargestellt. Je dunkler die blaue Farbe ist, desto höher ist

[1] Einige Informationen zum Topic-Modell: Das Modell wird auf eine Textsammlung trainiert, die aus Tagebüchern, der Autobiographie *Der Wendepunkt* und dem Roman *Flucht in den Norden* von Klaus Mann besteht. Es umfasst 30 Topics. Die Tagebücher-Sammlung besteht aus den Tagebucheinträgen von Klaus Mann zwischen dem 09.10.1931 und dem 31.12.1934. Jeder Eintrag ist ein Dokument. Der Roman *Flucht in den Norden* und die Autobiographie *Der Wendepunkt* wurden in Abschnitts-Chunks zerlegt. Insgesamt besteht die Textsammlung aus 2991 Dokumenten. Stoppwörter wurden aus dem Korpus entfernt und alle Texte sind lemmatisiert. Das Training des Modells wurde durch MALLET durchgeführt. Die vorgegebenen Parameter-Einstellungen von MALLET werden verwendet.

die Wahrscheinlichkeit, dass ein Topic in einem Dokument vorkommt. Es ist zu beobachten, dass das Topic 8 (rot markiert) mit hoher Wahrscheinlichkeit mit fast allen Roman-Segmenten verbunden ist. Im Vergleich dazu ist die Wahrscheinlichkeit einer Verbindung mit Topic 8 in Autobiografie-Segmenten kleiner, bei Tagebucheinträgen ist sie am geringsten. Dieses Phänomen entspricht dem allgemeinen Wissen über den Sprachstil der Textgenres: Die Sprache in Romanen ist eher literarisch, die in Tagebücher eher sachlich. In Autobiographien ist die Sprache weniger literarisch als in Romanen, aber doch weniger nüchtern als in Tagebüchern. Werden lediglich die Topic-Wörter beobachtet, lässt sich das Topic nur schwer interpretieren. Mithilfe von Metadaten aber kann es als „literarisches Sprachmittel" interpretiert werden.

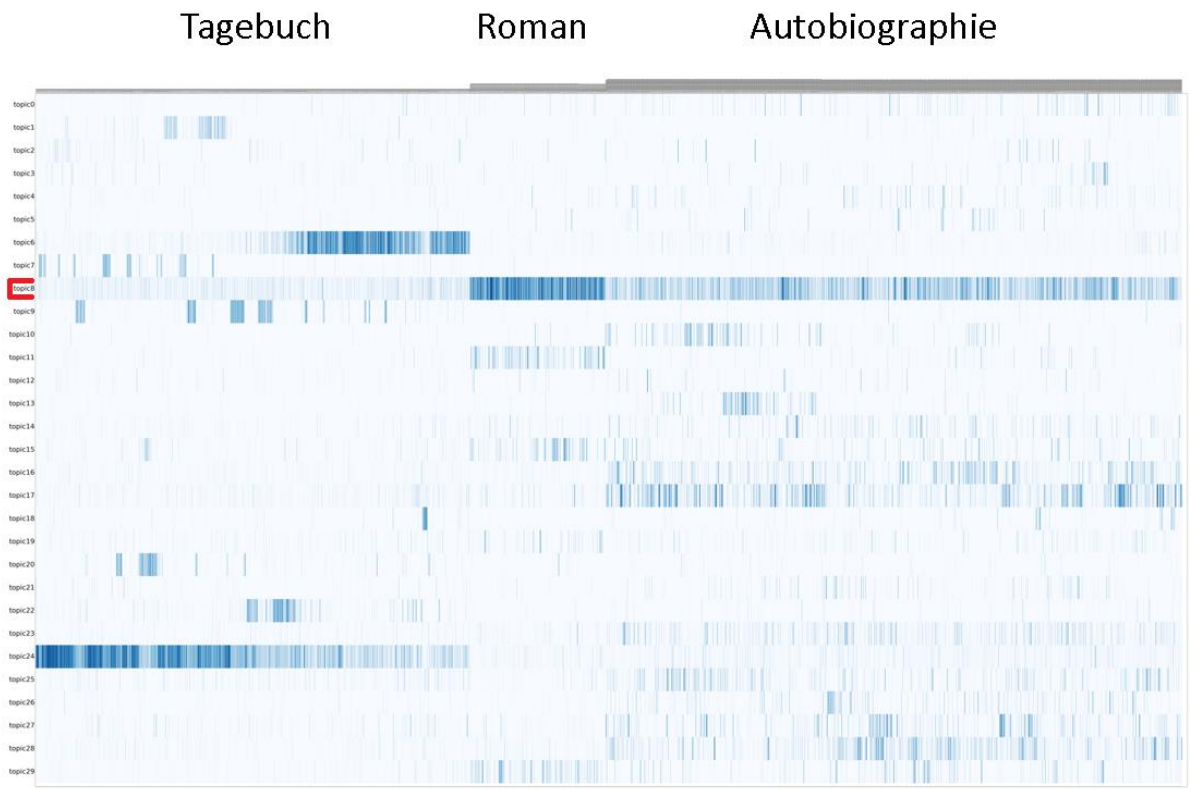

Abbildung 1.1 Topic-Dokument-Verteilung des Topic-Modells „Klaus Mann"

Bekannt ist, dass man durch Topic Modeling nicht immer Themen bekommt und die Begriffe „Topic Modeling" und „Topics" können dabei irreführend sein. Deshalb sollte man bei der Verwendung dieser Methode ständig daran denken, wie der Begriff „Topics" in Bezug auf die eigene Forschung zu verstehen ist. In Schöch (2017) wird z. B. eine Klassifikation von Topics vorgeschlagen: „In practice, individual words with the highest scores in a given topic are assumed to be semantically related words. This does not mean they must all belong to a common

abstract theme (such as justice or biology). Especially in literary texts, it is also common for the shared semantic basis of words in a topic to be a particular setting (such as interiors or natural landscape), a narrative motive (such as horse-riding or reading and writing letters), or a social group of characters (such as noblemen or family members).“ In Underwood (2012) z. B. wird vorgeschlagen, Topics als „discourse“ zu benennen: „I actually suspect that the topics identified by LDA probably always have the character of ‘discourses.’ They are, technically, ‘kinds of language that tend to occur in the same discursive contexts.’ But a ‘kind of language’ may or may not really be a ‘topic’.“

Aufgrund eines unzureichenden Verständnisses der Methode ist die leichte Skepsis angebracht, ob Topic Modeling ein stabiles Verfahren ist. Wenn ja, unter welchen Umständen ist diese Methode stabil? Aus diesem Grund ist es das Ziel dieser Dissertation, ein besseres und vertiefendes Verständnis des Topic Modeling zu gewinnen. Dabei gilt es, verschiedene Aspekte zu beachten. Als Vorarbeit müssen die Verwendung von Topic Modeling in Digital Humanities zuerst gut verstanden werden. Durch eine Analyse der bereits veröffentlichten Anwendungsbeispiele von Topic Modeling in Digital Humanities soll insbesondere geklärt werden, warum und wie Topic Modeling hier eingesetzt wird. So lässt sich ein Überblick über den Kenntnisstand von Topic Modeling in Digital Humanities gewinnen. Darüber hinaus soll der Untersuchungsgegenstand dieser Arbeit klar umrissen werden, es ist also zu bestimmen, welche Faktoren, die das Topic Modeling beeinflussen können, evaluiert werden sollen. Außerdem wird noch zusammengefasst, welche Methoden zur Evaluation von Topic-Modellen vorgeschlagen werden. Diese Zusammenfassung konzentriert sich dabei nicht nur auf die Evaluation des Topic-Modells selbst, sondern auch auf die Evaluation des Topics, die ja die zentrale Rolle bei der Exploration der Textdaten spielen. Ein solcher Überblick dient noch dazu, die geeignete Evaluationsmethode für die geplanten Untersuchungen herauszuarbeiten. Die Kernaufgabe der Dissertation umfasst die systematische Evaluation der zuvor bestimmten Faktoren. Das Ziel der Evaluation ist zu explorieren, wie sich die Qualität des Topic-Modells in Bezug auf die gewählten Evaluationskriterien verändert, wenn ein Faktor variiert wird. Dies bedeutet, dass lediglich der jeweils zu untersuchende Faktor geändert wird, die weiteren Parameter werden auf einen konstanten Wert gesetzt. Die Untersuchungen werden auf zwei deutschen Korpora (einem Sachtextkorpus und einem literarischen Textkorpus) durchgeführt, damit die möglichen Unterschiede zwischen den verschiedenen Textsorten dargestellt werden können.

Es ist wichtig hier zu betonen, dass das Ziel dieser Arbeit nur darin besteht, Topic Modeling aus zwei Perspektiven zu evaluieren, nämlich Topic-Modeling-basierte Dokumentklassifikation und Topic-Kohärenz. In dieser Arbeit wird nicht versucht, z. B. den Begriff „Topic“ präzis zu definieren, eine neue Evaluationsmethode des Topic Modeling vorzuschlagen, oder zwischen verschiedenen Arten von Topics zu unterschieden.

Diese Arbeit umfasst insgesamt acht Kapitel. Im Anschluss an diese Einführung wird im zweiten Kapitel ein Überblick über den Forschungsstand von Topic Modeling in Digital Humanities vorgestellt. Eine kurze Einführung in das Topic Modeling erfolgt dann im Unterkapitel 2.2. In den folgenden vier Unterkapiteln (2.3 bis 2.6) wird jeder der vier folgenden Aspekte der veröffentlichten Anwendungsbeispiele von Topic Modeling in Digital Humanities zusammengefasst: Forschungszwecke, Pre-processing des Textes, Training des Topic-Modells und Post-processing. Anschließend wird in Kapitel 3 die Evaluation des Topic Modeling vorgestellt. Das Unterkapitel 3.1 bezieht sich dann auf die Evaluation des Topic-Modells, das Unterkapitel 3.2 stellt die Evaluation des Topics vor. Nach dem theoretischen Teil folgt der praktische Teil dieser Dissertation. Zuerst wird in Kapitel 4 das Standardverfahren der automatischen Topic-Evaluation, Topic-Kohärenzmaß, analysiert, um die potenziellen Probleme dieser Evaluationsmethode genau zu verstehen. Kapitel 5 beschreibt die zwei Untersuchungskorpora und die für die Untersuchungen verwendeten Tools. Kapitel 6 und Kapitel 7 weisen dann eine identische Struktur mit je sechs Unterkapiteln auf, die die Untersuchungen der folgenden sechs Faktoren vorstellen:

- Anzahl der Topics
- Hyperparameter Alpha
- Hyperparameter-Optimierung
- Hyperparameter Beta
- Iteration des Gibbs-Samplings
- Chunk-length

Kapitel 6 wiederum bezieht sich auf die Untersuchungen eines deutschen Zeitungskorpus, Kapitel 7 beschreibt die Arbeit mit einem deutschen Romankorpus. Schließlich werden im 8. Kapitel die Untersuchungsergebnisse zusammengefasst, abschließend wird über noch offene Fragen bzw. potenzielle zukünftige Arbeiten diskutiert.

2. Forschungsstand zum Topic Modeling in den Digital Humanities

Topic Modeling ist ein probabilistisches Verfahren, das die latente semantische oder thematische Struktur einer Textsammlung aufdeckt. Durch Topic Modeling wird ein Zugang zur Verfügung gestellt, der es ermöglicht, den Inhalt von unstrukturierten Textdaten zu explorieren. Diese Methode stellt ein wertvolles Hilfsmittel für Forschungen im Bereich der Digital Humanities dar. Dies gilt nicht nur, weil so zahlreiche Textdaten exploriert werden können, sondern auch, weil darüber hinaus die semantischen Verknüpfungen zwischen mehreren Wörtern durch ein Topic widergespiegelt werden. Durch diese Topics werden mehr Informationen bereitgestellt, als dies bei wortorientierten Methoden wie z. B. der Kookkurrenz-Analyse der Fall ist. Aus diesem Grund ist Topic Modeling auch besser geeignet, um z. B. die narrativen Aspekte oder die Motive der vorliegenden Texte zu analysieren. Neben der inhaltlichen Textanalyse kann Topic Modeling auch für andere Aufgaben wie z. B. die Textklassifikation eingesetzt werden.

Als ein Werkzeug im Umfeld des „Distant Reading“ ist Topic Modeling (Blei, 2012) im Feld der Digital Huamnities weit verbreitet und wird häufig als ein möglicher Lösungsansatz für unterschiedliche Probleme der textanalytischen Forschungen eingesetzt (z. B. Rhody, 2012; Jockers, 2013; Hettinger et al., 2016; Schöch, 2017). Die Suchergebnisse für „topic model“ in „The Index of Digital Humanities Conferences“[2] zeigen (siehe Abbildung 2.1), dass Topic Modeling in den vergangenen Jahren deutlich zunehmend als eine Forschungsmethode in Digital Humanities eingesetzt wird. In diesem Kapitel wird eine Analyse der bisherigen Anwendungen im Forschungsfeld der Digital Humanities vorgestellt, um einen Überblick zum Einsatzbereich der Methode zu erhalten. Die Anwendung des Topic Modeling umfasst in der Regel folgende Schritte: Erstellung eines Untersuchungskorpus, die Vorverarbeitung (Pre-processing) der Texte, Training des Topic-Modells, die Evaluation des Modells und das Post-processing. Im Folgenden wird nun analysiert, wie die oben genannten einzelne Schritte durchgeführt werden. Das Ziel ist es dabei, Antworten auf folgende Fragen zu erhalten:

- Für welchen Forschungszwecke wird Topic Modeling im Bereich Digital Humanities eingesetzt?

[2] https://dh-abstracts.library.virginia.edu, (17.09.2023).

- Pre-processing: Welche Methoden werden für das Pre-processing des Textes verwendet und aus welchen Gründen?
- Modeling: Wie werden die Parameter (Topic-Anzahl, Hyperparameter, etc.) beim Training des Topic-Modells eingestellt? Welches Tool, welche Software wird verwendet, um Topic-Modelle zu trainieren?
- Post-processing: Was sind die weiteren Schritte nach dem Training des Topic-Modells?

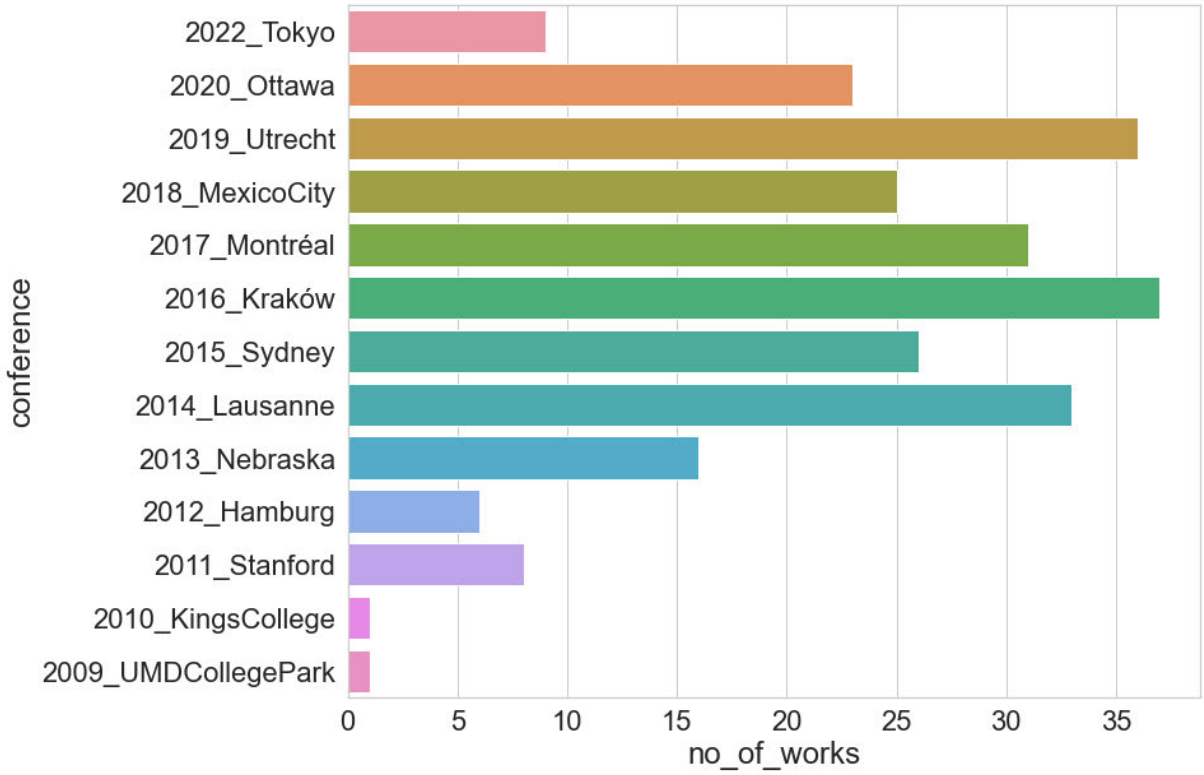

Abbildung 2.1 Die zeitliche Verteilung von Topic-Modeling-Anwendungsfällen veröffentlicht in den Jahrestagungen „Digital Humanities“ der ADHO zwischen 2009 und 2022

Um einen genaueren Überblick über die Anwendung des Topic Modeling im Feld der Digital Humanities zu gewinnen, werden 53 Anwendungsbeispiele aus den Books of Abstracts der Jahrestagung „Digital Humanities“ der Alliance of Digital Humanities Organizations (ADHO) zwischen 2011 und 2018 gesammelt. Neben den Abstracts aus DH-Tagungen wurden sieben Aufsätze als zusätzliche Anwendungsbeispiele aus zwei international von Experten begutachteten Zeitschriften (*Digital Scholarship in the Humanities* (DSH) und *Digital Humanities Quarterly* (DHQ)) gesammelt. Der Vorteil besteht hier darin, dass der Umfang der Aufsätze größer als der eines Abstracts (500 bis 1500 Wörter) ist. Zudem hat, während ein

Abstract für einen Vortrag auf einer Tagung häufig nur auflistet, welche Forschungsanstrengungen unternommen wurden, ein Aufsatz eine eher argumentative Grundstruktur. Es ist zu erwarten, hier ausführlichere Beschreibungen der Anwendung von Topic Modeling zu finden. Insgesamt wurden auf diese Weise 60 Anwendungsbeispiele/Papers für diese Metaanalyse gesammelt. Die Liste der 60 gesammelten Anwendungen ist im Anhang der Arbeit zu finden. Wenn die gesammelten Anwendungsbeispiele im folgenden Text erwähnt werden, werden sie fett geschrieben, wie z.B. **Fankhauser et al. 2016**.

In Kapitel 2.1 wird zunächst die theoretische Seite von Topic Modeling (LDA-Modell, Topic-Modeling-Tools usw.) kurz vorgestellt. Das Kapitel 2.2 liefert dann einen Überblick der gesammelten Anwendungen. Es wird analysiert, für welche Probleme im Bereich der Digital Humanities Topic Modeling eine geeignete Forschungsmethode darstellt. In den Abschnitten 2.3, 2.4 und 2.5 werden dann die Anwendungen aus drei Perspektiven analysiert werden, nämlich in Bezug auf das Pre-processing, das Modeling und das Post-processing.

2.1 Eine kurze Vorstellung des Topic Modeling

Bevor das Ergebnis der Metaanalyse detailliert vorgestellt wird, soll dieses Unterkapitel zunächst die theoretischen Grundlagen des Topic Modeling umreißen.

2.1.1 LDA und LDA Topic Modeling

Bei der Latent Dirichlet Allocation (LDA) handelt es sich um ein generatives Modell. Dieses modelliert einen fiktiven Prozess, in dem ein Dokument generiert wird. Es wird angenommen, dass ein Text eine Mischung von mehreren Topics ist, während jedes Topic eine Wahrscheinlichkeitsverteilung einer festen Menge von Wörtern repräsentiert. Ein Wort kann mit bestimmten Wahrscheinlichkeiten zu einem oder zu mehreren Topics gehören. Um ein Dokument zu generieren, wird zuerst zufällig eine Wahrscheinlichkeitsverteilung von Topics gewählt. Danach wird ein Topic zufällig aus den Topics gewählt und aus diesem Topic wird ein Wort zufällig gewählt. So wird ein einzelnes Wort des Dokumentes bestimmt. Dieser Prozess wird dann wiederholt, bis das Dokument fertig generiert wird (Blei, 2012).

Abbildung 2.2 zeigt eine graphische Darstellung des LDA-Modells von $\boldsymbol{M}$ Dokumenten und $\boldsymbol{K}$ Topics. Jedes Dokument enthält $\boldsymbol{N}$ Wörter. $\boldsymbol{\alpha}$ und $\boldsymbol{\beta}$ sind jeweils die A-priori-Verteilung der

Topic-Dokument-Verteilung $\boldsymbol{\theta}$ und der Topic-Wort-Verteilung $\boldsymbol{\varphi}$. $\boldsymbol{Z}$ steht hier für die Topic-Zuteilung jedes Wortes in jedem Dokument, $\boldsymbol{W}$ steht für Wörter im Dokument. Als die einzige sichtbare Variable wird $\boldsymbol{W}$ hier ausgegraut dargestellt.

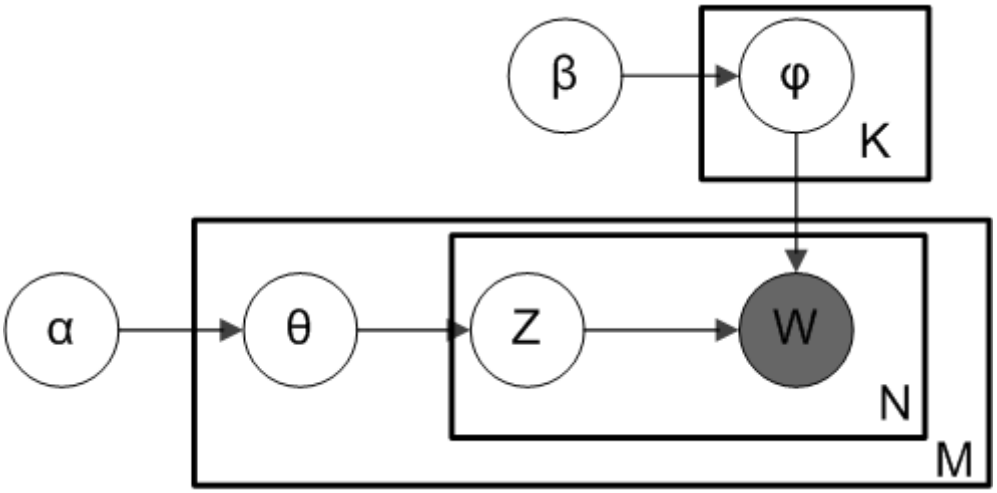

Abbildung 2.2 LDA-Modell[3]

Das LDA Topic Modeling kann als die Umkehrung des vorgestellten generativen Prozesses betrachtet werden. Gegeben ist eine Dokumentensammlung. Die unsichtbare Topic-Wort-Verteilung und die Topic-Dokument-Verteilung sollen durch Topic Modeling abgeleitet werden (Blei & Lafferty, 2009). Zwischen den Wörtern, die in einem Text oder Textabschnitt häufig gemeinsam vorkommen, besteht oft ein semantischer Zusammenhang. Sie werden durch Topic Modeling in einem Topic gruppiert. Deshalb können die Topics als Themen interpretiert werden und die inhaltliche Struktur der Textsammlung widerspiegeln. Die Ableitung der Topic-Wort- und der Topic-Dokument-Verteilung aufgrund des Korpus kann durch mehrere Verfahren erfolgen. Beim ursprünglichen LDA-Paper (Blei et al., 2003) wird ein approximatives variationales Inferenzverfahren eingesetzt. Alternativ können zwei weitere wichtige Methoden, nämlich die Expectation Propagation (Minka & Lafferty, 2002) und das Gibbs-Sampling (Griffiths & Steyvers, 2004) eingesetzt werden. Im Vergleich zum approximativen variationalen Inferenzverfahren ist das Gibbs-Sampling eine genauere Anpassungsmethode, während variationale Inferenzverfahren wie z. B. Online Variational Bayesian (Online VB) einfacher zu parallelisieren und zu konvergieren sind (Rehurek, 2014).

[3] https://upload.wikimedia.org/wikipedia/commons/4/4d/Smoothed_LDA.png, (20.02.2019).

2.1.2 Topic-Modeling-Tools

Es sind häufig Programmierkenntnisse erforderlich, wenn Topic Modeling durchgeführt werden soll. Die Vorverarbeitung des Textes, das Training des Topic-Modells sowie die Visualisierung und die Evaluation des Modells sollen automatisch durch Computerprogramme vorgenommen werden. Für die Vorverarbeitung des Textes wie z. B. die Lemmatisierung, die Segmentierung oder das Part-of-Speech (POS) -Tagging können Natural Language Processing (NLP) -Tools wie Stanford NLP[4], TreeTagger[5] oder das Python-Paket Spacy[6] eingesetzt werden. Für das Training eines Topic-Modells gibt es mehrere mögliche Optionen. In Tabelle 2.1 werden acht Topic-Modeling-Softwarelösungen aufgelistet, die die unterschiedlichen Inferenzalgorithmen (variationale Methoden, Gibbs-Sampling, etc.) in verschiedenen Programmiersprachen (C/C++, Java, etc.) implementieren. Nach der Erklärung bei Asuncion et al. (2009) hat die Wahl der Inferenzmethode nur geringe Auswirkungen auf die durch Topic Modeling abgeleiteten Topics.

Tool	**Inferenzalgorithmen**	**Programmiersprache**
C-Implementation[7]	Variational Bayesian Expectation Maximization (VEM)	C
MALLET (McCallum, 2002)[8]	Gibbs-Sampling	Java
Gensim (Rehurek & Sojka, 2010)[9]	Online Variational Bayes (VB)	Python
Python-LDA[10]	Collapsed Gibbs-Sampling	Python
Stanford Topic Modeling Toolbox (Ramage et al., 2009)[11]	Collapsed Gibbs-Sampling	Scala
R Package „topicmodels“ (Hornik & Grün, 2011)[12]	VEM/Gibbs-Sampling	R

[4] https://nlp.stanford.edu/software/, (20.02.2019).
[5] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/, (20.02.2019).
[6] https://spacy.io/, (12.03.2021).
[7] http://www.cs.columbia.edu/~blei/lda-c/, (20.02.2019).
[8] http://mallet.cs.umass.edu/topics.php, (20.02.2019).
[9] https://radimrehurek.com/gensim/, (20.02.2019).
[10] https://lda.readthedocs.io/en/latest/index.html, (20.02.2019).
[11] https://nlp.stanford.edu/software/tmt/tmt-0.4/, (20.02.2019).
[12] https://cran.r-project.org/web/packages/topicmodels/index.html, (20.02.2019).

R Package „lda“[13]	Collapsed Gibbs-Sampling	R
C++ Implementation (Phan & Nguyen, 2010)[14]	Gibbs-Sampling	C/C++

Tabelle 2.1 Topic-Modeling-Tools

2.1.3 Faktoren, die LDA Topic Modeling beeinflussen können

Ein Topic-Modeling-Prozess kann durch mehrere Faktoren beeinflusst werden. In diesem Unterkapitel wird ein Überblick über diese Faktoren gegeben, bevor sie separat analysiert werden.

Anzahl der Topics: Beim Training eines Topic-Modells wird zunächst die Frage gestellt, wie viele Topics trainiert werden sollen. Der Grund dafür liegt darin, dass eine unterschiedliche Anzahl der Topics unterschiedliche Topic-Modelle erzeugt. Abbildung 2.3 zeigt ein Beispiel der Ergebnisse der Dokumentklassifikation, die auf Topic-Modelle mit einer unterschiedlichen Anzahl von Topics basieren. Auf der X-Achse findet sich die Anzahl der Topics, die von 10 auf 500 erhöht wird. Auf der Y-Achse sind die Ergebnisse der Klassifikation, genauer gesagt die F1 (Makro)-Werte dargestellt. Für jede Einstellung der Anzahl der Topics werden zehn Topic-Modelle trainiert, die entsprechenden F1-Werte werden durch ein Boxplot dargestellt. In der Visualisierung lässt sich hier die Veränderung der Klassifikationsergebnisse mit der Erhöhung von Anzahl der Topics deutlich erkennen.

[13] https://cran.r-project.org/web/packages/lda/lda.pdf, (20.02.2019).
[14] http://gibbslda.sourceforge.net/, (20.02.2019).

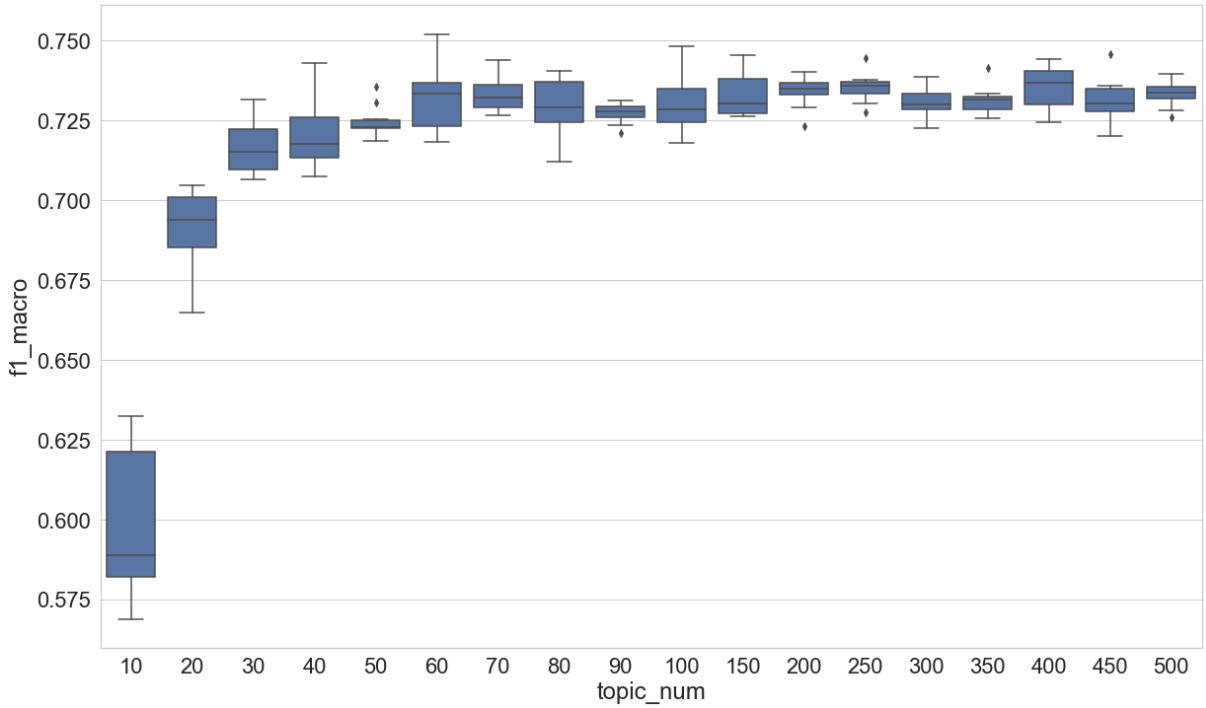

Abbildung 2.3 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation

Hyperparameter: Hyperparameter haben im Kontext von LDA Topic Modeling zwei unterschiedliche Bedeutungen: Sie stellen sowohl Parameter des LDA-Modells als auch Parameter des Inferenzalgorithmus dar.

- Parameter des LDA-Modells sind die Hyperparameter, die seine Eigenschaften bestimmen. Wie in der Abbildung 2.2 dargestellt, ist Alpha ($\boldsymbol{\alpha}$)der Hyperparameter einer Dirichletverteilung, die die A-priori-Verteilung der Topic-Dokument-Verteilung $\boldsymbol{\theta}$ ist. Beta ($\boldsymbol{\beta}$) ist der Hyperparameter einer anderen Dirichletverteilung, die die A-priori-Verteilung der Topic-Wort-Verteilung $\boldsymbol{\varphi}$ repräsentiert. Dies meint, dass die Form der beiden Dirichletverteilungen jeweils durch Alpha und Beta kontrolliert wird. Der Hyperparameter Alpha bzw. die A-priori-Verteilung der Topic-Dokument-Verteilung $\boldsymbol{\theta}$ entscheiden darüber, welche Topics nach unserer Erwartung mit welchen Wahrscheinlichkeiten in jedem Dokument vorkommen. Der Hyperparameter Beta bzw. die A-priori-Verteilung der Topic-Wort-Verteilung $\boldsymbol{\varphi}$ entscheiden darüber, welche Wörter nach unserer Erwartung mit welchen Wahrscheinlichkeiten in jedem Topic vorkommen.

- Parameter des Inferenzalgorithmus sind die Einstellungen, die das Verfahren des Topic Modeling bestimmen, mit dem wir die Parameter des LDA-Modells einstellen.[15] Als Parameter des Inferenzalgorithmus haben Alpha und Beta eine „Smoothing"-Wirkung auf die Topic-Dokument- und die Topic-Wort-Verteilung, die durch Topic Modeling abgeleitet werden sollen. Alpha stellt sicher, dass die Wahrscheinlichkeit jedes Topics in jedem Dokument während des gesamten Inferenzverfahren nicht 0 ist, während Beta garantiert, dass die Wahrscheinlichkeit jedes Wortes in jedem Topic während des kompletten Prozesses ungleich 0 ist. Noch wichtiger ist, dass Alpha und Beta jeweils die Annahme über die Daten repräsentieren, wie die Topics in Dokumenten und wie die Wörter in Topics verteilt sind. Anders gesagt, der Parameter Alpha beeinflusst, wie oft jedes Topic in jedem Dokument vorkommt und der Parameter Beta beeinflusst, wie oft jedes Wort in jedem Topic auftaucht.[16] Aus diesem Grund kann die Einstellung von Alpha und Beta die Inferenz der Topics und die Qualität des Topic-Modells beeinflussen.

Hyperparameter Alpha als potenzielle Störfaktor: Der Supervisor von MALLET, David Mimno, erklärte den Einfluss des Hyperparameters Alpha wie folgt: Ein großer Wert von Alpha kann die Topic-Dokument-Verteilung eher zu einer Uniformverteilung führen. Die Topics sind eher gleichermaßen wahrscheinlich auf alle Dokumente bezogen. Umgekehrt entsteht durch einen kleinen Wert von Alpha eine eher spärliche Topic-Dokument-Verteilung: Die Topics konzentrieren sich nicht auf alle, sondern jeweils auf einige Dokumente.[17] In MALLET werden zwei Parameter „SumAlpha" und „alpha_k" eingeführt. Der SumAlpha-Wert ist eine Konstante/Zahl und die vorgegebene Einstellung des Werts ist 5. Der alpha_k-Wert bezeichnet den eigentlichen Alpha-Wert jedes Topics. Er ist gleich SumAlpha-Wert geteilt durch die Anzahl der Topics. [18] Dies erlaubt, dass sich der Wert von Alpha mit der Änderung der Topic-Anzahl ändert. Werden zahlreiche Topics trainiert, ist der Alpha-Wert jedes Topics relativ klein. Umgekehrt ist der Wert größer, wenn weniger Topics trainiert werden. Wegen dieses Ansatzes der Implementierung ist bei der Verwendung von MALLET zu beachten, dass die Änderung der Topic-Anzahl gleichzeitig den Alpha-Wert jedes Topics verändert. Aus diesem Grund wird der Unterschied in den Klassifikationsergebnissen z. B. in Abbildung 2.3 eigentlich durch die

[15] Vgl. https://stackoverflow.com/questions/59458102/topic-modeling-with-mallet-topic-keys-output-parameter/59471230#59471230, (01.05.2020).
[16] Vgl. https://stackoverflow.com/questions/45162186/mallet-topic-modeling-topic-keys-output-parameter, (01.05.2020).
[17] Vgl. https://stackoverflow.com/questions/45162186/mallet-topic-modeling-topic-keys-output-parameter, (17.09.2019).
[18] https://github.com/mimno/Mallet/blob/master/src/cc/mallet/topics/LDA.java, (17.09.2019).

Änderung von zwei Faktoren (Anzahl und Alpha-Wert der Topics) verursacht. Werden Topic-Modelle mit MALLET trainiert und soll nur der Einfluss eines Faktors evaluiert werden, ist der Unterschied zwischen „SumAlpha“ und „alpha_k“ zu berücksichtigen.

Hyperparameter-Optimierung: Standardmäßig werden ein symmetrisches Alpha und symmetrisches Beta beim Training des Topic-Modells verwendet. Das bedeutet, dass der Alpha-Wert aller Topics gleich eingestellt wird und dass der Beta-Wert aller Wörter auch gleich eingestellt wird. Dadurch wird angenommen, dass alle Topics die gleiche Chance haben, einem Dokument zugeordnet zu werden, und dass alle Wörter die gleiche Chance haben, einem Topic zugeordnet zu werden. In Wallach et al. (2009) wurde allerdings gezeigt, dass ein asymmetrisches Alpha bessere Topic-Modelle erzielen kann, während Beta symmetrisch bleiben sollte. Bei der Inferenz der Topics wird zunächst voreingestellt, dass der Alpha-Wert aller Topics gleich ist. Mit der Hyperparameter-Optimierung wird der Alpha-Wert aller Topics im Inferenz-Prozess neu berechnet. Dadurch wird eine Annäherung des asymmetrischen Alpha erstellt. Die Hyperparameter-Optimierung erlaubt es dem Modell, sich besser an die Daten anzupassen, indem sie zulässt, dass einige Topics dominanter als andere sind. Anders gesagt, einige Topics werden eher allgemein und stark mit mehreren Dokumenten verbunden und anderen Topics treten eher spezifisch und konzentriert in einem bestimmten (kleinen) Teil der Dokumente auf. Da der Alpha-Wert aller Topics durch die Hyperparameter-Optimierung beim Topic Modeling ständig neu berechnet werden, ist der Einsatz der Hyperparameter-Optimierung ebenfalls ein Faktor, der das Ergebnis des Topic Modeling beeinflusst.

Iteration des Gibbs-Samplings: Die Iteration steuert, wie oft ein Topic-Modell auf dem Untersuchungskorpus trainiert werden soll. Da die Topic-Wort-Verteilung und die Topic-Dokument-Verteilung am Anfang des Trainings mit zufälligen Werten initialisiert werden, sind die beiden Verteilungen bei frühen Iterationen nicht unbedingt repräsentativ für die tatsächlichen Verteilungen. Aus diesem Grund ist es notwendig, Topic-Modelle mit einer höheren Anzahl von Iterationen zu trainieren. Allerding bedeutet dies, dass der Trainings-Prozess mehr Zeit in Anspruch nimmt. In einem Pilotversuch auf einem Korpus von 27 Millionen Tokens (ca. 234 Mb groß) hat es jeweils ca. 2:30 Stunden, 3:50 Stunden und 9:20 Stunden gedauert, um Topic-Modelle mit je 1000, 2000 und 5000 Iterationen zu trainieren. Der Test wurde auf einem normalen Laptop durchgeführt, das einen Intel® Core™ i5-4210M CPU @ 2.6GHz-Prozessor hatte, MALLET wurde als Topic-Modeling-Tool verwendet. Um das Training zu beschleunigen, kann die Multithreading-Funktion (das gleichzeitige/quasi-

gleichzeitige Abarbeiten mehrerer Threads) von MALLET verwendet werden. Das Training mit 1000 Iterationen auf demselben Korpus durch zwei Kernel-Threads hat nach dem Testergebnis ca. 1:45 Stunden in Anspruch genommen.

Chunk-Length: Beim Topic Modeling wird das Bag-of-Words-Modell vorausgesetzt und die Kookkurrenz der Wörter in jedem „Bag"/Dokument überprüft. Die Reihenfolge der Wörter im Dokument spielt deshalb keine Rolle. Die Länge des Dokuments könnte aber ein Topic-Modell beeinflussen. Je größer das „Bag" ist, desto mehr Wörter werden tendenziell zusammen in ihm gefunden. In den Digital Humanities (vor allem in den Computational Literary Studies) wird häufig mit längeren Texten wie z. B. Romanen gearbeitet. Zum Beispiel in *Sternstunden der Menschheit* von Stefan Zweig, werden zwar hauptsächlich die bedeutsamen Begebenheiten und ihre Auswirkungen erzählt. Es existieren in diesem Buch aber insgesamt 14 historische Geschichten, die thematisch durchaus unterschiedlich sind. Verschiedene Themen kommen deshalb entweder wiederholt als der „rote Faden" im Verlauf des gesamten Textes vor oder sie tauchen nur kurzzeitig in gewissen Teilen des Textes auf. Sind beide Arten von Themen für die inhaltliche Analyse sinnvoll, sollten längere Texte in mehrere Abschnitte segmentiert werden, da andernfalls die kurzzeitigen Themen nicht sichtbar werden. Deshalb ist der Einfluss der Chunk-Length auf das Topic Modeling ein wichtiger Faktor für die Anwendung entsprechender Modelle im Feld der Digital Humanities.

Zu diesem Thema wird bei Jockers (2013, S.134) diskutiert: „Novels tend to have some themes that run throughout and others that appear at specific points and then disappear. In order to capture these transient themes, it was useful to divide the novels into "chunks" and run the model over those chunks instead of over the entire text." Er hat Topic-Modelle auf eine Sammlung von 3346 englischen Romanen mit unterschiedlichen Chunking-Methoden trainiert, um den Einfluss der Chunk-Length zu analysieren. Die Romane werden in zehn gleich lange Chunks, in seitenbasierte Chunks, in paragraphbasierte Chunks, in 250-konsekutives-Substantiv-Chunks und in n-Token-Chunks zerlegt. Laut dem Autor kann ein 1000-Substantiv-Token-Chunk die am besten zu interpretierenden Topics produzieren. Außerdem wird nur ein geringer Unterschied beobachtet, wenn Topic-Modelle (mit 500 Topics) auf 1000-Substantiv-Token-Chunks oder auf 1500-Substantiv-Token-Chunks trainiert werden. In Boyd-Graber et al. (2017) wird zu dem Thema im Kapitel „What is a Document?" weiter diskutiert. Topic Modeling geht davon aus, dass das Verhältnis zwischen den Themen in einem Text oder in einem Dokument unverändert bleibt. Aber diese Annahme ist in der Realität nichtzutreffend.

Vor allem wird ein Roman nicht veröffentlicht, wenn sich keine thematische Veränderung/Entwicklung im Buch vorhanden sind. Aus diesem Grund werden ungenaue oder nicht-kohärente Topics durch Topic Modeling produziert, wenn ein Roman als ein einzelnes Dokument behandelt wird. Lange Texte sollten deshalb in Chunks segmentiert werden, um kürzere Kontexte zu erstellen. Eine perfekte Segmentierung sollte die Grenzen zwischen den thematischen Bereichen in einem Text identifizieren und den Text in Chunks zerlegen, die sich jeweils auf relativ wenige Themen fokussiert. Ein Textsegment sollte sich deswegen mit hoher Wahrscheinlichkeit auf weniger Topics konzentriert. Im Gegensatz dazu kann eine schlechte Segmentierung diese Grenzen in der Regel kaum berücksichtigen. Jeder Chunk wird deshalb mit hoher Wahrscheinlichkeit mit mehreren Topics verbunden. In Bezug auf dieser Problematik wird in Algee-Hewitt et al. (2015) eine komparative Analyse über die Effekte der Segmentierung auf das Topic Modeling durchgeführt. Ein Korpus von Romanen aus dem 19. Jh. wird in 200-Token-Segmente, 82-Token-Segmente[19] und in Paragraph-Segmente zerlegt. Der Herfindahl-Index wird dann für die Topic-Konzentration jedes Chunks eingesetzt. Dieser Index misst die Konzentration einer Verteilung. Höhere Werte deuten eine höhere Konzentration an. Das Testergebnis zeigt, dass das Topic-Modell jeweils die höchste und die niedrigste Topic-Konzentration hat, wenn das Korpus in Paragraph-Segmente und in 200-Token-Segmente zerlegt wird. Der Paragraph ist also, zumindest in diesem Romankorpus aus dem 19. Jh., die bessere thematische Einheit. Durch diese Untersuchung wird die allgemein bekannte Tatsache bestätigt, dass der Paragraph die thematische Einheit ist. Deshalb haben die Autoren vorgeschlagen: „If one wants to use topic modeling to analyze literature – then paragraphs are a better unit than “mechanical” segments[20], and should replace them in future research.“

2.1.4. Andere Topic-Modelle

Nachdem David Blei das LDA-Modell vorgestellt hatte, wurden mehrere Erweiterungen entwickelt und präsentiert, um Einschränkungen zu beheben oder den Ansatz durch zusätzliche Daten zu verbessern. Im Folgenden werden einige Erweiterungen des LDA-Modells aufgelistet: HMM-LDA(Griffiths et al., 2004), TagLDA (Zhu et al., 2006), Correlated Topic Model (Blei & Lafferty, 2006a); Dynamic Topic Model (Blei & Lafferty, 2006b), Hierarchical Dirichlet Process Model (Teh et al., 2006); Labeled LDA (Ramage et al., 2009), SentenceLDA (Balikas,

[19] 82 Token ist die durchschnittliche Länge der Paragraphen im Untersuchungskorpus.
[20] Hier ist der Begriff „‚mechanical segments“ das Synonym des n-Token-Segments.

Amini, et al., 2016), CopulaLDA (Balikas, Amoualian, et al., 2016). Parallel zu LDA und seinen Erweiterungen existieren einige andere Topic-Modeling-Algorithmen, die ebenfalls häufig in den Bereichen Information Retrieval oder Digital Humanities eingesetzt werden, z. B. Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Analysis (PLSA), Non-Negative Matrix Factorization (NMF) und BERTopic.

2.1.4.1. Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) wird in einigen Fällen auch als „Latent Semantic Analysis“ (LSA) bezeichnet. Aus theoretischer Sicht bezeichnen diese beiden Termini die gleiche Methode, in der Praxis besteht allerdings ein Unterschied. LSI und LSA können als die Verwendung einer Methode in unterschiedlichen Gebieten angesehen werden. LSI wird dabei im Bereich Indexierung, LSA dagegen für zahlreiche anderen möglichen Aufgaben wie z. B. die Textanalyse eingesetzt wird. LSI basiert auf der Approximation der Dokument-Term-Matrix. Thomas Hofmann betont in seiner Arbeit: „The key idea of LSA is to map documents (and by symmetry terms) to a vector space of reduced dimensionality, the latent semantic space“ (Hofmann, 1999). Für das Mapping bzw. die Approximation wird die Singulärwertzerlegung (auf Englisch: Singular Value Decomposition, SVD) eingesetzt. SVD kann als ein Faktorisierungsverfahren betrachtet werden und meint, dass ein Objekt in mehrere Faktoren zerlegt wird. Hier bezieht sich das Verfahren auf die Zerlegung einer Dokument-Term-Matrix. Durch SVD wird die originale Matrix durch weitaus weniger Dimensionen repräsentiert, ihre Komplexität wird verkleinert. Manning sagt hierzu: „When forced to squeeze the terms/documents down to a k-dimesional space, the SVD should bring together terms with similar co-occurrences“ (Manning et al., 2008). So wird die Untersuchung der Ähnlichkeiten zwischen den Begriffen in den Texten möglich.

Manning und Ko-Autoren legen in ihrem Buch auch die Vor- und Nachteile des LSI dar. Ein Nachteil der SVD ist der große Berechnungsaufwand. Außerdem konnte bisher kein Korpus mit dem LSI erfolgreich untersucht werden, wenn dieses mehr als eine Million Texte enthielt. Die Untersuchung von Manning wurde allerdings im Jahr 2008 veröffentlicht. Es wäre möglich, dass es inzwischen erfolgreiche Untersuchungen gibt. Manning betont: „This has been the biggest obstacle to the widespread adoption to LSI.“ Er vertritt auch die Meinung, dass das LSI als „soft clustering“ gesehen werden kann. Anders als beim „hard clustering“ kann ein Element bzw. Text beim soft clustering zu mehreren Clustern gehören. Die k-dimensionale Matrix

repräsentiert k verschiedene Cluster und „the value that a document has on that dimension as its fractional membership in that cluster“. Im engeren Sinn ist das LSI kein Topic Model, weil das Verfahren keine Topics produziert, die latente Variablen sind. Die Methode ermöglicht jedoch eine Berechnung der Term- und Dokumentvektoren. Auf der Grundlage des LSI konnte das Verfahren der Probabilistic Latent Semantic Analysis (PLSA) entwickelt werden.

2.1.4.2. Probabilistic Latent Semantic Analysis (PLSA)

Im Jahr 1999 analysierte Thomas Hofmann das LSI aus statistischer Sicht und entwickelte darauf aufbauend ein neues Modell, welches eine probabilistische Variante darstellt – die Probabilistic Latent Semantic Analysis (PLSA). Anders als die LSI, welche die Texte anhand von semantischen Räumen abbildet, stammt die PLSA nicht aus der linearen Algebra. Die PLSA nimmt an, dass ein Text mehrere unterschiedliche Themen (auf Englisch „Topics“) umfasst. Ein Text wird als eine Mischung von Themen gesehen, ein Topic ist eine Mischung aus Wörtern. Diese Mischung aus Wörtern lässt sich in Dokumente beobachten. Im Gegensatz dazu ist es unmöglich, die Mischung von Themen zu registrieren. Folglich liegt hier eine latente Variable vor. Nach dem grundlegenden Gedanken der PLSA wird ein Text in den folgenden Schritten generiert: Zunächst wird eine Mischung möglicher Themen der Textsammlung bestimmt. Für jedes Topic steht eine Reihe von Wörtern zur Verfügung. Dann wird jeder Text mit den entsprechenden Wörtern generiert.

Abbildung 2.4 zeigt eine graphische Darstellung des PLSA-Modells von ***M*** Dokumenten. Jedes Dokument enthält ***N*** Wörter. ***d*** ist ein Dokument, ***c*** ist die Topic-Zuteilung des Wortes ***w*** in diesem Dokument. Die zwei sichtbaren Variablen ***d*** und ***w*** werden ausgegraut dargestellt, während ***c*** eine unsichtbare Variable ist.

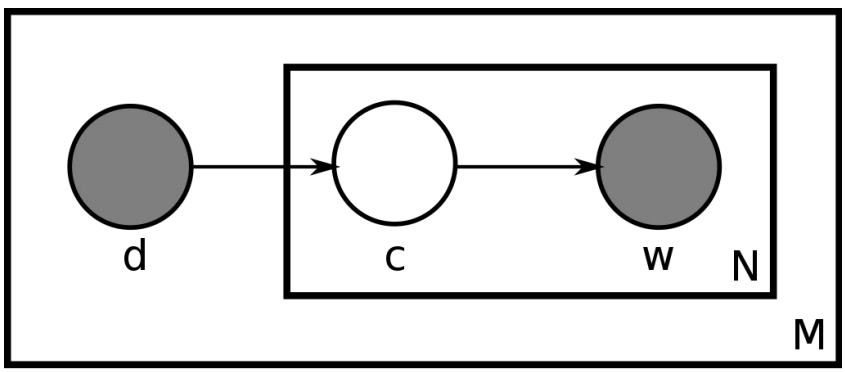

Abbildung 2.4 PLSA-Modell[21]

Beim PLSA-Topic Modeling wird der obige Prozess umgekehrt durchgeführt. Ähnlich wie bei der LSA wird versucht, ohne Wörterbücher oder Thesauri die Polysemien eindeutig zu machen und verwandte Wörter zu gruppieren. Der Expectation-Maximization-Algorithmus wird für die Anpassung des Modells eingesetzt, um die größte Wahrscheinlichkeit der latenten Variable („Topic“) abzuschätzen.

2.1.4.3. Non-Negative Matrix Factorization (NMF)

Die Non-Negative Matrix Factorization (NMF) ist dank der Untersuchungen in Paatero & Tapper (1994) und Lee & Seung (1999) bekannt geworden. NMF ist eine Faktorisierungsmethode, die eine Matrix in zwei Matrizen zerlegt. Die beiden Matrizen müssen non-negativ sein, d. h., alle Werte in diesen zwei Matrizen müssen größer oder gleich 0 sein. Wenn NMF für Topic Modeling eingesetzt wird, bezieht die Faktorisierung sich auf die Dokument-Term-Matrix ***D*** einer Textsammlung (Abbildung 2.5). Gelegentlich wird auch die tf-idf-gewichtete Dokument-Term-Matrix als Input der Faktorisierung genommen. Der Output bzw. das ***K***-Topics-Modell besteht aus Matrizen, nämlich der Topic-Term-Matrix ***U*** und der Topic-Dokument-Matrix ***V***.

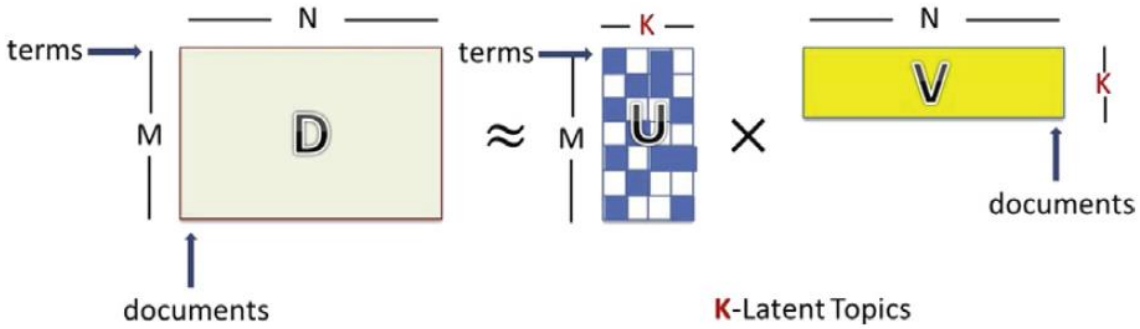

Abbildung 2.5 Non-Negative Matrix Factorization (Chen et al., 2019)

In Stevens et al. (2012) und Chen et al. (2019) werden LDA, LSA und NMF komparativ analysiert um herauszufinden, welches Modell bessere Topics produzieren kann. Beide

[21] https://en.wikipedia.org/wiki/Probabilistic_latent_semantic_analysis#/media/File:Plsi_1.svg, (17.09.2019).

Untersuchungen haben das UCI-Kohärenzmaß[22] als Evaluationsmethode der Topics verwendet. Das Korpus in Stevens et al. (2012) ist eine Sammlung von 92.600 Artikeln aus der „New York Times“, während das Korpus in Chen et al. (2019) eine Sammlung von kurzen Texten (die durchschnittliche Textlänge ist hier geringer als 15 Tokens) ist. Aus den Ergebnissen der Untersuchungen können folgende Schlussfolgerungen gezogen werden:

1. LSA produziert weniger kohärente Topics als NMF und LDA, die Methode ist aber für die Dokument-Klassifikation besser geeignet.
2. Der durchschnittliche Kohärenzwert der LDA-Topics ist höher als der der NMF-Topics. NMF produziert mehr nicht-kohärente Topics als LDA und LSA. Die besten 10 % der kohärenten NMF-Topics haben allerdings höhere durchschnittliche Kohärenzwerte als die besten 10 % der LDA-Topics.
3. Wenn ein Topic-Modell auf eine Sammlung von Kurztexten trainiert wird, haben NMF-Topics höhere durchschnittliche Kohärenzwerte als LDA-Topics. „More information encoded and definite algorithms probably make NMF tend to produce higher-quality topics than LDA (which leverages stochastic Gibbs sampling without enough word co-occurrences for learning and inference) from short texts.“ (Chen et al., 2019)

2.1.4.4 BERTopic

Neben den klassischen Bag-of-Words-basierten Topic-Modellen sind mit der schnellen Entwicklung der Word Embedding in den letzten Jahren auch Modelle entstanden, die z. B. Bidirectional Encoder Representations from Transformers (BERT, Devlin et al. 2018) einbeziehen. Ein typisches Beispiel dafür ist BERTopic (Grootendorst, 2022). Abbildung 2.6 veranschaulicht den Algorithmus sowie die Komponenten von BERTopic. Der Kern des Verfahrens besteht zunächst darin, jedes Dokument im Untersuchungskorpus mithilfe eines vortrainierten Sprachmodells (z. B. BERT-Modell) in eine Dokument-Embedding-Repräsentation zu transformieren. Auf Basis dieser Embedding-Repräsentation erfolgt ein Clustering-Prozess, um die Dokumente zu gruppieren. Für jedes Cluster wird eine Liste der bedeutendsten Wörter extrahiert, und diese Listen bilden die Topics.

In Egger & Yu (2022) werden vier Topic Modeling Algorithmen LDA, NMF, Top2Vec (Angelov, 2020) und BERTopic zur Analyse der Twitter-Daten verglichen. Aufgrund des

[22] In Kapitel 3.2.7 werden Topic-Kohärenz und. Kohärenzmaß ausführlich erläutert.

unterschiedlichen Charakters der Algorithmen, die Autoren haben nur LDA mit NMF und Top2Vec mit BERTopic verglichen. Basiert auf menschlicher Interpretation der trainierten Topics wurde festgestellt, wenn man Twitter-Daten analysieren möchten, sind BERTopic und NMF zu empfehlen, gefolgt von Top2Vec und LDA. Angesichts der Tatsache, dass es sich bei den Texten von Tweets im Wesentlichen um kurze Texte handelt, haben die Ergebnisse dieser Studie jedoch nur einen begrenzten Referenzwert für die Anwendungen in den Digital Humanities, die häufig lange Texte untersuchen.

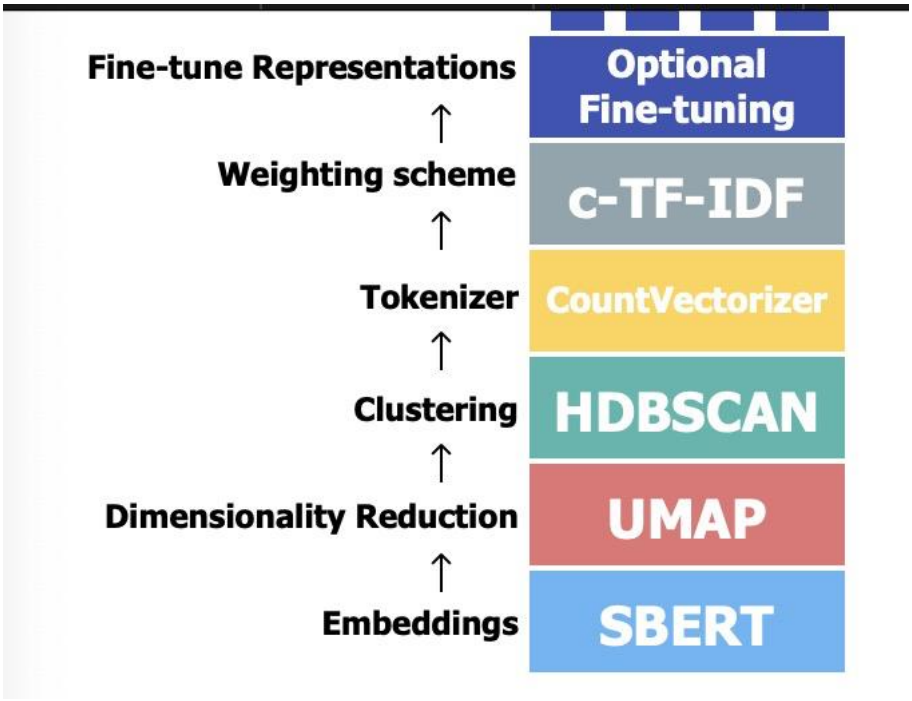

Abbildung 2.6 Der Algorithmus von BERTopic[23]

Disclaimer: Das Topic-Modell besteht, wie oben erläutert, aus einer Reihe von mehreren Modellen, die auf unterschiedlichen Algorithmen basieren. Da das LDA-Modell das am meisten verbreitete Topic-Modell im Feld der Digital Humanities ist, beziehen sich die Begriffe „Topic Modeling“ und „Topic-Modell“ in der vorliegenden Arbeit lediglich auf das LDA-Modell, wenn keine zusätzlichen Informationen angegeben werden.

2.2 Forschungszwecke

[23] https://maartengr.github.io/BERTopic/algorithm/algorithm.html, (15.10.2023).

Nach der theoretischen Einführung in das Topic Modeling gilt es nun, die Anwendungen dieser Methode im Feld der Digital Humanities zu betrachten. Zunächst stellt sich die Frage, für welche Forschungszwecke Topic Modeling in den gesammelten 60 Anwendungsbeispielen eingesetzt wird.

Topic Modeling wird hier häufig als eine Distant-Reading-Methode verwendet, wenn man die thematische Struktur einer großen Textsammlung analysieren möchten. Im Vergleich zu den anderen inhaltsanalytischen Methoden wie z.B. Key Word In Context (KWIC)-Indexing oder Kookkurrenz-Analyse, die die Wörter in ihren konkreten Kontext stellen und analysieren, kann Topic Modeling mehr Informationen aus unstrukturierten Texten extrahieren, da ein Topic eine Gruppe von semantisch oder thematisch kohärenten Wörtern enthält. Ein Thema oder ein Diskurs im Kontext der Kultur oder Geschichte kann dabei nur schwer durch einzelne Wörter widergespiegelt werden. Topic-Modelle (Topics und die Verbindung zwischen Topics und Dokumenten) können deshalb die textanalytische Anforderung erfüllen, die Erzählgegenstände, wiederkehrende Themen und Motive in Texten auf einer generellen Ebene zu verstehen (Jockers, 2013). In Boyd-Graber et al. (2014) wird kurz zusammengefasst, dass Topic Modeling auch für unterschiedliche computerlinguistischen Forschungsaufgaben eingesetzt werden kann: Zusammenfassung mehrerer Dokumente (Haghighi & Vanderwende, 2009), Disambiguierung der Wörter (Brody & Lapata, 2009), Sentimentanalyse (Titov & McDonald, 2008), maschinelle Übersetzung (Eidelman et al., 2012), Information Retrieval (Wei & Croft, 2006), Diskursanalyse (Purver et al., 2006) und Kennzeichnung von Bildern (Fei-Fei & Perona, 2005). Statt Topics direkt für die Untersuchungen zu verwenden, werden sie auch häufig als Feature in einem größeren Algorithmus benutzt. Im Vergleich dazu liegt der Fokus der Untersuchungen im Bereich Digital Humanities oft auf den Topics selbst. Die gesammelten Beispiele verdeutlichen, dass die Methode vor allem für geschichtliche Forschungen und in der Literaturwissenschaft verwendet wird; zwei Drittel der Anwendungsfälle entfallen auf diese beiden Forschungsbereiche. Darüber hinaus wird Topic Modeling auch z. B. in der Religionswissenschaft oder im Bereich der digitalen Archäologie eingesetzt.

Topic Modeling wird vor allem für den explorativen Textvergleich verwendet. Normalerweise werden Topic-Modelle trainiert und die Topics werden von Menschen interpretiert. Die interessantesten Topics werden anschließend mit Metadaten der Dokumente wie z. B. Nation, Gender, Autor, Gattung usw. kombiniert und analysiert, um inhaltliche Unterschiede/Ähnlichkeiten zwischen Textgruppen zu analysieren (z. B. **Chao et al., 2018; Du,**

2017; Miller et al., 2016; Roe et al., 2014; Jockers, 2011). Eine weitere beliebte Anwendung ist es, die zeitliche Entwicklung der Topics zu visualisieren und zu beobachten. Dadurch lässt sich zum Beispiel herausfinden, zu welchen Zeiten ein bestimmtes Topic häufig diskutiert wird und wann das Interesse wieder nachlässt (z. B **Fischer et al., 2018; Maryl & Eder, 2017; Schöch, 2015; Riddell, 2011; Blevins, 2011**). Darüber hinaus werden Topics in **Organisciak & Franklin (2017)** und **Bartz et al. (2014)** für die Disambiguierung der Wörter eingesetzt. In **Broadwell & Tangherlini (2015)** werden Topics mit geographischen Daten verbunden und in Zusammenhang mit einer Landkarte gebracht um herauszufinden, wie folkloristische Erzähler in ihren Texten Landschaften konzipiert haben. Neben der explorativen Untersuchung gibt es in **Falk (2016)** auch den Versuch, die Topic-Wörter in Texten zu beobachten, zu analysieren und das Close-Reading des Textes zu unterstützen. In **Binder & Jennings (2014)** bezieht die Analyse sich nicht auf die Verbindung zwischen Topics und Metadaten, sondern auf den Vergleich zwischen Topics und historischen thematischen Ressourcen (ein Sachregister aus dem 18. Jahrhundert). Eine spezielle Anwendung von Topic Modeling ist die Untersuchung in **Eder et al. (2018)**. Statt Textdaten zu analysieren, wird Topic Modeling hier auf unstrukturierten chemischen Daten verwendet, um sinnvolle chemische Strukturen zu entdecken. Dieser Ansatz basiert darauf, dass Topic Modeling sinnvolle thematische Strukturen aus unstrukturierten Textdaten extrahieren kann.

In den oben genannten Fällen liegt der Fokus der Forschung hauptsächlich auf der Suche nach einem oder mehreren interessanten Topics und der weiteren Analyse der Topics (Visualisierung, Vergleich, etc.). Im Vergleich dazu wird die Topic-Dokument-Verteilung des Topic-Modells in anderen Anwendungen eingesetzt, um Dokumentenklassifikationen oder Dokument-Clustering durchzuführen. In **Schöch (2017)** und **Hettinger et al. (2016)** wird zum Beispiel die Topic-Modeling-basierte Klassifikation von literarischen Untergattungen durchgeführt. Auf der Grundlage der Verhältnisse zu den entsprechenden Topics werden Zeitungsartikel in **Fitzgerald & Cordell (2018)** nach ihren Gattungen gruppiert. In **Fischer et al. (2018)**, **Wevers et al. (2018)** und **Fankhauser et al. (2016)** wird Topic Modeling auch für das Dokumenten-Clustering eingesetzt. Außerdem ist die Topic-Dokument-Verteilung eines Topic-Modells eine Repräsentation der Assoziationen zwischen jedem Topic und jedem Dokument, die man in einem Netzwerk für weitere Untersuchung visualisieren kann. In **Jockers (2012)** wird jedes Buch als ein Knoten eines Netzwerks betrachtet. Da jedes Buch ein N-dimensionaler Vektor ist (N ist die Topic-Anzahl des Modells), kann die Distanz zwischen zwei Vektoren ihre Beziehung zueinander repräsentieren. Diese Distanz wird dann als Kante im Netzwerk visualisiert, die

zwei Knoten verbindet. Ein ähnlicher Versuch ist in **Shawn (2013)** zu finden, wo jedes Topic als ein Knoten visualisiert wird.

2.3 Pre-processing des Textes

Bevor ein Topic-Modell auf einem Korpus trainiert wird, wird normalerweise eine Phase der Vorverarbeitung der Textdaten durchlaufen. In den 60 Anwendungsbeispielen wird in 22 Fällen die eingesetzte Vorverarbeitung des Textes vorgestellt. Ihr Ziel ist es vor allem, die Interpretierbarkeit der trainierten Topics zu erhöhen. Die unnötigen Informationen in Texten (sinnlose Zeichenketten, Satzzeichen, Stoppwörter, Zahlen, Groß- und Kleinschreibung des Wortes, evtl. XML-Tags, etc.) müssen entfernt werden, wenn sie nichts zur thematischen oder semantischen Interpretierbarkeit des Topics beitragen können. POS-Tagging wird häufig verwendet, um die weniger relevanten Wortarten (z. B. solche ohne lexikalische Bedeutung) zu identifizieren und aus dem Korpus herauszufiltern. Je nach dem aktuellen Forschungsziel werden verschiedene Wortarten aus dem Korpus entfernt oder beibehalten. In **Hammond & Brooke (2018), Norén & Mähler (2017)** und **Schöch (2015)** werden nur Substantive im Korpus beibehalten. Bei **Lee (2018)** verbleiben Substantive und Verben im Korpus. **Schöch (2017)** wiederum lässt auch Adjektive und Adverbien zu. Im Vergleich dazu werden in **Clivaz et al. (2017)** nur die Funktionswörter aus dem Korpus entfernt. In **Hettinger et al. (2016), Falk (2016)** und **Jautze et al. (2016)** werden neben den allgemeinen Stoppwörtern auch Personennamen entfernt, weil diese keine thematische Bedeutung haben. Im Gegensatz dazu werden Personennamen in **Yamada & Inoue (2015)** beibehalten, weil hier die Beziehungen zwischen Personen mithilfe des Topic Modeling analysiert werden sollen. In **Organisciak & Franklin (2017)** und **Jähnichen et al. (2015)** werden selten vorkommende Wörter auch als Stoppwörter betrachtet und aus dem Korpus entfernt. Neben der Entfernung der Stoppwörter ist es auch sinnvoll, eine Lemmatisierung einzusetzen, wenn mit Texten in einer flektierenden Sprache wie Französisch oder Deutsch gearbeitet wird, weil die unterschiedlichen Wortformen eines Wortes eigentlich dieselbe lexikalische Bedeutung ausdrücken.

Die Interpretierbarkeit des Topics zu erhöhen ist nicht das einzige Ziel der Verarbeitung der Textdaten vor dem Topic Modeling. Als eine andere typische Methode des Pre-processing wird besonders bei längeren Texten auch die Segmentierung eingesetzt, um die enthaltenen Themen genauer in den Blick nehmen zu können. In **Maryl & Eder (2017)** wird zum Beispiel berichtet, dass die Segmentierung nicht eingesetzt wird, weil die zu untersuchende Texte relativ kurz sind.

Insgesamt wird in 14 Anwendungsbeispielen die jeweils verwendete Strategie der Segmentierung vorgestellt. In **Hammond & Brooke (2018), Norén & Mähler (2017), Du (2017), Falk (2016)** und **Schöch (2015)** werden die Texte in Segmente mit einer Länge von ***n*** Tokens zerlegt. Die Einstellungen von ***n*** sind aber unterschiedlich (125, 150, 500 und 1000), wodurch die Länge der meisten Segmente gleichbleibt; lediglich das letzte Segment ist kürzer als ***n***. Diese Methode macht es möglich, dass ein Satz, ein Paragraph oder ein Kapitel in zwei Segmente zerlegt wird, was die inhaltliche Struktur brechen könnte. Deshalb werden die zu untersuchende Texte nach Kapiteln (**Armaselu, 2018**), Paragraphen (**Bailey & Rochester, 2016**), Seiten des Buchs (**Organisciak & Franklin, 2017; Organisciak et al., 2015**) oder Akt/Szene des Dramas (**Schöch, 2017**) segmentiert. Dadurch ergibt sich eine Segmentierung nach vollständigen thematischen Einheiten, während die Längen der Chunks unterschiedlich sind. In **Hettinger et al. (2016)** wird jeder Roman in zehn Segmente zerlegt, die gleich lang sind. In **Blevins (2011)** dagegen wird auf eine Segmentierung verzichtet, jeder Tagebucheintrag wird hier direkt als ein Dokument betrachtet.

2.4 Training des Topic-Modells

Da das Training eines Topic-Modells durch mehrere Faktoren beeinflusst werden kann, wird in diesem Unterkapitel analysiert, wie in den bisherigen Anwendungen mit folgenden Faktoren umgegangen wurde:

- Anzahl der Topics: Die Anzahl der Topics beim Training des Modells muss selbständig eingestellt werden. Dies bildet eine große Herausforderung, vor allem, wenn keine ausreichenden Kenntnisse über den zu untersuchenden Korpus vorliegen. Eine zu hohe Topic-Anzahl könnten dazu führen, dass die Topics nicht genug semantisch verwandte Wörter enthalten, um sinnvolle bzw. interpretierbare Themen zu bilden. Umgekehrt könnte eine zu niedrige Anzahl dazu führen, dass einige Topics zu allgemein sind und sie im gesamten Korpus vorkommen (Jockers, 2013), während einige Topics mehrere unterschiedliche Themen umfassen.
- Hyperparameter der Topic-Dokument-Verteilung und Topic-Wort-Verteilung: Größere Werte der Hyperparameter ermöglichen eine einheitlichere Topic-Dokument- und

Topic-Wort-Verteilung. Im Gegensatz dazu sorgen kleinere Werte bei beiden Verteilungen für mehr Sparsity.[24]

- Verwendung der Hyperparameter-Optimierung: [25] Wird die Hyperparameter-Optimierung beim Training des Topic-Modells verwendet, wird dieses mit einer asymmetrische A-priori-Verteilung der Topic-Dokument-Verteilung trainiert. Dadurch könnte das trainierte Modell bessere Leistungen erbringen (Wallach et al., 2009).
- Inferenzverfahren der Modellierung: Wie in Kapitel 2.2.1 vorgestellt, existieren mehrere Inferenzalgorithmen, um Topic-Modelle zu trainieren Das Gibbs-Sampling etwa ist im Vergleich zu Online VB die genauere Anpassungsmethode.
- Anzahl der Iterationen beim Modell-Updating: Eine größere Anzahl an Iterationen nimmt zwar mehr Zeit in Anspruch, führt aber zu einer besseren Qualität des Topic-Modells. Hier gilt es, einen guten Kompromiss zwischen Zeit und Qualität zu finden.[26]

In 30 der untersuchten 60 Anwendungsbeispielen werden Informationen in Bezug auf das Training des Topic-Modells vorgestellt, in 23 wird über das eingesetzte Topic-Modeling-Tool berichtet und 19 davon haben MALLET (McCallum, 2002) verwendet. In 27 Fällen wird über die Einstellung der Anzahl der Topics berichtet. In den folgenden zehn Arbeiten wird erwähnt, dass mehrere Topic-Modelle mit unterschiedlichen Einstellungen der Topic-Anzahl auf den zu untersuchenden Korpus trainiert wurden, um das beste Modell für weitere Analysen zu finden: **Lee (2018), Eder et al. (2018), Norén & Mähler (2017), Garcia & Pacios (2017), Schöch (2017), Hettinger et al. (2016), Jockers (2016), Bailey & Rochester (2016), Fankhauser et al. (2016), Roe et al. (2014)**. Es ist allerdings schwierig, aus diesen Daten eine Regel abzuleiten, die die Festlegung der Anzahl der Topics betrifft. Bei **Hettinger et al. (2016)** werden 100 bis 500 Topics auf eine Sammlung von 333 Romanen trainiert. Das Korpus in **Jautze et al. (2016)** hat eine ähnliche Größe (410 Romane), das trainierte Topic-Modell enthält allerdings lediglich 50 Topics. In **Hammond & Brooke (2018)** und **Jockers (2012)** werden Topic-Modelle mit jeweils 400 und 500 Topics trainiert. Zu beachten ist aber, dass die Korpora in **Hammond & Brooke (2018)** und **Jockers (2012)** mehr als zehnmal so groß sind wie der in **Hettinger et al. (2016)**: sie umfassen eine Sammlung von mehr als 4000 Science-Fiction-Erzählungen bzw. 3592 Romane. In **Fischer et al. (2018)** enthält das zu untersuchende Korpus 9779

[24] Vgl. https://stackoverflow.com/questions/45162186/mallet-topic-modeling-topic-keys-output-parameter, (08.04.2019).
[25] Vgl. http://mallet.cs.umass.edu/topics.php, (08.04.2019).
[26] Vgl. http://mallet.cs.umass.edu/topics.php, (08.04.2019).

wissenschaftliche Artikel (32 Millionen Tokens), während das Topic-Modell nur mit 24 Topics trainiert wird.

Neben der Anzahl der Topics werden in den meisten Papers nur wenige Informationen darüber geliefert, wie die anderen Faktoren beim Training des Topic-Modells behandelt werden. In **Klein et al. (2015)** und **Eisenstein et al. (2014)** wird berichtet, dass die vorgegebene Parametereinstellung von MALLET übernommen wird, in **Falk (2016)** und **Goldstone (2014)** wird die Hyperparameter-Optimierung von MALLET verwendet. Eine detaillierte Analyse über das Optimierungsintervall der Hyperparameter wird in **Schöch (2017)** durchgeführt, um das beste Modell für weitere Analysen zu finden. In **Organisciak & Franklin (2017)** wird eine asymmetrischen A-priori-Verteilung der Topic-Dokument-Verteilung verwendet, um die allgemeinen Topics und die speziellen Topics beim Training zu unterscheiden. Die allgemeinen Topics tauchen in allen Dokumenten mit hoher Wahrscheinlichkeit auf, während die speziellen Topics sich nur auf einige Dokumente konzentrieren. Es wird außerdem erläutert: „When training topic models, earlier texts have an outsize influence on the topics that emerge. " Um das Problem zu lösen, wird hier die Methode des „weighted training" eingesetzt. In **Maryl & Eder (2017), Schöch (2017), Schöch (2015)** wird darüber hinaus über die Einstellung der Iterationen berichtet (jeweils 1000, 10.000 und 6000). Die Standardeinstellungen der Iterationen in den Topic-Modeling-Tools MALLET, Python-LDA, R Package „topicmodels" und Gensim sind jeweils 1000[27], 2000[28], 2000[29] und 50[30]. Dabei ist jedoch zu beachten, dass die Einstellung in Gensim deutlich kleiner als die der anderen drei Programme ist, was sehr wahrscheinlich auf die Verwendung eines anderen Inferenzalgorithmus zurückzuführen ist (siehe Tabelle 2.1). Außerdem werden in zwei Arbeiten statt LDA andere Topic-Modelle verwendet. In **Nelson et al. (2012)** beispielsweise wird die Multilingual Supervised Latent Dirichlet Allocation (MLSLDA, (Boyd-Graber & Resnik, 2010)) eingesetzt, was nach Ansicht der Verfasser zu kohärenteren Topics führt. In **Smeets et al. (2016)** kommt dagegen die Non-Negative Matrix Factorization (NMF) zum Einsatz. Laut Choo et al. (2013) kann NMF für stabilere Ergebnisse sorgen.

2.5 Post-processing

[27] https://github.com/mimno/Mallet/blob/master/src/cc/mallet/topics/tui/TopicTrainer.java, (10.04.2019).
[28] https://github.com/ariddell/lda/blob/develop/lda/lda.py, (10.04.2019).
[29] https://cran.r-project.org/web/packages/topicmodels/topicmodels.pdf, S. 14, (10.04.2019).
[30] https://github.com/RaRe-Technologies/gensim/blob/develop/gensim/models/ldamodel.py, (10.04.2019).

Das Post-processing bezieht sich auf die Verarbeitung nach dem Training der Topic-Modelle. So wird zum Beispiel in **Klein et al. (2015)** eine interaktive Visualisierung des Topic-Modells vorgenommen, wogegen in **Binder & Jennings (2014)** die Topic-Wörter im Text für das Close Reading annotiert werden, um ihren Kontext im Text besser zu verstehen. Das Post-processing hat normalerweise zwei Aufgaben. Eine ist die Evaluation des trainierten Topic-Modells. Sowohl die Qualität des Modells als auch die der Topics werden geprüft. Genauer gesagt wird kontrolliert, wie gut das Modell die Textdaten anpasst und in welchem Maß die Topics für Menschen als Themen interpretierbar sind. Die andere Aufgabe ist die Visualisierung des Topic-Modells, was ein wichtiges Hilfsmittel der Exploration ist. Die Topic-Dokument- und die Topic-Wort-Verteilung eines Topic-Modells bilden in der Regel zwei sehr große, nicht durchschaubare Tabellen/Matrizen einer extrem großen Datenmenge. Die Visualisierung kann hier ein Topic-Modell anschaulich und verständlich darstellen. Softwarepakete wie z. B. LDAvis (Sievert & Shirley, 2014) in R[31] oder in Python[32], Topic Words in Context (TWiC)[33] oder dfr-browser [34] können für die Visualisierung der Topics und der Topic-Dokument-Verteilung eingesetzt werden. Neben Evaluation und Visualisierung wird auch ein Clustering der Topics oder der Dokumente verwendet, um die Assoziationen zwischen den Topics oder den Dokumenten zu analysieren.

Bei den gesammelten 60 Papers werden in 38 die eingesetzten Methoden des Post-processing vorgestellt. In 30 Arbeiten werden Topic-Modelle visualisiert, während die Evaluation des Modells in geringerem Umfang erwähnt wird. Diese Evaluation wird normalerweise durch eine externe Aufgabe/Ressource oder manuell durchgeführt. In **Eder et al. (2018), Schöch (2017), Hettinger et al. (2016), Mimno et al. (2014)** wird Textklassifikation als Evaluationsmethode verwendet. Ein besseres Klassifikationsergebnis steht hier für ein besseres Modell. In **Fankhauser et al. (2016)** wird die Stabilität der Topics als das Evaluationskriterium vorgestellt, um die richtige Anzahl der Topics herauszufinden. Diese Methode wurde erstmals in (Steyvers & Griffiths, 2007) vorgeschlagen, wobei das Topic Modeling mit verschiedenen Random-Seeds auf einem Untersuchungskorpus wiederholt durchgeführt wird. Man erhält die beste Parametereinstellung und das beste Modell, wenn die trainierten Topics am stabilsten bleiben können. In **Lee (2018), Garcia-Zorita & Pacios (2017), Norén & Mähler (2017), Jautze et al. (2016), Schöch (2015), Goldstone (2014), Blevins (2011)** wird berichtet, dass die Topics

[31] https://github.com/cpsievert/LDAvis, (20.02.2019).
[32] https://github.com/bmabey/pyLDAvis, (20.02.2019).
[33] https://github.com/jarmoza/twic, (20.02.2019).
[34] http://agoldst.github.io/dfr-browser/, (20.02.2019).

manuell überprüft oder als konkrete Themen interpretiert werden. Eine automatische Evaluation der Topics findet sich lediglich in **Rhody (2014)**, wo durch die Berechnung der Topic-Kohärenz ihre Qualität evaluiert wird. In **Wevers et al. (2018)** ist das zu untersuchende Korpus eine Sammlung von Werbungen, die nicht nur Texte, sondern auch Bilder enthalten. Die trainierten Topics und die Inhalte der Bilder werden manuell verglichen, um die Qualität der Topics zu evaluieren.

In einem Topic-Modell können ein Wort und ein Topic jeweils in mehreren Topics und in mehreren Dokumenten vorkommen. Aus diesem Grund ist die Netzwerkvisualisierung eine beliebte Methode für die bildliche Darstellung des jeweiligen Modells. In **Lee (2018)** werden Wörter und ihre Wahrscheinlichkeiten in jedem Topic als Knoten und Kanten eines Netzwerks visualisiert. Dadurch werden nicht nur die Topics, sondern auch die Überschneidungen zwischen den Topics dargestellt (Abbildung 2.7a). In **Fischer et al. (2018), Schöch (2017), Fankhauser et al. (2016), Bartz et al. (2014), Jockers (2012)** wird das Topic- oder Dokument-Clustering eingesetzt, um die Assoziationen zwischen Topics oder Dokumenten zu explorieren. Basierend auf der Topic-Dokument-Verteilung werden die Distanzen (z. B. die Jensen-Shannon-Divergence) zwischen den Topics oder zwischen den Dokumenten für das Clustering berechnet. In **Jockers (2012)** wiederum wird die Netzwerkvisualisierung für die Darstellung der Dokument-Cluster verwendet. Die Dokumente und die Distanzen werden jeweils als Knoten und Kanten visualisiert (Abbildung 2.7b und Abbildung 2.7c).

Für die Visualisierung des Topic-Modells spielen die Metadaten der Dokumente eine wichtige Rolle. Die Verknüpfung zwischen Topic-Dokument-Verteilung und Metadaten ermöglicht eine vertiefende explorative Analyse der Dokumente. Die Knoten (Romane) in Abbildung 2.7c können beispielsweise nach Gattung farbig visualisiert werden, um die potenziellen Gattungsgruppen hervorzuheben. Außerdem werden die Korrelationen zwischen Topics und Epoche der Dokumente häufig visualisiert, um die zeitliche Entwicklung der Topics darzustellen, z. B. in **Maryl & Eder (2017), Smeets et al. (2016), Goldstone (2014), Nelson et al. (2012)**. Ein anderer interessanter Ansatz findet sich in **Broadwell & Tangherlini (2015)**, wo die Topics anhand geographischer Informationen auf Landkarten für die weitere Analyse visualisiert werden.

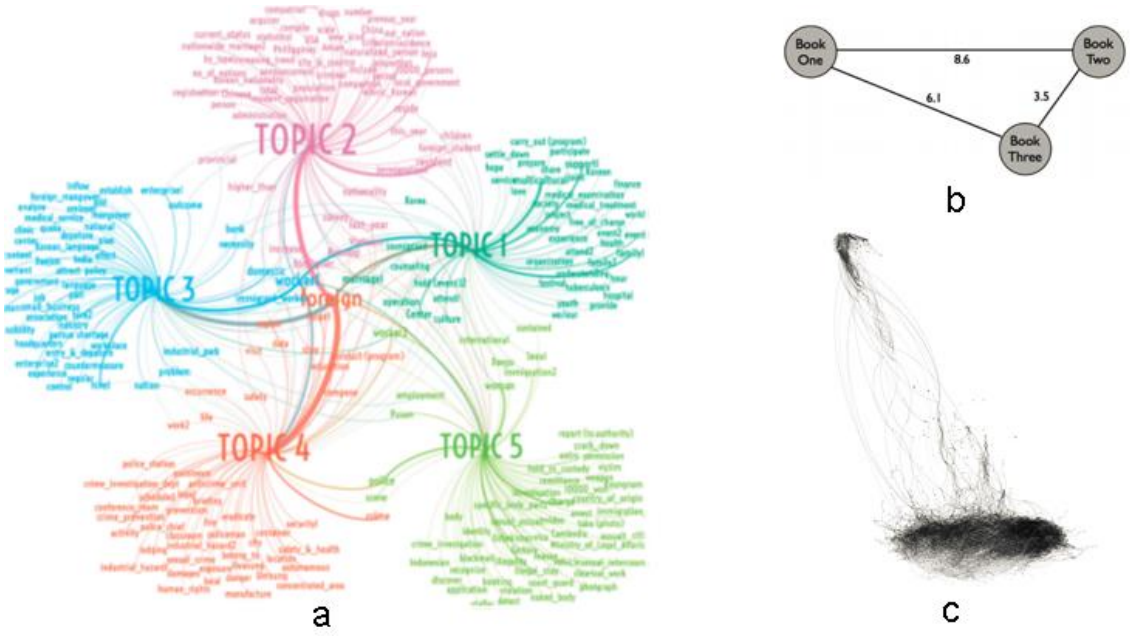

Abbildung 2.7 a): Netzwerkvisualisierung der Topic-Wort-Verteilung (**Lee, 2018**),
b) ein einfaches Netzwerk von drei Büchern (Jockers, 2013),
c) ein Netzwerk von 3592 englischsprachigen Romanen aus dem 19. Jh. (**Jockers, 2012**).

2.6 Zwischenfazit

Topic Modeling kann wertvolle, manchmal sogar unerwartete Ergebnisse produzieren. Für quantitative Textanalysen, vor allem, wenn eine große Textsammlung inhaltlich analysiert werden soll, ist das Topic Modeling eine geeignete, wichtige Methode. Die Exploration der 60 Anwendungsbeispiele des Topic Modeling stellt zusammenfassend dar, wie diese Methode für geisteswissenschaftlichen Forschungen bisher eingesetzt wird.

Betrachtet man die 60 ausgewählten Anwendungen näher lässt sich feststellen, dass die technischen Aspekte des Topic Modeling weniger berücksichtigt werden, obwohl Topic Modeling durch unterschiedliche technische Faktoren wie z.B. die Anzahl der Topics oder die Hyperparameter des Modells beeinflusst werden. Genauer gesagt, viele Parameter werden zwar variiert, meist aber ohne Evaluation, sodass die Praxis eher intuitiv als empirisch gesichert ist. Das liegt daran, dass es im Bereich der Digital Humanities kein gemeinsames Verständnis darüber gibt, wie man mit diesen Faktoren umzugehen soll. Zum Beispiel wird in Nichols et al. (2018) in Bezug auf die Segmentierung der Dokumente erklärt: „Following common practice using LDA on texts, we did not chunk or split the texts in our corpus for analysis.“ Im Gegensatz

dazu wird das Untersuchungskorpus bei Jockers (2013) in 1000-Substantive-Chunks zerlegt: „Novels tend to have some themes that run throughout and others that appear at specific points and then disappear. In order to capture these transient themes, it was useful to divide the novels into 'chunks'. " Ein weiteres Beispiel ist die Diskussion über die Entfernung von Stoppwörtern. Während die meisten Wissenschaftler vor dem Prozess des Topic Modeling die Stoppwörter entfernen, wird in Schofield et al. (2017) gezeigt, dass es ausreichend sein kann, nur die extrem häufigen Stoppwörter vor dem Training des Modells zu eliminieren. Der Grund dafür ist die Tatsache, dass nur diese extrem häufigen Stoppwörter (wie z. B. Funktionswörter) einen großen Einfluss auf die trainierten Topics haben. Im Vergleich dazu ist der Einfluss weniger häufiger Stoppwörter deutlich geringer. Es scheint deshalb nicht notwendig, viel Zeit zu investieren und vor dem Training des Topic-Modells die Stoppwortliste zu verfeinern, um dann alle Stoppwörter aus dem Untersuchungskorpus zu entfernen. Werden nach dem Training des Modells bestimmte Stoppwörter in Topics beobachtet, kann man diese dort eliminieren. Dadurch erhält man annähernd die gleichen Topics, als wäre das Modell komplett ohne diese Wörter trainiert worden.

Darüber hinaus beruhen die meisten Studien zu Topic Modeling auf englischen (Sach-)Texten (Details sieht Kapitel 3). Es ist jedoch wichtig, die Anwendung einer digitalen Methode an Texten in verschiedenen Sprachen zu testen (siehe z. B. Eder 2011, Evert 2017). Aus diesen Gründen werden die Beziehungen zwischen den möglichen Faktoren und der Qualität des Topic-Modells sowie die Qualität der Topics in dieser Arbeit durch systematische Untersuchungen auf zwei deutschen Korpora analysiert, um ein besseres Verständnis der Methode insgesamt zu erlangen.

3. Evaluation des Topic Modeling

In diesem Kapitel wird zusammengefasst, welche Methoden für die Evaluation des Topic Modeling bereits vorgeschlagen wurden. Topic Modeling ist ein unüberwachtes maschinelles Lernverfahren. Deshalb unterscheidet sich die Evaluation von derjenigen überwachter Lernverfahren. Der Vorteil eines überwachten Modells ist, dass die von Menschen annotierten Daten als Gold-Standard für die Evaluation eingesetzt werden können. Ein Teil der annotierten Daten wird als Testdaten zurückgehalten, ein überwachtes Modell wird auf den restlichen Daten trainiert. Danach kann das trainierte Modell auf den Testdaten eingesetzt werden. Wenn das Modell die Labels der Testdaten zu einem hohen Anteil richtig vorhersagen kann, ist das Modell ein gutes Modell. Im Vergleich dazu wird ein Topic-Modell nicht darauf trainiert, bestimmte Topics oder Themen vorherzusagen. Es gibt auch keine Topics als Gold-Standard, die für die Evaluation des Modells direkt verwendet werden könnten. Wenn ein Topic-Modell evaluiert werden soll, kann man das typische Verfahren von Computerlinguistik folgen. In Wallach et al. (2009) wird vorgestellt: „LDA is typically evaluated by either measuring performance on some secondary task, such as document classification or information retrieval, or by estimating the probability of unseen held-out documents given some training documents. A better model will give rise to a higher probability of held-out documents, on average.“ Diese Evaluation allein ist jedoch nicht ausreichend, weil sie weder die Qualität noch die Interpretierbarkeit der Topics widerspiegelt, die bei explorativen Textanalysen in den Digital Humanities eine sehr wichtige Rolle spielen. Die Evaluation des Topic Modeling bezieht sich deshalb nicht nur auf die Evaluation des LDA-Modells, sondern auch auf die Evaluation der Topics. Für die Evaluation der Topics sind andere Methoden notwendig.

3.1 Evaluation des Topic-Modells

3.1.1 Definition der Aufgabe

Das Ziel der Evaluation des Topic-Modells ist es herauszufinden, ob ein Topic-Modell und die jeweiligen Textdaten gut zusammenpassen oder auch, wie gut ein Topic-Modell die Textdaten repräsentiert. Es gibt sowohl interne als auch externe Evaluationsmethode. Die interne Evaluationsmaße sind unabhängig von externen Aufgaben und werden oft für eine schnellere Überprüfung der Modelle eingesetzt (Martin & Jurafsky, 2009). Die externe Evaluation ist in

der Praxis aufwendiger, aber sie geben einen direkten Hinweis auf die Leistung des Modells bei der entsprechenden externen Aufgabe.

3.1.2 Interne Evaluationsmethoden

3.1.2.1 Perplexität

Existieren zwei oder mehrere Topic-Modelle einer Textsammlung wird diese, wenn die Qualität der Modelle durch eine interne Evaluation verglichen werden soll, in Trainingsdaten und Testdaten zerlegt. Testdaten werden gelegentlich auch als „held out"-Dokumente bezeichnet, weil sie „hold out from the training data" sind (Martin & Jurafsky, 2009). Anschließend werden Parameter der Modelle auf Trainingsdaten trainiert und es wird überprüft, wie gut die Modelle mit den Testdaten zusammenpassen. Das Modell, das den Testdaten eine höhere Wahrscheinlichkeit zuweisen kann, gilt hier als das bessere: „Given two probabilistic models, the better model is the one that has a tighter fit to the test data or that better predicts the details of the test data, and hence will assign a higher probability to the test data." (Martin & Jurafsky, 2009)

In Wallach et al. (2009) wird eine interne Methode vorgestellt, mit der sich Topic-Modelle durch die Einschätzung der Wahrscheinlichkeit von Held-out-Dokumenten evaluieren lassen. In dem Artikel werden unterschiedliche maschinelle Lernmethoden vorgestellt, mit denen sich die Wahrscheinlichkeit der Held-out-Dokumente einschätzen lässt: „Importance sampling methods", „Harmonic mean method", „Annealed importance sampling", „Chib-style estimation" und „Left-to-right evaluation algorithm." Bessere Modelle erzielen hier höhere Wahrscheinlichkeiten bei den Held-out-Dokumenten. In der Praxis wird zumeist die Perplexität als das Evaluationsmaß anstelle der Wahrscheinlichkeit verwendet. „The perplexity (sometimes called PP for short) of a language model on a test set is the inverse probability of the test set, normalized by the number of words." (Martin & Jurafsky, 2009). Die Leistungsfähigkeit eines Sprachmodells bei externen Aufgaben wie z. B. der Spracherkennung oder der maschinellen Übersetzung korreliert oft mit der Perplexität. Je niedriger der Wert der Perplexität ist, desto leistungsfähiger ist das Modell. In Abbildung 3.1 wird ein Vergleich einiger Sprachmodelle dargestellt. Diese werden auf einen Datensatz des TREC AP-Korpus[35] trainiert, der aus 16.333 englischen Nachrichtenartikeln besteht. Es ist zu beobachten, dass LDA wesentlich

[35] http://www.daviddlewis.com/resources/testcollections/trecap/, (13.05.2019).

leistungsfähiger als andere Modelle ist. Außerdem lässt sich feststellen, dass sich die Perplexität aller Modelle mit der Erhöhung der Topic-Anzahl reduziert (Murphy, 2012). Vereinfacht ausgedrückt gilt hier: Modelle mit mehr Topics sind besser und leistungsfähiger.

