technische universität
dortmund

fakultät
statistik

Statistische Methoden zur Validierung von Inhaltsanalysen

Dissertation
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
an der Fakultät Statistik
der Technischen Universität Dortmund

Lars Koppers

03.03.2023

Gutachter*innen:
Prof. Dr. Jörg Rahnenführer,
Prof. Dr. Katja Ickstadt

Datum der mündlichen Prüfung:
01.12.2023

Inhaltsverzeichnis

1 Einleitung

Welche Rolle spielen Facebook und Twitter als Informationsquelle für journalistische Texte (Nordheim, Boczek und Koppers, 2018)? Wie wird über das transatlantische Handelsabkommen TTIP in den USA, Großbritannien und Deutschland berichtet (Nordheim, Boczek, Koppers und Erdmann, 2018) und wie funktioniert Journalismus in sozialen Medien (Boczek und Koppers, 2020)? Eine systematische Auswertung von Zeitungsarchiven kann bei der Beantwortung solcher Fragestellungen helfen. Idealerweise arbeiten dabei Wissenschaftler*innen der Journalistik und Methodenwissenschaftler*innen zusammen, um sowohl vom Fachwissen der Anwendung, wie auch