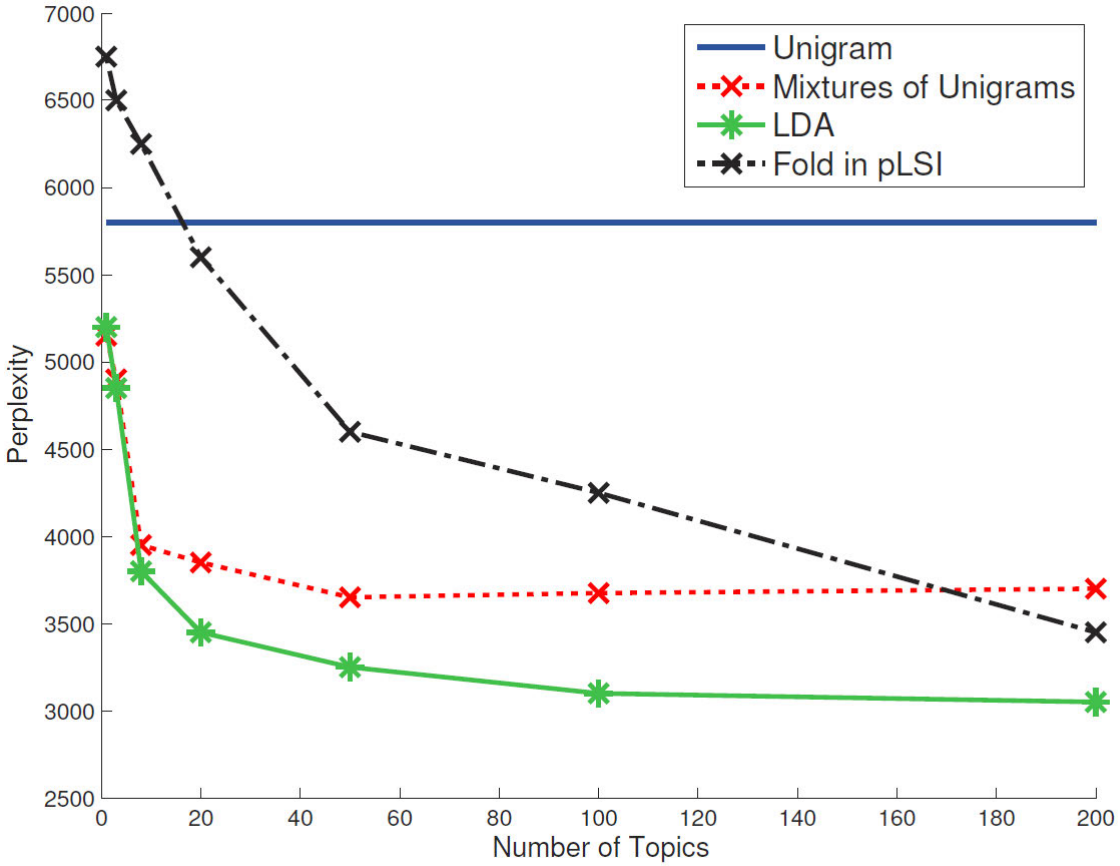

Abbildung 3.1 Perplexität vs. Topics-Anzahl (Murphy, 2012)

Der Vergleich der Perplexität zwischen zwei Modellen ist aber nur sinnvoll, wenn diese das gleiche Vokabular enthalten. Außerdem ist in der Praxis darauf zu achten, dass die Korrelation zwischen Perplexität und Leistungsfähigkeit des Modells keine Kausalität darstellt. Aus diesem Grund garantiert eine interne Verbesserung der Perplexität nicht unbedingt, dass das Modell bei den externen Aufgaben auf jeden Fall besser funktioniert. Eine End-to-End Evaluation ist deshalb immer erforderlich (Martin & Jurafsky, 2009).

3.1.2.2 Berechnung der Perplexität mit MALLET

In diesem Unterkapitel wird kurz vorgestellt, wie die Berechnung der Perplexität und die interne Evaluation der Topic-Modelle praktisch durchgeführt werden kann. [36] Es existieren verschiedene Tools, die die Perplexität eines Topic-Modells berechnen können, wie z. B. das R-Paket „topicmodels“.[37] Für die Demonstration wird hier MALLET verwendet, weil es ein weit verbreitetes Topic-Modeling-Tool ist.

Zunächst wird der gesamte Korpus importiert und in eine MALLET-Datei, **topic_model.mallet**, transformiert[38]. Anschließend wird der vorliegende Datensatz durch folgende Befehle in zwei Dateien aufgeteilt, nämlich Trainingsdaten (**training.mallet**) und Testdaten (**test.mallet**). „**--training-portion .8**“ bedeutet, dass das Verhältnis zwischen Trainings- und Testdaten 80 % zu 20 % beträgt.

```
bin/mallet split --input topic_model.mallet --training-file training.mallet --testing-file testing.mallet --training-portion .8
```

Der nächste Schritt besteht darin, ein Modell auf die Trainingsdaten zu trainieren. Für die Evaluation wird der Output durch „**-evaluator-filename = evaluator.mallet**“ gespeichert.

```
bin/mallet train-topics --input training.mallet --num-topics 10 --evaluator-filename = evaluator.mallet
```

Der dritte Schritt ist dann das Modell durch die Testdaten zu überprüfen. Die Output-Datei „**doc_probs.csv**“ wird durch „**--output-doc-probs**“ erstellt, die die Log-Wahrscheinlichkeit aller einzelnen Dokumenten in den Testdaten enthält. In der Output-Datei werden die Werte aufgelistet, jede Zeile enthält den Wert eines Dokuments.

```
bin/mallet evaluate-topics --evaluator evaluator.mallet --input testing.mallet --output-doc-probs doc_probs.csv
```

[36] Vgl. *Model Perplexity*, unter: https://mallet-dev.cs.umass.narkive.com/UQAYvpn2/model-perplexity, (16.05.2019).
[37] https://cran.r-project.org/web/packages/topicmodels/topicmodels.pdf, (16.05.2019).
[38] Hier wird nicht näher darauf eingegangen, wie MALLET verwendet werden kann. Ein detailliertes Tutorial über Topic Modeling mit MALLET von Shawn Graham, Scott Weingart und Ian Milligan befindet sich unter: https://programminghistorian.org/en/lessons/topic-modeling-and-mallet, (16.05.2019).

Abschließend wird nun die Perplexität berechnet. Dafür wird die Log-Wahrscheinlichkeit jedes Dokumentes durch die Dokumentlänge geteilt, das Ergebnis ist dann die Perplexität. Durch folgenden Befehl lässt sich die Dokumentlänge aller Dokumente in den Testdaten erhalten:

```
bin/mallet run cc.mallet.util.DocumentLengths --input testing.mallet
```

3.1.2.3 Weitere interne Evaluationsmethoden

Neben der Perplexität werden vier weitere interne Evaluationsmethoden in Deveaud et al. (2014); Arun et al. (2010); Cao et al. (2009) und Griffiths & Steyvers (2004) vorgestellt. Diese konzentrieren sich hauptsächlich auf die Korrelation zwischen den Evaluationsmaßen und der Anzahl der Topics. Das Ziel ist es, die richtige Anzahl der Topics durch das jeweilige Extrem (Maximum oder Minimum) zu ermitteln. Zum Beispiel berechnet die Methode in Griffiths & Steyvers (2004) die Wahrscheinlichkeit der Testdaten bei einem gegebenem Topic-Modell. Je größer die Wahrscheinlichkeit ist, desto besser ist das Modell. Die bei Arun et al. (2010) und Cao et al. (2009) verwendeten Methoden suchen durch die Verwendung zweier Minimierungsprobleme nach der optimalen Anzahl der Topics.

Alle vier Methoden sind in dem R-Paket „ldatuning“[39] implementiert. Murzintcev (2019) hat ein Test auf dem Datensatz „AssociatedPress“ durchgeführt. Dieser Datensatz stammt aus der ersten Text Retrieval Conference (TREC-1) 1992 und enthält 2246 Dokumente und 10.473 Terms (Hornik & Grün, 2011). Das Ergebnis wird in Abbildung 3.2 dargestellt. Drei Methoden zeigen im Ergebnis übereinstimmend, dass die optimale Anzahl der Topics zwischen 90 und 140 liegt, während die Methode aus Deveaud et al. (2014) lediglich eine abnehmende Kurve bei der Erhöhung der Topic-Anzahl zeigt.[40]

[39] https://cran.r-project.org/web/packages/ldatuning/index.html, (16.05.2019).
[40] Das Testergebnis wird in einem anderen Blog-post bestätigt: http://freerangestats.info/blog/2017/01/05/topic-model-cv, (16.05.2019).

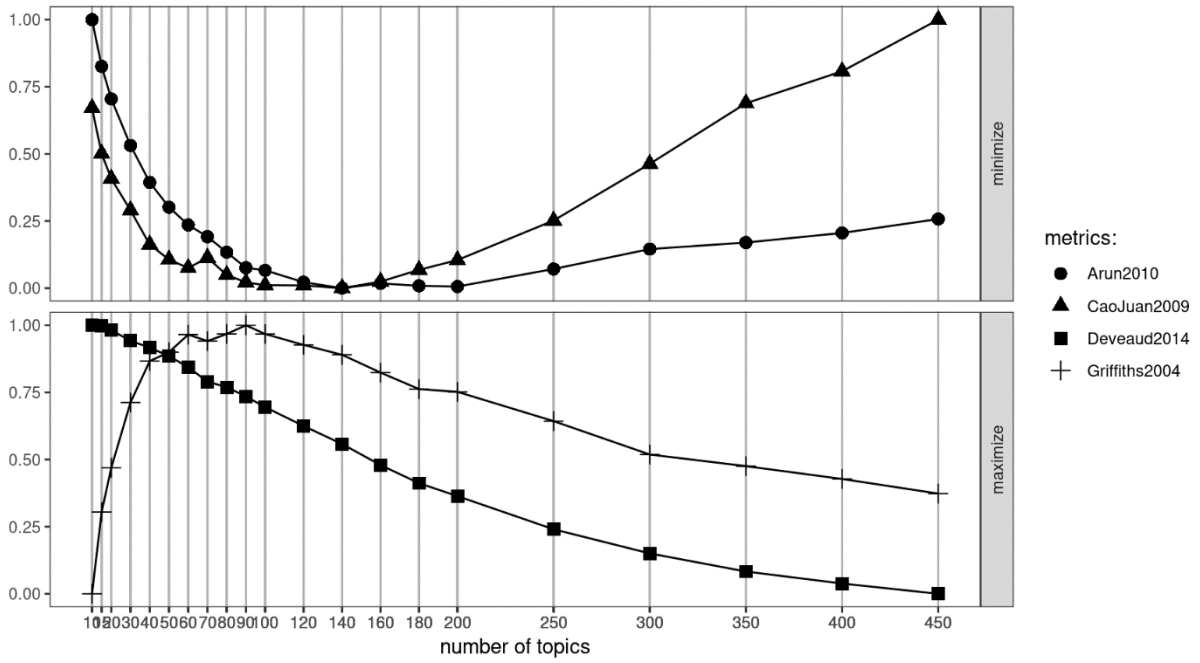

Abbildung 3.2 Interne Evaluation für die Auswahl der richtigen Topics-Anzahl (Murzintcev, 2019)

3.1.3 Externe Evaluationsmethoden

Externe Methoden der Evaluation beziehen sich auf die Anwendung des Modells in sekundären Aufgaben. Eine gute Leistung in den sekundären Aufgaben drückt hier die Qualität des Topic-Modells aus (Wallach et al., 2009). Eine typische externe Aufgabe ist beispielsweise die Topic-Modeling-basierte Dokumentklassifikation. Bei der einfachsten Bag-of-Words-basierten Dokumentklassifikation wird jeder Text durch einen Vektor von Wörtern repräsentiert. Die Komponenten jedes Vektors bezeichnen entweder, ob jedes Wort in einem Text vorkommt oder, wie häufig jedes Wort in einem Text enthalten ist. Bei der Topic-Modeling-basierten Textklassifikation ist jeder Text ein Vektor von ***N*** Topics und die Komponenten jedes Vektors bezeichnen, wie wahrscheinlich jedes Topic in einem Text vorkommt (Abbildung 3.3). ***N*** ist hier die Anzahl der Topics des Topic-Modells. Anhand der Dokument-Topic-Verteilung können die Dokumente dann klassifiziert werden.

	Topic0	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	dominant_topic
Doc0	0	0	0	0	0	0.14	0	0	0	0.84	9
Doc1	0	0.05	0	0	0.05	0.24	0	0.65	0	0	7
Doc2	0	0	0	0	0	0.08	0.2	0	0	0.71	9
Doc3	0	0.55	0	0	0	0.44	0	0	0	0	1
Doc4	0.16	0.29	0	0	0	0.53	0	0	0	0	5
Doc5	0	0	0.05	0	0	0	0	0.12	0	0.83	9
Doc6	0	0	0	0	0	0.88	0.1	0	0	0	5
Doc7	0	0	0	0	0	0.99	0	0	0	0	5
Doc8	0	0	0.08	0.67	0	0	0	0	0.24	0	3
Doc9	0	0	0.74	0	0	0.14	0	0	0.11	0	2
Doc10	0	0	0	0	0.41	0.16	0	0.06	0	0.36	4
Doc11	0	0	0	0	0	0	0	0.97	0	0	7
Doc12	0	0	0	0.44	0	0.04	0	0.27	0	0.24	3
Doc13	0.14	0	0	0	0	0.07	0.57	0.08	0	0.13	6
Doc14	0	0	0	0	0.78	0.22	0	0	0	0	4

Abbildung 3.3 Beispiel einer Dokument-Topic-Verteilung[41]

Bei Schöch (2017) wird eine beispielhafte Untersuchung vorgestellt, in der Topic-Modelle durch Dokumentklassifikation evaluiert werden. Für die Analyse der Textsammlung werden zunächst mehrere Topic-Modelle trainiert, während zwei Parameter (Anzahl der Topics und Optimize-Interval) beim Training unterschiedlich eingestellt werden. Im nächsten Schritt wird das beste Modell durch das Klassifikationsergebnis identifiziert. Abbildung 3.4 stellt die Verteilung der Klassifikationsergebnisse im Verhältnis zu den Parameter-Einstellungen dar. In dieser Untersuchung lieferte das Modell mit 60 Topics und 300 Optimize-Intervallen das beste Ergebnis. Die Topics des so ermittelten besten Topic-Modells werden dann nach der Evaluation für die Exploration und den Vergleich der Textsammlung verwendet.

[41] https://www.machinelearningplus.com/nlp/topic-modeling-python-sklearn-examples/, (04.08.2021).

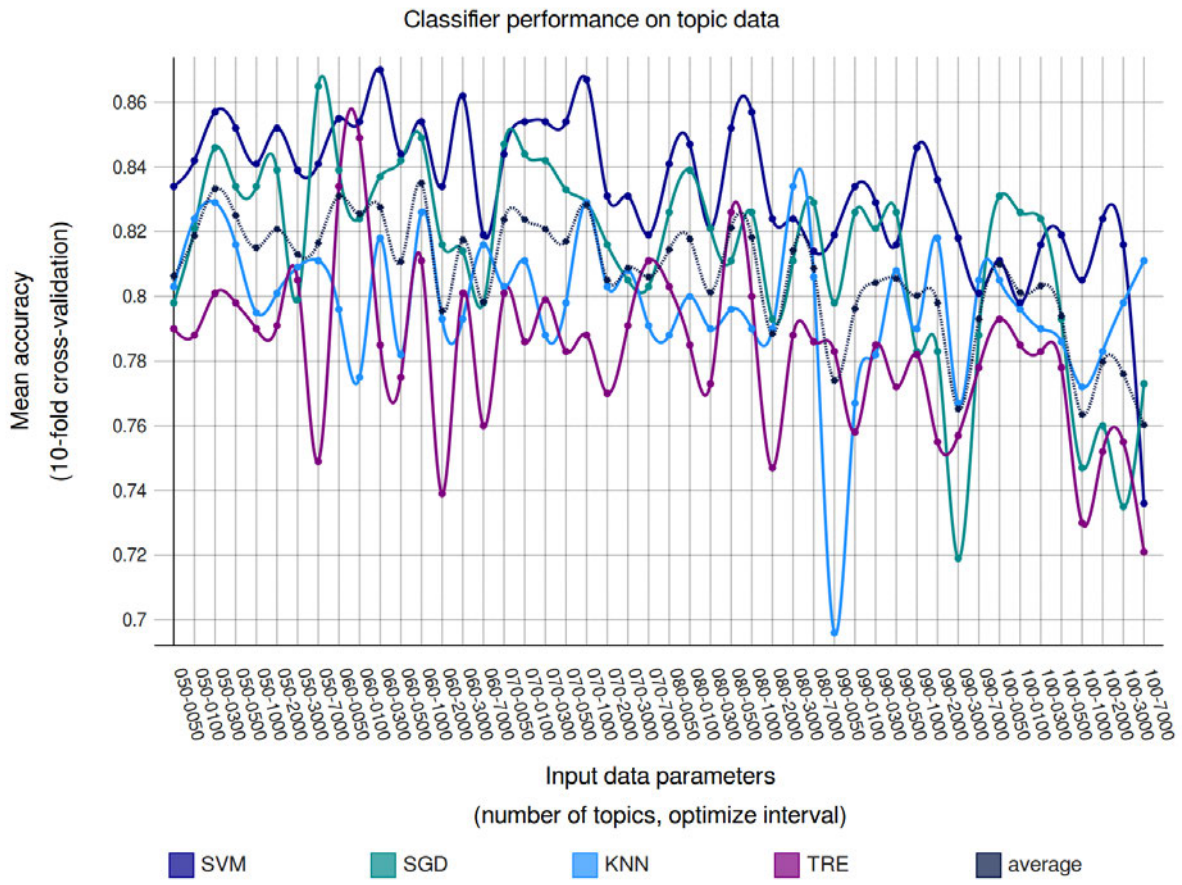

Abbildung 3.4 Ergebnis der Topic-Modell-basierten Klassifikation von Untergattungen (Schöch, 2017)

Eine andere verbreitete externe Aufgabe für die Evaluation von Topic-Modellen ist das Information Retrieval (IR). Es wird zum Beispiel in Liu & Croft (2004) vorgestellt, dass IR-Systeme besser funktionieren, wenn Sprachmodelle für Cluster-basiertes Retrieval eingesetzt werden. Da Topic-Modelle auch als Sprachmodelle betrachtet werden können, wird eine lineare Kombination von Dokument-Modell und LDA-Topic vorgestellt, um das IR-System zu verbessern. Hingewiesen wird dabei darauf, dass eine direkte Repräsentation der Dokumente durch ein LDA-Modell das IR-System nicht verbessern kann. Stattdessen wird ein Dokument-Modell eingesetzt, das auf dem LDA-Modell der Dokumentsammlung basiert. Je besser das Ergebnis des IR-Systems ist, desto besser ist auch das Topic-Modell, weshalb dieses Vorgehen auch für die Evaluation von Topic-Modellen eingesetzt werden kann, wenn es etwa darum geht, die richtigen Parameter-Einstellungen für das Training des Modells zu finden. In Abbildung 3.5 wird dargestellt, wie die durchschnittliche Precision sich ändert, wenn die Anzahl der

Iterationen des Gibbs-Sampling erhöht wird. Das Ergebnis zeigt, dass das IR-System bei dem Test ab 60 Iterationen ungefähr stabil bleiben kann.

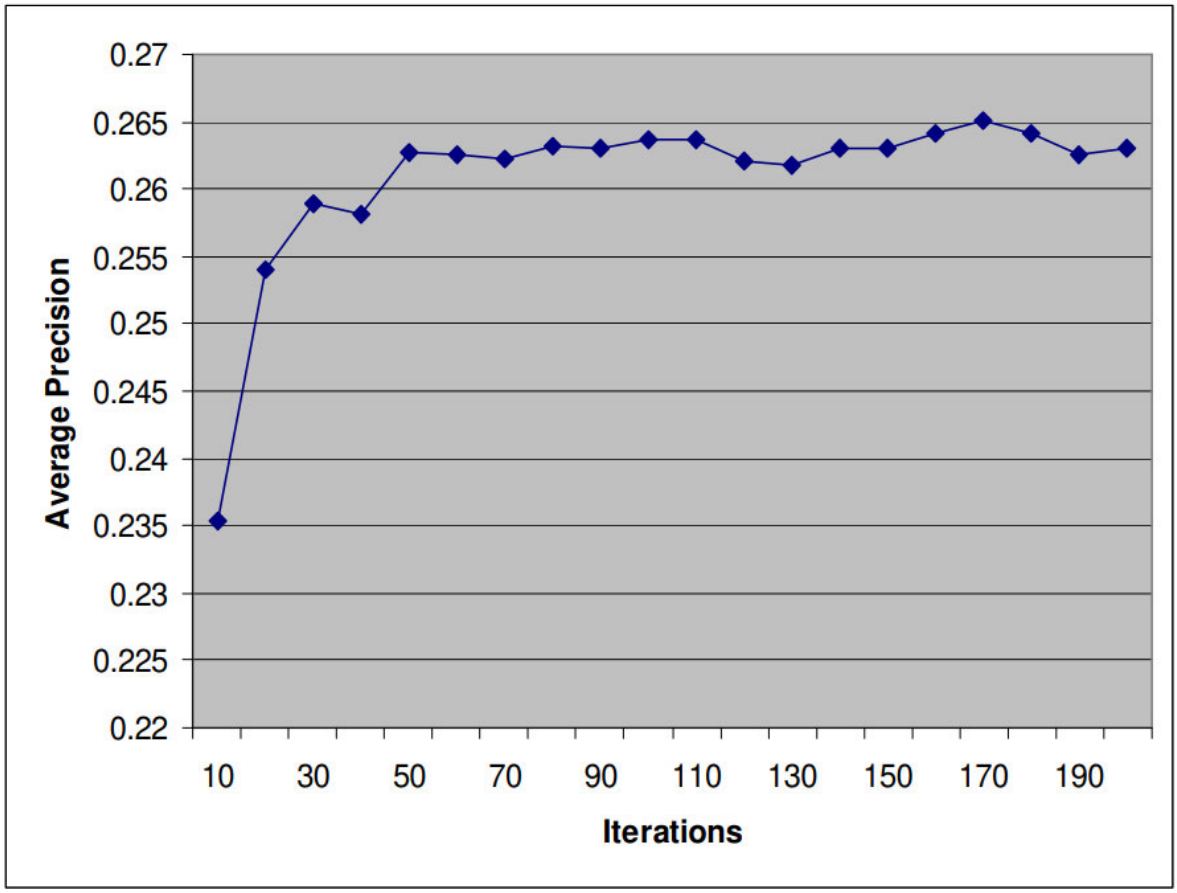

Abbildung 3.5 Das Ergebnis des IR-Systems im Verhältnis zur Anzahl der Iterationen des Gibbs-Sampling (Liu & Croft, 2004)

3.1.4 Zwischenfazit zur Evaluation des Topic-Modells

Aus dem Vergleich der vorhandenen Evaluationsmethode lassen sich die folgenden Schlussfolgerungen ziehen: Die internen Evaluationsmethoden sind zwar einfach, können aber die Leistungsfähigkeit des Topic-Modells bei den externen Aufgaben nicht garantieren. Im Vergleich dazu sind die externen Aufgaben (wie z. B. Textklassifikation) für die Evaluation aufwändiger, aber besser geeignet. Die Topic-Modeling-basierte Dokumentklassifikation wird bereits regelmäßig im Bereich der Digital Humanities verwendet. Diese Methode ist intuitiver als die internen Evaluationsmethoden, weil sich klarer erkennen lässt, welche Fortschritte die Evaluation erreicht und warum sie sinnvoll ist. Außerdem ist die Dokumentklassifikation eine

passende Evaluationsaufgabe, weil Topic Modeling in Digital Humanities häufig verwendet wird, um die thematischen Differenzierungen einer Textgruppe zu explorieren. Topic-Modeling-basierte Textklassifikationsergebnisse werden deshalb bei den Untersuchungen in Kapitel 6 und Kapitel 7 als Indikator für die Qualität der Topic-Modelle verwendet.

3.2 Evaluation der Topics

3.2.1 Definition der Aufgabe

In Chan & Akoglu (2013) betonen die Autoren die Bedeutung der Evaluation der Topics: „Low-quality topics can potentially degrade the performance of the applications; e.g. they could mislead topic-based document similarity, introduce noise in clustering, and cause poor semantic interpretation. This makes the evaluation of topic models a crucial task.“ Allerdings sind die in Kapitel 3.1 vorgestellten Evaluationsmethoden oder Metriken für diese Aufgabe nicht geeignet, weil die Qualität der Topics bei der Evaluation eines Topic-Modells nicht unbedingt überprüfen wird. Ein Beispiel ist die Untersuchung in Wehrheim (2019): „Die optimale Anzahl an Topics wurde zunächst mittels der R-Erweiterung ldatuning von Murzintcev (2016)[42] geschätzt, allerdings lieferte der daraus resultierende Wert von etwa 30 nur sehr unspezifische Topics. Daher wurde die Topic-Anzahl sukzessive erhöht und schließlich ein Wert von 70 festgelegt, welcher einen Kompromiss zwischen Spezifität und Redundanz der Topics darstellt.“ In Chang et al. (2009) beobachten die Autoren sogar, dass eine negative Korrelation zwischen menschlicher Bewertung der Topics und der internen statistischen Evaluation von Topic-Modellen existiert.

In Bezug auf die Qualität der Topics ist eine (automatische) Methode notwendig, um diese zu bewerten. In AlSumait et al. (2009) wird ein „Topic Significance Ranking“ (TSR) vorgestellt, welches die Topics nach ihrer semantischen Signifikanz sortiert. Unter der Qualität des Topics wird zumeist die menschliche Interpretierbarkeit des Topics verstanden, die auch als semantische Interpretierbarkeit bezeichnet wird. Topics, die aus einer Gruppe von meist semantisch kohärenten Worten bestehen, sind in der Regel besser interpretierbar und weisen so eine höhere Qualität auf. Im Folgenden werden die Evaluationsmethoden in AlSumait et al. (2009), Chang et al. (2009), Newman, Lau, et al. (2010), Chan & Akoglu (2013), Lau et al.

[42] Murzintcev, Nikita 2016: ldatuning (R package). Online verfügbar unter https://cran.r-project.org/web/packages/ldatuning/ldatuning.pdf, (08.01.2020).

(2014) und Röder et al. (2015) kurz vorgestellt. Sie konzentrieren sich auf die Messung der semantischen Interpretierbarkeit bzw. die semantische Kohärenz der Topics.

3.2.2 „Junk and Insignificant“ (J/I) -Topics

Der Ausgangspunkt der Methode ist der Versuch, die „Junk and Insignificant“ (J/I) -Topics zu definieren. Nach Steyvers & Griffiths (2007) ist ein „Junk“ -Topic ein „uninterpretable topic that picks out idiosyncratic word combinations“. Ein „Insignificant“ Topic wird in AlSumait et al. (2009) folgendermaßen definiert: „A topic that consists of general words, known as ‚background words‘, which are commonly used in general or across a broad range of documents within each corpus/domain.“ Da ein Topic eine Wahrscheinlichkeitsverteilung / ein Vektor ist, kann die Distanz zwischen einem Testtopic und den J/I-Topics gemessen werden. Je größer die Distanz ist, umso höher ist die Qualität des Testtopics, je geringer sie ist, umso uninteressanter ist das Testtopic. Wird ein Topic-Modell mit nur einem Topic trainiert, stellt dieses die Verteilung aller Wörter im Korpus dar. Das Topic kann als das einfachste J/I-Topic betrachtet und für die Evaluation verwendet werden. Darüber hinaus werden drei weitere Arten von J/I-Topics definiert:

- Uniform Distribution Over Words (W-Uniform): Das W-Uniform ist eine Gleichverteilung auf allen Wortformen des Korpus. Nach dem Zipf’s Law sollen die Wörter im ein interessanten oder spezifischen Topic nicht gleich verteilt sein. Weil alle Wörter (Wortformen) in einem W-Uniform-Topic mit der gleichen Wahrscheinlichkeit zu dem Topic gehört, ist das Topic eher ein J/I-Topic.
- The Vacuous Semantic Distribution (W-Vacuous): Die empirische Verteilung der Stichprobe (der gesamten Wortfrequenzen der Stichprobe) ist eine Konvexkombination der probabilistischen Verteilungen der latenten Themen, die keine signifikanten Informationen zeigt. Im Vergleich dazu ist zu erwarten, dass ein echtes Topic eine einzigartige Eigenschaft hat. Deshalb ist ein Topic semantisch weniger signifikant, wenn der Distanz zwischen dem Topic und der empirischen Verteilung der Stichprobe klein ist.
- The Background Distribution (D-BGround): Ein spezifisches Topic ist normalerweise mit einer kleinen Menge von Dokumenten verbunden. Im Gegensatz dazu ist ein allgemeines Topic mit einer großen Anzahl von Dokumenten verknüpft. Ein allgemeines Topic kann als „background topic“ (D-BGround) für die Evaluation

definiert werden. Wenn die Distanz zwischen dem Testtopic und dem D-BGround-Topic groß ist, ist das Testtopic eher interessant.

Um die Distanz zwischen einem Testtopic und einem J/I-Topic zu messen, sind Distanzmaße erforderlich. Es werden dazu drei Möglichkeiten vorgeschlagen, nämlich die Kullback-Leibler (KL)-Divergenz, die Kosinus-Distanz und die Pearson-Korrelation. Durch die Berechnung der Distanzen erhält jedes Testtopic einen Wert. Anhand dieser Werte werden am Ende alle Topics eines Topic-Modells eingeordnet. Allerdings ist unklar, inwieweit die Ergebnisse dieser Evaluationsmethode mit der menschlichen Bewertung der Topics übereinstimmen (im entsprechenden Paper findet sich keine manuelle Evaluation).

3.2.3 Word Intrusion

Für die Evaluation von Topics wird in Chang et al. (2009) zudem die Methode „Word Intrusion“ vorgestellt, die die Topics durch Befragung manuell evaluiert. Während der Evaluation wird ein zufälliges Wort zu den wichtigsten fünf Topic-Wörtern eines Topics hinzugefügt. Im Prinzip sollte dieses zufällige Wort mit den anderen fünf Wörtern semantisch nicht verwandt sein. Die Aufgabe der Befragten ist es, das Topic zu überprüfen und das „falsche“ Wort zu finden. Wenn ein Topic semantisch kohärent ist, ist die Aufgabe nicht kompliziert. Zum Beispiel dürften die meisten Befragten ohne Problem das Wort „apple“ im Topic {dog, cat, horse, apple, pig, cow} als „falsch“ identifizieren. Bei der Evaluation des Topics {car, teacher, platypus, agile, blue, Zaire} dagegen wählten die meisten Befragten willkürlich ein Wort aus, weil die Topic-Wörter semantisch miteinander nicht kohärent sind. Kann ein Topic nicht als ein Thema oder Konzept von Menschen interpretiert werden, bietet es eine nur geringe Qualität.

3.2.4 Externe Ressourcen-basierte Evaluation

In Newman, Lau, et al. (2010) wird vorgeschlagen, externe Ressourcen für die Evaluation der Topics einzusetzen. Die Grundidee ist, die ontologischen Ähnlichkeiten oder die semantischen Verlinkungen oder die Kookkurrenz-Informationen zwischen Wörtern aus den externen Ressourcen (wie z. B. WordNet[43] oder Wikipedia) für die Evaluation zu extrahieren. Anhand eines gegebenen Topics wird hier die Ähnlichkeit aller seiner Wortpaare berechnet. Die

[43] https://wordnet.princeton.edu/, (15.06.2019).

durchschnittliche Ähnlichkeit aller Wortpaare drückt dann die Qualität dieses Topics aus. Außerdem existieren auch Methoden, die auf der Suchmaschine Google beruhen. Hier werden Topics als eine Evaluationseinheit betrachtet, ihre Qualität wird anhand der Suchergebnisse zu ihnen bei Google bewertet. Es werden insgesamt 15 unterschiedliche Evaluationsmethoden getestet (neun WordNet- und vier Wikipedia-basierte sowie zwei Google-basierte Ansätze). Zugleich werden Datensätze von manuell bewerteten Topics aufgebaut. Die Topics werden nach ihrer manuellen Bewertung und parallel dazu nach der Bewertung durch die oben vorgestellten Methoden sortiert. Die Spearman'sche Rangkorrelationskoeffizient zwischen den Topic-Rankings zeigt, dass die auf Wörter-Kookkurrenz basierenden Kohärenzmaße mit menschlicher Bewertung am besten korrelieren, während die auf WordNet-basierten Methoden für die Evaluation nicht ideal sind.

3.2.5 Evaluation durch Graph Mining

Mit der in Chan & Akoglu (2013) vorgestellten Methode kann die externe Evaluation von Topics darüber hinaus mit der Hilfe von Graph Mining durchgeführt werden. In dieser Untersuchung diente die Verlinkung von Wikipedia-Seiten als externe Ressource, die die Relation der Topic-Wörter beschreibt. Es wird zuerst eine Sammlung von Topics manuell als „gut“ oder „schlecht“ annotiert. Die Struktur der Verlinkungen zwischen den Topic-Wörtern in Wikipedia wird anschließend graphisch modelliert und zwei Arten von Topic-Graphen werden definiert: der Topic Projection Graph (g_M) und der Topic Spanning Graph (g_S). Der Projection Graph ist eine direkte graphische Darstellung der Verlinkungen. Wenn keine Verlinkung zwischen einem Wort und anderen Wörtern existiert, steht das Wort an der Seite des Graphen. Anders als beim Projection Graph wird das Wort in einem Spanning Graph durch „connector nodes“ mit anderen Wörtern verbunden. Abbildung 3.6 stellt zwei Graphen von zwei Topics dar. Topic 1 ist „steam, engine, valve, piston, cylinder, pressure, boiler, air, pump, pipe“, und Topic 2 ist „cut, system, capital, pointed, opening, building, character, round, france, paris“. Wenn ein Topic-Wort einen Wikipedia-Eintrag besitzt, wird dies mit einem blauen Viereck markiert, andernfalls erhält es ein weißes Viereck. Die „connector nodes“ wiederum werden grau dargestellt. Hier ist auf Anhieb zu erkennen, dass, weil die meisten Wörter im Topic 1 miteinander verbunden sind, dieses im Vergleich zu Topic 2 semantisch kohärenter ist. Aus den Graphen werden anschließend Graph-basierte Features, wie z. B. die Anzahl der verbundenen Wörter, konstruiert. Anhand derartiger Features und einer manuellen Annotation wird in dem letzten Schritt ein Modell trainiert, um die Topics zu klassifizieren.

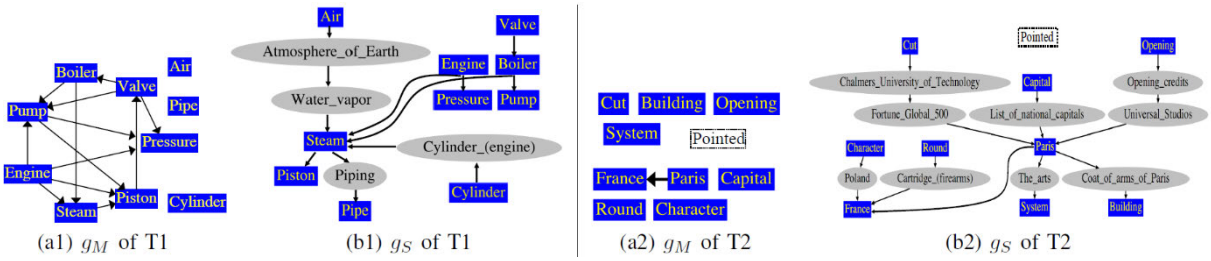

Abbildung 3.6 Projection Graph und Spanning Graph: zwei Beispiele

3.2.6 Deskriptive Metriken des Topics

Neben den oben vorgestellten Evaluationsmethoden werden noch einige deskriptive Metriken vorgeschlagen und in MALLET implementiert, die für die Evaluation von Topics hilfreich sein könnten.[44] Dabei ist zu beachten, dass diese Metriken die Qualität eines Topics nicht unbedingt definieren. Sie werden vielmehr verwendet, um die „interessanten" Topics zu finden oder um die „allgemeinen" von den „spezifischen" Topics zu unterscheiden. Allerdings sind diese Metriken sehr wohl Merkmale der Topics, die sich zum Beispiel für die Klassifikation der Topics als Feature nutzen lassen.

1. tokens: Es wird errechnet, wieviel Tokens einem Topic zugeordnet werden. Wenn das Topic zu viele oder zu wenige Tokens enthält, ist es eher uninteressant.
2. document entropy: Die Frequenz eines Topics über alle Dokumente wird errechnet und normalisiert, um eine Topic-Dokument-Distribution zu erstellen. Die Entropie dieser Distribution wird als „document entropy" bezeichnet. Ein Topic mit niedriger Entropie konzentriert sich auf einige Dokumente, es ist normalerweise ein spezifisches/ interessantes Topic.
3. word length: Davon ausgehend, dass längere Wörter normalerweise spezifische Bedeutungen haben ist anzunehmen, dass Topics mit einer kurzen durchschnittlichen Wortlänge eher nicht spezifisch sind.
4. coherence: Hier wird die Kohärenz der Wörter im Testtopic anhand von UMass-Kohärenzmaß berechnet[45]. Der Wert der Kohärenz ist immer kleiner als 0. Je näher der Wert an 0 liegt, desto eher tauchen die Wörter gemeinsam im Untersuchungskorpus auf.

[44] Die Berechnung dieser Metriken kann durch MELLET automatisch erledigt werden. Genauere Information unter: http://mallet.cs.umass.edu/diagnostics.php, (13.09.2020).
[45] Eine genauere Beschreibung des UMass-Kohärenzmaß befindet sich in Kapitel 3.2.7.

5. uniform_dist: Diese Metrik verwendet dieselbe Methode des W-Uniform in Kapitel 3.2.2. Dabei wird die Kullback-Leibler-Divergence für die Messung der Distanz verwendet; größere Werte weisen auf ein spezifischeres Topic hin.
6. corpus_dist: Die Verteilung aller Wörter im Korpus wird als ein J/I-Topic betrachtet, anschließend wird die Distanz zwischen einem Testtopic und dem J/I-Topic gemessen. Hier weisen größere Werte auf ein spezifischeres Topic hin.
7. effective number of words: Diese Metrik ist der uniform_dist ähnlich. Größere Werte weisen auf ein spezifischeres Topic hin.
8. token/document discrepancy: Diese Metrik sucht nach sogenannten „bursty words“. Diese gehören zwar zu einem Topic, können es aber nicht richtig repräsentieren, weil sie nur in einigen Dokumenten bzw. in bestimmten Kontexten sehr häufig auftauchen.
9. rank 1 documents: Wörter wie z. B. „paper, abstract, data“ sind in wissenschaftlichen Texten als „allgemein“ anzusehen, im Vergleich dazu sind Begriffe wie z. B. „cell, disease, dna, blood“ spezifischer. Ein spezifisches Topic wird in relativ wenigen Dokumenten vorkommen. Aber wenn es in diesen Dokumenten vorkommen, ist es oft das wichtigste Topic zu diesen Dokumenten. Daher werden diese Dokumente als „rank 1 documents“ des Topics bezeichnet. Mit dieser Metrik wird berechnet, wie oft ein Topic das häufigste Topic in einem Dokument ist. Das Ziel der Metrik ist es, die allgemeinen und die spezifischen Topics zu unterscheiden.
10. allocation count: Hier wird berechnet, wieviel Prozent der Wörter eines Dokuments einem Topic zugeordnet wurden.
11. allocation ratio: Diese Metrik gibt das Verhältnis der Zuteilungszahlen bei zwei verschiedenen Schwellenwerten an, die voreingestellt bei 50% und 2% liegen.
12. exclusivity: An dieser Stelle erfolgt eine Überprüfung, wie oft Top-Wörter eines Topics in den anderen Topics auftauchen.

3.2.7 Topic-Kohärenzmaße

Ein Topic-Kohärenzmaß wird für die Quantifizierung und Evaluation der Interpretierbarkeit von Topics verwendet. Der Grundgedanke hinter diesem Ansatz besteht darin, ein externes Referenzkorpus zu verwenden und die Häufigkeit des gleichzeitigen Vorkommens zweier Topic-Wörter im Referenzkorpus als Indikator für ihre semantische Relation zu betrachten. Je häufiger zwei Topic-Wörter im Referenzkorpus gemeinsam auftauchen, desto höher ist ihr Kohärenz-Wert. Der durchschnittliche Kohärenz-Wert aller Wortpaare eines Topics entspricht

dabei dem Kohärenz-Wert des Topics. Darüber hinaus gilt: Je höher dieser Wert ausfällt, desto besser ist das jeweilige Topic für Menschen interpretierbar.

In Strube & Ponzetto (2006) wird vorgeschlagen, Wikipedia wegen seines enzyklopädischen Charakters für die Berechnung der semantischen Relation der Wörter zu verwenden: „Wikipedia provides entries on a vast number of named entities and very specialized concepts. … providing a large coverage knowledge resource developed by a large community, which is very attractive for information extraction applications." In Newman, Lau, et al. (2010) wird nachgewiesen, dass das Wikipedia-basierte Topic-Kohärenzmaß (*UCI coherence*) unter den getesteten 15 Methoden die beste Evaluationsmethode ist. Dem folgten in den vergangenen Jahren mehrere Untersuchungen (z. B Mimno et al. (2011), Stevens et al. (2012), Aletras & Stevenson (2013), Röder et al. (2015)), die weitere Kohärenzmaße vorstellten. Die auf der Wort-Kookkurrenz basierenden Kohärenzmaße werden derzeit als das Standardverfahren betrachtet, um die Topic-Kohärenz und die Interpretierbarkeit von Topics automatisch zu evaluieren.

Um Klarheit darüber zu erlangen, wie gut ein Kohärenzmaß die menschliche Bewertung der Topics wiedergeben kann, werden die Topic-Kohärenzmaße auch evaluiert. Ein bereits verbreitetes Schema der Evaluation wird z. B. in Newman, Lau, et al. (2010), Aletras & Stevenson (2013), Lau et al. (2014), Rosner et al. (2014), Röder et al. (2015) und Xing et al. (2019) eingesetzt: Für die Evaluation wird die Interpretierbarkeit der Topics zunächst manuell bewertet, also hinterfragt, wie gut die Topics als Themen interpretierbar sind. Diese Bewertungen werden als Goldstandard betrachtet. Anschließend werden die Kohärenz-Werte aller Topics anhand des zu evaluierenden Kohärenzmaßes berechnet. Als Ergebnis ergeben die menschlichen Bewertungen und die Kohärenz-Werte zwei Rangfolgen der Topics. Die Korrelation der beiden Rangfolgen wird in einem weiteren Schritt berechnet. Eine gute Qualität eines Kohärenzmaßes wird durch eine hohe Rangkorrelation zwischen den menschlichen Bewertungen und den Kohärenz-Werten angezeigt.

Das einfachste Kohärenzmaß ist die *UCI coherence*. Sie basiert auf Pointwise Mutual Information (PMI, siehe unten). Der Zähler $\boldsymbol{P(w_i,w_j)}$ ist die beobachtete Kookkurrenz von $\boldsymbol{w_i}$ und $\boldsymbol{w_j}$ im Referenzkorpus, während der Nenner $\boldsymbol{P(w_i)P(w_j)}$ die erwartete Kookkurrenz von $\boldsymbol{w_i}$ und $\boldsymbol{w_j}$ im Referenzkorpus ist. Ein größerer Wert für die PMI spiegelt eine stärkere Assoziation zwischen $\boldsymbol{w_i}$ und $\boldsymbol{w_j}$, wider. Wenn $\boldsymbol{w_i}$ und $\boldsymbol{w_j}$ im Text nicht zusammen auftauchen, ist $\boldsymbol{P(w_i,w_j)}$ =

0. Um den Logarithmus von 0 zu vermeiden, wird ϵ hinzugefügt. Der Wert von ϵ wird auf 1 eingestellt. Für alle Wortpaare in den Top-N Topic-Wörtern eines Topics werden die PMI-Werte berechnet. Das arithmetische Mittel aller PMI-Werte ist der Kohärenz-Wert des Topics.

$$PMI(w_i, w_j) = \log_2 \frac{P(w_i, w_j) + \epsilon}{P(w_i)P(w_j)}$$

$$C_{\mathrm{UCI}} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI\ (w_i, w_j)$$

Ein anderes Beispiel für das Kohärenzmaß ist die *UMass coherence* aus Mimno et al. (2011). Der Ausgangspunkt dieses Maßes ist die Annahme, dass das Vorkommen eines Topic-Wortes in einem Topic von dem vorherigen Topic-Wort abhängt. Daher kommt ein Topic-Wort wahrscheinlicher in einem Dokument vor, wenn das vorherige Topic-Wort in demselben Topic bereits im Dokument auftaucht. Für jedes Wort in einem Topic wird der Logarithmus der Wahrscheinlichkeit berechnet, indem jedes weitere Top-Wort verwendet wird, welches eine höhere Top-Wort-Rangordnung als Bedingung erfüllt. Die Wahrscheinlichkeiten leiten sich aus der Häufigkeit des Vorkommens im Korpus ab. Der zusammengefasste Wert aller konditionalen Wahrscheinlichkeiten entspricht hier dem Kohärenz-Wert des Topics. Im Zähler wird hier ebenfalls ϵ addiert, um einen Logarithmus von 0 zu vermeiden. Später wurde bei Stevens et al. (2012) entdeckt, dass ein Kohärenzmaß (sowohl *UCI coherence* als auch *UMass coherence*) besser anwendbar ist, wenn für ϵ statt 1 ein deutlich kleinerer Wert (z.B. $\epsilon = 10^{-12}$) verwendet wird.

$$C_{\mathrm{UMass}} = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)}$$

Bei Röder et al. (2015) werden die existierenden Kohärenzmaße systematisch analysiert und in einem Kohärenzmaß-Framework vereint. Dieses besteht aus vier Teilen, nämlich

- Segmentierung (Segmentation of word subsets): Die Kohärenz eines Topics wird durch die Kohärenz der Topic-Wörter festgelegt. Obwohl der Kohärenzwert eines Topics sich bei den meisten Kohärenzmaßen aus den zusammengefassten Werten aller Wortpaare

im Topic ergibt, existieren andere Möglichkeiten. Nach Lund & Burgess (1996) tauchen „road“ und „street“ in englischen Text fast niemals zusammen in einem Text auf. Eine ähnliche Situation wird in Lemaire & Denhiere (2006) beschrieben. In einem französischen Korpus basierend auf Inhalten aus der Zeitung „Le Monde“ (das Korpus enthält 24 Millionen Wörter), kommen die Begriffe „internet“ und „web“ jeweils 131- und 94-mal vor, sie weisen allerdings eine Kookkurrenz von 0 auf. Offensichtlich kann die Wort-Wort-Kookkurrenz die semantische Verwandtschaft von zwei Komponenten eines Topics nicht immer problemlos widerspiegeln. Dies lässt sich umgehen, wenn die Kohärenz zwischen einem Topic-Wort und einer Teilmenge anderer Topic-Wörter, zwischen einem Topic-Wort und allen anderen Topic-Wörtern im Topic oder zwischen zwei Teilmengen der Topic-Wörter berechnet wird. Auf diese Weise können auch die „higher-order of co-occurrences“-Informationen (Niemann et al., 2012) berücksichtigt werden. Aus diesem Grund besteht der erste Schritt der Kohärenzberechnung darin, Topic-Wörter zu segmentieren.

- Berechnung der Wahrscheinlichkeit (Probability Calculation): In diesem Schritt wird die Kookkurrenz der zwei Wörter im Referenzkorpus berechnet. Es wird überprüft, wie oft diese beiden Wörter jeweils im Korpus vorkommen und wie häufig sie in einem Textfenster gemeinsam auftauchen. Die Fenstergröße ist dabei frei definierbar, sie kann sich auf ein Dokument, ein Paragraph, einen Satz oder ***n*** Wörter beziehen.
- Bestätigungsmaß (Confirmation Measure): Ein Bestätigungsmaß verbindet die berechneten Wahrscheinlichkeiten und gibt die Assoziation zwischen zwei Wörtern an. Dabei wird zwischen einem direkten und einem indirekten Bestätigungsmaß unterschieden. Ein direktes Bestätigungsmaß, wie z. B. *UMass coherence*, berechnet unmittelbar die Assoziation. Das Problem der direkten Bestätigungsmaße ist, dass manche Wörter sehr selten zusammen in einem Text vorkommen, obwohl sie semantisch verwandt sind (bereits erwähnt wurde, dass „internet“ und „web“ null Kookkurrenz haben). Die Assoziation dieser zwei Wörter kann dennoch durch ihre Kookkurrenzen mit anderen Wörtern wie „computer“, „WWW“, „IT“ usw. erfasst werden. Dies stellt genau die Idee der indirekten Bestätigungsmaße dar: Jedes Wort wird durch einen Vektor repräsentiert, der die Kohärenzwerte zwischen diesem Wort und allen anderen Wörtern im Topic enthält.

$$\overrightarrow{w_i} = \left\{ \sum_{\substack{1 \le i \le n-1 \\ i+1 \le j \le n}} assoc(w_i, w_j) \right\}$$

Die Kohärenz von zwei Wörtern entspricht hier der Ähnlichkeit (vergleichbar mit der Kosinus-Ähnlichkeit) ihrer Vektoren.

$$Sim_{cos}(\overrightarrow{w_i}, \overrightarrow{w_j}) = 1 - \frac{\overrightarrow{w_i} \cdot \overrightarrow{w_j}}{\| \overrightarrow{w_i} \| \| \overrightarrow{w_j} \|}$$

- Aggregation: Im letzten Schritt werden die Kohärenz-Werte aller Wortpaare zu einem Topic-Kohärenzwert zusammengefasst. Hier wird zumeist der arithmetische Mittelwert für die Aggregation verwendet, es lassen sich aber auch andere Maße wie z. B. Median, Maximum oder Minimum einsetzen.

In Röder et al. (2015) wird die Effektivität von 237.912 unterschiedlichen Kohärenzmaßen systematisch getestet. Das Ergebnis legt nahe, dass ein Kohärenzmaß die Assoziation zwischen einem Topic-Wort und einer großen Teilmenge der restlichen Topic-Wörter messen soll. Das Textfenster soll mindestens 50 Wörter enthalten, für die Aggregation sollte der arithmetische Mittelwert oder der Median verwendet werden. Für die Berechnung der Kookkurrenz wird statt des Topic-Modeling-Korpus ein Referenzkorpus wie z. B. Wikipedia verwendet. Das beste Kohärenzmaß $\boldsymbol{C_V}$ [46], welches eine Korrelation mit der menschlichen Interpretation von durchschnittlich 0,731 erreichte, ist eine Kombination von folgenden Komponenten:

- Berechnung der Assoziation zwischen einem Topic-Wort und einer Teilmenge von anderen Topic-Wörtern,
- Textfenster mit einer Größe von 110 Wörtern,
- die Kosinus-Ähnlichkeit der Wortvektoren, die Normalized Pointwise Mutual Information (NPMI, siehe unten) zwischen Topic-Wörtern enthalten, als indirektes Bestätigungsmaß. NPMI wird erstmals in Bouma, (2009) vorgestellt. Im Vergleich zu PMI kann NPMI das Bias gegenüber weniger häufigen Wörtern reduzieren. Außerdem

[46] Auf https://github.com/dice-group/Palmetto/issues/13 (17.08.2021) wird berichtet, dass die $\boldsymbol{C_V}$-Kohärenz-Werte im Paper nicht reproduzierbar sind. Deshalb wird vom Autor später vorgeschlagen, statt $\boldsymbol{C_V}$ andere Kohärenzmaße wie. z.B. NPMI oder UCI zu verwenden.

hat NPMI einen standardisierten Wertbereich von [-1, 1]. NPMI = 1 steht dafür, dass w_i und w_j immer im Text gemeinsam vorkommen. NPMI = 0 bedeutet, dass w_i und w_j zufällig in einem Text gemeinsam auftauchen; NPMI = -1 bedeutet, dass w_i und w_j nicht zusammen vorkommen. Laut Hao et al. (2018) ist das NPMI-basierte Kohärenzmaß das erfolgreichste Maß für die automatische Evaluation der Topic-Kohärenz,

$$NPMI(w_i, w_j) = \left(\frac{\log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}}{-\log(P(w_i, w_j) + \epsilon)} \right)$$

- arithmetischer Mittelwert für die Aggregation.

In Martin & Jurafsky (2009) wird als eine weitere Möglichkeit vorgeschlagen, den T-Test für die Evaluation von Topics zu verwenden. Formal ist die T-Test-Assoziation folgendermaßen definiert:

$$assoc_{t-test}(w_i, w_j) = \frac{P(w_i, w_j) - P(w_i) \cdot P(w_j)}{\sqrt{P(w_i) \cdot P(w_j)}}$$

Wie bei der Definition von PMI ist $P(w_i,w_j)$ hier die Beobachtung der Kookkurrenz von w_i und w_j. Das $P(w_i)P(w_j)$ zeigt die Erwartung der Kookkurrenz von w_i und w_j. Der Unterschied zwischen der Beobachtung und der Erwartung wird berechnet und normalisiert durch den Nenner $\sqrt{P(w_i) \cdot P(w_j)}$. Größere Werte zeigen eine stärkere Kohärenz zwischen w_i und w_j.

3.2.8 Zwischenfazit zur Evaluation der Topics

Aus dem Vergleich der vorhandenen Evaluationsmethode der Topics kann festgestellt werden, dass die Evaluation der Topics hauptsächlich auf zwei Arten durchgeführt werden. Die eine besteht darin, die Topics auf der Grundlage ihrer verschiedenen statistischen Eigenschaften zu bewerten (e.g. J/I-Topics oder die deskriptive Metriken in MALLET). Die andere besteht darin, die (semantische) Verbindung zwischen den im Topic enthaltenen Wörtern zu bewerten (e.g. Word Intrusion oder Topic Kohärenzmaße). Für die Anwendung von Topic Modeling in den Digital Humanities ist die zweite Art der Evaluation besser geeignet, weil diese Methode die

Interpretierbarkeit der Topics widerspiegelt. Aus diesem Grund wird Topic Kohärenzmaße für die Evaluation der Topics in dieser Arbeit verwendet. Da in früheren Forschungen mehrere Kohärenzmaße vorgeschlagen wurden, werden die Maße im nächsten Kapitel genauer analysiert, um festzustellen, welches Maß zur Evaluation der Topics verwendet werden soll.

4.Analyse des Topic-Kohärenzmaßes

Wird Topic Modeling als eine explorative textanalytische Methode für digitale geisteswissenschaftliche Forschungen eingesetzt, erwartet man eine Reihe von Wortlisten (Topics), die einen Einblick der thematischen Struktur eines Textkorpus ermöglichen. Allerdings kann Topic Modeling nicht garantieren, dass alle Topics für Menschen als Themen interpretierbar sind. Deshalb ist die Evaluation der Topics, vor allem in Digital Humanities, eine sehr wichtige Aufgabe. Zurzeit ist die Kohärenz-basierte Evaluationsmethode das Standardverfahren der automatischen Topic-Evaluation. Laut Hoyle et al. (2021) wird die menschliche Bewertung der Topics durch die Kohärenz-basierte automatische Evaluation komplett ersetzt. In diesem Kapitel wird der Ansatz der Topic-Kohärenzmaße durch Untersuchungen analysiert, um vor allem die potenziellen Probleme dieser Evaluationsmethode genau zu verstehen.

4.1 Variabilität der Korrelation zwischen den menschlichen Bewertungen und den Kohärenz-Werten

Wie bereits vorgestellt, repräsentiert die Korrelation zwischen den menschlichen Bewertungen und den Kohärenz-Werten der Topics, wie gut ein Kohärenzmaß die menschliche Bewertung der Topics wiedergibt. Für die Berechnung der Korrelation können der Pearson-Korrelationskoeffizient, der Spearman'sche Rangkorrelationskoeffizient oder der Kendall'sche Rangkorrelationskoeffizient eingesetzt werden. Der Pearson-Korrelationskoeffizient wird etwa in Lau et al. (2014) und Röder et al. (2015) verwendet, während der Spearman'sche Rangkorrelationskoeffizient in Aletras & Stevenson (2013) zum Einsatz kommt. Der theoretische Wertbereich aller drei Korrelationskoeffizienten liegt zwischen -1 und +1, wobei 0 keine Korrelation impliziert. Eine positive Korrelation bedeutet, dass **Y** mit der Zunahme von **X** ebenfalls zunimmt, umgekehrt bedeutet eine negative Korrelation, dass **Y** mit der Zunahme von **X** abnimmt. [47] Der Spearman'sche Rangkorrelationskoeffizient ist dabei weniger empfindlich gegenüber Ausreißern als der Pearson-Korrelationskoeffizient. Außerdem basiert der Pearson-Korrelationskoeffizient auf den Rohdaten der Variablen und berechnet die lineare Beziehung zwischen zwei kontinuierlichen Variablen. Im Vergleich dazu basieren die Rangkorrelationskoeffizienten auf den Rangwerten für jede Variable und berechnen die monotone Beziehung zwischen zwei kontinuierlichen oder ordinalen Variablen. In einer

[47] Vgl. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html, (05.05.2020).

monotonen Beziehung neigen die Variablen dazu, sich gemeinsam zu verändern, aber nicht unbedingt mit einer konstanten Rate. Deshalb sind die Rangkorrelationskoeffizienten streng genommen die bessere Auswahl für die Evaluation des Kohärenzmaßes. Der Pearson-Korrelationskoeffizient wird etwa in Xing et al. (2019) eingesetzt, wobei das Untersuchungsergebnis zeigt, dass die „Variability"-basierte Evaluationsmethode mit der menschlichen Bewertung der Topics besser korreliert. Allerdings kann sich das Ergebnis anders darstellen, wenn der Spearman'sche Rangkorrelationskoeffizient eingesetzt wird (Abbildung 4.1).[48]

Korrelationskoeffizient	**Kohärenzmaß**	**20NG**	**Wiki**	**NYT**	**Mean**
Pearson	PMI	0,602	0,550	0,623	0,592
	NPMI	0,640	0,567	0,639	0,615
	Variability	0,678	0,701	0,773	**0,717**
Spearman	PMI	0,646	0,555	0,583	0,595
	NPMI	0,671	0,547	0,594	0,604
	Variability	0,639	0,571	0,537	**0,582**

Tabelle 4.1 Pearson- und Spearman'sche Korrelation zwischen menschlichen Bewertungen und drei Topic-Kohärenzmaßen nach Xing et al. (2019)

Werden die Evaluationsergebnisse der Kohärenzmaße verglichen, ist zu beobachten, dass diese zu unterschiedlichen Ergebnisse in verschiedenen Untersuchungen führen können. In Aletras & Stevenson (2013) kann NPMI die Rangkorrelationen von 0,74, 0,76 und 0,76 für die folgenden drei Datensätze erzielen: Topics aus der „New York Times" (NYT), aus der 20 News Group Data Collection (20NG) und aus wissenschaftlichen Artikeln, die in 49 Zeitschriften von MEDLINE veröffentlicht wurden (Genomics). Im Vergleich dazu liegen die Rangkorrelationen für die gleichen Datensätze in Röder et al. (2015) auf 0,806 für NYT, 0,780 für 20NG und 0,594 für Genomics. Hier wurde die gleiche Methode mit identischen Daten getestet, aber das Ergebnis ist unterschiedlich. Der Autor erklärt diese Abweichung folgendermaßen: „Results differing to Aletras and Stevenson might be caused by a different preprocessing and different versions of the Wikipedia." Darüber hinaus werden in den beiden Arbeiten unterschiedliche

[48] Die in Xing et al. (2019) veröffentlichen Daten befinden sich unter: https://github.com/lxing532/topic_variability, (24.11.2020).

Rangkorrelationskoeffizienten verwendet, was ein weiterer Grund für die unterschiedlichen Ergebnisse sein könnte.

Die oben erwähnte Variabilität liegt innerhalb akzeptabler Grenzen, weil die Abweichung nicht besonders groß ist. Im Gegensatz dazu ist eine weitere Variabilität bemerkenswerter. C_P aus Röder et al. (2015) erzielt im Originalpaper Rangkorrelationen von 0,757 für NYT und 0,825 für 20NG. In Xing et al. (2019) werden allerdings nur Werte von 0,061 für NYT und 0,378 für 20NG erreicht. *DS* [49] aus Aletras & Stevenson (2013) erzielt im Originalpaper Rangkorrelationen von 0,76 für NYT und 0,8 für 20NG und kommt in Röder et al. (2015) auf 0,739 für NYT und 0,747 für 20NG. Die Abweichung ist nicht groß, weil C_P und *DS* in den beiden Artikeln mit den gleichen Daten getestet werden. In Xing et al. (2019) betragen die Werte allerdings jeweils 0,365 und 0,461. Die Gründe für so eine große Varianz werden in dem entsprechenden Paper leider nicht erläutert.

Technische Faktoren wie zum Beispiel unterschiedliches Preprocessing oder verschiedene Korrelationskoeffizienten können zur Variabilität des Evaluationsergebnisses beigetragen haben. Ein anderer wichtiger Faktor ist die menschlichen Bewertungen der Topics. In fünf Papers werden Topics für die Evaluation der Kohärenzmaße manuell bewertet (Tabelle 3.1). Durch eine genauere Überprüfung kann festgestellt werden, dass die manuelle Bewertung der Topics sehr unterschiedlich durchgeführt wird.

Zunächst unterscheiden sich die eingesetzten Korpora der fünf Untersuchungen in Korpusgröße, Textart und thematischem Gebiet. Zweitens sind die Methoden der Bewertung unterschiedlich. In vier Untersuchungen wird die Qualität der Topics auf einer 3- (oder 4-) Punkte-Skala bewertet, während in Lau et al. (2014) die Qualität durch Word-Intrusion bewertet wird. Selbst bei den vier Untersuchungen, die die Topics in Skalen bewertet haben, gibt es große Unterschiede in der konkreten Umsetzung. In Newman et al. (2010) wird ein Topic als „useful“ bewertet, wenn es für den Annotator sinnvoll, interpretierbar und themenüberschriftenähnlich ist und der Annotator sich vorstellen kann, es in einer Suchoberfläche zu verwenden, um Dokumente zu einem bestimmten Thema zu finden. Im gegenteiligen Fall ist ein Topic „useless“. In Mimno et al. (2011) werden die Topics zuerst manuell in drei Arten kategorisiert („research“, „grant mechanisms and publication types“ und „general“). Ein Topic wird dann als „good“ annotiert, wenn es die Wörter enthält, die als ein

[49] Das Maß wird in (Röder et al., 2015) als C_A bezeichnet.

einziges kohärentes Konzept interpretiert werden können. Zusätzlich kann ein „research"-Topic nur als „good" betrachtet werden, wenn die mit diesem Topic verbundenen Dokumente die Texte enthalten, die sich auf das von ihm interpretierte Konzept beziehen. In Aletras & Stevenson (2013) ist ein Topic „useful", wenn es semantisch kohärent, sinnvoll und interpretierbar ist. Ein Topic ist „average", wenn einige Wörter des Topics kohärent und interpretierbar sind, „useless" ist es dann, wenn die enthaltenen Wörter zufällig erscheinen und ohne Bezug zueinander sind. In Xing et al. (2019) werden die Kriterien für die Bewertung der Qualität der Topics nicht erläutert. Drittens werden in allen fünf Untersuchungen 100 bis 300 Topics manuell bewertet. Allerdings werden die Topics durch eine unterschiedliche Anzahl von Worten repräsentiert (fünf bis 30). Da viertens unterschiedliche Personen ein Topic unterschiedlich bewerten könnten, wird jedes einzelne stets von mehreren Annotatoren begutachtet. So werden zum Beispiel die Topics bei Chang et al. (2009) von acht Annotatoren bewertet. Die gleichen Topics werden in Lau et al. (2014) von durchschnittlich 11,7 Annotatoren zusätzlich beurteilt. Danach werden alle Bewertungen aggregiert und als Goldstandard für die Qualität der Topics betrachtet. In vier Untersuchungen werden die Bewertungen der Topics zumeist von Annotatoren selbstständig auf einer Crowdsourcing-Plattform erledigt. Im Vergleich dazu müssen die zwei Annotatoren in Mimno et al. (2011) miteinander über ihre Bewertungen eines Topics diskutieren und sich auf ein einheitliches Urteil einigen, wenn sie ein Topic zunächst unterschiedlich bewertet haben. Außerdem wird in Mimno et al. (2011) und Aletras & Stevenson (2013) entschieden, die Topics aus medizinischen Bereichen nur von Annotatoren bewerten zu lassen, die entsprechenden Fachkenntnisse haben.

Der oben vorgestellte Vergleich zeigt, dass die manuelle Bewertung der Topics in mehreren Ebenen unterschiedlich durchgeführt wird. Deshalb ist es wenig verwunderlich, dass ein Kohärenzmaß in verschiedenen Untersuchungen unterschiedlich funktioniert. Die Variabilität der Evaluationsergebnisse der Kohärenzmaße ist ein wichtiger Faktor, der nicht ignoriert werden darf. Aus diesem Grund ist es wenig ratsam, die in verschiedenen Arbeiten berichteten Testergebnisse undifferenziert zu vergleichen.

	Newman, Lau, et al. (2010)	Mimno et al. (2011)	Aletras & Stevenson (2013)	Lau et al. (2014)	Xing et al. (2019)
Korpora für Topic Modeling	55.000 Zeitungsartikel aus dem englischen Gigaword, 12.000 Bücher aus Internet-Archiven	300.000 Abstracts von Zeitschriftenartikeln des National Institutes of Health	47.229 Nachrichtenartikel der „New York Times“, 20.000 Newsgroup-Emails, 30.000 wissenschaftliche Artikel aus 49 MEDLINE-Zeitschriften	8447 Nachrichtenartikel der „New York Times“, 10.000 ausgewählte Artikel aus Wikipedia (Daten aus Chang et al. (2009))	9347 Absätze aus Newsgroup-Emails, 10.773 Artikel aus Wikipedia, 8764 Nachrichtenartikel der „New York Times“
Methode	Qualität des Topics wird auf einer 3-Punkte-Skala bewertet	Qualität des Topics wird auf einer 3-Punkte-Skala bewertet	Qualität des Topics wird auf einer 3-Punkte-Skala bewertet	Word-Intrusion	Qualität des Topics wird auf einer 4-Punkte-Skala bewertet
Objekt	237 Topics	148 Topics, Top-30 Topic-Wörter	300 Topics, Top-10 Topic-Wörter	300 Topics, Top-5 Topic-Wörter	100 Topics, Top-10 Topic-Wörter
Annotator	9 Annotatoren	2 Fachgebietsexperten	12 bis 26 Universitätsmitarbeiter und graduierte Studenten einer Crowdsourcing-Plattform	8 + 11, 7 Annotatoren einer Crowdsourcing-Plattform	5 Annotatoren einer Crowdsourcing-Plattform

Tabelle 3.1 wichtige Daten der manuellen Bewertung von Topics in fünf Papers

4.2 NPMI-basierte Evaluation von deutschen Topics:

Alle verfügbaren Evaluationen von Kohärenzmaßen basieren auf englischen Texten. Da die in dieser Arbeit folgenden Untersuchungen des Topic Modeling aber auf deutschen Textdaten basieren, ist es sinnvoll, ein Verständnis dafür zu entwickeln, wie sich das Ergebnis darstellt, wenn das Kohärenzmaß für die Evaluation von deutschen Topics verwendet wird. Aus diesem Grund wurde ein Topic-Modell mit 100 Topics auf eine Sammlung deutscher Zeitungsartikel[50] trainiert. Die 100 Topics wurden in zwei Gruppen geteilt. Jede Gruppe enthält 50 Topics und sie wurden von sechs Annotatoren bzw. Studierenden bewertet, die Digital Humanities studieren. Diese Studierenden verstehen die Grundlage von Topic Modeling und wissen deshalb, dass ein Topic im Kontext des Topic Modeling nicht unbedingt ein Thema ist. Tabelle 4.2 zeigt die ersten vier Zeilen des verwendeten Fragebogens (der vollständige Fragebogen ist im Anhang der Arbeit enthalten). Am Anfang des Fragebogens wurden zwei bereits annotierte Beispieltopics angeführt, im Anschluss wurden die 100 zu annotierenden Topics in einer Tabelle aufgelistet.

Ein alltägliches Szenario für die Anwendung von Topic Modeling in Digital Humanities ist, dass zunächst eine Textsammlung aufgebaut wird, wobei meist schon eine grobe Vorstellung vom Inhalt vorhanden ist (z. B. eine Sammlung von Science-Fiction-Romanen oder eine Sammlung von Sportnachrichten). Mithilfe des Topic Modeling wird dann der Inhalt der Textdaten im Detail exploriert. Ausgehend von einem solchen Szenario wird im Fragebogen der Hinweis gegeben, dass die zu bewertenden Topics sich auf eine Sammlung von deutschen Zeitungsartikeln beziehen, wobei die Artikel aus den zehn Themengebieten „Digital“, „Gesellschaft“, „Karriere“, „Kultur“, „Lebensart“, „Politik“, „Reisen“, „Sport“, „Studium“ und „Wirtschaft“ stammen. Jedes Topic enthält die zehn wichtigsten, originalen Topic-Wörter und ein zufälliges falsches Wort, das hinzugefügt wurde. Die erste Aufgabe besteht in der Word-Intrusion, also darin, das falsche Wort zu identifizieren und es in das entsprechende Feld des Fragebogens einzutragen. Danach müssen die Annotatoren die verbleibenden zehn Wörter im Topic überprüfen und kontrollieren, ob das Topic zu einem der oben genannten Themengebiete gehört. Ist dies der Fall, tragen sie „ja“ und das vermutete Themengebiet im Feld „interpretierbar?“ bzw. „Thema“ ein. Die letzte Aufgabe erfordert es, im Feld „Anzahl der Topic-Wörter“ einzutragen, wie viele der zehn Topic-Wörter zum jeweiligen Thema gehören.

[50] Mehr Details zum Korpus siehe Kapitel 5.1.1 „Der Zeitungskorpus: eine Sammlung von 2000 Zeitungsartikeln“.

Ist dies bei keinem der Fall, sollen „nein“, „NA“ und „NA“ in den entsprechenden drei Feldern eintragen werden.

id	Topic	Ein falsches Wort	Interpretierbar?	Thema	Anzahl der Topic-Wörter
Bsp1	spiel, spieler, computerspiel, spielen, islam, ps, games, xbox, game, one, medium	islam	Ja	Digital	8
Bsp2	sagen, geben, mal, wissen, sehen, verlag, finden, einfach, denken, lassen, stehen	verlag	Nein	NA	NA
1	abmahnung, coupland, urmann, wandergeselle, streaming, ip, film, strafanzeige, redtube, adressen, illegal				
2	andersch, tscherkesse, wand, freundschaft, nachbar, tscherkessen, berzona, tscherkessisch, fatimat, organisationen, anderschs				

Tabelle 4.2 Fragebogen der manuellen Bewertung der Topics

Bei der Auswertung der annotierten Daten wird die Spearman’sche Rangkorrelation zwischen dem NPMI-Wert und den folgenden vier Bewertungen aller Topics berechnet: Wie viele Annotatoren können das falsche Wort finden, für wie viele von ihnen ist das Topic interpretierbar, wie viele Annotatoren können sich maximal auf das Themengebiet einigen und wie viele Topic-Wörter gehören zum jeweiligen Themengebiet? Die Korrelationen zwischen NPMI und den manuellen Bewertungen liegen zwischen 0,41 und 0,57 (Tabelle 4.3). Auch die Korrelationen zwischen den manuellen Bewertungen werden berechnet. Interessant ist hier, dass die Word-Intrusion-basierte Rangfolge der Topics mit allen anderen Rangfolgen relativ schwach (ca. 0,5) korreliert, während die anderen drei manuellen Bewertungen viel besser (über 0,85) miteinander korrelieren.

	NPMI	Ein falsches Wort	Interpretierbar?	Thema	Anzahl der Topic-Wörter
NPMI	1	0,490	0,514	0,417	0,567
Ein falsches Wort	0,490	1	0,494	0,447	0,527
Interpretierbar?	0,514	0,494	1	0,880	0,947
Thema	0,417	0,447	0,880	1	0,857
Anzahl der Topic-Wörter	0,567	0,527	0,947	0,857	1

Tabelle 4.3 Rangkorrelationen zwischen den Bewertungen

Werden die manuellen Bewertungen genauer betrachtet, lassen sich einige interessante Details erkennen (Tabelle 4.4). Das Topic 19 etwa ist für alle sechs Annotatoren interpretierbar und wurde dem Thema „Digital“ zugeordnet. Unter den zehn Topic-Wörtern gehören im Durchschnitt mehr als sieben zum Themengebiet. Allerding wurde das hinzugefügte falsche Wort von keinem der Annotatoren gefunden, alle haben „orange“ statt „lückemann“ als das falsche Wort gewählt. Im Vergleich zu „lückemann“ ist „orange“ deutlich auffälliger, weil das Wort als Frucht interpretiert wird und so scheinbar nicht zum Thema „Digital“ gehört. Tatsächlich aber ist die Orange S.A. das größte Telekommunikationsunternehmen in Frankreich und „orange“ deshalb ein Wort, das thematisch mit „Digital“ verbunden ist. Das zweite Beispiel (Topic 45) zeigt einen anderen Aspekt. Während die anderen drei Aufgaben fast perfekt erledigt werden, konnten die Annotatoren sich hier nicht auf das Themengebiet einigen. Zwei Annotatoren haben sich für „Kultur“ entschieden, „Lebensart“ und „Politik“ wurden jeweils von einem Annotator gewählt. Die beiden anderen Annotatoren haben „Islam“ und „Religion“ eingetragen. Das letzte Beispiel, Topic 94, ist für die meisten Annotatoren nicht interpretierbar. Nur von einem wird das Topic dem Themengebiet „Reisen“ zugeordnet. Interessant ist, dass alle das falsche Wort „charlene“ richtig erkannt haben. Die Beispiele zeigen, dass zu empfehlen ist, bei der Verwendung von Word-Intrusion ein Topic mehrfach zu evaluieren. Dabei sollten für jeden Durchgang andere „Intruder“-Worte verwendet werden. Das aggregierte Evaluationsergebnis kann dadurch eine geringere Zufälligkeit aufweisen.

id	Topic	Ein falsches Wort	Interpretierbar?	Thema	Anzahl der Topic-Wörter

19	facebook, nutzer, netzwerk, profil, tumblr, timeline, zuckerberg, schrems, lückemann, schenken, orange	lückemann / 0%	100%	Digital	7,67
45	islam, islamisch, redlich, muslime, religiös, kopftuch, tragen, khorchide, moschee, salafisten, muslimischen	redlich / 100%	100%	Kultur / Lebens-art / Politik	9,33
94	welt, wissen, vergessen, lassen, zeigen, bleiben, auge, erinnern, charlene, schön, wahr	charlene / 100%	17%	Reisen / NA	0,17

Tabelle 4.4 Beispiele der manuell bewerteten Topics

4.3 Vorsichtsmaßnahmen bei Verwendung des Topic-Kohärenzmaßes

Wird die Topic-Kohärenz verwendet, um die Qualität der Topics zu evaluieren, ist es sehr wichtig zu bedenken, dass der Topic-Kohärenz-Wert nur eine Annäherung in Bezug auf die Interpretierbarkeit des Topics darstellt. Hier ist Vorsicht geboten, weil der Kohärenz-Wert nicht nur zu Missverständnissen führen kann, sondern auch durch zwei Parameter beeinflusst wird. Diese beziehen sich auf die Anzahl der Wörter in den Topics. Es handelt sich dabei um die Mächtigkeit oder Kardinalität des Topics und Out-of-Vocabulary-Wörter, die in den Kapiteln 4.3.1 und 4.3.2 ausführlich behandelt werden.

Bevor der Einfluss der beiden Parameter vorgestellt wird, soll NPMI als ein Beispiel für ein Topic-Kohärenzmaß genauer betrachtet werden. Zur besseren Übersicht wird die Formel von NPMI hier nochmal wiedergegeben.

$$NPMI(w_i, w_j) = \left(\frac{\log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}}{-\log(P(w_i, w_j) + \epsilon)} \right)$$

Für die Berechnung des NPMI-Werts zwischen w_i und w_j sind $P(w_i)$, $P(w_j)$ und $P(w_i, w_j)$ notwendig. Hier konzentrieren wir uns zuerst auf $P(w_i, w_j)$ und schauen wir das folgende Beispiel an: es gibt zwei Wörter, die jeweils nur einmal in einem Referenzkorpus vorkommen

und sie kommen im Referenzkorpus nicht gemeinsam vor. Der Referenzkorpus enthält zehn Millionen Wörter. Deshalb sind $\boldsymbol{P(w_i)}$ und $\boldsymbol{P(w_j)}$ jeweils gleich eins geteilt durch zehn Millionen und $\boldsymbol{P(w_i, w_j)}$ ist gleich 0. ϵ ist gleich 10^{-12}. Der NPMI-Wert der beiden Wörter aus unserem Beispiel ist dann ca. 0,167 (der Nenner ist 12 und der Zähler ist 2). Nach der Definition bedeutet ein NPMI größer als 0, dass die beiden Begriffe in gewisser Weise zusammenhängen, was jedoch nicht der Fall ist. Dieses Problem kann sich auf die Evaluation der Topics auswirken. Die Interpretierbarkeit eines Topics, das mehrere im Referenzkorpus nicht zusammen auftauchende Wörter enthält, könnte wegen des NPMI-Werts falsch beurteilt werden.

4.3.1 Kardinalität des Topics

Wie bereits erwähnt, werden für die Berechnung der Topic-Kohärenz eines Topics die Top-$\boldsymbol{N}$ Topic-Wörter verwendet. Die Einstellung von $\boldsymbol{N}$ wird in Lau & Baldwin (2016) als Kardinalität des Topics (auf Englisch: „cardinality of the topic“) beschrieben. Laut dieser Arbeit wird die Kardinalität oft übersehen und willkürlich ausgewählt: In Chang et al. (2009) wird $\boldsymbol{N}$ auf 5 eingestellt, während der Wert in Newman, Lau, et al. (2010), Aletras & Stevenson (2013), Lau et al. (2014) auf zehn festgelegt wird. In Röder et al. (2015) wird $\boldsymbol{N}$ bei drei Datensätzen auf zehn und bei drei weiteren auf fünf eingestellt.

In Lau & Baldwin (2016) wird zudem untersucht, wie groß der Einfluss der Kardinalität auf die Kohärenz eines Topics ist. 300 Topics werden auf Wikipedia und „New York Times“-Artikeln trainiert und manuell als nicht kohärent, neutral und kohärent bewertet. Das Setting der Kardinalität ist $\boldsymbol{N} \in \{5, 10, 15, 20\}$ und die Top-$\boldsymbol{N}$ Topic-Wörter werden evaluiert. Jedes Topic wird durch mehrere Befragte bewertet, der durchschnittliche Wert ist dann der von Menschen ermittelte Kohärenzwert. Die menschlichen Bewertungen zeigt mit der Erhöhung von $\boldsymbol{N}$ eine schwache, aber systematische Absenkung: Die durchschnittliche Bewertung der Wikipedia-Topics sinkt von 2,42 über 2,37 und 2,35 auf 2,29 ab, die Bewertung der „New York Times“-Topics sinkt von 2,49 über 2,46 und 2,42 auf 2,39. Dies zeigt: Je mehr Wörter in einem Topic betrachtet werden, desto weniger wird es als kohärent wahrgenommen. Anschließend wird NPMI als ein Beispiel des Kohärenzmaßes für die folgende Untersuchung verwendet. Die Pearson-Rangkorrelation zwischen den automatisch berechneten NPMI-Werten und der menschlichen Annotation der 300 „New York Times“-Topics wird zusätzlich überprüft, wobei Wikipedia als Referenzkorpus für die NPMI-Berechnung verwendet wird. Hier lässt sich keine

systematische Änderung beobachten; die Korrelationen liegen jeweils auf 0,62, 0,68, 0,68 und 0,65, wenn ***N*** auf 5, 10, 15 und 20 eingestellt wird.

In Lau & Baldwin (2016) wird allerdings nicht getestet, wie groß der Einfluss der Kardinalität auf den Kohärenz-Wert eines Topics ist. Deshalb wird hier eine kleine Untersuchung durchgeführt. Topic-Modelle werden auf drei lemmatisierten deutschen Textsammlungen trainiert, die eine ähnliche Größe (ca. 20 Mb) haben: eine Sammlung von 2000 Zeitungsartikeln („Zeit“), eine Sammlung von 17 Romanen („pyDelta“) und eine Sammlung von 77 Dramen. Die durchschnittliche Länge der Zeitungsartikel beträgt 1800 Wörter, alle Romane und Dramen werden daher ebenfalls in 1800-Token-Segmente zerlegt. Die Wörter sind kleingeschrieben und die Standard-Stoppwörter werden entfernt. Die Topic-Modelle werden mit MALLET mit Standardparametereinstellungen trainiert. Das Setting der Kardinalität ist $\boldsymbol{N} \in \{3, 5, 10, 20\}$, das Setting der Topics-Anzahl ist $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$. Für jede Einstellungskombination von ***N*** und ***T*** werden zehn Topic-Modelle trainiert. Im Folgenden Lau & Baldwin (2016) wird hier das NPMI-basierte Kohärenzmaß verwendet. Der NPMI-Wert aller Topics wird berechnet, wenn ***N*** auf 3, 5 und 10 eingestellt wird. Bei $\boldsymbol{N} = 20$ werden die NPMI-Werte von zufällig ausgewählten 1080 Topics berechnet, weil die NPMI-Berechnung sehr zeitaufwendig ist.

Abbildung 4.1 zeigt, dass die durchschnittlichen NPMI-Werte der Topics aus allen drei Textsammlungen mit der Erhöhung von ***N*** ähnlich wie die manuellen Bewertungen in Lau & Baldwin (2016) systematisch absinken. Wenn man überprüft, wie die NPMI-Werte aller Topics verteilt sind (Abbildung 4.2), ist zudem zu beobachten, dass sich die Spannweite bzw. die Varianz der NPMI-Verteilungen mit der Erhöhung von ***N*** systematisch verkleinert. Mit der Erhöhung von ***N*** sinken die NPMI-Werte der Topics also nicht immer ab; die höchsten und die niedrigsten NPMI-Werte sind hier weniger extrem.

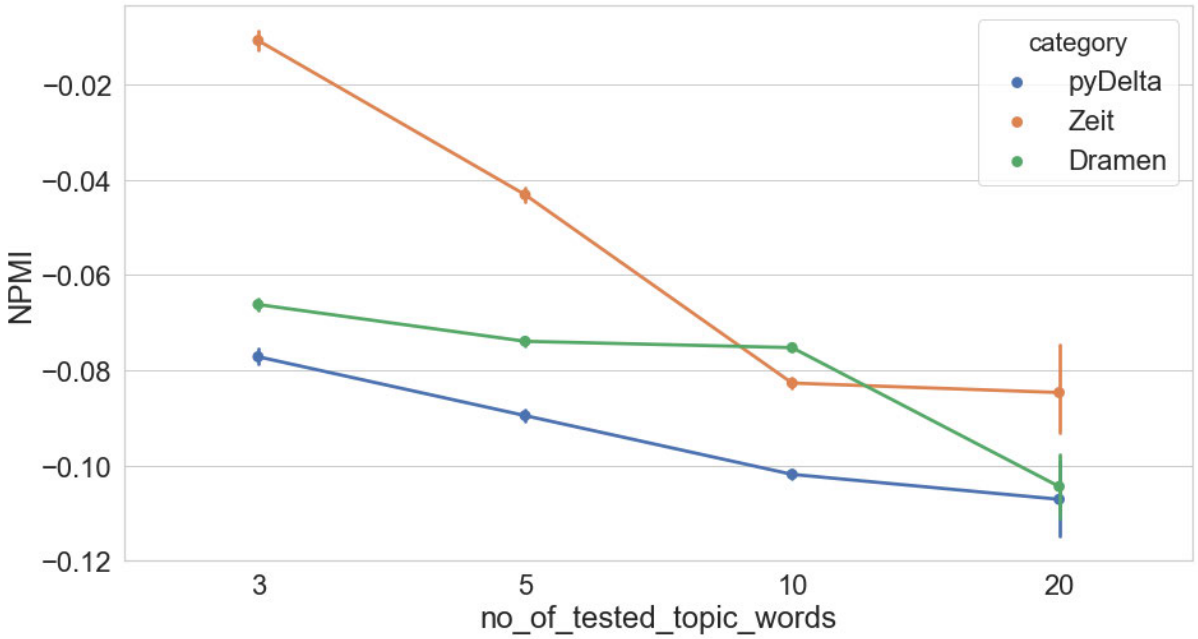

Abbildung 4.1 Durchschnittliche NPMI-Werte im Verhältnis zur Kardinalität der Topics

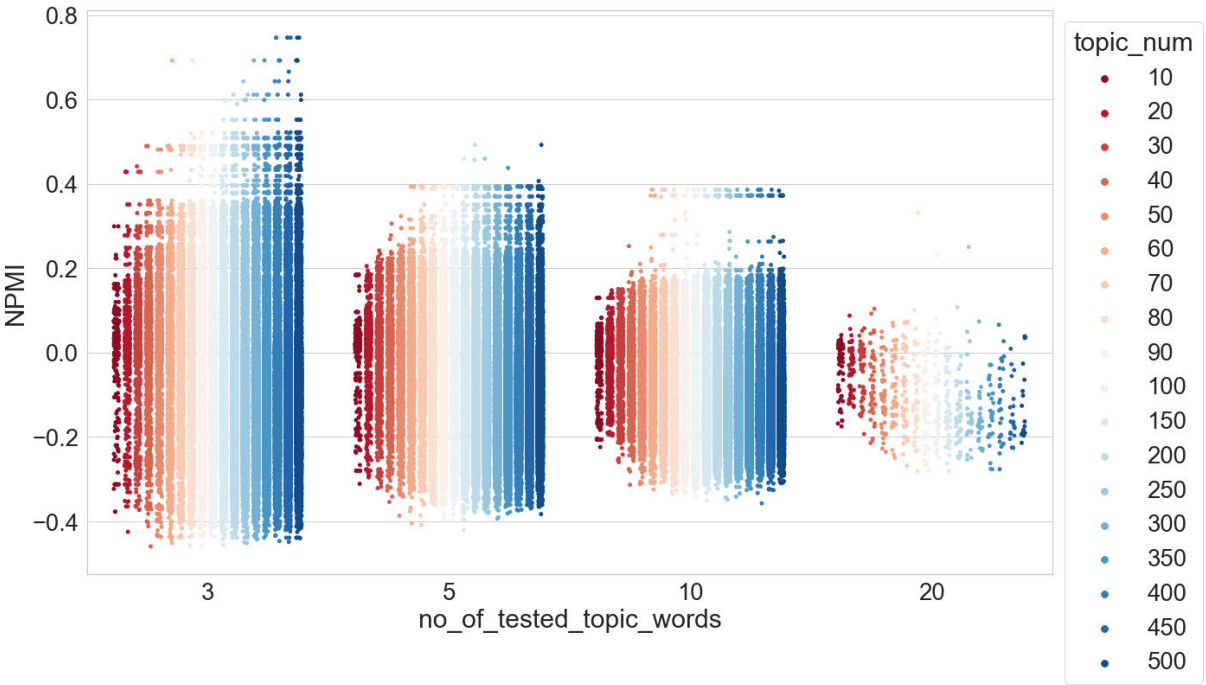

Abbildung 4.2 NPMI-Werte-Verteilung der Topics im Verhältnis zu Kardinalität der Topics und Anzahl der Topics

4.3.2 Out-of-Vocabulary-Wörter

Wie bereits erläutert ist der Grundgedanke des Topic-Kohärenzmaßes, das gleichzeitige Vorkommen zweier Topic-Wörter in einem externen Referenzkorpus zu überprüfen. Ein Problem dieses Ansatzes ist, dass das gemeinsame Auftauchen nicht überprüft werden kann, wenn ein Wort (oder die beiden Wörter) im Referenzkorpus nicht existieren. In dieser Arbeit werden diese Wörter als Out-of-Vocabulary (OoV)-Worte bezeichnet. Der Kohärenz-Wert dieser zwei Wörter kann deshalb nicht mehr berechnet werden, er wird stattdessen künstlich auf einen Wert gesetzt. In Palmetto, dem beliebtesten Werkzeug zur Berechnung der Kohärenz-Werte, wird dieser Wert auf 0 festgelegt, weil dies die Situation widerspiegelt, dass das System über keine Informationen zu diesen Wörtern verfügt.[51] Dadurch kann jedoch Verwirrung entstehen, weil ein Wert von 0 für NPMI eigentlich für das zufällige gemeinsame Vorkommen von zwei Wörtern in Text steht. Das Problem wird nicht nur bei der Verwendung des NPMI-basierten Kohärenzmaßes auftreten, sondern auch immer dann, wenn ein Kohärenzmaß zum Einsatz kommt, das auf Informationen aus externen Referenzkorpora basiert.

Vor allem sind OoV-Worte besonders problematisch, wenn Topic Modeling für Computational Literary Studies eingesetzt wird. Weil literarische Texte lexikalisch komplexer als Sachtexte (z. B. Zeitungsartikel, Tagebücher usw.) sind ist anzunehmen, dass ein Topic-Modell einer literarischen Textsammlung zahlreiche OoV-Wörter enthält. Das folgende Topic mit Top-10-Wörtern stammt aus einem Topic-Modell einer deutschen Romansammlung. Hier wird das deutsche Wikipedia aus dem Jahr 2017 als Referenzkorpus verwendet und es verwundert nicht, dass einige Topic-Wörter OoV-Wörter sind. Selbst wenn ein noch größerer Referenzkorpus verwendet wird, ist das Auftauchen von OoV-Wörter unvermeidlich.

> **„politicus, machiavellus, antiquus, candidus, appetitus, uranius, rationalis, gentiletus, sedulus, eruditus“**

Um die Annahme zu überprüfen, dass Topic-Modelle von literarischen Texten besonders viele OoV-Wörter enthalten können, wird im Folgenden eine kleine Untersuchung durchgeführt. Wie bereits dargelegt, werden für die Untersuchung der Kardinalität des Topics mehrere Topic-Modelle auf drei deutschen Textsammlungen trainiert: eine Zeitungsartikelsammlung, eine Romansammlung und eine Dramasammlung. Alle Topics aus den trainierten Topic-Modellen werden für diese Untersuchung verwendet. Anschließend wird gezählt, wie viele OoV-Wörter in den ersten $\boldsymbol{N} \in \{3, 5, 10, 20\}$ Topic-Wörtern jedes Topics existiert. Das Ergebnis wird in

[51] https://github.com/dice-group/Palmetto/issues/33, (19.08.2021).

Abbildung 4.3 visualisiert. Die schwarzen Punkte in der Grafik stellen den Median der OoV-Wörter jedes Settings der Anzahl der Topics dar. In den Topic-Modellen aller drei Textsammlungen können OoV-Wörter gefunden werden. Die Topic-Modelle enthalten besonders viel OoV-Wörter, wenn die Anzahl der Topics hoch eingestellt wird. Der Median der OoV-Wörter bleibt mit der Erhöhung der Anzahl der Topics und der Anzahl der Topic-Wörter ständig auf 0, wenn die Topic-Modelle auf Zeitungsartikeln trainiert werden. Im Vergleich dazu steigt der Median beim Drama- und beim Romankorpus langsam an und erreicht einen Höchststand von 8. Diese Untersuchung bestätigt die frühere Annahme, dass mehr OoV-Wörter in Topic-Wörter erscheinen, wenn Topic-Modelle auf literarischen Textsammlungen trainiert werden.

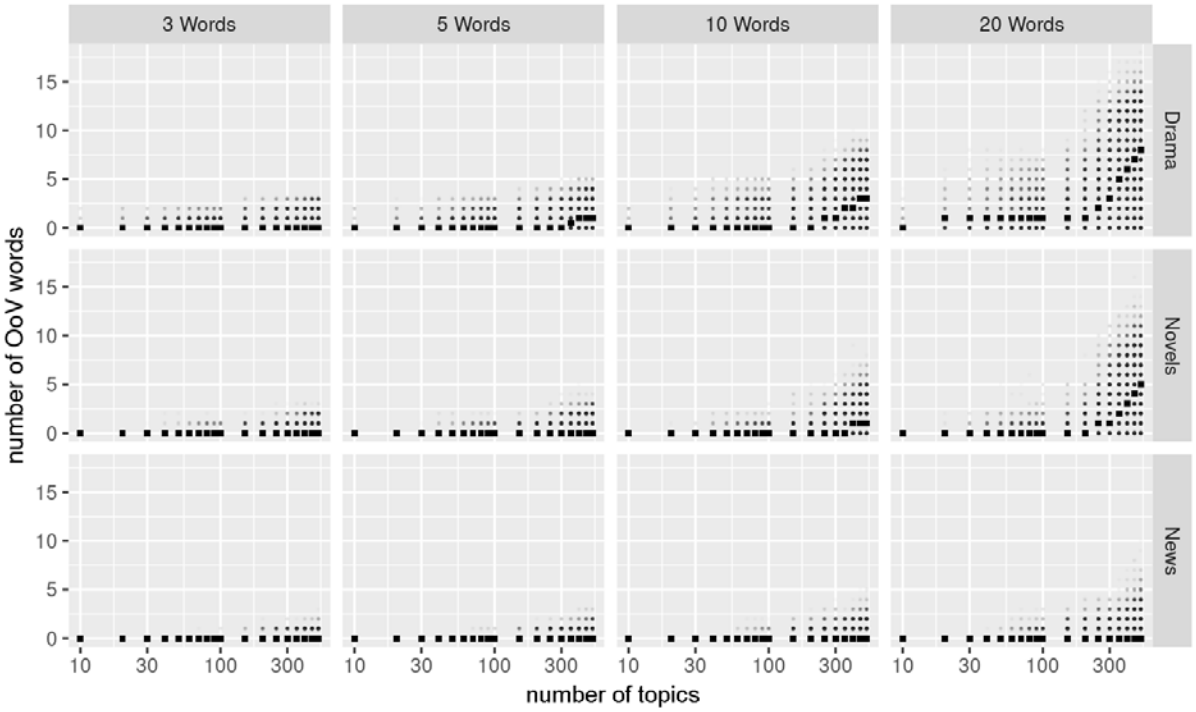

Abbildung 4.3 Anzahl der OoV-Wörter in den ersten N Topic-Wörtern

Nachdem die OoV-Wörter entdeckt worden sind, besteht der nächste Schritt darin, ihren Einfluss auf die Topic-Kohärenz zu explorieren. Für diese Untersuchung werden 50 Topics bei jedem Setting der Topic-Anzahl $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ aus den trainierten Topic-Modellen zufällig ausgewählt. Es wird zunächst kontrolliert, dass die zehn Topic-Wörter aller ausgewählten Topics keine OoV-Wörter sind.

Dann wird ein zufälliges Topic-Wort in den ersten $\boldsymbol{N} \in \{3, 5, 10\}$[52] Topic-Wörtern jedes ausgewählten Topics durch ein OoV-Wort[53] ersetzt und der NPMI-Wert des manipulierten Topics wird berechnet. Dieser Prozess wird dann wiederholt durchgeführt, wobei die Anzahl der OoV-Wörter erhöht wird, bis nur noch ein ursprüngliches Topic-Wort übrig ist. In diesem Fall kann der NPMI-Wert eines Topics nicht mehr berechnet werden und wird auf 0 gesetzt. Das Ergebnis wird in Abbildung 4.4 visualisiert. In der Grafik steht jedes Boxplot für die NPMI-Werte der 50 Topics. Die Topic-Anzahl wird hier auf 100 eingestellt. Mit der Erhöhung der Anzahl der OoV-Wörter ist zu beobachten, dass die NPMI-Werte aller Topics sich allmählich 0 nähern. Die Variabilität der NPMI-Werte nimmt deshalb ab, wobei dieser Trend unabhängig von der Art des Textes ist.

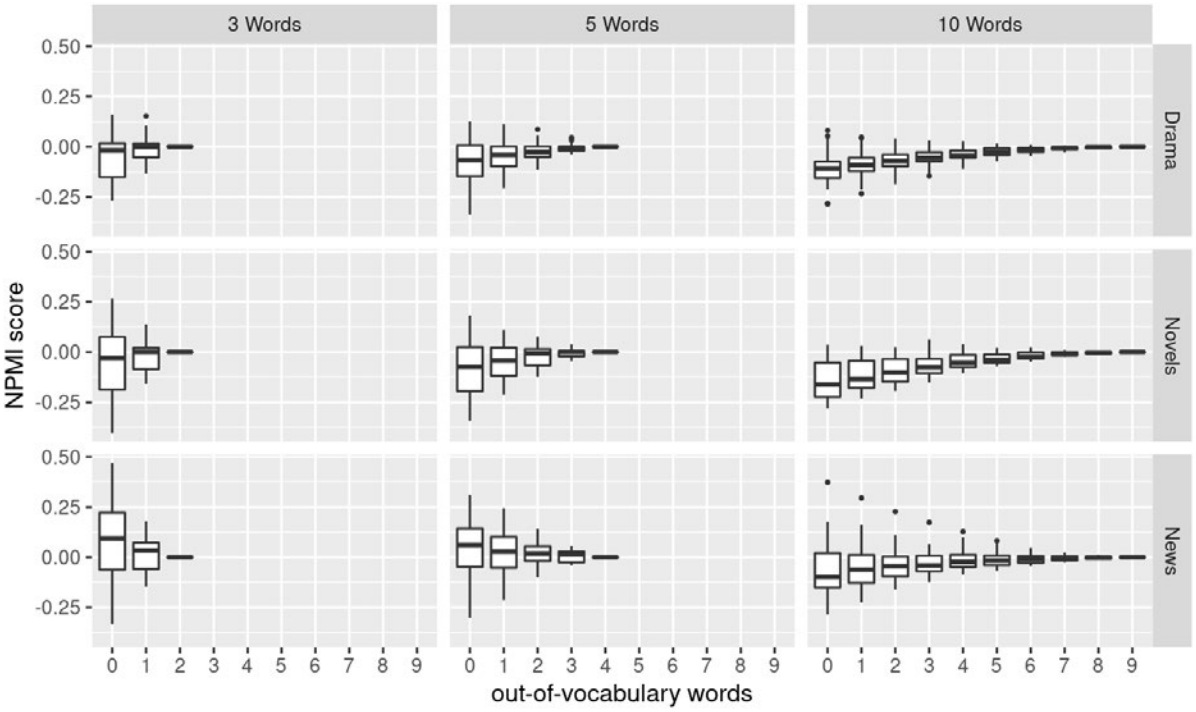

Abbildung 4.4 NPMI-Verteilungen im Verhältnis zur Anzahl der OoV-Wörter in den ersten N Topic-Wörtern

Es gibt mehrere offensichtliche Strategien, die zum Umgang mit dem Problem der OoV-Wörter angewandt werden könnten. In der oben vorgestellten Untersuchung ist zu beobachten

[52] In dieser Untersuchung wird $\boldsymbol{N} = 20$ nicht mehr getestet, weil die Berechnung der NPMI-Werte für 20 Topic-Wörter sehr zeitaufwendig ist.

[53] Diese sehr lange, bedeutungslose Zeichenkette „aifjpoisjfpoasijfpoinsfopnofijweopfijoishfoh“ wird als OoV-Wort verwendet, um sicherzustellen, dass dieses Wort in unserem Referenzkorpus nicht vorkommt.

(Abbildung 4.3), dass OoV-Wörter weniger häufig auftauchen, wenn man Topic-Modelle mit weniger Topics trainiert – dann kann ihr Einfluss ignoriert werden. Eine weitere Möglichkeit besteht darin, statt Wikipedia ein anderes Referenzkorpus für die automatische Berechnung der Topic-Kohärenz zu verwenden. Idealerweise soll das Referenzkorpus dem Untersuchungskorpus sprachlich und inhaltlich ähnlich sein. Die dritte Möglichkeit ist, alle OoV-Wörter als Stoppwörter zu betrachten und beim Training des Topic-Modells aus dem Untersuchungskorpus zu entfernen. Allerdings bringen diese drei Optionen Nachteile mit sich und sie sind wahrscheinlich für zahlreiche geisteswissenschaftliche Forschungsprojekte in der Praxis nicht anwendbar. Die vierte Möglichkeit ist, die OoV-Wörter in einem Topic zu entfernen und die Kohärenz der restlichen Topic-Wörter zu berechnen. Dadurch lässt sich eine Annäherung an den echten Kohärenzwert erhalten. Die Frage ist, wie weit diese Annäherung von den tatsächlichen Werten abweicht und inwiefern die echten Evaluationsergebnisse dadurch dargestellt werden können. Um diese Frage zu beantworten, lässt sich überprüfen, wie sich die Rangfolge der Topics nach den Topic-Kohärenz-Werten ändert, wenn die Anzahl der Topic-Wörter schrittweise reduziert wird.

Deshalb wird das folgende Experiment durchgeführt: Wie in der letzten Untersuchung werden 50 Topics bei jedem Setting der Topics-Anzahl $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ aus den trainierten Topic-Modelle zufällig ausgewählt. Es wird kontrolliert, dass die ersten zehn Topic-Wörter aller ausgewählten Topics keine OoV-Wörter sind. Die NPMI-Werte der Topics werden anhand der ersten zehn Topic-Wörter des Topics berechnet, diese werden nach ihren NPMI-Werten sortiert. Dadurch erhält man die originale Rangfolge der Topics. Dann werden $\boldsymbol{P} \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%\}$ der 50 Topics bei jedem Setting von $\boldsymbol{T}$ zufällig ausgewählt. Danach werden $\boldsymbol{W} \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ zufällige Topic-Wörter aus den zehn Topic-Wörtern der ausgewählten Topics entfernt. Die NPMI-Werte der manipulierten Topics werden anschließen anhand der restlichen Wörter in den zehn Topic-Wörtern berechnet. Nach der Berechnung aller NPMI-Werte werden die Topics nochmals nach ihren NPMI-Werten sortiert. Dadurch erhält man eine manipulierte Rangfolge der Topics. Der letzte Schritt der Untersuchung besteht darin, die Spearman'sche Rangkorrelation zwischen den originalen und den manipulierten Rangfolgen zu berechnen. Je größer die Rangkorrelation ist, desto weniger Auswirkungen hat die Reduzierung der Topic-Wörter auf die Rangfolge der Topics. Das bedeutet, dass die Annäherung der Kohärenz-Werte auch die originale Rangfolge der Topics darstellen kann. Das Experiment wird

auf alle drei Korpora (Zeitungskorpus, Dramakorpus und Romankorpus) durchgeführt, um die Auswirkung der Textarten zu explorieren.

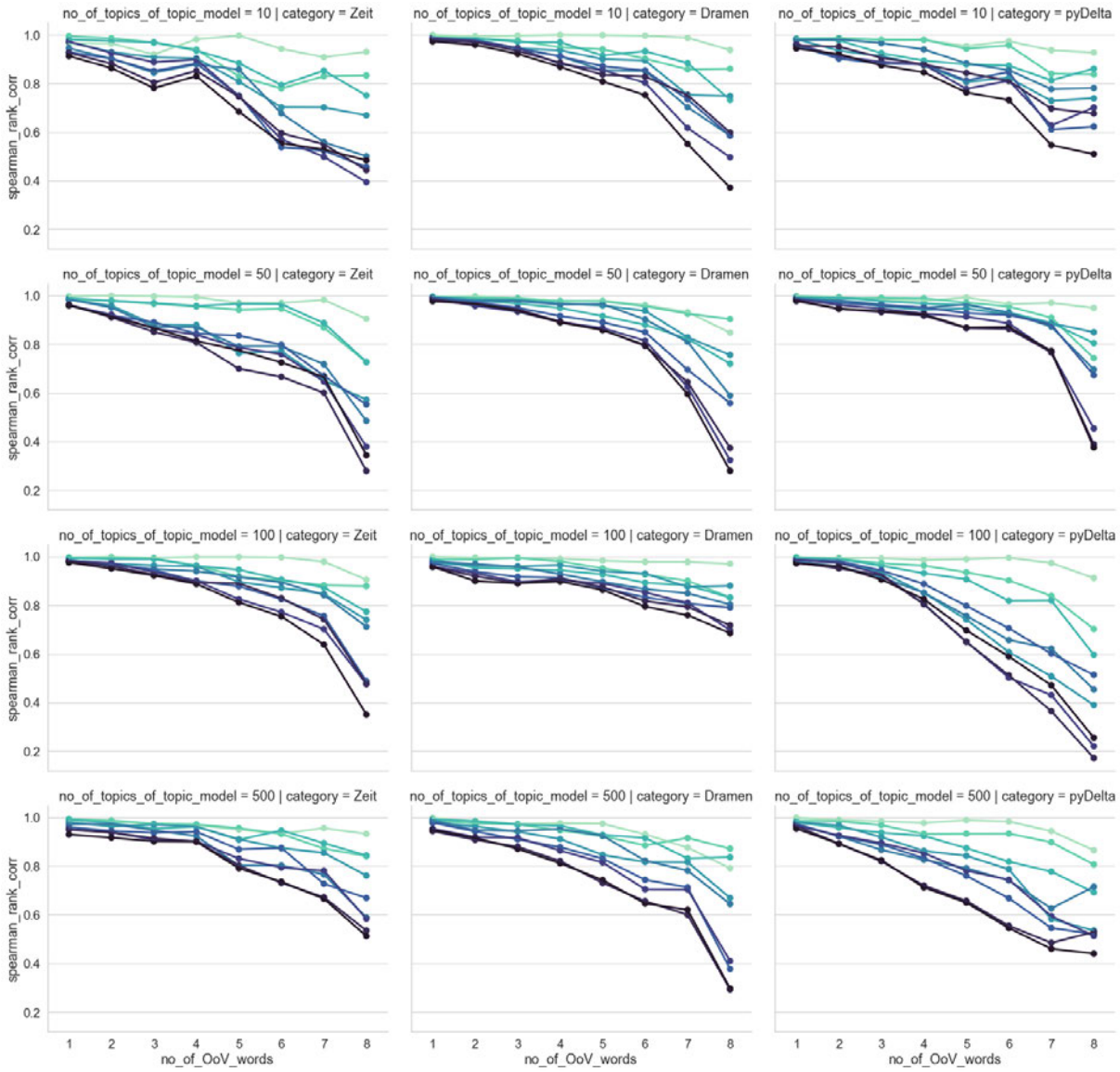

Abbildung 4.5 Entwicklung der NPMI-basierten Topic-Rangfolge beim Entfernen von OoV-Wörtern

Die Untersuchungsergebnisse werden in Abbildung 4.5 visualisiert. Jede Spalte in der Abbildung stellt die Ergebnisse eines Korpus dar. Vor links nach rechts abgebildet werden der Zeitungskorpus, der Dramakorpus und der Romankorpus. Jede Zeile in der Abbildung stellt die Ergebnisse für ein Setting der Anzahl der Topics $\boldsymbol{T}$ dar. Von oben nach unten angeordnet sind:

$\boldsymbol{T}$ = 10, $\boldsymbol{T}$ = 50, $\boldsymbol{T}$ = 100, $\boldsymbol{T}$ = 500[54]. Die Farbe der Linien in der Abbildung zeigt den Anteil der manipulierten Topics $\boldsymbol{P}$. Ein Verlauf von hell nach dunkel repräsentiert einen Anstieg des $\boldsymbol{P}$ von 10 % auf 90 %. Innerhalb jedes Liniendiagramms wiederum stellt die x-Achse die Anzahl der entfernten Topic-Wörter (von 1 bis 8) und die y-Achse die Rangkorrelation dar. Bezüglich des Ergebnisses der Untersuchung ist zuerst festzustellen, dass, unabhängig von der Anzahl der Topics und der Textart, die Korrelationen einen deutlichen absteigenden Trend zeigen. Dieser Trend hängt sowohl mit der Anzahl der entfernten Topic-Wörter als auch mit dem Anteil der manipulierten Topics zusammen: Je mehr Topic-Wörter aus Topics entfernt werden, desto niedriger ist die Rangkorrelation. Dies gilt auch, je mehr manipulierte Topics es gibt.

Aus diesem Experiment können folgende Schlussfolgerungen gezogen werden: Wenn die OoV-Wörter bei der Berechnung der Topic-Kohärenz entfernt werden, weicht der Kohärenz-Wert ab. Allerdings kann dieser abweichende Kohärenz-Wert die originale Rangfolge der Topics immer noch widerspiegeln, wenn nicht zu viele OoV-Wörter im Topic enthalten sind. Auch, wenn ein Topic-Modell problematische Topics enthält, die viele OoV-Wörter aufweisen, kann man der Rangfolge der Topics nach den abweichenden Kohärenz-Werten immer noch vertrauen, wenn das Topic-Modell nicht zu viele solcher problematischen Topics enthält. Zum Beispiel liegt die Rangfolgekorrelation in dieser Untersuchung häufig höher als 0,9, wenn 40 % der Topics bis zu fünf OoV-Wörter enthalten. Deshalb sollte bei der automatischen Evaluation der Topics zuerst überprüft werden, wie viele OoV-Wörter in den Topics eines Topic-Modells vorhanden sind. Wenn viele Topics mehrere OoV-Wörter enthalten, müssen die Menge der Textdaten im Referenzkorpus erhöht und die Textarten des Referenzkorpus vielfältiger gestaltet werden, um die Anzahl der OoV-Wörter zu reduzieren und eine zuverlässige Evaluation der Topics sicherzustellen. Dies gilt insbesondere im Feld der Digital Humanities. Da die hier in der Forschung verwendeten Texte sehr unterschiedlich sein können, kann die Auswahl des Referenzkorpus je nach dem sprachlichen Charakter des Untersuchungskorpus, dem Zweck der Evaluation oder anderen Faktoren variieren. So ist zum Beispiel dann, wenn ein Topic-Modell auf literarische Texte aus dem 19. Jahrhundert trainiert wird, Wikipedia offensichtlich für die Evaluation der Topics nicht mehr ausreichend geeignet.

Es gibt darüber hinaus eine andere Möglichkeit, um das OoV-Wörter-Problem zu vermeiden: die Verwendung des UMass-Kohärenzmaßes (Mimno et al., 2011). Wie bereits erläutert,

[54] Die Ergebnisse bei den anderen Settings der Topic-Anzahl $\boldsymbol{T}$ sind ähnlich und werden deshalb nicht visualisiert.

benötigt diese Methode für die Berechnung des Kohärenz-Wertes kein externes Referenzkorpus. Allerdings weist eine empirische Studie darauf hin, dass das UMass-Kohärenzmaß weniger mit der menschlichen Wahrnehmung der Interpretierbarkeit als vielmehr mit dem NPMI-Kohärenzmaß korreliert (Aletras & Stevenson 2013, Röder et al. 2015). Die Verwendung des UMass-Kohärenzmaßes kann zudem als widersprüchlich zur Zielstellung angesehen werden, den besten verfügbaren quantitativen Proxy für die Interpretierbarkeit der Topics zu verwenden.

4.3 Zwischenfazit

Zusammenfassend ist bei der Verwendung des Kohärenzmaßes aus mehreren Gründen Vorsicht geboten. Zunächst kann die Definition der Evaluationsaufgabe, nämlich die „Qualität“ oder „Interpretierbarkeit“ des Topics zu überprüfen, leicht unterschiedlich verstanden werden. Das hat zur Folge, dass Evaluationsmethoden entwickelt werden, die der menschlichen Bewertung des Topics entsprechen sollten, wobei diese doch wegen der möglichen Variabilität nicht genau quantifiziert werden kann. Die Ergebnisse dieser Evaluationsmethoden enthalten deshalb ebenfalls ein hohes Maß an Variabilität. Allerdings korrelieren die Ergebnisse der automatischen Topic-Evaluation trotz der oben vorgestellten Ungenauigkeiten doch positiv mit der menschlichen Topic-Bewertung. Bisher existiert keine perfekte Methode, um Topics automatisch zu evaluieren. Wahrscheinlich ist es auch nicht möglich, ein einheitliches Evaluationsverfahren für Topics zu entwickeln, weil die Evaluationsaufgaben aufgrund der jeweils unterschiedlichen Forschungsziele nicht immer identisch sind. Zurzeit jedenfalls ist die Nutzung eines Topic-Kohärenzmaßes die beste existierende Evaluationsmethode. In Aletras & Stevenson (2013), Röder et al. (2015) und Xing et al. (2019) werden verschiedene Topic-Kohärenzmaße getestet. Das NPMI-basierte Topic-Kohärenzmaß korreliert in allen drei Untersuchungen immer am zweitbesten mit den menschlichen Evaluationen. Im Vergleich dazu ist die Korrelation zwischen den anderen Methoden und den menschlichen Evaluationen nicht sehr stabil. Deshalb wird der NPMI-basierte Kohärenzwert des Topics bei den Untersuchungen in Kapitel 6 und Kapitel 7 als Indikator ihrer Qualität verwendet.

5. Korpora und Tools

Bevor die Untersuchungen und die Evaluationsergebnisse betrachtet werden, bietet dieses Kapitel einen Überblick über die zwei Untersuchungskorpora. Darüber hinaus wird kurz erläutert, welche Tools bzw. Computerprogramme für die Untersuchungen eingesetzt werden.

5.1 Korpora

5.1.1 Das Zeitungskorpus: eine Sammlung von 2000 Zeitungsartikeln

Für die Untersuchung wird ein Korpus von 2000 Texten aufgebaut. Es handelt sich dabei um deutsche Zeitungsartikel aus DIE ZEIT, die zwischen 2001 und 2014 publiziert wurden. Jeder Zeitungsartikel stellt hier ein Dokument dar. Die Artikel wurden vom Verlag in zehn thematischen Klassen eingeteilt, von denen jede 200 Dokumente enthält. Die einzelnen Klassen sind: „Digital“, „Gesellschaft“, „Karriere“, „Kultur“, „Lebensart“, „Politik“, „Reisen“, „Sport“, „Studium“ und „Wirtschaft“. Das Korpus enthält insgesamt über 3,4 Millionen Tokens, die durchschnittliche Textlänge beträgt knapp 1800 Wörter. Alle Texte sind darüber hinaus lemmatisiert für das Training des Topic-Modells. In Abbildung 5.1 wird genauer dargestellt, wie sich die Textlängen verteilen. Die meisten Dokumente enthalten ungefähr 1400 bis 2000 Lemmata.

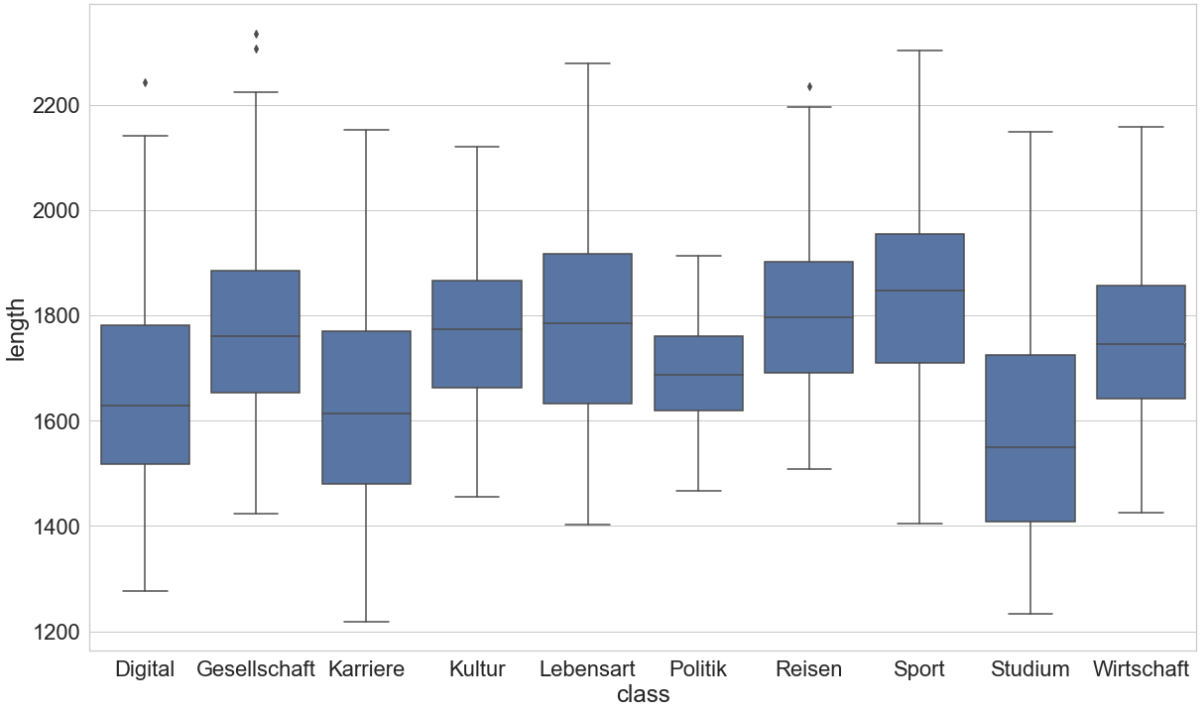

Abbildung 5.1 Verteilung der Textlänge in den zehn Klassen der Zeitungsartikel

Da das Topic-Modeling-basierte Klassifikationsergebnis als eines der Kriterien für die Evaluation herangezogen wird, wird zunächst eine Bag-of-Words (BoW) -basierte Klassifikation des Korpus durchgeführt, um eine Baseline der Klassifikation des Korpus zu definieren. Für die Beurteilung der Klassifikationsergebnisse kann neben Accuracy auch F1 (Makro) oder F1 (Mikro) verwendet werden. F1 (Makro) gibt bei einer Multiklassen-Klassifikation allen Klassen die gleiche Wichtigkeit, während F1 (Mikro) jedem Dokument die gleiche Wichtigkeit beimisst. F1 (Mikro) wird häufig verwendet, wenn es Ungleichgewichte in den Klassen gibt. Da alle Klassen im Testkorpus 200 Dokumente umfassen, wird für diese Analyse F1 (Makro) verwendet. Die Tests erfolgten als 10-fache Kreuzvalidierung mit linearer SVM [55] und naivem Bayes-Klassifikator. [56] Das Ergebnis zeigt, dass die lineare SVM-Klassifikation etwas besser funktioniert. Die durchschnittliche Accuracy und der F1 (Makro)-Wert betrugen jeweils zwischen 0,72 und 0,77 bzw. lag zwischen 0,72 und 0,76 (Tabelle 5.1 und Tabelle 5.2).

	Linear Support Vector Classification	*Multinomial Naive Bayes*
BoW	0,753	0,757
tf-idf gewichtete BoW	**0,765**	0,727

Tabelle 5.1 Accuracy der BoW-basierten Klassifikation

	Linear Support Vector Classification	*Multinomial Naive Bayes*
BoW	0,747	0,753
tf-idf gewichtete BoW	**0,758**	0,723

Tabelle 5.2 F1 (Makro)-Wert der BoW-basierten Klassifikation

Die Konfusionsmatrix (auch Wahrheitsmatrix genannt) in Abbildung 5.2 bietet zusätzliche Informationen über das Klassifikationsergebnis. Alle möglichen Kombinationen von vorhergesagten Klassen und wahren Klassen werden in der Matrix eingetragen. Die diagonalen

[55] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html, (19.09.2020).
[56] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html, (19.09.2020).

Matrixwerte sind die korrekt vorhergesagten Dokumente, während die falsch klassifizierten Dokumente in den übrigen Zellen der Matrix liegen. Aus der Konfusionsmatrix lässt sich ableiten, dass die Erkennung von „Lebensart“-Dokumenten für Computer besonders schwer ist. Nur 37 % der Dokumente können hier richtig klassifiziert werden, während mehr als 50 % der „Lebensart“-Dokumente fälschlicherweise „Gesellschaft“, „Karriere“, „Kultur“, „Reisen“ und „Sport“ zugeordnet werden. Vermutlich liegt der Grund darin, dass diese Dokumente eine große Überlappung von Inhaltswörtern haben. Außerdem werden mehr als ein Viertel der „Gesellschaft“-, „Karriere“- und „Wirtschaft“-Dokumente falsch klassifiziert. Die Erkennung von „Digital“, „Reisen“ und „Sport“ erreicht dagegen ca. 90 %, wahrscheinlich, weil hier domänenspezifische Inhaltswörter vorkommen. Vermutlich wird die Dokument-Klassifikation durch Topic Modeling ähnliche Ergebnisse erbringen, weil die Unterschiede zwischen den Artikeln hauptsächlich auf der Ebene des Themas liegen.

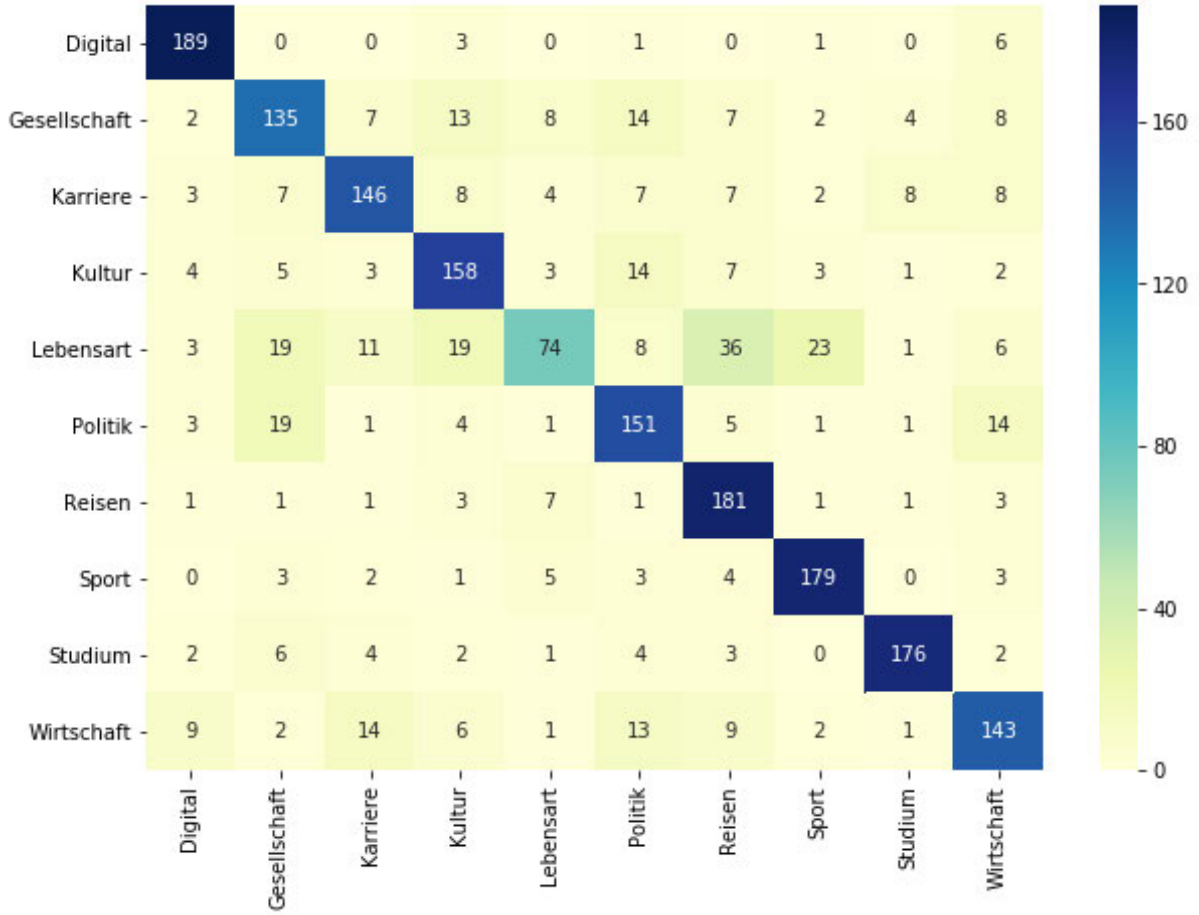

Abbildung 5.2 Konfusionsmatrix der SVM-Klassifikation des Zeitungskorpus

5.1.2 Der Romankorpus: eine Sammlung von 439 Heftromanen

Neben der Sammlung von Zeitungsartikeln wird ein zweites Korpus von literarischen Texten aufgebaut und analysiert. Das Korpus besteht aus 439 deutschen Heftromanen aus den Jahren 1961 und 2016. Sie teilen sich in fünf thematischen Klassen/Untergattungen auf, nämlich 100 Fantasyromane (fantasy), 51 Horrorromane (horror), 88 Kriminalromane (krimi), 100 Liebesromane (liebes) und 100 Science-Fiction (scifi). Das Korpus enthält insgesamt ca. 13,4 Millionen Wörter, die durchschnittliche Textlänge ist ca. 30.000 Wörter. In Abbildung 5.3 wird genauer dargestellt, wie sich die Textlänge der Romane verteilt. Die meisten Liebesromane enthalten mehr als 35.000 Wörter, die meisten Romane aus den anderen vier Gattungen etwas weniger (zwischen 25.000 und 35.000 Tokens).

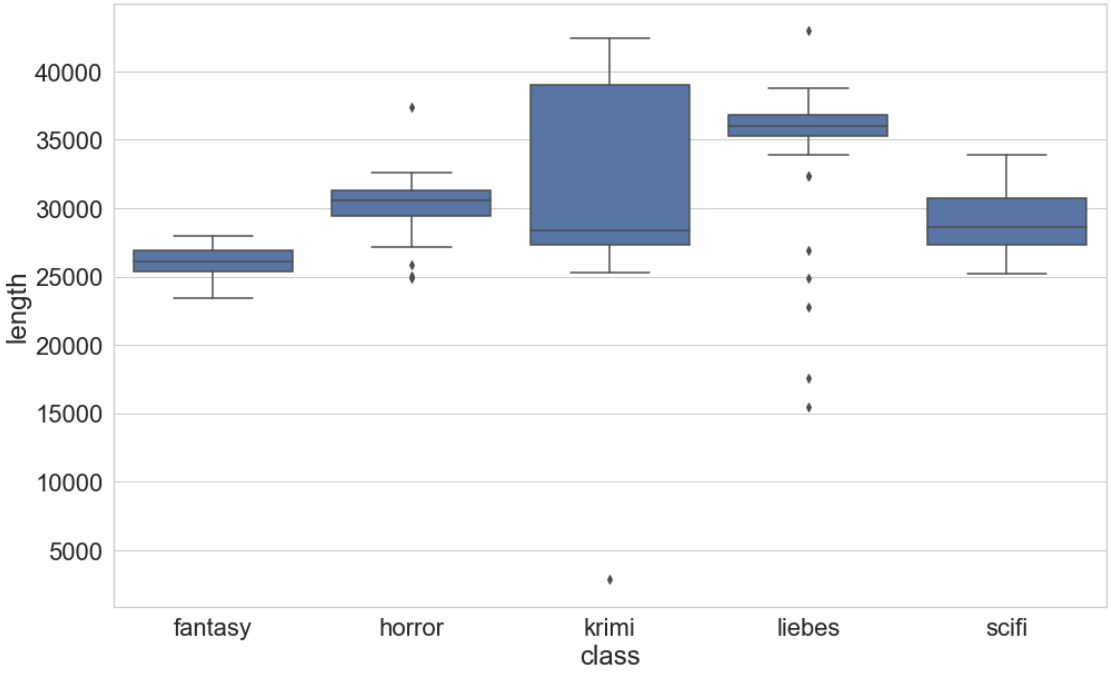

Abbildung 5.3 Verteilung der Textlängen in den vier Klassen der Heftromane

Auch hier wird eine Bag-of-Words (BoW) -basierte Klassifikation durchgeführt, um eine Baseline der Klassifikation zu definieren. Die Tests erfolgten als 10-fache Kreuzvalidierung mit linearer SVM und naivem Bayes-Klassifikator. Die Accuracy und der F1 (Makro)-Wert können beide 1,0 erreichen (Tabelle 5.3 und Tabelle 5.4). Hier ist das Ergebnis der linearen SVM-Klassifikation besser. Besonders interessant ist es zu sehen, dass die Klassifikation durch den naiven Bayes-Klassifikator deutlich schlechter funktioniert, wenn die Tf-idf-Gewichtung

auf der Dokument-Wort-Matrix eingesetzt wird. Da die Klassifikation perfekt funktioniert, wird keine Konfusionsmatrix berechnet bzw. visualisiert.

	Linear Support Vector Classification	*Multinomial Naive Bayes*
BoW	**1,0**	**1,0**
tf-idf gewichtete BoW	**1,0**	0,877

Tabelle 5.3 Accuracy der BoW-basierten Klassifikation

	Linear Support Vector Classification	*Multinomial Naive Bayes*
BoW	**1,0**	**1,0**
tf-idf gewichtete BoW	**1,0**	0,754

Tabelle 5.4 F1 (Makro)-Wert der BoW-basierten Klassifikation

Wie bereits erwähnt, wenn Topic-Modelle auf langen Texten trainiert werden, sollten diese in Chunks zerlegt werden.[57] Aus diesem Grund werden die Romane für die Untersuchungen im Kapitel 6 in Chunks zerlegt. Da die durchschnittliche Dokumentlänge der Zeitungsartikel 1800 Wörter ist, wird die Chunk-Length auf diesen Wert eingestellt. Dadurch ist die Chunk-Length kein Störfaktor mehr, wenn die Untersuchungsergebnisse aus dem Zeitungskorpus in Kapitel 5 und die Untersuchungsergebnisse aus den Romankorpus in Kapitel 6 verglichen werden.

Eine Bag-of-Words (BoW) -basierte Klassifikation wird deshalb in Bezug auf die zerlegten Chunks durchgeführt, um eine Baseline der Klassifikation zu definieren. Die Tests erfolgten als 10-fache Kreuzvalidierung mit linearer SVM und naivem Bayes-Klassifikator. Die Accuracy und der F1 (Makro)-Wert erreichen jeweils höchstens 0,993 und 0,992 (Tabelle 5.5 und Tabelle 5.6). Das Ergebnis zeigt, dass die lineare SVM-Klassifikation etwas besser funktioniert, wenn die Tf-idf-Gewichtung auf der Dokument-Wort-Matrix eingesetzt wird. Ohne die Tf-idf-Gewichtung wiederum funktioniert der naive Bayes-Klassifikator besser. Die Konfusionsmatrix in Abbildung 5.4 liefert weitere Informationen: 18 Fantasy-Chunks werden als Scifi klassifiziert, während 27 Scifi-Chunks der Klasse Fantasy zugeordnet werden. Außerdem lässt sich beobachten, dass die Klassifikation vor allem bei einer kleinen Menge von

[57] Siehe Kapitel 2.2.3 Faktoren, die LDA Topic Modeling beeinflussen können – Chunk-Length.

Horror-, Krimi- und Scifi-Chunks die Gattungen durcheinandergebracht hat. Die Ergebnisse dieser Klassifikation entsprechen den Erwartungen, wenn man die möglichen Ähnlichkeiten oder Überschneidungen in den Handlungen dieser Untergattungen bedenkt. Insgesamt ist der Unterschied zwischen den Texten aus diesen fünf Untergattungen für den Computer klar erkennbar. Es ist zu erwarten, dass die Topic-Modeling-basierte Klassifikation auf dieser literarischen Textsammlung ähnlich gut funktioniert, weil die Unterschiede zwischen den Untergattungen hauptsächlich auf der Ebene des Themas liegen.

	Linear Support Vector Classification	*Multinomial Naive Bayes*
BoW	0,979	**0,993**
tf-idf gewichtete BoW	**0,993**	0,867

Tabelle 5.5 Accuracy der BoW-basierten Klassifikation

	Linear Support Vector Classification	*Multinomial Naive Bayes*
BoW	0,977	0,991
tf-idf gewichtete BoW	**0,992**	0,757

Tabelle 5.6 F1(Makro)-Wert der BoW-basierten Klassifikation

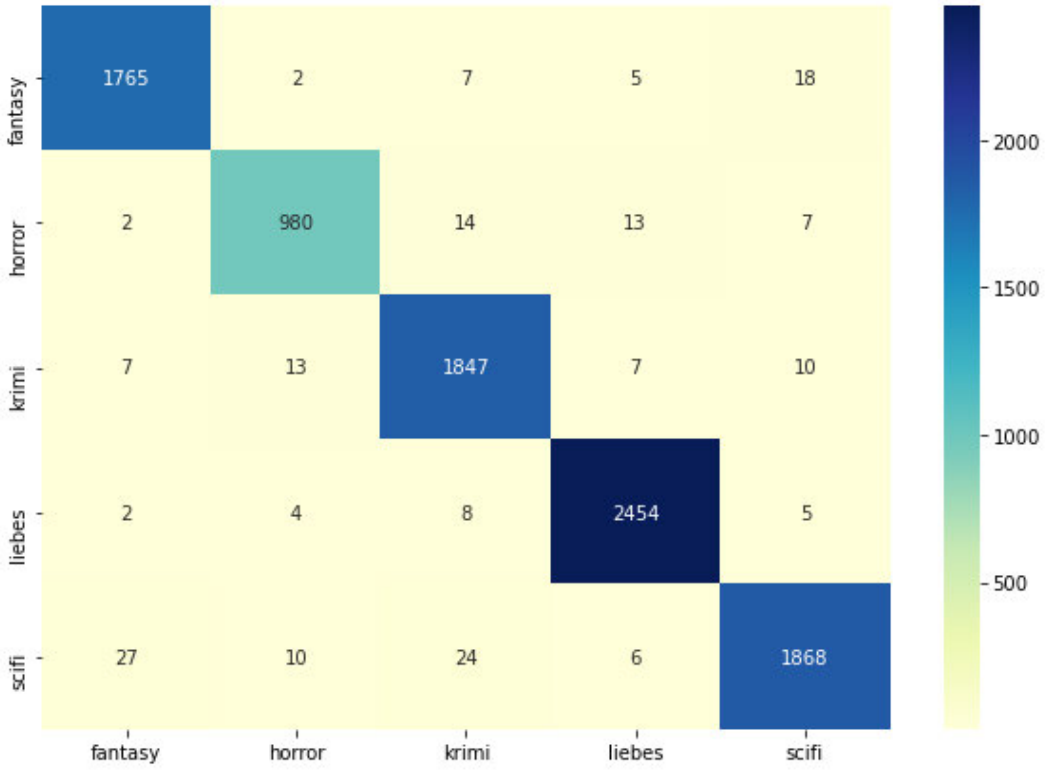

Abbildung 5.4 Konfusionsmatrix der SVM-Klassifikation des Romankorpus bei Chunks

5.2 Tools

In dieser Arbeit werden die meisten Aufgaben der computerstützten Datenanalyse (wie z. B. Chunking des Textes, Dokumentklassifikation oder Visualisierung der Evaluationsergebnisse) durch vom Autor selbst geschriebene Python-Skripte erledigt. Darüber hinaus werden drei Programme eingesetzt, nämlich MALLET für die Durchführung des Topic Modeling, Palmetto für die Berechnung der Topic-Kohärenz und Treetagger für die Lemmatisierung des Textes.

5.2.1 MALLET für Topic Modeling

Das Topic Modeling wird mit dem Java-basierten Programm MALLET[58] durchgeführt. Die Durchführung besteht aus zwei Schritten: die Texte werden importiert und die Topics werden trainiert. Für das Importieren der Textdateien soll folgende Anweisung in der Kommandozeile eingegeben werden:

> **mallet import-dir --input <Korpus-Pfad> --output <Output-Datei> --keep-sequence --token-regex "[\p{L}]+" --remove-stopwords --stoplist-file <Stoppwörter-Datei>**

Nach dem **„--output“** muss der Name der Output-Datei eingegeben werden. Hier muss darauf geachtet werden, dass die Dateinamenerweiterung der Output-Datei **„.mallet“** ist. Damit kann diese Datei im nächsten Schritt beim Training der Topics eingelesen werden. Die Stoppwortliste wird beim Importieren der Dateien durch die Anweisung **„--remove-stopwords --stoplist-file <Stoppwörter-Datei>“** hinzugefügt. Nach **„--stoplist-file“** wird der Pfad eingegeben. Wenn ein Korpus aus deutschen Texten besteht, muss zusätzlich die Anweisung **„--token-regex "[\p{L}]+"“** hinzufügt werden. Ohne diese Anweisung kann MALLET die deutschen Zeichen „ö“, „ä“, „ü“ und „ß“ nicht einlesen. Wie in Abbildung 5.5 zu sehen ist, enthält das Ergebnis des Topic Modeling dann zahlreiche Fehler. Die in rot markierten Zeichenketten im Ergebnis sind keine vollständigen deutschen Wörter, wie z.B. „gew“, „rte“, „verh“ oder „hrte“.

[58] http://mallet.cs.umass.edu/topics.php, (21.08.2019).

3	2,5	zu von ein mit als n te ohne linovsky t wieder g st welche konnte alexandern aller seite wirklich
4	2,5	das h nur eine leben l auf als um wieder ja hl rte verh indem leicht wollen allen innern
5	2,5	und durch war von den des gew unter nen es hatten kaum zwar stillen kr erinnerung vers freuden strebte
6	2,5	ich zu sie mir mich alexander ihnen r mein hat meiner nicht meine du meines soll dich darf kann
7	2,5	der wie da vor einem dieser einen diese sehen zum werde m vielleicht bis sah z her sprach trat
8	2,5	der die in sich einer auch oder gef sondern en ganz gro deren machte ganze nahm jede eigenen unwillk
9	2,5	und des in aber z auge e nden dem sanft cken dessen gott wieder hrend ver herzens herab kaiser
10	2,5	zu das sie den wie mit so ihr liebe es wu etwas um te art herz willfried u furcht
11	2,5	so da nicht sie auch r sehr bei dies rde doch wenig re sinn weil ernst gern tante nur
12	2,5	ihm in es seinen an vor wohl dann gegen ob leise selbst einmal sonst hrte und blick nicht irgend
13	2,5	der die eine gr ein sch fin bl im ge entgegen viel ersten seine blumen zu andere jeden kind
14	2,5	den auf mit aus an dem zur so aber d durch fand wo denen eben t kraft alles himmel
15	2,5	und f nicht jetzt alexander r wurde keine tief fl diesem gab he l pl ne zwischen weit dem
16	2,5	die der wenn so gl zur ck gen gem wie th lebens welt tr oft immer bald diese denn
17	2,5	die als noch k nicht ihrem mehr da nun habe was alle einen sagte nach daher zum hoffnung recht
18	2,5	sie erna dem s des seele aber schien am konnte mu worte anblick bereits auguste ersch the freundin einige
19	2,5	und ist fr m man es seyn denn uns von wir nnen hier werden sind lassen nichts gewi sollte

Abbildung 5.5 Ergebnis des Topic Modeling

Nach dem Importieren der Textdaten kann ein Topic-Modell mit der folgenden Anweisung trainiert werden:

mallet train-topics --input <Mallet-Datei> --num-topics <Topic-Anzahl> --output-topic-keys <Topic-Datei> --output-doc-topics <Topic-Dokument-Datei> --num-top-words <Topic-Wörter-Anzahl>

Die Anweisung **„--num-topics“** fordert die Anzahl der Topics an. MALLET wird durch die Anweisung daraufhin konfiguriert, wie viele Topics zu trainieren sind. Die weiteren Anweisungen dienen der Erstellung des Ergebnisses. Die Anweisung **„--output-topic-keys <Topic-Datei>“** gibt die Topics aus. Die Ausgangsdatei enthält eine Reihe von Wortlisten und die passenden Topic-ID. In jeder Zeile steht eine Liste / ein Topic. Die Anweisung „**--num-top-words**“ stellt die Anzahl der Topic-Wörter fest, die ein Topic repräsentieren. Durch die Anweisung „**--output-doc-topics <Topic-Dokument-Datei>**“ wird die Topic-Dokument-Verteilung als eine Matrix gespeichert.

5.2.2 Palmetto für die Berechnung der Topic-Kohärenz

Für die automatische Berechnung der Topic-Kohärenz wird das Java-basierte Programm Palmetto[59] verwendet. Außerdem ist ein Referenzkorpus notwendig. Der Referenzkorpus wird indiziert, die Kookkurrenz-Informationen werden in einem Lucene-Index [60] gespeichert. Danach können die Berechnung durch die folgende Anweisung im Kommandozeilen durchgeführt werden:

java -jar palmetto-0.1.0-jar-with-dependencies.jar <Lucene-Index-Pfad> <Kohärenzmaß> <Topic-Datei-Pfad>

In dem Programm werden sechs Kohärenzmaße implementiert.[61] Die Topic-Datei kann mehrere Topics enthalten, jedes Topic umfasst eine Zeile. In jeder Zeile werden die Top-Wörter des Topics aufgelistet und durch Leerzeichen getrennt.[62] In der Topic-Datei (Input-Datei) müssen die Topics folgendermaßen aussehen:

internet, geben, online, neu, facebook, datum, netz, information, google, nutzen
sagen, universität, geben, hochschule, deutsch, schule, studium, uni, arbeiten, studieren
land, deutsch, geben, kirche, finden, welt, staat, krieg, neu, westen

Der vorgegebene Lucene-Index des Programms basiert auf der englischsprachigen Wikipedia. Der Anleitung des Programms[63] folgend, wurde das deutschsprachige Wikipedia indiziert und ein deutscher Index aufgebaut, um die Topic-Kohärenz der deutschen Topics zu berechnen. Für die Indizierung wurde jeder Wikipedia-Artikel als ein Dokument betrachtet. Die Texte wurden zudem alle lemmatisiert, weil dies auch bei den Texten für das Topic Modeling der Fall ist. In Tabelle 5.7 sind sechs Beispieltopics mit ihren NPMI-Werten dargestellt.

NPMI-Wert	*Top 10 Wörter des Topics*
0.37235	the, of, and, it, is, for, we, you, that, on
0.28507	flugzeug, passagier, flughafen, fliegen, maschine, air, flug, airlines, fluggesellschaft, lufthansa

[59] http://aksw.org/Projects/Palmetto.html, (21.08.2019).
[60] https://lucene.apache.org/core/, (21.08.2019).
[61] https://github.com/dice-group/Palmetto/wiki/Coherences, (21.08.2019).
[62] Vgl. „How Palmetto can be used“, unter: https://github.com/dice-group/Palmetto/wiki/How-Palmetto-can-be-used, (21.08.2019).
[63] „How to create a new index“, unter: https://github.com/dice-group/Palmetto/wiki/How-to-create-a-new-index, (21.08.2019).

0.19565	hochschule, universität, uni, studium, studieren, studierende, semester, studiengang, fach, bachelor
-0.3031	peng, party, jimmy, obertilliach, ruhpolding, arena, biathlon, sparkasse, camp, treffer
-0.28392	fahrrad, kind, new, nase, eisenhüttenstadt, book, dach, raum, online, lust
-0.27796	mark, campus, pfennig, ikea, schokolade, redakteurin, stempel, farbe, malen, verdienen

Tabelle 5.7 Beispieltopics und ihre NPMI-Werte

Interessant ist, dass das Fremdwörter-Topic „the, of, and, it, is, for, we, you, that, on“ einen hohen NPMI-Wert bekam, obwohl es nicht als Thema interpretiert werden kann. Da das Referenzkorpus für die Berechnung der Topic-Kohärenz das deutschsprachige Wikipedia ist, tauchen die englischen Wörter im Referenzkorpus selbstverständlich viel häufiger zusammen auf als die deutschen Wörter. Es ist deshalb wenig erstaunlich, dass das Topic einen hohen Wert erhält. Wenn das Referenzkorpus das englischsprachige Wikipedia ist, beträgt der NPMI-Wert dieses Topics -0.04640. Dies zeigt, dass die Auswahl des Referenzkorpus die automatische Berechnung der Topic-Kohärenz stark beeinflussen kann.

5.2.3 TreeTagger und Python-Treetaggerwrapper

In Kapitel 2.3 wurde bereits dargelegt, dass die Lemmatisierung des Textes oft vor dem Topic Modeling eingesetzt wird, um die unterschiedlichen Flexionsformen eines Wortes auf eine Grundform zu reduzieren. Dadurch können sich die Topics besser auf die semantische Ebene des Wortes konzentrieren. Für die Lemmatisierung werden der TreeTagger (Schmid 2013)[64] und der Python-Treetaggerwrapper[65] eingesetzt. Der TreeTagger nimmt Plaintext als Input und führt POS-Tagging und Lemmatisierung des Textes gleichzeitig durch. Der Output enthält drei Spalten (Tabelle 5.8): Das (Original)-Wort, die POS-Annotation und das Lemma des Wortes. Die Lemmata in der dritten Spalte werden dann verwendet, um die lemmatisierten Texte zu rekonstruieren und Topic-Modelle zu trainieren.

Wort	**POS**	**Lemma**

[64] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/, (21.08.2019).
[65] https://treetaggerwrapper.readthedocs.io/en/latest/, (25.08.2020).

Fremden	NN	Fremde
und	KON	Und
Neuzugezogenen	NN	Neuzugezogene
macht	VVFIN	machen
das	ART	Die
Internet	NN	Internet
oft	ADV	Oft
Angst	NN	Angst
.	$.	.

Tabelle 5.8 Beispiel des TreeTagger-Outputs

Mit den entsprechenden Parameter-Dateien kann der TreeTagger für das POS-Tagging und die Lemmatisierung von mehr als 20 Sprachen verwendet werden. Allerdings müssen die Installation und die Durchführung des Programms über die Kommandozeile ausgeführt werden. Um die Verwendung des Programms zu vereinfachen, wurden unterschiedliche Lösungen entwickelt wie z. B. grafische Benutzeroberflächen[66], der R-Wrapper[67] oder der Python-Treetaggerwrapper. In dieser Arbeit wird der Python-Treetaggerwrapper eingesetzt. Ein deutschsprachiger Treetaggerwarpper kann durch folgende Anweisung erstellt werden:

tagger = treetaggerwrapper.TreeTagger(TAGLANG='de')

Anschließend können das POS-Tagging und die Lemmatisierung eines Textes durch die Funktion „**tagger.tag_text()**“ durchgeführt werden. Der Output der Funktion stellt sich wie die Ergebnisse in Tabelle 5.8 dar. Anschließend können die lemmatisierten Texte rekonstruiert werden, um Topic-Modelle zu trainieren.

An dieser Stelle sind alle notwendigen Vorbereitungen abgeschlossen, und die eigentliche systematische Evaluation von Topic Modeling kann nun beginnen.

[66] https://www3.smo.uhi.ac.uk/oduibhin/oideasra/interfaces/winttinterface.htm, (25.08.2020).
[67] https://github.com/daniel-jach/treetag-fertilizer, (25.08.2020).

6. Evaluation von Topic Modeling in den Digital Humanities am Beispiel eines Zeitungskorpus

Im Bereich der Digital Humanities wird Topic Modeling eingesetzt, um die Unterschiedlichkeit (oder die Ähnlichkeit) von Dokumenten zu identifizieren. Dies wird häufig bei der Textklassifikation genutzt. Grundsätzlich ist hier die Qualität der Topics ein wichtiger Aspekt, bilden diese doch den Kern des Topic Modeling. In der Praxis interagieren bei Untersuchungen in Digital Humanities die menschlichen Nutzer direkt mit Topics, um große Textsammlung zu explorieren. Für die Quantifizierung der Interpretierbarkeit von Topics wird die Topic-Kohärenz eingeführt. Je höher die Topic-Kohärenz eines Topics ist, desto besser ist dieses interpretierbar. Das Ziel der Experimente in diesem Kapitel ist es, die Auswirkung verschiedener Faktoren auf die Qualität des Modells zu evaluieren. Die Topic-Modeling-basierte Dokumentklassifikation und die Topic-Kohärenzwerte werden als Methoden zur Darstellung der Qualität des Modells eingesetzt.

In den folgenden Unterkapiteln wird jeweils die Untersuchung eines Faktors vorgestellt. Bei der Analyse wird der Wert des untersuchenden Faktors variiert, während alle anderen Faktoren Kontrollvariablen sind und unverändert gehalten werden. Neben den Experimenten in Bezug auf die Chunk-Length (Kapitel 6.6) wird beim Training des Topic-Modells jeder Zeitungsartikel als ein Dokument betrachtet. Das Korpus der Untersuchung besteht aus 2000 Zeitungsartikeln mit zehn thematischen Klassen. Eine genauere Beschreibung des Korpus findet sich in Kapitel 5.1.1. Wenn keine zusätzlichen Hinweise vorhanden sind, werden die Kontrollvariablen folgendermaßen eingestellt: Iteration des Gibbs-Samplings $\boldsymbol{I}$ = 2000, Hyperparameter Alpha jedes Topics $\boldsymbol{\alpha}$ = 0,05, Hyperparameter Beta $\boldsymbol{\beta}$ = 0,01. Die Stoppwörter werden vor dem Training des Modells aus dem Korpus entfernt, die Modelle werden ohne Hyperparameter-Optimierung trainiert. Die Dokumentklassifikation mit linearer SVM erfolgt gemäß einer 10-fachen Kreuzvalidierung, zudem wird das NPMI-basierte Kohärenzmaß (C_{NPMI}) für die Evaluation der Topics verwendet. Das C_{NPMI} vergleicht zunächst paarweise die Kohärenz der Top-10 Topic-Wörter. Der durchschnittliche Kohärenz-Wert aller Wortpaare bildet dann den Kohärenz-Wert des Topics.

Darüber hinaus wird ein NPMI-Kontrollwert als eine Art von Baseline festgelegt, der die Topic-Kohärenz der „Topics vor Topic Modeling“ repräsentiert. 18 Topic-Modelle werden zunächst mit nur einer Iteration auf die 2000 Zeitungsartikel trainiert, die jeweils $\boldsymbol{T} \in \{10, 20, 30, 40, 50,$

60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500} Topics enthalten. Die NPMI-Werte der Top-10 Wörter aller diesen 3150 Topics werden berechnet, der Durchschnittswert ist dann der Kontrollwert: -0,0619. Der NPMI-Kontrollwert wird in den folgenden Visualisierungen durch eine schwarze Linie dargestellt.

Aus technischen Gründen (die zufällige Initialisierung bei der Zuweisung von Topics und das Gibbs-Sampling) sind zwei Topic-Modelle eines Korpus nicht vollkommen identisch, selbst wenn alle Parameter beim Training der Modelle gleich eingestellt werden. Um den möglichen Unterschied zwischen den Modellen mit gleicher Parametereinstellung sichtbar darzustellen, werden zehn Modelle für jedes Parametersetting trainiert.

In den nächsten Kapiteln werden Untersuchungen in Bezug auf die Anzahl der Topics, den Hyperparameter Alpha, die Hyperparameter Optimierung, den Hyperparameter Beta, die Anzahl der Iterationen des Gibbs-Samplings und die Chunk-Length vorgestellt.

6.1 Anzahl der Topics

In diesem Kapitel bezieht sich das Experiment auf den Einfluss der Topic-Anzahl $\boldsymbol{T}$ auf die Qualität des Modells. Das Setting der Anzahl der Topics ist $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$.

6.1.1 Dokumentklassifikation

In Abbildung 6.1 werden die Topic-Modeling-basierten Klassifikationsergebnisse visualisiert. Es ist zu beobachten, dass die Klassifikationen mit der Erhöhung von $\boldsymbol{T}$ besser funktionieren. Wenn $\boldsymbol{T}$ größer als 80 eingestellt wird, ist allerdings keine deutliche Verbesserung der Klassifikationsergebnisse mehr zu erkennen. Ein weiteres interessantes Phänomen ist die Tatsache, dass sich eine große Abweichung in dieser Untersuchung beobachten lässt, wenn $\boldsymbol{T}$ auf 10 eingestellt wird. Es ist festzustellen, dass die zufällige Initialisierung und das Gibbs-Sampling größere Auswirkung auf die Qualität des Modells haben, wenn es mit weniger Topics trainiert wird. In dieser Untersuchung erzielt die Klassifikation im besten Fall eine Accurarcy von 0,761 und einen F1-Wert von 0,753. Ein besseres Ergebnis als die BoW-basierte Baseline (Accurarcy von 0,765 / F1-Wert von 0,758) wird nicht beobachtet.

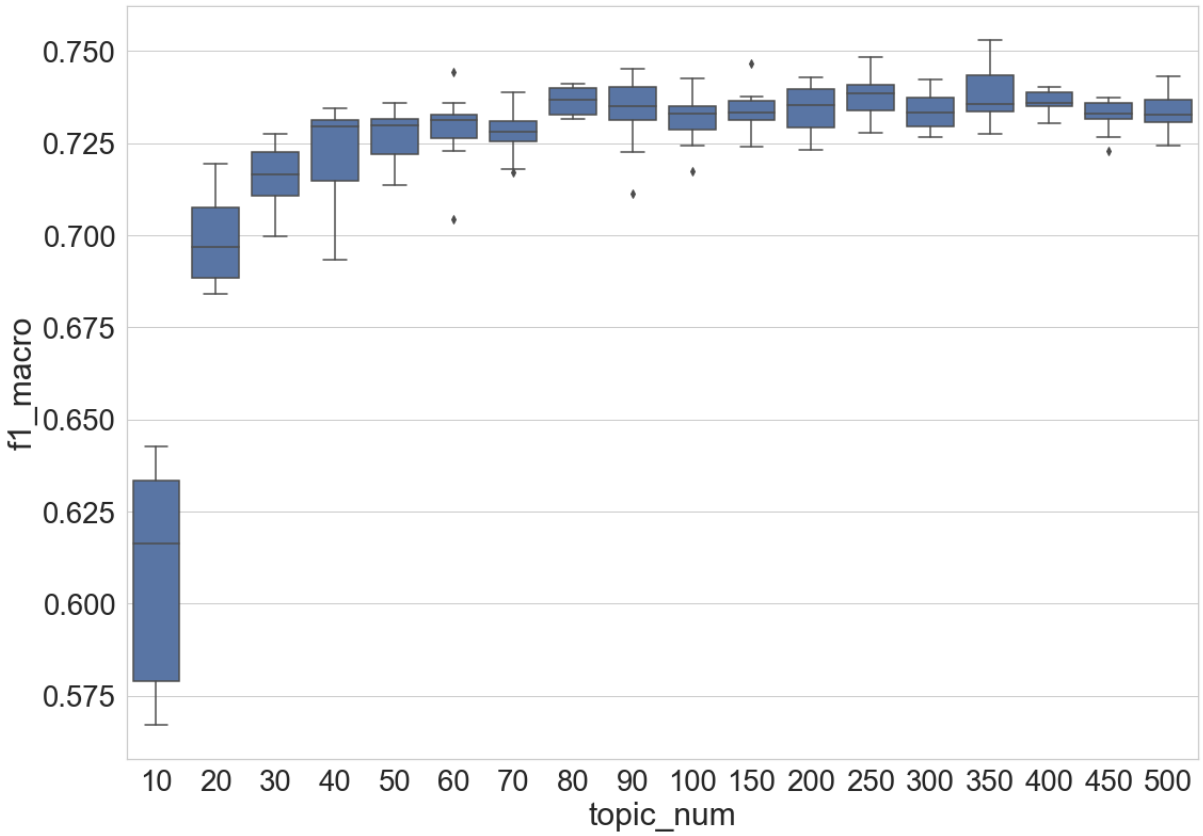

Abbildung 6.1 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zur Anzahl der Topics

6.1.2 Topic-Kohärenz

Abbildung 6.2 zeigt die Verteilungen der Kohärenz-Werte der Topics im Verhältnis zu ihrer Anzahl. Es ist zu beobachten, dass der Median der NPMI-Werte mit der Erhöhung von ***T*** sinkt. Der Wertebereich der mittleren 50 % der Daten (der Wertebereich zwischen dem 0,25- und 0,75-Quartil) und die Spannweite (gesamter Wertebereich des Datensatzes) sind beide breiter geworden, wenn ***T*** von 10 auf 90 steigt. Die mittleren 50 % der NPMI-Werte sinken mit der weiteren Erhöhung von ***T***, ab ***T*** = 350 zeigt die NPMI-Verteilung keine deutlichen Änderungen mehr. Die mittleren 50 % der NPMI-Werte liegen zwischen der Baseline (-0.0619) und einem Wert von ca. -0.15. In diesem Test ist zu erkennen, dass das trainierte Topic-Modell einen zunehmenden Anteil an nicht-kohärenten Topics enthält, wenn die Anzahl der Topics 90 übersteigt.

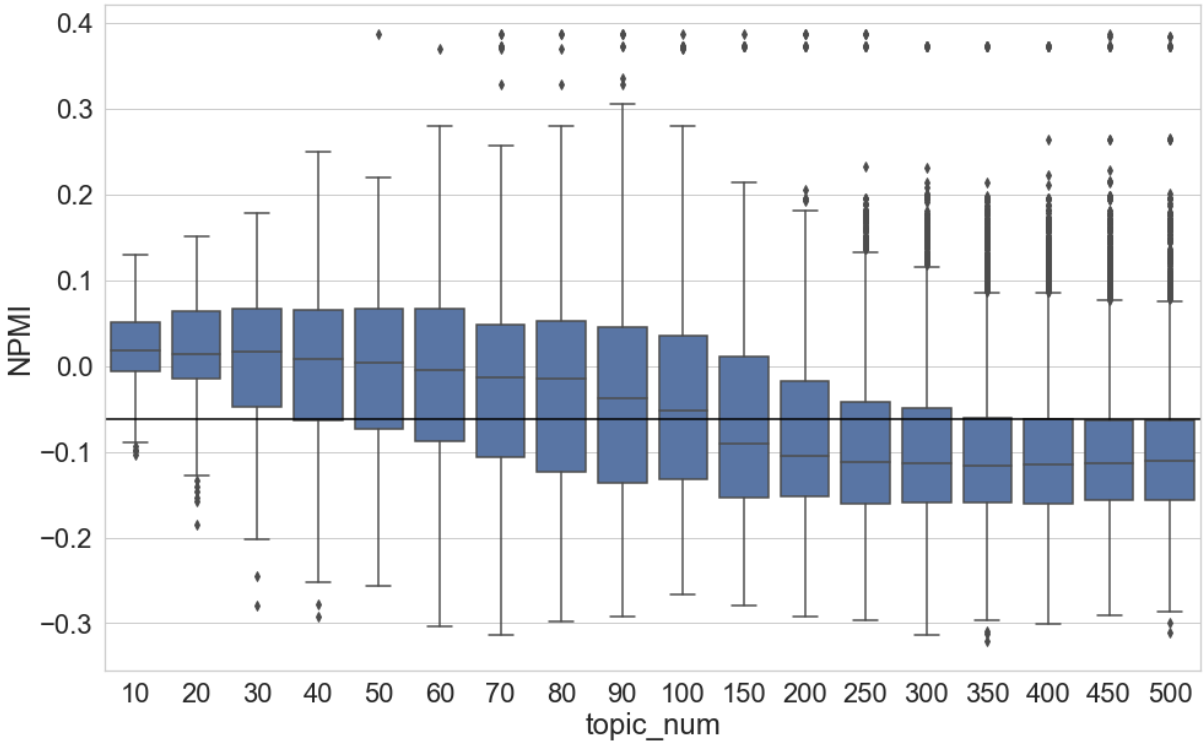

Abbildung 6.2 NPMI-Werte-Verteilung der Topics im Verhältnis zu ihrer Anzahl

Wie bereits erläutert, werden für jedes Setting zehn Topic-Modelle trainiert. Da die Anzahl der Topics bei unterschiedlichen Settings ungleich ist, werden in Abbildung 6.2 von links nach rechts 10 * ***T***, also 100 bis 5000 Datenpunkte visualisiert. Um sicherzustellen, dass die beobachteten Veränderungen nicht durch eine ungleiche Anzahl von Datenpunkten verursacht werden, wird folgende Untersuchung durchgeführt: Für jedes Setting in $\boldsymbol{T} \in \{10, 100, 500\}$ werden jeweils 500, 50 und 10 Topic-Modelle trainiert. Die NPMI-Werte der Topics werden dann berechnet und visualisiert. In Abbildung 6.3 enthält deshalb jedes Boxplot 5000 Datenpunkte/NPMI-Werte, die Verteilungen der Daten ist dabei fast identisch mit der in Abbildung 6.2: Der Median und die mittleren 50 % der NPMI-Werte sinken mit der Erhöhung von ***T*** ab.

In Abbildung 6.2 lässt sich zudem erkennen, dass es zunehmend mehr Ausreißer gibt. Es handelt sich hier vor allem um Topics, die einen hohen NPMI-Wert haben. In Abbildung 6.4 (links) wird die Situation besonders deutlich dargestellt: Die absolute Anzahl der Topics, deren NPMI-Wert größer als der NPMI-Kontrollwert ist, steigt mit der Erhöhung von ***T*** von weniger als 200 auf mehr als 1200 auf. Wenn die absolute Anzahl normalisiert wird (geteilt durch die

Anzahl der Topics), ist umgekehrt eine absteigende Tendenz deutlich zu erkennen (Abbildung 6.4, rechts). Der Prozentwert ist von über 90 auf weniger als 30 % gesunken.

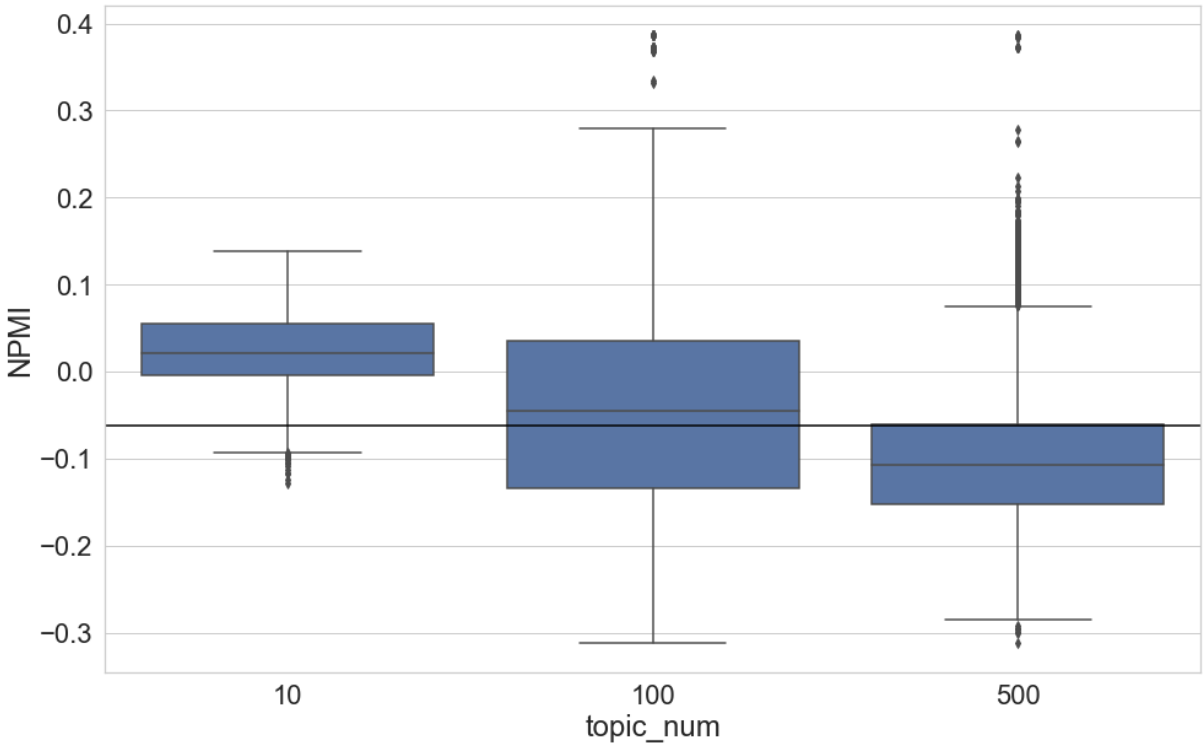

Abbildung 6.3 NPMI-Werte-Verteilung von drei Gruppen, die jeweils 5000 Topics enthalten

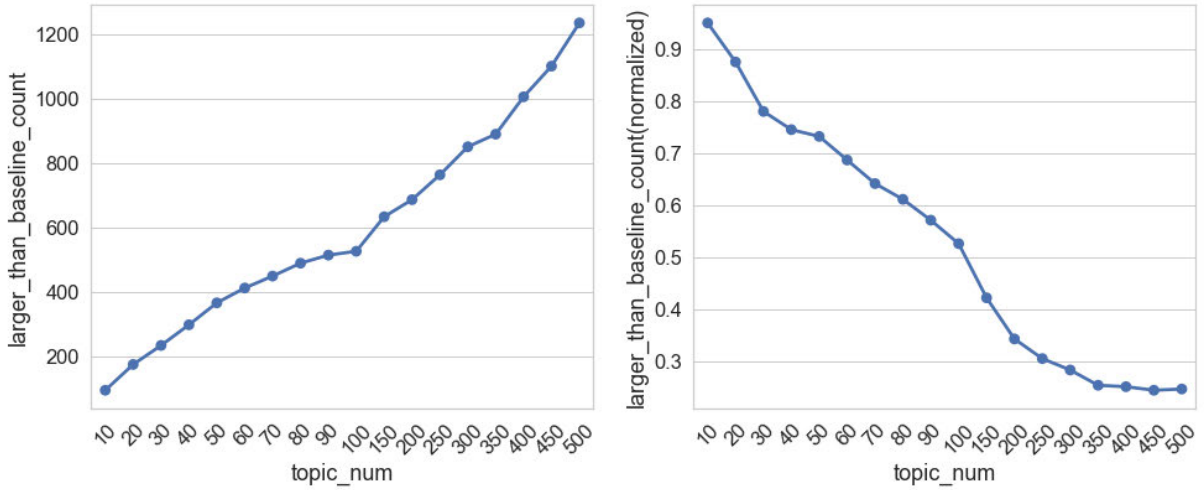

Abbildung 6.4 Anzahl der Topics, deren NPMI-Wert größer als der NPMI-Kontrollwert ist (links: absolute Anzahl; rechts: Prozentzahl)

Die Untersuchungen in diesem Kapitel haben folgendes gezeigt: Wenn die Topic-Modeling-basierte Dokumentklassifikation deutlich schlechter als die BoW-basierte Klassifikation

funktioniert, sollte die Anzahl der Topics erhöht und das Modell neu trainiert werden. Die Anzahl der Topics sollte auch erhöht werden, wenn mehrere Topic-Modelle mit gleicher Parameter-Einstellung trainiert werden, aber die Abweichung zwischen deren Klassifikationsergebnissen sehr groß ist. Außerdem ist es möglichst zu vermeiden, Topic-Modelle mit einer sehr hohen Anzahl von Topics zu trainieren. Die Klassifikation wird zwar in dieser Untersuchung durch eine hohe Anzahl der Topics nicht beeinflusst, eine große Anzahl von wenig kohärenten Topics mit weniger häufigen Wörtern könnte allerdings für die inhaltliche Exploration der Textdaten sehr problematisch sein. Werden die obigen Untersuchungsergebnisse kombiniert ist festzustellen, dass die Anzahl der Topics für die zu untersuchende Artikelsammlung auf 80 oder 90 eingestellt werden sollte. Die Klassifikation funktioniert fast ebenso gut wie die Baseline und die Modelle enthalten zudem auch nicht sehr viele weniger kohärente Topics.

6.2 Hyperparameter Alpha

In diesem Kapitel bezieht sich das Experiment auf den Einfluss des Hyperparameters Alpha[68] auf die Qualität des Topic-Modells. Das Setting von Alpha ist $\alpha \in \{0{,}01, 0{,}05, 0{,}1, 0{,}5, 1, 5, 10, 20, 30, 40, 50, 100\}$, die Anzahl der Topics wird zuerst auf 80 und 90 eingestellt, die in der letzten Untersuchung als ideal angesehen wurde.

6.2.1 Dokumentklassifikation

In Abbildung 6.5 wird das Ergebnis der Klassifikationen visualisiert. Jedes Boxplot stellt die F1-Werte der zehn Topic-Modelle dar, die mit derselben Parameter-Einstellung trainiert werden. Die F1-Werte bleiben zuerst zwischen 0,705 und 0,745, wenn Alpha von 0,01 auf 1,0 erhöht wird. Dann zeigt die Verteilung der F1-Werte einen deutlich absteigenden Trend, wenn Alpha von 1 höhere Werte, bis zu 100, gesetzt wird. Darüber hinaus gibt es keinen systematischen Unterschied zwischen den Klassifikationsergebnissen, wenn die Anzahl der Topics auf 80 oder 90 eingestellt wird.

Auch wenn die Anzahl der Topics stärker variiert wird, können in Abbildung 6.6 ähnliche absteigende Trends beobachtet werden. Wenn Alpha kleiner als 0.5 ist, können die Topic-Modelle mit mehr Topics bessere Klassifikationen erzielen. Wenn Alpha größer als 1 ist,

[68] Hier bezieht der Alpha-Wert sich auf den jedes einzelnen Topics.

verändern sich jedoch die Klassifikationsergebnisse. Je größer die Anzahl der Topics und der Wert von Alpha sind, desto schneller sinken die F1-Werte ab. Im Vergleich dazu werden die Topic-Modelle mit weniger Topics in geringerem Maße beeinflusst. Die höchste Accuracy und der höchste F1-Wert liegen in diesem Test bei 0,761 und 0,753, beide Werte sind geringfügig schlechter als die Baseline (0,765 / 0,758).

Laut Griffiths & Steyvers (2004) hat ein Topic-Modell die bestmögliche Qualität, wenn die Summe der Alpha-Werte aller Topics gleich 50 und Beta gleich 0,01 ist. Vermutlich war in MALLET 2.0.7 deswegen der vorgegebene Wert der Summe der Alphas auch auf 50 eingestellt, während der Wert in MALLET 2.0.8 auf 5 verkleinert wird. David Mimno betont: „The general experience was that 50 was too large, and that 5 is a better default".[69] Mit dem Ergebnis der oben dargestellten Untersuchung kann auf der Klassifikationsebene die Einschätzung von Mimno bestätigt werden, dass ein kleiner Alpha-Wert (vor allem kleiner als 1) für das Topic Modeling besser geeignet ist.

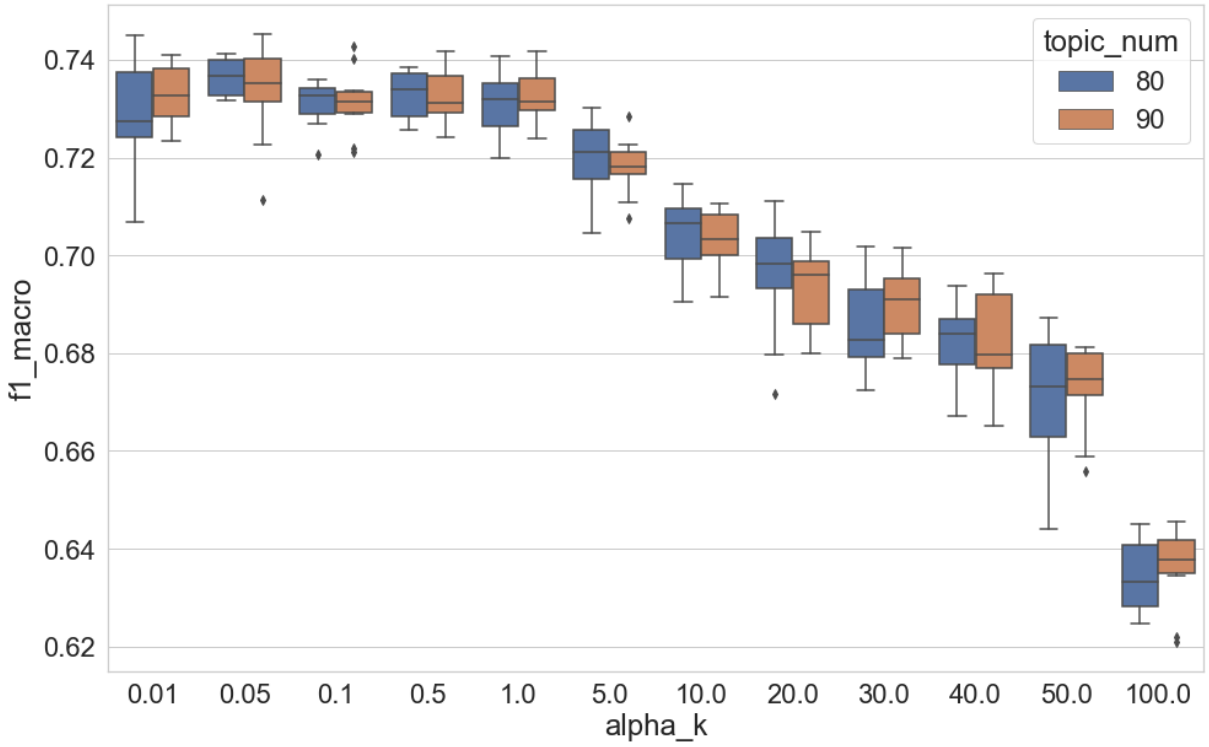

Abbildung 6.5 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Alpha

[69] https://stackoverflow.com/questions/45162186/mallet-topic-modeling-topic-keys-output-parameter, (17.09.2019).

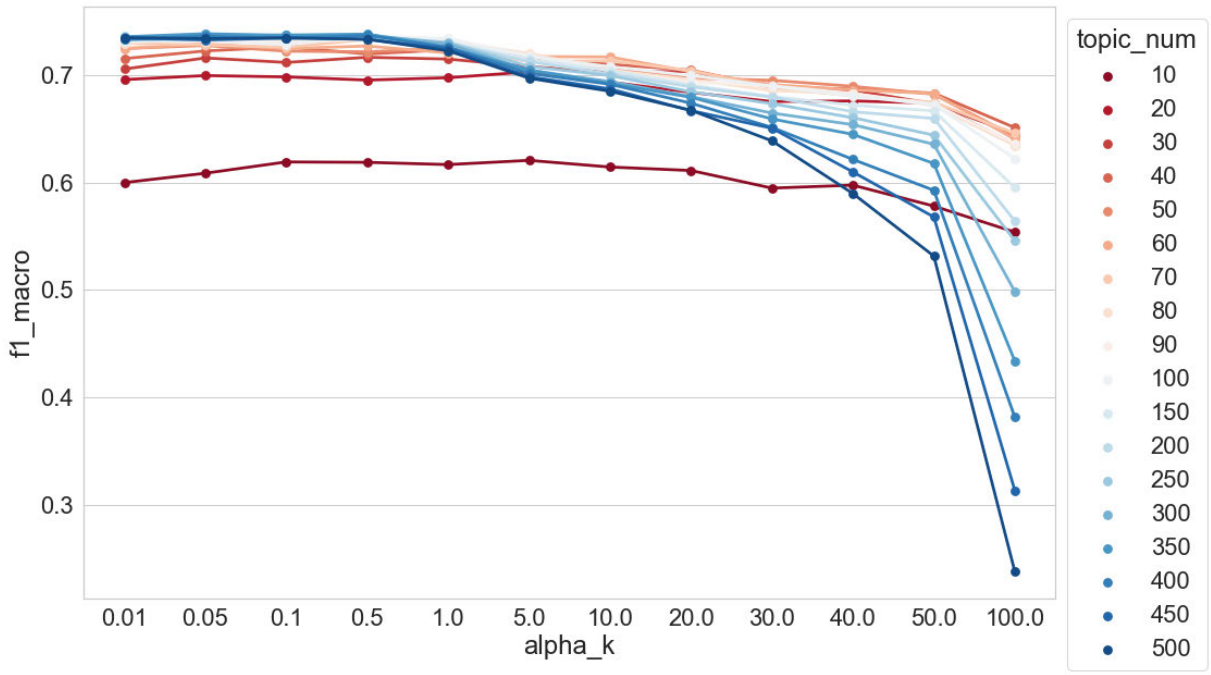

Abbildung 6.6 Durchschnittliche F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Alpha und Anzahl der Topics

6.2.2 Topic-Kohärenz

In Abbildung 6.7 wird die Verteilung der NPMI-Werte im Verhältnis zu Alpha visualisiert. Der Median der Verteilungen steigt zuerst etwas auf, wenn Alpha von 0,01 auf 1,0 erhöht wird. Die NPMI-Werte von mehr als 50 % der Topics sind höher als der Kontrollwert. Mehr als drei Viertel der Topics haben NPMI-Werte, die höher sind als der Kontrollwert, wenn der Wert von Alpha auf 1 und auf 5 eingestellt wird. Wenn Alpha weiter auf 100 erhöht wird, sinken die Mediane der Verteilungen ab und liegen unter dem NPMI-Kontrollwert. Zudem erreicht die Spannweite der Boxplots das Maximum bei Alpha = 0.05, also unterscheiden die NPMI-Werte sich stärker. Im Vergleich dazu ist die Varianz der NPMI-Werte kleiner, wenn Alpha einen großen Wert hat (z. B. α = 100). Dies bedeutet, dass ein kleiner Alpha-Wert zu einer spärlichen NPMI-Verteilung führt, während ein großer Alpha-Wert eine dichte NPMI-Verteilung zur Folge hat. Ähnlich wie bei den Klassifikationsergebnissen gibt es keine großen Unterschiede, wenn die Anzahl der Topics auf 80 und 90 eingestellt wird. Aufgrund dieses Ergebnisses ist festzustellen, dass der Wert von Alpha nicht zu groß sein darf, wenn durch Topic Modeling mehr kohärente Topics erhalten werden sollen.

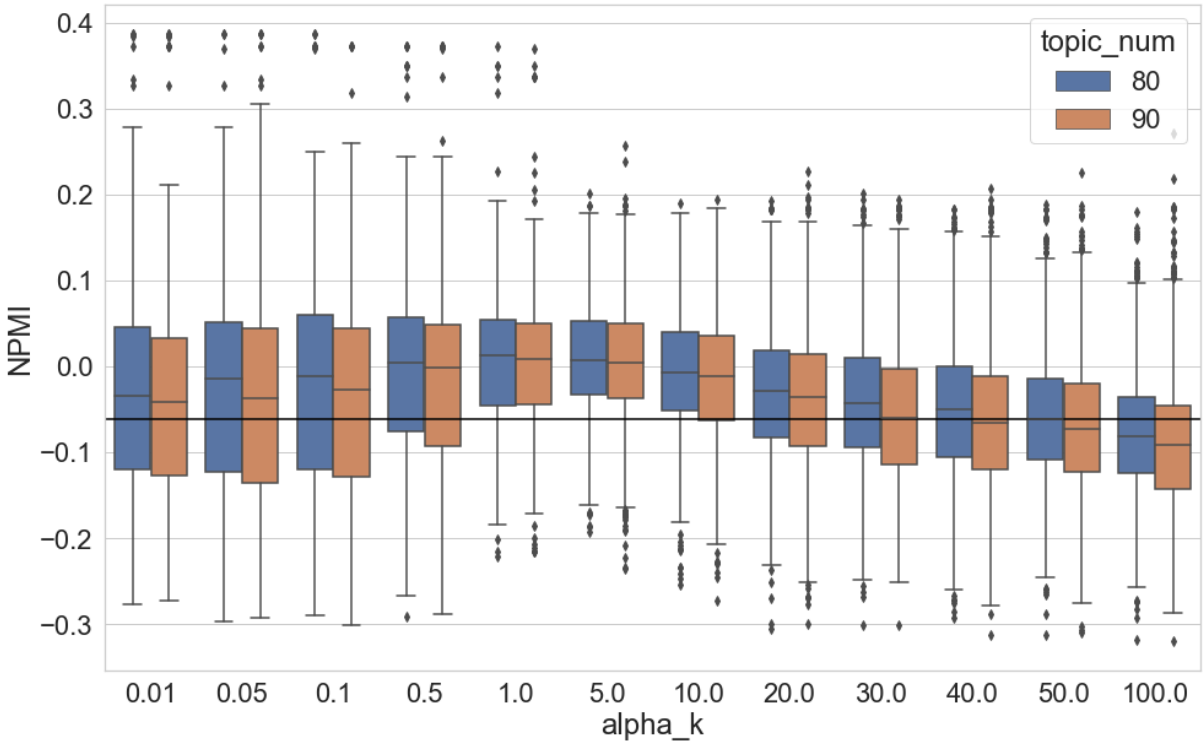

Abbildung 6.7 NPMI-Werte-Verteilung der Topics im Verhältnis zu Alpha

Wenn die Anzahl der Topics stärker variiert wird, ist in Abbildung 6.8[70] zunächst zu beobachten, dass ein Topic-Modell umso mehr nicht-kohärente Topics enthält, je mehr Topics es hat. Wenn z. B. die Anzahl der Topics auf 500 eingestellt wird, haben mindestens 75 % von ihnen einen NPMI-Wert unter der Baseline. Dieses Phänomen ist unabhängig von der Einstellung von Alpha. Die NPMI-Werte-Verteilungen werden durch Alpha weniger beeinflusst, wenn die Anzahl der Topics zu klein oder zu groß eingestellt wird (in dieser Untersuchung $\boldsymbol{T}$ = 10 und $\boldsymbol{T}$ = 500). Im Vergleich dazu ist die Änderung der Spannweite der NPMI-Werte-Verteilungen mit der Erhöhung von Alpha deutlicher, wenn die Anzahl der Topics zwischen 50 und 150 beträgt. Bei $\boldsymbol{\alpha}$ = 0.01 oder 0.05 liegen die NPMI-Werte meist zwischen -0,3 und 0,3, während sie bei $\boldsymbol{\alpha}$ = 20 oder 50 zwischen ca. 0,15 und -0,33 betragen.

Werden die Klassifikationsergebnisse und die NPMI-Werte-Verteilungen kombiniert kann man feststellen, dass es nicht ideal ist, den Alpha-Wert jedes Topics beim Training des Modells größer als 1 einzustellen. In den folgenden Untersuchungen wird der Alpha-Wert jedes Topics immer auf 0,05 eingestellt.

[70] Die Ausreißer wurden bei der Visualisierung der Daten aus den Boxplots entfernt.

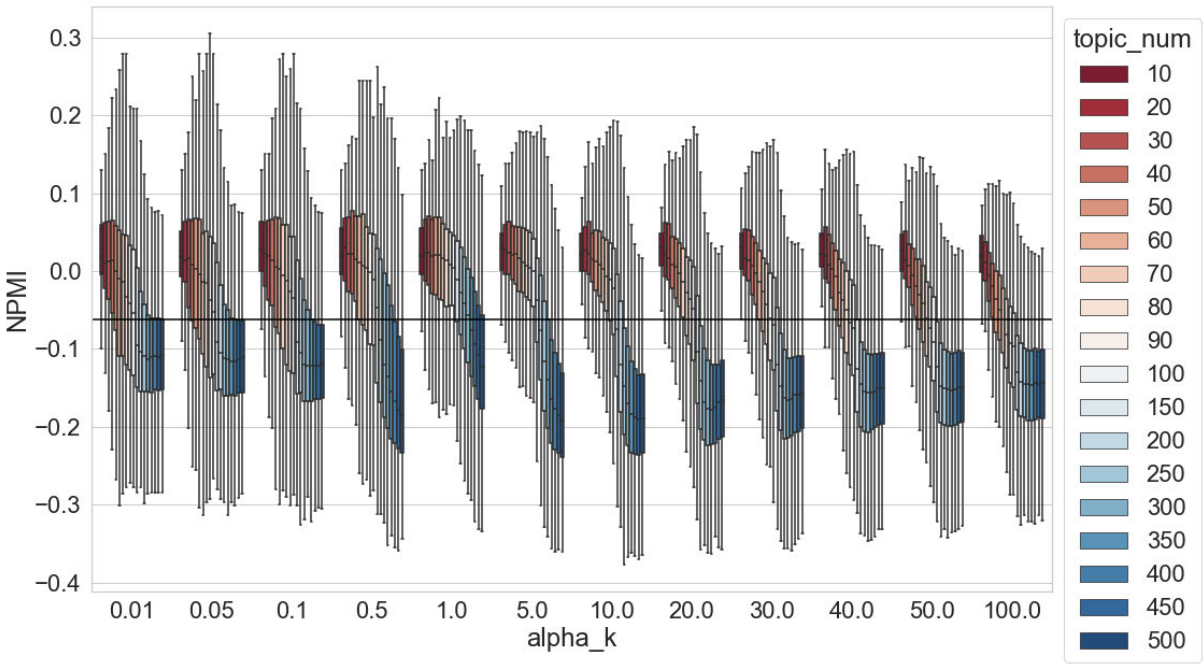

Abbildung 6.8 NPMI-Werte-Verteilung der Topics im Verhältnis zu Alpha und Anzahl der Topics

6.3 Hyperparameter-Optimierung

In diesem Kapitel untersucht das Experiment den Einfluss der Hyperparameter-Optimierung auf die Qualität des LDA-Modells. Die Optimierung bezieht sich nur auf den Hyperparameter der Topic-Dokument-Verteilung. Wenn Topic Modeling mit MALLET durchgeführt wird, müssen zwei Parameter eingestellt werden, um die Hyperparameter-Optimierung einzusetzen:[71] „Optimize Burn-in“ und „Optimize Interval“. Der erste Parameter legt die Anzahl der Iterationen fest, die vor der ersten Schätzung des Hyperparameters ausgeführt werden müssen, der zweite bestimmt die Anzahl der Iterationen zwischen der Neuschätzung der Hyperparameter. In dieser Arbeit werden die zwei Parameter „Optimize Burn-in“ und „Optimize Interval“ als ***OB*** und ***OI*** bezeichnet. Das Setting der Parameter in dieser Untersuchung ist ***OB*** ∈ {20, 50, 100, 200, 300, 400, 500} und ***OI*** ∈ {20, 50, 100, 200, 300, 400, 500}. Durch die Kombination der beiden Parameter wird analysiert, wie sich die Qualität des Topic-Modells verändert, wenn Hyperparameter beim Topic Modeling ganz früh (***OB*** =20) oder etwas später (***OB*** =500) optimiert werden. Darüber hinaus wird betrachtet, wie die Qualität

[71] „Hyperparameter Optimization“, unter http://mallet.cs.umass.edu/topics.php, (01.05.2020).

des Topic-Modells sich ändert, wenn Hyperparameter beim Topic Modeling sehr häufig (***OI*** =20) oder weniger häufig (***OI*** =500) optimiert werden.

6.3.1 Dokumentklassifikation

In Abbildung 6.9 werden die Klassifikationsergebnisse bei allen Kombinationen von ***OB*** und ***OI*** visualisiert. Die höchste Accuracy und der höchste F1-Wert erreichen in dieser Untersuchung 0,764 und 0,756, was etwas unter der Baseline (0,765 / 0,758) liegt. Hier lässt sich keine systematische Veränderung bei der Erhöhung der beiden Parameter ***OB*** und ***OI*** beobachten. In dieser Untersuchung wurde zudem die Anzahl der Topics variiert, das Setting ist $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$, wobei nur die Ergebnisse bei $\boldsymbol{T} = 80$ und 90 visualisiert werden, weil die Ergebnisse bei den anderen Settings von Anzahl der Topics auch keine systematische Veränderung zeigen. Um redundante Informationen zu vermeiden, werden die anderen Ergebnisse nicht mehr visualisiert. Schlussfolgerung aus dieser Untersuchung ist, dass es kein Zusammenhang zwischen der Topic-Modeling-basierten Dokumentklassifikation und der Einstellung der beiden Parameter ***OB*** und ***OI*** festzustellen gibt. Es lässt sich also festhalten, dass die Klassifikationsergebnisse unabhängig davon sind, wann und wie oft die Hyperparameter-Optimierung eingesetzt wird.

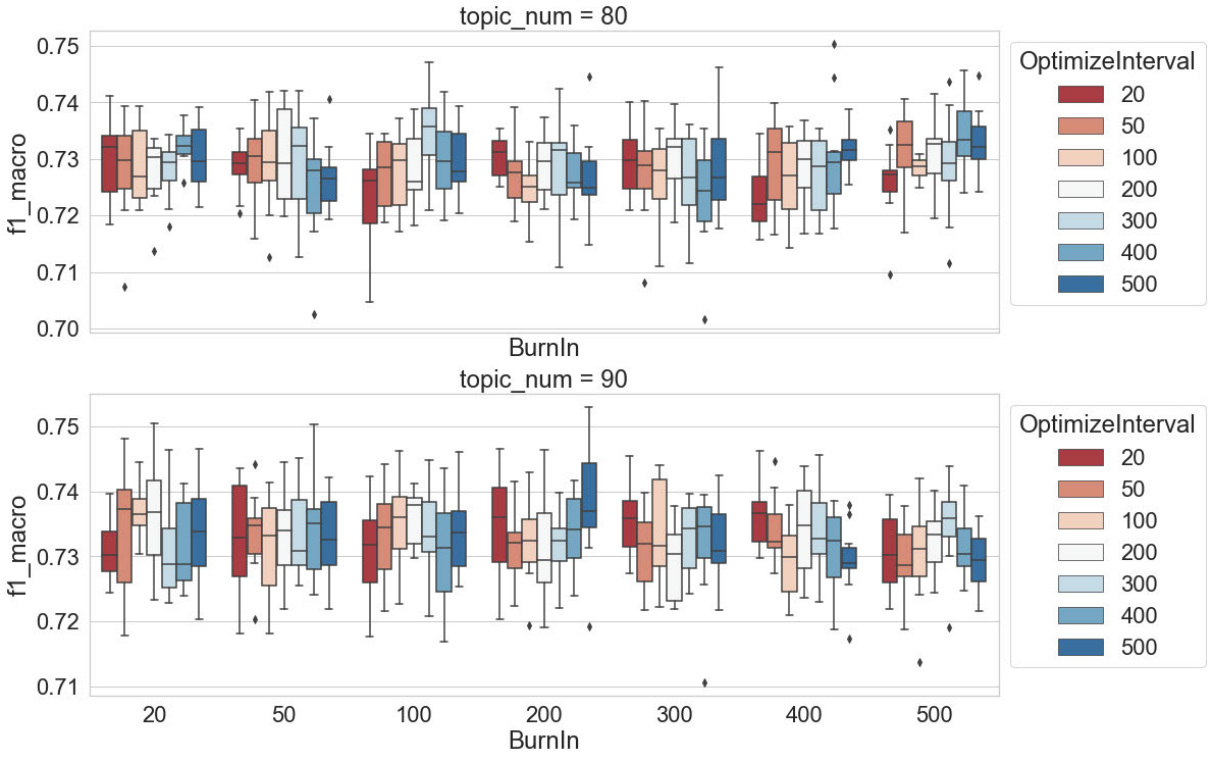

Abbildung 6.9 F1(Makro)-Werte der Dokumentklassifikation im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ (oben: ***T*** = 80, unten: ***T*** = 90)

Zusätzlich werden auch die Klassifikationsergebnisse verglichen, wenn Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. In dieser Untersuchung wird die Anzahl der Topics variiert und die ***OB***- und ***OI***-Werte jeweils auf 200 eingestellt. Durch den Vergleich in Abbildung 6.10 lässt sich erkennen, dass sich die Klassifikation mit der Erhöhung der Anzahl der Topics verbessert, unabhängig davon, ob die Hyperparameter-Optimierung beim Topic Modeling eingesetzt wird. Die Abbildung zeigt zudem, dass die Klassifikation deutlich bessere Ergebnisse erbringt, wenn Topic-Modelle mit 80, 250 oder 350 Topics und ganz ohne Hyperparameter-Optimierung trainiert werden. Das stellt aber keinen aussagekräftigen Beweis dafür dar, dass es in jedem Fall vorteilhaft ist, Topic-Modelle ohne Hyperparameter-Optimierung zu trainieren. In der Mehrzahl der Fälle gibt es keinen wesentlichen Unterschied bei den Ergebnissen der Klassifikationen.

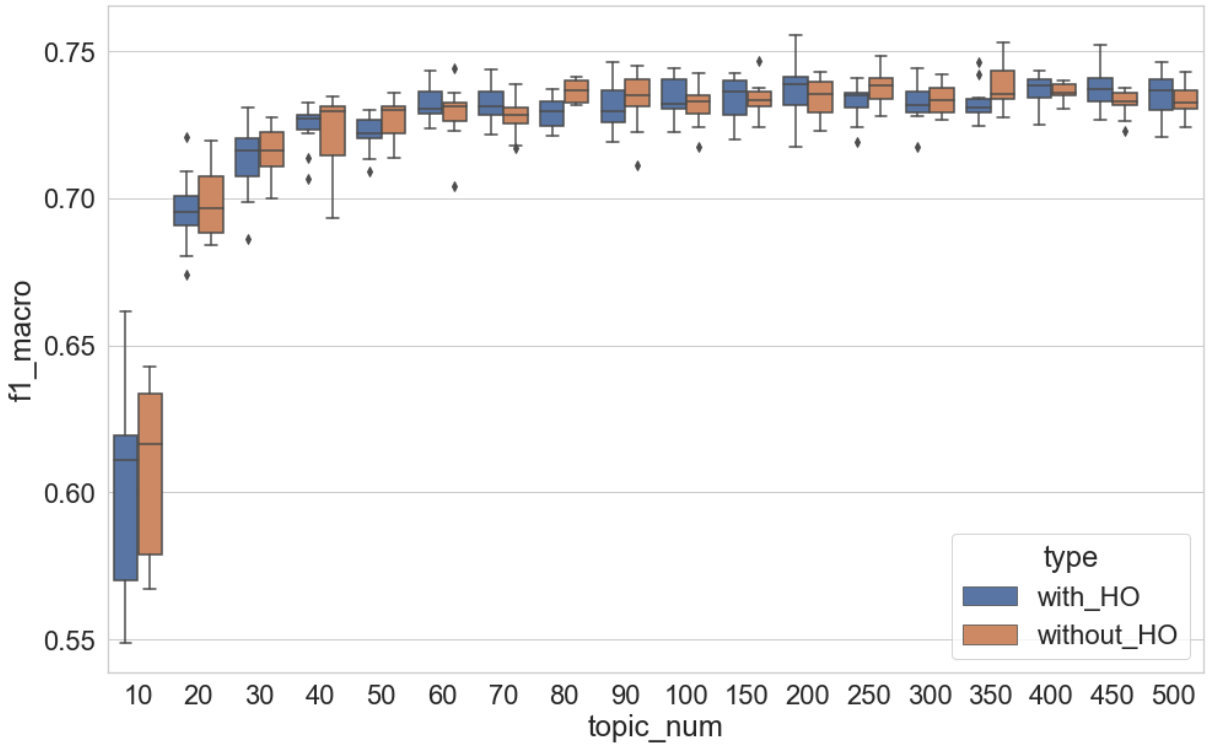

Abbildung 6.10 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics

Bei der Inferenz der Topics wird in MALLET voreingestellt, dass der Hyperparameter Alpha für alle Topics gleich ist. Durch den Einsatz der Hyperparameter-Optimierung wird Alpha für alle Topics während des Inferenz-Prozesses neu berechnet. Dadurch lässt sich eine Annäherung des asymmetrischen Alpha erreichen. Das Ergebnis in Wallach et al. (2009) zeigt, dass ein asymmetrischer Hyperparameter Alpha bessere Topic-Modelle bei einer Erhöhung der Topics-Anzahl erzielen kann. Allerdings stimmt die Schlussfolgerung mit dem hier vorliegenden Untersuchungsergebnis nicht überein. Die Hyperparameter-Optimierung hat in dieser Untersuchung offenbar keinen genau definierbaren Einfluss auf die Topic-Modeling-basierte Dokumentklassifikation. Um die Ursache der Differenzen zwischen der Untersuchung in dieser Arbeit und der in Wallach et al. (2009) zu explorieren, wird das Setting der beiden Versuche verglichen. Hier sind besonders die Evaluationsmethode und die Datenmenge interessant.

So werden unterschiedliche Methoden verwendet, um die Qualität des Topic-Modells zu evaluieren. In Wallach et al. (2009) wird die Log-Probability des Held-out-Dokumentes für die Evaluation des Modells berechnet, während in diesem Unterkapitel die Topic-Modeling-basierte Klassifikation als Evaluationsmethode verwendet wurde. Daraus ergibt sich als erste Annahme, dass der Unterschied zwischen den Untersuchungsergebnissen auf die unterschiedlichen Evaluationsmethoden zurückzuführen ist. Darüber hinaus könnten die beobachteten Differenzen durch unterschiedliche Datenmengen verursacht werden. Das LDA-Topic-Modeling basiert auf der bayesschen Statistik, die die folgenden Schritte umfasst: Zunächst wird die A-priori-Verteilung definiert, die die subjektiven Überzeugungen von Menschen über einen Parameter einbezieht. Danach wird die A-priori-Verteilung mit Daten aktualisiert, um die A-posteriori-Verteilung zu erhalten, die unsere aktualisierten Überzeugungen über den Parameter darstellt, nachdem die Daten gesehen wurden. Im Kontext des Topic Modeling bilden die Hyperparameter Alpha und Beta die A-priori-Verteilungen der Topic-Dokument-Verteilung und Topic-Wort-Verteilung. Nach dem Topic-Modeling-Prozess entsprechen die Topic-Dokument-Verteilung und die Topic-Wort-Verteilung wiederum der A-posteriori-Verteilungen. Dabei ist zu beachten, dass bei einer großen Datenmenge die Daten die A-posteriori-Verteilung dominieren – die A-priori-Verteilung hat einen geringeren oder gar keinen Einfluss auf die A-posteriori-Verteilung, wenn bei der Aktualisierung ein großer Datensatz existiert. In Wallach et al. (2009) werden drei Korpora für die Untersuchung verwendet, die jeweils 540, 1016 und 1768 Dokumente enthalten. Die durchschnittliche Dokumentlänge beträgt 101,87, 148,17 und 270,06 Wörter. Im Vergleich dazu enthält das in dieser Arbeit verwendete Zeitungskorpus 2000 Zeitungsartikel, die durchschnittliche

Dokumentlänge bei ca. 1800 Wörter liegt. Das bedeutet, dass das Korpus in dieser Arbeit sieben- bis 60-mal größer ist. Zugleich ist die durchschnittliche Chunk-Length mindestens 6-mal größer. Aus diesen Werten lässt sich die zweite Annahme ableiten: Wenn beim Training des Topic-Modells eine große Datenmenge vorhanden ist, ist der Einfluss des Hyperparameter Alpha auf die Topic-Dokument-Verteilung geringer. Auch die Hyperparameter-Optimierung hat aus diesem Grund einen nur geringen Einfluss.

Um die beiden Annahmen zu testen, werden die im Folgenden erläuterten Untersuchungen durchgeführt, die sich auf Evaluationsmethode und Datenmenge konzentrieren. Die Datenmenge kann durch die Größe des Untersuchungskorpus oder die Länge der einzelnen Dokumente geändert werden. Deshalb wird der Einfluss der beiden Parameter in den folgenden Untersuchungen getestet.

Test der Evaluationsmethode: Zunächst muss sichergestellt werden, dass die zu beobachtenden Unterschiede nicht auf die angewandten Evaluationsmethoden zurückzuführen sind. Deshalb wird das Korpus in 80 % Trainingsdaten (1600 Dokumente) und 20 % (400 Dokumente) Testdaten geteilt. Zwei Topic-Modelle werden auf den Trainingsdaten trainiert. Die Anzahl der Topics und die Iteration des Gibbs-Samplings werden jeweils auf 80 und 2000 eingestellt. Ein Modell wird mit Hyperparameter-Optimierung trainiert, während ***OB*** und ***OI*** jeweils auf 200 und 200 eingestellt werden. Ein weiteres Topic-Modell wird ohne Optimierung mit einem symmetrischen Hyperparameter Alpha trainiert. Danach wird die Log-Probability des Held-out-Dokumentes (also der Testdaten) berechnet. Dieser Vorgang wird zehnmal wiederholt. Die Ergebnisse der zehn Tests werden in der Abbildung 6.11 visualisiert. Jedes Boxplot stellt die Verteilung der Log-Probability der 400 Dokumente dar. In der Visualisierung ist zu erkennen, dass keine Differenzen in den zehn Tests auftreten, wenn Topic-Modelle mit oder ohne Hyperparameter-Optimierung trainiert werden.

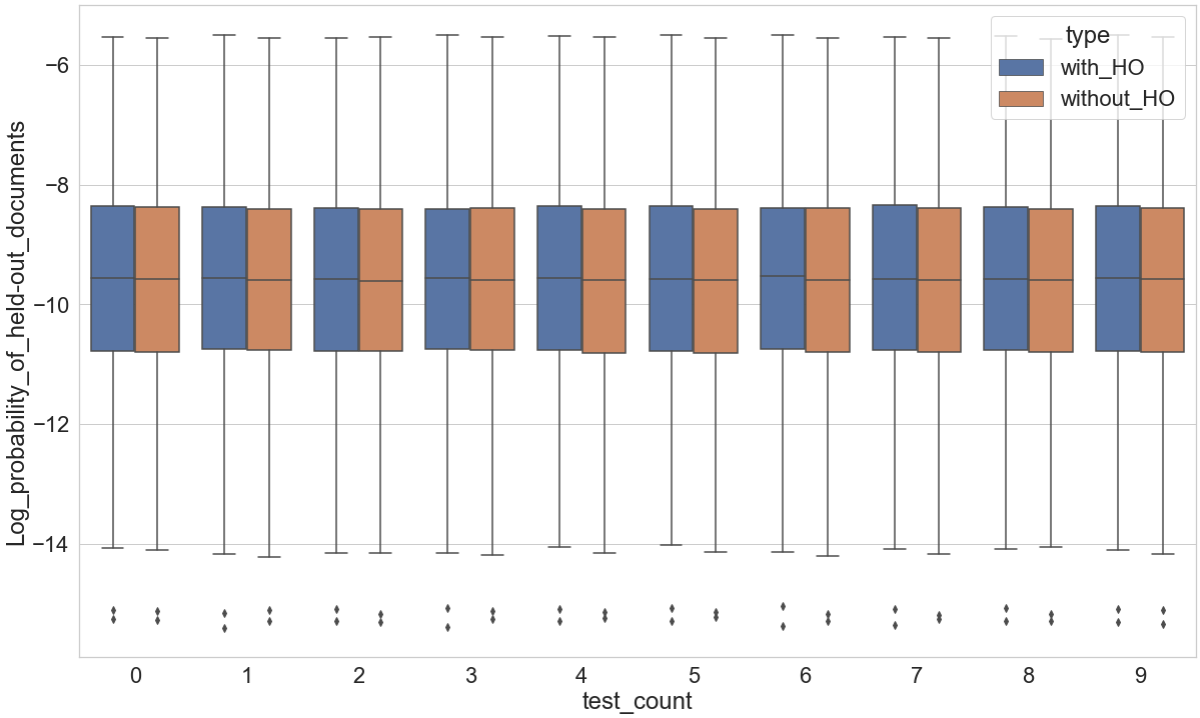

Abbildung 6.11 Log-Probability des Held-out-Dokumentes

Test der Korpusgröße: Die nächste Aufgabe besteht nun darin, den Einfluss der Korpusgröße zu analysieren. Aus dem Zeitungskorpus werden zehn Dokumente jeder thematischen Klasse zufällig ausgewählt, um ein Untersuchungskorpus von 100 Dokumenten aufzubauen. Das Korpus enthält dann ca. 180.000 Wörter und ist größer als das kleinste Korpus, aber zugleich kleiner als das größte Korpus in Wallach et al. (2009). Die Settings der Parameter in dieser Untersuchung sind jeweils $\boldsymbol{T} \in \{10, 30, 50, 80, 100, 120, 150, 180, 200, 250, 300, 400, 500\}$, $\boldsymbol{OB} \in \{20, 50, 100, 200, 300, 400, 500\}$ und $\boldsymbol{OI} \in \{20, 50, 100, 200, 300, 400, 500\}$. Für jedes Setting werden auch hier zehn Topic-Modelle trainiert. Zuerst werden die Klassifikationsergebnisse in Bezug auf die Einstellung von ***OB*** und ***OI*** in Abbildung 6.12 visualisiert. Die Anzahl der Topics wird auf 80 eingestellt. Die Resultate der anderen Settings von ***T*** sind ähnlich und werden deshalb nicht visualisiert. Es zeigt sich hier ein ähnliches Bild wie bei den vorherigen Untersuchungen. Die Abbildung lässt dabei keinen deutlichen Trend erkennen. Im Vergleich dazu zeigen die Klassifikationsergebnisse in Bezug auf die Anzahl der Topics in Abbildung 6.13 eine interessante Situation: Die Klassifikation kann gelegentlich sogar bessere Ergebnisse erzielen, wenn Topic-Modelle ohne Hyperparameter-Optimierung trainiert werden. Das Phänomen wird dann beobachtet, wenn Topic-Modelle weniger als 150 Topics enthalten. Ab $\boldsymbol{T} = 200$ ist der Unterschied zwischen den zwei beiden nicht mehr klar zu

beobachten. Die Settings von ***OB*** und ***OI*** waren hier jeweils auf 200 eingestellt. Die Ergebnisse der anderen Settings sind ähnlich und werden deshalb nicht visualisiert.

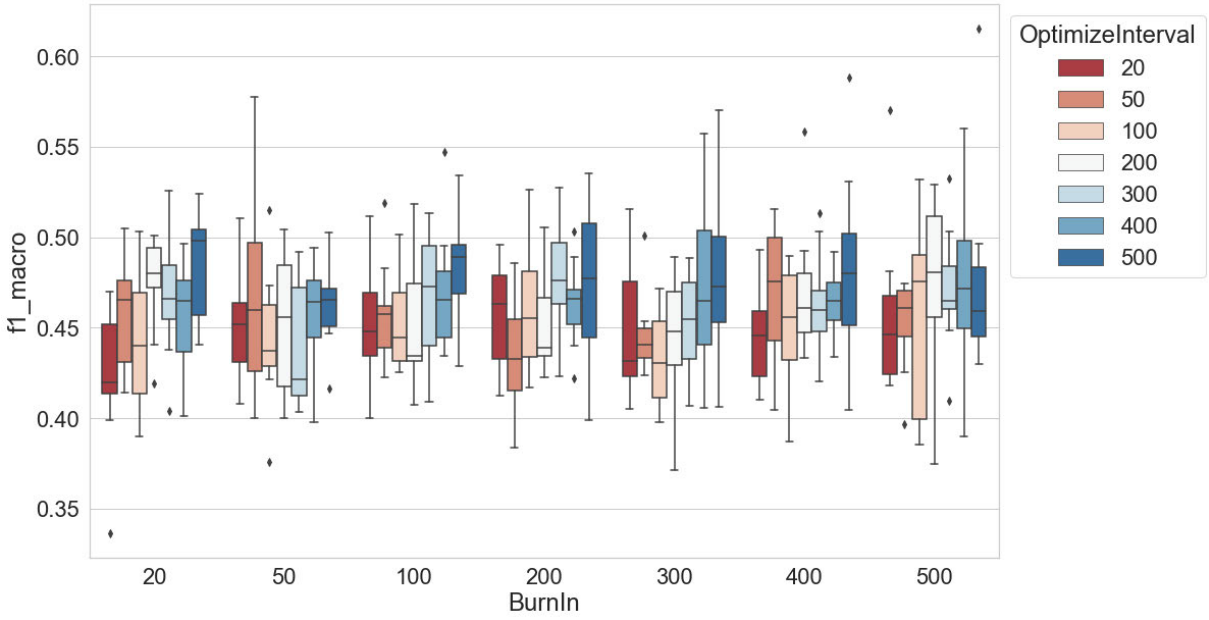

Abbildung 6.12 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ (100 Dokumente)

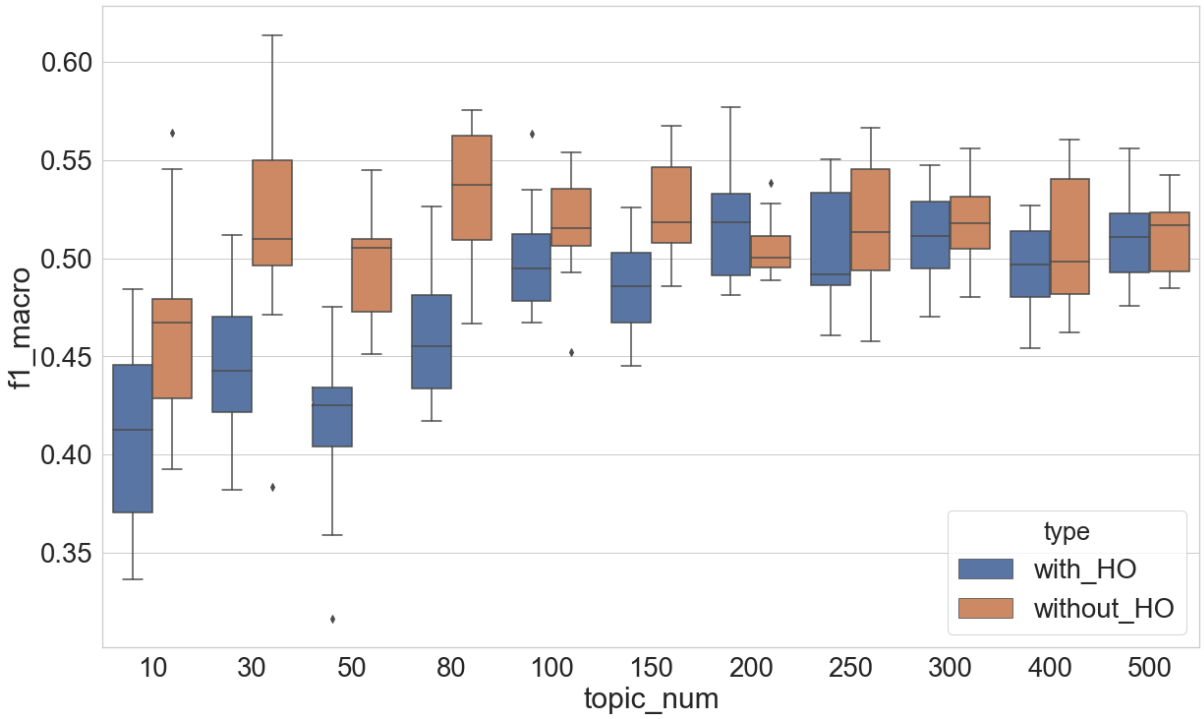

Abbildung 6.13 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics (100 Dokumente)

Test der Chunk-Length: Die nächste Untersuchung hat das Ziel, den Einfluss der Chunk-Length zu analysieren. Dafür werden alle 2000 Zeitungsartikel in Chunks zerlegt, die Chunk-Length wird auf ***C*** = 100 eingestellt. Das Setting der anderen wichtigen Parameter in dieser Untersuchung ist identisch zu den bisherigen Untersuchungen: ***T*** ∈ {10, 30, 50, 80, 100, 120, 150, 180, 200, 250, 300, 400, 500}, ***OB*** ∈ {20, 50, 100, 200, 300, 400, 500} und ***OI*** ∈ {20, 50, 100, 200, 300, 400, 500}. Für jedes Setting werden hier wieder zehn Topic-Modelle trainiert. Die Klassifikationsergebnisse in Bezug auf ***OB*** und ***OI*** werden in Abbildung 6.14 visualisiert. Die Anzahl der Topics wird auf 80 eingestellt. Die Ergebnisse von weiteren Settings von ***T*** sind ähnlich und werden nicht visualisiert. Trotz einiger Schwankungen lässt sich in den Ergebnissen ein schwacher Trend erkennen: Anscheinend verschlechtern sich die Klassifikationsergebnisse mit der Erhöhung von ***OI***, wenn ***OB*** beim Topic Modeling größer als 100 eingestellt wird. Der Unterschied zwischen den F1-Verteilungen ist aber nicht besonders interessant. Die Klassifikationsergebnisse werden auch verglichen, wenn die Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. In dieser Untersuchung wird die Anzahl der Topics variiert, die Settings von ***OB*** und ***OI*** sind hier jeweils 200. Die Ergebnisse der anderen Settings sind ähnlich und werden deshalb nicht visualisiert. In Abbildung 6.15 ist deutlich zu erkennen, dass die Klassifikation stets bessere Ergebnisse erzielt, wenn Topic-Modelle mit einer Hyperparameter-Optimierung trainiert werden.

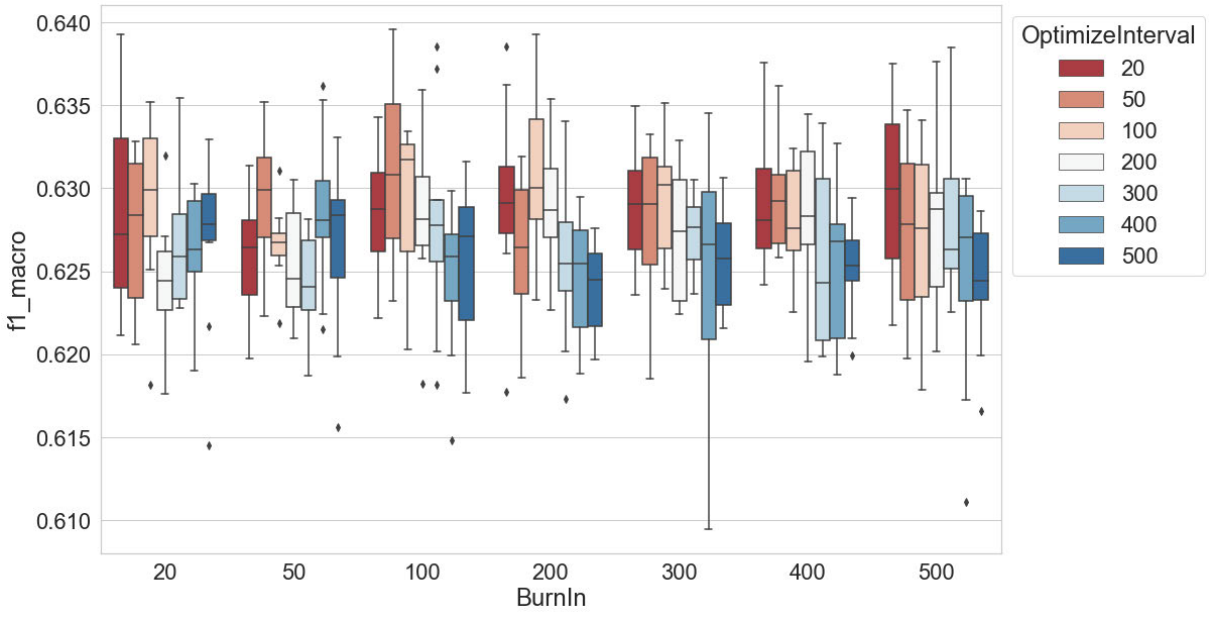

Abbildung 6.14 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ (100 Chunk-Length)

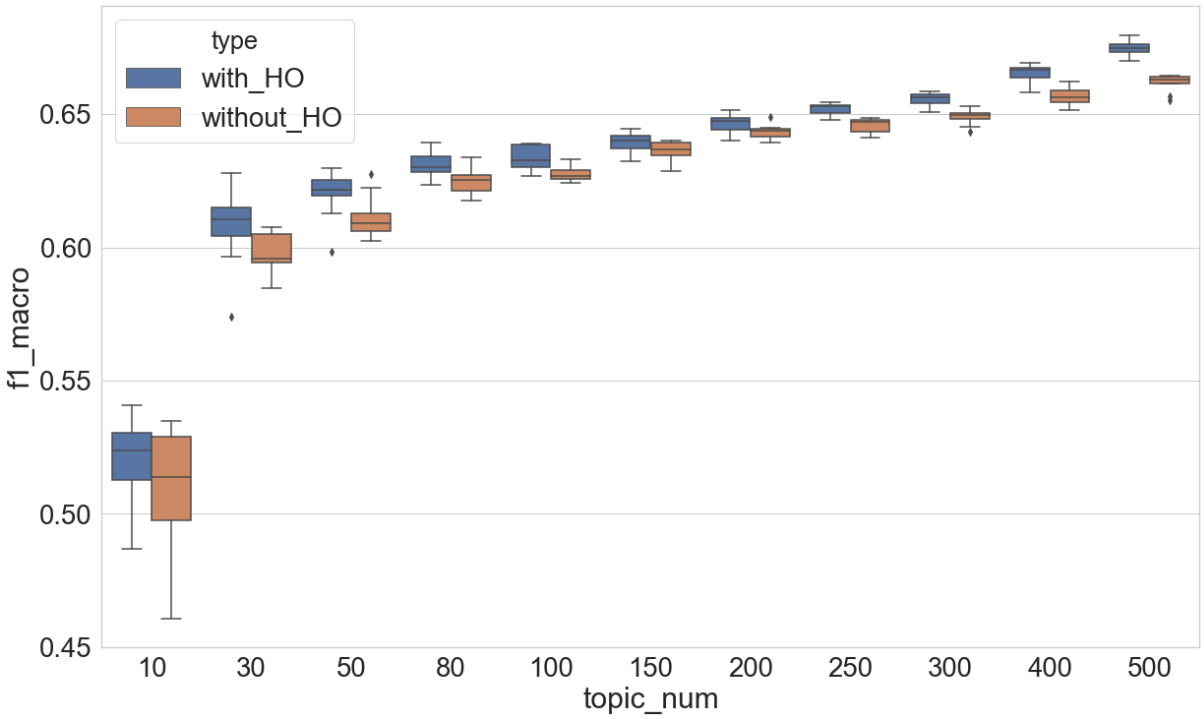

Abbildung 6.15 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics (100 Chunk-Length)

Das Ergebnis der früheren Untersuchung zeigt, dass die Hyperparameter-Optimierung keinen klaren Einfluss auf die Topic-Modelle hat, wenn die durchschnittliche Chunk-Length ca. 1800 Tokens beträgt. Wird die Chunk-Length dagegen auf 100 eingestellt, kann die Hyperparameter-Optimierung die Klassifikationsergebnisse bzw. die Topic-Modelle verbessern. Angesichts der Ergebnisse aller Untersuchungen ist anzunehmen, dass der Einfluss der Hyperparameter und der Hyperparameter-Optimierung auf die Topic-Modeling-basierte Dokumentklassifikation mit der Erhöhung der Chunk-Length abnimmt, weil die A-posteriori-Verteilung durch die Zunahme der Textdaten dominiert wird. Um diese Annahme zu bestätigen, wird eine weitere Untersuchung durchgeführt: Die 2000 Zeitungsartikel werden in Chunks zerlegt, das Setting der Chunk-Length ist $\boldsymbol{C} \in \{10, 20, 50, 80, 100, 120, 150, 180, 200, 250, 300, 350, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1700\}$. Bei jedem Setting von $\boldsymbol{C}$ werden zehn Topic-Modelle mit und ohne Hyperparameter-Optimierung trainiert. Die ***OB*** und ***OI*** werden jeweils auf 200 eingestellt, wenn die Hyperparameter-Optimierung beim Training eingesetzt wird; die Anzahl der Topics wird auf 80 eingestellt. Andere Settings der Anzahl der Topics wurden

ebenfalls getestet, die Ergebnisse sind ähnlich. Danach werden die Topic-Modeling-basierten Dokumentklassifikationen durchgeführt. Die Ergebnisse werden in Abbildung 6.16 visualisiert und bestätigen die früheren Annahmen. Es ist zu beobachten, dass die Klassifikationsergebnisse stets besser sind, wenn ***C*** zwischen 10 und 150 eingestellt wird und die Topic-Modelle mit Hyperparameter-Optimierung trainiert werden. Ab ***C*** = 180 ist der Unterschied zwischen den beiden Gruppen meist nicht mehr relevant. In einigen Situationen allerdings, beispielsweise bei ***C*** = 600, 1000 oder 1700, kann die Klassifikation bessere Ergebnisse erzielen, wenn keine Hyperparameter-Optimierung eingesetzt wird.

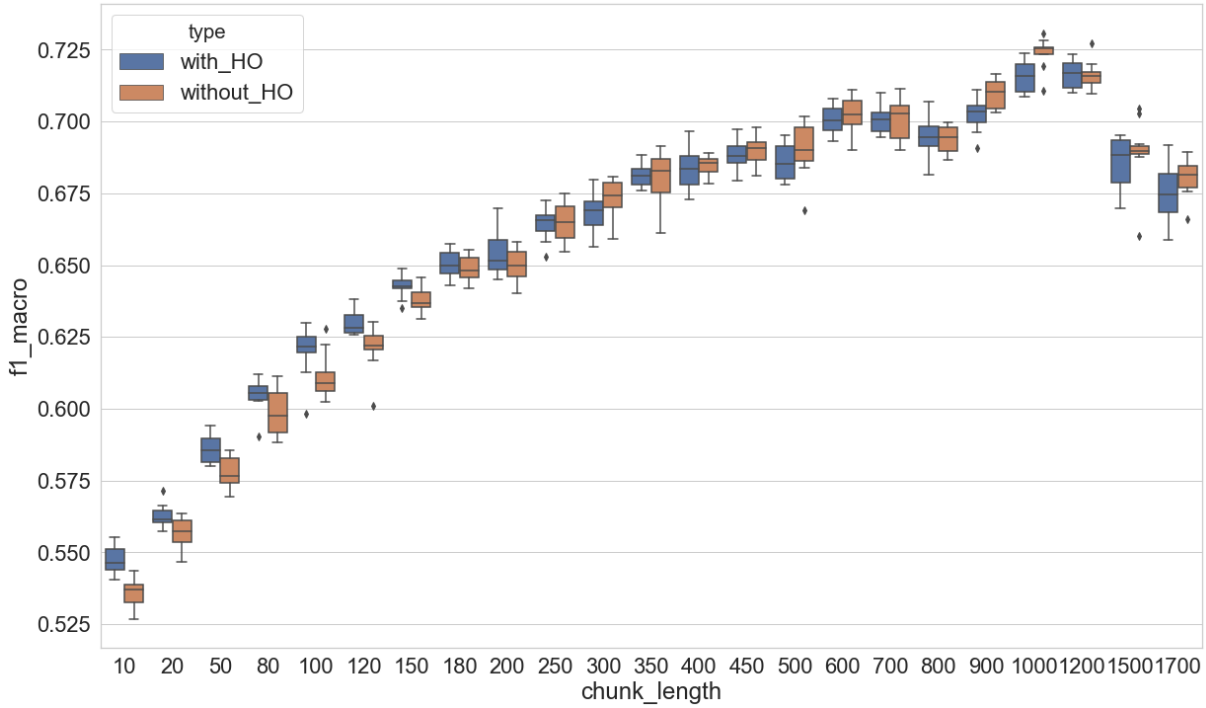

Abbildung 6.16 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Hyperparameter-Optimierung und Chunk-Length (80 Topics)

Aus den oben beschriebenen Tests geht hervor, dass die die Topic-Modeling-basierte Klassifikation durch den Einsatz der Hyperparameter-Optimierung verbessert werden kann, wenn keine sehr große Datenmenge in jedem Dokument beim Training vorliegt. Anders formuliert kann eine asymmetrische A-priori-Verteilung der Topic-Dokument-Verteilung eine Verbesserung der Klassifikationsergebnisse nicht garantieren, wenn die einzelnen Dokumente sehr lang sind.

6.3.2 Topic-Kohärenz

Nach den Untersuchungsergebnissen im letzten Unterkapitel wird in diesem Kapitel der Einfluss der Hyperparameter-Optimierung auf die Topic-Kohärenz überprüft und zuerst auf nicht segmentierten Texten. In Abbildung 6.17 werden die NPMI-Verteilungen bei allen Kombinationen von ***OB*** und ***OI*** visualisiert, während die Anzahl der Topics auf 80 und 90 eingestellt wird. In der Visualisierung ist auch hier kein klarer Trend zu beobachten. Die Anzahl der Topics wird ebenfalls variiert, das Setting ist $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$. Die so erhaltenen Ergebnisse sind allerdings ähnlich und werden deshalb nicht visualisiert.

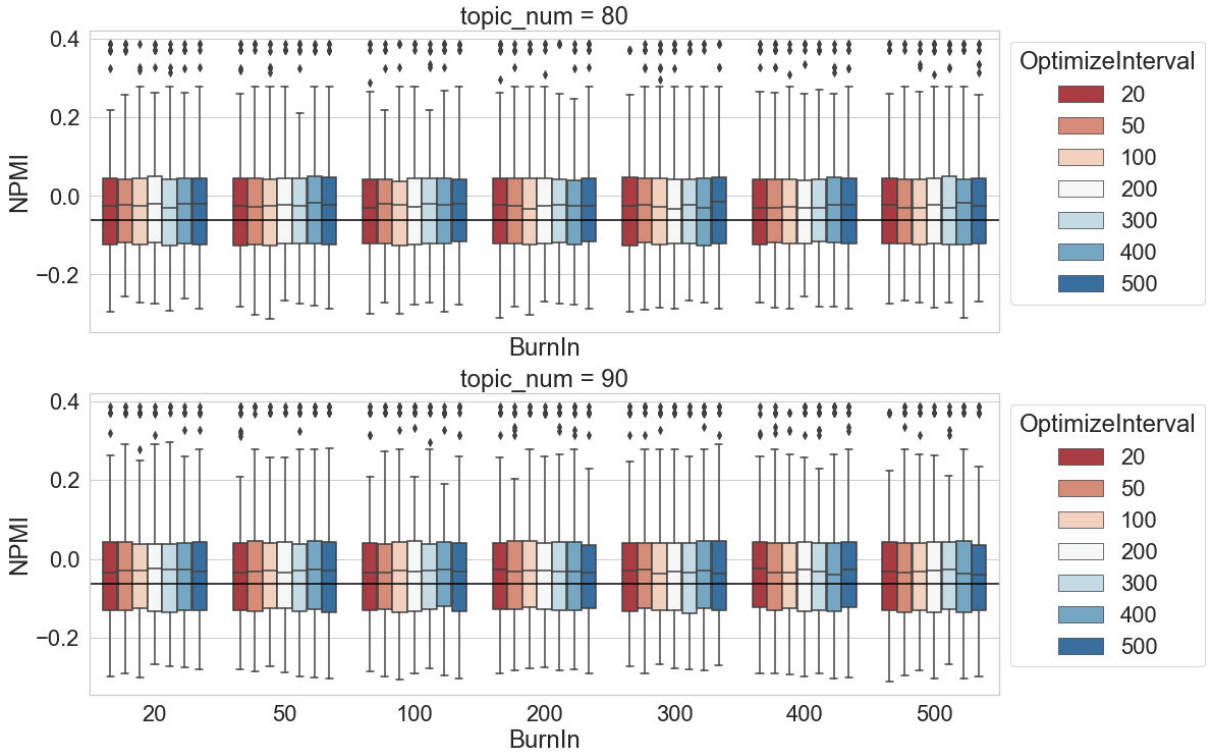

Abbildung 6.17 NPMI-Werte-Verteilung der Topics im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ und Anzahl der Topics (oben: $\boldsymbol{T} = 80$, unten: $\boldsymbol{T} = 90$)

Zusätzlich wird die Kohärenz der Topics verglichen, wenn Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. In dieser Untersuchung wird die Anzahl der Topics variiert: $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$. Der Hyperparameter-Alpha, ***OB*** und ***OI*** werden jeweils auf 0,05, 200 und 200 eingestellt (die Ergebnisse bei den anderen Settings von ***OB*** und ***OI*** sind ähnlich). Das Resultat in

Abbildung 6.18 zeigt, dass kein systematischer Unterschied zwischen den Topic-Modellen besteht, die mit und ohne Hyperparameter-Optimierung trainiert werden. Die Veränderung der NPMI-Werte-Verteilung hängt offenbar also nicht davon ab, ob eine Hyperparameter-Optimierung beim Topic Modeling eingesetzt wird.

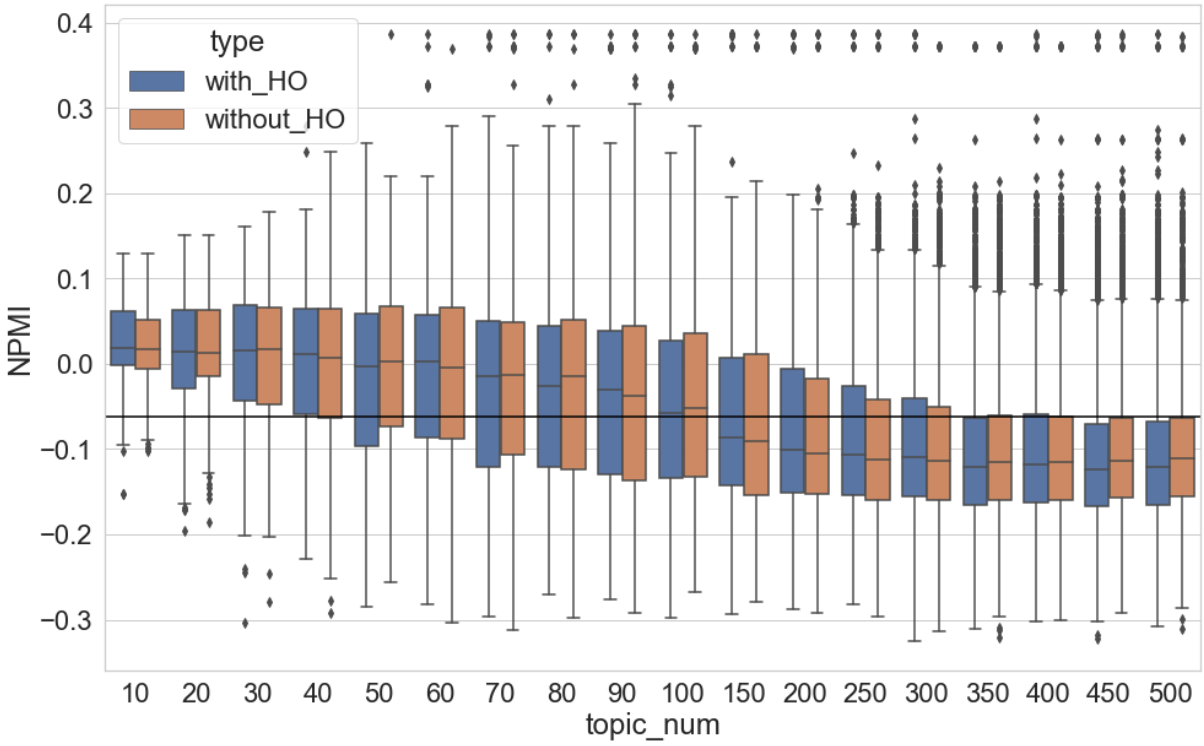

Abbildung 6.18 NPMI-Werte-Verteilung der Topics im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics

Test der Korpusgröße: In der Untersuchung auf dem nicht zerlegten Korpus wird kein klarer Einfluss der Hyperparameter-Optimierung beobachtet. Der nächste Schritt besteht darin, die Datenmenge zu reduzieren und den Einfluss der Hyperparameter-Optimierung nochmals zu überprüfen. Zunächst wird dafür das Untersuchungskorpus auf die 100 zufällig ausgewählten Zeitungsartikel reduziert. Das Setting der Parameter ist wie zuvor ***OB*** ∈ {20, 50, 100, 200, 300, 400, 500} und ***OI*** ∈ {20, 50, 100, 200, 300, 400, 500}, die Anzahl der Topics wird auf 80 eingestellt. Für jedes Setting werden zehn Topic-Modelle trainiert. Das Ergebnis in Abbildung 6.19 zeigt, dass es keinerlei systematischen Veränderungen der NPMI-Werte-Verteilungen bei der Veränderung der beiden zwei Parameter ***OB*** und ***OI*** gibt. Mehr als 75 % der Topics weisen NPMI-Werte unter dem NPMI-Kontrollwert auf.

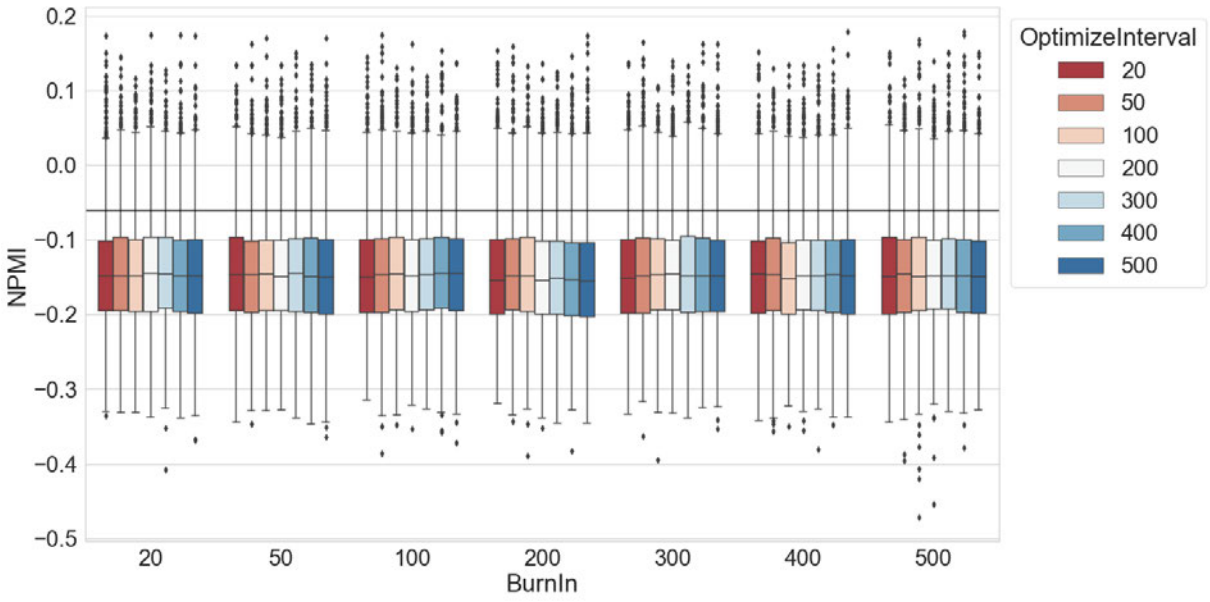

Abbildung 6.19 NPMI-Werte-Verteilung der Topics im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ (100 Dokumente)

Die Verteilungen der NPMI-Werte werden daraufhin nochmals in den Fällen verglichen, in denen Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. Das Setting für das Training der Topic-Modelle entspricht dem bisherigen: $\boldsymbol{T} \in \{10, 30, 50, 80, 100, 120, 150, 180, 200, 250, 300, 400, 500\}$, $\boldsymbol{OB} = 200$ und $\boldsymbol{OI} = 200$. Die Ergebnisse bei den anderen Settings von ***OB*** und ***OI*** sind ähnlich. Aus dem in Abbildung 6.20 dargestellten Ergebnis lässt sich nur schwer eine Antwort auf die Frage finden, ob die Topic-Kohärenz durch die Verwendung der Hyperparameter-Optimierung verbessert werden kann. Wenn die Anzahl der Topics kleiner als 50 oder größer als 250 ist, wird sogar eine höhere Anzahl von nicht-kohärenten Topics produziert, wenn Topic-Modelle mit Hyperparameter-Optimierung trainiert werden.

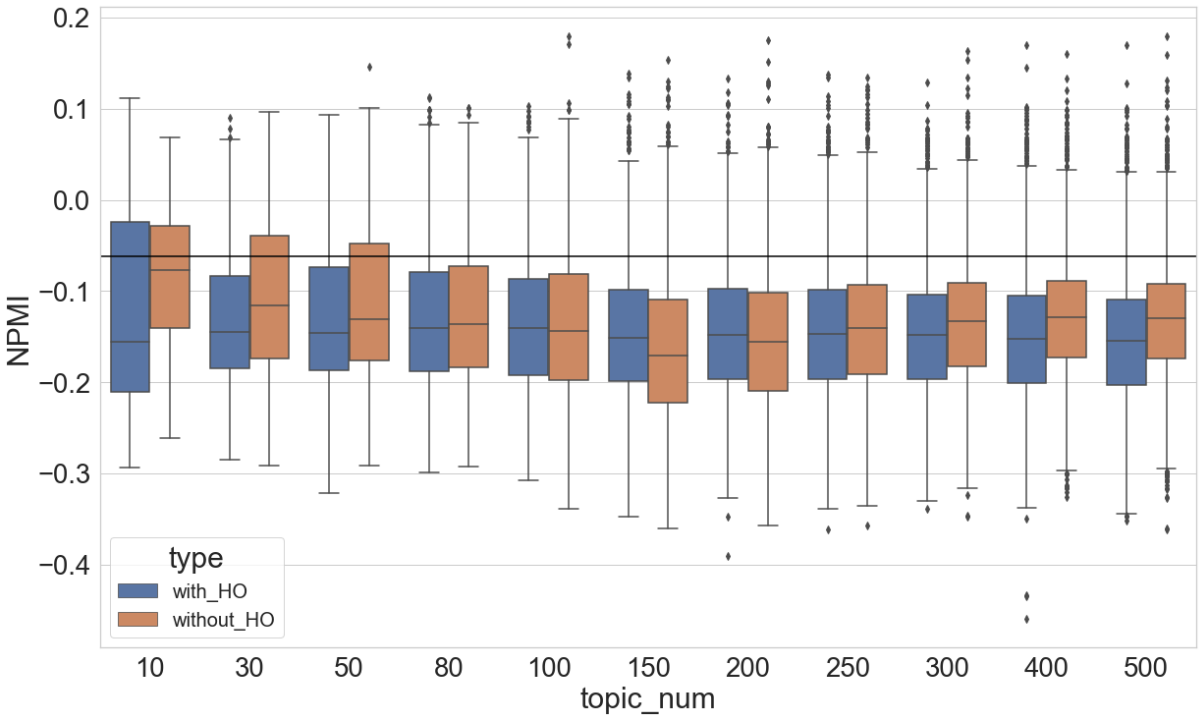

Abbildung 6.20 NPMI-Werte-Verteilung der Topics im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics (100 Dokumente)

Test der Chunk-Length: Der nächste Schritt besteht darin, die Chunk-Length jedes Dokuments im Untersuchungskorpus zu reduzieren und den Einfluss der Hyperparameter-Optimierung zu testen. Für diese Untersuchung werden alle Zeitungsartikel in Chunks zerlegt und die Chunk-Length auf ***C*** = 100 eingestellt. Das Setting der anderen Parameter bleibt stabil, es liegt also bei ***OB*** ∈ {20, 50, 100, 200, 300, 400, 500} und ***OI*** ∈ {20, 50, 100, 200, 300, 400, 500}. Die Anzahl der Topics wird auf 80 eingestellt. Für jedes Setting von ***C*** werden zehn Topic-Modelle trainiert. In Abbildung 6.21 werden die Verteilungen der NPMI-Werte in Bezug auf ***OB*** und ***OI*** visualisiert. Ein interessantes Phänomen ist, dass der Median aller NPMI-Werte-Verteilungen höher ist, wenn das Optimierungsintervall größer ist. Mit der Erhöhung von ***OI*** steigt der Median der NPMI-Werte-Verteilungen über den NPMI-Kontrollwert, während die Spannweite der Verteilungen keine deutliche Veränderung aufweist. Die Topic-Kohärenz ist im Ergebnis besser, wenn der Hyperparameter Alpha beim Training des Topic-Modells weniger häufig optimiert wird. Im Vergleich zum ***OI*** kann keine systematische Veränderung beobachtet werden, wenn das ***OB*** von 20 auf 500 erhöht wird.

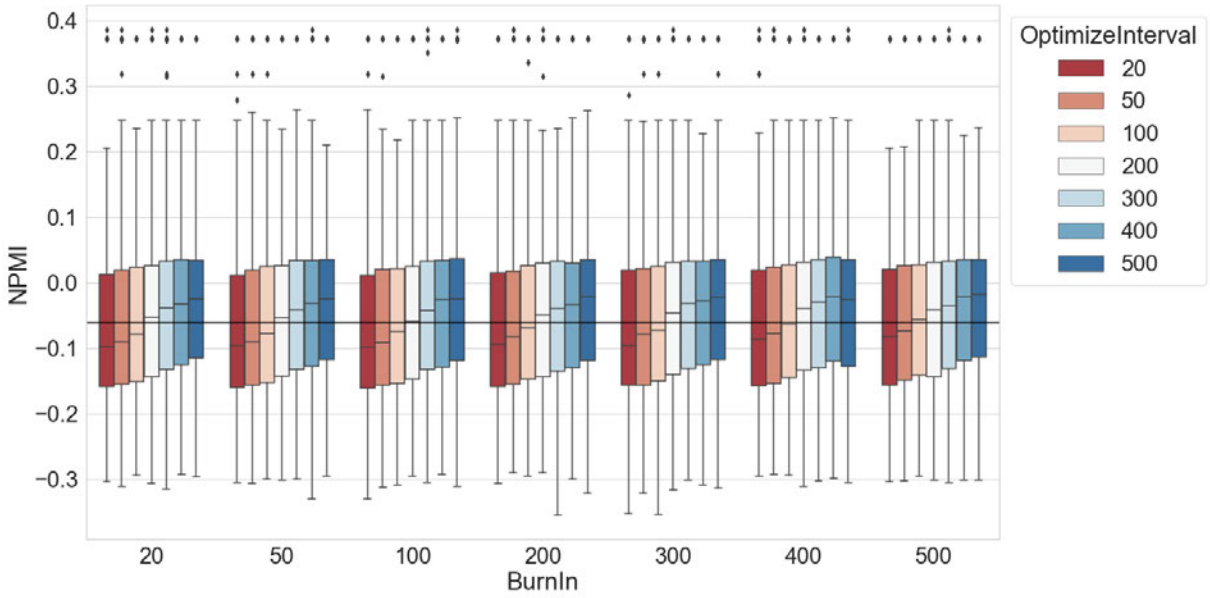

Abbildung 6.21 NPMI-Werte-Verteilung der Topics im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ (100 Chunk-Length)

Wie in den Untersuchungen zuvor werden auch die Verteilungen der NPMI-Werte verglichen (Abbildung 6.22), wenn Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. Die Settings der Parameter beim Training der Topic-Modelle sind: $\boldsymbol{T} \in \{10, 30, 50, 80, 100, 120, 150, 180, 200, 250, 300, 400, 500\}$, $\boldsymbol{OB} = 200$ und $\boldsymbol{OI} = 200$. Die Ergebnisse bei den anderen Settings von ***OB*** und ***OI*** sind ähnlich. Zunächst fällt in der Visualisierung auf, dass die Spannweite der NPMI-Werte-Verteilungen meist größer ist, wenn Topic-Modelle mit Hyperparameter-Optimierung trainiert werden. Das bedeutet, dass die kohärentesten Topics durch die Verwendung der Hyperparameter-Optimierung noch kohärenter geworden sind. Vice versa zeigt sich, dass die am wenigsten kohärenten Topics noch schlechter werden. In Bezug auf das Setting der Anzahl der Topics ist festzustellen, dass der mediane Unterschied der NPMI-Werte-Verteilungen zwischen den beiden Gruppen mit der Erhöhung der Topic-Anzahl größer wird.

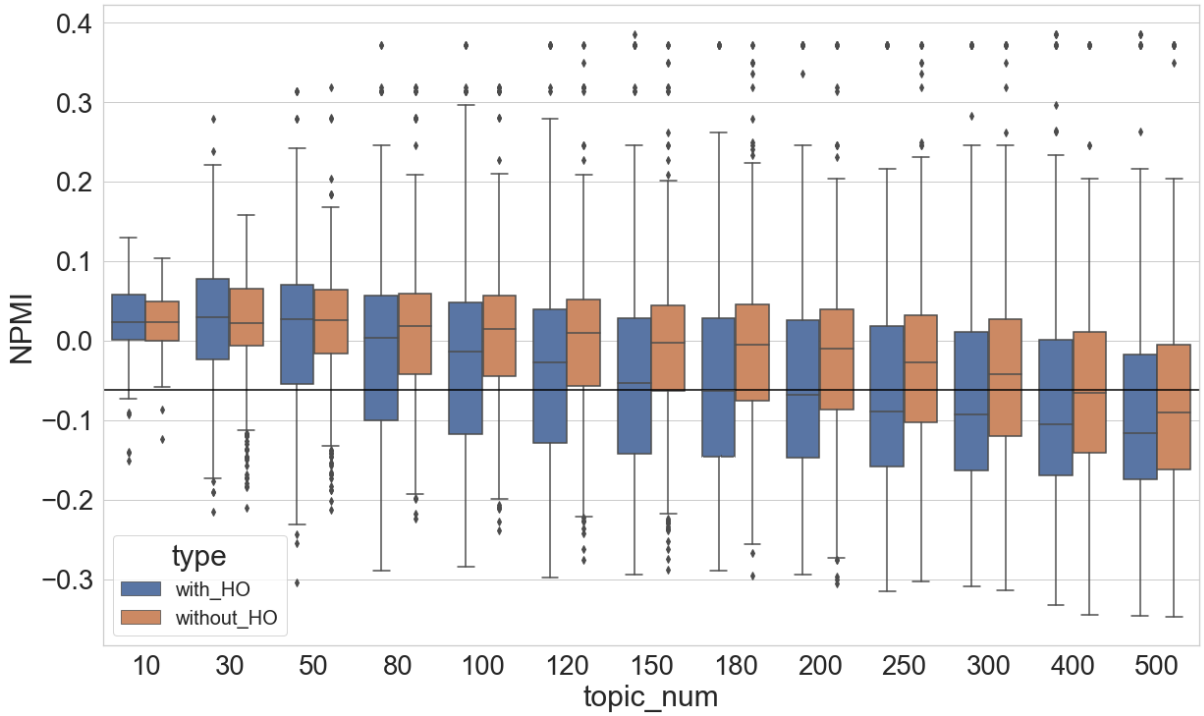

Abbildung 6.22 NPMI-Werte-Verteilung der Topics im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics (Chunk-Length 100)

Angesichts der Ergebnisse der Untersuchungen ist wie bei der Dokumentklassifikation anzunehmen, dass der Einfluss der Hyperparameter und der Hyperparameter-Optimierung auf die Topic-Kohärenz auch mit der Erhöhung der Chunk-Length abnimmt. Um diese Annahme zu bestätigen, wird eine weitere Untersuchung durchgeführt: Die 2000 Zeitungsartikel werden in Chunks zerlegt, das Setting der Chunk-Length ist $\boldsymbol{C} \in \{10, 20, 50, 80, 100, 120, 150, 180, 200, 250, 300, 350, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1700\}$. Bei jedem Setting von $\boldsymbol{C}$ werden zehn Topic-Modelle mit und ohne Hyperparameter-Optimierung trainiert. ***OB*** und ***OI*** werden hier jeweils auf 200 eingestellt, wenn die Hyperparameter-Optimierung beim Training eingesetzt werden soll. Die Anzahl der Topics wird auf 80 eingestellt. Auch andere Settings der Topic-Anzahl wurden getestet, die Ergebnisse sind ähnlich. Anschließend wird die NPMI-Werte aller Topics berechnet. Die entsprechenden Ergebnisse werden in Abbildung 6.23 visualisiert. Es ist zu erkennen, dass die Mediane und die Spannweite der NPMI-Werte-Verteilungen der zwei Gruppen bei $\boldsymbol{C} = 10$ sehr unterschiedlich sind. Ab $\boldsymbol{C} = 300$ sind die NPMI-Werte-Verteilungen ähnlich und es kann keine systematische Veränderung mehr beobachtet werden.

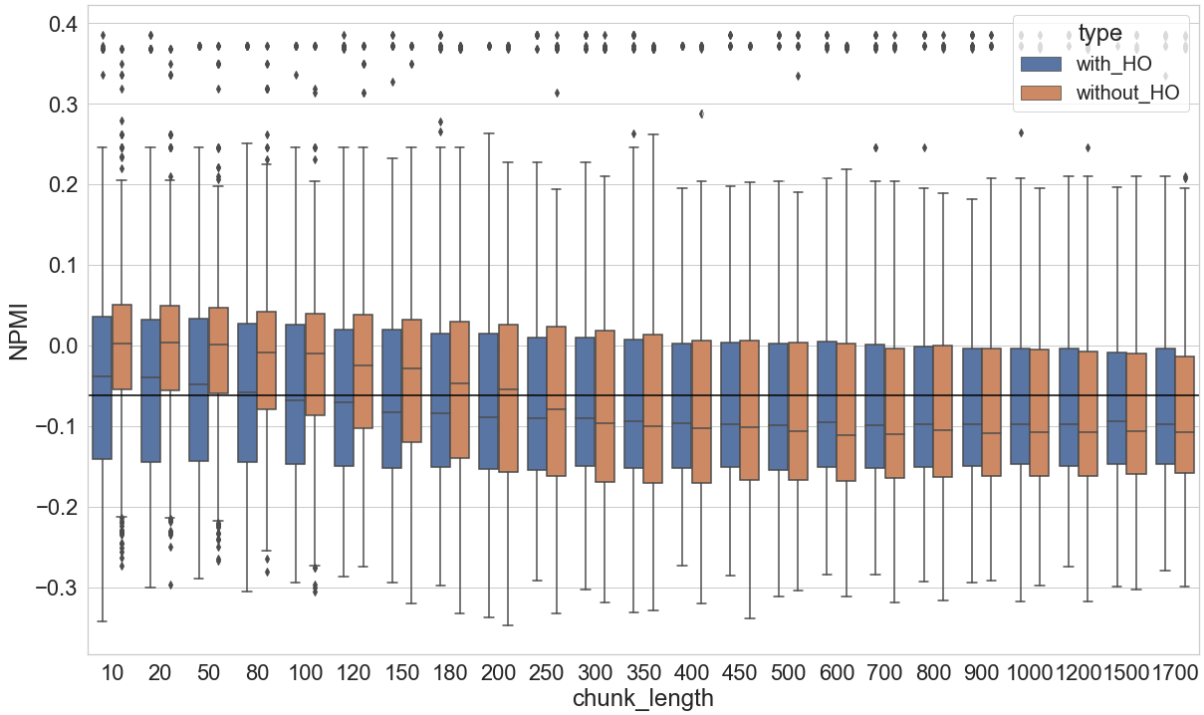

Abbildung 6.23 NPMI-Werte-Verteilung der Topics im Verhältnis zu Hyperparameter-Optimierung und Chunk-Length

In diesem Unterkapitel wird analysiert, wie sich die Hyperparameter-Optimierung auf die Qualität der trainierten Topics auswirkt. Da die Qualität der Topics in Wallach et al. (2009) nicht evaluiert wurde, sind die Ergebnisse hier eine Ergänzung zu dem Artikel. Die oben beschriebenen Tests zeigen, dass der Einfluss der Hyperparameter-Optimierung auf die Topic-Kohärenz besonders stark ist, wenn die einzelnen Chunks in einem Korpus kurz sind (in dieser Untersuchung kürzer als 200 Tokens). Dieser Einfluss verringert sich mit zunehmender Länge des Chunks.

6.4 Hyperparameter Beta

In diesem Kapitel beziehen sich die durchgeführten Experimente auf den Einfluss des Hyperparameters Beta auf die Qualität des Topic-Modells. Das Setting von Beta ist $\boldsymbol{\beta} \in$ {0,00001 (1e-05), 0,00005 (5e-05), 0,0001, 0,0005, 0,001, 0,005, 0,01, 0,02, 0,05, 0,08, 0,1, 0,2, 0,5, 0,8, 1, 2, 5, 8, 10, 20, 50}. In dieser Untersuchung wird die Anzahl der Topics zunächst auf 80 und 90 eingestellt.

6.4.1 Dokumentklassifikation

In Abbildung 6.24 werden die Klassifikationsergebnisse im Verhältnis zu Beta visualisiert. Mit der Erhöhung von Beta zeigen die Boxplots zunächst einen aufsteigenden Trend. Die beste Klassifikation wird erzielt, wenn Beta auf einen Wert zwischen 0,001 und 0,01 eingestellt wird. Das Ergebnis in dieser Untersuchung ist eine Accuracy von 0,7675 und ein F1-Wert von 0,7604. Diese Werte sind geringfügig besser als die BoW-basierte Baseline (Accuracy 0,765 / F1-Wert 0,758). Diese Verbesserung ist zwar nur marginal, wird in den anderen vorgestellten Untersuchungen aber noch nicht beobachtet. Mit der weiteren Erhöhung von Beta geht dagegen keine zusätzliche Verbesserung einher, im Gegenteil sinkt der F1-Wert zuerst langsam bis auf 0,7 ab, wenn Beta von 0,01 auf 0,1 erhöht wird. Die Klassifikation verschlechtert sich anschließend deutlich schneller, wenn Beta von 0,1 auf 5 erhöht wird. Außerdem ist hier zu beobachten, dass die Abweichung zwischen den zehn Topic-Modellen mit dem gleichen Parameter-Setting größer wird. Offenbar wird der Trainings-Prozess hier wegen des Settings von Beta durch die zufällige Initialisierung bei der Zuweisung der Topics und des Gibbs-Samplings stärker beeinflusst. Der F1-Wert sinkt von ca. 0,7 bis auf Werte unter 0,1 ab. Bei einer weiteren Erhöhung von Beta bis 50 bleiben die F1-Werte unterhalb von 0,1, was darauf hinweist, dass die trainierten Topic-Modelle für die Klassifikationsaufgabe nicht geeignet sind.

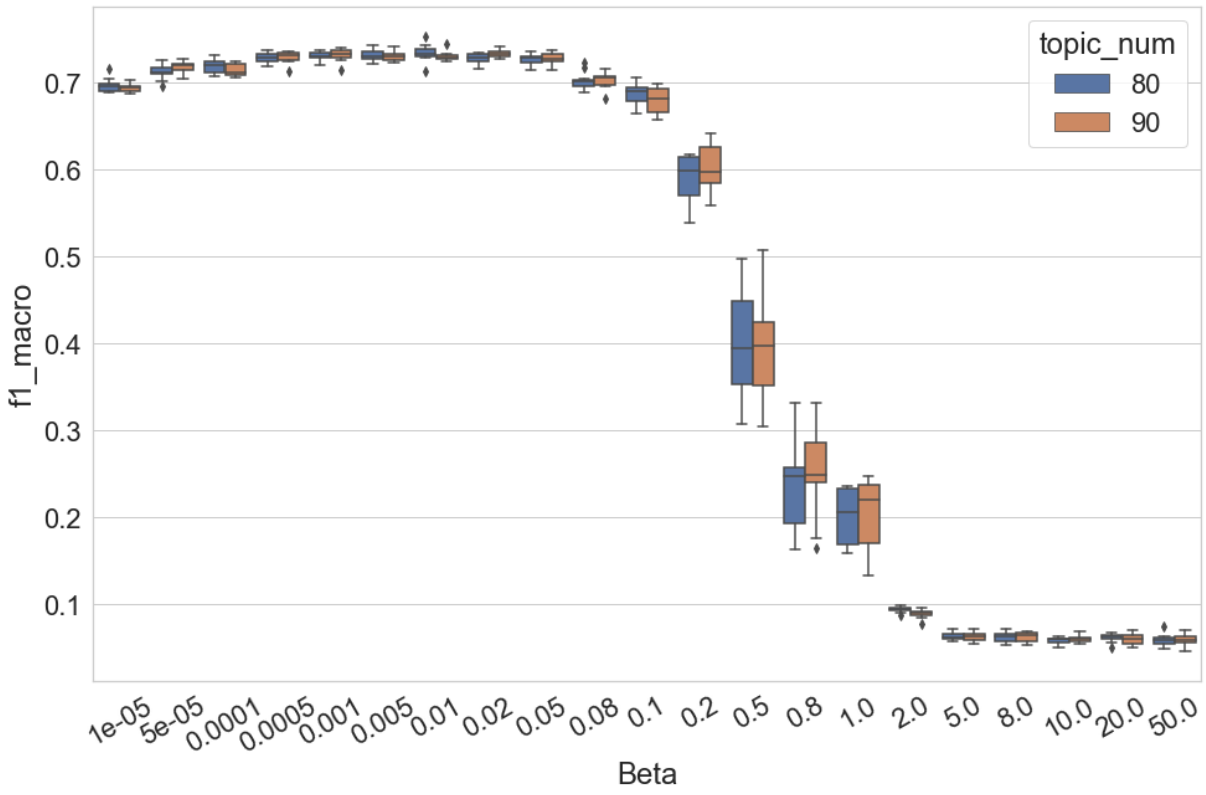

Abbildung 6.24 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Beta

Auch wenn die Anzahl der Topics stärker variiert wird, können ähnliche Verteilungen der F1-Werte in Abbildung 6.25 beobachtet werden. Alle Kurven steigen zuerst auf und erreichen den höchsten Punkt, wenn Beta zwischen 0,001 und 0,01 eingestellt wird. Wenn der Wert von Beta größer als 0,1 eingestellt wird, sinken sämtliche Kurven rasch ab, die Klassifikation erfolgt in sehr geringer Qualität. In der Visualisierung ist darüber hinaus zu erkennen, dass die Topic-Modelle mit mehr Topics bessere Klassifikationsergebnisse erzielen, wenn der Wert von Beta kleiner als 0,1 ist. Allerdings können diese Modelle bei einer weiteren Erhöhung von Beta die Dokumente schlechter klassifizieren. Offenbar werden die Topic-Modelle durch eine solche Erhöhung stärker beeinflusst, wenn sie eine hohe Anzahl von Topics enthalten.

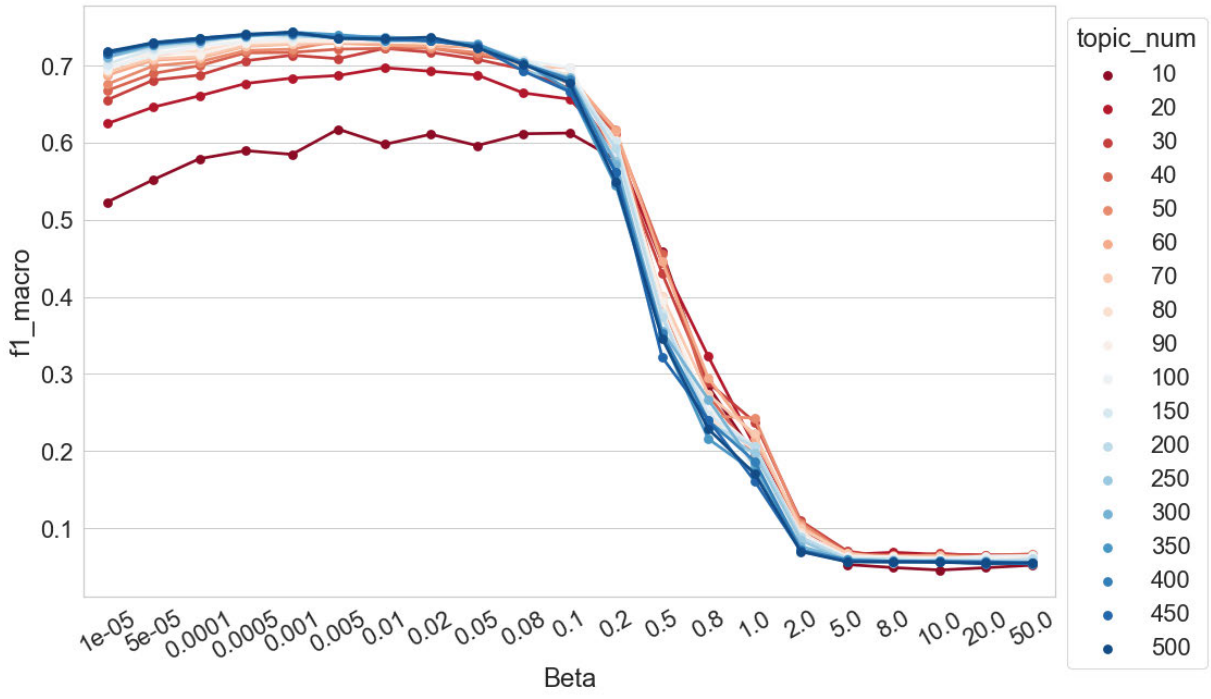

Abbildung 6.25 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Beta und Anzahl der Topics

6.4.2 Topic-Kohärenz

Abbildung 6.26 zeigt die Verteilungen der NPMI-Werte aller Topics im Verhältnis zu Beta, wobei die Anzahl der Topics zuerst auf $\boldsymbol{T}$ = 80 und 90 eingestellt wird. Wenn Beta von 1e-05 (0,00001) auf 0,001 erhöht wird, steigen die Mediane der Verteilungen auf und die Spannweiten

der Verteilungen sind größer. Bei einer weiteren Erhöhung von Beta bis auf 0,02 sind die NPMI-Werte der besten kohärenten Topics höher, obwohl die Mediane der Verteilungen absinken. Mit einer weiteren Erhöhung von Beta (auf bis zu 50) wird der Bereich der Verteilungen schrittweise auf Werte zwischen ca. 0,05 und -0,1 reduziert. Die Mediane fallen zunächst unter den NPMI-Kontrollwert und kehren dann über diesen zurück, wenn Beta von 0,001 auf 50 erhöht wird.

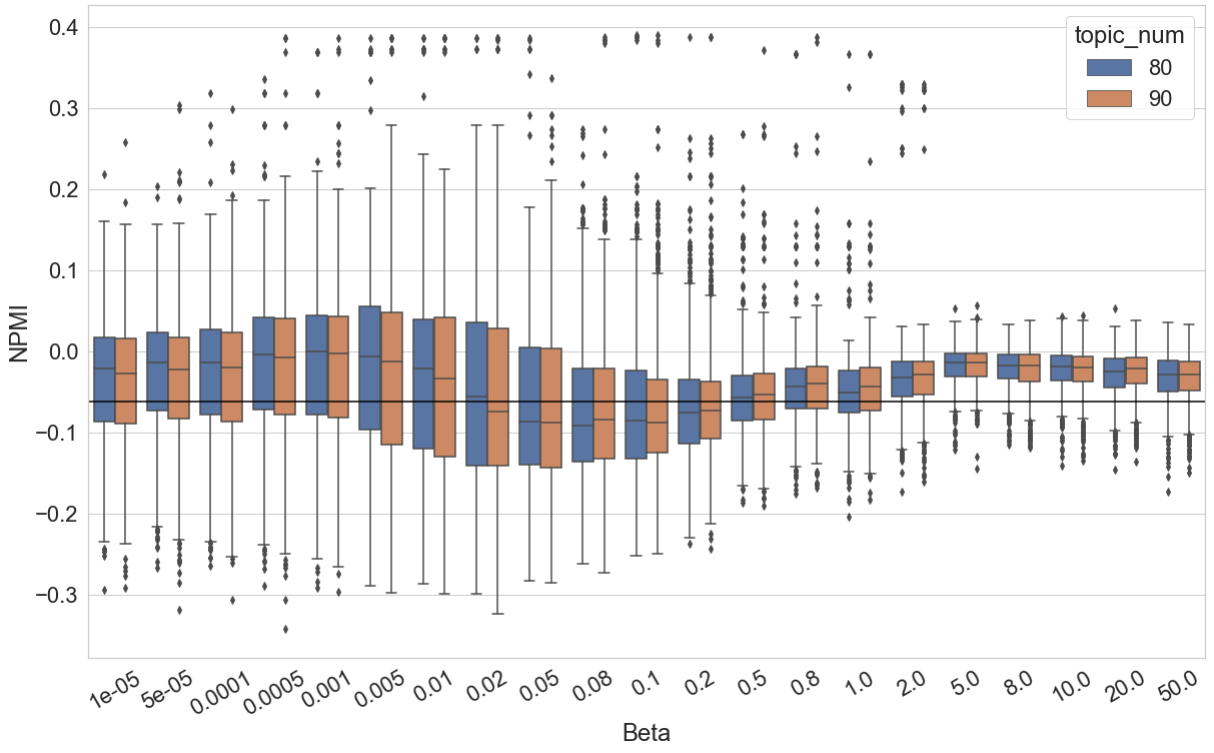

Abbildung 6.26 NPMI-Werte-Verteilung der Topics im Verhältnis zu Beta

Wenn die Anzahl der Topics stärker variiert wird, ist die Situation ähnlich (Abbildung 6.27): Bis zu $\boldsymbol{\beta}$ = 0,001 zeigen die Verteilungen insgesamt einen aufsteigenden Trend und die Spannweiten der Verteilungen sind größer. Ab $\boldsymbol{\beta}$ = 0,02 beginnen sich die Verteilungen bei jedem Setting der Topic-Anzahl zu verengen. Besonders bei den Topic-Modellen mit einer hohen Anzahl von Topics (z. B. $\boldsymbol{T}$ = 450 oder 500) ist zu beobachten, dass die Verengung der NPMI-Werte-Verteilungen bei $\boldsymbol{\beta}$ = 0,005 beginnt. Ab $\boldsymbol{\beta}$ = 5 ist der Unterschied in den Verteilungen nicht mehr deutlich sichtbar. Es stellt sich allerdings die Frage nach der Bedeutung der Verengung der NPMI-Werte-Verteilung. Das Ergebnis scheint darauf hinzudeuten, dass der Wert von Beta beim Training des Topic-Modells auch höher angesetzt werden könnte. Dadurch verliert man zwar die kohärentesten Topics, gleichzeitig gibt es aber

auch weniger nicht-kohärente Topics. Vor allem sind die NPMI-Werte-Verteilungen sehr stabil und werden nur in geringem Maß durch die Anzahl der Topics beeinflusst, wenn Beta (in dieser Untersuchung) auf einen Wert größer als 5 eingestellt wird. Um diese Annahme zu überprüfen, werden die Topic-Modelle aus einer zusätzlichen Perspektive kontrolliert. Der Parameter Beta entspricht dem Hyperparameter der Topic-Wort-Verteilung und kontrolliert, wie oft jedes Wort in einem Topic vorkommt. Deshalb ist es sinnvoll zu überprüfen, wie viele Wörter jedem Topic zugeordnet sind.

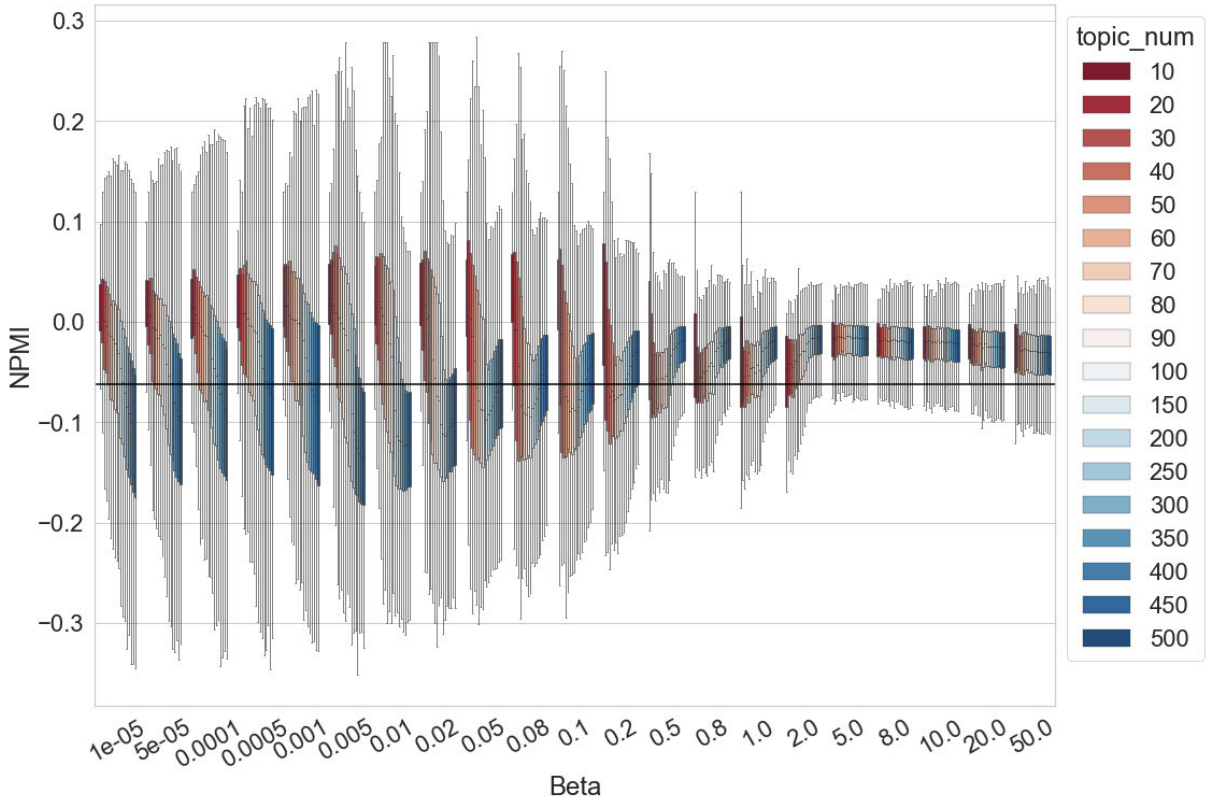

Abbildung 6.27 NPMI-Werte-Verteilung der Topics im Verhältnis zu Beta und Anzahl der Topics

In Abbildung 6.28 wird die Anzahl der zugeordneten Wörter in jedem Topic[72] visualisiert. Als klare Veränderung lässt sich feststellen, dass die Anzahl der in einem Topic enthaltenen Tokens bis zu einem Faktor von ca. 10 variiert, wenn Beta kleiner als 0,001 ist. Mit der weiteren Erhöhung von Beta (bis zu $\boldsymbol{\beta} = 2$) wird die Differenz zwischen den Topics größer. Die Anzahl der in einem Topic enthaltenen Tokens kann von einigen Hundert über Tausende bis hin zu Zehntausenden reichen. Ab $\boldsymbol{\beta} = 5$ hat sich die Situation polarisiert. Unabhängig davon, wie viele Topics ein Topic-Modell enthält, gibt es nur zwei Gruppen von Topics im Modell. Die Topics

[72] Mithilfe von „Mallet – Topic model diagnostics“ können diese Daten ermittelt werden.

enthalten entweder mehr als Hunderttausend Tokens oder nur ca. ein Hundert. Wenn ein Topic im Vergleich zu anderen eine sehr geringe Anzahl von Tokens enthält, ist es möglicherweise nicht zuverlässig, weil wir nicht über ausreichend viele Beobachtungen verfügen, um seine Wortverteilung valide zu evaluieren. Umgekehrt ist ein Topic mit einer sehr großen Anzahl von Tokens auch nicht ideal. So enthält zum Beispiel in einem Topic-Modell mit 50 Topics bei $\boldsymbol{\beta}$ = 50 ein Topic 1.180.922 Wörter, während alle anderen 49 Topics nur 75 bis 129 Tokens umfassen. In diesem Fall werden fast alle Wörter im Untersuchungskorpus einem Topic zugeordnet. Das große Topic und das Topic-Modell sind für die Exploration des Untersuchungskorpus daher offensichtlich nicht sinnvoll. Das Ergebnis zeigt, dass beim Training des Topic-Modells der Wert von Beta nicht deutlich zu groß (in dieser Untersuchung größer als 5) eingestellt werden sollte. Sowohl die Topic-Modeling-basierten Klassifikationsergebnisse als auch die Qualität der Topics selbst sind nicht befriedigend, wenn Beta zu groß ist.

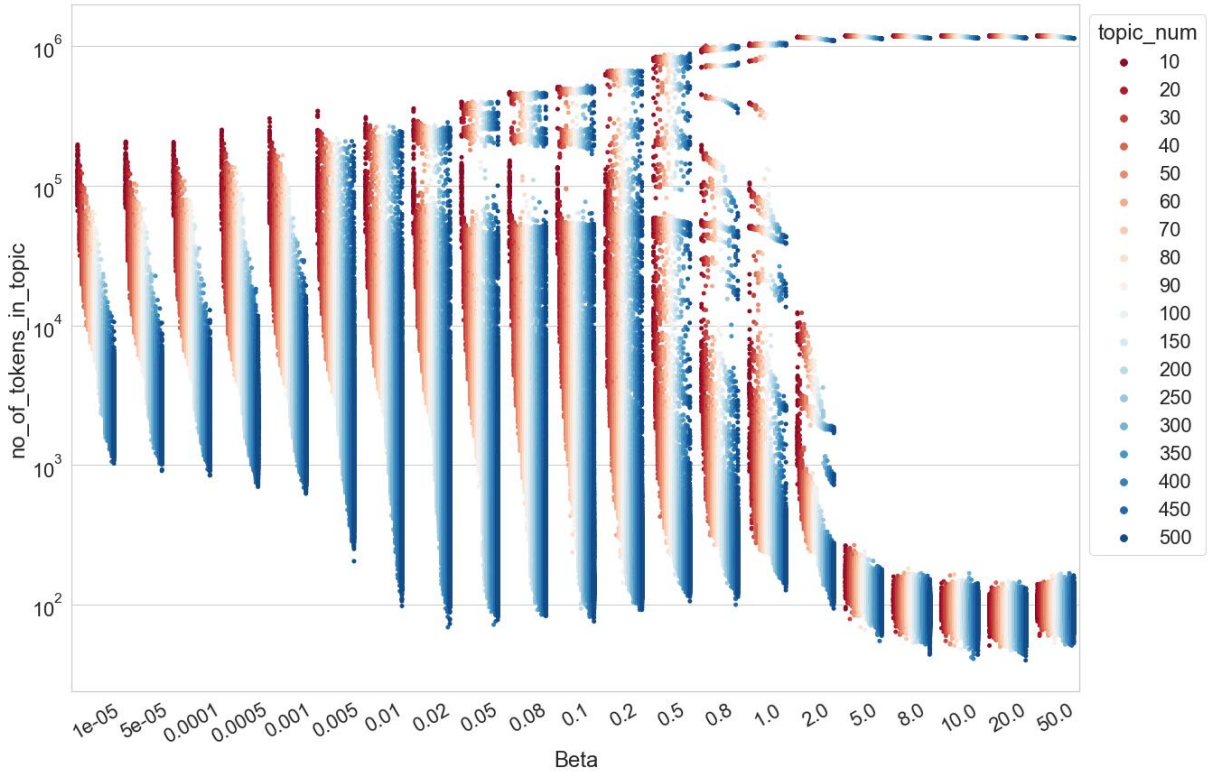

Abbildung 6.28 Anzahl der zugeordneten Tokens in jedem Topic im Verhältnis zu Beta und Anzahl der Topics (y-Achse logarithmisch skaliert)

6.5 Iteration des Gibbs-Samplings

In diesem Kapitel bezieht sich das durchgeführte Experiment sich auf den Einfluss der Iteration des Gibbs-Samplings ***I*** auf die Qualität des Topic-Modells. Das Setting der Anzahl der Iterationen ist ***I*** ∈ {20, 50, 100, 200, 300, 400, 500, 700, 1000, 1200, 1500, 1800, 2000, 2200, 2500, 2800, 3000}, die Anzahl der Topics wird wie zuvor auf 80 und 90 eingestellt.

6.5.1 Dokumentklassifikation

In Abbildung 6.29 werden die Klassifikationsergebnisse visualisiert. Die Klassifikation zeigt allmählich bessere Ergebnisse, wenn ***I*** von 20 auf 200 erhöht wird. Die F1-Werte sind um etwa 0,1 gestiegen. Ab ***I*** = 300 kann keine systematische Änderung der Klassifikationsergebnisse mehr beobachtet werden, die F1-Werte liegen meist zwischen 0,72 und 0,74. An einigen Stellen (z. B. ***I*** = 200 oder ***I*** = 300) sind die Klassifikationsergebnisse deutlich unterschiedlich, wenn die Anzahl der Topics auf 80 und 90 eingestellt wird. Es handelt sich hier aber offenbar nicht um einen systematischen Unterschied. In der Mehrzahl der Fälle existiert kein systematischer Unterschied zwischen ***T*** = 80 und ***T*** = 90.

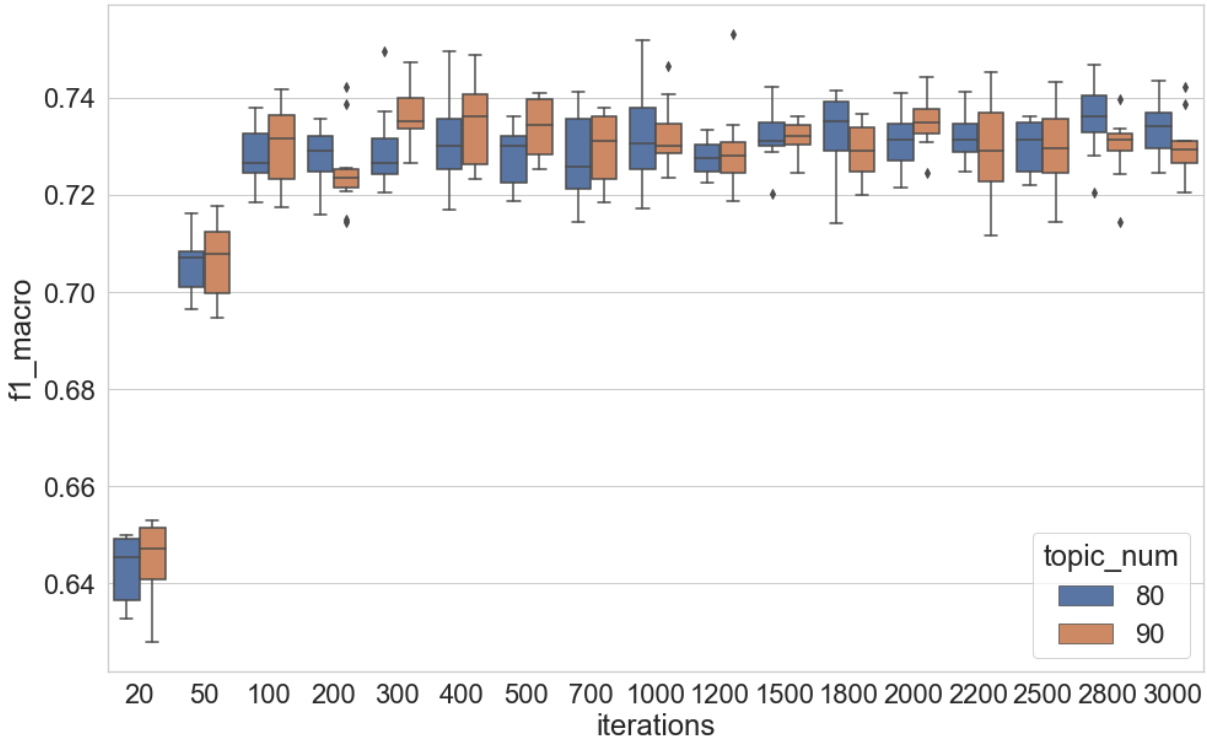

Abbildung 6.29 F1 (Makro)-Werte der Topic-Modell-basierten Dokumentklassifikation im Verhältnis zur Iteration des Gibbs-Samplings

Auch, wenn die Anzahl der Topics stärker variiert wird, bleibt der aufsteigend Trend unabhängig von der Anzahl der Topics stabil, wenn ***I*** von 20 auf 200 erhöht wird (Abbildung 6.30). Ab ***I*** = 300 kann keine deutliche Verbesserung beobachtet werden. Je höher die Anzahl der Topics ist, desto kleiner ist die Schwankung der Kurve. Die F1-Werte bleiben meist zwischen 0,70 und 0,75. In diesem Test wird kein Ergebnis beobachtet, das besser als die Baseline ist. Die Accuracy und der F1-Wert können höchstens 0,7640 und 0,7551 erreichen, was geringfügig unter den Werten der Baseline (0,765 / 0,758) liegt.

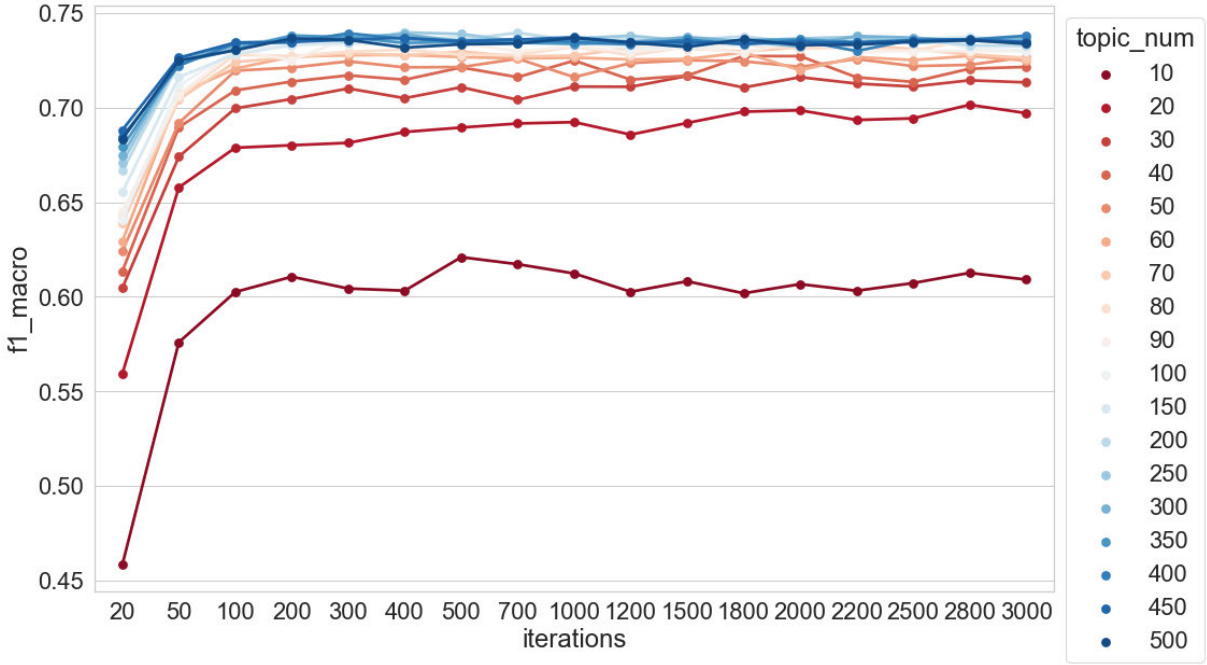

Abbildung 6.30 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zur Iteration des Gibbs-Samplings und Anzahl der Topics

6.5.2 Topic-Kohärenz

Abbildung 6.31 zeigt die Verteilung der NPMI-Werte im Verhältnis zur Iteration des Gibbs-Samplings. Zunächst ist zu beobachten, dass die Mediane der NPMI-Werte-Verteilungen konstant über dem NPMI-Kontrollwert liegen, sie verändern sich mit der Erhöhung von ***I*** nicht deutlich. Bei ***T*** = 80 sind die Mediane der NPMI-Werte-Verteilungen immer etwas höher als die Mediane bei ***T*** = 90. Der Wertebereich der NPMI-Werte-Verteilungen ist jedoch bei ***T*** = 80 und ***T*** = 90 nicht deutlich unterschiedlich. Außerdem ist zu erkennen, dass die Spannweite der Verteilungen größer geworden ist, wenn ***I*** von 20 auf 200 erhöht wird. Das bedeutet, dass die kohärenten Topics mit der Erhöhung von ***I*** stets eine höhere Qualität aufweisen. Um diese

Änderung deutlicher darzustellen, werden die NPMI-Werte-Verteilungen der kohärentesten 10 % der Topics in Abbildung 6.32 erneut gesondert visualisiert. Alle Werte sind höher als der NPMI-Kontrollwert und eine klare Steigerung kann beobachtet werden, wenn ***I*** von 20 auf 200 erhöht wird. Bei einer weiteren Erhöhung von ***I*** ist dann aber keine deutliche systematische Änderung mehr zu erkennen.

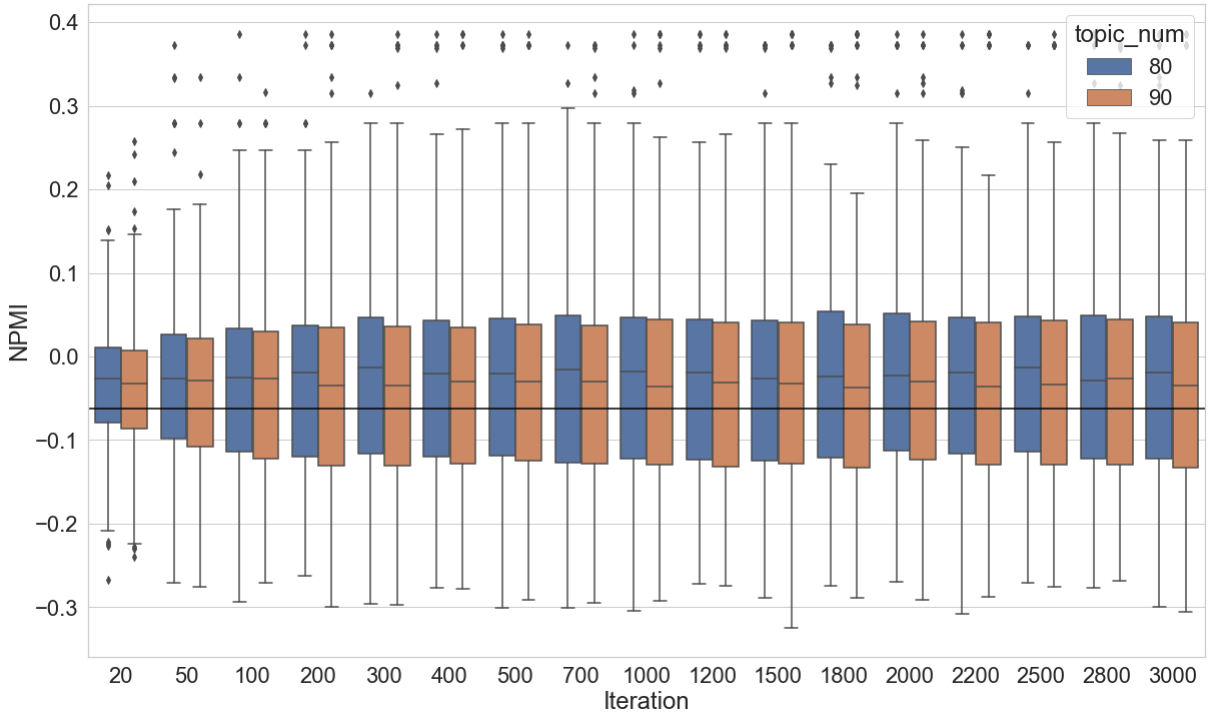

Abbildung 6.31 NPMI-Werte-Verteilung der Topics im Verhältnis zur Iteration des Gibbs-Samplings

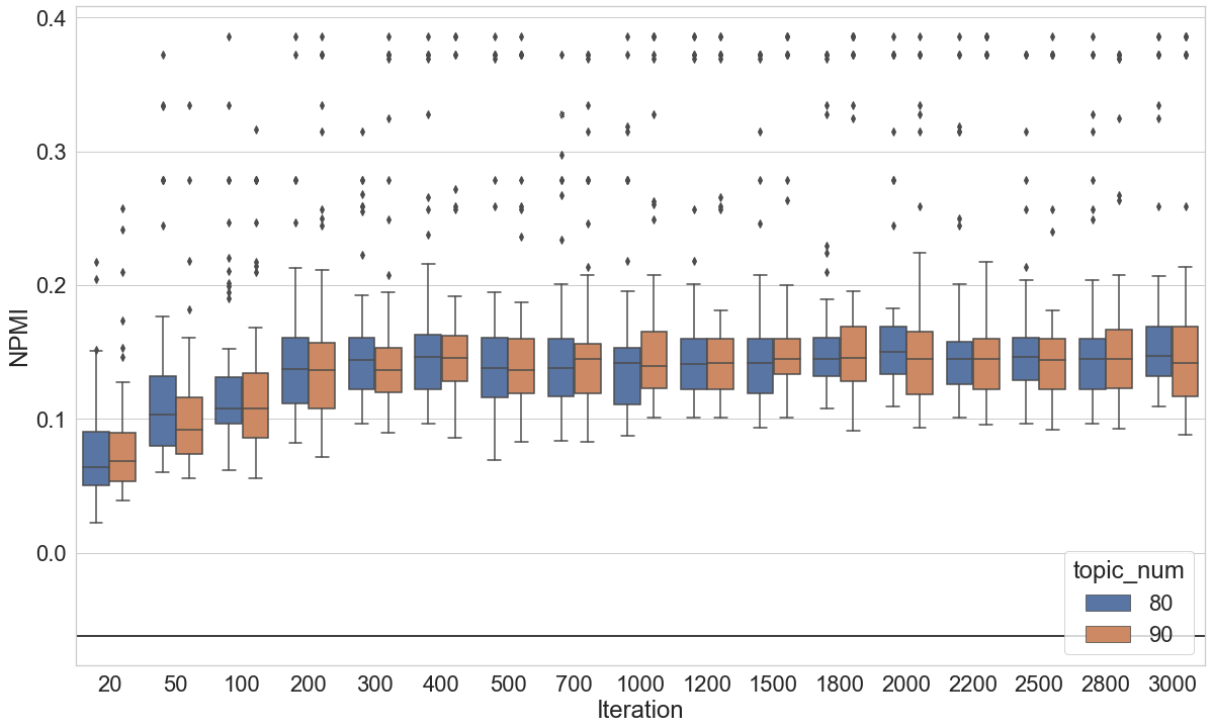

Abbildung 6.32 NPMI-Werte-Verteilung der Top 10 % der kohärentesten Topics im Verhältnis zur Iteration des Gibbs-Samplings

Wenn die Anzahl der Topics in höherem Maße variiert wird, stellt sich die Situation ähnlich dar (Abbildung 6.33): Es gibt keine systematische Änderung beim Setting der Anzahl der Topics, wenn ***I*** von 300 auf 3000 erhöht wird. Wenn ***I*** von 20 auf 300 erhöht wird, steigen die Minimal- und die Maximalwerte der Verteilungen jeweils von ca. -0,35 auf etwa -0,3 und von ca. 0,15 auf knapp 0,3. Die Verteilungen der Top 10 % der kohärentesten Topics zeigen ebenfalls einen aufsteigenden Trend, wenn ***I*** von 20 auf 300 erhöht wird (Abbildung 6.34). Der aufsteigende Trend ist jedoch bei einer weiteren Steigerung des Werts von ***I*** nicht mehr deutlich. In dieser Untersuchung wird wie bereits bei den vorherigen Experimenten beobachtet, dass die Topic-Modelle mit vielen Topics (z. B. bei $\boldsymbol{T}$ = 500) deutlich mehr nicht-kohärente Topics enthalten.

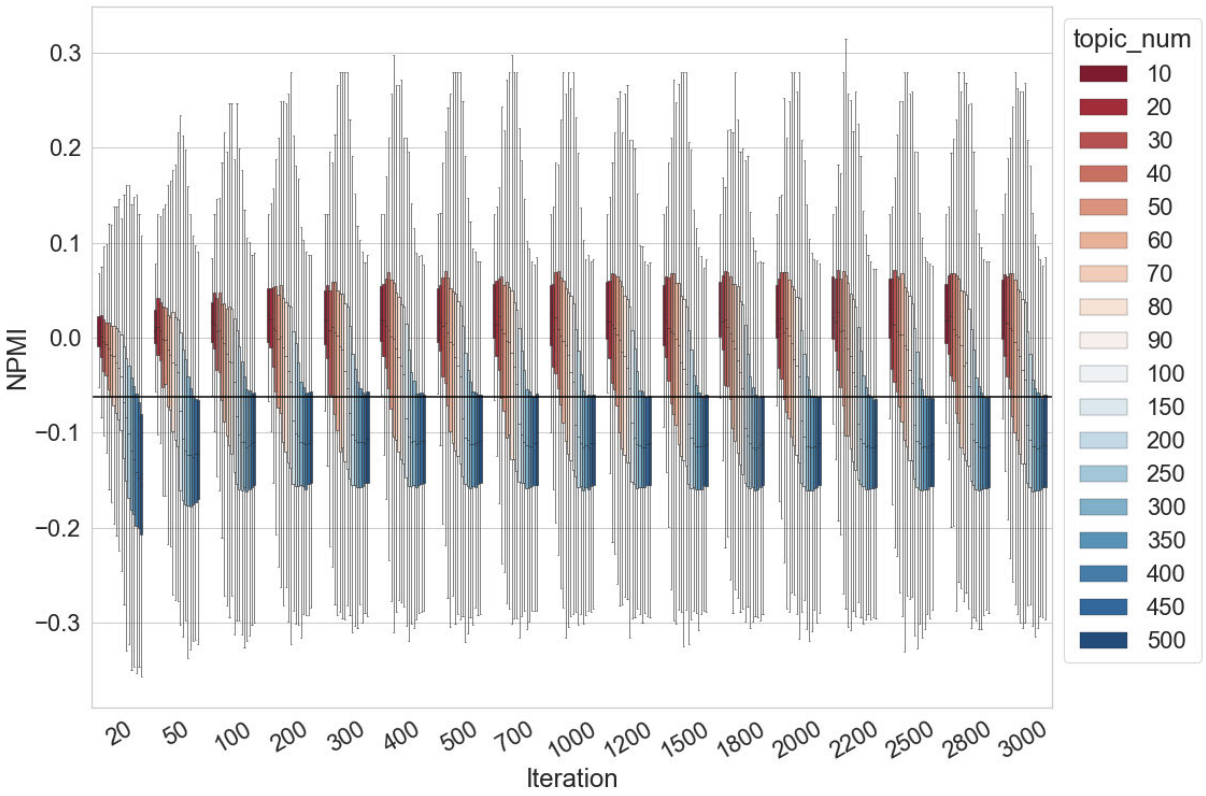

Abbildung 6.33 NPMI-Werte-Verteilung der Topics im Verhältnis zur Iteration des Gibbs-Samplings und Anzahl der Topics

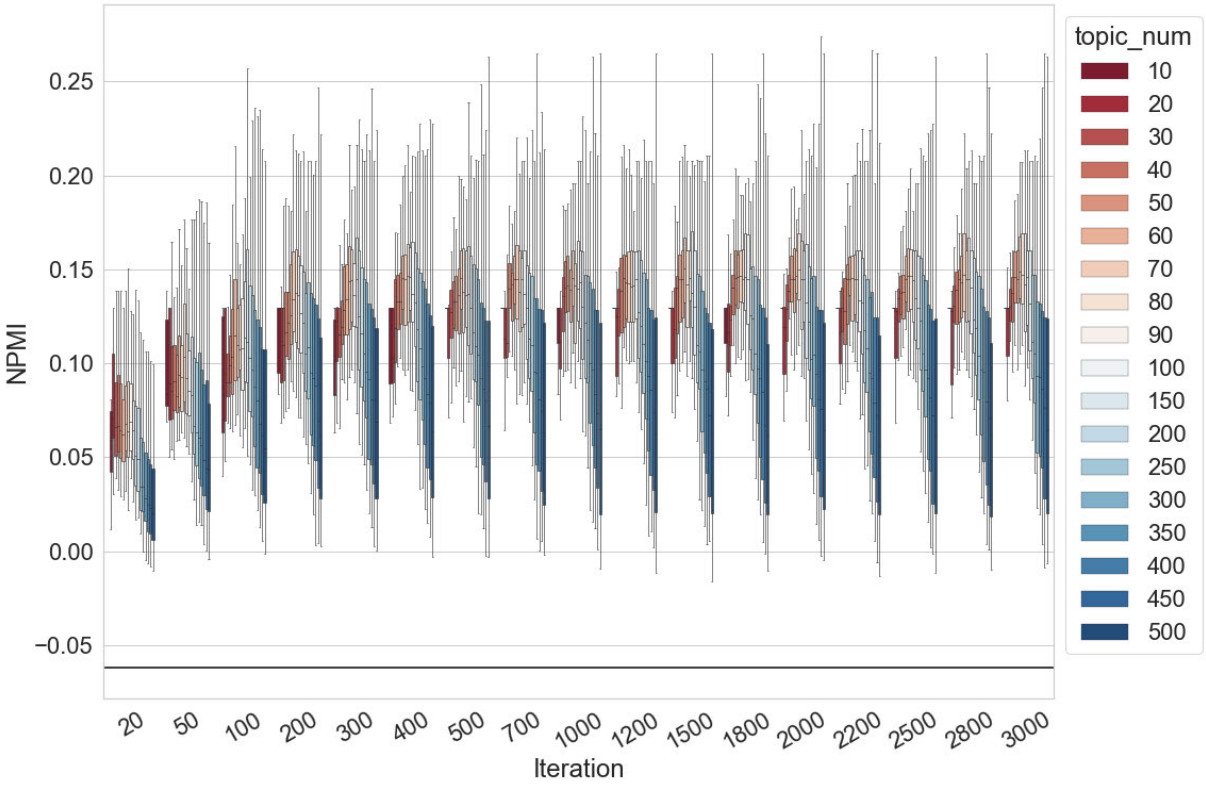

Abbildung 6.34 NPMI-Werte-Verteilung der Top 10 % der kohärentesten Topics im Verhältnis zur Iteration des Gibbs-Samplings und Anzahl der Topics

Aus den Experimenten in diesem Kapitel lässt sich ableiten, dass unabhängig von der Anzahl der Topics eine hohe Anzahl von Iterationen beim Training eines Topic-Modells eine erfolgsversprechende Methode ist. Durch die Erhöhung der Zahl der Iterationen kann die Qualität des Topic-Modells allerdings nicht unendlich verbessert werden, ab einem bestimmten Wert (in diesem Test ab 300 Iterationen) lässt sich keine deutliche Änderung mehr beobachten. In Raftery & Lewis (1991) wird vorgeschlagen: „Reasonable accuracy may often be achieved with 5000 iterations or less; this can frequently be reduced to less than 1000 if the posteriror tails are known to be light.“ Hinsichtlich der vorgegebenen Einstellungen in Bezug auf die Iterationen in MALLET (***I*** = 1000), der Empfehlungen der Fachleute und auch der gerade beschriebenen Beobachtung sollten Topic-Modelle also mit mindestens 1000 Iterationen trainiert werden. Aus diesem Grund wird die Anzahl der Iterationen in allen Untersuchungen in dieser Arbeit immer auf 2000 eingestellt, wenn die Anzahl der Iterationen eine Kontrollvariable ist und konstant gehalten werden soll.

6.6 Chunk-Length

6.6.1 Chunking auf Paragraph-Ebene

In diesem Kapitel bezieht sich das Experiment auf den Einfluss der Chunk-Length des Dokuments auf das Topic Modeling. Dem Vorschlag in Algee-Hewitt et al. (2015) folgend, werden die Texte in Paragraphen zerlegt, um Dokumente für das Training der Topic-Modellen vorzubereiten. Um den Einfluss der Chunk-Length zu testen, wird eine kleinste Längeneinheit des Dokuments ***C*** eingeführt. Jeder Zeitungsartikel wird zuerst in Paragraphen zerlegt. Wenn ein Paragraph P_n ***C*** Wörter oder mehr als ***C*** Wörter enthält, wird er als ein Chunk (ein Dokument für das Topic-Modeling-Verfahren) gespeichert. Wenn ein Paragraph P_n weniger als ***C*** Wörter enthält, wird der nächste Paragraph P_{n+1} zu P_n hinzugefügt, um einen Chunk aufzubauen. Wenn die beiden Paragraphen insgesamt noch immer weniger als ***C*** Wörter enthalten, wird der folgende Paragraph P_{n+2} für den Aufbau eines Chunks hinzugefügt und so weiter.

Das Setting der kleinsten Längeneinheit ist ***C*** ∈ {10, 20, 50, 80, 100, 120, 150, 180, 200, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1700}. Je größer die kleinste

Segmentlänge ***C*** ist, desto eher ist zu erwarten, dass die gesamte Anzahl der Dokumente geringer ausfällt. In Abbildung 6.35 wird die Veränderung der Anzahl der Dokumente und der durchschnittlichen Dokumentlänge mit der Erhöhung von ***C*** dargestellt. Die Anzahl der Chunks (die grüne Linie) wurde von knapp 50.000 auf ca. 3000 reduziert, während die durchschnittliche Chunk-Length (die blaue Linie) von ca. 80 Tokens auf etwa 1200 zunimmt. Wenn ***C*** auf 1700 eingestellt wird, werden die ursprünglichen 2000 Zeitungsartikel in 2892 Chunks zerlegt. Das bedeutet, dass mehr als die Hälfte der Zeitungsartikel weniger als 1700 Wörter enthalten, sie werden nicht zerlegt. Der Wert von ***C*** wird lediglich bis auf maximal 1700 gesteigert, da eine weitere Erhöhung wenig Sinn ergibt.

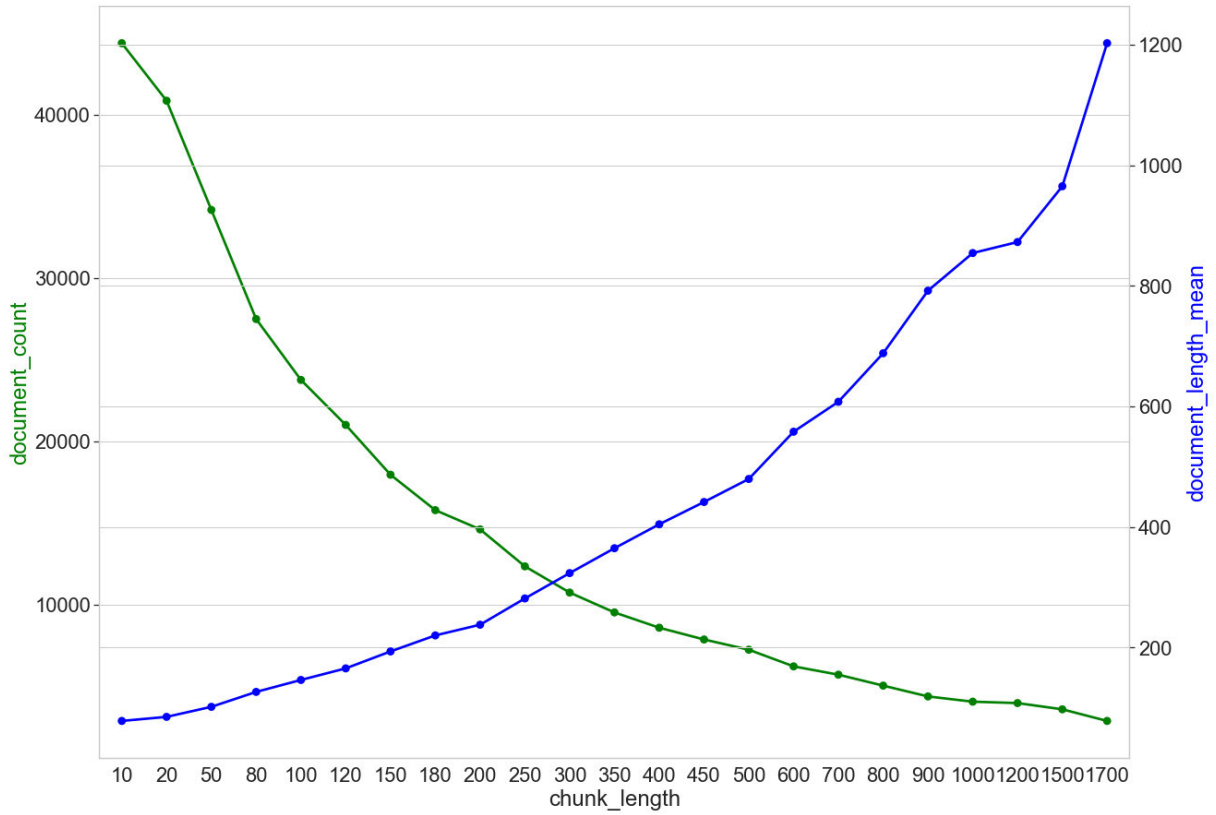

Abbildung 6.35 Anzahl der Dokumente (document_count) und die durchschnittliche Dokumentlänge (document_length_mean) im Verhältnis zur Chunk-Length ***C***

6.6.1.1 Dokumentklassifikation

Ein sehr wichtiger Punkt, der vor der Vorstellung der Testergebnisse hervorgehoben werden muss, ist, dass sich die Untersuchungen in Bezug auf die Chunk-Length in einer Hinsicht von den vorherigen Tests unterscheiden. In den bisherigen Experimenten konnte der Unterschied

der Qualität zwischen den unterschiedlichen Topic-Modellen durch den Unterschied zwischen den unterschiedlichen Klassifikationsergebnissen widergespiegelt werden. Durch Veränderungen der Einstellungen der Parameter werden verschiedene Topic-Modelle auf dem gleichen Korpus trainiert. Deshalb beziehen sich die Klassifikationsaufgaben immer auf die Klassifikation der 2000 Zeitungsartikel und ein gutes Ergebnis kann auf eine hohe Qualität des Modells hinweisen. Da die Topic-Modelle in diesem Unterkapitel aber auf unterschiedliche Sammlungen von Chunks trainiert werden, sind die Klassifikationsaufgaben bei jedem Setting von $\boldsymbol{C}$ unterschiedlich (z. B. müssen bei $\boldsymbol{C}$ = 100 knapp 50.000 Chunks zehn thematischen Klassen zugeordnet werden, während bei $\boldsymbol{C}$ = 1700 weniger als 3000 Chunks klassifiziert werden müssen). Beim gleichen Setting von $\boldsymbol{C}$ zeigt das bessere Klassifikationsergebnis eine höhere Qualität des Topic-Modells an. Bei unterschiedlichen Settings von $\boldsymbol{C}$ kann der Unterschied in den Klassifikationsergebnissen die Differenzen in der Modellqualität jedoch nicht repräsentieren, da die Klassifikationsaufgaben unterschiedlich sind. Die in den folgenden Abbildungen dargestellten Ergebnisse können deshalb lediglich beschreiben, wie sich die Topic-Modeling-basierten Klassifikationsergebnisse verändern, wenn das gleiche Korpus unterschiedlich zerlegt wird.

In Abbildung 6.36 ist die F1-Verteilung der zehn Einzeldurchläufe der jeweiligen Settings durch Boxplots dargestellt. Die Anzahl der Topics wird zuerst auf 80 und 90 eingestellt. Die Klassifikationsergebnisse zeigen einen aufsteigenden Trend, wenn $\boldsymbol{C}$ von 10 auf 1000 erhöht wird. Ab $\boldsymbol{C}$ = 1000 kann eine weitere Verbesserung nicht mehr beobachtet werden. Die Kurve senkt sich dabei mit einer Steigerung des Werts von $\boldsymbol{C}$ ab. An dieser Stelle muss betont werden, dass die Verschlechterung der Klassifikationen sehr wahrscheinlich auf die große Anzahl von kurzen Texten zurückzuführen ist. Wenn $\boldsymbol{C}$ auf 1700 eingestellt wird, wird z. B. ein Zeitungsartikel mit 1750 Tokens in zwei Chunks zerlegt. Das erste Chunk enthält mindestens 1700 Tokens, während das zweite Chunk weniger als 50 enthält. In der Tat existieren mehr als 600 Dokumente, die weniger als 200 Tokens enthalten, während das gesamte Korpus 2892 Chunks umfasst. In Abbildung 6.36 ist zu beobachten, dass die Topic-Modeling-basierte Klassifikation auf kurzen Dokumenten unzureichende Ergebnisse erbringt. Dies erklärt auch die Verschlechterung der Klassifikationsergebnisse ab $\boldsymbol{C}$ = 1000. Das Ergebnis zeigt, dass die Topic-Modeling-basierte Klassifikation umso bessere Ergebnisse erzielt, je länger die Chunks sind, wenn nicht durch die Segmentierung eine zu hohe Anzahl an kurzen Dokumenten produziert wird.

Auch wenn die Anzahl der Topics in einem höheren Maße variiert wird, bleibt der aufsteigende Trend unabhängig von der Anzahl der Topics stabil, wenn ***C*** von 10 auf 1000 erhöht wird (Abbildung 6.37). Je höher die Anzahl der Topics ist, desto besser sind die Klassifikationsergebnisse. Die Accuracy und der F1-Wert erreichen in diesem Test Höchstwerte von 0,7669 und 0,7615.

In den Abbildung 6.36 undAbbildung 6.37 wird zusätzlich eine grüne Linie visualisiert, die als Baseline ausschließlich für diese Untersuchung dient. Während die Baseline (Accuracy: 0,765 / F1-Wert: 0,758) durch eine Klassifikationsaufgabe von 2000 Dokumenten erstellt wird, beziehen sich die Klassifikationen hier nicht mehr auf dasselbe Korpus, sondern auf die segmentierten Korpora. Deshalb wird BoW-basierte Klassifikation auf den 23 segmentierten Korpora nochmals durchgeführt. Die Klassifikationen erfolgten wie zuvor als 10-fache Kreuzvalidierung mit linearer SVM. Interessanterweise kann in den beiden Abbildungen beobachtet werden, dass die Topic-Modeling-basierte Dokumentklassifikation besser als die BoW-basierte Klassifikation funktioniert. Wenn die Anzahl der Topics ***T*** größer als 100 ist, sind die Klassifikationen bei jedem Setting der Chunk-Length besser als die Baseline. Es ist also festzustellen, dass Topic-Modelle ebenfalls eine gute Möglichkeit sein können, um Dokumente zu klassifizieren.

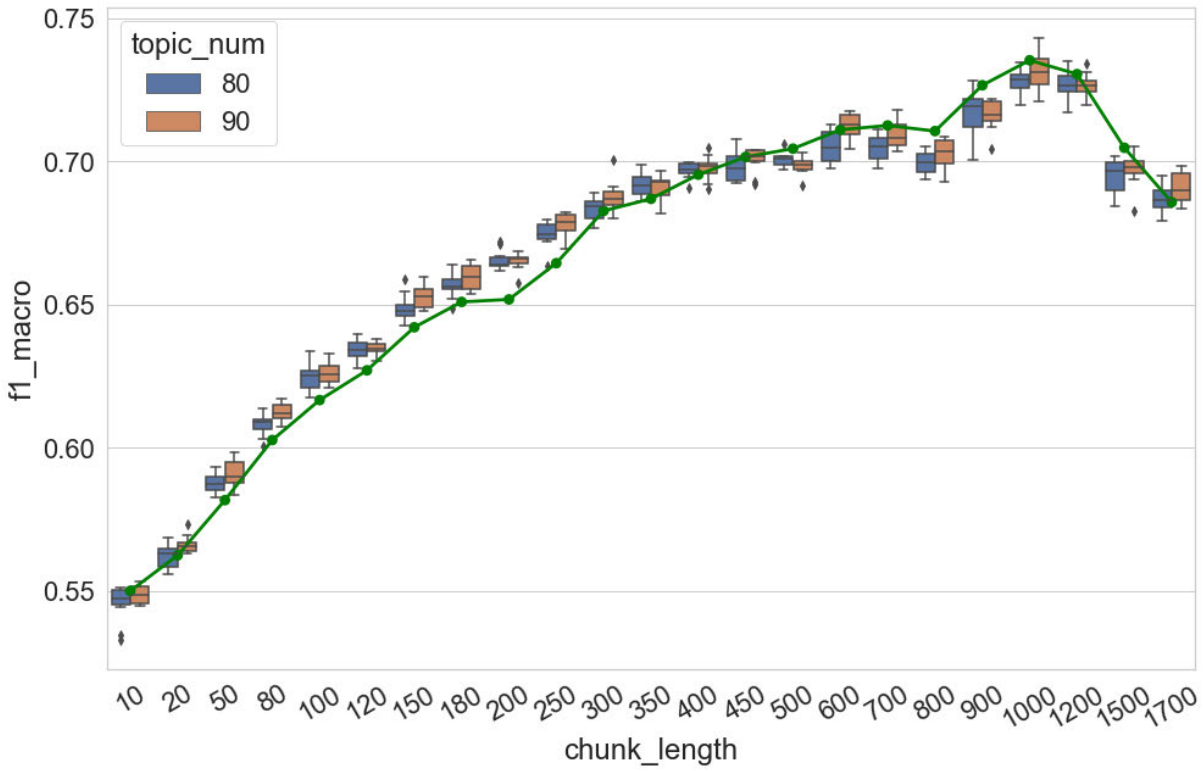

Abbildung 6.36 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation im Verhältnis zur Chunk-Length ***C***

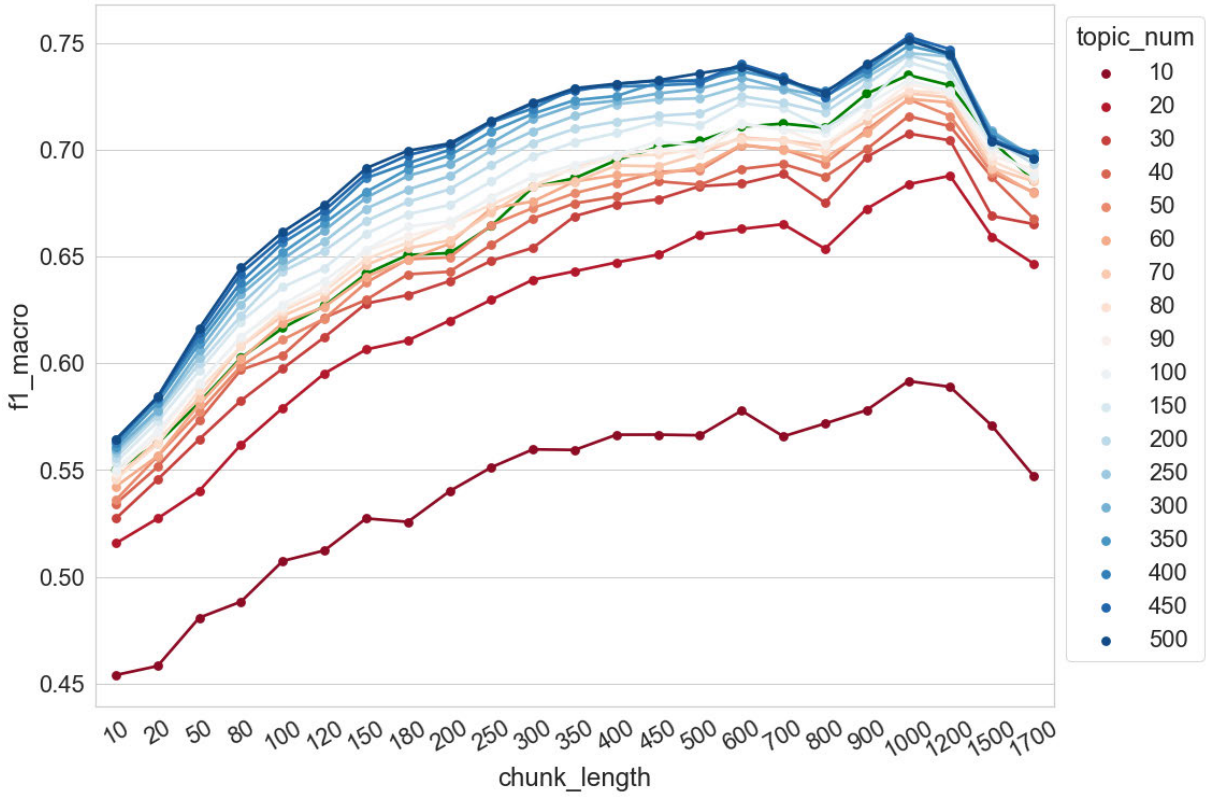

Abbildung 6.37 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Chunk-Length ***C*** und Anzahl der Topics

6.6.1.2 Topic-Kohärenz

Laut Chen et al. (2019) wird LDA nur dann als geeignet für das Topic Modeling angesehen, wenn das Korpus statistisch ausreichend ist. Ist dies nicht der Fall, enthält das Korpus häufig lediglich Kurztexte, was typischerweise einen Mangel an Informationen über die Kookkurrenz von Wörtern mit sich bringt. Eine solche Situation ist für statistische Topic-Modelle wie LDA nicht vorteilhaft. Der Grund besteht darin, dass die in der LDA angewandte Kernmethode des Gibbs-Samplings zu großen Abweichungen bei der Inferenz der Topics führen würde, insbesondere für kurze Texte. Im Ergebnis werden beim Einsatz der LDA für das Topic Modeling auf Kurztexten zahlreiche nicht-kohärente Topics produziert. Zusätzlich zu Chen et al. (2019) liefert das Experiment in diesem Unterkapitel weitere Informationen. Die Änderung der NPMI-Verteilungen in Abbildung 6.38 verdeutlicht dabei zwei Trends. Zunächst sinken die

Mediane und das untere Quartil der Verteilungen mit der Erhöhung von ***C*** ab, während das obere Quartil der Verteilungen nahezu unverändert bleibt. Zweitens werden die Spannweiten und der Quartilsabstand mit der Erhöhung von ***C*** breiter. Das Ergebnis zeigt: Im Vergleich zu auf sehr kurzen Chunks (in dieser Untersuchung z. B. ***C*** = 10 oder 20) trainierten Topic-Modellen können die kohärentesten Topics eine noch höhere Kohärenz haben, wenn die Modelle auf längeren Chunks (in dieser Untersuchung z. B. ***C*** = 700 oder 800) trainiert werden. Gleichzeitig zeigt die Kohärenz der am wenigsten kohärenten Topics eine weiter absinkende Qualität. Gleichzeitig verdeutlicht das Ergebnis, dass die Spannweite der NPMI-Werte-Verteilungen sich nicht unendlich vergrößern lässt. Ab ***C*** = 450 ist keine klare systematische Veränderung mehr erkennbar.

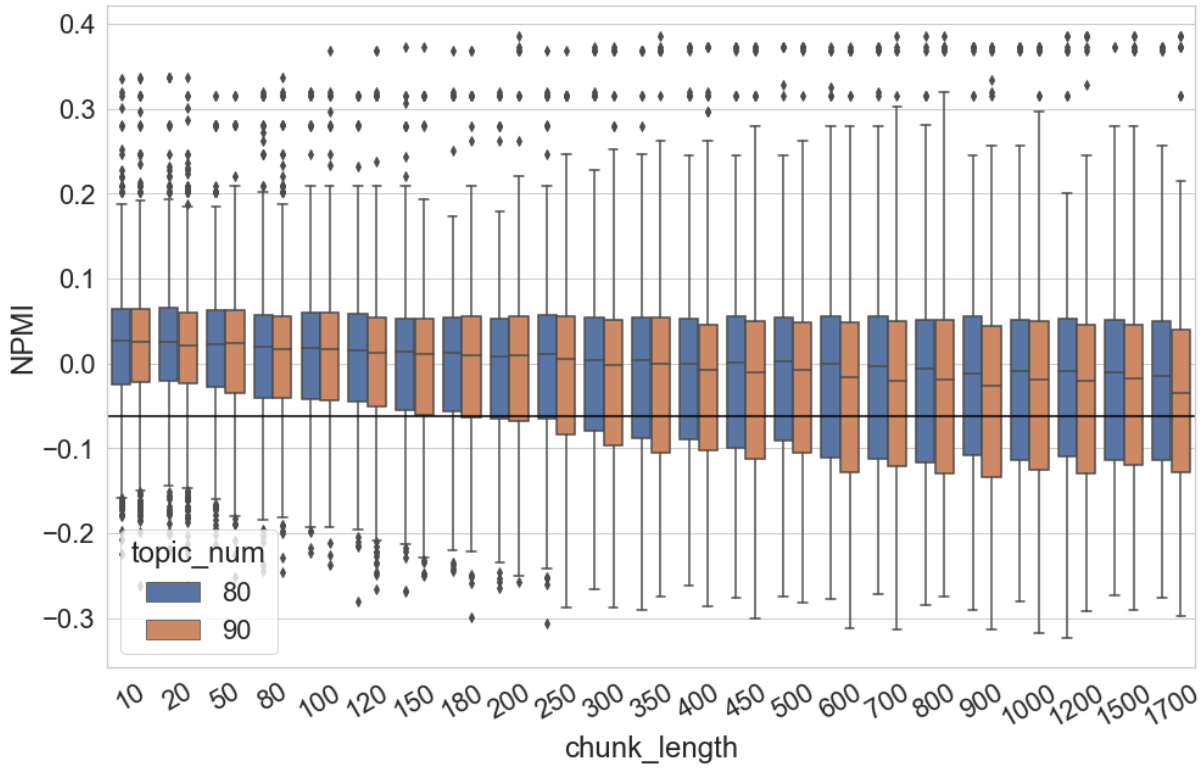

Abbildung 6.38 NPMI-Werte-Verteilung der Topics im Verhältnis zur Chunk-Length ***C***

Wenn die Anzahl der Topics stärker variiert wird, ist in der Abbildung 6.39 zunächst zu erkennen, dass die Spannweite der Verteilungen breiter ist, wenn sich ***C*** von 10 auf 300 erhöht. Die NPMI-Werte liegen zwischen ca. 0,25 und -0,35. Bei weiteren Erhöhungen bis auf ***C*** = 800 liegen die NPMI-Werte zwischen 0,3 und -0,3. Danach wird der Wertbereich der Verteilungen wieder auf 0,25 und -0,3 eingegrenzt. Insbesondere die Modelle mit einer hohen Anzahl von

Topics zeigen eine besonders ausgeprägte Verengung und Absenkung des Wertbereichs ihrer NPMI-Werte-Verteilung. Bei ***C*** = 1700 sind die NPMI-Werte von mehr als 75 % der Topics niedriger als der NPMI-Kontrollwert, wenn die Anzahl der Topics auf 500 eingestellt wird. Darüber hinaus ist zu beobachten, dass die NPMI-Werte-Verteilung weniger durch die Veränderung von ***C*** beeinflusst wird, wenn die Topic-Modelle eine geringe Anzahl von Topics (z. B. bei ***T*** = 10 oder 20) aufweisen. Die höchsten NPMI-Werte werden erreicht, wenn die Anzahl der Topics weder zu groß noch zu klein ist (in dieser Untersuchung zwischen ***T*** = 60 und ***T*** = 100) und die Chunk-Length ***C*** einen bestimmten Wert nicht unterschreitet (in dieser Untersuchung ***C*** größer als 450).

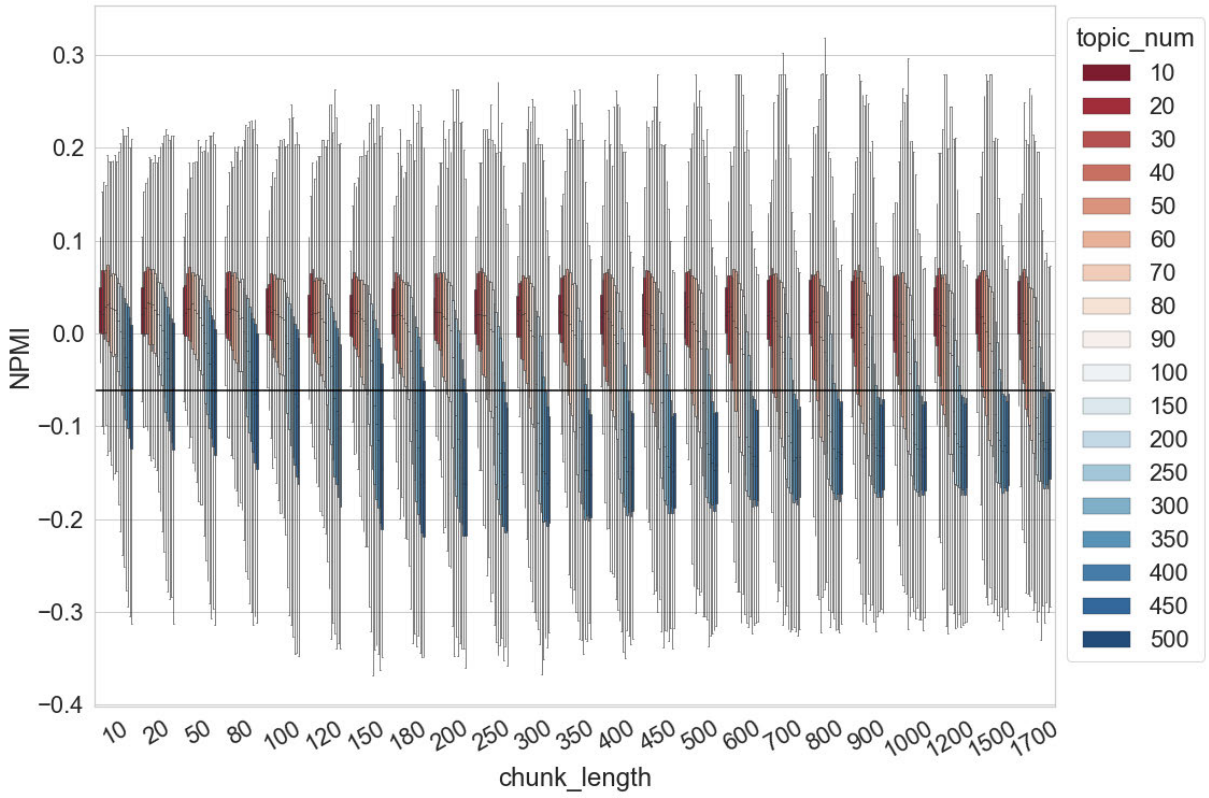

Abbildung 6.39 NPMI-Werte-Verteilung der Topics im Verhältnis zu Chunk-Length ***C*** und Anzahl der Topics

6.6.2 N-Token-Chunking

Neben der Zerlegungsstrategie in Algee-Hewitt et al. (2015) gibt es noch weitere Möglichkeiten, Texte zu zerlegen und Dokumente für das Training des Topic-Modells vorzubereiten: Statt Texte in Paragraphen wurden diese in Jockers (2013) in 1000-Token-

Chunks zerlegt. Um den Einfluss dieser Zerlegungsstrategie auf die Qualität des Topic-Modells zu untersuchen, wird in diesem Unterkapitel eine Längeneinheit des Dokuments ***C_n*** eingeführt. Jeder Zeitungsartikel im Untersuchungskorpus wird in Chunks zerlegt, von denen wiederum jedes genau ***C_n*** Wörter enthält.

Das Setting der Längeneinheit ist ***C_n*** ∈ {10, 20, 50, 80, 100, 120, 150, 180, 200, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1700}. Je größer ***C_n*** ist, desto eher ist zu erwarten, dass die gesamte Anzahl der Dokumente geringer wird. In Abbildung 6.40 wird die Veränderung der Anzahl der Dokumente und der durchschnittlichen Dokumentlänge mit der Erhöhung von ***C_n*** dargestellt. Die Anzahl der Dokumente (die grüne Linie) reduziert sich von ca. 350.000 auf weniger als 50.000, während die durchschnittliche Dokumentlänge (die blaue Linie) von 10 Tokens auf mehr als 1000 Tokens ansteigt. Wie im Unterkapitel 5.6.1 wird ***C_n*** maximal auf 1700 eingestellt.

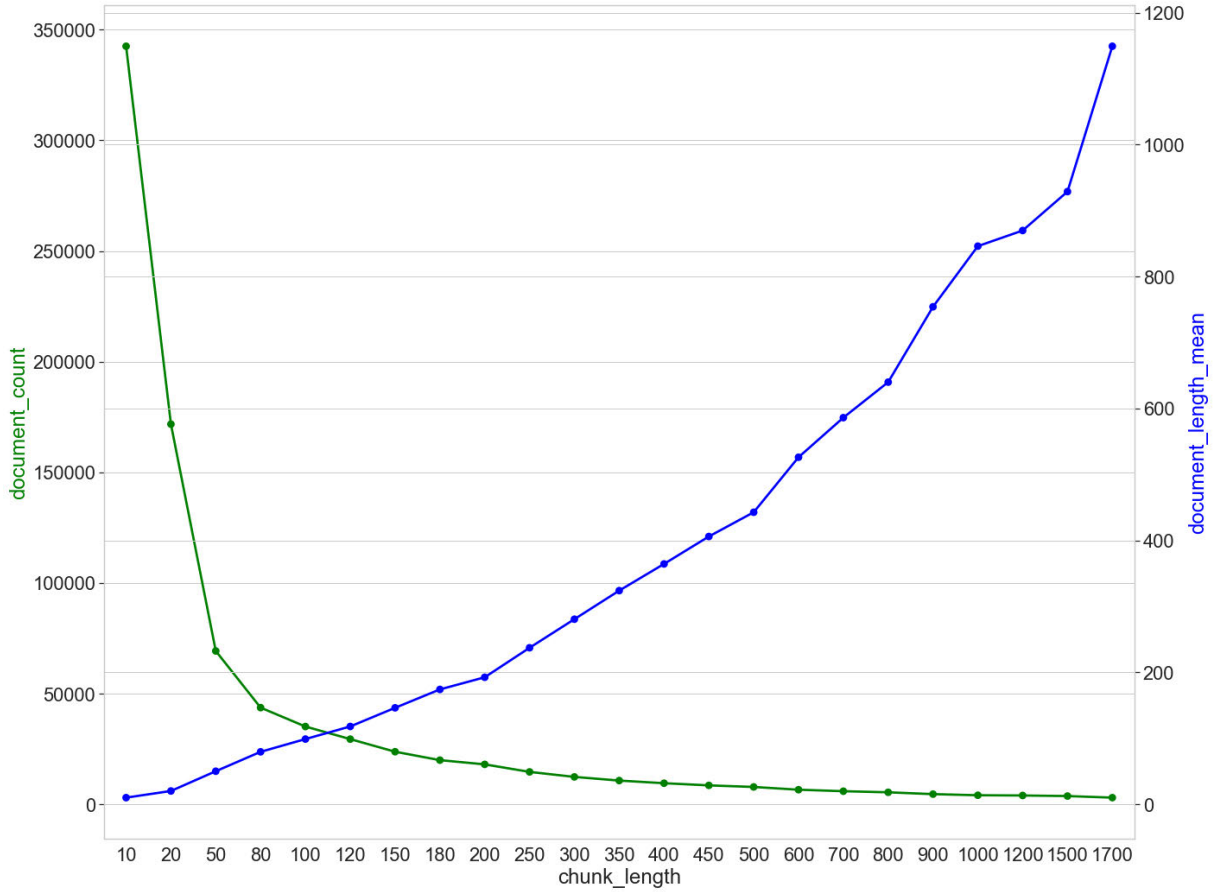

Abbildung 6.40 Anzahl der Dokumente (document_count) und die durchschnittliche Dokumentlänge (document_length_mean) im Verhältnis zur Chunk-Length ***C_n***

6.6.2.1 Dokumentklassifikation

Es sei hier nochmals betont werden, dass bei unterschiedlichen Settings von ***C_n*** der Unterschied in den Klassifikationsergebnissen den Unterschied in der Modellqualität nicht wirklich repräsentieren kann, weil die Klassifikationen auf unterschiedlichen Korpora (in Bezug auf die Anzahl der Dokumente) durchgeführt werden. Die in der Abbildung dargestellten Ergebnisse beschreiben also lediglich, wie sich die Klassifikationsergebnisse verändern, wenn der gleiche Korpus unterschiedlich zerlegt wird. Für diese Untersuchung wird die BoW-basierte Klassifikation auf die 23 segmentierten Korpora durchgeführt, um die Baseline der Klassifikation neu zu erstellen. Die Klassifikationen erfolgten wie zuvor als 10-fache Kreuzvalidierung mit linearer SVM. Die 23 F1 (Makro)-Werte werden in Abbildung 6.41 und Abbildung 6.42 durch eine grüne Linie dargestellt.

In Abbildung 6.41 ist das Klassifikationsergebnis dargestellt. Die Anzahl der Topics wird zuerst auf ***T*** = 80 und 90 eingestellt. Ähnlich wie bei den Testergebnissen im letzten Unterkapitel wird auch hier eine aufsteigende Tendenz der F1-Werte beobachtet, wenn ***C_n*** von 10 auf 1200 erhöht wird: Je größer die Segmentlänge ***C_n*** ist, desto erfolgreicher kann die Klassifikation funktionieren. Ab ***C_n*** = 1200 kann eine Verbesserung nicht mehr beobachtet werden, die Kurve senkt sich mit weiterer Erhöhung von ***C_n*** ab. Das Ergebnis ist dem der vorherigen Untersuchung ähnlich und die Verschlechterung der Klassifikation ab ***C_n*** = 1200 beruht sehr wahrscheinlich nicht auf der Erhöhung von ***C_n***. Wenn ***C_n*** auf einen hohen Wert eingestellt wird, werden zahlreiche kurze Dokumente durch das Chunking erzeugt. Die Klassifikation verschlechtert sich, weil die Anzahl der kurzen Dokumente gestiegen ist. Diese Untersuchung zeigt: Je länger die Chunks sind, desto erfolgreicher ist die Topic-Modeling-basierte Klassifikation, wenn die Dokumentsammlung nicht zu viel kurze Dokumente enthält. Eine weitere Beobachtung, die den in der vorherigen Untersuchung gemachten Erfahrungen ähnelt ist, dass die Topic-Modeling-basierte Klassifikation besser als die BoW-basierte funktionieren kann. Zwischen ***C_n*** = 120 und ***C_n*** = 350 liegt die grüne Linie in der Abbildung 6.41 unterhalb der Boxplots. Wenn im Korpus ausschließlich kurze Dokumente enthalten sind (z. B. ***C_n*** = 10 oder 20), kann das BoW-basierte Verfahren die Dokumente deutlich besser klassifizieren. Das Ergebnis zeigt, dass im Vergleich zur Topic-Modeling-basierten Methode das BoW-Verfahren besser für die Klassifikation der Kurztexte geeignet ist.

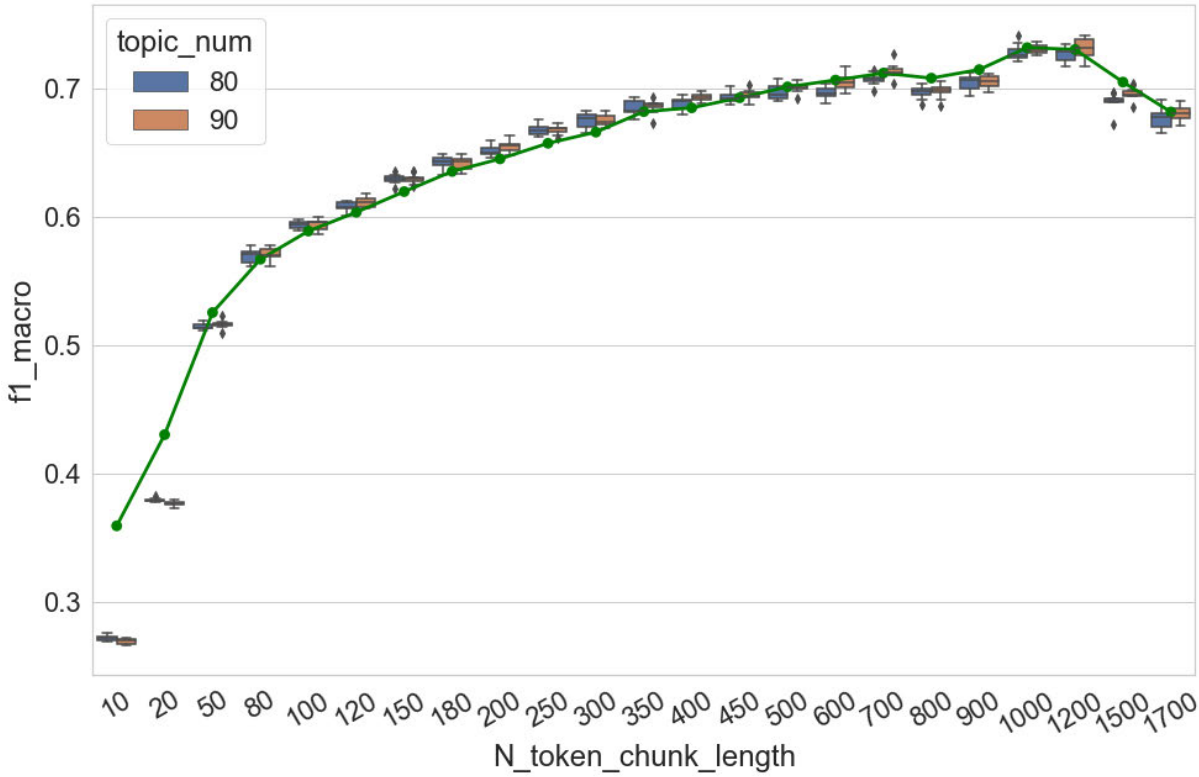

Abbildung 6.41 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation im Verhältnis zur Chunk-Length ***C_n***

Der aufsteigende Trend der F1-Werte ist auch sichtbar, wenn die Anzahl der Topics stärker variiert wird (Abbildung 6.42). Die meisten F1-Werte liegen zwischen 0,25 und 0,75. Wenn der Wert von ***C_n*** kleiner als 50 ist, sind die Ergebnisse der Topic-Modeling-basierten Klassifikation bei jedem Setting der Anzahl der Topics schlechter als die der BoW-basierten. Insbesondere ist bei ***C_n*** kleiner als 50 zu beachten, dass die Topic-Modelle, die eine höhere Anzahl von Topics enthalten, nicht die beste Klassifikation erreichen. Hier wird nochmal bestätigt, dass das Topic-Modeling-basierte Verfahren für die Klassifikation der Kurztexte nicht geeignet ist. Beginnend mit einem ***C_n*** größer als 50 lässt sich erkennen, dass die Klassifikation bessere Ergebnisse erzielt, wenn die Anzahl der Topics größer ist. Die Topic-Modeling-basierte Klassifikation kann besser als die BoW-basierte Baseline funktionieren, wenn Topic-Modelle mit mehr als 100 Topics trainiert werden. In diesem Test lagen die höchste Accuracy und der höchste F1-Wert bei 0,762 bzw. 0,758.

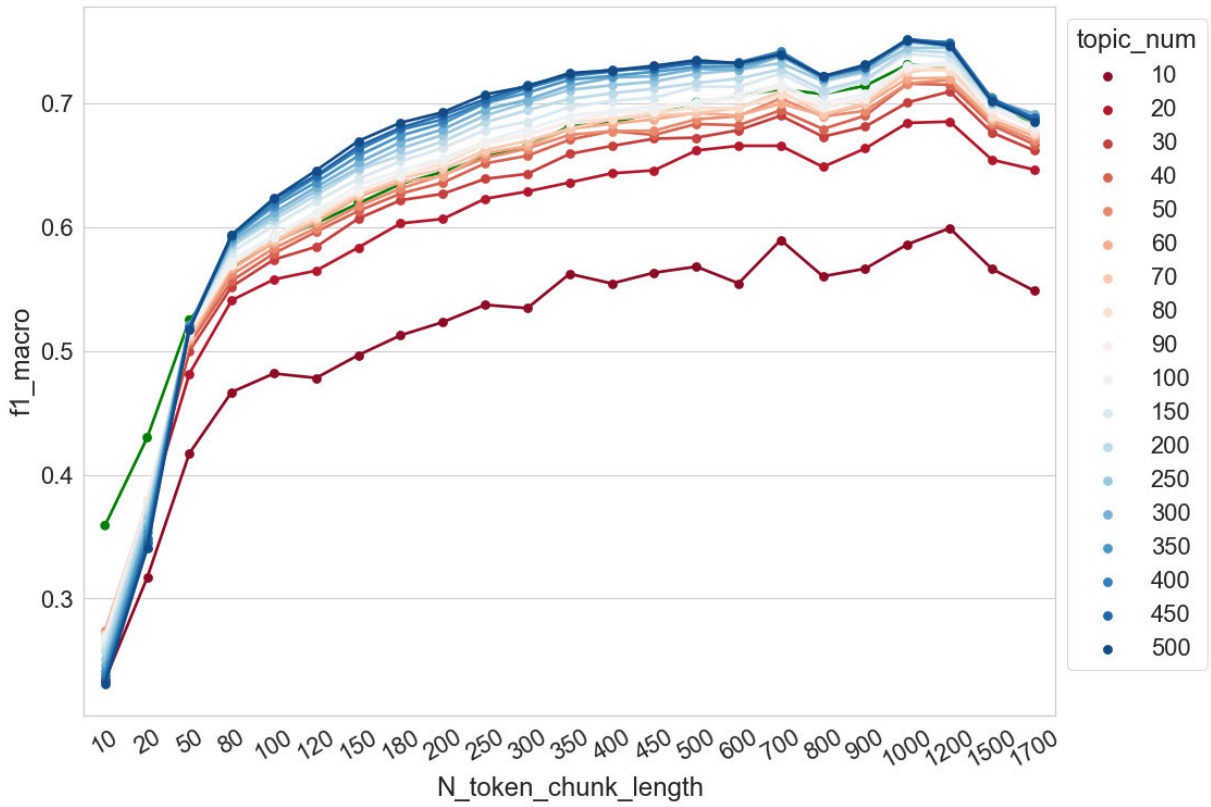

Abbildung 6.42 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zur Chunk-Length ***C_n*** und Anzahl der Topics

6.6.2.2 Topic-Kohärenz

In Abbildung 6.43 werden die NPMI-Verteilungen visualisiert. Die Anzahl der Topics wird zuerst auf ***T*** = 80 und 90 eingestellt. Die Veränderung der NPMI-Werte-Verteilungen folgt einen ähnlichen Trend wie bei der Untersuchung in Bezug auf die Paragraph-Chunks. Unabhängig von der Anzahl der Topics wird die Spannweite der Verteilungen breiter, wenn ***C_n*** von 10 auf 700 erhöht wird. Der Wertbereich erweitert sich zwischen -0,2 und 0,2 auf zwischen -0,3 und 0,3. Mit einer weiteren Erhöhung von ***C_n*** kann keine systematische Veränderung der Spannweite (trotz vorhandener Schwankungen) beobachtet werden. Außerdem sinken die Mediane der Verteilungen mit der Erhöhung von ***C_n*** allmählich unter 0 ab. Deshalb ist in der Abbildung auch zu beobachten, dass es mit der Erhöhung von ***C_n*** mehr Topics gibt, deren NPMI-Werte niedriger als der NPMI-Kontrollwert sind.

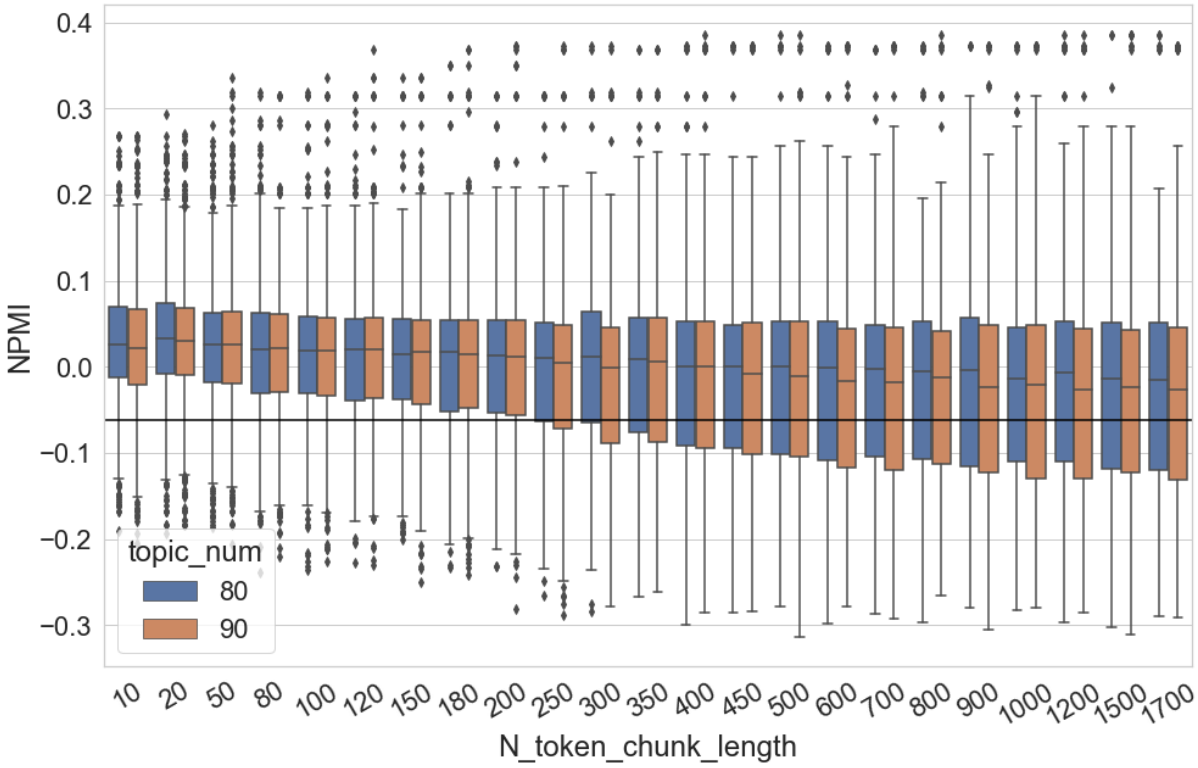

Abbildung 6.43 NPMI-Werte-Verteilung der Topics im Verhältnis zur Chunk-Length ***C_n***

Wenn die Anzahl der Topics in höherem Maße variiert wird, ist in Abbildung 6.44 zu sehen, dass die NPMI-Werte-Verteilungen durch die Veränderung von ***C_n*** weniger beeinflusst werden, wenn die Anzahl der Topics geringer ist. Die NPMI-Werte liegen fast immer zwischen 0,1 und -0,1 bei jedem Setting von ***C_n***, wenn die Anzahl der Topics z. B. auf zehn eingestellt wird. Im Gegensatz dazu gilt: Je größer die Anzahl der Topics eingestellt wird, desto ausgeprägter ist die Veränderung der Verteilungen. Wenn die Anzahl der Topics z. B. auf 500 eingestellt wird, liegt die Verteilung der NPMI-Werte bei ***C_n*** = 10 zwischen ca. 0,15 und -0,32. Mit der Erhöhung von ***C_n*** erweitert sich der Wertbereich der Verteilung bei ***C_n*** = 200 auf zwischen ca. 0,23 und -0,35. Ab ***C_n*** = 450 fällt die Verteilung der NPMI-Werte auf einen Bereich zwischen ca. 0,1 und -0,3 und mehr als 75 % der NPMI-Werte sind niedriger als der entsprechende Kontrollwert. Ähnlich wie bei der vorherigen Untersuchung werden die höchsten NPMI-Werte erreicht, wenn die Anzahl der Topics weder zu groß noch zu klein ist (in dieser Untersuchung zwischen ***T*** = 60 und ***T*** = 120) und die Chunk-Length ***C_n*** ein bestimmtes Maß nicht unterschreitet (in dieser Untersuchung größer als 450).

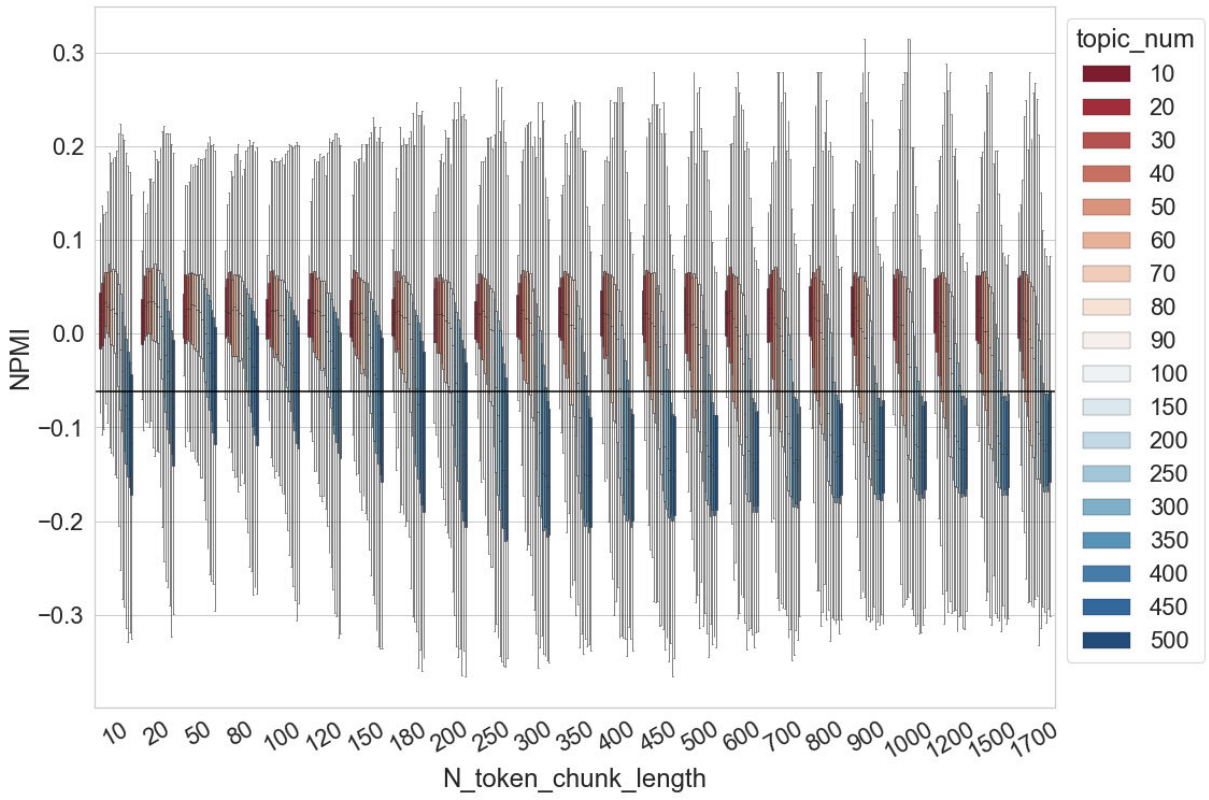

Abbildung 6.44 NPMI-Werte-Verteilung der Topics im Verhältnis zu Chunk-Length ***C_n*** und Anzahl der Topics

6.6.3 Chunking auf Satzebene

Neben den beiden oben genannten Strategien kann auch der Satz als eine thematische Einheit betrachtet werden und das Chunking auf Satzebene durchzuführt werden. Deshalb wird ein dritter Test durchgeführt. Bei diesem Experiment wird eine kleinste Längeneinheit des Dokuments ***C_s*** eingeführt. Ein Text wird zunächst in Sätze zerlegt. Wenn ein Satz S_n genau ***C_s*** Tokens oder mehr als ***C_s*** Tokens enthält, wird der Satz S_n als ein Chunk für das Topic Modeling gespeichert. Wenn der Satz S_n weniger als ***C_s*** Tokens enthält, wird der nächste Satz S_{n+1} zu S_n hinzugefügt, um einen Chunk aufzubauen. Wenn die beiden Sätze insgesamt immer noch weniger als ***C_s*** Tokens enthalten, wird der nächste Satz S_{n+2} für den Aufbau eines Chunks hinzugefügt und so weiter. Nach der Segmentierung des Textes können Topic-Modelle auf den zerlegten Korpora trainiert werden. Das Setting der kleinsten Längeneinheit war ***C_s*** ∈ {10, 20, 50, 80, 100, 120, 150, 180, 200, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1700}. Je größer die kleinste Segmentlänge ***C_s*** ist, desto geringer ist die Anzahl der Dokumente. In Abbildung 6.45 wird die Veränderung der Anzahl der Dokumente und der

durchschnittlichen Dokumentlänge mit der Erhöhung von ***C_s*** dargestellt. Die Anzahl der Dokumente (die grüne Linie) reduziert sich von ca. 120.000 auf weniger als 20.000, während die durchschnittliche Dokumentlänge (die blaue Linie) von knapp 30 Tokens auf etwa 1700 zunimmt. Wie in der vorherigen zwei Untersuchungen wird in diesem Experiment ***C_s*** maximal auf 1700 eingestellt.

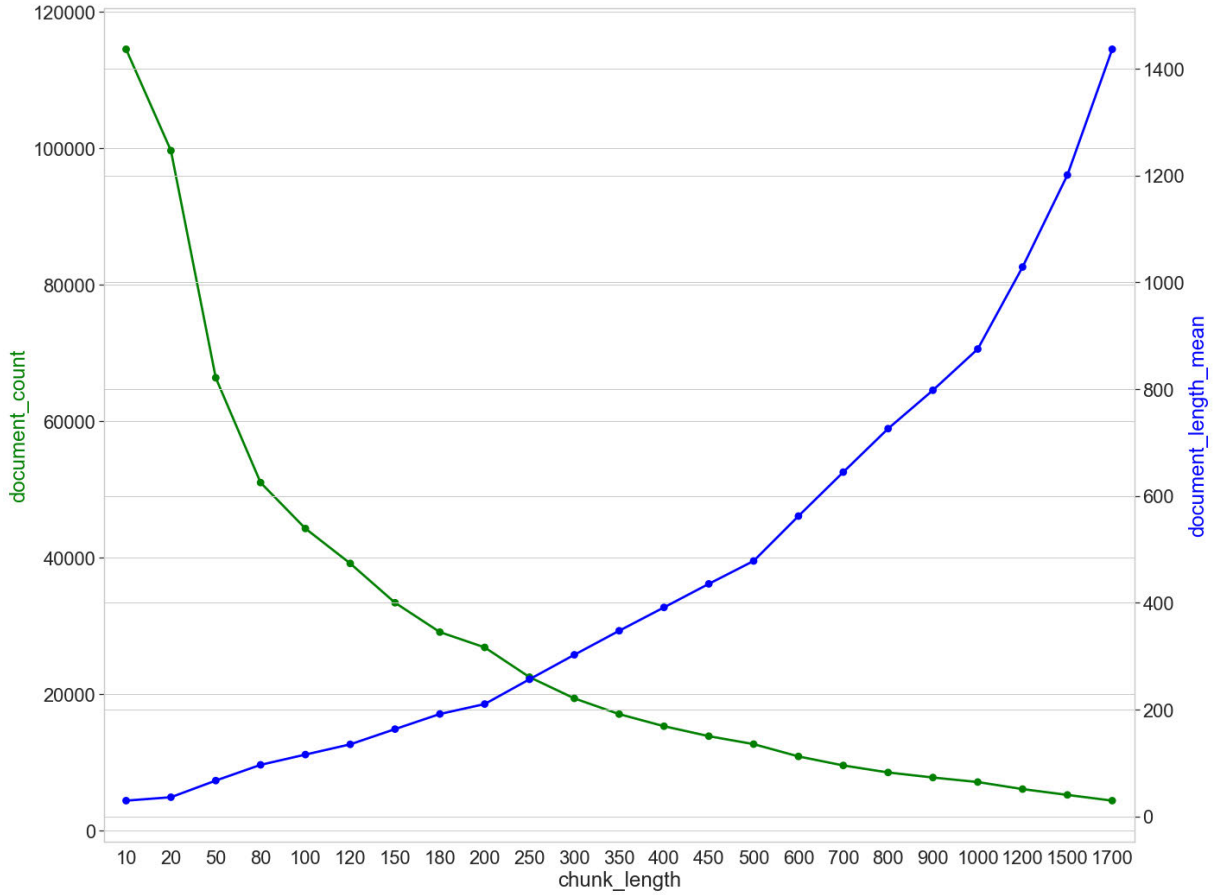

Abbildung 6.45 Anzahl der Dokumente (document_count) und die durchschnittliche Dokumentlänge (document_length_mean) im Verhältnis zur Chunk-Length ***C_s***

6.6.3.1 Dokumentklassifikation

Wie in den vorherigen beiden Untersuchungen kann der Unterschied in den Klassifikationsergebnissen den Unterschied in der Modellqualität nicht widerspiegeln, wenn das Setting von ***C_s*** nicht identisch ist. Die in der Abbildung dargestellten Ergebnisse beschreiben also lediglich, wie sich die Klassifikationsergebnisse ändern, wenn der gleiche Korpus unterschiedlich zerlegt wird. Für diese Untersuchung wird zudem eine BoW-basierte Klassifikation auf den 23 segmentierten Korpora durchgeführt, um die Baseline der

Klassifikation neu zu erstellen. Die Klassifikationen erfolgten wie vorher als 10-fache Kreuzvalidierung mit linearer SVM. Die 23 F1 (Makro)-Werte werden in Abbildung 6.46 und Abbildung 6.47 durch eine grüne Linie dargestellt.

In Abbildung 6.46 ist die F1-Verteilung der zehn Einzeldurchläufe des jeweiligen Settings von ***C_s*** durch Boxplots dargestellt. Die Anzahl der Topics wird zuerst auf 80 und 90 festgelegt. Ähnlich wie bei den Testergebnissen der beiden vorherigen Untersuchungen ist mit der Erhöhung der Chunk-Length ***C_s*** ein aufsteigender Trend der F1-Werte zu beobachten. Diese steigen von ca. 0,42 auf 0,70. Die Topic-Modeling-basierte Klassifikation funktioniert ungefähr gleich gut wie die BoW-basierte Klassifikation, wenn ***C_s*** zwischen 50 und 700 eingestellt wird. In den vorherigen beiden Untersuchungen wurde beobachtet, dass die Klassifikationsergebnisse ab einer Chunk-Length von 1200 stetig schlechter werden. Dieses Phänomen lässt sich hier allerdings nicht deutlich erkennen.

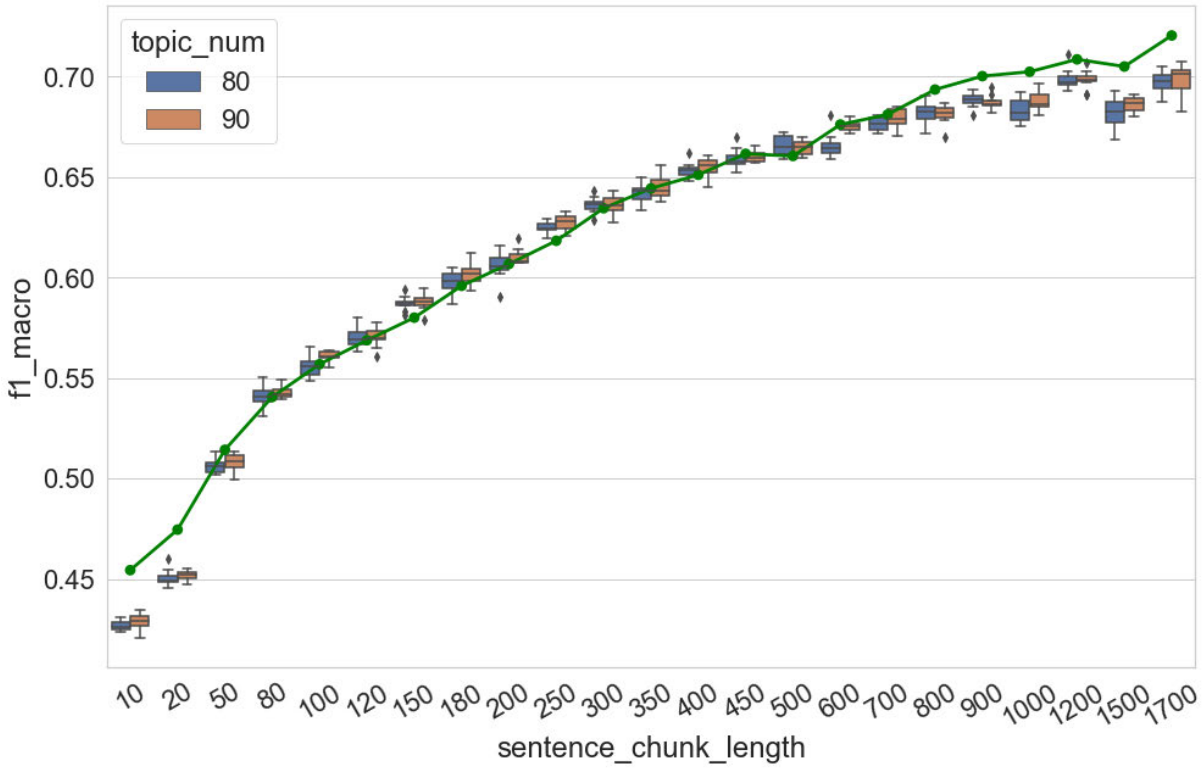

Abbildung 6.46 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation im Verhältnis zur Chunk-Length ***C_s***

Wenn die Anzahl der Topics stärker variiert wird, ist in Abbildung 6.47 zu beobachten, dass sich die Klassifikation mit der Erhöhung von ***C_s*** verbessert. In diesem Test liegen die meisten

F1-Werte zwischen 0,35 und 0,75; die höchste Accuracy und der höchste F1-Wert liegen bei 0,741 bzw. 0,739. Ab ***T*** = 80 sind die Ergebnisse der Topic-Modeling-basierten Klassifikation besser als die der BoW-basierten Klassifikation, wenn ***C_s*** größer als 50 ist. Je größer die Anzahl der Topics ist, desto erfolgreicher ist die Klassifikation. In dieser Untersuchung wird zudem erneut bestätigt, dass das Topic-Modeling-basierte Verfahren für die Klassifikation von Kurztexten nicht geeignet ist.

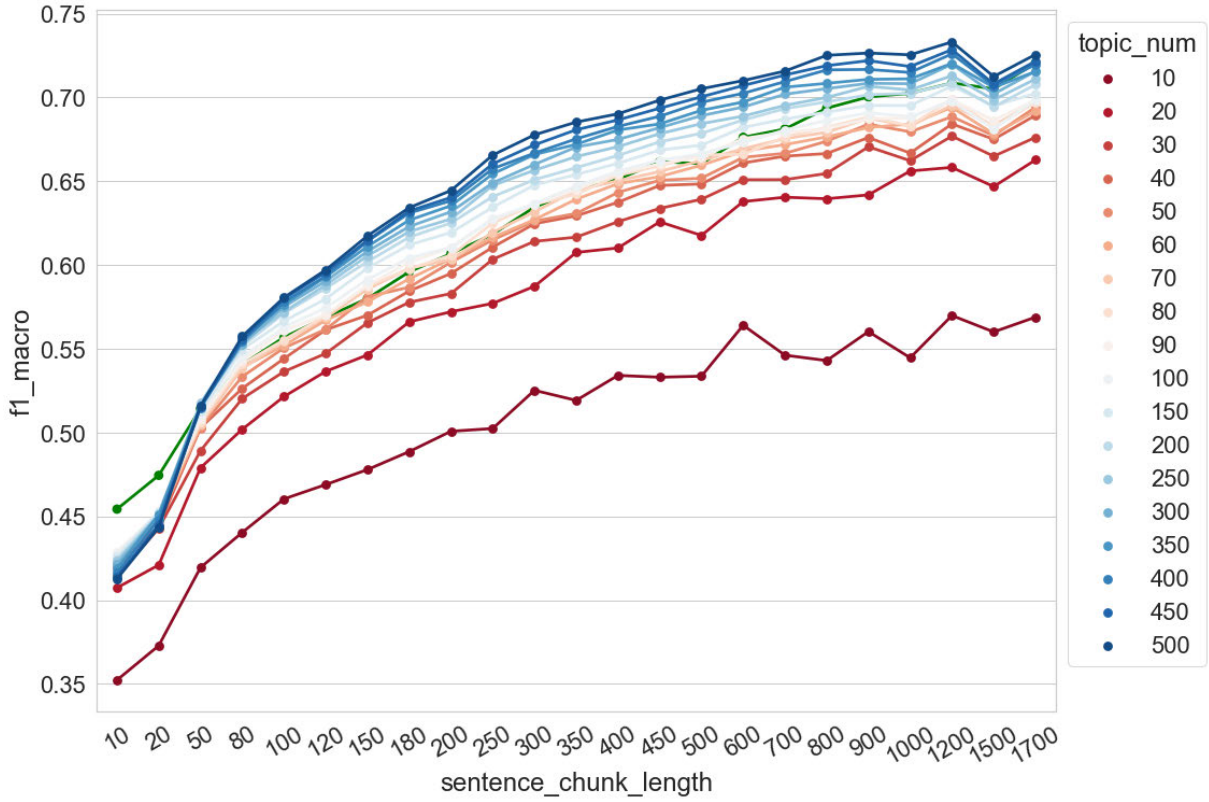

Abbildung 6.47 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Chunk-Length ***C_s*** und Anzahl der Topics

Zusammenfassend betrachtet lässt sich feststellen, dass die Ergebnisse der drei oben vorgestellten Untersuchungen in Bezug auf die Chunk-Length einander ähneln. Werden die Resultate verglichen, so ist in Abbildung 6.48 zu beobachten, dass das Chunking auf Satzebene im Vergleich zu den anderen beiden Zerlegungsstrategien zu einer etwas schlechteren Klassifikation führt. Hier wird die Anzahl der Topics auf 90 eingestellt. Die Ergebnisse bei den anderen Settings von Anzahl der Topics sind ähnlich und werden deshalb nicht visualisiert. Wenn die Chunk-Length größer als 500 ist, funktioniert das Chunking auf Paragraph-Ebene und das Chunking auf N-Token-Ebene fast gleich gut.

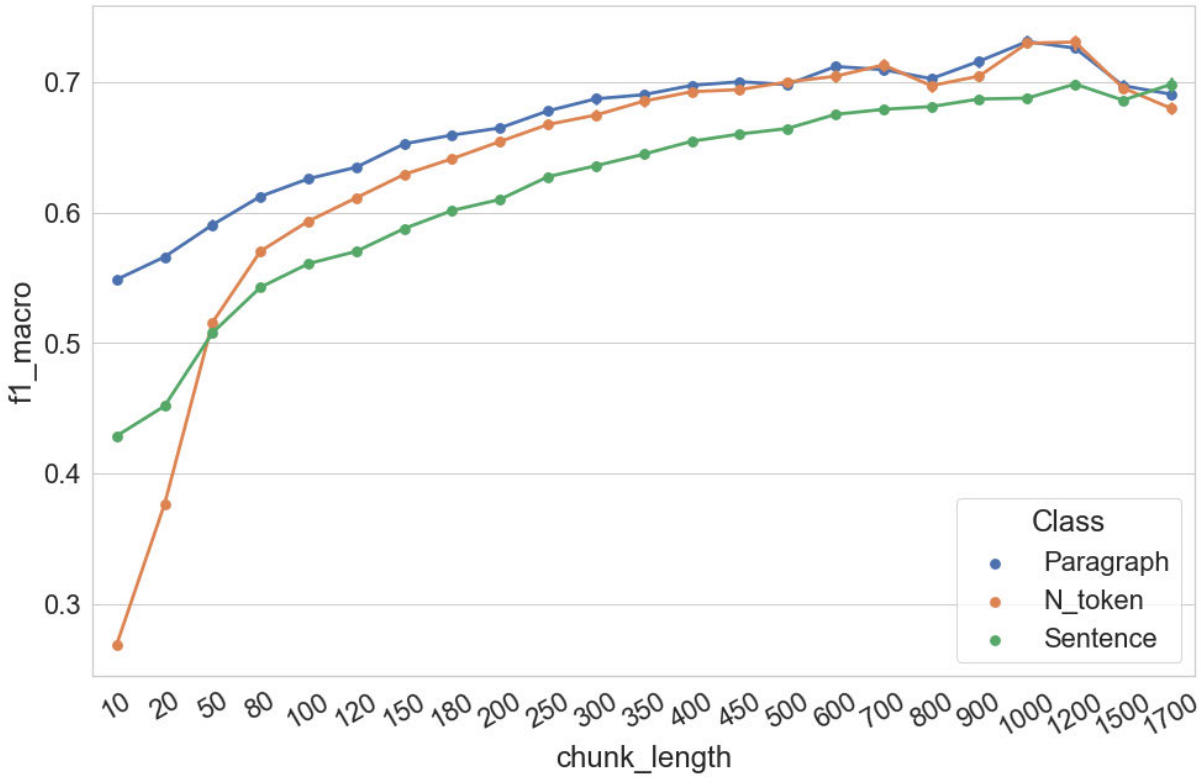

Abbildung 6.48 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation im Verhältnis zu den drei Chunking- Strategien

6.6.3.2 Topic-Kohärenz

In Abbildung 6.49 wird die Änderung der NPMI-Verteilung bei einer Erhöhung von ***C_s*** visualisiert. Die Anzahl der Topics wird zuerst auf 80 und 90 eingestellt. Der Median der Verteilungen ist fast bei jedem Setting von ***C_s*** höher, wenn die Anzahl der Topics auf 80 eingestellt wird. Ähnlich wie in den vorherigen beiden Untersuchungen ist hier zu beobachten, dass die Spannweite der NPMI-Werte-Verteilungen breiter wird, wenn ***C_s*** von 10 auf 800 erhöht wird. Der Wertbereich erweitert sich von zwischen ca. -0,12 und knapp 0,2 auf zwischen ca. -0,3 und knapp 0,3. Mit einer weiteren Erhöhung von ***C_s*** kann keine systematische Veränderung der Spannweite trotz der Schwankung bei ***C_s*** = 1000 beobachtet werden. Außerdem sinken die Mediane der Verteilungen mit der Erhöhung von ***C_s*** allmählich auf ca. -0,05 ab. Deshalb existieren mit der Erhöhung von ***C_s*** mehr Topics, deren NPMI-Werte niedriger als der NPMI-Kontrollwert sind. Bei ***C_s*** = 1700 z. B. liegen fast 50 % der NPMI-Werte unter dem Kontrollwert.

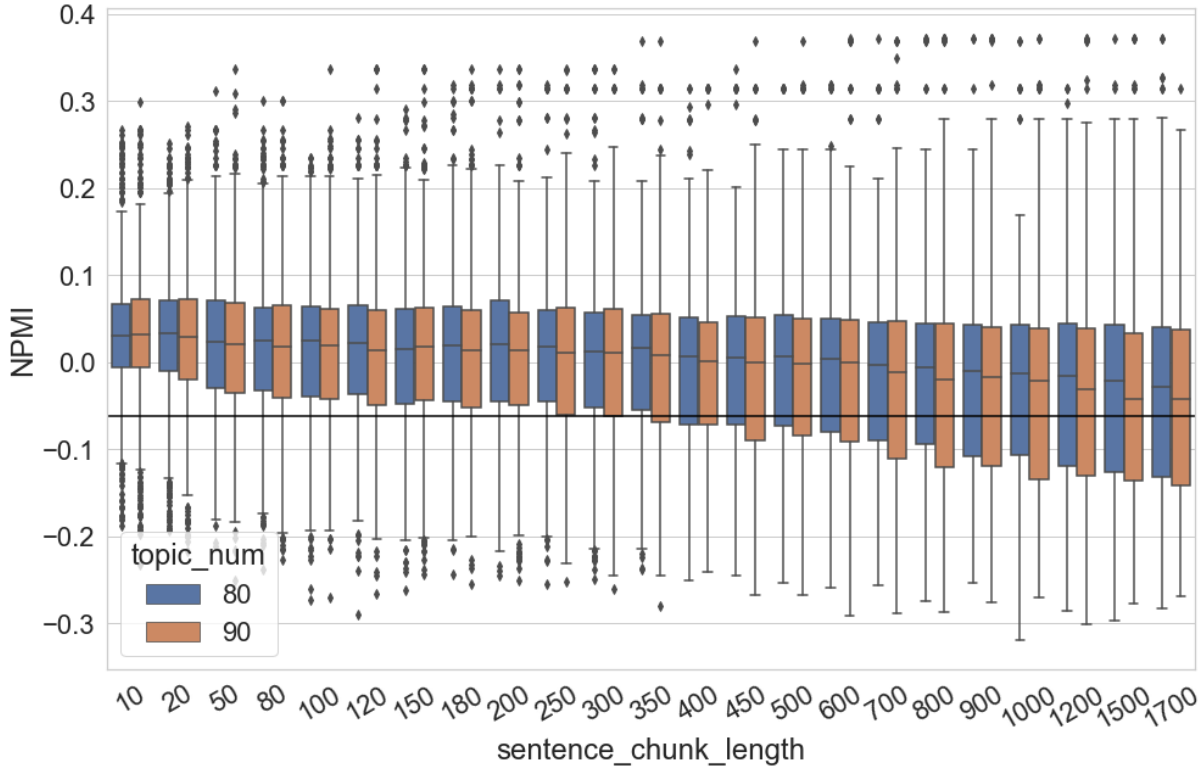

Abbildung 6.49 NPMI-Werte-Verteilung der Topics im Verhältnis zur Chunk-Length ***C***

Wenn die Anzahl der Topics in größerem Umfang variiert wird, ist in Abbildung 6.50 zunächst zu erkennen, dass die Spannweite der Verteilungen breiter ist, wenn sich ***C_s*** von 10 auf 50 erhöht. Die NPMI-Werte liegen zwischen ca. 0,25 und -0,35. Mit weiteren Steigerungen bis zu einem Wert von ***C_s*** = 1700 liegen die NPMI-Werte zwischen ca. 0,28 und -0,31. Insbesondere die Modelle mit einer hohen Anzahl von Topics weisen die Tendenz auf, den Wertbereich ihrer NPMI-Werte-Verteilung mit der Erhöhung von ***C_s*** zu verengen und abzusenken. Ab ***C_s*** = 700 sind z. B. mehr als 75 % der NPMI-Werte niedriger als der Kontrollwert, wenn die Anzahl der Topics auf 500 eingestellt wird. Ähnlich wie in den vorherigen Untersuchungen werden die NPMI-Werte-Verteilungen weniger durch die Veränderung von ***C_s*** beeinflusst, wenn die Topic-Modelle über eine geringe Anzahl von Topics (z. B. bei ***T*** = 10 oder 20) verfügen. Die höchsten NPMI-Werte werden erreicht, wenn die Anzahl der Topics weder zu groß noch zu klein eingestellt wird (in dieser Untersuchung zwischen ***T*** = 60 und ***T*** = 100) und die Chunk-Length ***C_s*** einen nicht zu geringen Wert aufweist (in dieser Untersuchung ***C_s*** größer als 800).

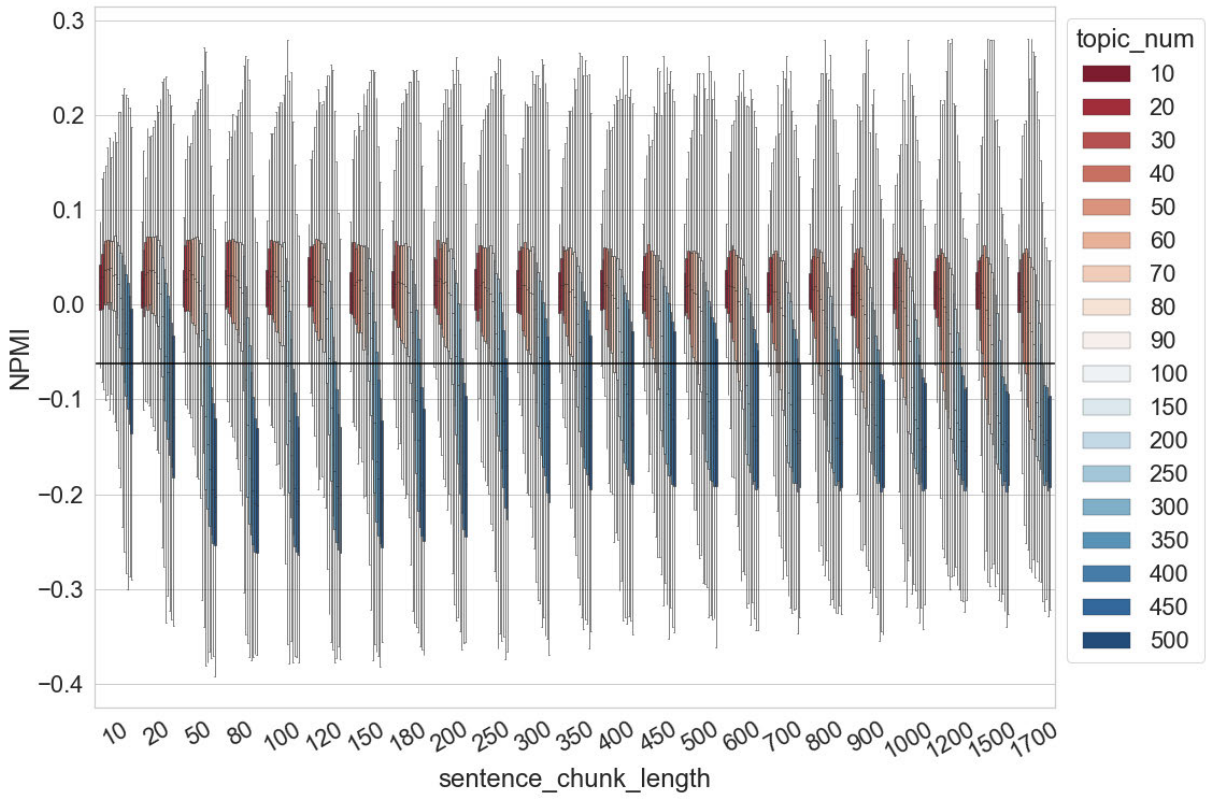

Abbildung 6.50 NPMI-Werte-Verteilung der Topics im Verhältnis zu Chunk-Length ***C_s*** und Anzahl der Topics

Zusammenfassend lässt sich festhalten, dass die Ergebnisse der drei oben vorgestellten Untersuchungen ähnlich sind und die Unterschiede zwischen den drei Zerlegungsstrategien eine lediglich geringe Auswirkung auf die Topic-Kohärenz haben. Als ein Beispiel werden in Abbildung 6.51 die NPMI-Werte-Verteilungen visualisiert, wenn die Anzahl der Topics auf 90 eingestellt wird. Es ist zu beobachten, dass die Spannweiten der NPMI-Verteilungen größer werden, wenn die Chunk-Length von 10 auf 800 erhöht wird. Der Median der NPMI-Verteilungen sinkt von ungefähr 0,05 auf knapp -0,05 ab, wenn die Chunk-Length von 10 auf 1700 erhöht wird. Die Verteilungen der drei Zerlegungsstrategien sind zwar bei jedem Setting der Chunk-Length leicht abweichend, ein systematischer Unterschied wird hier jedoch nicht beobachtet.

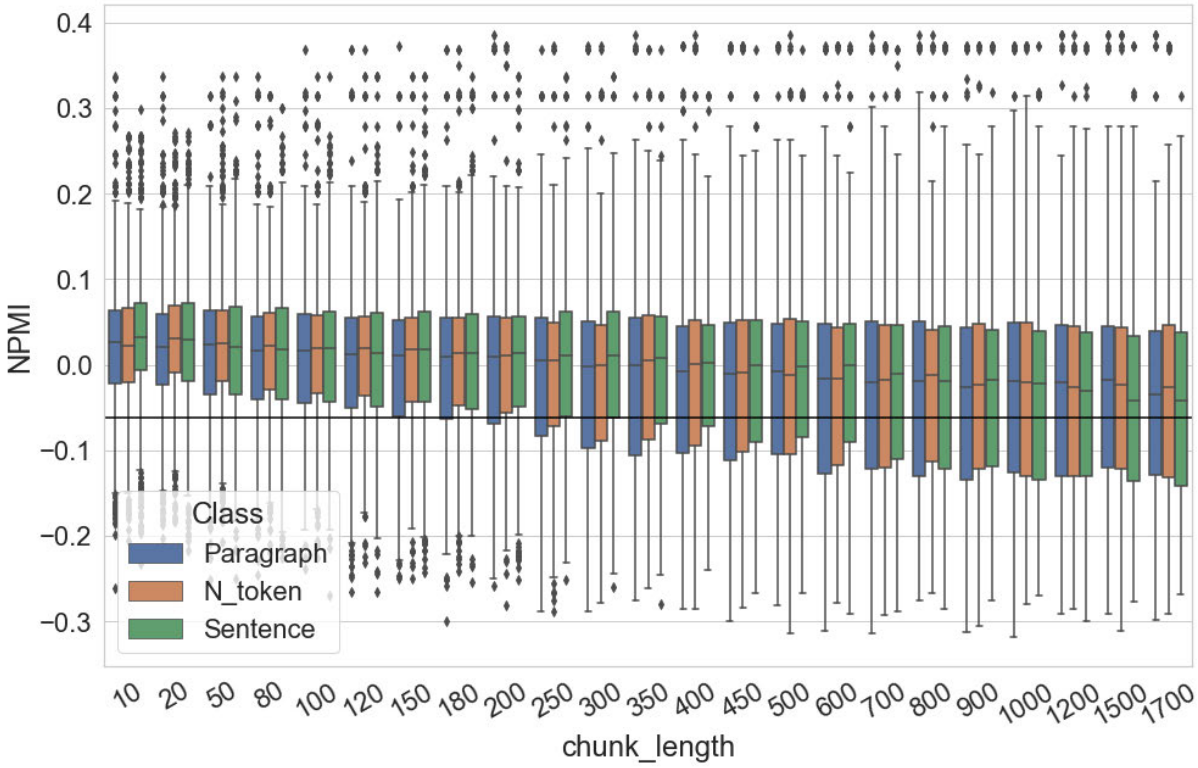

Abbildung 6.51 NPMI-Werte-Verteilung der Topics im Verhältnis zu den drei Chunking-Strategien

6.7 Zwischenfazit

In diesem Kapitel wird der Einfluss von sechs Faktoren (die Anzahl der Topics, der Hyperparameter Alpha, die Hyperparameter Optimierung, der Hyperparameter Beta, die Anzahl der Iterationen des Gibbs-Samplings und die Chunk-Length) auf Topic Modeling systematisch evaluiert. Die Ergebnisse der Untersuchungen zeigen, dass der Einfluss dieser Faktoren bestimmte Muster aufweist. So kann sich beispielsweise eine zu hohe oder zu niedrige Einstellung von dem Hyperparameter Alpha und dem Hyperparameter Beta negativ auf die Qualität des trainierten Topic-Modells auswirken. Außerdem gilt: Je mehr kohärente Topics man durch das Training des Topic-Modells erhält, desto mehr nicht-kohärente Topics erhält man gleichzeitig. Eine Frage, die weiter untersucht werden sollte, ist natürlich, wie viele der in diesem Kapitel beobachteten interessanten Phänomene mit dem untersuchenden Zeitungskorpus zusammenhängen. Würden die Ergebnisse gleich bleiben, wenn ein neues Korpus verwendet würde? Um diese Frage zu beantworten, wäre es notwendig, eine weitere Evaluation ihres Einflusses auf das Topic Modeling an einem neuen Korpus vorzunehmen.

7. Evaluation des Topic Modeling in den Digital Humanities am Beispiel eines Romankorpus

Im Folgenden werden sämtliche Untersuchungen aus dem Kapitel 6 erneut durchgeführt, beim Korpus handelt es sich hier um eine Sammlung deutscher Heftromane. Eine genauere Beschreibung des Korpus befindet sich in Kapitel 5.1.2. Das Ziel in diesem Kapitel ist es vor allem zu analysieren, ob sich die Ergebnisse der Experimente in Bezug auf den Einfluss der verschiedenen Faktoren stark verändern, wenn es sich beim Untersuchungskorpus um eine andere Textsammlung handelt, die aus literarischen Texten besteht. An dieser Stelle ist zu betonen, dass der Hauptunterschied zwischen den fünf verschiedenen Untergattungen der vorliegenden Romansammlung auf der thematischen Ebene zu finden ist. Deshalb ist Topic Modeling eine geeignete Forschungsmethode, um das Korpus zu explorieren. Dies heißt aber nicht automatisch, dass die Methode des Topic Modeling für Gattungsbestimmungen oder -vergleiche in jedem Fall geeignet ist.

In diesem Kapitel wird erarbeitet, wie sich die Ergebnisse der Topic-Modeling-basierten Dokument-Klassifikation und die Topic-Kohärenz unter dem Einfluss verschiedener Faktoren verändern. Die gleichen Untersuchungen wie im vorangegangenen Kapitel wird erneut durchgeführt. In jedem Unterkapitel wird die Untersuchung eines Faktors vorgestellt. Bei der Analyse wird der Wert des zu untersuchenden Faktors variiert, während alle anderen Faktoren Kontrollvariablen sind und unverändert gehalten werden. Wenn keine zusätzlichen Hinweise gegeben werden, sind die Kontrollvariablen folgendermaßen eingestellt: Iteration des Gibbs-Samplings $\boldsymbol{I}$ = 2000, Hyperparameter Alpha jedes Topics $\boldsymbol{\alpha}$ = 0,05, Hyperparameter Beta $\boldsymbol{\beta}$ = 0,01. Um sicherzustellen, dass die Testergebnisse in diesem Kapitel mit früheren Tests in Kapitel 6 auf dem Zeitungskorpus vergleichbar sind, werden die Romane in Chunks zerlegt und die Chunk-Length wird auf 1800 Wörter[73] eingestellt. Die Stoppwörter werden vor dem Training des Modells aus dem Korpus entfernt und die Modelle werden ohne Hyperparameter-Optimierung trainiert. Ein NPMI-Kontroll-Wert, der die Topic-Kohärenz der „Topics ohne Topic Modeling“ repräsentiert, wird in diesem Kapitel ebenfalls festgelegt. 18 Topic-Modelle werden zunächst mit nur einer Iteration auf den Romankorpus trainiert, sie enthalten jeweils $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 150, 180, 200, 250, 300, 400, 500\}$ Topics. Die NPMI-Werte der Top-10-Wörter der 2650 Topics werden dann berechnet, der

[73] 1800 ist die durchschnittliche Textlänge der 2000 Zeitungsartikel.

Durchschnittswert repräsentiert den Kontroll-Wert: -0,1153. Der NPMI-Kontroll-Wert wird in den folgenden Visualisierungen durch eine schwarze Linie dargestellt.

Aus technischen Gründen (die zufällige Initialisierung bei der Zuweisung von Topics und aufgrund des Gibbs-Samplings) sind zwei Topic-Modelle eines Korpus nicht völlig identisch, selbst wenn alle Parameter beim Training der Modelle gleich eingestellt werden. Um die möglichen Abweichungen zwischen den Modellen mit gleicher Parametereinstellung sichtbar zu machen, werden für jedes Parametersetting zehn Modelle trainiert.

In den nächsten Kapiteln werden Untersuchungen in Bezug auf die Anzahl der Topics, den Hyperparameter Alpha, die Hyperparameter Optimierung, den Hyperparameter Beta, die Anzahl der Iterationen des Gibbs-Samplings und die Chunk-Length vorgestellt.

7.1 Anzahl der Topics

In diesem Kapitel bezieht sich das Experiment auf den Einfluss der Topic-Anzahl $\boldsymbol{T}$ auf die Qualität des Modells. Das Setting der Anzahl der Topics war $\boldsymbol{T} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$.

7.1.1 Dokumentklassifikation

In Abbildung 7.1 werden die Klassifikationsergebnisse durch Boxplots visualisiert. Zunächst ist zu beobachten, dass der niedrigste F1-Wert knapp 0,990 beträgt, während diese Kennzahl ansonsten meist zwischen 0,992 und 0,998 liegt. Wenn $\boldsymbol{T}$ von 10 auf 150 erhöht wird, steigen die Mediane der F1-Werte von ca. 0,9955 auf knapp 0,997. Da der F1-Wert der BoW-basierten Klassifikation bei 0,992 liegt ist es wenig erstaunlich, dass die Topic-Modeling-basierte Klassifikation hier erfolgreich ist. In den meisten Fällen erzielt sie sogar bessere Ergebnisse als die BoW-basierte Klassifikation. Im Vergleich zum Ergebnis der Untersuchung auf dem Zeitungskorpus in Kapitel 6.1.1, dass die Klassifikationsergebnisse ab $\boldsymbol{T} = 80$ sich weder zum Besseren noch zum Schlechteren verändert haben, lässt sich hier zudem ein unterschiedliches Phänomen beobachten. Nachdem die F1-Werte den höchsten Punkt erreicht hat, ist eine Absenkung der F1-Werte ab $\boldsymbol{T} = 150$ in der Visualisierung sofort zu beobachten. Offenbar ist die Leistungsfähigkeit der Topic-Modeling-basierte Dokumentklassifikation abhängig vom Untersuchungskorpus und die Topic-Modelle mit mehr Topics können nicht immer die

besseren Klassifikationsergebnisse garantieren. Außerdem haben die F1-Werte eine große Abweichung, wenn die Anzahl der Topics auf zehn eingestellt wird. Im Vergleich zu den anderen Settings der Anzahl der Topics haben die zufällige Initialisierung und das Gibbs-Sampling offenbar größere Auswirkungen auf die Topic-Modelle mit zehn Topics. In dieser Untersuchung erzielt die Klassifikation im besten Fall eine Accuracy von 0,9976 und einen F1-Wert von 0,9975, was über der BoW-basierten Baseline (Accuracy von 0,993 / F1-Wert von 0,992) liegt.

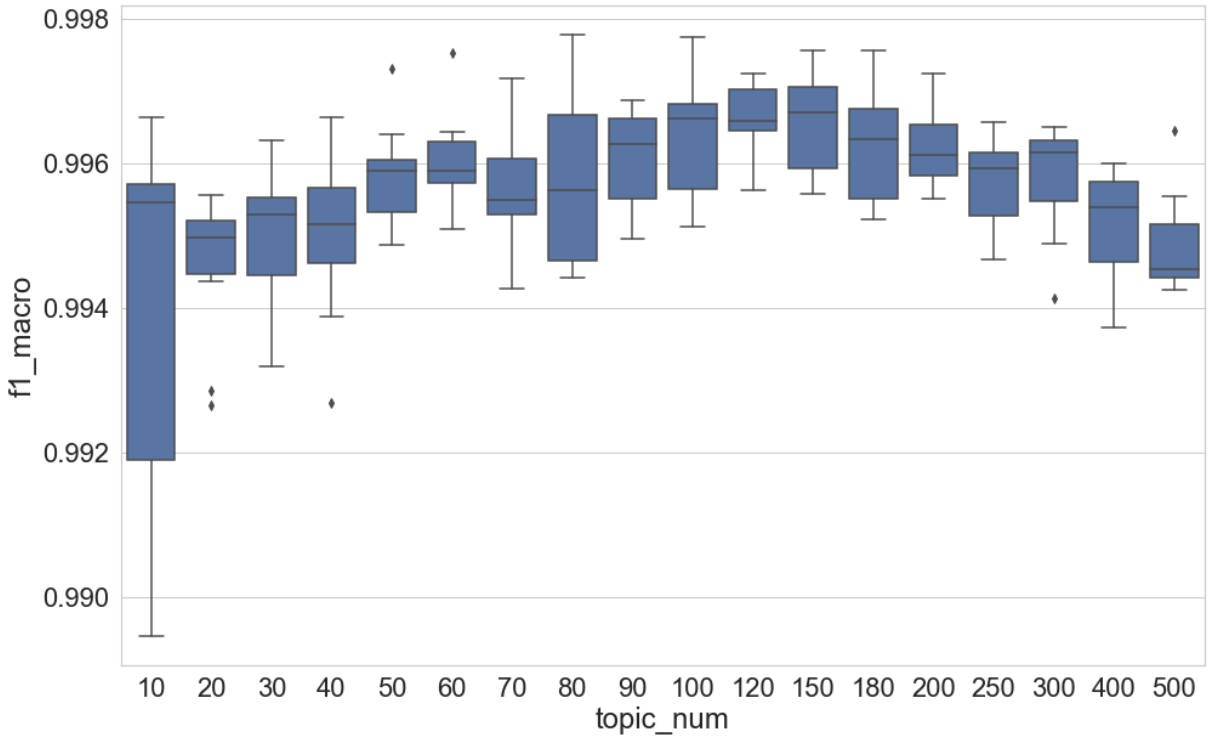

Abbildung 7.1 F1 (Makro)-Werte der Dokument-Klassifikation im Verhältnis zur Anzahl der Topics

7.1.2 Topic-Kohärenz

In Abbildung 7.2 wird die Veränderung der NPMI-Werte-Verteilungen im Verhältnis zur Anzahl der Topics visualisiert. Wenn die Anzahl der Topics von zehn auf 80 erhöht wird, vergrößert sich die Spannweite der Verteilung. Die Maximal- und Minimalwerte der Verteilung werden jeweils von ca. 0,03 und ca. -0,3 auf ca. 0,1 bzw. ca. -0,42 erweitert. Ansonsten zeigt das Ergebnis leichte Abweichungen zu dem der Untersuchung auf dem Zeitungskorpus. Es lassen sich in der Abbildung keine weiteren systematischen Änderungen beobachten: Der

Median der Verteilung bleibt trotz Schwankungen bei ca. -0,15 und es gibt bei jedem Setting der Topic-Anzahl mindestens 25 % der Topics, deren NPMI-Wert höher als der NPMI-Kontroll-Wert ist. Im Vergleich dazu ist in Abbildung 7.3 der klare Trend zu erkennen, dass es mit der Erhöhung der Anzahl der Topics immer mehr von ihnen gibt, deren NPMI-Wert größer als der NPMI-Kontroll-Wert ist. Werden allerdings die absoluten Zahlen normalisiert (also geteilt durch die Anzahl der Topics), ist trotz einiger Schwankungen ein absinkender Trend zu beobachten. Dieses Ergebnis stimmt mit dem Resultat der Untersuchungen auf dem Zeitungskorpus überein. Je mehr Topics beim Topic Modeling trainiert werden, desto mehr kohärente Topics gibt es. Gleichzeitig erhält man eine noch höhere Anzahl nicht-kohärenter Topics.

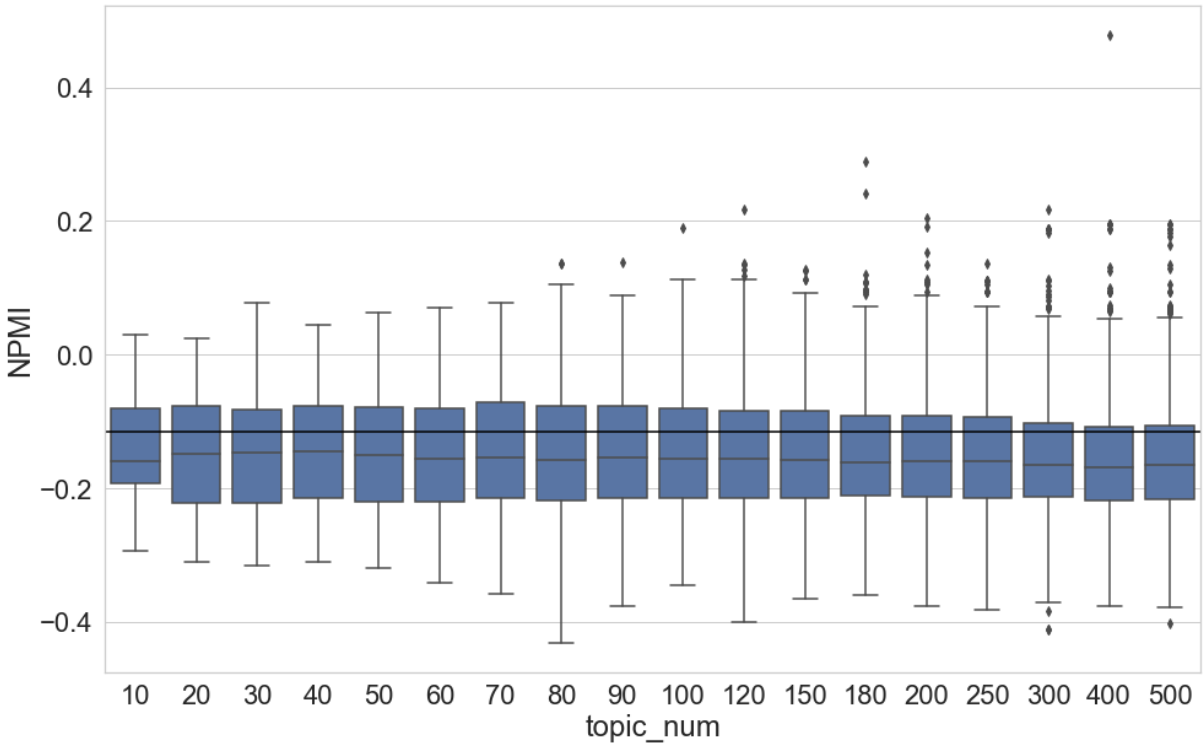

Abbildung 7.2 NPMI-Wert-Verteilung der Topics im Verhältnis zur Anzahl der Topics

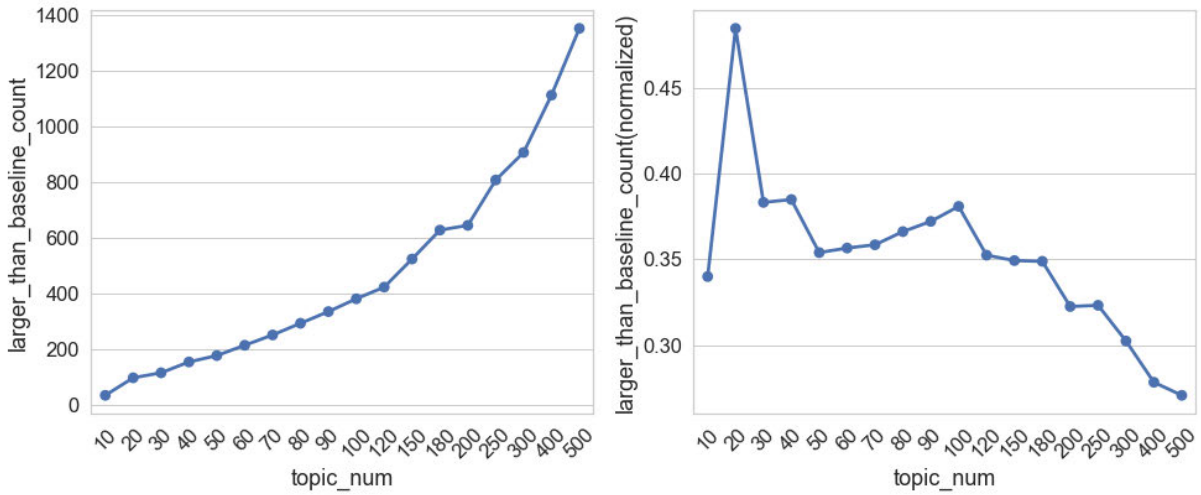

Abbildung 7.3 Anzahl der Topics, deren NPMI-Wert größer als der NPMI-Kontroll-Wert ist (links: absolute Anzahl; rechts: Prozentzahl)

Werden die obigen Untersuchungsergebnisse kombiniert ist festzustellen, dass die Anzahl der Topics für die zu untersuchende Artikelsammlung auf 80 oder 150 eingestellt werden könnte. Die Klassifikationsergebnisse erreichen ihr Maximum und ihren maximalen Median bei $\boldsymbol{T} = 80$ bzw. $\boldsymbol{T} = 150$ und die Modelle enthalten zudem auch nicht sehr viele weniger kohärente Topics.

7.2 Hyperparameter Alpha

In diesem Kapitel bezieht sich das Experiment auf den Einfluss des Hyperparameters Alpha[74] auf die Qualität des Topic-Modells. Das Setting von Alpha ist $\boldsymbol{\alpha} \in \{0{,}0001, 0{,}0005, 0{,}001, 0{,}005, 0{,}01, 0{,}05, 0{,}1, 0{,}5, 1, 5, 10, 20, 30, 40, 50, 100\}$.

7.2.1 Dokumentklassifikation

In Abbildung 7.4 wird das Untersuchungsergebnis visualisiert. Die Anzahl der Topics wird zuerst auf 80 und 150 eingestellt. Auf der linken Seite der Abbildung sind die gesamten Ergebnisse der Klassifikation dargestellt. Es ist zu beobachten, dass die Klassifikation fast immer perfekt funktionieren kann, wenn Alpha zwischen 0,0001 und 1,0 eingestellt wird. Im Vergleich dazu sinken die F1-Werte schrittweise ab (von ca. 0,97 auf kleiner als 0,1), wenn Alpha von 5,0 auf 100,0 erhöht wird. Die Verteilungen der F1-Werte bei einem Alpha zwischen

[74] Hier bezieht sich das Alpha auf den Alpha-Wert jedes einzelnen Topics.

0,0001 und 5,0 werden auf der rechten Seite der Abbildung erneut visualisiert, um die Ergebnisse genauer darzustellen. Eine perfekte Klassifikation wird zwar nicht erzielt, dennoch liegen die F1-Werte sämtlich über 0,99, wenn Alpha größer als 0,0005 und kleiner als 0,5 ist. Darüber hinaus ist ein aufsteigender Trend zu beobachten, wenn Alpha von 0,0001 auf 0,1 erhöht wird. Bei α = 5,0 funktioniert die Klassifikation deutlich schlechter, wenn die Anzahl der Topics auf 150 eingestellt wird. Vermutlich werden die Topic-Modelle mit mehr Topics stärker durch die Veränderung von Alpha beeinflusst. Diese Annahme kann durch die Ergebnisse in Abbildung 7.5 bestätigt werden.

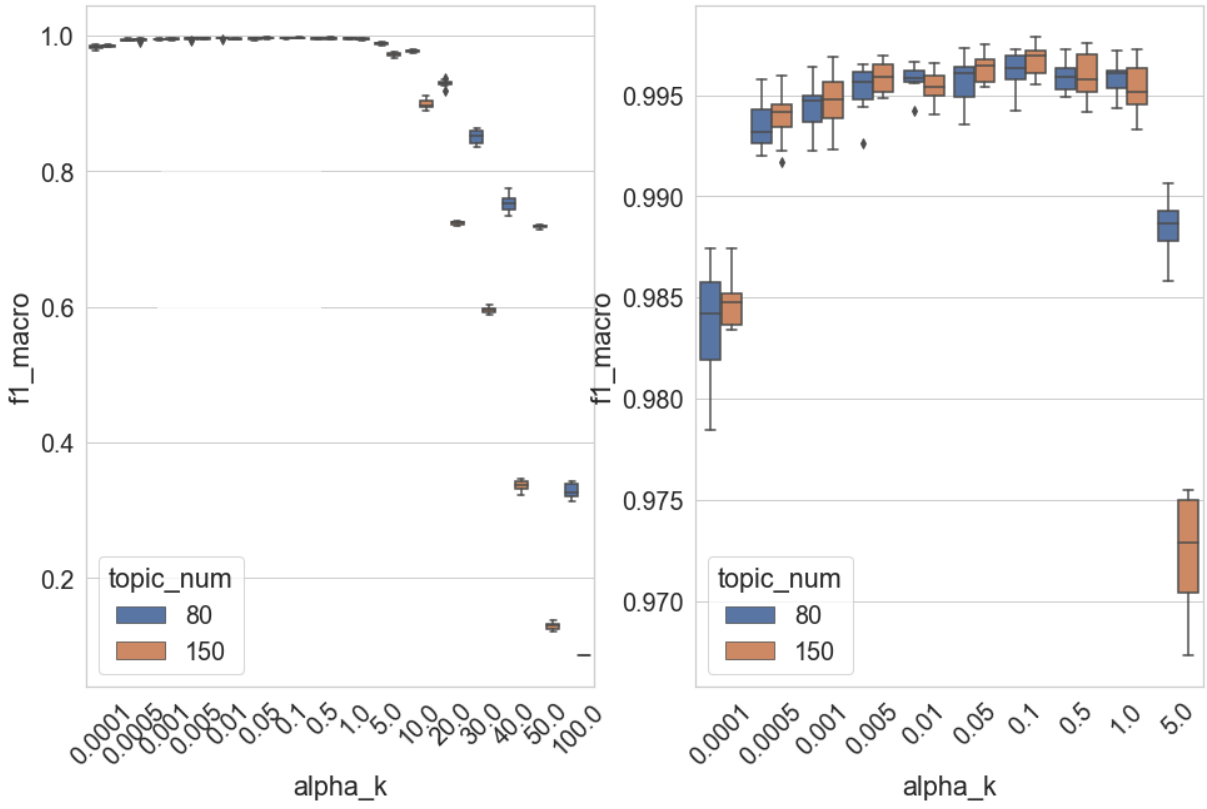

Abbildung 7.4 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Alpha

Wenn die Anzahl der Topics stärker variiert wird, ist in Abbildung 7.5 zu beobachten, dass die Klassifikationsergebnisse ab α = 0,5 schlechter werden. Wie in der vorherigen Abbildung werden die Gesamtergebnisse auf der linken Seite visualisiert. Unabhängig von der Anzahl der Topics kann der absinkende Trend mit der Erhöhung von Alpha beobachtet werden. Offenbar werden die Klassifikationsergebnisse stärker beeinflusst, wenn Topic-Modelle mit mehr Topics trainiert werden. Bei den Topic-Modellen mit zehn Topics bleiben die F1-Werte noch immer höher als 0,9, wenn Alpha auf 100 eingestellt wird. Im Gegensatz dazu fallen sie bei α = 100

unter 0,1, wenn die Anzahl der Topics größer als 90 eingestellt wird. Die Verteilungen der F1-Werte bei Alpha zwischen 0,0001 und 0,1 werden auf der rechten Seite der Abbildung nochmals gesondert gezeigt. Hier ist ein aufsteigender Trend mit der Erhöhung von Alpha bei fast jedem Setting der Topic-Anzahl sichtbar. Ab α = 0,0005 liegen alle Werte über 0,99, der maximale F1-Wert wird bei α = 0,1 erzielt.

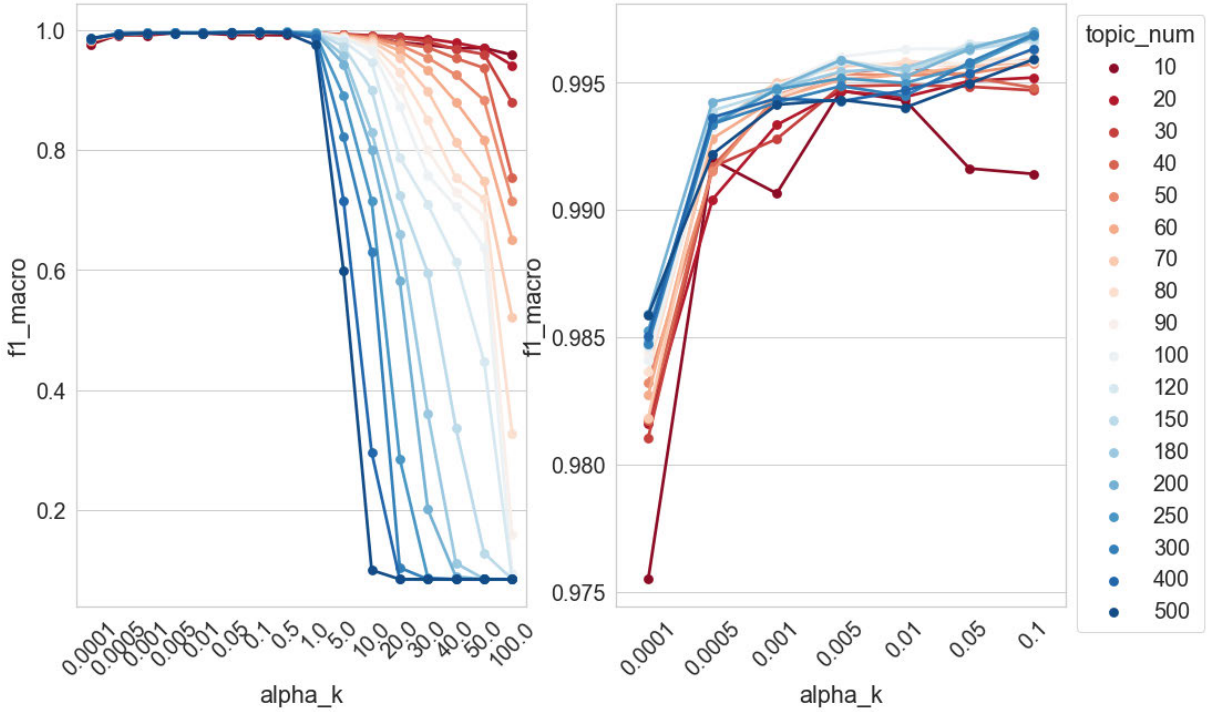

Abbildung 7.5 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Alpha und Anzahl der Topics

Durch den Vergleich der obigen Ergebnisse mit den vorherigen Tests in Kapitel 6.2.1 (Untersuchung auf dem Zeitungskorpus) lässt sich feststellen, dass die Einstellung von Alpha immer zwischen 0,05 und 0,1 liegen sollte, unabhängig von dem Korpus, auf dem ein Topic-Modell trainiert wird. Der Wert von Alpha darf eventuell noch kleiner als 0,05, aber nicht größer als 5 sein. Auch wenn der Wert von Alpha auf größer als 5 eingestellt werden muss, sollte das Topic-Modelle mit weniger Topics trainiert werden. Dadurch wird sichergestellt, dass die besten Topic-Modeling-basierten Klassifikationsergebnisse erzielt werden können.

7.2.2 Topic-Kohärenz

In Abbildung 7.6 wird die Verteilung der NPMI-Werte im Verhältnis zu Alpha visualisiert. Die Anzahl der Topics wird zuerst auf 80 und 150 eingestellt. Als klarer Trend ist zunächst zu beobachten, dass es zunehmend mehr nicht-kohärente Topics gibt, wenn der Wert von Alpha schrittweise von 1 bis zu 100 erhöht wird. Ab α = 5 ist der Unterschied zwischen den blauen und den gelben Boxplots deutlich zu erkennen. Ab α = 20 liegen mehr als 75 % der NPMI-Werte bei jeder Kombination von Alpha und der Anzahl der Topics unterhalb des NPMI-Kontroll-Werts. Die Verteilung der mittleren 50 % der NPMI-Werte haben sich kaum verändert, wenn Alpha zwischen 0,0001 und 0,1 eingestellt wird. Die Spannweite der Verteilung ist größer, wenn Alpha von 0,0001 auf 0,1 erhöht wird. Außerdem erreicht der Median der Verteilung den höchsten Punkt, wenn Alpha zwischen 0,5 und 1 beträgt.

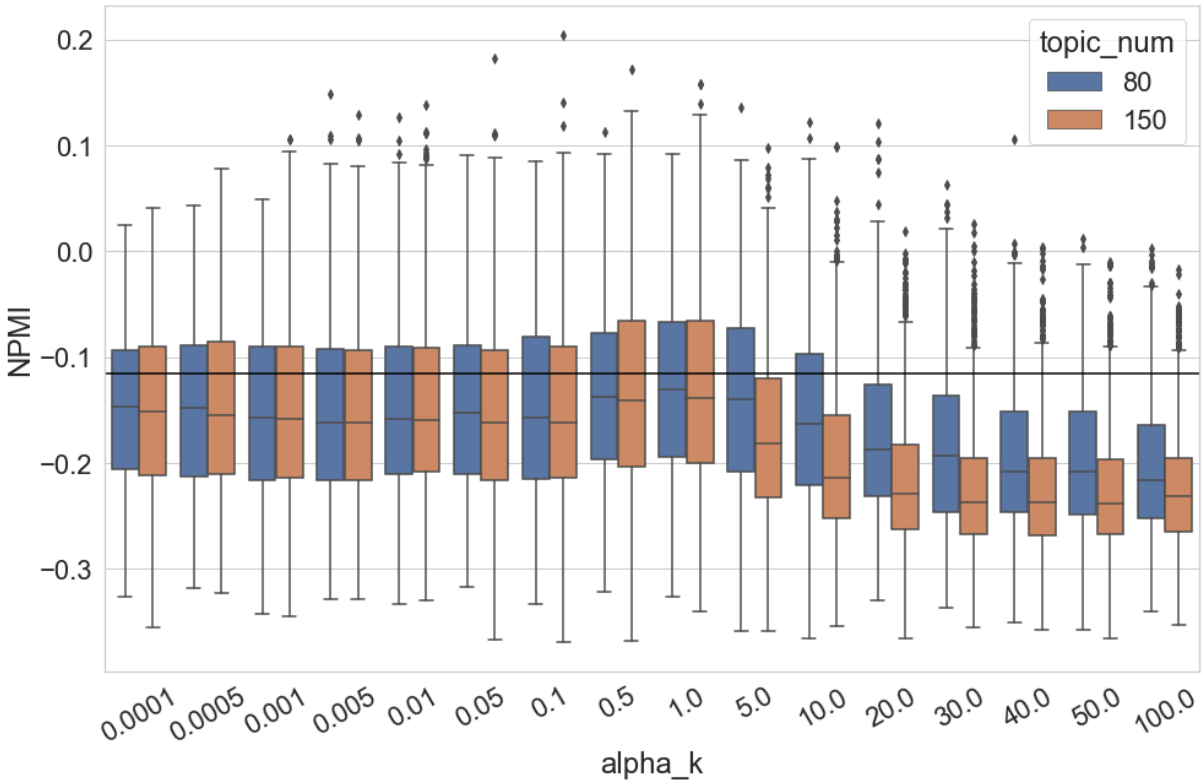

Abbildung 7.6 NPMI-Werte-Verteilung der Topics im Verhältnis zu Alpha

Wenn die Anzahl der Topics in höherem Maß variiert wird, ist in Abbildung 7.7 zunächst zu beobachten, dass die mittleren 50 % der Daten bei jedem Setting der Topic-Anzahl meistens zwischen ca. -0,08 und -0,22 liegen, wenn Alpha zwischen 0,0001 und 0,01 beträgt. Die Spannweite der Verteilung ist dabei größer, wenn Topic-Modelle mit mehr Topics trainiert werden. Topics mit den höchsten NPMI-Werten tauchen immer in den Topic-Modellen auf, die

weder zu viele noch zu wenig Topics enthalten (in dieser Untersuchung zwischen 100 und 200 Topics). Mit einer weiteren Erhöhung von Alpha zeigen die Verteilungen einen absinkenden Trend, wenn die Anzahl der Topics größer als 30 eingestellt wird. Bei $\boldsymbol{T}$ = 500 z. B. sinken die mittleren 50 % der NPMI-Werte mit der Erhöhung von Alpha allmählich ab. Bei $\boldsymbol{\alpha}$ = 100 liegen die meisten NPMI-Werte unterhalb des Kontroll-Werts. Im Vergleich dazu wird die NPMI-Werte-Verteilung der Topic-Modelle mit weniger Topics durch die Veränderung von Alpha weniger beeinflusst. Bei $\boldsymbol{T}$ = 10 steigt der Median mit der Erhöhung von Alpha von ca. -0,14 auf ca. -0,11, während sich die Spannweite der Verteilung kaum ändert.

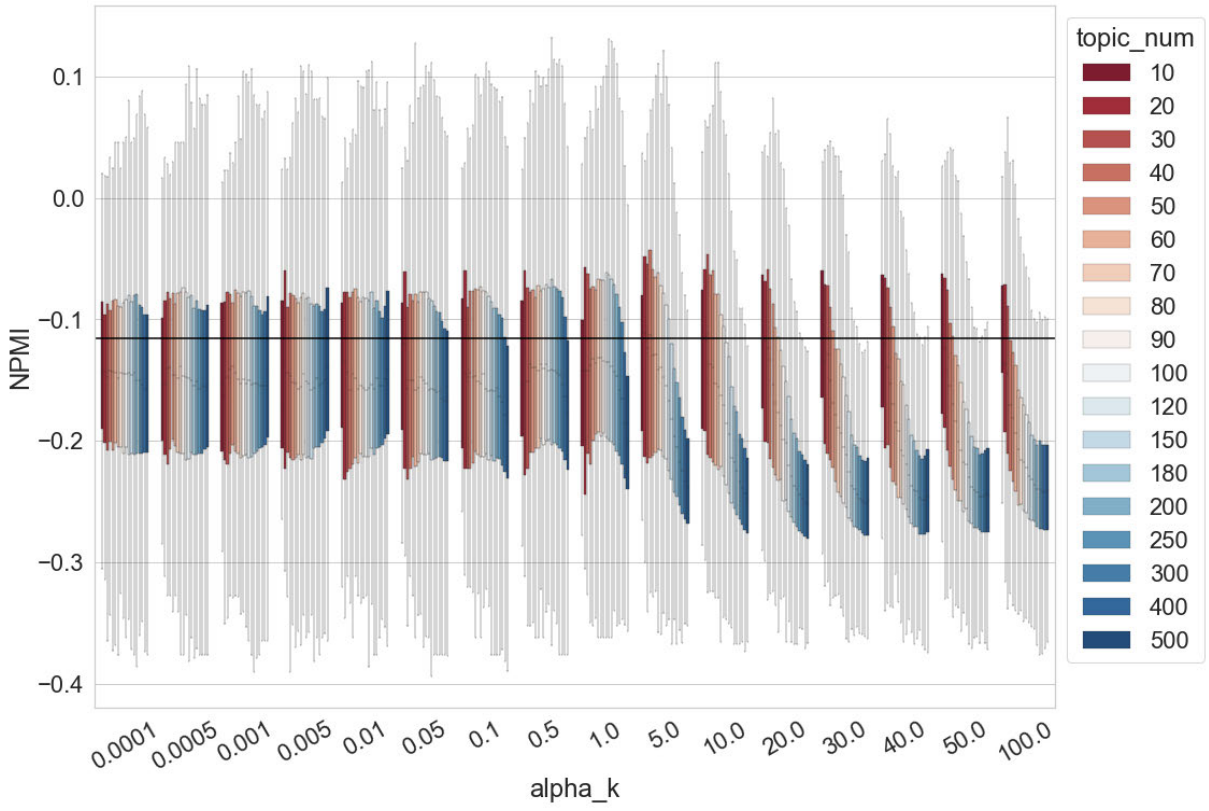

Abbildung 7.7 NPMI-Werte-Verteilung der Topics im Verhältnis zu Alpha und Anzahl der Topics

Ähnlich wie bei den Schlussfolgerungen in Kapitel 6.2.2 lässt sich mit Blick auf die Resultate der Untersuchung in diesem Kapitel feststellen, dass es eine gute Wahl ist, den Wert von Alpha jedes Topics beim Training eines Topic-Modells kleiner als 1 einzustellen, wenn es das Ziel ist, mehr kohärente Topics zu erhalten.

7.3 Hyperparameter-Optimierung

In diesem Kapitel zielt das Experiment auf den Einfluss der Hyperparameter-Optimierung auf die Qualität des LDA-Modells. Wenn Topic Modeling mit MALLET durchgeführt wird, müssen zwei Parameter eingestellt werden: „Optimize Burn-in“ und „Optimize Interval“, um die Hyperparameter-Optimierung durchzuführen.[75] Der erste Parameter legt die Anzahl der Iterationen fest, die vor der ersten Schätzung des Hyperparameters Alpha ausgeführt werden müssen. Der zweite Parameter definiert die Anzahl der Iterationen zwischen der Neuschätzung des Hyperparameters Alpha. Wie im Kapitel 6.3 werden die beiden Parameter „Optimize Burn-in“ und „Optimize Interval“ hier auch als ***OB*** und ***OI*** bezeichnet. Das Setting der Parameter in dieser Untersuchung ist ***OB*** ∈ {20, 50, 100, 200, 300, 400, 500} und ***OI*** ∈ {20, 50, 100, 200, 300, 400, 500}. Durch die Kombination von ***OB*** und ***OI*** wird analysiert, wie die Qualität des Topic-Modells sich ändert, wenn der Hyperparameter Alpha beim Topic Modeling bereits sehr früh (***OB*** =20) und etwas später (***OB*** =500) optimiert werden. Außerdem wird untersucht, inwiefern sich die Qualität des Topic-Modells verändert, wenn der Hyperparameter Alpha beim Topic Modeling sehr häufig (***OI*** =20) oder weniger häufig (***OI*** =500) optimiert werden.

7.3.1 Dokumentklassifikation

In Abbildung 7.8 werden die Klassifikationsergebnisse in Bezug auf „Optimize Burn-in“ und „Optimize Interval“ dargestellt. Die Anzahl der Topics wird auf 80 und 150 eingestellt. Bei den anderen Settings der Anzahl der Topics (10 bis 500) sind die Ergebnisse ähnlich, sie werden deshalb nicht visualisiert. Das Ergebnis stimmt mit der Untersuchung auf dem Zeitungskorpus überein und verdeutlicht, dass es keinen systematischen Zusammenhang zwischen der Topic-Modeling-basierten Dokumentklassifikation und der Einstellung der beiden Parameter ***OB*** und ***OI*** gibt. Es ist also festzuhalten, dass die Topic-Modeling-basierte Dokumentklassifikation in dieser Untersuchung nicht systematisch davon beeinflusst wird, wann und wie oft der Hyperparameter Alpha optimiert wird.

Die Klassifikationsergebnisse werden hier auch verglichen, wenn Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. In dieser Untersuchung wird die Anzahl der Topics beim Training der Modelle variiert, ***OB*** und ***OI*** werden jeweils auf 200 eingestellt. In Abbildung 7.9 ist zu beobachten, dass die meisten F1-Werte über 0,99 liegen, ein Einfluss der

[75] „Hyperparameter Optimization“, unter http://mallet.cs.umass.edu/topics.php, (07.06.2020).

Hyperparameter-Optimierung ist schwerlich festzustellen. Wenn die Anzahl der Topics z. B. auf 50, 60, 70 oder 150 eingestellt wird, funktioniert die Klassifikation besser ohne Hyperparameter-Optimierung. Im Gegensatz dazu kann die Hyperparameter-Optimierung eine bessere Klassifikation garantieren, wenn die Anzahl der Topics größer als 300 ist. Bei $\boldsymbol{T}$ = 20, 90, oder 200 gibt es keinen wesentlichen Unterschied zwischen den Ergebnissen der beiden Gruppen.

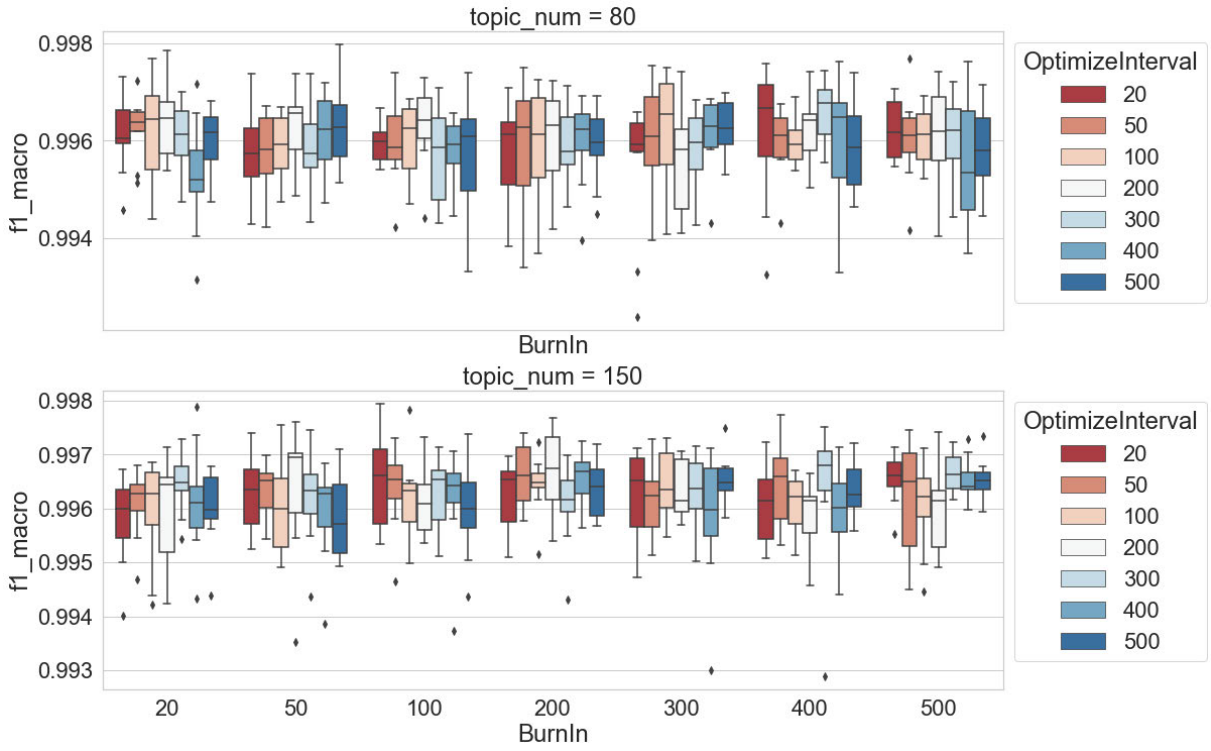

Abbildung 7.8 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ (oben: $\boldsymbol{T}$ = 80, unten: $\boldsymbol{T}$ = 150)

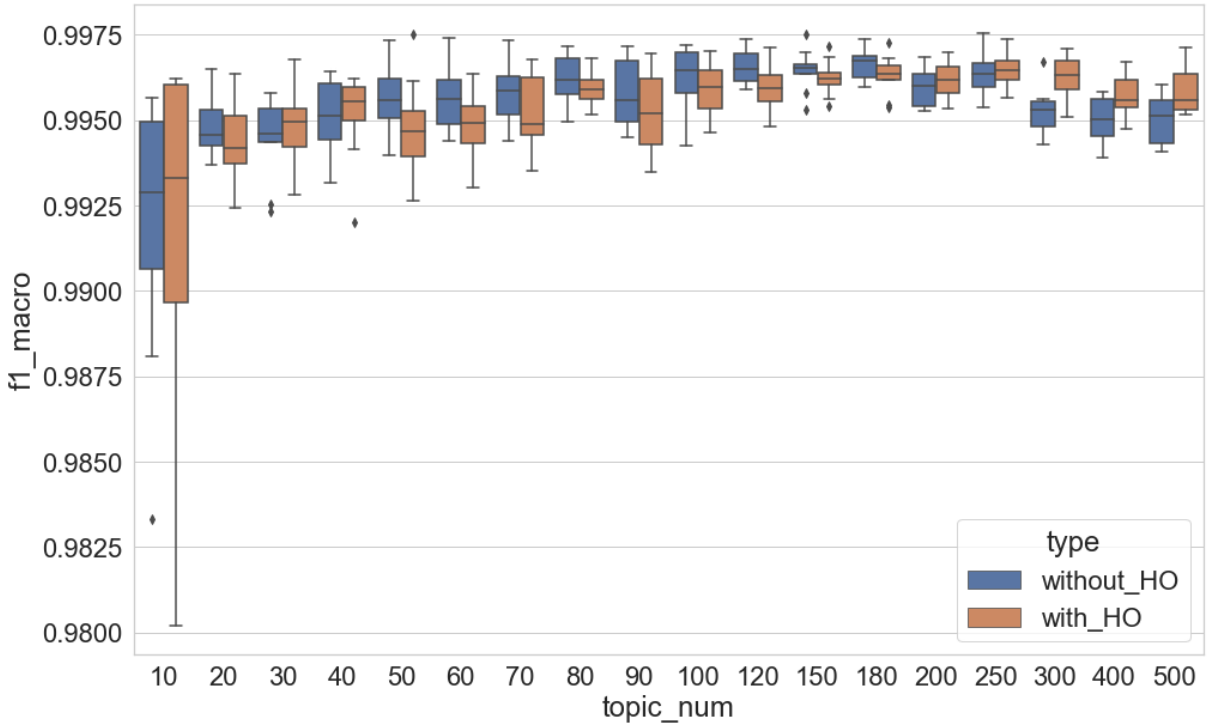

Abbildung 7.9 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics

Das Ergebnis im Kapitel 6.3.1 zeigt, dass, wenn die Chunk-Length größer als 200 Tokens ist, der Unterschied zwischen den Topic-Modellen, die ohne oder mit Hyperparameter-Optimierung trainiert werden, nicht mehr deutlich ist. Der Untersuchungskorpus in diesem Kapitel besteht aus Textsegmenten, die mindestens 1800 Tokens enthalten. Bis jetzt stimmen die Untersuchungsergebnisse auf dem Romankorpus mit den Untersuchungen auf dem Zeitungskorpus überein. Sie besagen, dass die Hyperparameter-Optimierung auf die Topic-Modeling-basierte Klassifikation von langen Textsegmenten keinen klaren Einfluss hat. Der nächste Schritt ist es deshalb zu überprüfen, ob die Untersuchung auf dem Romankorpus das bereits zuvor erzielte Resultat bestätigt, nämlich, dass die Hyperparameter-Optimierung die Klassifikation von kurzen Textsegmenten verbessern kann.

Für die Untersuchungen werden die 439 Heftromanen in Chunks zerlegt; das Setting der Chunk-Length ist $\boldsymbol{C} \in \{10, 20, 50, 80, 100, 120, 150, 180, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1800, 2000\}$. Bei jedem Setting werden zehn Topic-Modelle mit und ohne Hyperparameter-Optimierung trainiert. Die ***OB*** und ***OI*** werden jeweils auf 200 eingestellt, wenn die Hyperparameter-Optimierung beim Training eingesetzt wird. Die Anzahl der Topics

wird auf 80 festgelegt. Die Ergebnisse bei anderen Settings der Topic-Anzahl sind ähnlich und werden deshalb nicht visualisiert. Die Klassifikationsergebnisse werden in Abbildung 7.10 dargestellt. Die Klassifikationen können klar bessere Ergebnisse erzielen, wenn die Hyperparameter-Optimierung eingesetzt wird und die Chunk-Length kürzer als 200 ist. Ab ***C*** = 300 ist der systematische Unterschied nicht mehr klar zu beobachten. Zusammenfassend kann man feststellen, dass es besonders sinnvoll ist, die Hyperparameter-Optimierung einzusetzen, wenn Topic-Modelle auf einer Sammlung von kurzen Texten (kürzer als 200 Tokens) trainiert werden. Dies können bessere Klassifikationsergebnisse unabhängig vom Korpus garantieren.

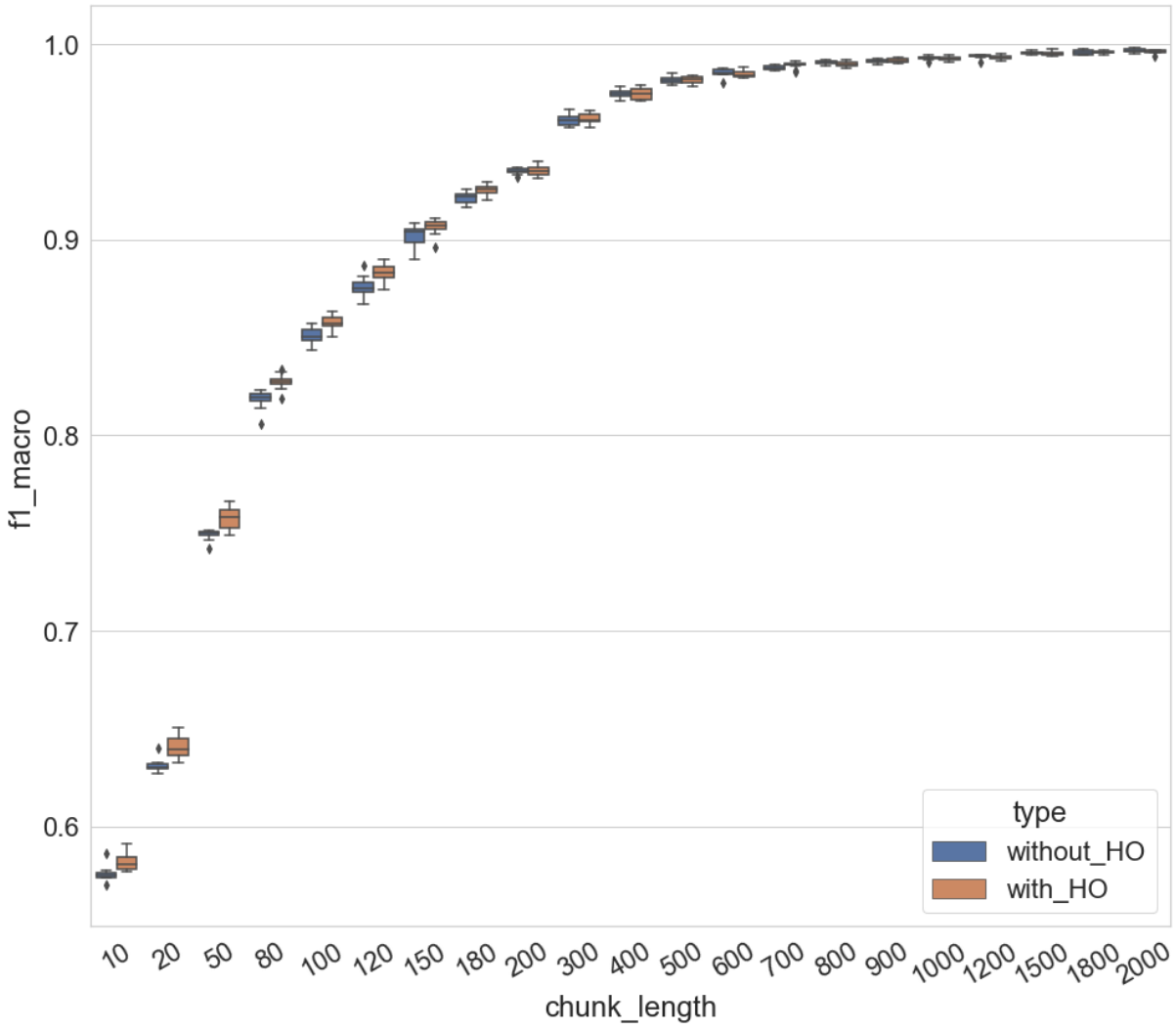

Abbildung 7.10 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Hyperparameter-Optimierung und Chunk-Length (80 Topics)

7.3.2 Topic-Kohärenz

Nach den Untersuchungsergebnissen im letzten Unterkapitel ist zu erwarten, dass die Hyperparameter-Optimierung auch keinen deutlichen Einfluss auf die Topic-Kohärenz hat, wenn Topic-Modelle auf dem Romankorpus trainiert werden. Abbildung 7.11 zeigt die NPMI-Werte-Verteilungen bei allen Kombinationen von ***OB*** und ***OI***, während die Anzahl der Topics auf 80 und 150 eingestellt wird. In der Darstellung lässt sich erkennen, dass die NPMI-Werte-Verteilungen fast immer im gleichen Bereich liegen. Bei den anderen Settings der Anzahl der Topics (10 bis 500) sind die Ergebnisse ähnlich, sie werden deshalb nicht visualisiert. Die Ergebnisse hier decken sich mit den Resultaten der Untersuchung auf dem Zeitungskorpus: Die Topic-Kohärenz wird nicht systematisch davon beeinflusst, wann und wie oft die Hyperparameter der Topic-Dokument-Verteilung optimiert werden.

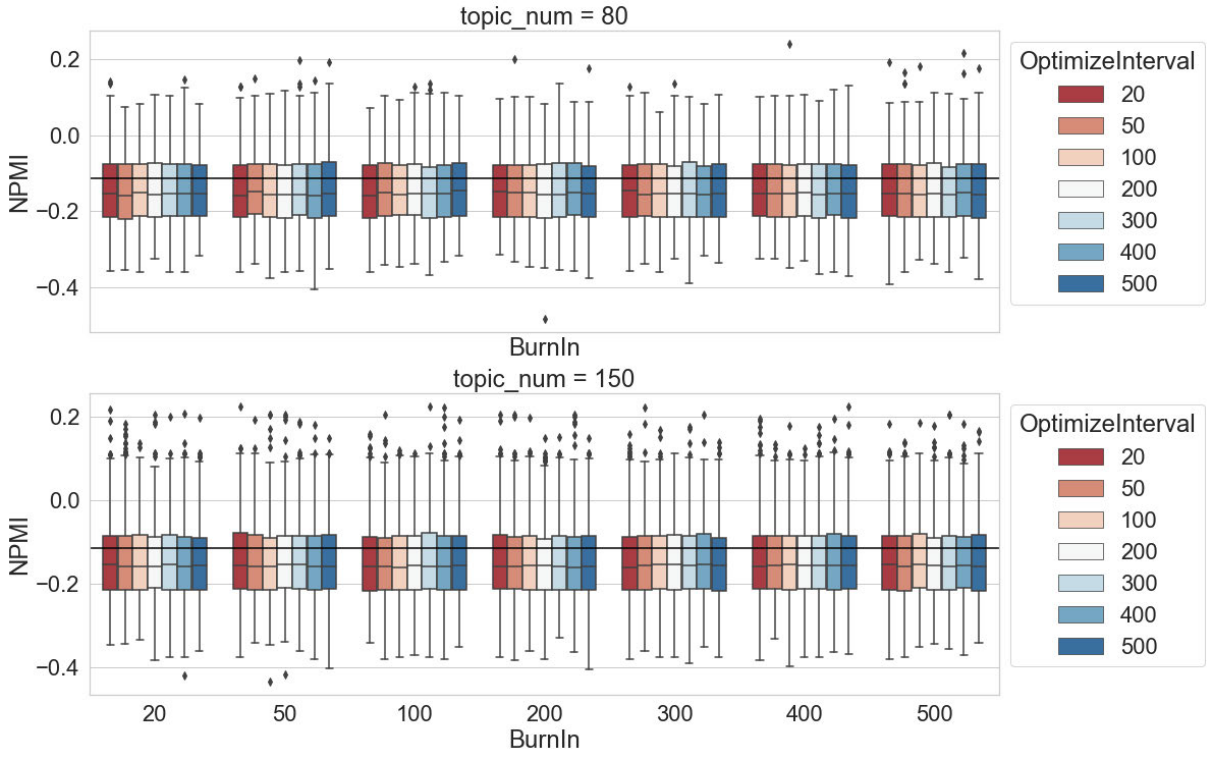

Abbildung 7.11 NPMI-Werte-Verteilung der Topics im Verhältnis zu „Optimize Interval“ und „Optimize Burn-in“ und Anzahl der Topics (oben: ***T*** = 80, unten: ***T*** = 150)

Die Topic-Kohärenz wird auch verglichen, wenn Topic-Modelle ohne und mit Hyperparameter-Optimierung trainiert werden. In dieser Untersuchung wird die Anzahl der Topics beim Training der Modelle variiert und die ***OB*** und ***OI*** werden jeweils auf 200 eingestellt. In Abbildung 7.12 wird das Vergleichsergebnis visualisiert. Die blauen und gelben Boxplots stehen jeweils für die Topic-Kohärenz der Modelle, die ohne und mit

Hyperparameter-Optimierung trainiert werden. Es ist zu beobachten, dass es keinen systematischen Unterschied zwischen den beiden Gruppen gibt. Bei ***T*** = 20 z. B. ist der Median der blauen Boxplots höher, bei ***T*** = 60 und 90 ist die Spannweite der blauen Boxplots größer. Derlei Differenzen korrelieren allerdings nicht mit der Veränderung des Parameter-Settings.

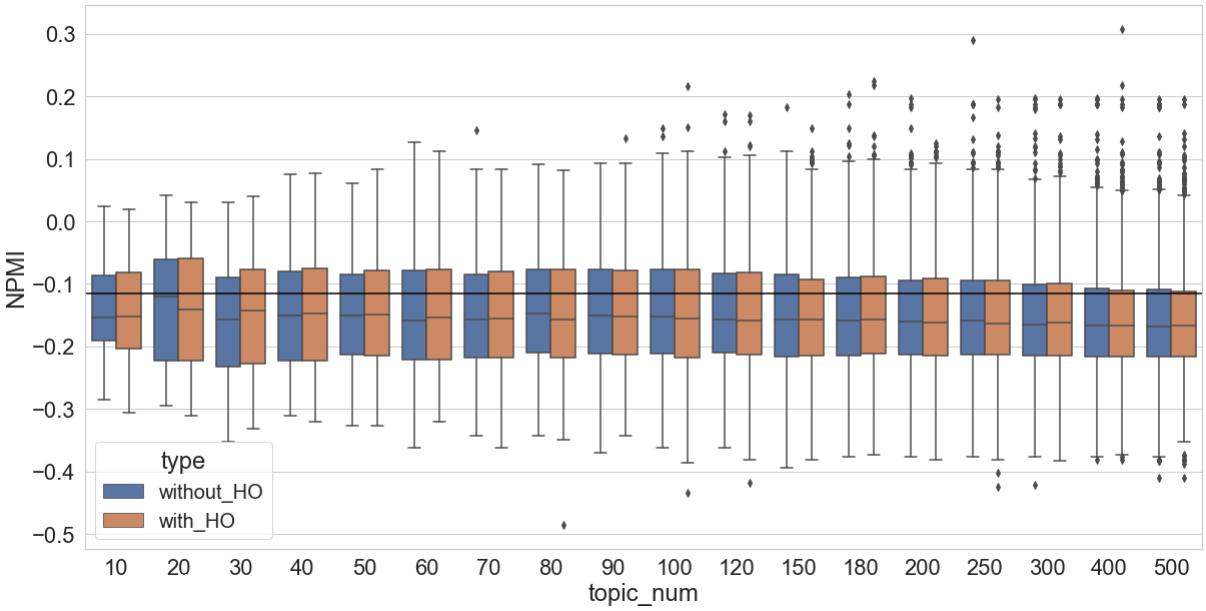

Abbildung 7.12 NPMI-Werte-Verteilung der Topics im Verhältnis zu Hyperparameter-Optimierung und Anzahl der Topics

Die letzte Untersuchung in diesem Kapitel soll überprüfen, ob die Verteilungen der Topic-Kohärenz-Werte beim Einsatz der Hyperparameter-Optimierung variieren, wenn Topic-Modelle auf Textsammlungen mit unterschiedlichen Chunk-Lengths trainiert werden. Für die Untersuchungen wird der Heftromankorpus in Chunks zerlegt, das Setting der Chunk-Length ist ***C*** ∈ {10, 20, 50, 80, 100, 120, 150, 180, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1800, 2000}. Bei jedem Setting der Chunk-Length werden zehn Topic-Modelle mit und ohne Hyperparameter-Optimierung trainiert. Die ***OB*** und ***OI*** werden jeweils auf 200 eingestellt, wenn die Hyperparameter-Optimierung beim Training eingesetzt wird. Die Anzahl der Topics wird auf 80 festgelegt.

Im Gegensatz zu den Ergebnissen der Klassifikation im vorangegangenen Kapitel (Abbildung 7.10) ist in Abbildung 7.13 kein klarer Unterschied erkennbar. Unabhängig von der Einstellung der Chunk-Length und davon, ob die Hyperparameter-Optimierung eingesetzt wird oder nicht, existieren keine deutlichen Differenzen bei den NPMI-Werte-Verteilungen. Das Ergebnis

unterscheidet sich von dem der Untersuchung auf dem Zeitungskorpus in Kapitel 6.3.2. Die Hyperparameter-Optimierung erlaubt dem Modell, sich besser an die Daten anzupassen, indem zugelassen wird, dass einige Topics dominanter als andere sind. Anders formuliert werden hier einige Topics eher allgemein und stark mit mehreren Dokumenten verbunden, andere sind eher spezifisch und konzentrieren sich auf einen bestimmten, kleinen Teil der Dokumente. Offenbar hat diese Unterscheidung, anders als bei den Topics bei der Untersuchung des Zeitungskorpus, weniger Einfluss auf die Kohärenz der Topics aus diesem Heftromankorpus.

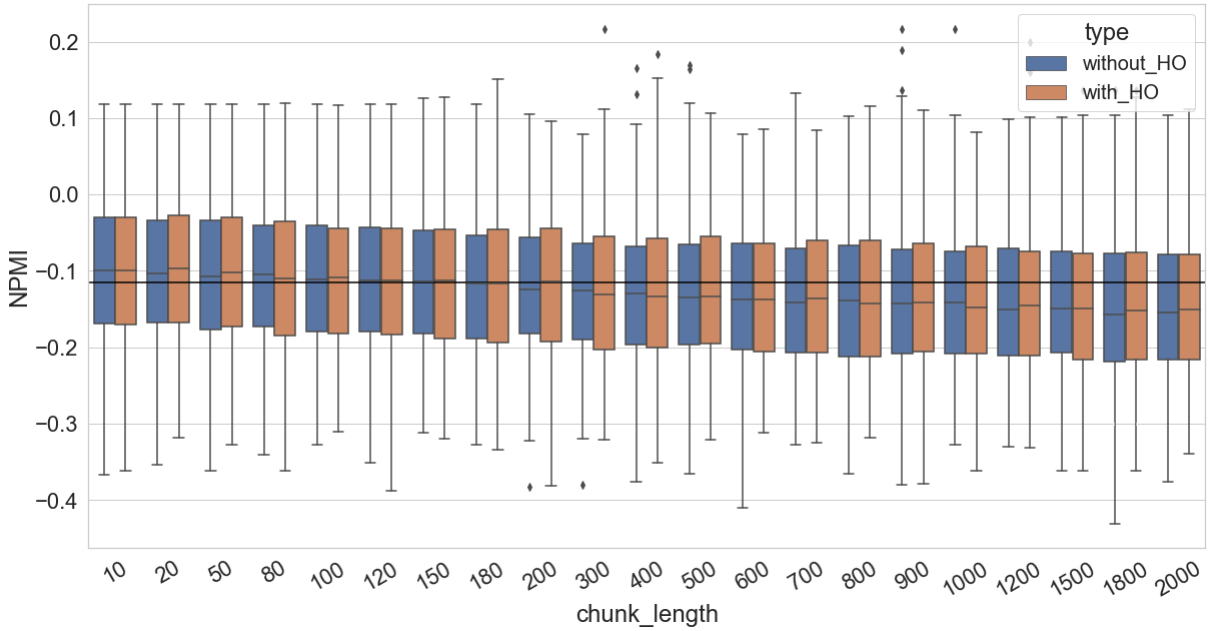

Abbildung 7.13 NPMI-Werte-Verteilung der Topics im Verhältnis zu Hyperparameter-Optimierung und Chunk-Length

7.4 Hyperparameter Beta

In diesem Kapitel bezieht sich das Experiment auf den Einfluss des Hyperparameters Beta auf die Qualität des Topic-Modells. Das Setting von Beta ist $\boldsymbol{\beta} \in$ {0,00001 (1e-05), 0,00005 (5e-05), 0,0001, 0,0005, 0,001, 0,005, 0,01, 0,02, 0,05, 0,08, 0,1, 0,2, 0,5, 0,8, 1, 2, 5, 8, 10, 20, 50}. In dieser Untersuchung wird die Anzahl der Topics zunächst auf 80 und 150 eingestellt.

7.4.1 Dokumentklassifikation

In Abbildung 7.14 werden die Klassifikationsergebnisse im Verhältnis zu Beta visualisiert. Auf der linken Seite der Abbildung ist das Gesamtergebnis der Untersuchung dargestellt: Die

Klassifikation kann fast perfekt funktionieren, wenn Beta zwischen 0,00005 und 0,1 eingestellt wird. Mit einer weiteren Erhöhung von Beta sinken die F1-Werte sehr schnell ab. Sie liegen sämtlich unterhalb von 0,1, wenn Beta größer als 5 eingestellt wird. Darüber hinaus ist zu beobachten, dass es eine große Abweichung zwischen den F1-Werten bei $\boldsymbol{\beta}$ = 2 gibt. Das bedeutet, dass der Trainingsprozess hier wegen der Einstellung von Beta durch die zufällige Initialisierung bei der Zuweisung von Topics und Gibbs-Sampling stärker beeinflusst wird. Auf der rechten Seite der Abbildung werden die F1-Werte-Verteilungen nochmals visualisiert, wenn Beta zwischen 0,00005 und 0,5 liegt. Außer bei $\boldsymbol{\beta}$ = 0.5 sind fast alle anderen F1-Werte höher als 0,99. Es lässt sich zudem ein aufsteigender Trend beobachten, wenn Beta von 0,00005 auf 0,01 erhöht wird. Mit einer weiteren Erhöhung von Beta werden die Klassifikationsergebnisse allmählich schlechter.

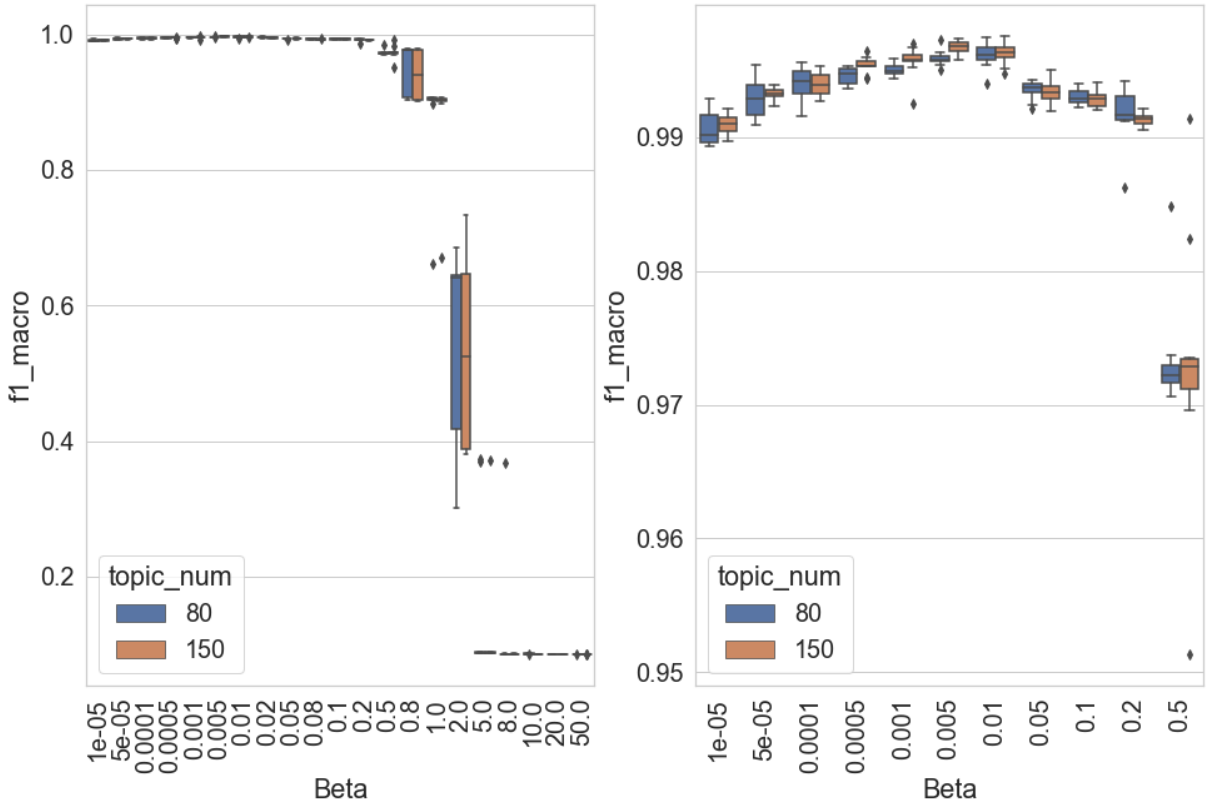

Abbildung 7.14 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Beta

Wenn die Anzahl der Topics stärker variiert wird, sind die Ergebnisse ähnlich. Auf der linken Seite der Abbildung lässt sich erkennen, dass die Klassifikationen sehr gute Resultate erzielen, wenn Beta kleiner als 0,1 ist. Ab $\boldsymbol{\beta}$ = 0,2 beginnen die Kurven abzusinken, bei $\boldsymbol{\beta}$ = 5 liegen die meisten Kurven unterhalb von 0,2. Auf der rechten Seite wird die Situation genauer dargestellt,

wenn Beta zwischen 0,00005 und 0,5 eingestellt wird. Alle Kurven steigen zuerst mit der Erhöhung von Beta auf und beginnen dann ab einem bestimmten Punkt abzusinken. Im Vergleich zu den Topic-Modellen mit weniger Topics (die roten Kurven) können Topic-Modelle mit mehr Topics (die blauen Kurven) bessere Klassifikationsergebnisse erzielen, wenn Beta kleiner als 0,001 ist. Sie werden allerdings zugleich durch die weitere Erhöhung von Beta stärker beeinflusst, ihre Klassifikationsergebnisse verlieren ab $\boldsymbol{\beta}$ = 0,005 schneller an Qualität.

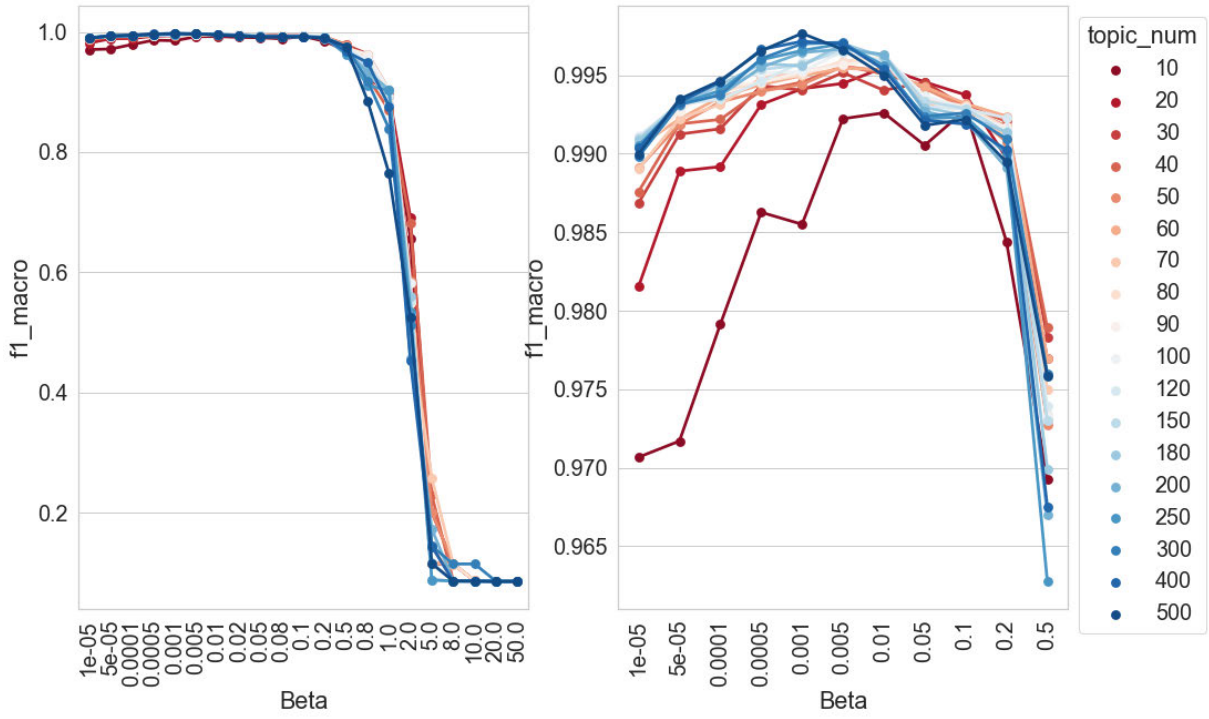

Abbildung 7.15 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Beta und Anzahl der Topics

Durch den Vergleich der obigen Ergebnisse mit den vorherigen Tests in Kapitel 6.4.1 (die Untersuchung auf dem Zeitungskorpus) kann man feststellen, dass Beta unabhängig vom Korpus zwischen 0,001 und 0,01 eingestellt werden sollte. Bei Werten größer als 0,1 ergibt sich ein deutlich schlechteres Klassifikationsergebnis. Wenn beim Training des Modells die Anzahl der Topics groß ist, sollte der Wert von Beta klein eingestellt werden. Der vorgegebene Wert von Beta in MALLET z. B. ist $\boldsymbol{\beta}$ = 0,01, es handelt sich dabei aus dieser Perspektive um eine vernünftige Voreinstellung.

7.4.2 Topic-Kohärenz

In Abbildung 7.16 werden die Verteilungen der NPMI-Werte aller Topics im Verhältnis zu Beta visualisiert, während die Anzahl der Topics zuerst auf $\boldsymbol{T}$ = 80 und 150 eingestellt wird. Hier ist zu beobachten, dass der Unterschied zwischen den beiden Einstellungen der Anzahl der Topics gering ist. Wenn Beta von 1e-05 (0,00001) auf 0,02 erhöht wird, ist die Spannweite der NPMI-Werte-Verteilungen etwas größer, während sich der Median der Verteilungen minimal verändert. Ab $\boldsymbol{\beta}$ = 0,05 wird der Wertbereich der Verteilungen immer kleiner. Mit einer weiteren Erhöhung von Beta (bis auf einen Wert von 50) wird der Bereich der Verteilungen schrittweise auf Werte zwischen ca. 0,05 und -0,1 reduziert.

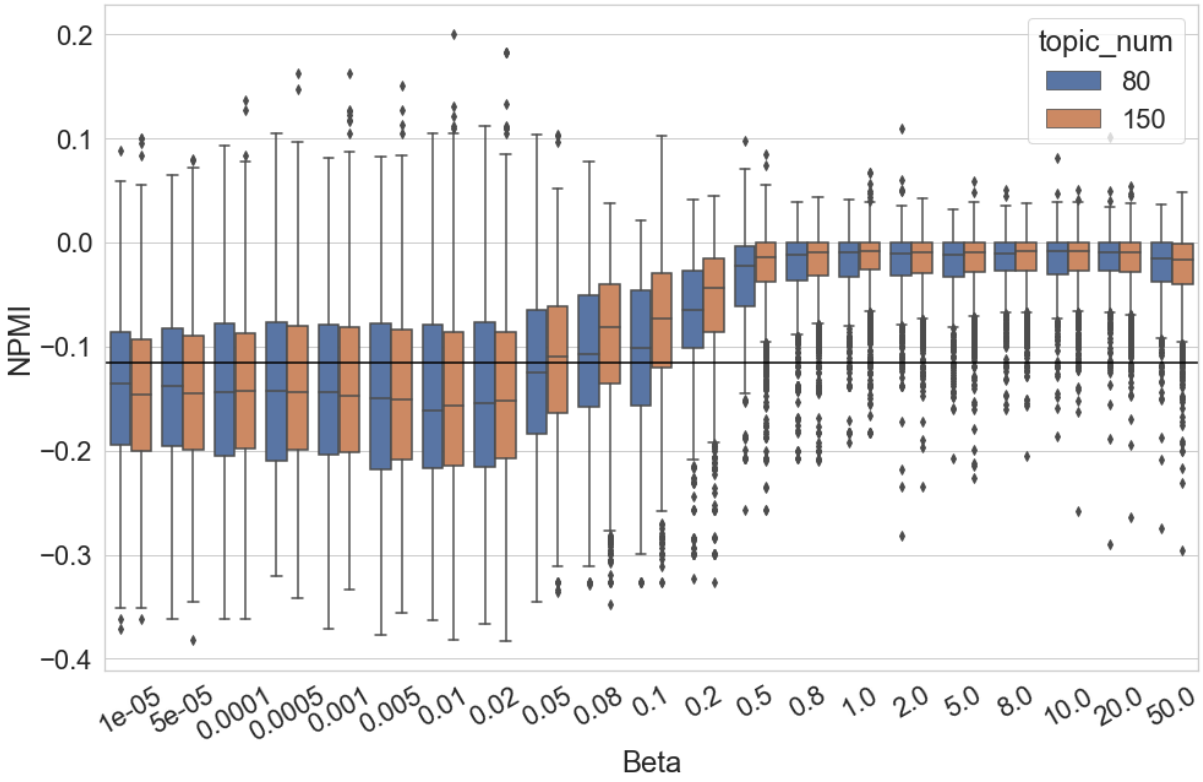

Abbildung 7.16 NPMI-Werte-Verteilung der Topics im Verhältnis zu Beta

Wenn die Anzahl der Topics stärker variiert wird, ändern sich die NPMI-Werte-Verteilungen ähnlich wie bei der vorherigen Situation. In Abbildung 7.17 wird deutlich, dass die Spannweite der NPMI-Werte-Verteilungen größer wird, wenn sich Beta von 1e-05 (0,00001) auf 0,02 erhöht. Die mittleren 50 % der NPMI-Werte sind etwas höher, wenn die Topic-Modelle mit weniger Topics trainiert werden (die roten Boxplots). Ab $\boldsymbol{\beta}$ = 0,02 beginnen sich die Verteilungen bei jedem Setting der Anzahl der Topics zu erhöhen und zu verengen. Unabhängig von der Anzahl der Topics liegen die meisten NPMI-Werte ab $\boldsymbol{\beta}$ = 5 zwischen ca. 0,05 und -

0,1. Das gleiche Phänomen – also, dass sich die NPMI-Werte-Verteilungen mit der Erhöhung von Beta verengen – wurde bereits in Kapitel 6.4.2 (die Untersuchung auf dem Zeitungskorpus) beobachtet. Der Parameter Beta ist der Hyperparameter der Topic-Wort-Verteilung; er beeinflusst, wie oft jedes Wort in einem Topic vorkommt. Deshalb wird in dieser Untersuchung auch überprüft, wie viele Wörter jedem Topic zugeordnet sind.

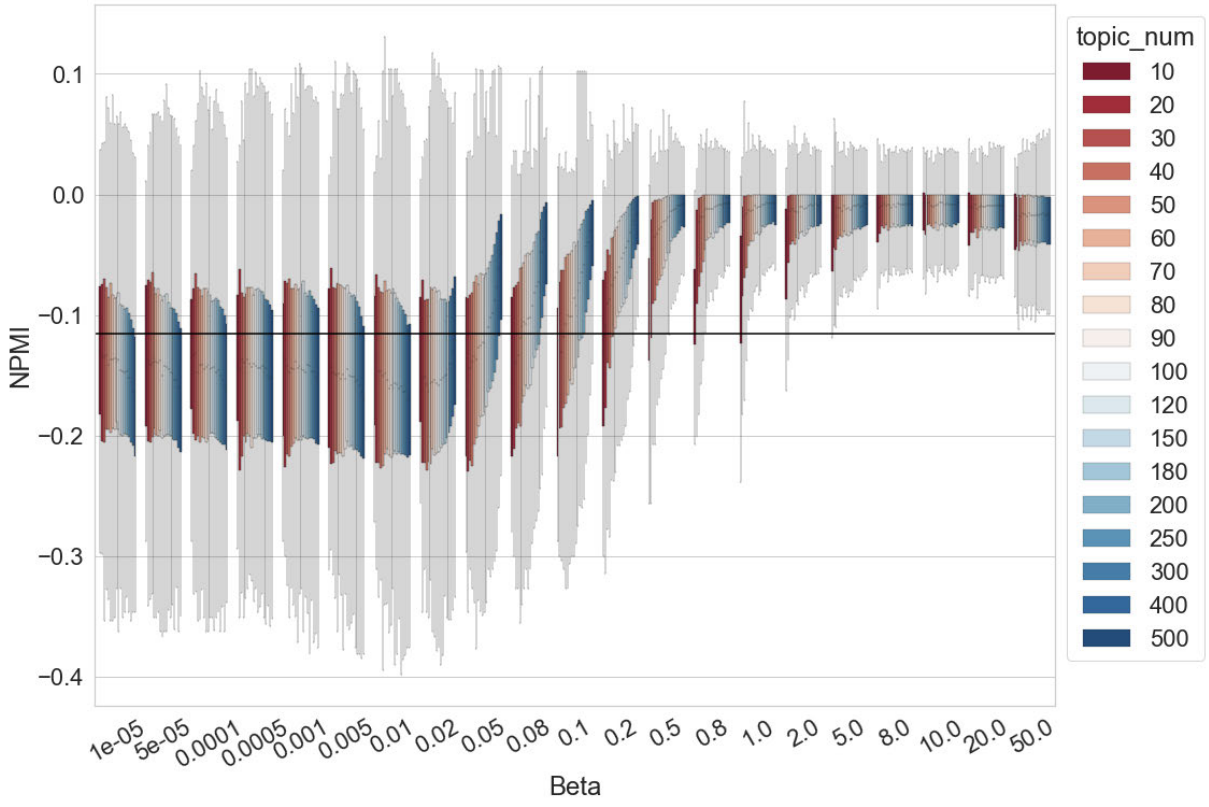

Abbildung 7.17 NPMI-Werte-Verteilung der Topics im Verhältnis zu Beta und Anzahl der Topics

In Abbildung 7.18 wird die Anzahl der jedem Topic zugeordneten Tokens visualisiert. Wenn Beta sehr klein eingestellt wird, variiert die Anzahl der Wörter in den Topics bei der gleichen Parameter-Setting-Kombination nicht sehr stark. Zum Beispiel enthalten alle Topics zwischen 5000 und 50.000 Tokens, wenn Beta auf 0,0001 und die Anzahl der Topics auf 500 eingestellt werden. Mit der Erhöhung von Beta ist der Unterschied unter den Topics immer größer, sie teilen sich allmählich in zwei Gruppen: Topics, die mehr als ein Million Tokens enthalten und solche, die weniger als 1000 Tokens umfassen. So existieren zum Beispiel in einem Topic-Modell ($\boldsymbol{T}$ = 80 und $\boldsymbol{\beta}$ = 50) ein Topic mit 3.573.120 Tokens, ein Topic mit 578 Tokens und 78 Topics mit lediglich 204 bis 282 Tokens. Eine ähnliche Situation ließ sich bereits in der

Untersuchung auf dem Zeitungskorpus beobachten und es ist festzustellen, dass ein solches Topic-Modell für die Exploration eines Korpus nicht sinnvoll ist.

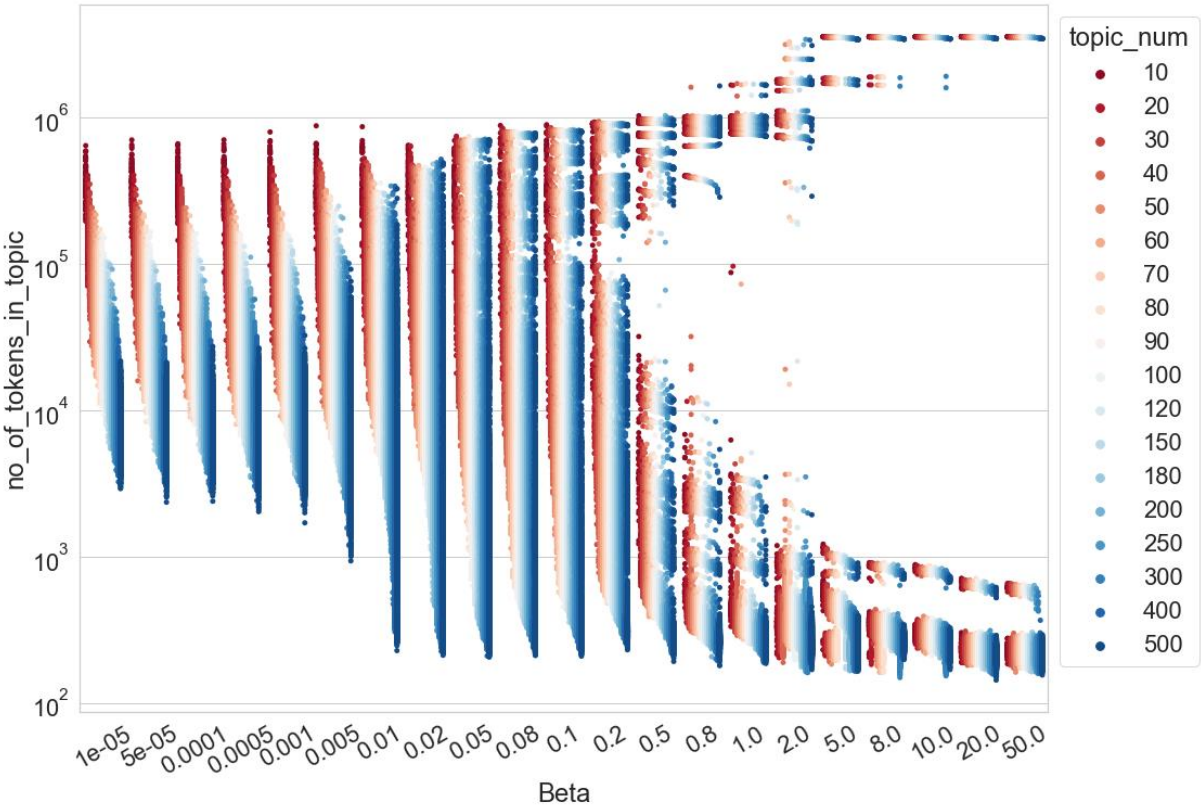

Abbildung 7.18 Anzahl der zugeordneten Tokens in jedem Topic im Verhältnis zu Beta und Anzahl der Topics (y-Achse logarithmisch skaliert)

Die oben genannte Beobachtung erklärt jedoch noch nicht, warum die NPMI-Werte meist zwischen 0,05 und -0,1 liegen, wenn Beta großer als 5 eingestellt wird. In Kapitel 3.2.6.3 wird beschrieben, dass ein NPMI-Wert trotzdem berechnet werden kann, auch wenn zwei Wörter bei der Berechnung des NPMI-Wertes im Referenzkorpus nicht gemeinsam vorkommen. Folgende Situation wird deshalb für eine weitere Untersuchung simuliert: Gegeben sind die Wörter A und B, sie tauchen im Referenzkorpus niemals zusammen auf, also ***P(A, B)*** = 0. Die Häufigkeit von A und B im Referenzkorpus ist ***Freq_A*** ∈ {1, 11, 21, 31, … 971, 981, 991} und ***Freq_B***. ∈ {1, 11, 21, 31, … 971, 981, 991}. Das Referenzkorpus (die deutschsprachige Wikipedia) enthält 209.566.852 Wörter, deshalb ist ***P(A)*** = ***Freq_A*** / 209.566.852 und ***P(B)*** = ***Freq_B*** / 209.566.852. Für jede Kombination von ***Freq_A*** und ***Freq_B*** kann ein NPMI-Wert berechnet werden, als Ergebnis werden insgesamt 10.000 NPMI-Werte ermittelt. Die Verteilung der 10.000 Werte wird in Abbildung 7.19 visualisiert, die meisten von ihnen liegen

in einem ähnlichen Wertbereich wie in Abbildung 7.17 bei einem Beta größer als 5. Nach der manuellen Überprüfung der Topic-Wörter ist festzustellen, dass es keine starke semantische Verbindung zwischen den wichtigsten Wörtern in den Topics mit 200 bis 300 Tokens gibt, obwohl sie durch Topic Modeling zu einem Topic zugeordnet werden. So sind zum Beispiel die ersten zehn Topic-Wörter eines Topics: „pfeiflaute, schräggelegt, footballmannschaft, schwerkraftprojektor, krebskorb, miesepetrigen, financier, pannendienst, flachdachgebäudes, ehrentag“. Es ist schwer vorstellbar, dass die Wörter „footballmannschaft“, „krebskorb“ und „ehrentag“ in irgendeinem Kontext gemeinsam in einem Text erscheinen. Das lässt die Vermutung zu, wenn Beta größer als 5 eingestellt wird, dass die meisten NPMI-Werte im Wertbereich zwischen 0,05 und -0,1 liegen, weil die Top-10 Topic-Wörter in den meisten Topics im Referenzkorpus nicht gemeinsam auftauchen.

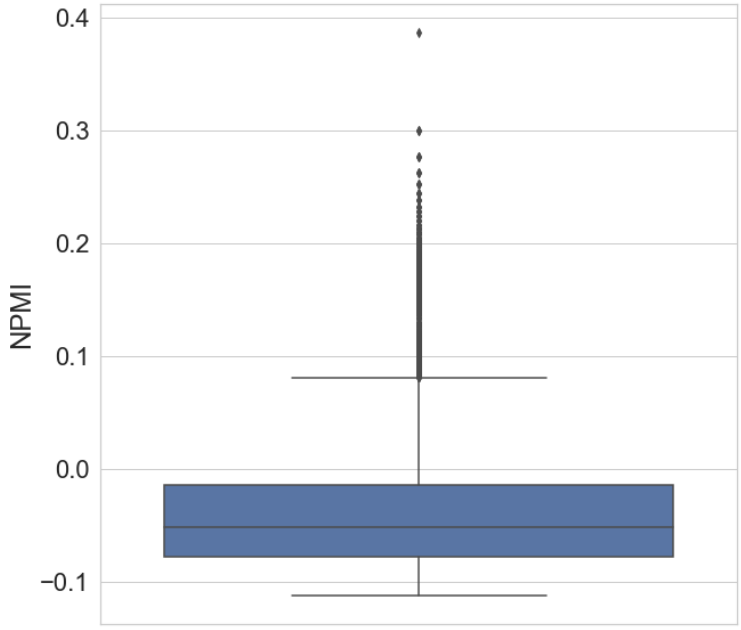

Abbildung 7.19 Verteilung der NPMI-Werte von zwei im Referenzkorpus nicht gemeinsam vorkommenden Wörtern

Um die Annahme zu testen, werden alle Topics aus den Topic-Modelle mit Beta größer als 5 überprüft. Das Ziel ist es herauszufinden, wie viele Wortpaare in den ersten zehn Wörtern jedes Topics in jedem Topic-Modell im Referenzkorpus nicht gemeinsam vorkommen. Mithilfe eines zusätzlichen Programms von Palmetto[76] kann die Kookkurrenz aller Wortpaare überprüft werden. So werden beispielsweise in einem Topic-Modell mit zehn Topics die ersten zehn

[76] Siehe: https://github.com/dice-group/Palmetto/issues/46, (26.02.2022).

Topic-Wörter in jedem Topic überprüft. Zehn Wörter in Paaren zu je zwei Wörtern ergeben 45 Wortpaare. Das Topic-Modell enthält deshalb 45 * 10 = 450 Wortpaare. Es kann angenommen werden, dass 90 Wortpaare existieren, die im Referenzkorpus nicht gemeinsam auftauchen; der Anteil beträgt dann 90 / 450 = 0,2. Nach Überprüfung aller Topic-Modelle werden die Ergebnisse in Abbildung 7.20 visualisiert. Für jedes Parametersetting werden zehn Topic-Modelle trainiert. Deshalb stellt jedes Boxplot in der Abbildung zehn Werte dar. In den meisten Fällen gibt es mehr als 75 % der Wortpaare, die im Referenzkorpus nicht zusammen vorkommen. Außerdem lässt sich beobachten, dass der Anteil mit der Erhöhung der Anzahl der Topics größer wird. Das Ergebnis entspricht der vorherigen Annahme. Zusammenfassend lässt sich feststellen, dass beim Training von Topic-Modellen der Wert von Beta nicht zu groß eingestellt werden sollte, weil dadurch schlechte Modelle trainiert werden. Außerdem kann dies dazu führen, dass die NPMI-Werte die Interpretierbarkeit der Topics nicht mehr richtig repräsentieren können, was deren Evaluation ruiniert.

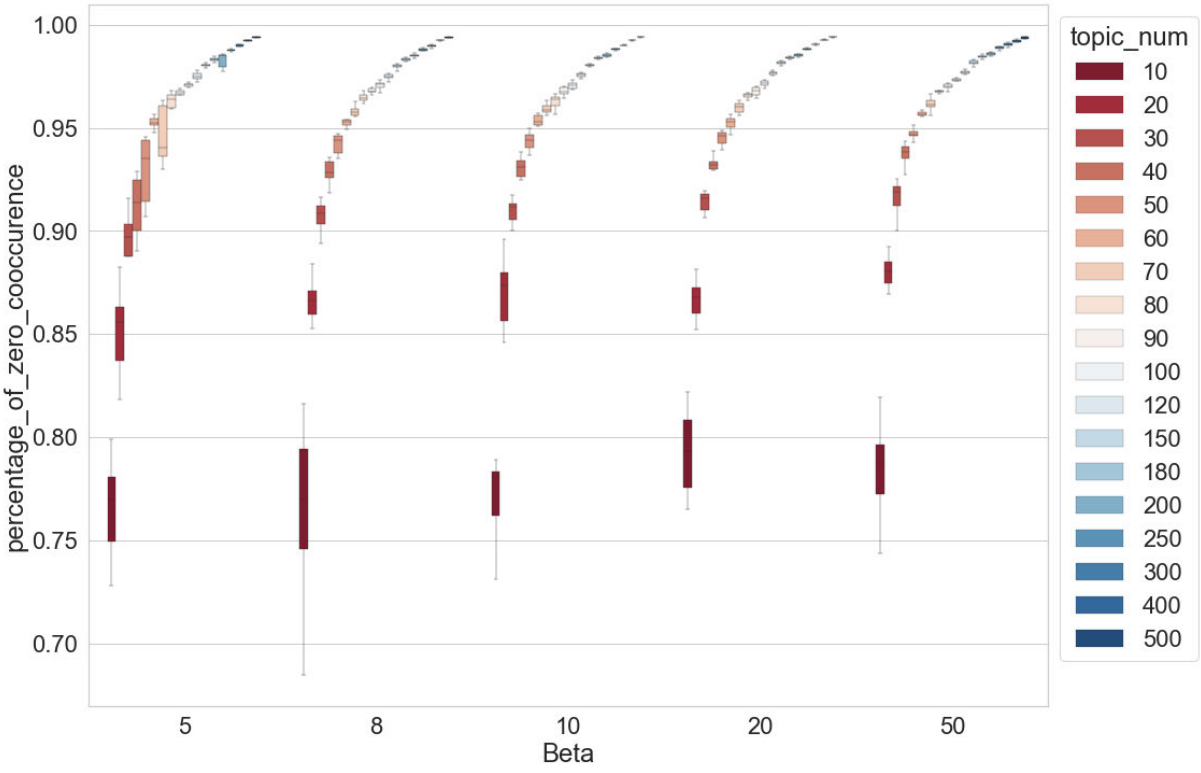

Abbildung 7.20 Anteil der Wortpaare, die in Wikipedia nicht gemeinsam vorkommen

7.5 Iteration des Gibbs-Samplings

In diesem Kapitel bezieht sich das Experiment auf den Einfluss der Iteration des Gibbs-Samplings ***I*** auf die Qualität des Topic-Modells. Das Setting von Anzahl der Iterationen ist ***I*** ∈ {20, 50, 100, 200, 300, 400, 500, 700, 1000, 1200, 1500, 1800, 2000, 2200, 2500, 2800, 3000}, die Anzahl der Topics wird wie zuvor auf 80 und 150 eingestellt.

7.5.1 Dokumentklassifikation

In Abbildung 7.21 werden die Klassifikationsergebnisse visualisiert. Wenn ***I*** von 20 auf 200 erhöht wird, steigen die F1-Werte von etwa 0,950 auf ca. 0,995. Ab ***I*** = 300 ist eine deutliche Verbesserung der Klassifikationsergebnisse trotz der Schwankungen nicht mehr zu beobachten, die F1-Werte bleiben auf ca. 0,995. Die Abweichung der F1-Werte zwischen den mit denselben Einstellungen trainierten Topic-Modelle ist besonders groß, wenn ***I*** auf 20 eingestellt wird. Das bedeutet, Topic-Modelle sollen mit mindestens 100 Iterationen trainiert werden, um den Einfluss der zufälligen Initialisierung bei der Zuweisung der Topics und beim Gibbs-Sampling zu vermindern. An einigen Stellen (z. B. ***I*** = 500 oder 2200) ist ersichtlich, dass die Klassifikation bei unterschiedlichen Einstellungen der Topic-Anzahl gleich gut funktioniert. Generell fallen die Klassifikationsergebnisse aber relativ besser aus, wenn die Anzahl der Topics auf 150 eingestellt wird.

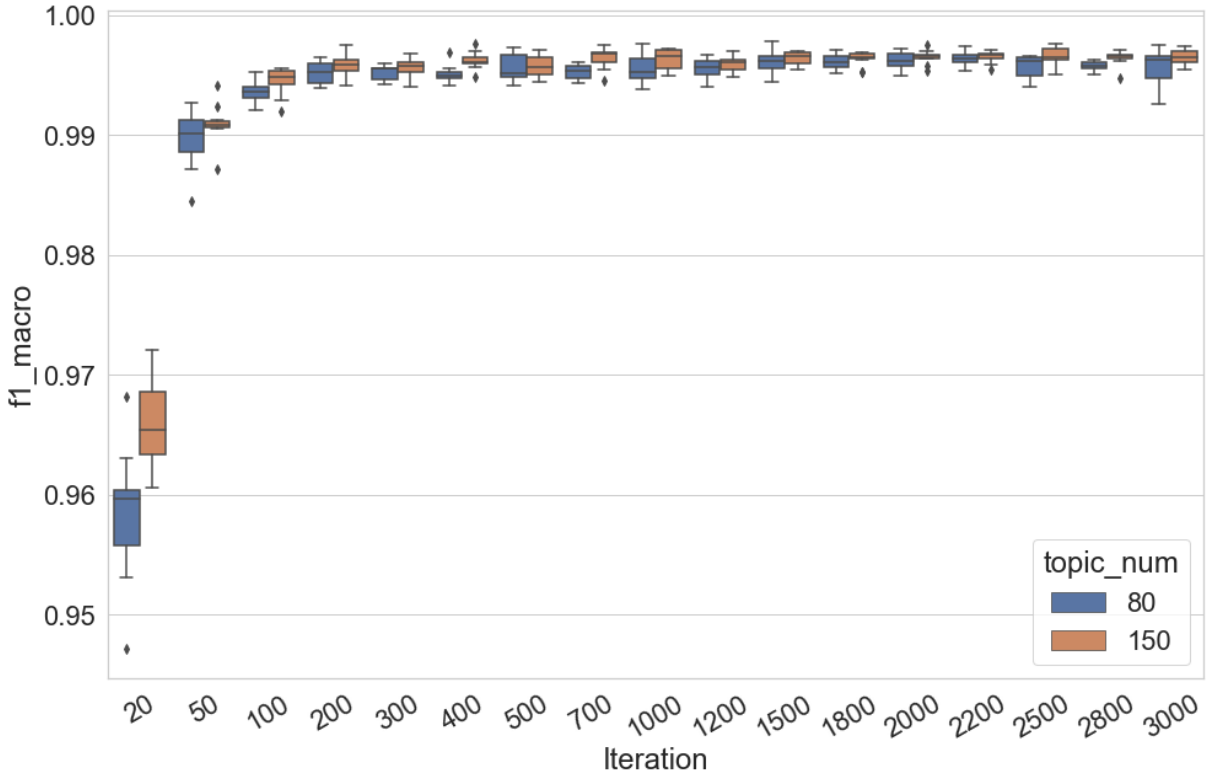

Abbildung 7.21 F1 (Makro)-Werte der Topic-Modell-basierten Dokumentklassifikation im Verhältnis zur Iteration des Gibbs-Samplings

Auch wenn die Anzahl der Topics stärker variiert wird, ist der aufsteigend Trend unabhängig von der Anzahl der Topics zu beobachten, wenn ***I*** schrittweise von 20 auf 300 erhöht wird (Abbildung 7.22). Auf der linken Seite wird das Gesamtergebnisse visualisiert, während die Veränderungen der durchschnittlichen F1-Werte ab ***I*** = 300 rechts nochmals dargestellt werden. Ab ***I*** = 300 kann keine systematische Verbesserung beobachtet werden. Neben den Topic-Modellen mit zehn Topics können alle anderen Modelle einen durchschnittlichen F1-Wert über 0,993 erzielen. In dieser Untersuchung wird auch beobachtet, dass die Topic-Modelle mit der höchsten Anzahl von Topics nicht das beste Klassifikationsergebnis garantieren. Im Vergleich dazu können die Topic-Modelle mit 100 bis 180 Topics höhere F1-Wert erzielen.

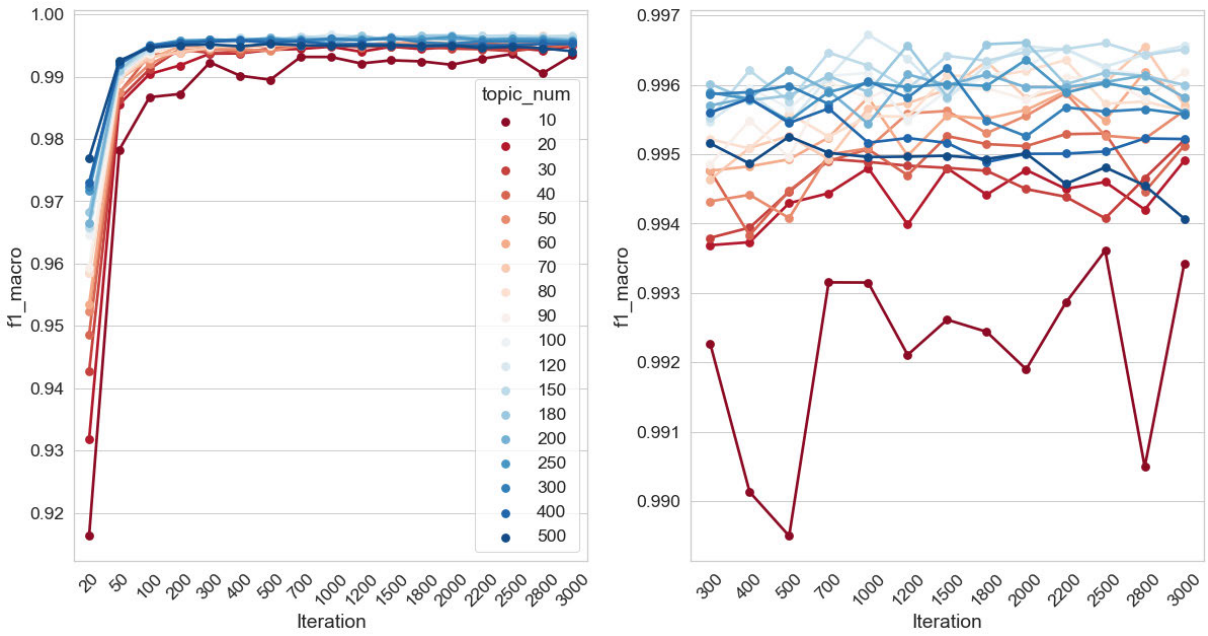

Abbildung 7.22 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Iteration des Gibbs-Samplings und Anzahl der Topics

Durch den Vergleich der obigen Ergebnisse mit den vorherigen Tests in Kapitel 6.5.1 (die Untersuchung auf dem Zeitungskorpus) lässt sich feststellen, dass dann, wenn ein Topic-Modell mit mehr als 300 Iterationen trainiert wird, die Klassifikation basierend auf diesem Modell bereits bestmögliche Ergebnisse erzielt. Die Voreinstellung der Iteration des Gibbs-Samplings

in MALLET (***I*** = 1000) ist aus diesem Grund sehr zuverlässig. Um Zeit beim Training des Modells zu sparen, lässt sich dieser Wert ggf. auch etwas niedriger einstellen.

7.5.2 Topic-Kohärenz

In Abbildung 7.23 werden die Verteilungen der NPMI-Werte aller Topics im Verhältnis zur Iteration des Gibbs-Samplings visualisiert, während die Anzahl der Topics zuerst auf ***T*** = 80 und 150 eingestellt wird. Das Ergebnis zeigt, dass sich die Spannweite der NPMI-Werte-Verteilungen vergrößert, wenn ***I*** von 20 auf 200 erhöht wird. Im Ergebnis wird also die Kohärenz der kohärentesten Topics durch die Erhöhung von ***I*** verbessert, während die Kohärenz der am wenigsten kohärenten Topics absinkt. Ab ***I*** = 300 kann keine deutliche Veränderung mehr beobachtet werden. Es führt auch zu keinen erheblichen Unterschieden, ob die Anzahl der Topics auf 80 oder 150 eingestellt wird. Insgesamt ist die Kohärenz von etwas mehr als 25 % der Topics besser als die NPMI-Baseline. Die NPMI-Werte-Verteilungen der besten 10 % der kohärentesten Topics werden in der Abbildung 7.24 nochmals dargestellt. Die Veränderung der Verteilungen ist leicht unterschiedlich zur Situation in Abbildung 7.23. Der aufsteigende Trend ist erst ab ***I*** = 400 nicht mehr deutlich zu erkennen.

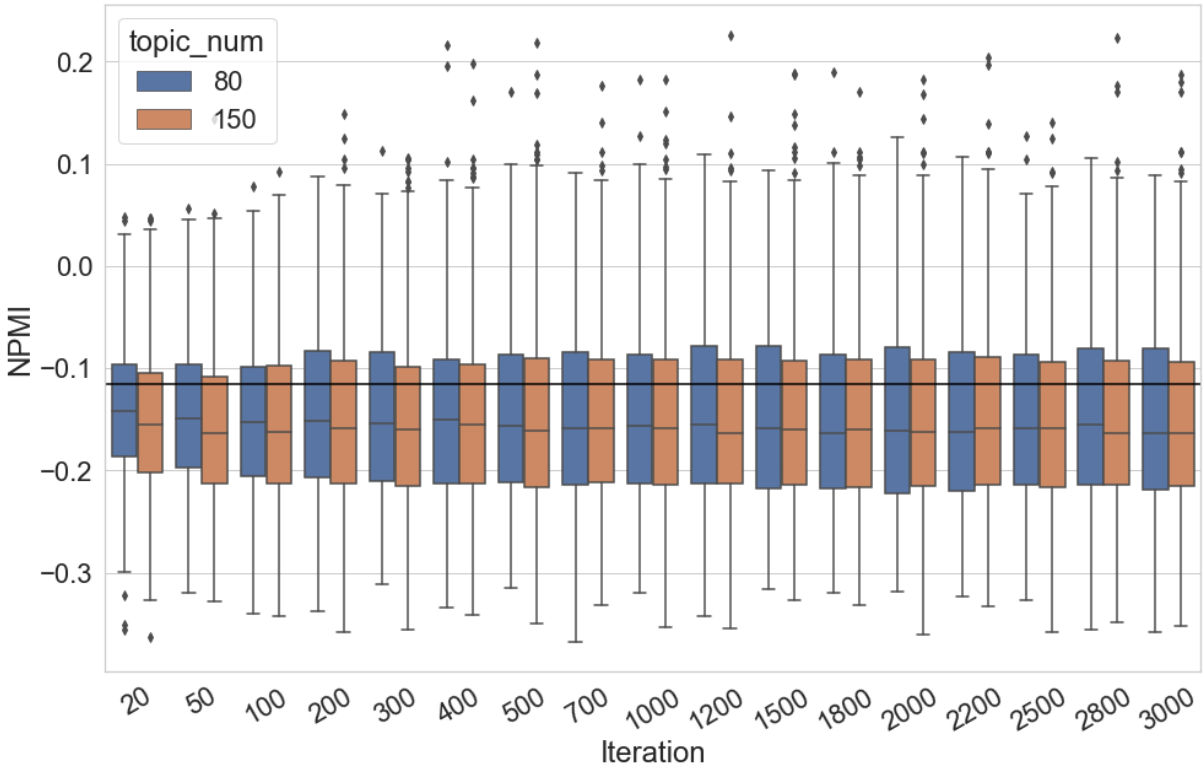

Abbildung 7.23 NPMI-Werte-Verteilung der Topics im Verhältnis zur Iteration des Gibbs-Samplings

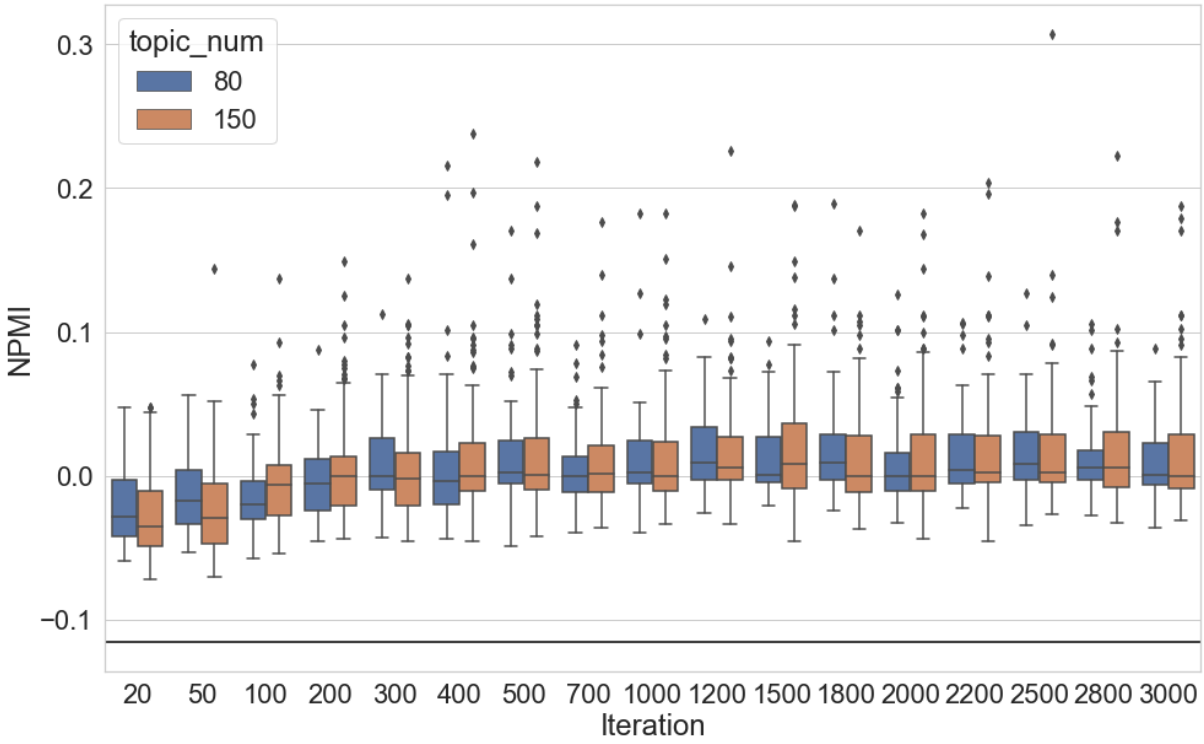

Abbildung 7.24 NPMI-Werte-Verteilung der Top 10 % kohärentesten Topics im Verhältnis zur Iteration des Gibbs Samplings

Auch wenn die Anzahl der Topics in einem höheren Maße variiert wird, lässt sich in Abbildung 7.25 beobachten, dass sich die Spannweite der NPMI-Werte-Verteilungen vergrößert, wenn ***I*** von 20 auf 400 erhöht wird. Die Verbesserung der Kohärenz der kohärentesten Topics lässt sich bis ***I*** = 1200 erkennen. Die NPMI-Werte eines kleinen Teils der Topics werden durch die Erhöhung von ***I*** auf über 0,1 gesteigert, wenn die Anzahl der Topics zwischen 80 und 150 eingestellt wird. Mit weiteren Erhöhungen von ***I*** ist keine deutliche systematische Verbesserung der Topic-Kohärenz beobachtbar. Im Vergleich dazu, unabhängig von der Einstellung von Anzahl der Topics, ändert sich die gesamte NPMI-Verteilung mit der Erhöhung von ***I*** nicht wesentlich. Die mittleren 50 % der NPMI-Werte liegen meist zwischen ca. -0,13 und -0,22, die Mediane der Verteilungen auf ca. -0,15. Das Ergebnis hier bestätigt auch, dass es keine gute Wahl ist, die Topic-Modelle mit zu vielen Topics zu trainieren, weil die NPMI-Werte von ca. 75 % der Topics bei jedem Setting von ***I*** niedriger als die NPMI-Baseline sind. Die NPMI-

Werte-Verteilung der Top 10 % kohärentesten Topics werden in Abbildung 7.26 nochmal dargestellt. Hier ist noch klarer zu erkennen, dass es sinnvoll ist, Topic-Modelle mit mindestens 1000 Iterationen zu trainieren, weil die kohärentesten Topics dadurch recht zuverlässig sein werden. Darüber hinaus ist bei fast jedem Setting von ***I*** zu sehen, das die Topics die höchsten NPMI-Werte aufweisen, wenn ihre Anzahl auf Werte zwischen 80 und 150 eingestellt wird.

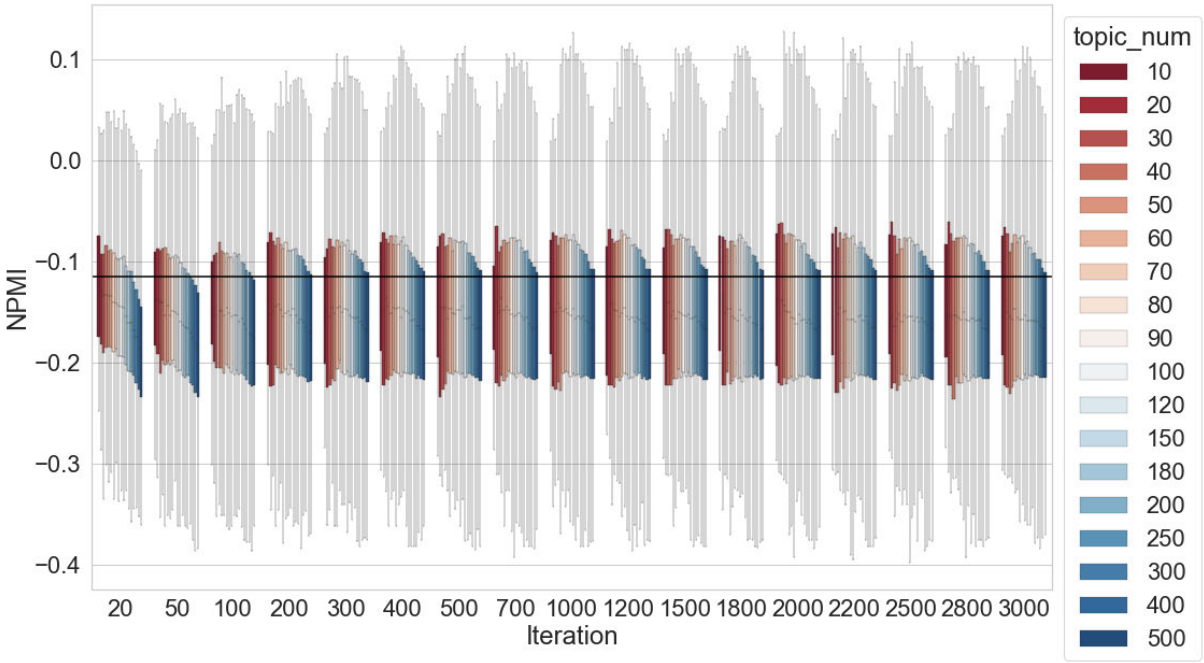

Abbildung 7.25 NPMI-Werte-Verteilung der Topics im Verhältnis zu Iteration des Gibbs-Samplings und Anzahl der Topics

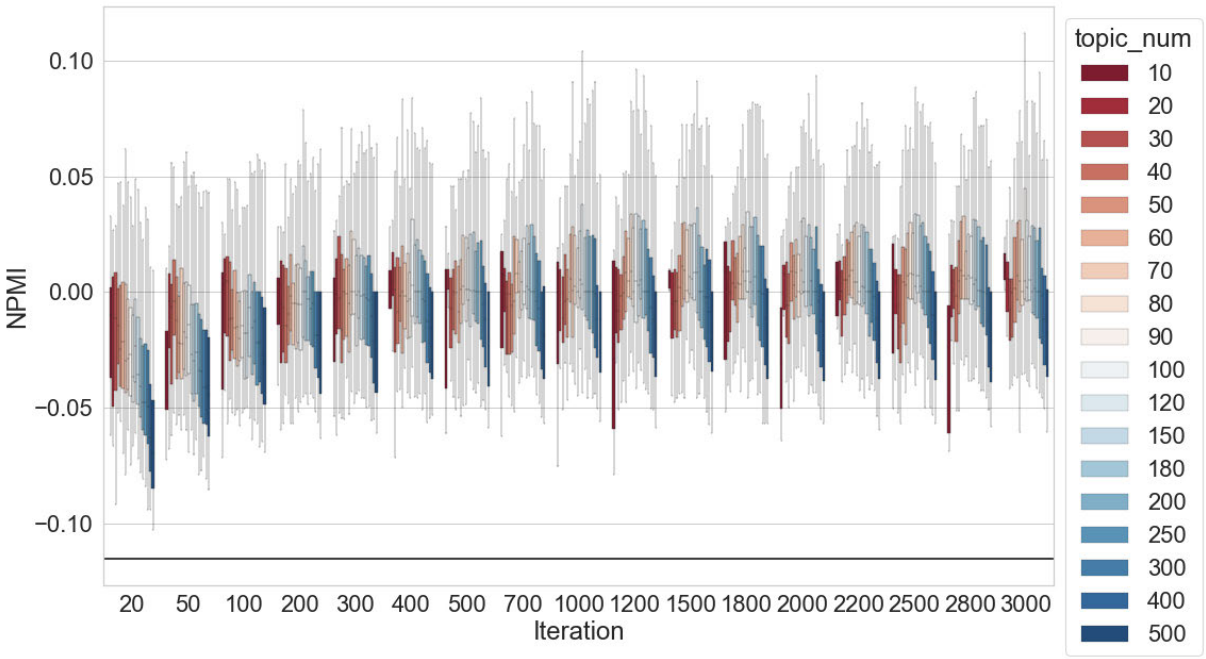

Abbildung 7.26 NPMI-Werte-Verteilung der Top 10 % der kohärentesten Topics im Verhältnis zu Iteration des Gibbs-Samplings und Anzahl der Topics

Durch den Vergleich der obigen Ergebnisse mit den vorhergegangenen Tests in Kapitel 6.5.2 (die Untersuchung auf dem Zeitungskorpus) lässt sich feststellen, dass es ausreichend ist, Topic-Modelle mit 500 Iterationen zu trainieren, um Zeit beim Training des Modells zu sparen, weil die gesamte NPMI-Verteilung sich ab ***I*** = 500 nicht wesentlich ändert. Allerdings ist es empfehlenswert, Topic-Modelle mit mindestens 1000 Iterationen zu trainieren, weil dadurch die kohärentesten Topics noch etwas weiter verbessert werden. Die Voreinstellung der Iteration des Gibbs-Sampling in MALLET (***I*** = 1000) ist aus diesem Grund sehr zuverlässig.

7.6 Chunk-Length

7.6.1 Chunking auf Paragraph-Ebene

In diesem Kapitel bezieht das Experiment sich auf den Einfluss der Chunk-Length des Dokuments auf die Qualität des Topic-Modells. Um den Einfluss der Chunk-Length zu testen, wird eine kleinste Längeneinheit des Dokuments ***C*** eingeführt. Die Romane werden beim Aufbau der Chunks zuerst in Paragraphen zerlegt. Wenn ein Paragraph P_n ***C*** oder mehr als ***C*** Tokens enthält, wird der Paragraph P_n als ein Chunk gespeichert. Wenn Paragraph P_n weniger als ***C*** Tokens enthält, wird der nächste Paragraph P_{n+1} zu P_n hinzugefügt, um einen Chunk

aufzubauen. Wenn die zwei Paragraphen insgesamt immer noch weniger als ***C*** Tokens enthalten, wird der nächste Paragraph P_{n+2} für den Aufbau eines Chunks hinzugefügt und so weiter. Danach werden Topic-Modelle auf die zerlegten Chunks trainiert.

Die Chunk-Length wird wie folgt eingestellt: $\boldsymbol{C} \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1800, 2000, 3000, 4000, 5000, 8000, 10000, 20000\}$. In Kapitel 5.1.2 wurde bereits erläutert, dass die durchschnittliche Textlänge des Korpus ca. 38.000 Tokens ist. Wenn man ***C*** höher als 19.000 einstellt, wird jeder Roman trotzdem nur in zwei Chunks zerlegt. Deshalb wird ***C*** höchstens auf 20.000 festgelegt. In Abbildung 7.27 wird die Veränderung der Anzahl der Dokumente und die durchschnittliche Chunk-Length mit der Erhöhung von ***C*** dargestellt. Die Anzahl der Dokumente (die grüne Linie) wird von mehr als 500.00 auf ca. 400 reduziert, während die durchschnittliche Dokumentlänge (die blaue Linie) von ca. 120 auf mehr als 16.000 Tokens zunimmt.

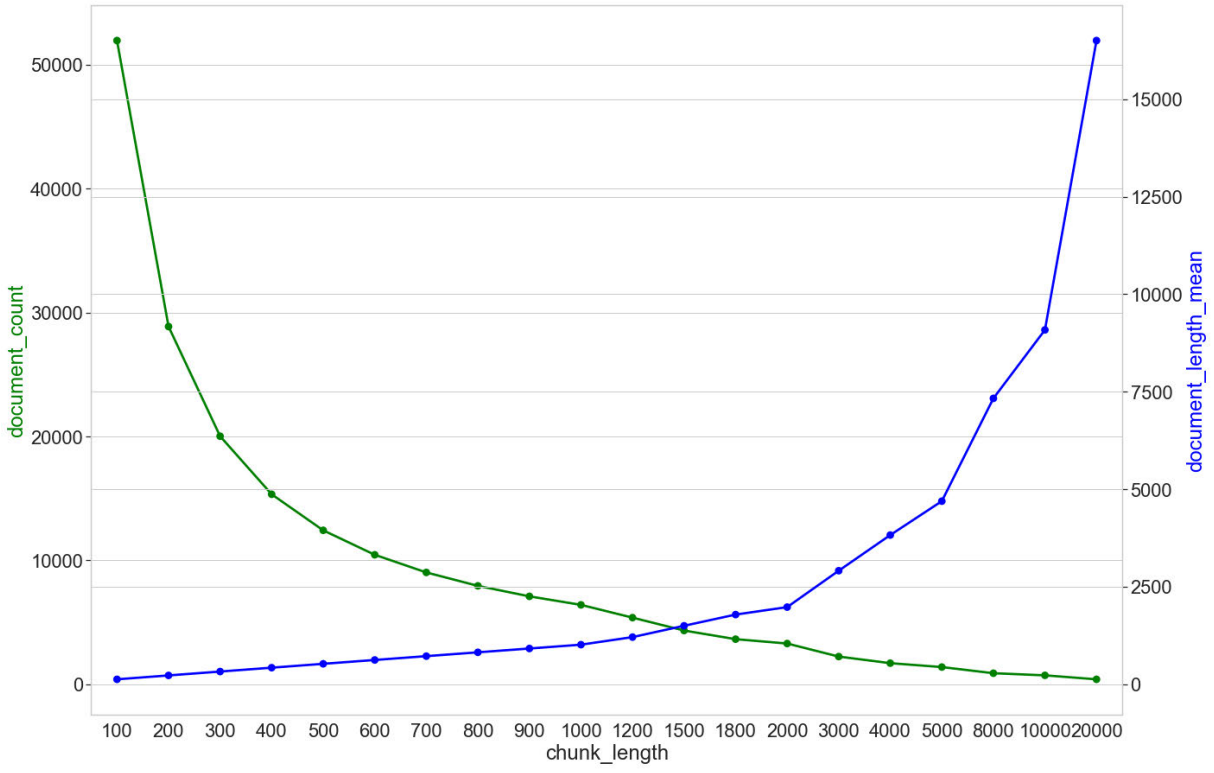

Abbildung 7.27 Anzahl der Dokumente (document_count) und die durchschnittliche Dokumentlänge (document_length_mean) im Verhältnis zur Chunk-Length ***C***

7.6.1.1 Dokumentklassifikation

Bevor die Untersuchungsergebnisse vorgestellt werden, ist es zunächst wichtig zu betonen, dass der Unterschied zwischen den Klassifikationsergebnissen in diesem Kapitel nicht immer den Unterschied der Qualität der Topic-Modelle repräsentieren kann. In den oben vorgestellten Untersuchungen bezieht sich die Klassifikationsaufgabe immer auf dieselbe Sammlung von Roman-Chunks, während die Chunk-Length auf 1800 eingestellt wird. Im Vergleich dazu sind die Klassifikationsaufgaben in diesem Kapitel wegen der Veränderung der Chunk-Length unterschiedlich. So sollen z. B. bei $\boldsymbol{C}$ = 100 die mehr als 50.000 Chunks in fünf Untergattungen klassifiziert werden, während bei $\boldsymbol{C}$ = 600 ca. 10.000 Chunks klassifiziert werden müssen. Daher ist der Vergleich dieser beiden Ergebnisse aus Sicht der Evaluation der Qualität der Topic-Modelle nicht sinnvoll. Die in den folgenden Abbildungen dargestellten Ergebnisse kann nur beschreiben, wie sich die Topic-Modeling-basierten Klassifikationsergebnisse ändern, wenn das gleiche Korpus unterschiedlich zerlegt wird.

In Kapitel 5.1.2 wurde eine Baseline der Klassifikation erstellt; der F1-Wert beträgt 0,993. Allerdings beziehen sich die Klassifikationen in diesem Kapitel nicht mehr auf dieselbe Chunks-Sammlung, sondern auf unterschiedliche Sammlungen mit verschiedener Chunk-Length. Aus diesem Grund wird die BoW-basierte Klassifikation auf den 20 segmentierten Korpora nochmals durchgeführt. Die Klassifikationen erfolgten wie zuvor als 10-fache Kreuzvalidierung mit linearer SVM. Die 20 F1 (Makro)-Werte werden in Abbildung 7.28 und Abbildung 7.29 durch eine grüne Linie visualisiert.

In Abbildung 7.28 werden die Klassifikationsergebnisse der zehn Einzeldurchläufe der jeweiligen Einstellungen der Chunk-Length durch Boxplots dargestellt, während die Anzahl der Topics auf 80 und 150 eingestellt wird. Der aufsteigende Trend der F1-Werte-Verteilung zeigt ein eindeutiges Ergebnis: Mit der Erhöhung von $\boldsymbol{C}$ erzielt die Klassifikation bessere Resultate. Wenn $\boldsymbol{C}$ größer als 10.000 ist, können die F1-Werte 1,0 erreichen. Interessant ist hier zu beobachten, dass die BoW-basierte Klassifikation bei $\boldsymbol{C}$ = 100 und $\boldsymbol{C}$ = 20.000 besser als die Topic-Modeling-basierte Klassifikation funktioniert. Allerdings kann dieses deutlich bessere Ergebnis erbringen, wenn $\boldsymbol{C}$ auf Werte zwischen 300 und 2000 eingestellt wird. Bei $\boldsymbol{C}$ = 5000 liefern z. B. die beiden Methoden ungefähr gleich gute Resultate. Der Unterschied zwischen den zwei Klassifikationsmethoden ist allerdings sehr gering (ca. 0,01 bis 0,04). Die Differenz der Klassifikationsergebnisse zwischen den Topic-Modellen mit 80 und 150 Topics erscheint hier nicht systematisch. Bei $\boldsymbol{C}$ = 200 oder 8000 erzielen die Topic-Modelle mit 80 Topics

bessere Klassifikationsergebnisse, während bei ***C*** = 700 oder 1000 die Topic-Modelle mit 150 Topics höhere F1-Werte erreichen können.

Wenn die Anzahl der Topics in einem größeren Umfang variiert wird, ist die Verbesserung der Klassifikationsergebnisse mit der Erhöhung von ***C*** klar zu beobachten (Abbildung 7.29). Wenn ***C*** zwischen 400 und 3000 liegt, kann die Topic-Modeling-basierte Klassifikation besser als die BoW-basierte Klassifikation funktionieren, wenn die Anzahl der Topics auf Werte größer als 10 eingestellt wird. An manchen Stellen, wie z. B. bei ***C*** = 1800 oder 2000, erzielen die Topic-Modelle mit zehn Topics ebenfalls bessere Ergebnisse als die BoW-basierte Klassifikation.

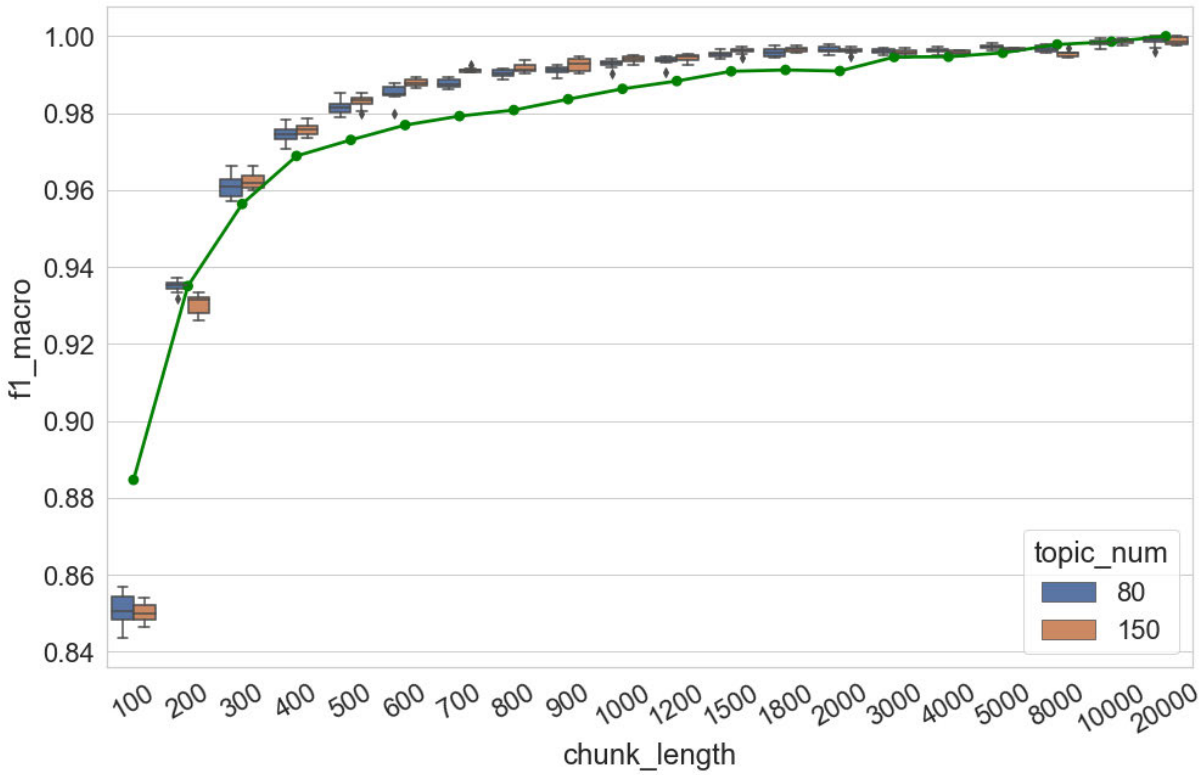

Abbildung 7.28 F1 (Makro)-Werte der Dokument-Klassifikation im Verhältnis zur Chunk-Length ***C***

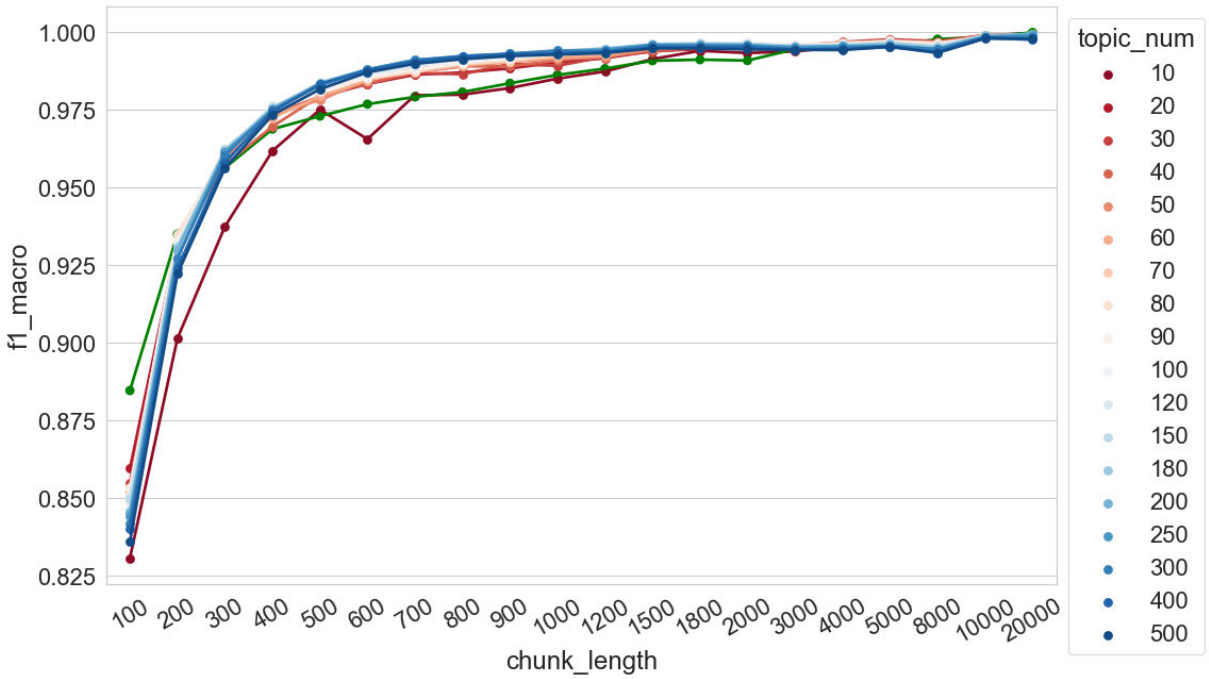

Abbildung 7.29 F1 (Makro)-Werte der Dokument-Klassifikation im Verhältnis zu Chunk-Length ***C*** und Anzahl der Topics

In den vorangegangenen Unterkapiteln wurde beobachtet, dass sich die Klassifikationsergebnisse nicht unbedingt mit der Erhöhung von ***T*** verbessern. Dieses Phänomen kann auch in Abbildung 7.29 beobachtet werden. So können die Topic-Modelle mit weniger Topics (die roten Linien) zum Beispiel dann, wenn ***C*** auf 5000 eingestellt wird, höhere F1-Werte erzielen. Da die Linien in dieser Visualisierung stark überlappend sind, werden die Daten in Abbildung 7.30 nochmal visualisiert, um eine bessere Ansicht der Klassifikationsergebnisse zu ermöglichen. Jede Linie stellt das Klassifikationsergebnis eines Settings der Chunk-Length dar. Die linke Seite der Abbildung zeigt die Änderung der F1-Werte mit der Erhöhung von Anzahl der Topics, während $\boldsymbol{C} \in \{100, 200, 300, 400, 500, 600\}$ eingestellt ist. Die rechte Hälfte der Abbildung dokumentiert die Änderung der F1-Werten bei der Erhöhung der Anzahl der Topics, während $\boldsymbol{C} \in \{700, 800, 900, 1000, 1200, 1500, 1800, 2000, 3000, 4000, 5000, 8000, 10000, 20000\}$ ausgewählt ist. In beiden Abbildungen ist zu beobachten, dass fast alle Kurven sich nach Erreichen des höchsten Punktes mit der Erhöhung der Anzahl der Topics etwas verschlechtern. Die Parametereinstellungen, die jede Kurve zu ihrem höchsten Punkt bringen, sind dabei unterschiedlich. Bei $\boldsymbol{C} = 5000$ und $\boldsymbol{C} = 8000$ z. B. wird der höchste Punkt erreicht, wenn die Anzahl der Topics auf 30 eingestellt wird. Im

Gegensatz dazu liefern die Topic-Modelle mit 300 Topics die beste Klassifikation, wenn die Chunk-Length auf 700 oder 800 eingestellt wird.

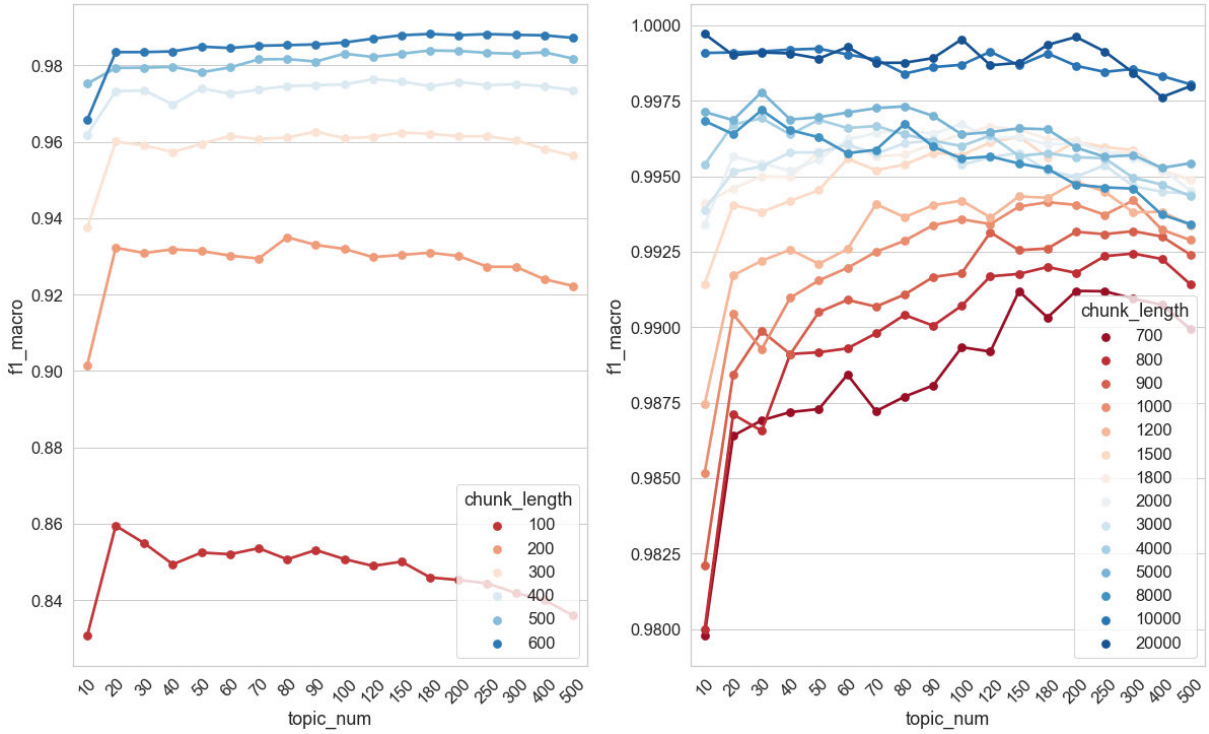

Abbildung 7.30 F1 (Makro)-Werte der Dokument-Klassifikation im Verhältnis zu Chunk-Length ***C*** und Anzahl der Topics

Durch den Vergleich der obigen Ergebnisse mit den vorherigen Tests in Kapitel 6.6.1.1 (die Untersuchung auf dem Zeitungskorpus) können folgende Beobachtungen zusammengefasst werden: Im Vergleich zum BoW-Modell kann das Topic-Modell auch für die Dokumentklassifikation eingesetzt werden, die Topic-Modeling-basiert Klassifikation kann sogar gelegentlich bessere Resultate erbringen. Allerdings ist es schwer, genau zu prognostizieren, welche Methode in welcher Situation besser ist. Die Untersuchung auf dem Zeitungskorpus zeigt, dass die Topic-Modeling-basierte Klassifikation bei jedem Setting der Chunk-Length besser funktioniert, wenn Topic-Modelle mit vielen Topics trainiert werden. Im Vergleich dazu zeigen die Untersuchungsergebnisse in Bezug auf das Romankorpus, dass Topic-Modeling-basierte Klassifikationen nur dann bessere Ergebnisse liefern, wenn die Chunk-Length auf einen bestimmten Wertbereich festgelegt wird. Vermutlich hängt das Ergebnis mit dem Untersuchungskorpus zusammen, allgemeine Regeln lassen sich hier daher eher nicht aufstellen. Außerdem ist es beim Training des Topics-Modells nicht empfehlenswert,

die Chunk-Length zu kurz einzustellen, da dies zu schlechteren Klassifikationsergebnissen führt.

7.6.1.2 Topic-Kohärenz

In Abbildung 7.31 wird die Änderung der NPMI-Verteilungen mit der Erhöhung der Chunk-Length visualisiert. Die Anzahl der Topics wird zuerst auf 80 und 150 eingestellt. Zunächst ist ein absinkender Trend der NPMI-Werte-Verteilungen bei einer Erhöhung der Chunk-Length zu beobachten. Bei ***C*** = 100 haben mehr als 50 % der Topics einen NPMI-Wert, der höher ist als der NPMI-Kontroll-Wert. Im Vergleich dazu liegen ca. 75 % der NPMI-Werte unterhalb des Kontroll-Wertes, wenn die Chunk-Length auf 20.000 eingestellt wird. Mit der Erhöhung der Chunk-Length sinken die Mediane der Verteilungen zudem, und zwar von ca. -0,1 allmählich auf etwa -0,16. Die Spannweite der Verteilungen ist oft (z. B: bei ***C*** = 100, 200, 300, 3000 oder 4000) breiter, wenn die Anzahl der Topics auf 150 eingestellt wird. Insgesamt aber ist kein großer Unterschied zwischen den Modellen mit 80 und 150 Topics zu beobachten. Insbesondere die Verteilungen der mittleren 50 % der NPMI-Werte sind fast immer identisch.

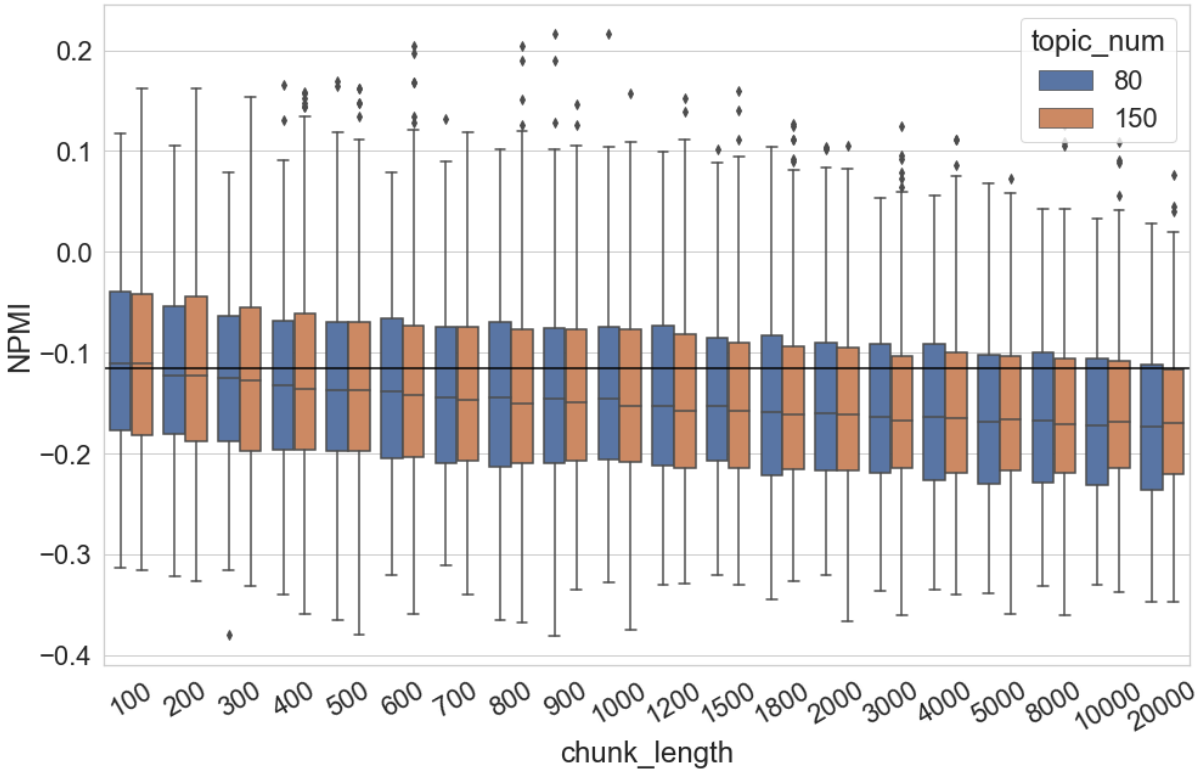

Abbildung 7.31 NPMI-Wert-Verteilung der Topics im Verhältnis zur Chunk-Length ***C***

Wenn die Anzahl der Topics stärker variiert wird, lässt sich dieser absinkende Trend auch in Abbildung 7.32 beobachten. Wenn die Chunk-Length von 100 auf 20.000 erhöht wird, sinkt der NPMI-Wert der kohärentesten Topics von ca. 0,17 auf etwa 0,05. Unabhängig von der Anzahl der Topics sinkt der Anteil der Topics, deren NPMI-Werte den NPMI-Kontroll-Wert überschreiten, mit der Erhöhung der Chunk-Length von 50 % auf etwa 26 % ab. Im Vergleich dazu zeigt das untere Quartil der meisten NPMI-Werte-Verteilungen bei der Erhöhung der Chunk-Length geringere Veränderungen, die Werte liegen hier meistens zwischen -0,2 und ca. -0,36. Wenn die Chunk-Length gleich eingestellt wird, lassen sich die höchsten NPMI-Werte zunächst mit der Erhöhung der Anzahl der Topics erreichen, anschließend fallen sie etwas ab. Darüber hinaus liegen die Verteilungen der mittleren 50 % der NPMI-Werte meist im gleichen Wertbereich, unabhängig von der Einstellung von Anzahl der Topics. Außerdem wird eine Erhöhung der NPMI-Werte ab ***C*** = 3000 beobachtet, wenn die Anzahl der Topics auf 500 eingestellt wird. Durch die Überprüfung der Topics wird festgestellt, dass es sich nicht um Verbesserung der Topics handelt, sondern um viele Topic-Wörter, die im Referenzkorpus nicht zusammen vorkommen (Siehe Kapitel 7.4.2 für eine ausführliche Erklärung).

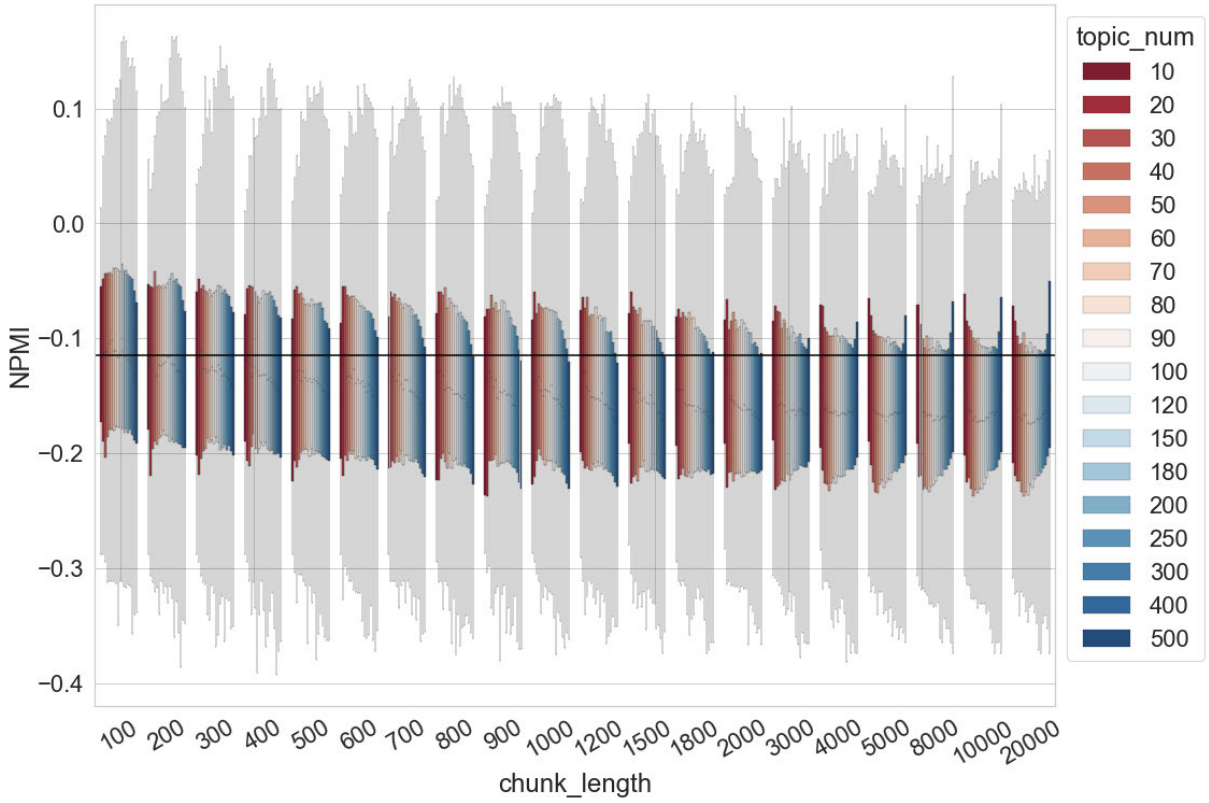

Abbildung 7.32 NPMI-Wert-Verteilung der Topics im Verhältnis zu Chunk-Length ***C*** und Anzahl der Topics

7.6.2 N-Token-Chunking

In den Untersuchungen in Kapitel 6.6 wird beobachtet, dass die trainierten Topic-Modelle wegen der eingesetzten verschiedenen Chunking-Strategien unterschiedlich sind. In diesem Kapitel werden die Romane in N-Token-Chunks zerlegt und der Einfluss dieser Zerlegungsstrategie auf die Qualität des Topic-Modells wird untersucht. Wie in Kapitel 6.6.2 wird eine kleinste Längeneinheit des Chunks ***C_n*** eingeführt. Jeder Roman im Untersuchungskorpus wird in Chunks zerlegt, jedes Chunk enthält genau ***C_n*** Tokens.

Die Chunk-Length wird wie folgt eingestellt: ***C_n*** ∈ {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1800, 2000, 3000, 4000, 5000, 8000, 10000, 20000}. Wie im vorangegangenen Kapitel wird ***C_n*** auf höchstens 20.000 eingestellt. In Abbildung 7.33 werden die Veränderung der Anzahl der Dokumente und die durchschnittliche Chunk-Length mit der Erhöhung von ***C_n*** dargestellt. Die Anzahl der Dokumente (die grüne Linie) wird von mehr als 140.000 auf ca. 1000 reduziert, während die durchschnittliche Dokumentlänge (die blaue Linie) von ca. 100 Tokens auf etwas mehr als 15.000 Tokens zunimmt.

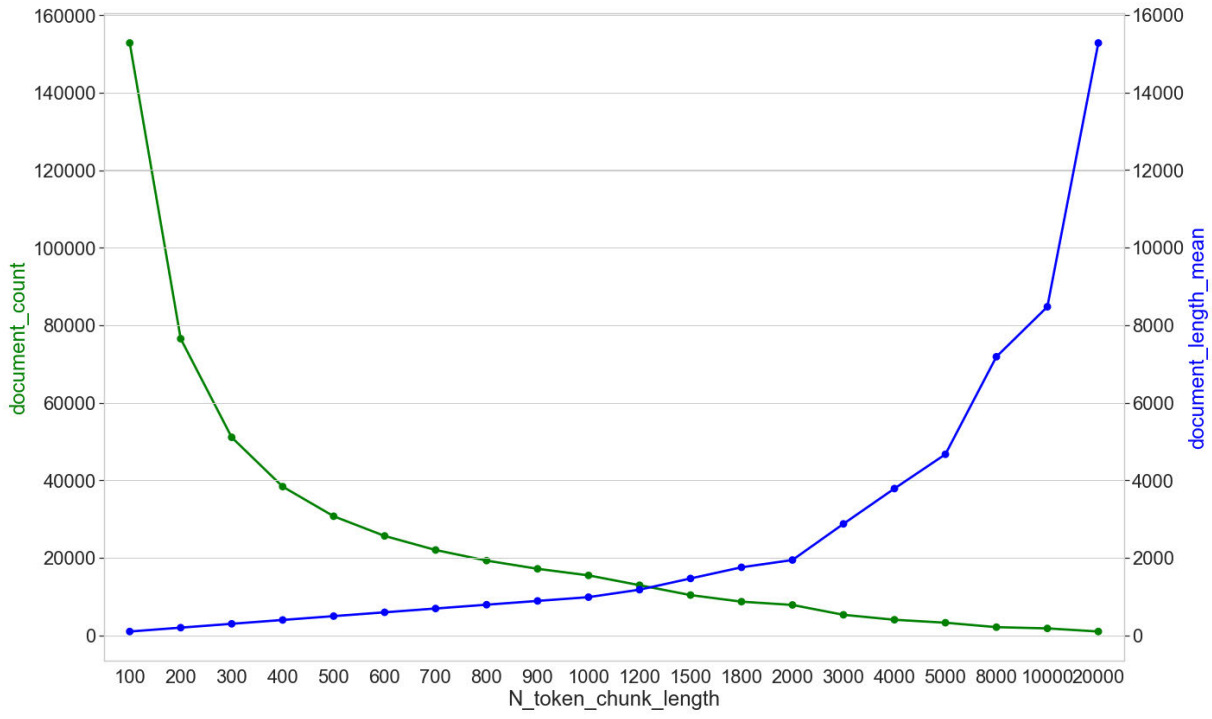

Abbildung 7.33 Anzahl der Dokumente (document_count) und die durchschnittliche Dokumentlänge (document_length_mean) im Verhältnis zur Chunk-Length ***C_n***

7.6.2.1 Dokumentklassifikation

In Abbildung 7.34 werden die Klassifikationsergebnisse der zehn Einzeldurchläufe der jeweiligen Einstellungen der Chunk-Length durch Boxplots dargestellt, während die Anzahl der Topics auf 80 und 150 eingestellt wird. Die grüne Linie stellt die Ergebnisse der BoW-basierten Klassifikation als Baseline dar. Auf den ersten Blick ist zu beobachten, dass sich die Klassifikationsergebnisse mit der Erhöhung von ***C_n*** verbessern. Während dieser aufsteigende Trend identisch mit den Ergebnissen der Untersuchung in Bezug auf die Paragraph-Chunks ist, zeigt der Vergleich zwischen den Topic-Modeling-basierten Klassifikationsergebnissen und den Baselines eine andere Situation. Die Topic-Modeling-basierte Klassifikation kann ab ***C_n*** = 500 zunehmend bessere Resultate erzielen. Die Klassifikationsergebnisse zwischen den Topic-Modellen mit 80 und 150 Topics weisen zudem keinen systematischen Unterschied auf. Bei ***C_n*** = 200 oder 10.000 können die Topic-Modelle mit 80 Topics beispielsweise bessere Klassifikationsergebnisse erzielen, während zwischen ***C_n*** = 500 und ***C_n*** = 1000 die Topic-Modelle mit 150 Topics höhere F1-Werte erreichen können.

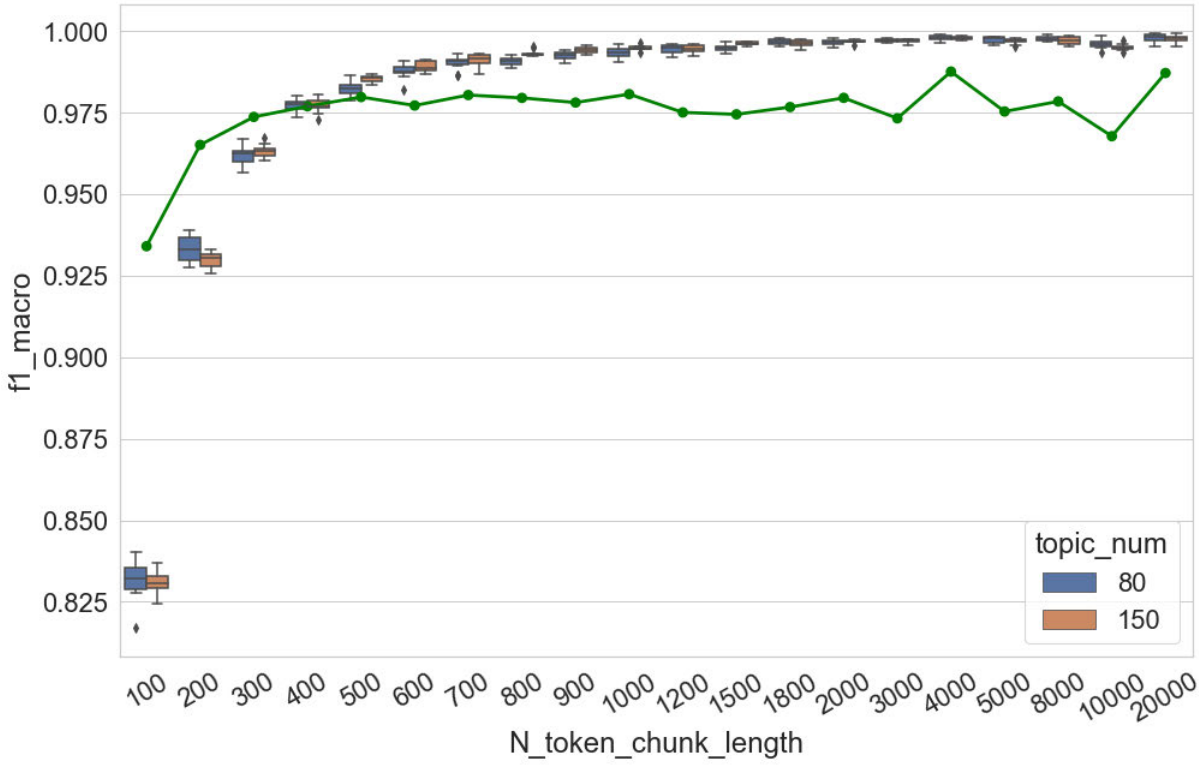

Abbildung 7.34 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation im Verhältnis zur Chunk-Length ***C_n***

Auch wenn die Anzahl der Topics deutlicher variiert wird, ist der aufsteigende Trend der F1-Werte mit der Erhöhung von ***C_n*** eindeutig zu beobachten (Abbildung 7.35). Im Vergleich zu anderen Settings der Topic-Anzahl erreichen die Modelle mit zehn Topics oft etwas schlechtere Klassifikationen. Selbst dann aber können die Modelle mit zehn Topics bessere Klassifikationsergebnisse liefern als die BoW-basierte Baseline, wenn ***C_n*** größer als 1000 eingestellt wird. Wenn der Wert von ***C_n*** größer als 500 ist, sind die meisten F1-Werte höher als die Baseline, und zwar unabhängig von der Anzahl der Topics. Darüber hinaus ist zu beobachten, dass die Modelle mit weniger Topics an mehreren Stellen (z. B. ***C_n*** = 10 oder ***C_n*** = 10.000) bessere Klassifikationsergebnis erzielen.

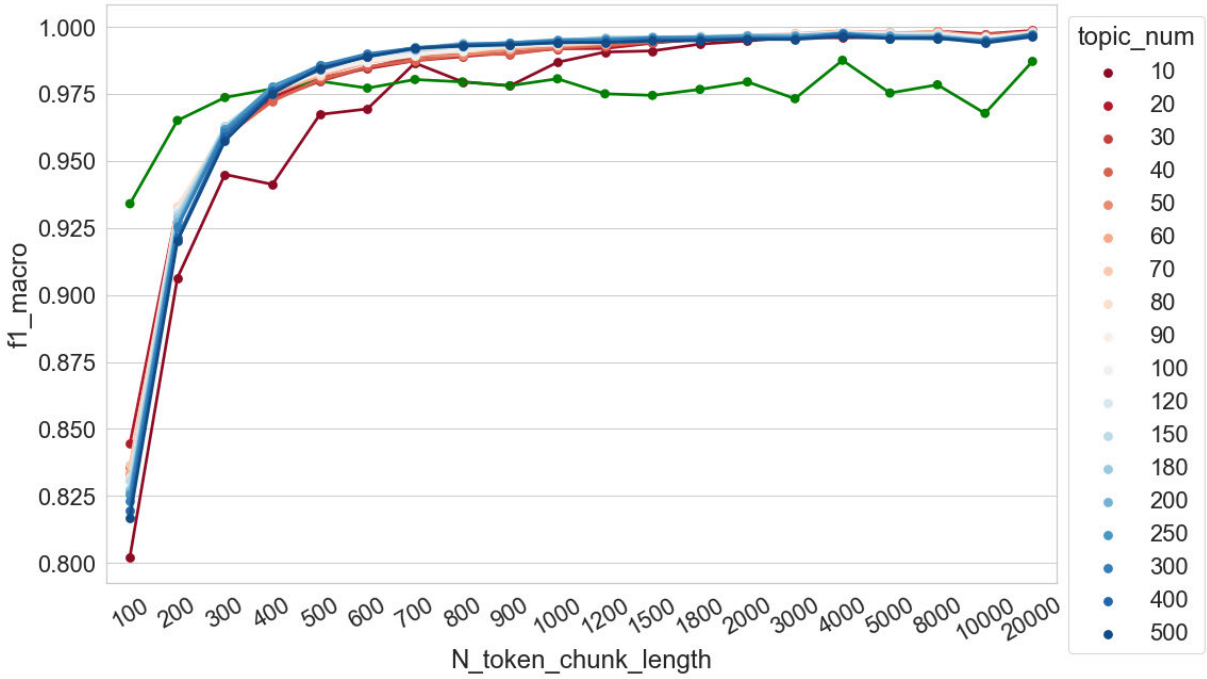

Abbildung 7.35 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Chunk-Length ***C_n*** und Anzahl der Topics

Um diese Befunde genauer zu präsentieren, werden die Ergebnisse in Abbildung 7.36 nochmals visualisiert. Auf der linken Seite der Darstellung sind die Änderungen der F1-Werte mit der Erhöhung der Anzahl der Topics zu sehen, während ***C_n*** ∈ {100, 200, 300, 400, 500, 600} beträgt. Auf der rechten Seite der Abbildung sind die Änderung der F1-Werte mit der Erhöhung der Anzahl der Topics dargestellt, während ***C_n*** ∈ {700, 800, 900, 1000, 1200, 1500, 1800, 2000, 3000, 4000, 5000, 8000, 10000, 20000} beträgt. Die Kurven zeigen zunächst eine

steigende und dann eine fallende Tendenz. Interessant ist, dass bei ***C_n*** = 100 und ***C_n*** größer als 8000 die höchsten F1-Werte erreicht werden, wenn die Anzahl der Topics auf 20 eingestellt wird. Im Vergleich dazu müssen Topic-Modelle mit mehr Topics (150 bis 300) trainiert werden, um die besten Klassifikationsergebnisse zu erzielen, wenn ***C_n*** auf einen mittleren Wert (z. B. zwischen 500 und 1200) eingestellt wird.

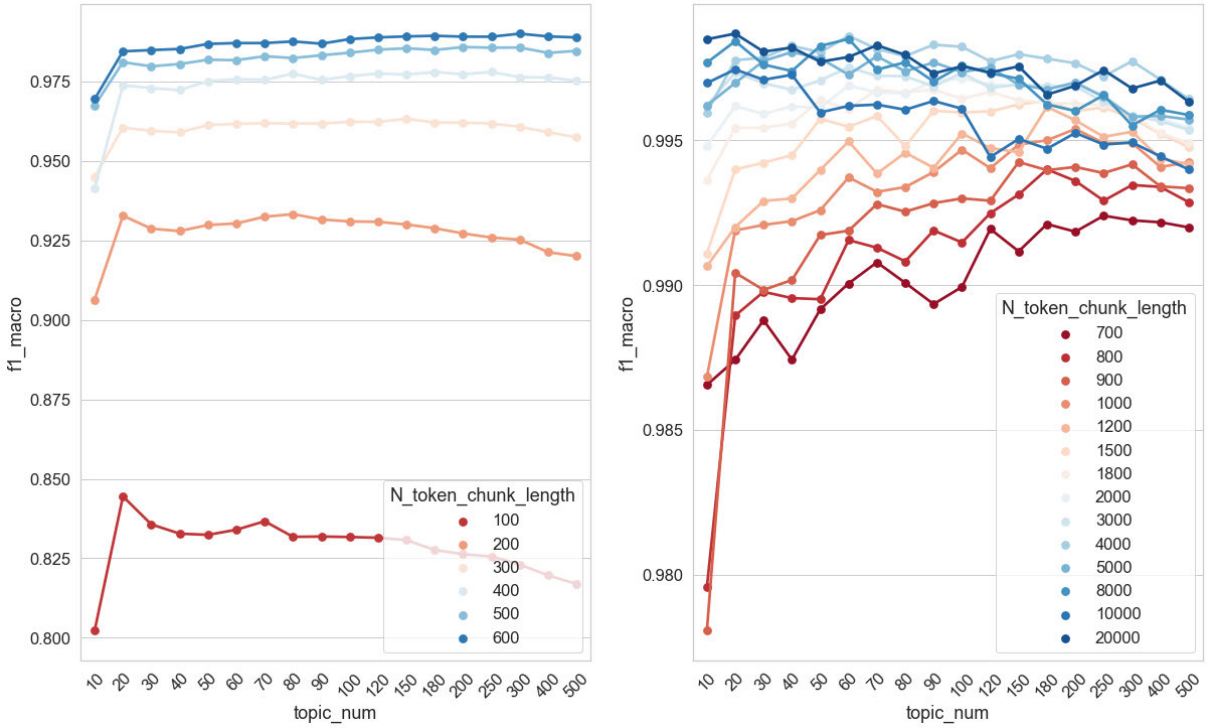

Abbildung 7.36 F1 (Makro)-Werte der Dokument-Klassifikation im Verhältnis zu Chunk-Length ***C_n*** und Anzahl der Topics

7.6.2.2 Topic-Kohärenz

In Abbildung 7.37 wird die Änderung der NPMI-Verteilungen bei der Erhöhung der Chunk-Length visualisiert, die Anzahl der Topics wird zuerst auf 80 und 150 eingestellt. Mit der Erhöhung von ***C_n*** zeigen das Maximum und der Median der NPMI-Werte-Verteilungen sowie die mittleren 50 % der NPMI-Werte einen absinkenden Trend, während das Minimum der NPMI-Werte-Verteilungen keine systematische Veränderung aufweist. Bei ***C_n*** = 100 ist der Median der beiden Verteilungen höher als den NPMI-Kontroll-Wert, während bei ***C_n*** = 20.000 etwas mehr als 25 % der NPMI-Werte über dem Kontroll-Wert liegen. Es zeigen sich darüber hinaus nur geringe Unterschiede zwischen den Modellen mit 80 und 150 Topics. An einigen Stellen, z. B. bei ***C_n*** kleiner als 500, haben die NPMI-Werte-Verteilungen eine größere

Spannweite, wenn die Anzahl der Topics auf 150 eingestellt wird. Wenn ***C_n*** dagegen beispielsweise auf 900 festgelegt wird, sind die beiden Verteilungen fast identisch.

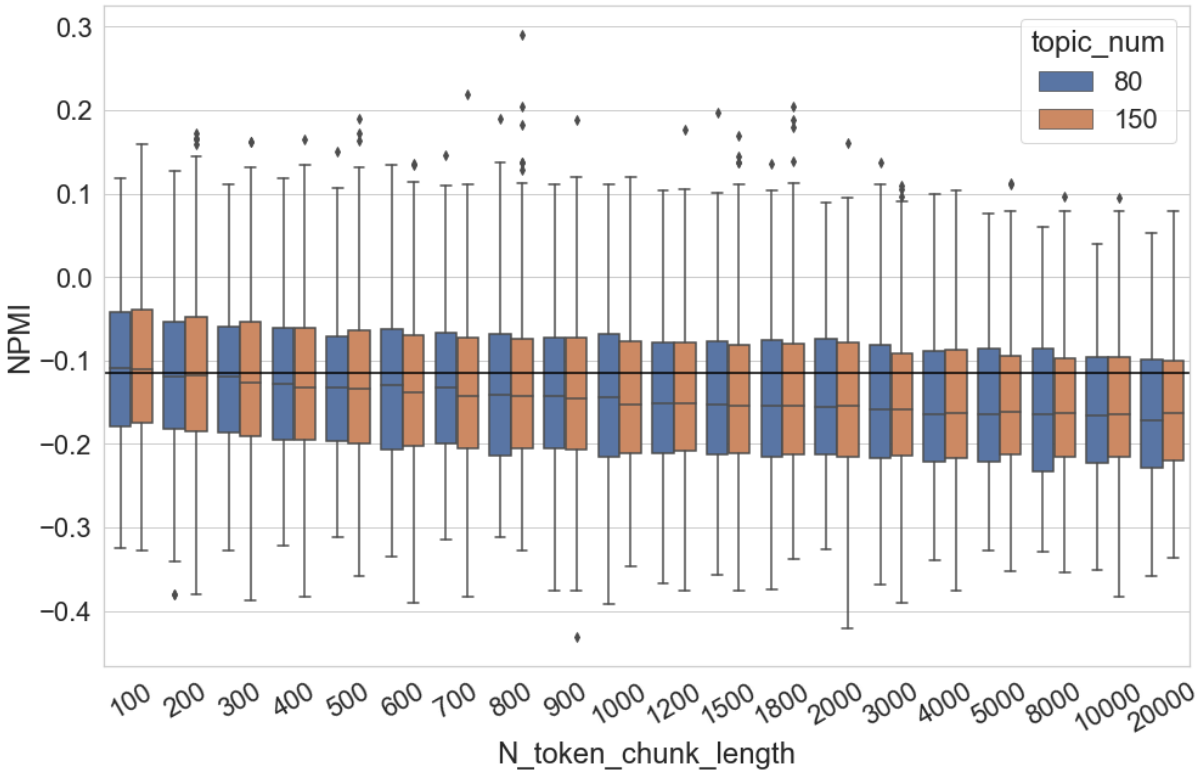

Abbildung 7.37 NPMI-Wert-Verteilung der Topics im Verhältnis zur Chunk-Length ***C_n***

Wenn die Anzahl der Topics deutlicher variiert wird, ist der absinkende Trend der Maximalwerte mit der Erhöhung von ***C_n*** auch in Abbildung 7.38 zu beobachten. Bei jedem Setting der Topic-Anzahl existieren mit der Erhöhung von ***C_n*** auch weniger Topics, deren NPMI-Werte den Kontroll-Wert überschreiten. Im Vergleich dazu zeigt das untere Quartil der meisten NPMI-Werte-Verteilungen geringere Veränderungen, es liegt zumeist zwischen -0,2 und ca. -0,38. Bei jedem Setting von ***C_n*** ist zudem zu beobachten, dass sich die Spannweite der Verteilungen mit der Erhöhung der Topic-Anzahl allmählich verbreitet. Der Maximalwert wird zunächst erreicht, anschließend fällt der Wert etwas ab. Die Verteilungen der mittleren 50 % der NPMI-Werte liegen bei jedem Setting von ***C_n*** in einem ähnlichen Wertbereich, unabhängig davon, wie die Anzahl der Topics eingestellt wird. Ähnlich wie in Abbildung 7.32 wird hier bei ***C_n*** größer als 3000 auch eine Erhöhung der NPMI-Werte beobachtet, wenn die Anzahl der Topics auf 500 eingestellt wird. Durch die Überprüfung der Topics wird festgestellt, dass es sich nicht um Verbesserung der Topics handelt, sondern um viele Topic-Wörter, die im

Referenzkorpus nicht zusammen vorkommen (Siehe Kapitel 7.4.2 für eine ausführliche Erklärung).

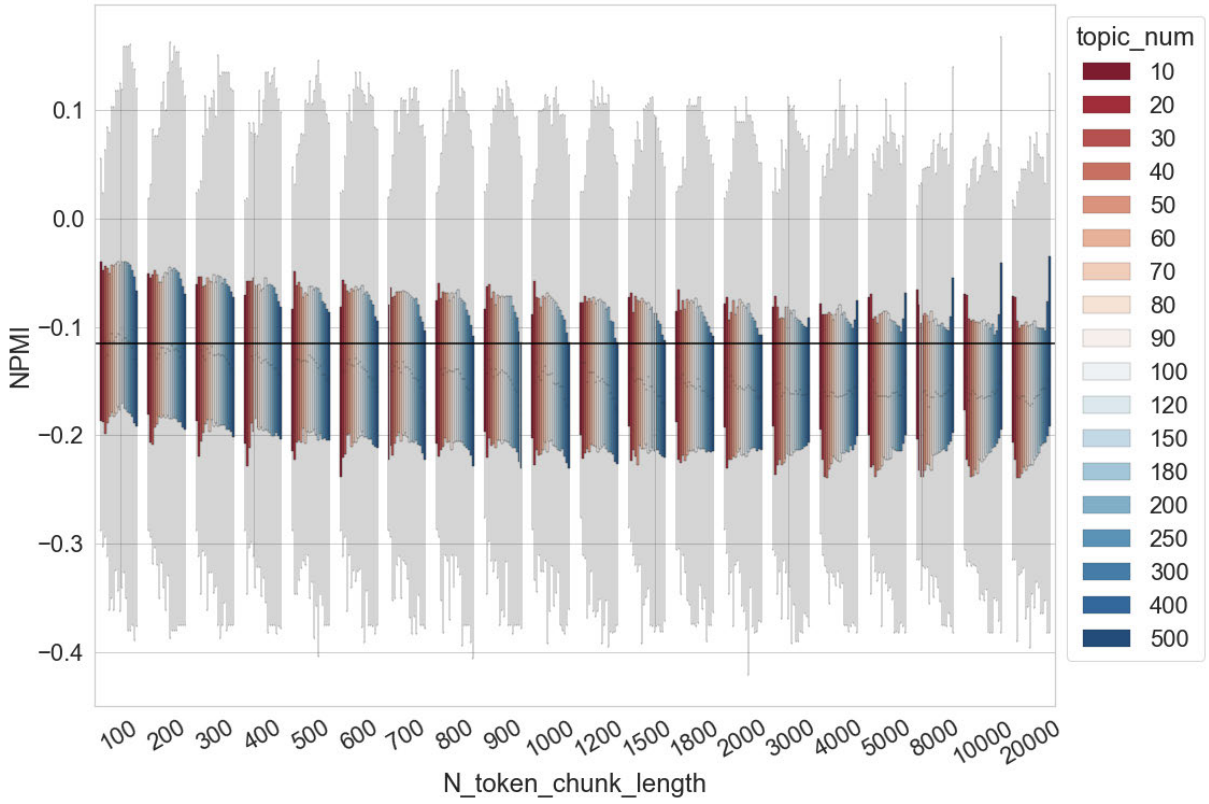

Abbildung 7.38 NPMI-Wert-Verteilung der Topics im Verhältnis zu Chunk-Length ***C_n*** und Anzahl der Topics

7.6.3 Chunking auf Satzebene

Wie in Kapitel 6.6.3 wird auch in diesem Kapitel untersucht, welchen Einfluss es auf die Qualität des Topic-Modells hat, wenn das Chunking des Textes auf der Satz-Ebene durchgeführt wird. Hier wird die kleinste Längeneinheit des Dokuments ***C_s*** verwendet. Jeder Roman wird zuerst in Sätzen zerlegt, jedes Chunk enthält mindestens ***C_s*** Tokens. Danach werden Topic-Modelle auf die zerlegten Korpora trainiert. Das Setting der kleinsten Längeneinheit ist ***C_s*** ∈ {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1800, 2000, 3000, 4000, 5000, 8000, 10000, 20000}. Je größer die kleinste Segmentlänge ***C_s*** ist, desto weniger Chunks enthält der zerlegte Korpus. In Abbildung 7.39 werden die Veränderung der Anzahl der Dokumente und die durchschnittliche Chunk-Length mit der Erhöhung von ***C_s*** dargestellt. Die Anzahl der Dokumente (die grüne Linie) sinkt von mehr als 200.000 auf 1630,

während die durchschnittliche Dokumentlänge (die blaue Linie) von etwa 110 Tokens auf mehr als 17.600 Tokens zunimmt.

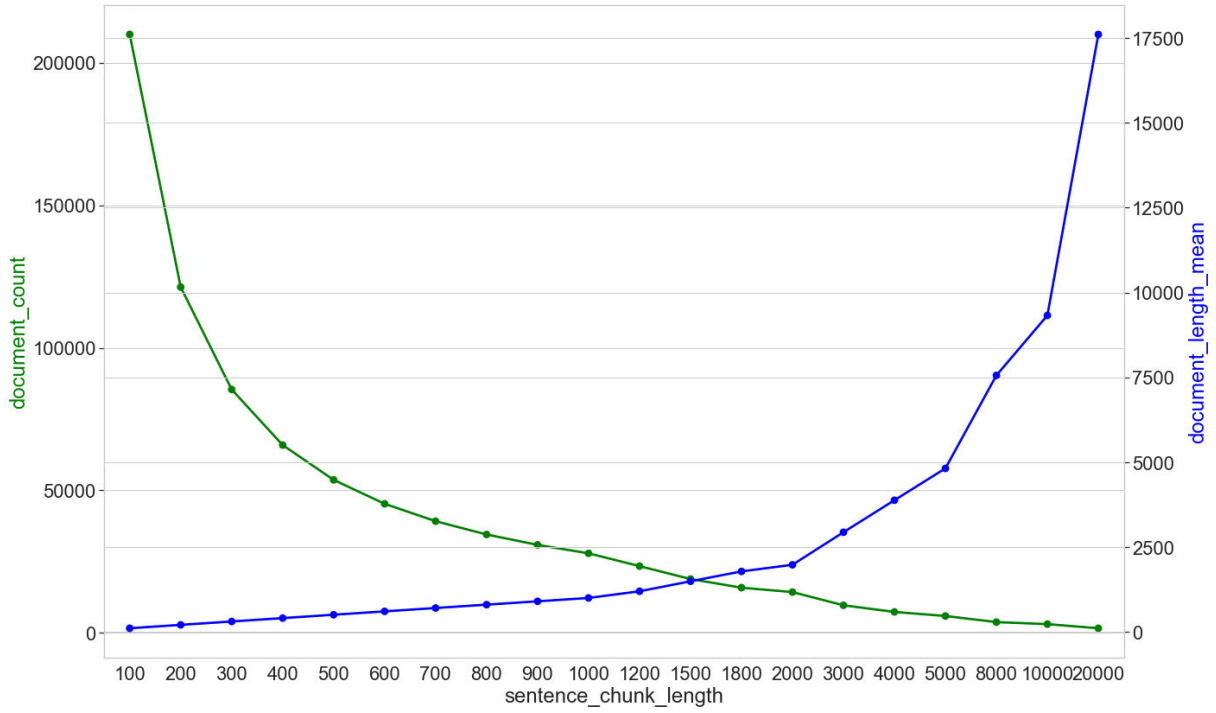

Abbildung 7.39 Anzahl der Dokumente (document_count) und die durchschnittliche Dokumentlänge (document_length_mean) im Verhältnis zur Chunk-Length ***C_s***

7.6.3.1 Dokumentklassifikation

In Abbildung 7.40 werden die Klassifikationsergebnisse mit der Veränderung der Chunk-Length durch Boxplots dargestellt, während die Anzahl der Topics zunächst auf 80 und 150 eingestellt wird. Die grüne Linie stellt die Ergebnisse der BoW-basierten Klassifikation als Baseline dar. Ein deutlicher Trend ist hier, dass die Klassifikation mit der Erhöhung von ***C_s*** zunehmend besser funktioniert. Die F1-Werte steigen von ca. 0,75 auf 1, ab ***C_s*** = 500 sind sie höher als 0,95. Außerdem ist zu beobachten, dass die Topic-Modelle mit 80 Topics die Dokumente deutlich besser klassifizieren, wenn ***C_s*** kleiner als 300 eingestellt wird. Dieser Unterschied ist bei anderen Einstellungen von ***C_s*** nicht zu beobachten. Der Vergleich mit der Baseline zeigt, dass die Topic-Modeling-basierte Klassifikation höhere F1-Werte erzielen kann, wenn ***C_s*** größer als 700 eingestellt wird.

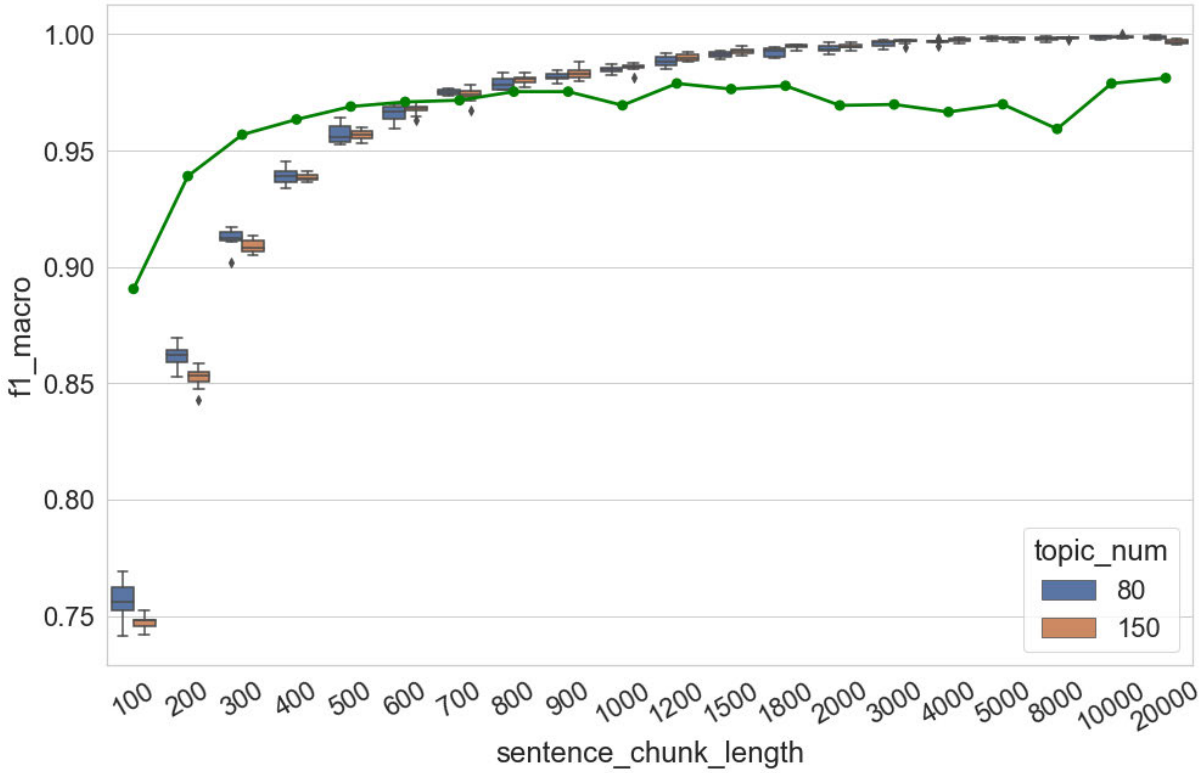

Abbildung 7.40 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation im Verhältnis zur Chunk-Length ***C_s***

Wenn die Anzahl der Topics in größerem Umfang variiert wird, ist in Abbildung 7.41 der aufsteigende Trend der F1-Werte mit der Erhöhung von ***C_s*** bei jedem Setting der Anzahl der Topics deutlich zu beobachten. Ab ***C_s*** = 1200 erbringt die Topic-Modeling-basierte Klassifikation bei jedem Setting der Anzahl der Topics bessere Ergebnisse als die BoW-basierte Methode. Ein interessantes Phänomen ist hier zudem, dass die Klassifikationsergebnisse etwas schlechter sind, wenn ***C_s*** kleiner als 800 und die Anzahl der Topics größer als 100 eingestellt wird. Im Vergleich dazu können Topic-Modelle mit mehr Topics (z. B. 400 oder 500 Topics) deutlich bessere Klassifikation erzielen, wenn ***C_s*** zwischen 800 und 8000 eingestellt wird. Der Unterschied zwischen den verschiedenen Einstellungen der Topic-Anzahl wird allerdings allmählich kleiner, bei ***C_s*** = 20000 ist er nicht mehr vorhanden.

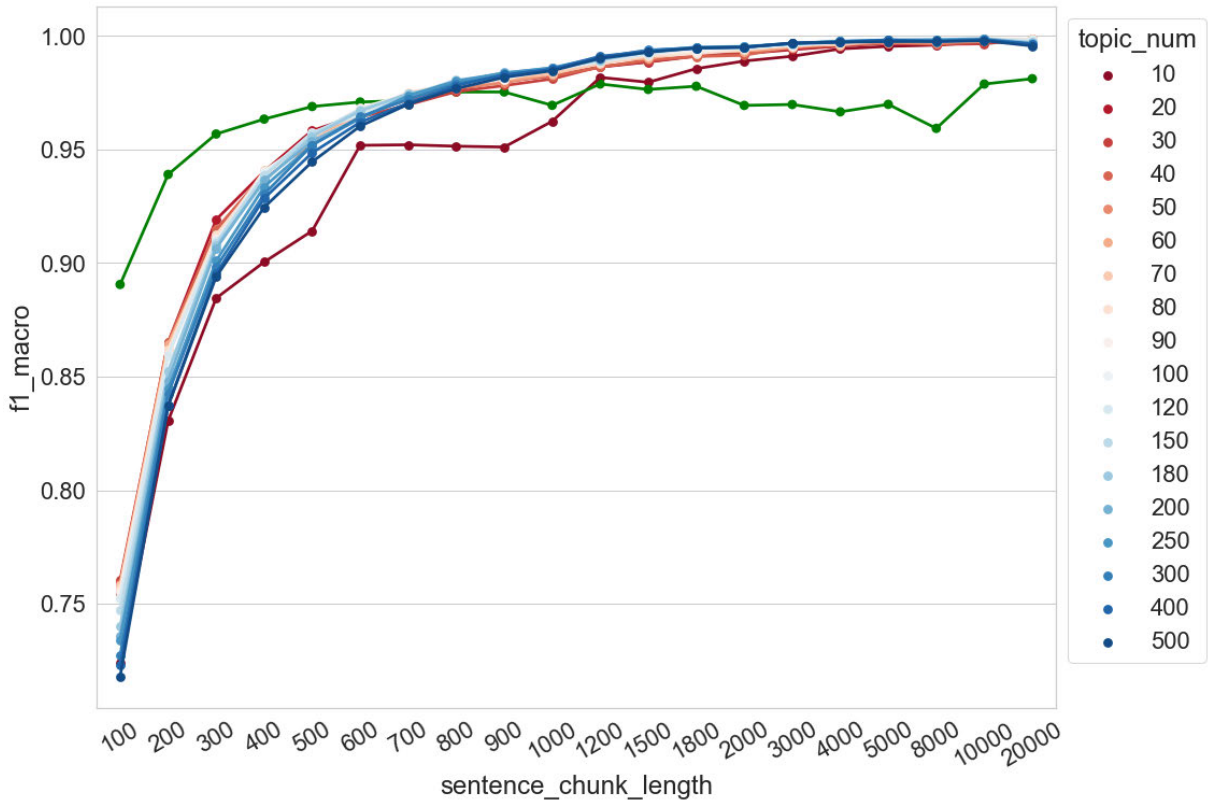

Abbildung 7.41 F1 (Makro)-Werte der Dokumentklassifikation im Verhältnis zu Chunk-Length ***C_s*** und Anzahl der Topics

Um die sich überlappenden Kurven in Abbildung 7.41 deutlicher darzustellen, werden die Ergebnisse in Abbildung 7.42 nochmal visualisiert. Auf der linken Seite des Diagramms werden die Änderungen der F1-Werte bei der Erhöhung der Anzahl der Topics dargestellt, während ***C_s*** auf Werte zwischen 100 und 800 eingestellt wird. Auf der rechten Seite werden die Veränderungen der F1-Werte bei der Erhöhung der Topic-Anzahl gezeigt, wenn ***C_s*** größer als 900 ist. Als allgemeiner Trend ist hier abzulesen, dass die Topic-Modelle mit zehn Topics bei jedem Setting von ***C_s*** das schlechteste Klassifikationsergebnis liefern. Bei ***C_s*** kleiner als 500 wird die beste Klassifikation durch Topic-Modelle mit 20 oder 30 Topics erzielt, die Resultate verschlechtern sich dann mit einer weiteren Erhöhung der Topic-Anzahl bis zu 500. Im Vergleich dazu ist der absinkende Trend bei anderen Settings von ***C_s*** weniger deutlich zu beobachten. Hier verringern sich die F1-Werte häufig nur leicht, wenn die Modelle mehr als 300 Topics enthalten.

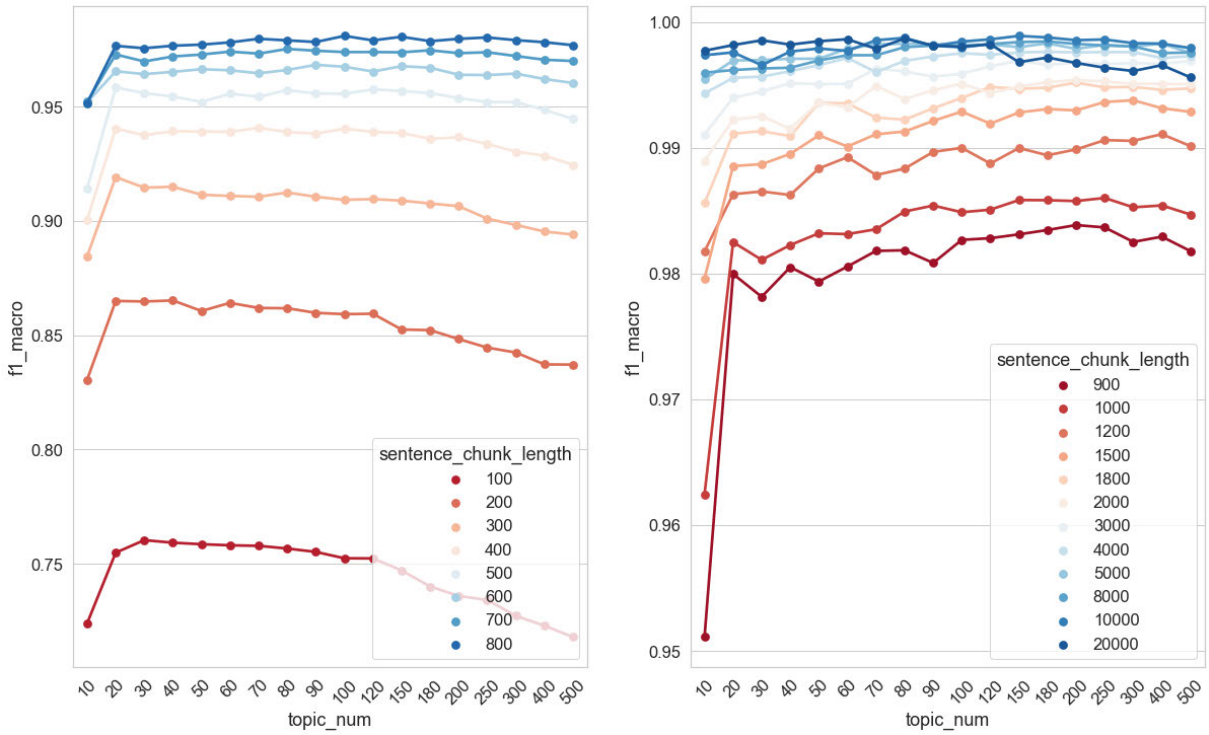

Abbildung 7.42 F1 (Makro)-Werte der Dokument-Klassifikation im Verhältnis zu Anzahl der Topics und Chunk-Length ***C_s***

Vergleicht man die Klassifikationsergebnisse der Topic-Modelle, die mit diesen drei verschiedenen Chunkingsstrategien trainiert wurden, so lässt sich feststellen, dass für sie alle gilt, dass die Klassifikation umso besser funktioniert, je größer die Chunk-Length ist. In Abbildung 7.43 ist zu beobachten, dass das Chunking auf Satzebene im Vergleich zu den beiden anderen Chunkingsstrategien zu einer etwas schlechteren Klassifikation führt, wenn die Chunk-Length kleiner als 2000 eingestellt ist. Hier beträgt die Anzahl der Topics 150. Die Ergebnisse bei den anderen Settings der Anzahl der Topics sind ähnlich und werden deshalb nicht visualisiert. Das Chunking auf Paragraph-Ebene und das Chunking auf N-Token-Ebene erbringen fast bei jedem Setting der Chunk-Length gleich gute Resultate. All diese Beobachtungen stimmen mit den Untersuchungsergebnissen auf dem Zeitungskorpus in Kapitel 5.6 überein. Offensichtlich ist das Chunking des Textes in zu kurze Sätze weniger dazu geeignet, die thematische Struktur einer Textsammlung gut zu modellieren und eine gute Topic-Modeling-basierte Klassifikation zu garantieren.

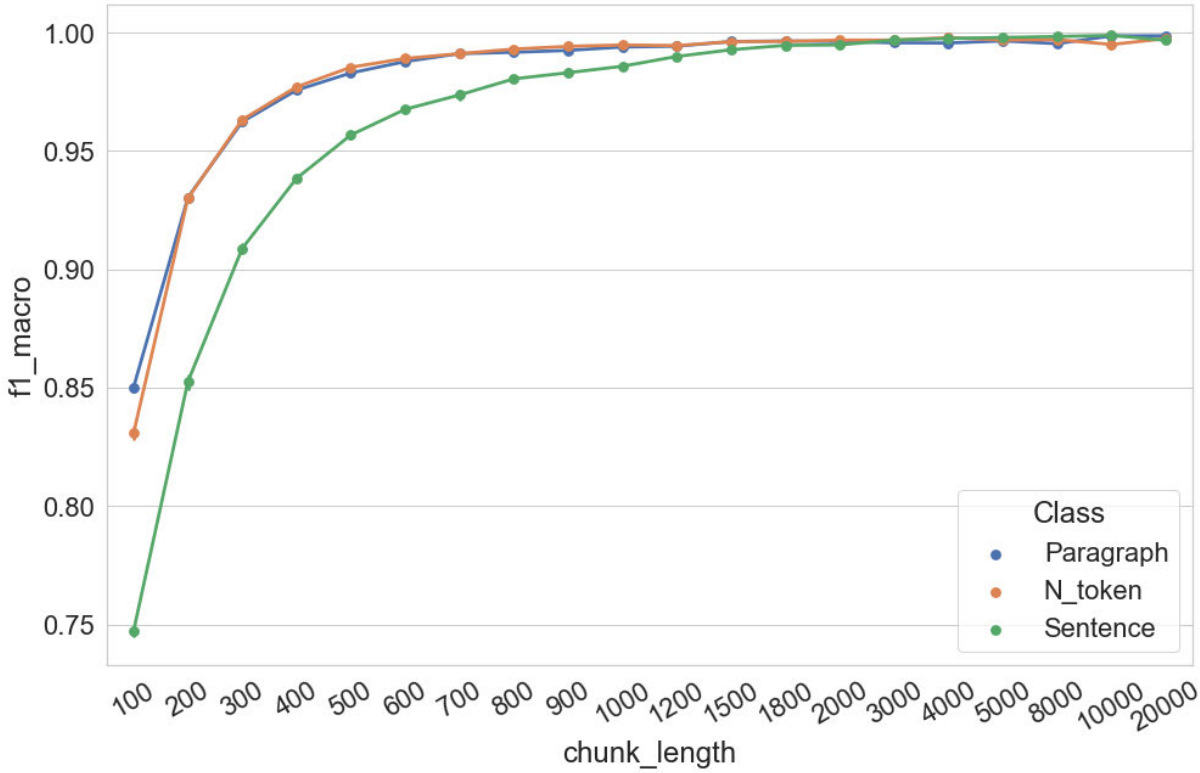

Abbildung 7.43 F1 (Makro)-Werte der Topic-Modeling-basierten Dokumentklassifikation bei drei Chunking-Strategien

7.6.3.2 Topic-Kohärenz

In Abbildung 7.44 wird die Veränderung der NPMI-Wert-Verteilungen bei der Erhöhung der Chunk-Length ***C_s*** visualisiert. Die meisten NPMI-Werte der Topics liegen zwischen 0,1 und -0,4. Im besten Fall hat knapp die Hälfte der Topics einen NPMI-Wert, der über dem NPMI-Kontroll-Wert liegt. Wenn die Anzahl der Topics auf 80 eingestellt ist, zeigt die NPMI-Wert-Verteilung bei einer Erhöhung von ***C_s*** einen absinkenden Trend. Der Median der Verteilung sinkt von ca. -0,13 auf -0,17 ab und bei ***C_s*** = 20.000 sind etwas mehr als 25 % der NPMI-Werte höher als der Kontroll-Wert. Wenn die Anzahl der Topics auf 150 eingestellt ist, zeigt die NPMI-Wert-Verteilung zunächst einen aufsteigenden Trend, wenn ***C_s*** von 100 auf 500 erhöht wird. Mit der weiteren Erhöhung von ***C_s*** ist hier wiederum ein Absinken der Werte zu beobachten. Bei ***C_s*** = 20.000 erreichen zudem nur ca. 25 % der Topics einen NPMI-Wert über dem Kontroll-Wert. Darüber hinaus ist zu beobachten, dass der Maximalwert der Verteilung bei den beiden Settings der Topic-Anzahl mit der Steigerung des Werts von ***C_s*** sinkt, während der Minimalwert der Verteilung trotz einiger Schwankungen keine deutliche systematische Veränderung zeigt.

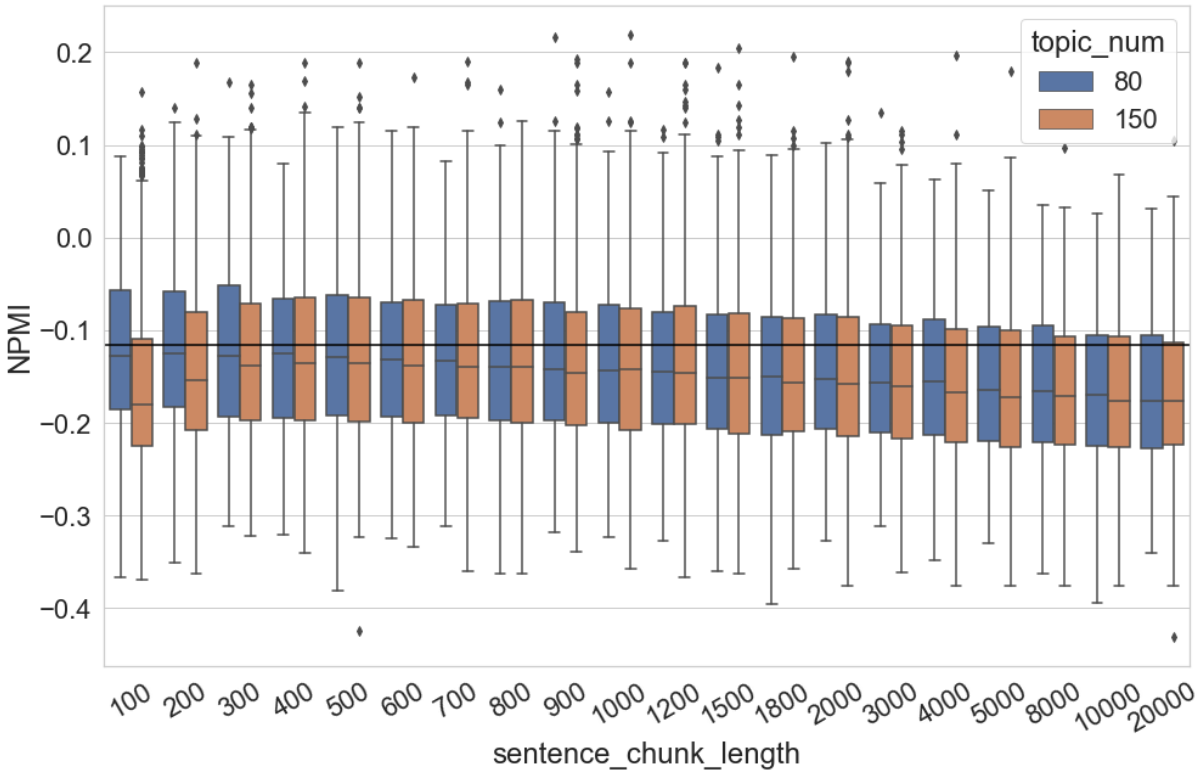

Abbildung 7.44 NPMI-Wert-Verteilung der Topics im Verhältnis zur Chunk-Length ***C_s***

Wenn die Anzahl der Topics stärker variiert wird, ist der absinkende Trend der Maximalwerte mit der Erhöhung von ***C_s*** noch deutlicher zu beobachten (Abbildung 7.45). Der Maximalwert der Verteilung sinkt von ca. 0,15 auf knapp 0,05, während die Minimalwerte der Verteilung keine systematischen Veränderungen zeigen. Bei jedem Setting von ***C_s*** erhält man die Topics mit dem höchsten NPMI-Wert, wenn ihre Anzahl auf Werte zwischen 80 und 150 eingestellt ist. Eine weitere Beobachtung ist, dass die Kombination von kurzen Chunks und einer großen Anzahl von Topics zu einer großen Anzahl von nicht kohärenten Topics führt. Bei ***C_s*** kleiner als 700 und einer Topic-Anzahl von 500 weisen z. B. mindestens 75 % der Topics einen NPMI-Wert auf, der unter dem Kontroll-Wert liegt. Im Gegensatz dazu existieren mehr Topics mit höheren NPMI-Werten, wenn die Modelle weniger Topics enthalten. Dieser Unterschied zeigt sich aber mit der Erhöhung von ***C_s*** weniger deutlich. Bei ***C_s*** = 20.000 haben fast in allen Fällen etwa 75 % der Topics einen NPMI-Wert unter dem Kontroll-Wert, unabhängig von der Einstellung von Anzahl der Topics.

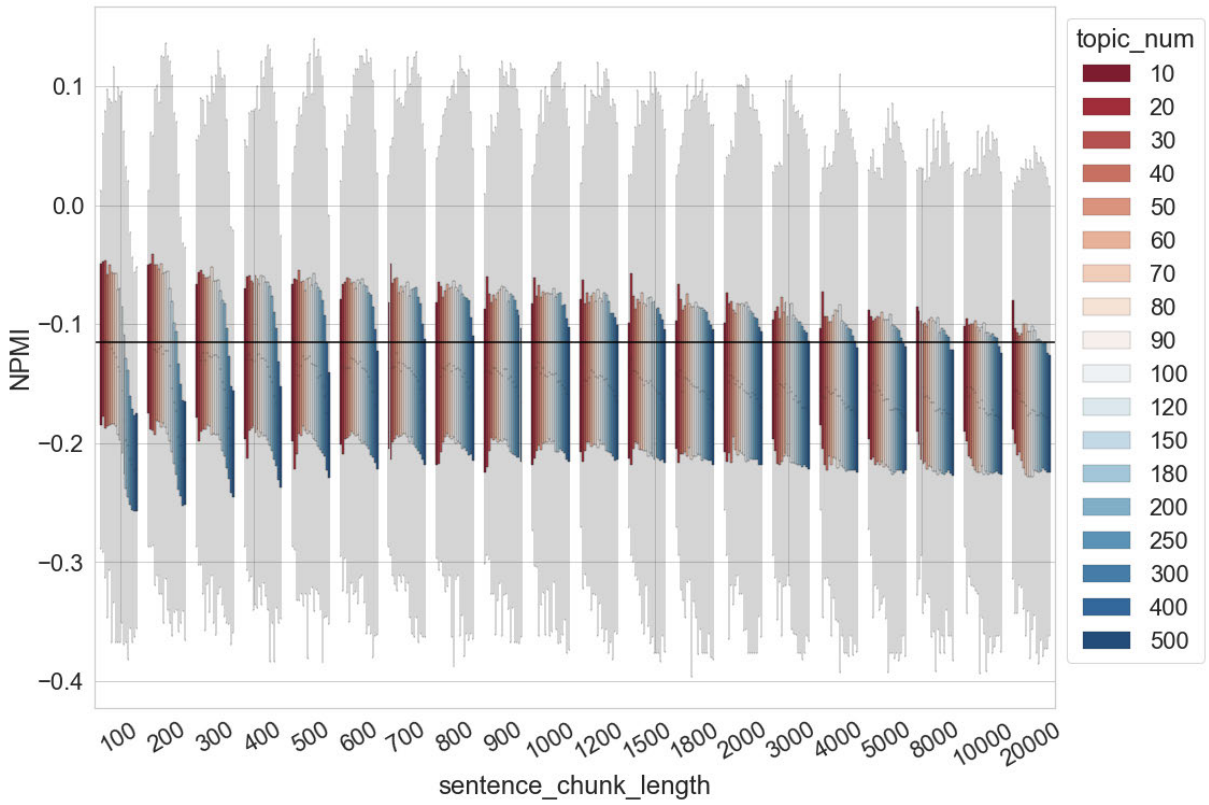

Abbildung 7.45 NPMI-Wert-Verteilung der Topics im Verhältnis zu Chunk-Length ***C_s*** und Anzahl der Topics

Vergleicht man die NPMI-Wert-Verteilungen der Topics der Topic-Modelle, die mit diesen drei verschiedenen Chunkingstrategien trainiert wurden, so lässt sich feststellen, dass es stets umso mehr kohärente Topics gibt, je kürzer die Chunk-Length ist. Als Ausnahme ist anzumerken, dass im Vergleich zu den anderen beiden Strategien das Chunking auf Satzebene zu mehr nicht-kohärenten Topics führt, wenn die Chunk-Length kürzer eingestellt ist. Abbildung 7.46 zeigt, dass die NPMI-Werte-Verteilungen der drei Gruppen (Paragraph, N_token und Sentence) ab einer Chunk-Length größer als 400 nahezu identisch sind. Hier wird die Anzahl der Topics auf 150 eingestellt. Die Ergebnisse bei den anderen Settings der Topic-Anzahl sind ähnlich und werden deshalb nicht mehr visualisiert. Offensichtlich gilt es also zu vermeiden, den Text beim Training des Topic-Modells in zu kurze Sätze aufzuteilen, wenn Topic-Modelle mit mehr kohärenten Topics garantiert werden sollen.

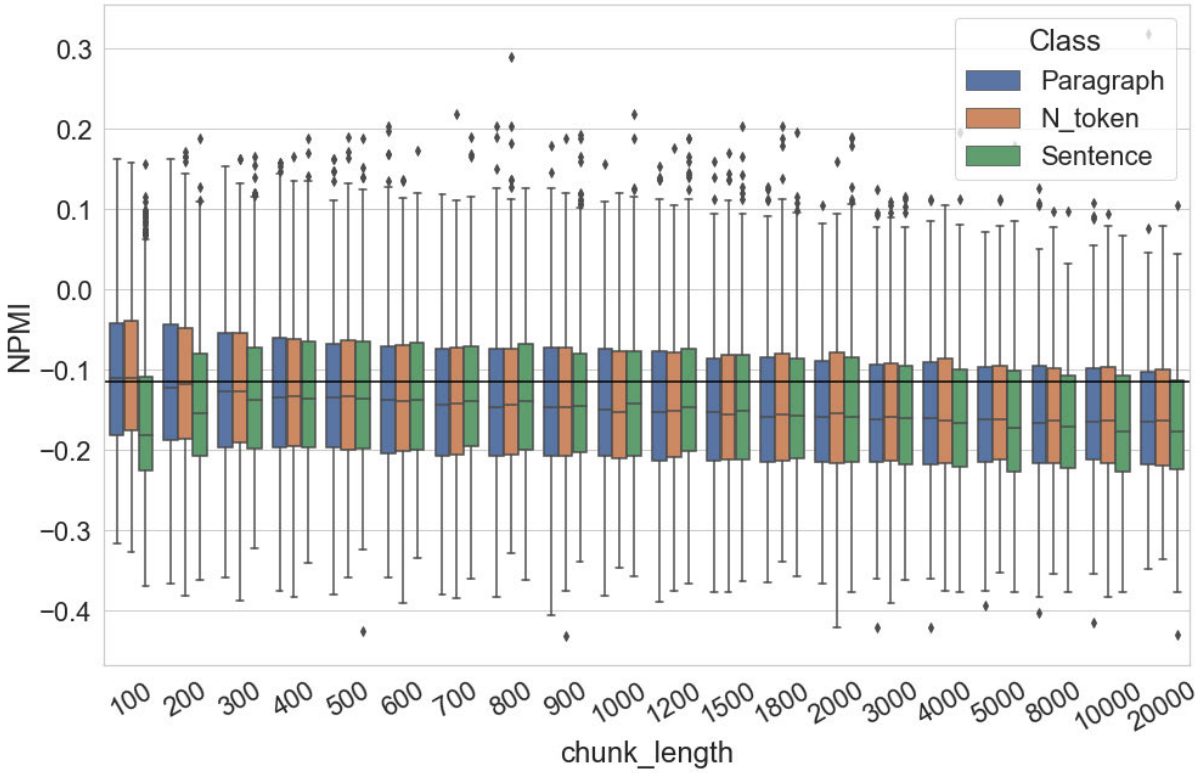

Abbildung 7.46 NPMI-Werte-Verteilung der Topics im Verhältnis der drei Chunking-Strategien

7.7 Zwischenfazit

In diesem Kapitel wird der Einfluss von sechs Faktoren (die Anzahl der Topics, der Hyperparameter Alpha, die Hyperparameter Optimierung, der Hyperparameter Beta, die Anzahl der Iterationen des Gibbs-Samplings und die Chunk-Length) auf Topic Modeling an einem Romankorpus vorgenommen. Einerseits zeigen die Untersuchungen in diesem Kapitel, dass sich der Einfluss einiger Faktoren (z. B. der Hyperparameter Alpha und Beta, die Iteration des Gibbs Samplings) auf das Topic Modeling durch die Änderung des für die Untersuchungen verwendeten Korpus nicht verändert haben. Andererseits haben sich die Ergebnisse einiger Faktoren geändert, wie z. B. die Tatsache, dass eine zu hohe Anzahl von Topics zu schlechteren Klassifikationsergebnissen führte. Grundsätzlich lässt sich feststellen, dass eine mittelgroße Einstellung (nicht zu groß oder zu klein) für fast alle Parameter am besten ist. Allzu extreme Parametereinstellungen sollten vermieden werden.

8. Fazit

In dieser Dissertation wurden das Topic Modeling, genauer gesagt, sechs entscheidende Faktoren durch Experimente evaluiert, nämlich die Anzahl der Topics, Hyperparameter Alpha, Hyperparameter-Optimierung, Hyperparameter Beta, Iteration des Gibbs-Samplings und Chunk-Length. Das Ziel ist die Beantwortung der Frage, unter welchen Umständen das Topic Modeling stabil ist, und somit einen Einblick in die Empfindlichkeit der Methode gegenüber Parametereinstellungen zu geben.

Die Untersuchungen wurden wie folgt durchgeführt: Beim Training der Topic-Modelle wird ein Faktor unterschiedlich eingestellt, während alle anderen Parameter unverändert bleiben. Zusätzlich wird die Anzahl der Topics bei der Untersuchung jedes Faktors variiert. Der Grund dafür ist, wenn die technische Seite des Topic Modeling den Anwendern nicht vertraut ist, können diese zwar der vorgegebene Parameter-Einstellung des Topic-Modeling-Programms vertrauen und sie übernehmen. Allerdings gilt es stets selbst zu entscheiden, wie viele Topics trainiert werden sollen. Aus diesem Grund wurde die Interaktion zwischen den einzelnen Faktoren und der Anzahl der Topics auch evaluiert.

Um die Qualität des Topic-Modells und die der Topics widerzuspiegeln, werden die Topic-Modeling-basierte Klassifikation und die Topic-Kohärenz als die beiden Evaluationsmethoden eingesetzt. Die Verwendung der Klassifikationsergebnisse als Kriterium für die Beurteilung der Qualität des Modells ist einfach zu interpretieren: Bessere Ergebnisse deuten auf eine bessere Modellqualität hin. Im Vergleich dazu ist die Interpretation der Topic-Kohärenz komplizierter. Die Bewertungskriterien für die Qualität der einzelnen Topics sind klar, je höher der Topic-Kohärenz-Wert ist, desto ein Topic ist besser. Es wurde aber auch meistens beobachtet, dass die kohärenten und die nicht-kohärenten Topics in einem Topic-Modell immer gleichzeitig zunehmen. Ein sehr anschauliches Beispiel dafür ist, dass die Spannweite der NPMI-Verteilungen größer geworden ist, wenn die Iteration des Gibbs-Samplings bis zu 300 erhöht wird. Angesichts der Ergebnisse ist es deshalb unrealistisch zu erwarten, dass die Topic-Kohärenz aller Topics in einem Topic-Modell durch die Veränderung der Parametereinstellung ständig verbessert werden.

Die Evaluationen wurden auf einem deutschen Zeitungskorpus und einem deutschen Romankorpus durchgeführt. Dadurch ergab sich die Möglichkeit zu beobachten, ob die

Auswirkungen der einzelnen Faktoren auf die Ergebnisse von Korpus zu Korpus unterschiedlich sind. Beim Vergleich der Testergebnisse ist festzustellen, dass die Ergebnisse tatsächlich auch vom zu untersuchenden Korpus abhängig sind. Die Aussage „abhängig vom Korpus“ stellt allerdings eine sehr unpräzise Schlussfolgerung dar, weil die beiden Korpora sich in mehreren Aspekten wie z. B. Textarten oder Epochen unterscheiden. Im Idealfall sollten z. B. mehrere Romankorpora aufgebaut werden, die in allen anderen Aspekten weitgehend gleich sind. Dann können Evaluationen auf diesen Korpora durchgeführt werden, um herauszufinden, ob die Veränderungen in der Topic-Kohärenz doch einem identischen Trend folgen, wenn alle Topic-Modelle anhand derselben Textart trainiert werden.

Die Evaluationsergebnisse der sechs Faktoren zeigen insgesamt unterschiedliche Situationen. Anhand der Resultate lässt sich zunächst feststellen, dass drei Faktoren doch unabhängig von den zwei Korpora folgendermaßen eingestellt werden sollten, um ein gutes Topic-Modeling-basiertes Klassifikationsergebnis und mehr kohärente Topics sicherzustellen.

- Der Alpha-Wert jedes Topics soll nicht größer als 1 eingestellt werden,
- der Beta-Wert jedes Wortes in jedem Topic soll zwischen 0,001 und 0,01 festgelegt werden,
- die Iteration des Gibbs-Samplings soll mindestens auf 300 eingestellt werden.

Es besteht eine hohe Übereinstimmung zwischen diesen Ergebnissen und den voreingestellten Werten in MALLET:

- Die Summe der Alpha-Werte aller Topics ist 5 (also ist, solange mehr als fünf Topics trainiert werden, der Alpha-Wert jedes Topics kleiner als 1),
- der Beta-Wert jedes Wortes in jedem Topic ist 0,01,
- die Iteration des Gibbs-Samplings ist 1000.

Die Übereinstimmung zeigt, dass die Voreinstellungen dieser drei Parameter in MALLET zuverlässig sind, sie müssen also beim Training des Topic-Modells nicht eigens eingestellt werden. Vermutlich wurden diese Voreinstellungen nach Untersuchungen auf englischsprachige Textdaten festgelegt. Da Englisch und Deutsch beide zu den westgermanischen Sprachen gehören, sind sie sehr eng verbunden. Es ließe sich deshalb leider nicht einfach annehmen, dass diese Parameter beim Training des Topic-Modells immer so eingestellt werden sollten, unabhängig von der Sprache des Textes. Um diese Annahme zu überprüfen, sind weitere Untersuchungen auf mehreren mehrsprachigen Korpora in der Zukunft offensichtlich notwendig.

Die Untersuchungen zu den drei oben genannten Parametern stimmen mit den bisherigen Erkenntnissen über LDA Topic Modeling überein. Im Vergleich dazu liefern die Ergebnisse der Untersuchung in Bezug auf die Hyperparameter-Optimierung neue Informationen: Der Einfluss der Hyperparameter-Optimierung bzw. der Einfluss einer asymmetrische A-priori-Verteilung der Topic-Dokument-Verteilung wird mit der Erhöhung der Chunk-Length der Dokumente im Korpus immer kleiner. Darüber hinaus wird beobachtet, dass die Frage, wann und wie oft die Hyperparameter-Optimierung eingesetzt wird, keinen systematischen Einfluss auf das trainierte Topic-Modell hat. In den Untersuchungen auf beiden Korpora wird beobachtet, dass die Klassifikation durch die Anwendung der Hyperparameter-Optimierung nur dann verbessert wird, wenn die Dokumente im Korpus kürzer als 300 Tokens sind. Zudem zeigt auch die Untersuchung auf dem Zeitungskorpus, dass die Verteilungen der Topic-Kohärenz-Werte größere Spannweiten und einen kleineren Median haben, wenn die Topic-Modelle mit Hyperparameter-Optimierung trainiert werden. Dieses Phänomen wird allerdings in der Untersuchung auf dem Romankorpus nicht beobachtet. In der Voreinstellung in MALLET ist die Hyperparameter-Optimierung nicht aktiviert. Es ist allerdings empfehlenswert, die Hyperparameter-Optimierungsfunktion von MALLET beim Training des Topic-Modells zu verwenden. Dies gilt auch dann, wenn die Dokumente im Korpus länger als 300 Tokens sind. Im Output von MALLET, das die trainierten Topics speichert, befindet sich ein Dezimalzahl-Wert vor jedem Topic (Abbildung 8.1). Dieser Wert stellt den Hyperparameter Alpha des Topics dar. Auf der linken Seite der Abbildung sind alle Werte gleich, weil das Modell ohne Hyperparameter-Optimierung trainiert wurde. Die Werte auf der rechten Seite sind aber aufgrund der Anwendung der Hyperparameter-Optimierung unterschiedlich. Anhand dieser verschiedenen Werte ist einen groben Überblick über die Topics möglich. Ein Topic mit einem großen Wert enthält „Beinahe-Stoppwörter“, die in den meisten Dokumenten häufig vorkommen und nicht viele Beziehungen zu Inhaltswörtern haben. Topics mit sehr kleinen Werten sind oft ungewöhnlich und sehr spezifisch. Solche mit Werten im mittleren Bereich stellen dagegen häufig die interessantesten Topics dar, die für die Exploration des Korpus vergleichsweise nützlich sind.

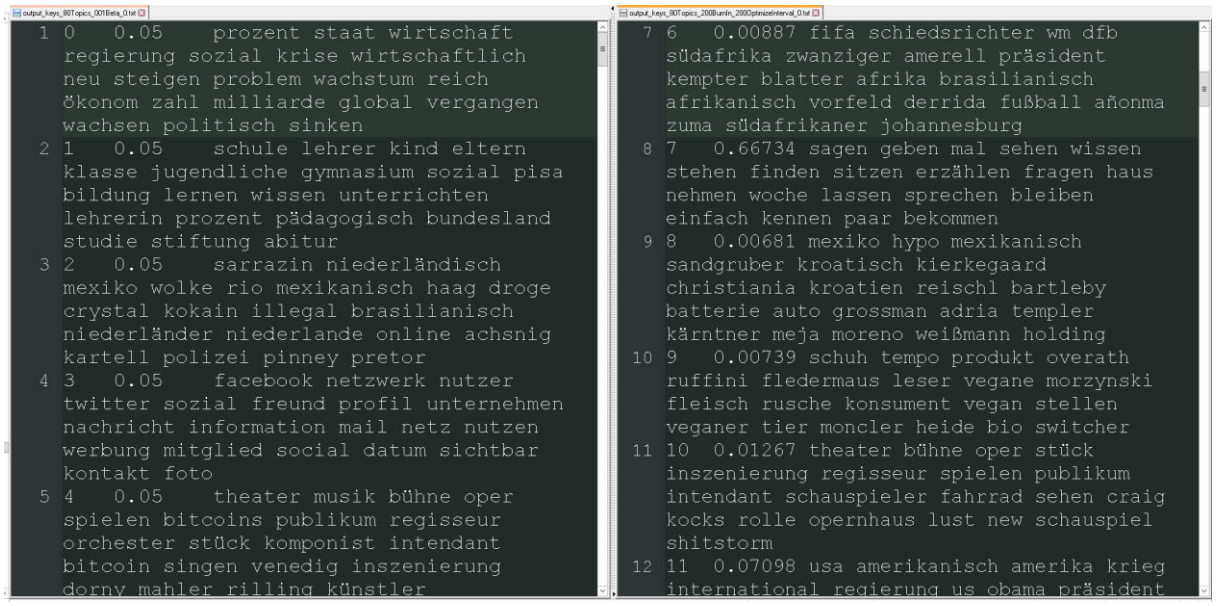

Abbildung 8.1 Output der Topics von MALLET
(links: Topic-Modell trainiert ohne Hyperparameter-Optimierung, rechts: Topic-Modell trainiert mit Hyperparameter-Optimierung)

Im Vergleich zu den oberen vier Faktoren haben allerdings die Ergebnisse der Untersuchungen der beiden anderen Faktoren (Anzahl der Topics und Chunk-Length) keine eindeutige Empfehlung für die Einstellungen ergeben.

Ausgehend von einer einfachen, intuitiven Einschätzung ließe sich vermuten, dass die Anzahl der Topics mit der Größe des Korpus zusammenhängt. Je größer das Korpus ist, desto mehr echte Themen existieren, weshalb die Topic-Modelle mit mehr Topics trainiert werden sollten. Die Untersuchungen zeigen allerdings gemischte Ergebnisse. Das Romankorpus ist zwar ungefähr viermal so groß wie das Zeitungskorpus, die besten Klassifikationsergebnisse der beiden Korpora werden jedoch mit Topic-Modellen erzielt, bei denen die Anzahl der Topics in einem ähnlichen Bereich (80 bis 150 Topics) eingestellt war. Wenn die Topic-Modelle mit noch mehr Topics (bis zu 500) trainiert werden, können die Klassifikationsergebnisse für das Zeitungskorpus auf dem bestmöglichen Niveau gehalten werden, während die Resultate für das Romankorpus eine Tendenz zur Verschlechterung zeigen. Aus der Perspektive der Topic-Kohärenz stehen wir vor einem Dilemma in Bezug auf die Anzahl der Topics: Je mehr Topics trainiert werden, desto mehr kohärente Topics werden erhalten, aber gleichzeitig existiert eine noch größere Zahl von Topics mit geringer Kohärenz. Darüber hinaus wird ein interessantes Phänomen in fast allen Untersuchungen beobachtet, dass die Topic-Modelle, die eine größere

Anzahl von Topics enthalten, empfindlicher gegenüber Veränderungen der anderen Faktoren sind. Die Schlussfolgerung, die man aus den Experimenten in dieser Arbeit ziehen kann, lautet daher, dass die Anzahl der Topics weder zu klein noch zu groß eingestellt werden sollte. Hier stellt sich nun natürlich die Frage, was genau ist „zu klein“ oder „zu groß“? Die voreingestellte Anzahl der Topics in MALLET beträgt beispielsweise 10. Anhand der Untersuchungen auf beiden Korpora kann festgestellt werden, dass dieser voreingestellte Wert eher zu klein ist. Um Hinweise zu einer möglicherweise zu hohen Anzahl von Topics zu finden, könnte beim Training des Modells verschiedene Einstellungen versucht werden. Wenn die Klassifikationsergebnisse schlechter werden oder der Anteil der nicht-kohärenten Topics zu hoch ist, bedeutet dies, dass der Wert zu groß ist.

Der letzte zu untersuchende Faktor ist die Chunk-Length. Hier ist die Schlussfolgerung eindeutig: Je länger die Chunks sind, desto besser sind die Resultate der Klassifikation. Darüber hinaus lässt sich beobachten, dass die Topic-Modeling-basierte Klassifikation genauso gut und manchmal sogar besser als die BoW-basierte Klassifikation funktionieren kann. Interessant wäre es in Zukunft zu überprüfen, ob ähnliche Ergebnisse bei der Topic-Modeling-basierte Klassifikation der Autorenschaft oder Epoche zu beobachten sind. Im Vergleich zur Klassifikation ist das Evaluationsergebnis der Topics weniger eindeutig. In den Untersuchungen auf beiden Korpora wird beobachtet, dass Topic-Modelle mit der Erhöhung der Chunk-Length allmählich mehr nicht-kohärente Topics enthalten. Allerdings wurden die Topic-Modelle mit den kohärentesten Topics trainiert, wenn die Chunk-Length bei der Untersuchung des Zeitungskorpus auf Werte zwischen 800 und 1500 eingestellt wurde. Im Vergleich dazu wurden bei der Untersuchung des Romankorpus die Modelle mit den kohärentesten Topics mit einer Chunk-Length zwischen 100 und 500 trainiert. Schließlich muss auch nochmals betont werden: In den Experimenten wurde jeder Untersuchungskorpus mehrmals segmentiert, während die Chunk-Length unterschiedlich eingestellt war. Die Gesamtzahl der segmentierten Chunks ist deshalb bei jedem Setting der Chunk-Length unterschiedlich. Der Unterschied in der Qualität der trainierten Topic-Modelle ist deshalb auf zwei Faktoren zurückzuführen, die sich nicht voneinander trennen lassen: die Chunk-Length und die Gesamtzahl der Chunks. Die Folge dieses Problems ist, dass der Unterschied zwischen den Topic-Modeling-basierten Klassifikationsergebnissen beim Vergleich der Qualität der Modelle nicht wirklich den realen Unterschied zwischen ihnen widerspiegeln kann: Da die Klassifikationen für verschiedene Datensätze erfolgen, die aus einer unterschiedlichen Anzahl von Dokumenten unterschiedlicher Länge bestehen (d. h. die Klassifikationsaufgaben sind

unterschiedlich), gibt es einen inhärenten Unterschied zwischen den Klassifikationsergebnissen. Daher ist es unmöglich, den Einfluss der unterschiedlichen Qualität der Topic-Modelle auf die Klassifikationsergebnisse zu messen. Wie der Einfluss dieser beiden Faktoren auf das Topic Modeling getrennt untersucht werden kann, ist eine Frage, die in Zukunft geklärt werden muss.

In dieser Arbeit wurde das NPMI-basierte Topic-Kohärenzmaß in dieser Arbeit trotz seiner Nachteile für die Evaluation der Interpretierbarkeit des Topics verwendet, weil es zurzeit die beste vorliegende Evaluationsmethode ist. Es ist aber zu erwarten, dass mit der Weiterentwicklung der Computertechnik in Zukunft neue robuste Methode vorgeschlagen und etabliert werden können. Es könnte natürlich lohnenswert sein, die Qualität der Topics anhand anderen Methoden zu evaluieren. So wird beispielsweise die Type-Token Ratio (TTR) der Topic-Wörter in Bennett et al. (2021) eingesetzt, um die Vielfältigkeit der Topic-Wörter zu überprüfen. Diese interessante Methode wird im Paper als „Topic Gap" bezeichnet. Für die Berechnung des Topic-Gaps eines Modells werden die Top-10 Topic-Wörter aller $\boldsymbol{K}$ Topics zusammengesetzt, der Wert wird dann nach **Anzahl der einzigartigen Wörter / 10$\boldsymbol{K}$** berechnet. So werden z. B. in einem Topic-Modell des Zeitungskorpus zwei Topics über das Thema „Literatur" beobachtet. Die Top-10 Topic-Wörter dieser beiden Topics sind „schreiben, buch, roman, literatur, lesen, autor, werk, schriftsteller, text, gedicht" und „buch, schreiben, lesen, autor, stehen, version, online, erscheinen, roman". Die thematische Struktur eines Korpus lässt sich allerdings nur schwer explorieren, wenn ein Topic-Modell zu viele ähnliche Topics enthält. Das Topic-Gap bzw. die TTR nun können diese Situation mathematisch gut darstellen, wenn es zwischen den Topics Überschneidungen in Topic-Wörtern gibt. Ein Nachteil der TTR ist aber, dass sie von der Textlänge abhängig ist (Tweedie & Baayen, 1998), was den Vergleich des Topic-Gaps zwischen Topic-Modellen mit einer unterschiedlichen Anzahl von Topics problematisch macht. Um das Problem zu vermeiden, kann wiederum die Standardized Type-Token Ratio (STTR, Evert et al. (2017)) eingesetzt werden. In einem Pilottest[77] (Abbildung 8.2) auf dem Zeitungskorpus wird z. B. beobachtet, unabhängig von der Einstellung der Anzahl von Topics, dass die STTR mit der Erhöhung der Chunk-Length $\boldsymbol{C}$ aufsteigt. Intuitiv kann man zwar davon ausgehen, dass es für die explorative Untersuchung eines Korpus normalerweise attraktiver und hilfreicher ist, wenn die Topics vielfältiger sein könnten. Es existieren allerdings keine validen Bewertungskriterien, um zu erklären, welcher Wert der STTR das beste Topic-

[77] Zur Berechnung von STTR werden alle Topic-Wörter in Segmente mit einer festen Größe von 100 unterteilt, dann wird TTR für jedes Segment berechnet. Der Durchschnittswert aller TTRs ist hier die STTR des Topic-Modells.

Modell darstellen kann. In Schöch (2017) werden z. B. in vier Topics die Wörter „Liebe“ „Herz“ und „lieben“ beobachtet, während diese vier Topics durch die anderen Top-Wörter in jedem Topic in unterschiedliche Kontexte gestellt werden, die mit den verschiedenen Untergattungen (Tragödie, Komödie und Tragkomödie) zusammenhängen. Daraus lässt sich feststellen, dass ein STTR-Wert von 1 ebenfalls nicht unbedingt das beste Ergebnis ist. Um diese Art einer deskriptiven Evaluationsmethode, wie es das Topic-Gap darstellt, zuverlässig für die Evaluation der Topics verwenden zu können, ist die vertiefende Forschung in Zukunft notwendig.

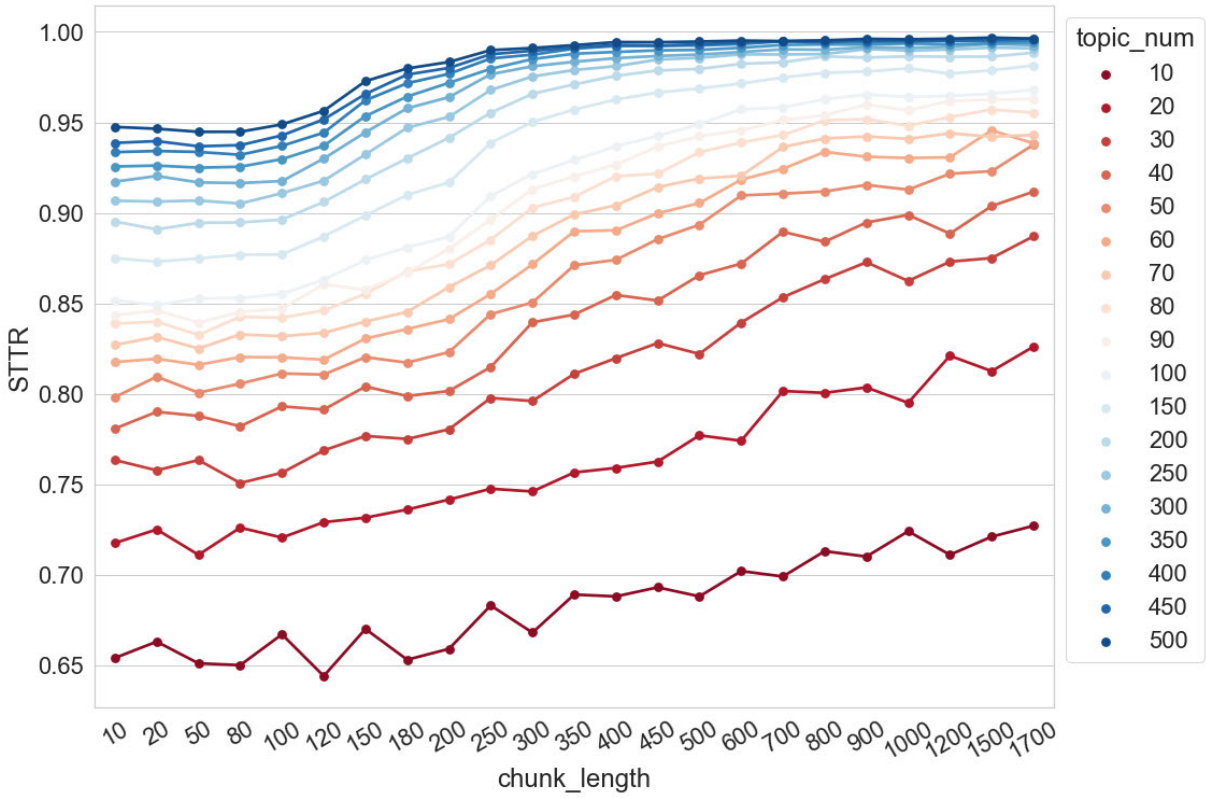

Abbildung 8.2 STTR der Topics (Top-10 Topic-Wörter) im Verhältnis zu Chunk-Length C und Anzahl der Topics (Zeitungskorpus)

Eine andere gute Möglichkeit ist Word Embedding, weil sich diese Methode in den vergangenen Jahren erheblich weiterentwickelt hat und vor allem die semantischen Aspekte der Wörter gut repräsentieren kann. In einer meinen Pilotuntersuchung wird die Leistungsfähigkeit des vortrainierten Modells von fastText (Bojanowski et al., 2017) bei der Aufgabe „Word-Intrusion“ getestet. Das Ergebnis erreichte noch nicht die Qualität der Topic-Kohärenz-

basierten Evaluation[78]. Mit dem Aufkommen von leistungsfähigeren Sprachmodellen wie BERT (Devlin et al., 2018) ist allerdings zu erwarten, dass die Verwendung von Word Embedding für die Evaluation der Topics in Zukunft eine zuverlässigere Option werden könnte.

Diese Arbeit konzentriert sich nur auf LDA Topic Modeling. Im Vergleich dazu haben sich in den letzten Jahren Algorithmen wie BERTopic herausgebildet, die Word Embedding einbeziehen. Neben den Evaluationsmethode muss deshalb die Frage, welcher Algorithmus von Topic Modeling der bessere ist, in Zukunft genauer untersucht werden. In einer meinen Pilotuntersuchung wurden Topic-Modelle auf dem in Kapitel 6 verwendeten Zeitungskorpus trainiert, wobei sowohl LDA als auch BERTopic verwendet wurden. Anschließend wurden die NPMI-Werte der trainierten Topics berechnet und visualisiert. In der Abbildung 8.3 ist zu beobachten, dass die mit BERTopic trainierten Topics im Vergleich zu LDA-Topics deutlich weniger kohärent sind. Vor allem haben die kohärentesten LDA-Topics viel höheren NPMI-Werte, unabhängig davon, wie viele Topics trainiert wurden. Um zu verstehen, warum dieses Ergebnis auftrat, wurde BERTopic genauer analysiert. Topic Modeling mit BERTopic besteht aus fünf Schritte, die nacheinander ausgeführt werden: „Embeddings“, „Dimensionality Reduction“, „Clustering“, „Tokenizer“ und „Weighting scheme“. Zusätzlich kann das „Representation Tuning“ noch hinzugefügt werden. BERTopic geht von einer gewissen Unabhängigkeit zwischen diesen Schritten aus und ist recht modular aufgebaut, so dass der Benutzer für jeden Schritt verschiedene Modelle oder Methoden verwenden kann (siehe Abbildung 8.4). Bei der obigen Untersuchung wurden für die Schritte „Embeddings“ und „Clustering“ das Spacy-Modell (de_core_news_md) [79] bzw. k-Means verwendet. Offensichtlich ist es kein Modell, das kohärente Topics liefert. Es kann sein, dass mit anderen Kombinationen von Methoden/Modellen unter diesen fünf Schritten bessere Topics trainiert werden, aber dies erfordert vom Benutzer ein umfassenderes Verständnis der technischen Seite (z. B. Vor- und Nachteile der verschiedenen Embedding-Modelle, Sprachmodelle und

[78] Einige Details zu diesem Test: Das fastText-Modell bekommt 100 Topics mit jeweils zehn Wörtern (Aletras & Stevenson, 2013). In jedem Topic wird ein zufälliges Wort (Intruder-Wort) hinzugefügt. Dann nimmt das Modell die (10 +1) Wörter jedes Topics als Eingabe, berechnet den Mittepunkt der elf Vektoren und entfernt das Wort, das am weitesten vom Mittepunkt entfernt ist. Dieser Prozess wird ***X***-mal wiederholt, bis das echte Intruder-Wort aus dem Topic entfernt wird. Je öfter der Prozess ausgeführt wird, desto schwieriger ist es, das Intruder-Wort zu identifizieren. Das Topic hat deshalb eine geringere Qualität bzw. Interpretierbarkeit. Jedes Topic wird durch 1000 zufällig ausgewählte Intruder-Wörter überprüft und die 100 Topics werden dann nach ihren durchschnittlichen ***X***-Werten sortiert. Schließlich wird die Rangkorrelation zwischen Word-Intrusion-basiertem Ranking und einem auf menschlichen Bewertungen basierendem Ranking berechnet. Das Ergebnis beträgt hier 0,7. Im Vergleich dazu beträgt die Rangkorrelation in Aletras & Stevenson (2013) zwischen dem Topic-Kohärenz-basierten Evaluationsergebnis und der menschlichen Bewertung 0,8.

[79] https://spacy.io/models/de#de_core_news_md, (07.07.2023).

Clustering-Algorithmen). Es ist daher notwendig, an dieser Stelle zu betonen, dass dieser Test nicht beweist, dass BERTopic der LDA unterlegen ist. Trotzdem ist diese Untersuchung sehr sinnvoll, weil sie die Zielsetzung der vorliegenden Dissertation bestätigt. Es ist notwendig, die Kenntnisse über eine digitale Methode durch die systematischen Evaluationen zu vertiefen, um zu vermeiden, dass die Methode durch eine ungeeignete Anwendung in Frage gestellt wird.

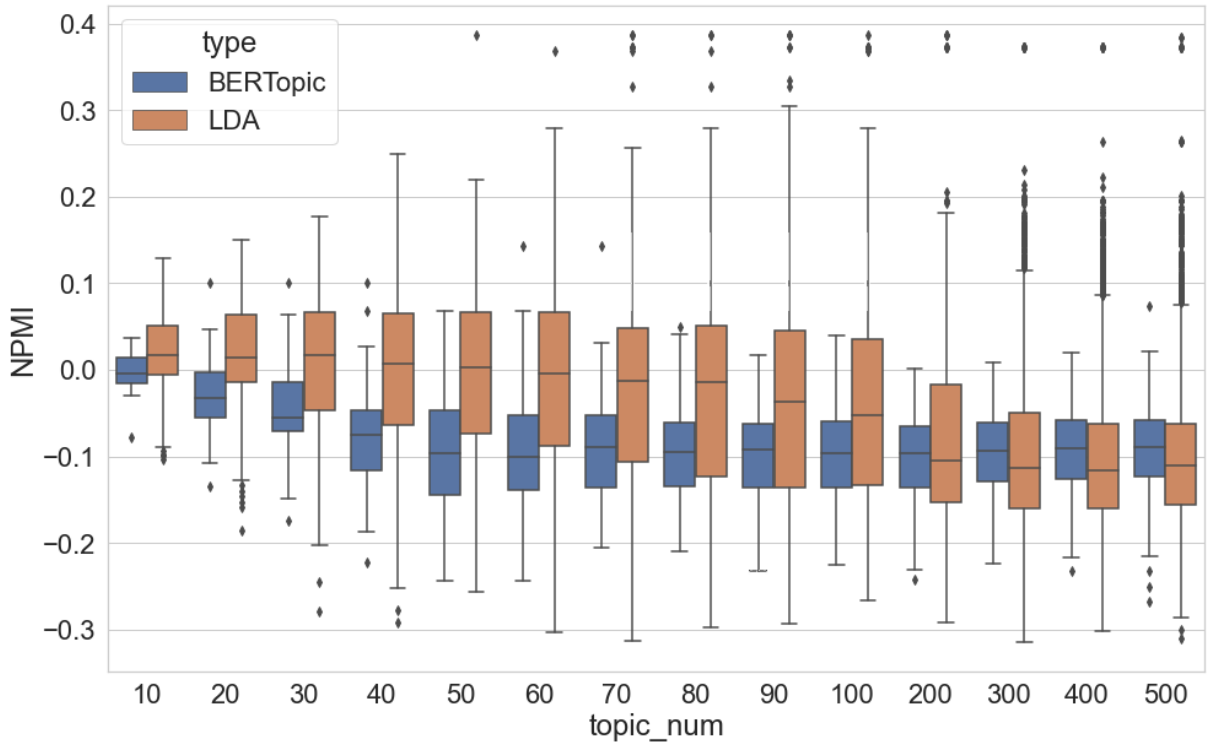

Abbildung 8.3 NPMI-Verteilungen der mit BERTopic und LDA trainierten Topics

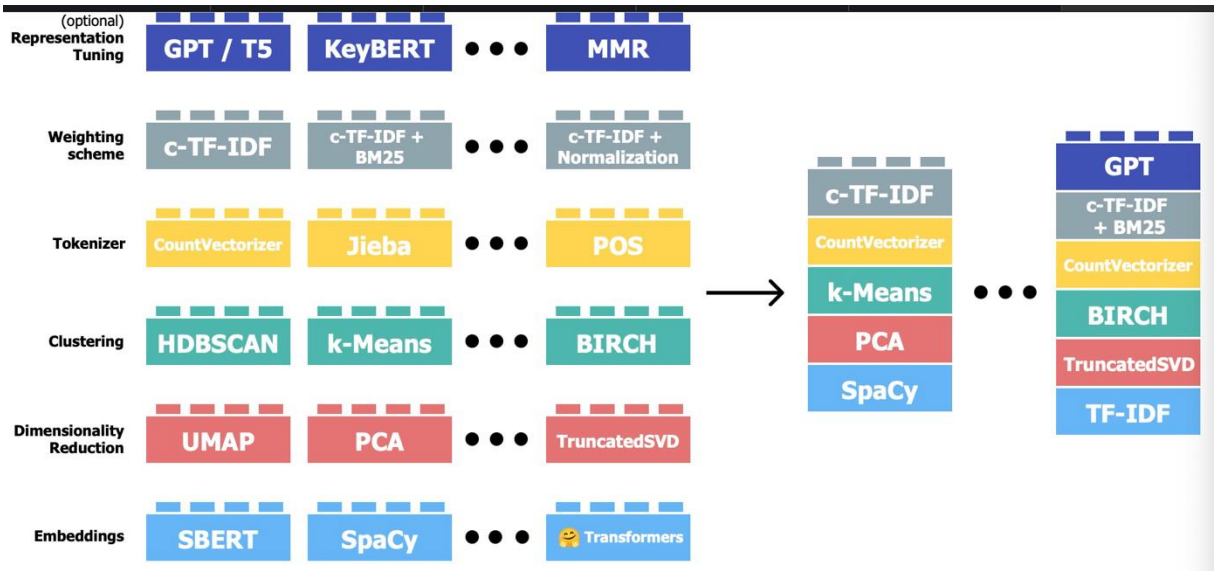

Abbildung 8.4 Modularität von BERTopic[80]

Zuletzt seien noch einige Einschränkungen dieser Arbeit genannt. Zuerst sei hier noch darauf hingewiesen, dass alle Untersuchungen in dieser Arbeit sich nur auf die Interaktion zwischen Anzahl der Topics und den anderen fünf Faktoren überprüft wurde. Vor allem wurde die Dokumentlänge auf 1800 Tokens eingestellt. Um zu verstehen, wie Alpha, Hyperparameteroptimierung, Beta, Iteration des Gibbs Samplings und Chunk-Length miteinander interagieren, sind weitere Experimente erforderlich. Darüber hinaus führen dieselben Tests in dieser Arbeit nicht zu exakt denselben Ergebnissen in beiden Korpora. Zum Beispiel kann die die Verschlechterung der Klassifikationsergebnisse mit der Erhöhung von Anzahl der Topics nur in der Untersuchung auf dem Romankorpus beobachtet werden. In Schöch (2017) wurde das gleiche Phänomen beobachtet. Betrachtet man alle Testergebnisse zusammen, so scheint dieses Phänomen ein Merkmal des literarischen Textes zu sein. Es besteht natürlich auch die Möglichkeit, dass dieses Phänomen nicht mit der Art der Sprache (literarischer Text vs. Sachtext) zusammenhängt, sondern eher mit der Tatsache, dass die Ergebnisse von Korpus zu Korpus variieren. Drittens gibt es neben den sechs in dieser Arbeit evaluierten Faktoren natürlich noch weitere Faktoren, die sich ebenfalls auf die Ergebnisse des Modells auswirken können, wie z. B. die Größe des Korpus oder die anderen Inferenzalgorithmen als das Gibbs Sampling. Um LDA Topic Modeling vollständig zu verstehen, sind Tests in großem Umfang und auf der Grundlage mehrerer Korpora erforderlich. Natürlich ist es unwahrscheinlich, dass eine einzelne Person in der Lage ist, LDA Topic Modeling auf mehrere Korpora in verschiedenen Sprachen zu evaluieren. Es ist daher sehr empfehlenswert, dass alle Anwender:innen dieser Methode im eigenen Forschungsbericht ausführlich beschreiben, wie die Methode eingesetzt und evaluiert wird. Auf diese Weise können wir hoffentlich ein wirklich umfassendes Verständnis der Methode erreichen, indem wir alle Anwendungsfälle zusammenfassen.

[80] https://maartengr.github.io/BERTopic/algorithm/algorithm.html, (07.07.2023).

Literaturverzeichnis

Aletras, N., & Stevenson, M. (2013). Evaluating Topic Coherence Using Distributional Semantics. *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, 13–22. https://www.aclweb.org/anthology/W13-0102

Algee-Hewitt, M., Heuser, R., & Moretti, F. (2015). *On paragraphs: Scale, themes and narrative form*. https://eems.stanford.edu/workspace/druid:xp408qn5109/LiteraryLabPamphlet10.pdf

AlSumait, L., Barbará, D., Gentle, J., & Domeniconi, C. (2009). Topic significance ranking of LDA generative models. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 67–82.

Angelov, D. (2020). Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*.

Armaselu, F. (2018). The Magnifying Glass and the Kaleidoscope. Analysing Scale in Digital History and Historiography. In: *DH*.

Arun, R., Suresh, V., Veni Madhavan, C. E., & Narasimha Murthy, M. N. (2010). On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations. In M. J. Zaki, J. X. Yu, B. Ravindran, & V. Pudi (Hrsg.), *Advances in Knowledge Discovery and Data Mining* (Bd. 6118, S. 391–402). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13657-3_43

Asuncion, A., Welling, M., Smyth, P., & Teh, Y. W. (2009). On smoothing and inference for topic models. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 27–34.

Bailey, S., & Rochester, E. (2016). Testing the Doctrine of Election: A Computational Approach to Karl Barth's Church Dogmatics. In: *DH*.

Balikas, G., Amini, M.-R., & Clausel, M. (2016). On a topic model for sentences. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 921–924.

Balikas, G., Amoualian, H., Clausel, M., Gaussier, E., & Amini, M. R. (2016). Modeling topic dependencies in semantically coherent text spans with copulas. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 1767–1776.

Bartz, T., Beißwenger, M., Pölitz, C., Radtke, N., & Storrer, A. (2014, July). Neue Möglichkeiten der Arbeit mit strukturierten Sprachressourcen in den Digital Humanities mithilfe von Data-Mining. In: *DH*.

Bennett, A., Misra, D., & Than, N. (2021). Have you tried Neural Topic Models? Comparative Analysis of Neural and Non-Neural Topic Models with Application to COVID-19 Twitter Data. *arXiv:2105.10165 [cs]*. http://arxiv.org/abs/2105.10165

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77. https://doi.org/10.1145/2133806.2133826

Blei, D. M., & Lafferty, J. D. (2006a). Correlated topic models. *In Proceedings of the 23rd International Conference on Machine Learning*, 113–120.

Blei, D. M., & Lafferty, J. D. (2006b). Dynamic topic models. *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, 113–120. https://doi.org/10.1145/1143844.1143859

Blei, D. M., & Lafferty, J. D. (2009). Topic models. In *Text Mining* (S. 101–124). Chapman and Hall/CRC.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Blevins, C. (2011). Topic modeling historical sources: Analyzing the diary of Martha Ballard. In: *DH*.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, *30*, 31–40.

Boyd-Graber, J., Hu, Y., & Mimno, D. (2017). Applications of Topic Models. *Foundations and Trends® in Information Retrieval*, *11*(2–3), 143–296. https://doi.org/10.1561/1500000030

Boyd-Graber, J., Mimno, D., & Newman, D. (2014). Care and feeding of topic models: Problems, diagnostics, and improvements. *Handbook of mixed membership models and their applications*, *225255*.

Boyd-Graber, J., & Resnik, P. (2010). Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 45–55.

Broadwell, P., & Tangherlini, T. R. (2015). ElfYelp: Geolocated topic models for pattern discovery in a large folklore corpus. In: *DH*.

Brody, S., & Lapata, M. (2009). Bayesian Word Sense Induction. *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 103–111. https://www.aclweb.org/anthology/E09-1013

Burton, M. (2013). Data Driven Documentation of Digital Humanities Discourse. In: *DH*.

Cao, J., Xia, T., Li, J., Zhang, Y., & Tang, S. (2009). A density-based method for adaptive LDA model selection. *Neurocomputing*, *72*(7–9), 1775–1781. https://doi.org/10.1016/j.neucom.2008.06.011

Chan, H., & Akoglu, L. (2013). External evaluation of topic models: A graph mining approach. *2013 IEEE 13th International Conference on Data Mining*, 973–978.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 288–296.

Chao, A. S., Li, Q., & Liu, Z. (2018). Integrating Latent Dirichlet Allocation and Poisson Graphical Model: A Deep Dive into the Writings of Chen Duxiu, Co-Founder of the Chinese Communist Party. In: *DH*.

Chen, Y., Zhang, H., Liu, R., Ye, Z., & Lin, J. (2019). Experimental explorations on short text topic mining between LDA and NMF based Schemes. *Knowledge-Based Systems*, *163*, 1–13.

Choo, J., Lee, C., Reddy, C. K., & Park, H. (2013). Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, *19*(12), 1992–2001.

Clivaz, C., Clark, E. S., Dilley, P., Faull, K. M., McBride-Lindsey, R., & Phillips, P. (2017). Digital Religion-Digital Theology. In: *DH*.

Da, N. Z. (2019). The Computational Case against Computational Literary Studies. *Critical Inquiry*, *45*(3), 601–639. https://doi.org/10.1086/702594

Deveaud, R., SanJuan, E., & Bellot, P. (2014). Accurate and effective latent concept modeling for ad hoc information retrieval. *Document Numérique*, *17*(1), 61–84. https://doi.org/10.3166/dn.17.1.61-84

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Du, K. (2016). Testing Delta on Chinese Texts. *Digital Humanities 2016: Conference Abstracts*, 781–783. https://dh2016.adho.org/abstracts/15

Du, K. (2017). Authorship of Dream of the Red Chamber: A Topic Modeling Approach. In: *DH*.

Eder, M. (2011). Style-markers in authorship attribution: a cross-language study of the authorial fingerprint. *Studies in Polish Linguistics*, 6(1).

Eder, M., Winkowski, J., Wozniak, M., Górski, R. L., & Grzybowski, B. (2018). Text Mining Methods to Solve Organic Chemistry Problems, or Topic Modeling Applied to Chemical Molecules. In: *DH*.

Egger, R., & Yu, J. (2022). A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in sociology*, 7, 886498.

Eidelman, V., Boyd-Graber, J., & Resnik, P. (2012). Topic Models for Dynamic Translation Model Adaptation. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 115–119. https://www.aclweb.org/anthology/P12-2023

Eisenstein, J., Sun, I., & Klein, L. F. (2014). Exploratory thematic analysis for historical newspaper archives. In: *DH*.

Evert, S., Proisl, T., Jannidis, F., Reger, I., Pielström, S., Schöch, C., & Vitt, T. (2017). Understanding and explaining Delta measures for authorship attribution. *Digital Scholarship in the Humanities*, 32(suppl_2), ii4-ii16.

Evert, S., Wankerl, S., & Nöth, E. (2017). Reliable measures of syntactic and lexical complexity: The case of Iris Murdoch. *Proceedings of the Corpus Linguistics 2017 Conference, Birmingham, UK*.

Falk, M. G. (2016). Faraway, So Close!: Reading Adeline Mowbray Closely Using Topic Modelling. In: *DH*.

Fankhauser, P., Knappen, J., & Teich, E. (2016). Topical diversification over time in the royal society corpus. In: *DH*.

Fei-Fei, L., & Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2*, 524–531 Bd. 2. https://doi.org/10.1109/CVPR.2005.16

Fischer, S., Knappen, J., & Teich, E. (2018). Using Topic Modelling to Explore Authors' Research Fields in a Corpus of Historical Scientific English. In: *DH*.

Fitzgerald, J. D., & Cordell, R. (2018). Stranger Genres: Computationally Classifying Reprinted Nineteenth Century Newspaper Texts. In: *DH*.

Fyfe, P. (2013). Counting Words with Henry James: Towards a Quantitative Hermeneutics. In: *DH*.

Garcia, A. S. (2018). Interrogating the Roots of American Settler Colonialism: Experiments in Network Analysis and Text Mining. In: *DH*.

Goldstone, A. (2014). Let DH Be Sociological! [Short Paper]. In: *DH*.

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, *101*(suppl 1), 5228–5235.

Griffiths, T. L., Steyvers, M., Blei, D. M., & Tenenbaum, J. B. (2004). Integrating topics and syntax. *NIPS*, *4*, 537–544.

Haghighi, A., & Vanderwende, L. (2009). Exploring Content Models for Multi-Document Summarization. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 362–370. https://www.aclweb.org/anthology/N09-1041

Hammond, A., & Brooke, J. (2018). SciFiQ," Twinkle, Twinkle": A Computational Approach to Creating" the Perfect Science Fiction Story". In: *DH*.

Hao, S., Boyd-Graber, J., & Paul, M. J. (2018). Lessons from the Bible on Modern Topics: Low-Resource Multilingual Topic Model Evaluation. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1090–1100. https://doi.org/10.18653/v1/N18-1099

Hettinger, L., Jannidis, F., Reger, I., & Hotho, A. (2016). Significance Testing for the Classification of Literary Subgenres. *Digital Humanities 2016: Conference Abstracts*. https://dh2016.adho.org/abstracts/173

Hofmann, T. (1999). Probabilistic latent semantic analysis. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 289–296.

Hornik, K., & Grün, B. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30.

Hoyle, A., Goel, P., Hian-Cheong, A., Peskov, D., Boyd-Graber, J., & Resnik, P. (2021). Is Automated Topic Model Evaluation Broken? The Incoherence of Coherence. *Advances in Neural Information Processing Systems*, *34*.

Jähnichen, P., Oesterling, P., Liebmann, T., Heyer, G., Kuras, C., & Scheuermann, G. (2015). Exploratory search through interactive visualization of topic models. In: *DH*.

Jautze, K., van Cranenburgh, A., & Koolen, C. (2016). Topic modeling literary quality. In: *DH*.

Jennings, C., & Binder, J. M. (2013). Eighteenth-and Twenty-First-Century Genres of Topical Knowledge. In: *DH*.

Jockers, M. L. (2011). Detecting and characterizing national style in the 19th century novel. In: *DH*.

Jockers, M. L. (2012). Computing and visualizing the 19th-century literary genome. In: *DH*.

Jockers, M. L. (2013). *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press.

Kaufman, M. (2015). Everything on paper will be used against me: Quantifying Kissinger. In: *DH*.

Lau, J. H., & Baldwin, T. (2016). The Sensitivity of Topic Coherence Evaluation to Topic Cardinality. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–487. https://doi.org/10.18653/v1/N16-1057

Lau, J. H., Newman, D., & Baldwin, T. (2014). Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 530–539. https://doi.org/10.3115/v1/E14-1056

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791. https://doi.org/10.1038/44565

Lemaire, B., & Denhiere, G. (2006). Effects of high-order co-occurrences on word semantic similarity. *Current psychology letters. Behaviour, brain & cognition*, *18, Vol. 1, 2006*.

Liu, X., & Croft, W. B. (2004). Cluster-based retrieval using language models. *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval - SIGIR '04*, 186. https://doi.org/10.1145/1008992.1009026

Liu, A., Champagne, A., Douglass, J., Kleinman, S., Russell, J., & Thomas, L. (2017). Open, Shareable, Reproducible Workflows for the Digital Humanities: The Case of the 4Humanities. org" WhatEvery1Says" Project. In: *DH*.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*(2), 203–208. https://doi.org/10.3758/BF03204766

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval* (Anniversary). Cambridge University Press.

Martin, J. H., & Jurafsky, D. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River.

Maryl, M., & Eder, M. (2017). Topic Patterns in an Academic Literary Journal: The Case Of "Teksty Drugie". In: *DH*.

McCallum, A. K. (2002). *Mallet: A machine learning for language toolkit*.

Meneses, L., & Mallen, E. (2017). Semantic Domains in Picasso's Poetry. In: *DH*.

Meneses, L., Martin, J., Furuta, R., & Siemens, R. (2018). Part Deux: Exploring the Signs of Abandonment of Online Digital Humanities Projects. In: *DH*.

Miller, B., Berger, C., Bhattcharyya, S., Caselli, T., Kelman, D., Olive, J., & Rajiva, J. (2016). Contextualizing Receptions of World Literature by Mining Multilingual Wikipedias. In: *DH*.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. *Proceedings of the conference on empirical methods in natural language processing*, 262–272.

Mimno, D., Broadwell, P. M., & Tangherlini, T. R. (2014). The Telltale Hat: LDA and Classification Problems in a Large Folklore Corpus. In: *DH*.

Minka, T., & Lafferty, J. (2002). Expectation-propagation for the generative aspect model. *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, 352–359.

Montague, J., Simpson, J., Rockwell, G., Ruecker, S., & Brown, S. (2015). Exploring large datasets with topic model visualizations. In: *DH*.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.

Murzintcev, N. (2019, Dezember 5). *Select number of topics for LDA model*. https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html

Nelson, R. K., Mimno, D., & Brown, T. (2012). Topic Modeling the Past. In: *DH*.

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 100–108.

Newman, D., Noh, Y., Talley, E., Karimi, S., & Baldwin, T. (2010). Evaluating topic models for digital libraries. *Proceedings of the 10th annual joint conference on Digital libraries*, 215–224.

Nichols, R., Slingerland, E., Nielbo, K., Bergeton, U., Logan, C., & Kleinman, S. (2018). Modeling the Contested Relationship between Analects, Mencius, and Xunzi:

Preliminary Evidence from a Machine-Learning Approach. *The Journal of Asian Studies*, *77*(1), 19–57.

Niemann, K., Schmitz, H.-C., Kirschenmann, U., Wolpers, M., Schmidt, A., & Krones, T. (2012). Clustering by usage: Higher order co-occurrences of learning objects. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12*, 238. https://doi.org/10.1145/2330601.2330659

Norén, F., & Mähler, R. (2017). Text Mining the History of Information Politics Through Thousands of Swedish Governmental Official Reports. In: *DH*.

Olesen, C. G., & Kisjes, I. (2018). OCR'ing and classifying Jean Desmet's business archive: methodological implications and new directions for media historical research. In: *DH*.

Organisciak, P., & Franklin, S. (2017). Modeling Creativity: Tracking Long-term Lexical Change. In: *DH*.

Organisciak, P.; Auvil, L.; Downie, J. S. (2015). Remembering books: A within-book topic mapping technique. In: *DH*.

O'Sullivan, J., Shade, M., & Rowles, B. (2016). Player-Driven Content: Analysing Textual Communications in Online Roleplay. In: *DH*.Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*(2), 111–126.

Phan, X., & Nguyen, C. (2010). *A c/c++ implementation of latent dirichlet allocation (lda) using gibbs sampling for parameter estimation and inference*. https://github.com/sinkovit/GibbsLDAPlusPlus

Purver, M., Körding, K. P., Griffiths, T. L., & Tenenbaum, J. B. (2006). Unsupervised Topic Modelling for Multi-Party Spoken Discourse. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 17–24. https://doi.org/10.3115/1220175.1220178

Quamen, H., & Hjartarson, P. (2014). Big Data and the Literary Archive: Topic Modeling the Watson-McLuhan Correspondence. In: *DH*.

Raftery, A. E., & Lewis, S. (1991). *How many iterations in the Gibbs sampler?* WASHINGTON UNIV SEATTLE DEPT OF STATISTICS.

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 248–256. https://www.aclweb.org/anthology/D09-1026

Rehurek, R. (2014). *Tutorial on Mallet in Python | RARE Technologies*. https://rare-technologies.com/tutorial-on-mallet-in-python/

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Rhody, L. (2012). Topic Modeling and Figurative Language. *CUNY Academic Works - Publications and Research*. https://academicworks.cuny.edu/gc_pubs/452

Riddell, A. B. (2011). Toward a Demography of Literary Forms: Building on Moretti's Graphs. In: *DH*.

Roe, G., Gladstone, C., & Morrissey, R. (2014). Discourses and disciplines in the enlightenment: Topic modeling the french encyclopédie. In: *DH*.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. *Proceedings of the eighth ACM international conference on Web search and data mining*, 399–408.

Rosner, F., Hinneburg, A., Röder, M., Nettling, M., & Both, A. (2014, März 25). *Evaluating topic coherence measures*. Neural Information Processing Systems Foundation (NIPS 2013) - Topic Models Workshop. http://arxiv.org/abs/1403.6397

Savonick, D., & Tagliaferri, L. (2018). The Purpose of Education: A Large-Scale Text Analysis of University Mission Statements. In: *DH*.

Schmid, H. (2013). Probabilistic part-ofispeech tagging using decision trees. *New methods in language processing*, 154.

Schöch, C. (2015). Topic Modeling French Crime Fiction. In: *DH*.

Schöch, C. (2017). Topic Modeling Genre: An Exploration of French Classical and Enlightenment Drama. *DHQ: Digital Humanities Quarterly*, *11*(2).

Schofield, A., Magnusson, M. ans, & Mimno, D. (2017). Pulling out the stops: Rethinking stopword removal for topic models. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 432–436.

Schreibman, S., Kerr, S., & McGarry, S. (2017). The Third Way: Discovery Beyond Search and Browse in Letters of 1916. In: *DH*.

Shadrova, A. (2021). Topic models do not model topics: Epistemological remarks and steps towards best practices. *Journal of Data Mining & Digital Humanities, 2021*, 7595. https://doi.org/10.46298/jdmdh.7595

Shaw, R. (2012). Contours of the Past: Computationally Exploring Civil Rights Histories. In: *DH*.

Shawn, G (2013). Topic Modeling Time and Space: Archaeological Datasets as Discourses. In: *DH*.

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, 63–70.

Smeets, J., Scholtes, J. C., Rasterhoff, C., & Schavemaker, M. (2016). SMTP: Stedelijk Museum Text Mining Project. In: *DH*.

Stevens, K., Kegelmeyer, P., Andrzejewski, D., & Buttler, D. (2012). Exploring topic coherence over many models and many topics. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 952–961.

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, *427*(7), 424–440.

Strube, M., & Ponzetto, S. P. (2006). WikiRelate! Computing semantic relatedness using wikipedia. *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, 1419–1424.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, *101*(476), 1566–1581. https://doi.org/10.1198/016214506000000302

Titov, I., & McDonald, R. (2008). A Joint Model of Text and Aspect Ratings for Sentiment Summarization. *Proceedings of ACL-08: HLT*, 308–316. https://www.aclweb.org/anthology/P08-1036

Tweedie, F. J., & Baayen, R. H. (1998). How Variable May a Constant Be? Measures of Lexical Richness in Perspective. *Computers and the Humanities*, *32*(5), 323–352. https://www.jstor.org/stable/30200474

Underwood, Ted. (2012, April 1). What kinds of "topics" does topic modeling actually produce? *The Stone and the Shell*. https://tedunderwood.com/2012/04/01/what-kinds-of-topics-does-topic-modeling-actually-produce/

Wallach, H. M., Mimno, D. M., & McCallum, A. (2009). Rethinking LDA: Why priors matter. *Advances in neural information processing systems*, 1973–1981.

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. *Proceedings of the 26th annual international conference on machine learning*, 1105–1112.

Wehrheim, L. (2019, März 16). *Von Wirtschaftsweisen undTopic Models: 50 Jahreökonomische Expertiseaus einer Text MiningPerspektive*. DHd 2019 Digital Humanities multimedial und multimodal. 6. Tagung des Verbands „Digital Humanities im deutschsprachigen Raum“, Frankfurt am Main und Mainz. https://doi.org/10.5281/zenodo.4622001

Wei, X., & Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '06*, 178. https://doi.org/10.1145/1148170.1148204

Wevers, M., Smits, T., & Impett, L. (2018). Modeling the Genealogy of Imagetexts: Studying Images and Texts in Conjunction using Computational Methods. In: *DH*.

Xing, L., Paul, M. J., & Carenini, G. (2019). Evaluating Topic Quality with Posterior Variability. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3471–3477. https://doi.org/10.18653/v1/D19-1349

Yamada, T., Inoue, S. (2015). Detection of People Relationship Using Topic Model from Diaries in Medieval Period of Japan. In: *DH*.

Zhu, X. J., Blei, D., & Lafferty, J. (2006). *TagLDA: Bringing a document structure knowledge into topic models*. University of Wisconsin-Madison Department of Computer Sciences.

Anhang

60 Anwendungen des Topic Modeling im Feld der Digital Humanities

Armaselu, F. (2018). The Magnifying Glass and the Kaleidoscope. Analysing Scale in Digital History and Historiography. In: DH.

Bailey, S., & Rochester, E. (2016). Testing the Doctrine of Election: A Computational Approach to Karl Barth's Church Dogmatics. In: DH.

Bartz, T., Beißwenger, M., Pölitz, C., Radtke, N., & Storrer, A. (2014, July). Neue Möglichkeiten der Arbeit mit strukturierten Sprachressourcen in den Digital Humanities mithilfe von Data-Mining. In: DH.

Binder, J. M., & Jennings, C. (2014). Visibility and meaning in topic models and 18th-century subject indexes. In: Literary and Linguistic Computing, 29(3), 405-411

Blevins, C. (2011). Topic modeling historical sources: Analyzing the diary of Martha Ballard. In: DH.

Broadwell, P., & Tangherlini, T. R. (2015). ElfYelp: Geolocated topic models for pattern discovery in a large folklore corpus. In: DH.

Burton, M. (2013). Data Driven Documentation of Digital Humanities Discourse. In: DH.

Chao, A. S., Li, Q., & Liu, Z. (2018). Integrating Latent Dirichlet Allocation and Poisson

Graphical Model: A Deep Dive into the Writings of Chen Duxiu, Co-Founder of the Chinese Communist Party. In: DH.

Clivaz, C., Clark, E. S., Dilley, P., Faull, K. M., McBride-Lindsey, R., & Phillips, P. (2017). Digital Religion-Digital Theology. In: DH.

Croxall, B. (2017). Digital humanities from scratch: A pedagogy-driven investigation of an incopyright corpus. In: DH.

Du, K. (2017). Authorship of Dream of the Red Chamber: A Topic Modeling Approach. In: DH.

Eder, M., Winkowski, J., Wozniak, M., Górski, R. L., & Grzybowski, B. (2018). Text Mining Methods to Solve Organic Chemistry Problems, or Topic Modeling Applied to Chemical Molecules. In: DH.

Eisenstein, J., Sun, I., & Klein, L. F. (2014). Exploratory thematic analysis for historical newspaper archives. In: DH.

Falk, M. G. (2016). Faraway, So Close!: Reading Adeline Mowbray Closely Using Topic Modelling. In: DH.

Fankhauser, P., Knappen, J., & Teich, E. (2016). Topical diversification over time in the royal society corpus. In: DH.

Fischer, S., Knappen, J., & Teich, E. (2018). Using Topic Modelling to Explore Authors' Research Fields in a Corpus of Historical Scientific English. In: DH.

Fitzgerald, J. D., & Cordell, R. (2018). Stranger Genres: Computationally Classifying Reprinted Nineteenth Century Newspaper Texts. In: DH.

Fyfe, P. (2013). Counting Words with Henry James: Towards a Quantitative Hermeneutics. In: DH.

Garcia, A. S. (2018). Interrogating the Roots of American Settler Colonialism: Experiments in Network Analysis and Text Mining. In: DH.

Garcia-Zorita, C., & Pacios, A. R. (2017). Topic modelling characterization of Mudejar art based on document titles. In: Digital Scholarship in the Humanities.

Goldstone, A. (2014). Let DH Be Sociological! [Short Paper]. In: DH.

Hammond, A., & Brooke, J. (2018). SciFiQ," Twinkle, Twinkle": A Computational Approach to Creating" the Perfect Science Fiction Story". In: DH.

Hettinger, L., Jannidis, F., Reger, I., & Hotho, A. (2016). Significance Testing for the Classification of Literary Subgenres. In: DH.

Jähnichen, P., Oesterling, P., Liebmann, T., Heyer, G., Kuras, C., & Scheuermann, G. (2015). Exploratory search through interactive visualization of topic models. In: DH.

Jähnichen, P., Oesterling, P., Heyer, G., Liebmann, T., Scheuermann, G., & Kuras, C. (2015). Exploratory search through visual analysis of topic models. In: Digital Humanities Quarterly (special issue).

Jautze, K., van Cranenburgh, A., & Koolen, C. (2016). Topic modeling literary quality. In: DH.

Jennings, C., & Binder, J. M. (2013). Eighteenth-and Twenty-First-Century Genres of Topical Knowledge. In: DH.

Jockers, M. L. (2011). Detecting and characterizing national style in the 19th century novel. In: DH.

Jockers, M. L. (2012). Computing and visualizing the 19th-century literary genome. In: DH.

Jockers, M. L. (2016). The Ancient World in Nineteenth-Century Fiction; or, Correlating Theme, Geography, and Sentiment in the Nineteenth Century Literary Imagination. In: DHQ: Digital Humanities Quarterly, 10(2).

Kaufman, M. (2015). Everything on paper will be used against me: Quantifying Kissinger. In: DH.

Klein, L. F., Eisenstein, J., & Sun, I. (2015). Exploratory thematic analysis for digitized archival collections. In: Digital Scholarship in the Humanities, 30(suppl_1), i130-i141.

Lee, C. (2018). How are ‘immigrant workers’ represented in Korean news reporting? — A text mining approach to critical discourse analysis. In: Digital Scholarship in the Humanities.

Liu, A., Champagne, A., Douglass, J., Kleinman, S., Russell, J., & Thomas, L. (2017). Open, Shareable, Reproducible Workflows for the Digital Humanities: The Case of the 4Humanities.org" WhatEvery1Says" Project. In: DH.

Maryl, M., & Eder, M. (2017). Topic Patterns in an Academic Literary Journal: The Case Of “Teksty Drugie”. In: DH.

Meneses, L., & Mallen, E. (2017). Semantic Domains in Picasso's Poetry. In: DH.

Meneses, L., Martin, J., Furuta, R., & Siemens, R. (2018). Part Deux: Exploring the Signs of Abandonment of Online Digital Humanities Projects. In: DH.

Miller, B., Berger, C., Bhattcharyya, S., Caselli, T., Kelman, D., Olive, J., & Rajiva, J. (2016). Contextualizing Receptions of World Literature by Mining Multilingual Wikipedias. In: DH.

Mimno, D., Broadwell, P. M., & Tangherlini, T. R. (2014). The Telltale Hat: LDA and Classification Problems in a Large Folklore Corpus. In: DH.

Montague, J., Simpson, J., Rockwell, G., Ruecker, S., & Brown, S. (2015). Exploring large datasets with topic model visualizations. In: DH.

Nelson, R. K., Mimno, D., & Brown, T. (2012). Topic Modeling the Past. In: DH.

Norén, F., & Mähler, R. (2017). Text Mining the History of Information Politics Through Thousands of Swedish Governmental Official Reports. In: DH.

Olesen, C. G., & Kisjes, I. (2018). OCR'ing and classifying Jean Desmet's business archive: methodological implications and new directions for media historical research. In: DH.

Organisciak, P., & Franklin, S. (2017). Modeling Creativity: Tracking Long-term Lexical Change. In: DH.

Organisciak, P.; Auvil, L.; Downie, J. S. (2015). Remembering books: A within-book topic mapping technique. In: DH.

O'Sullivan, J., Shade, M., & Rowles, B. (2016). Player-Driven Content: Analysing Textual Communications in Online Roleplay. In: DH.

Quamen, H., & Hjartarson, P. (2014). Big Data and the Literary Archive: Topic Modeling the Watson-McLuhan Correspondence. In: DH.

Rhody, L. (2014). The Story of Stopwords: Topic Modeling an Ekphrastic Tradition. In: DH.

Riddell, A. B. (2011). Toward a Demography of Literary Forms: Building on Moretti's Graphs. In: DH.

Roe, G., Gladstone, C., & Morrissey, R. (2014). Discourses and disciplines in the enlightenment: Topic modeling the french encyclopédie. In: DH.

Savonick, D., & Tagliaferri, L. (2018). The Purpose of Education: A Large-Scale Text Analysis of University Mission Statements. In: DH.

Schöch, C. (2015). Topic Modeling French Crime Fiction. In: DH.

Schöch, C. "Topic Modeling Genre: An Exploration of French Classical and Enlightenment Drama." In: Digital Humanities Quarterly 11, no. 2 (2017): §1-53.

Schreibman, S., Kerr, S., & McGarry, S. (2017). The Third Way: Discovery Beyond Search and Browse in Letters of 1916. In: DH.

Shaw, R. (2012). Contours of the Past: Computationally Exploring Civil Rights Histories. In: DH.

Shawn, G (2013). Topic Modeling Time and Space: Archaeological Datasets as Discourses. In: DH.

Smeets, J., Scholtes, J. C., Rasterhoff, C., & Schavemaker, M. (2016). SMTP: Stedelijk Museum Text Mining Project. In: DH.

Wevers, M., Smits, T., & Impett, L. (2018). Modeling the Genealogy of Imagetexts: Studying Images and Texts in Conjunction using Computational Methods. In: DH.

Wolff, M. (2013). Surveying a corpus with alignment visualization and topic modeling. In: DH.

Yamada, T., Inoue, S. (2015). Detection of People Relationship Using Topic Model from Diaries in Medieval Period of Japan. In: DH.

Fragebogen: Topic-Evaluation

In der unteren Tabelle steht eine Wortliste aus einer Textsammlung. Die Textsammlung enthält Zeitungsartikel aus 10 Themengebieten:
„**Digital**“, „**Gesellschaft**“, „**Karriere**“, „**Kultur**“, „**Lebensart**“, „**Politik**“, „**Reisen**“, „**Sport**“, „**Studium**“ und „**Wirtschaft**“.

In der Spalte „Topic“ enthält jedes Topic 11 Wörter. Bitte lesen Sie die Wörter durch, und beantworten Sie folgende Fragen nach Ihrem Bauchgefühl:

1. In den 11 Wörtern gibt es **1 zufälliges, falsches Wort**, das hinzugefügt wurde. Thematisch gehört das Wort nicht zu dem Topic. Bitte finden Sie das Wort und tragen Sie das Wort im Feld „**ein falsches Wort**“ ein.

2. Außer dem zufällgen Wort gibt es noch **10 Topic-Wörter**. Wenn Sie diese 10 Wörter lesen, können Sie erraten, zu welchem der oben genannten Themengebieten das Topic gehört? Wenn ja, tragen Sie bitte „**ja**“ im Feld „**interpretierbar?**“ ein. Dann tragen Sie bitte **das Thema** im Feld „**Thema**“ ein. Wenn nein, tragen Sie bitte „**nein**“ im Feld „**Thema**“ ein. Dann tragen Sie bitte „**NA**“ im Feld „**Thema**“ ein.

3. **Wenn Sie ein Thema erraten können**, könnte es sein, dass nicht alle 10 Topic-Wörter nach Ihrem Gefühl zu diesem Thema gehören. Tragen Sie bitte im Feld „**Anzahl der Topic-Wörter**“ ein, **wieviel Wörter** zu diesem Thema gehören. **Wenn Sie ein Thema nicht erraten können**, tragen Sie im Feld „**NA**“ ein.

Die ersten zwei Topics sind zwei Beispiele und sie sind schon evaluiert. Bitte fangen Sie mit dem **dritten** Topic an.

id	**Topic**	**ein falsches Wort**	**interpretierbar?**	**Thema**	**Anzahl der Topic-Wörter**
Bsp1	spiel, spieler, computerspiel, spielen, islam, ps, games, xbox, game, one, medium	islam	ja	Digital	8
Bsp2	sagen, geben, mal, wissen, sehen, verlag, finden,	verlag	nein	NA	NA

	einfach, denken, lassen, stehen				
1	abmahnung, coupland, urmann, wandergeselle, streaming, ip, film, strafanzeige, redtube, adressen, illegal				
2	andersch, tscherkesse, wand, freundschaft, nachbar, tscherkessen, berzona, tscherkessisch, fatimat, organisationen, anderschs				
3	arabisch, israelisch, palästinenser, jerusalem, palästinensisch, israeli, land, khreschatyk, aviv, jüdisch, jude				
4	argentinien, aires, buenos, geschäft, argentinisch, letzt, stellen, lassen, horrenden, reporter, beginnen				
5	artikel, wichtig, versuchen, audio, gesamt, bereit, wermutstropfen, reden, körperlich, situation, wenden				
6	bezeichnen, verantwortlich, reihe, wahrheit, vergangenheit, schätzen, platzeck, räumen, mitmachen, ahnen, dänisch				
7	bitcoins, naturgemäß, bitcoin, währung, automat, rechner, nutzen, dogecoin, überweisung, dollar, börse				
8	brücke, fledermaus, atmosphäre, liberianischen, ankommen, austin, irgendwie, riesig, nieder, innerlich, kühl				
9	buch, schreiben, lesen, autor, roman, literatur, text, museumscafé, verlag, erscheinen, werk				

10	cdu, spd, grüne, csu, politiker, union, politisch, koalition, bundestag, dato, fdp				
11	computer, programm, fifa, neu, club, chaos, schoko, ccc, trojaner, software, hacker				
12	denken, größe, nachhaltig, gedenken, instrument, hingegen, energie, gestalten, reihe, mosambik, polarstern				
13	deutlich, leistung, hälfte, entscheidend, ähnlich, schwach, hegen, heinevetter, auftreten, stochl, regelmäßig				
14	erreichen, frauenfeindlich, ziel, leistung, vergleichbar, niveau, meiste, insgesamt, konzentrieren, einsetzen, angehen				
15	erstellung, euro, million, kosten, pro, milliarde, zahlen, dollar, bezahlen, bekommen, prozent				
16	euro, sagen, prozent, million, kosten, blutig, neu, pro, vergangen, milliarde, zahlen				
17	bedrückend, europäisch, eu, krise, verhindern, frankreich, italien, mai, hoffnung, hart, kommend				
18	facebook, internet, nutzer, netz, netzwerk, kindergärtner, nutzen, datum, google, information, twitter				
19	facebook, nutzer, netzwerk, profil, tumblr, timeline, zuckerberg, schrems, lückemann, schenken, orange				
20	fallen, mirabaud, modell, leisten, gehören, glänzen, tübingen, kolleg, kollegin, spitze, entblößen				

21	fernsehen, tv, sehen, öffentlich, sendung, kommend, zuschauer, ard, laufen, zdf, übertragen				
22	feuer, kongress, lentz, kühn, alternative, neugründung, wahren, untergang, sofortbildkamera, entwerfen, satz				
23	film, kino, drehen, spielen, regisseur, sehen, serie, fernsehen, lächerlich, rolle, hollywood				
24	fluxus, kierkegaard, raddatz, begriff, adressat, irren, nacht, einsehen, umgekehrt, kette, carsten				
25	fühlen, situation, gefallen, fallen, kreisen, stolz, rong, festhalten, weitermachen, jahrzehntelang, relativ				
26	fußball, spieler, bayern, state, verein, fc, trainer, spielen, fan, münchen, hoeneß				
27	geben, erzielt, ahnung, zuständig, irgendetwas, gefallen, fest, erwecken, dran, außerhalb, wechselnd				
28	geben, böse, wissen, gewissensbiss, gewiss, empfinden, eigentlich, hölle, irgendwie, sagen, unheimlich				
29	hustenbonbon, geben, engagement, zumindest, rest, möglichst, neu, lehnen, lernen, zufrieden, saal				
30	geben, führen, erscheinen, gespräch, finden, buch, fall, nennen, ausdrucken, heißen, bleiben				
31	geben, versuchen, hart, organisieren, verdacht, ebenfalls, bestimmt,				

	verlieren, schmal, augenblick, folgend				
32	sprechend, gemeinde, konzept, bergen, zaun, strand, grossman, aufstellen, ausgesprochen, ans, faltin				
33	genau, verwurzelt, gehören, handeln, inzwischen, innerhalb, wobei, bezeichnen, betonen, mal, eher				
34	gewerkschaft, vertrag, arbeitgeber, arbeitnehmer, befristet, mitarbeiter, beschäftigte, metall, befristung, wetzel, icx				
35	gewinnen, kuppel, spielen, nowitzki, sieg, weltmeister, spiel, kampf, boxen, gegner, geschwindner				
36	gewiss, eher, weise, falsch, erfahrung, finden, liegen, problem, unterschied, besonder, philet				
37	hamburg, hamburger, zoo, astoria, glücklicherweise, bikini, berliner, kowalke, fischereihafen, elbe, gedächtniskirche				
38	hochschule, semester, bachelor, abschluss, bewerber, stipendium, brotzeitbrettl, fachhochschule, rektor, studiengebühr, studienplatz				
39	hochschule, studium, uni, universität, studieren, lernen, ausbildung, zusätzlich, international, unangreifbar, fach				
40	hotel, einheimsen, gast, haus, ort, grand, restaurant, tourist, central, berühmt, bahnhof				

41	idee, neu, sagen, lassen, entwicklung, technisch, vorstellungstermin, sogar, welt, längst, inzwischen				
42	hauptgrund, internat, end, kirchhof, sinn, kater, happy, längst, east, vergangenheit, hönisch				
43	international, global, nachname, weltweit, un, zugang, vereint, nation, bekämpfen, protokoll, entwicklungsland				
44	irland, irisch, christiania, ire, mcgeever, ira, fracking, dublin, pusher, kopenhagen, ertmann				
45	islam, islamisch, redlich, muslime, religiös, kopftuch, tragen, khorchide, moschee, salafisten, muslimischen				
46	kilometer, fliegen, luft, wald, strecke, entlang, ehrfurcht, flugzeug, warm, schiff, flug				
47	kirche, glaube, pfarrer, gemeinde, christlich, katholisch, priester, evangelisch, christentum, riesenrad, gottesdienst				
48	kirche, katholisch, souverän, glaube, staat, christlich, geben, welt, heißen, stehen, wort				
49	gebären, koze, brother, wickeln, neffe, lake, überwältigen, segen, bevorzugen, aufrecht, fluch				
50	anzughose, krieg, osten, westen, ddr, weste, regierung, amerika, volk, amerikaner, arabisch				
51	krieg, soldat, afghanistan, bundeswehr, militärisch,				

	armee, nato, irak, konvertieren, einsatz, truppe				
52	land, krieg, arabisch, russisch, soldat, russland, deutsch, canon, westen, regierung, iran				
53	meter, liegen, sehen, stehen, wind, kilometer, lassen, re, paar, himmel, weg				
54	meter, projektleiter, luft, unten, wind, see, eis, steuern, gipfel, ufer, rasen				
55	kaasche, missfeldt, bentwisch, landschaft, heidelberg, deich, oste, urlaubsbild, friesisch, sturmflut, hinwendung				
56	mitarbeiter, streit, werten, interpretieren, gold, vorliebe, ausgerechnet, einschränken, schauspiel, weich, zeitgeschichte				
57	mitarbeiter, unternehmen, arbeiten, sagen, arbeit, erhaben, chef, arbeitgeber, firma, neu, kollege				
58	münchen, deutsch, justizministerin, frankfurt, bayerisch, bayern, main, münchner, köln, düsseldorf, neu				
59	musik, song, singen, band, neu, album, spielen, lied, pop, zittern, hören				
60	nsa, kommunikation, geheimdienst, jahrzehnte, sicherheit, überwachung, kontrolle, information, überwachen, kommunizieren, sammeln				
61	obama, republikaner, barack, öffentlich, amerika, party, demokrat, konservativ,				

	republikanisch, obamas, harmonisch				
62	irrwitzig, passieren, gewiss, persönlich, gefallen, sitzen, woche, urlaub, geschehen, hoffnung, außerhalb				
63	pisa, deutsch, leistung, international, lesen, vergleich, testen, studie, sozial, weltweit, propagandist				
64	fabregas, politisch, politik, europäisch, politiker, eu, regierung, spd, krise, staat, wählen				
65	politisch, politik, europäisch, politiker, eu, regierung, spd, krise, stahlwerk, staat, wählen				
66	prassl, außenseiter, dichand, ostdeutsch, austropop, schnitzelbeat, stapel, finanzmafia, geheimorden, recording, rockabilly				
67	programm, skype, ccc, club, trojaner, computer, gutshaus, hacker, behörde, software, rechner				
68	renner, fleisch, tier, vegane, stadionverbot, essen, gemach, vegan, veganer, vegetarier, veganismus				
69	ring, diktatorisch, apfel, ringen, finden, kauder, mark, rest, verändern, alltag, übel				
70	roma, viertel, central, grand, fish, künstler, braamfontein, galerie, vorbehalten, café, poblenou				
71	roman, erzählen, zukunftsfähigkeit, letzt, geheimnis, fremd, welt, wirken, einst, voll, bloß				

72	russisch, land, russland, krieg, iran, westen, arabisch, putin, ukraine, syrien, architekt				
73	handelshochschule, sagen, fragen, wissen, mal, erzählen, leute, kennen, haus, denken, sitzen				
74	spannender, sagen, kind, haus, eltern, arbeiten, kennen, erzählen, lernen, wissen, besuchen				
75	schlafen, nacht, sms, ii, schmetterling, flats, matratze, efterskolen, königlich, privat, umarmung				
76	schließlich, fällen, unbedingte, ziehen, spitze, beinahe, wunder, mitmachen, treten, vergangenheit, januar				
77	schnell, tempo, annmarie, langsam, mal, paar, merken, beginnen, punkt, sekunde, minute				
78	schule, lehrer, kind, eltern, klasse, lernen, welser, jugendliche, gymnasium, unterrichten, wissen				
79	schwedisch, stockholm, schweden, eis, schwach, tanz, win, schwede, kufe, hlaing, nats				
80	schweiz, ausland, heimat, kanton, wennemer, einsam, mirabaud, tatsächlich, trapp, massiv, gedicht				
81	spielen, frankfurt, absender, gegner, position, feld, spielerin, birgit, jungs, neid, potsdam				
82	sport, maya, geeignet, begegnen, jahreszeit, studio, knie, neonazi, orientalisch, tanz, yoga				

83	sprache, seifart, turgi, abriß, shteyngart, verschwinden, gary, erhältlich, spanisch, abreißen, obrist				
84	steigen, bilden, ziehen, stil, hundert, versprechen, draußen, kochen, schätzen, landjäger, abend				
85	suchen, erfahren, leisten, perfekt, kochen, leiten, stecken, betreiben, region, denken, herkunft				
86	theater, musik, feien, oper, bühne, spielen, stück, publikum, regisseur, geben, orchester				
87	tier, geben, sagen, kaufen, fleisch, produkt, lassen, lebensmittel, ute, meist, halten				
88	twitter, sympathie, welt, dienst, nachricht, media, verbinden, verstehen, score, gleichzeitig, messen				
89	verstricken, un, international, haag, organisation, vereint, wieczorek, rot, stiftung, khmer, aids				
90	unternehmen, firma, gift, sagen, kunde, verkaufen, geschäft, konzern, million, gehören, markt				
91	unternehmen, firma, sagen, million, verkaufen, geschäft, konventionell, kunde, euro, kaufen, prozent				
92	unternehmen, prozent, whiskey, arbeiten, mitarbeiter, stellen, studie, sagen, geben, beruf, häufig				
93	welt, lassen, wissen, nehmen, letzt, böse, sterben, puritanismus, geschehen, weg, jen				

94	welt, wissen, vergessen, lassen, zeigen, bleiben, auge, erinnern, charlene, schön, wahr				
95	werk, baum, verkauf, gebäude, gagosian, carolinensiel, galerie, verkäufer, kunstwerk, schauen, fälschung				
96	wien, wiener, österreichisch, österreicher, wissen, schnitzel, jährig, erzählen, verbleibend, jährige, ehepaar				
97	wort, begriff, anzahl, sprechen, benutzen, klingen, rede, bedeuten, bloß, reden, tatsächlich				
98	exploration, www, hotel, meter, kilometer, fliegen, insel, reise, himmel, meer, sonne				
99	vielerlei, zeitung, magazin, leser, woche, spiegel, journalist, verlag, zeitschrift, medium, online				
100	zürich, auffallen, jarosch, durchgeführt, demnächst, weihnachtsfeier, ordnen, smalltalk, woanders, wahnsinnig, überlassen				

Versicherung an Eides Statt

Ich, Keli Du, Anschrift: Heinrich-Seliger-Str. 24, 60528 Frankfurt, Matrikel-Nr. 1725270, versichere an Eides Statt durch meine Unterschrift, dass ich die Dissertation *Zum Verständnis des LDA Topic Modeling: eine Evaluation aus Sicht der Digital Humanities* selbständig und ohne fremde Hilfe angefertigt, alle Stellen, die ich wörtlich oder dem Sinne nach aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht und ich auch keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe.

Ich versichere an Eides Statt durch meine Unterschrift, dass ich die Regeln der Universität Würzburg über gute wissenschaftliche Praxis eingehalten habe, insbesondere, dass ich die Gelegenheit zum Promotionsvorhaben nicht kommerziell vermittelt bekommen und insbesondere nicht eine Person oder Organisation eingeschaltet habe, die gegen Entgelt Betreuer bzw. Betreuerinnen für die Anfertigung von Dissertationen sucht.

Ich versichere an Eides Statt, dass ich die vorgenannten Angaben nach bestem Wissen und Gewissen gemacht habe und dass die Angaben der Wahrheit entsprechen und ich nichts verschwiegen habe.

Die Strafbarkeit einer falschen eidesstattlichen Versicherung ist mir bekannt, namentlich die Strafandrohung gemäß § 156 StGB bis zu drei Jahren Freiheitsstrafe oder Geldstrafe bei vorsätzlicher Begehung der Tat bzw. gemäß § 161 Abs. 1 StGB bis zu einem Jahr Freiheitsstrafe oder Geldstrafe bei fahrlässiger Begehung.

Frankfurt, 10.05.2024
Ort, Datum

Unterschrift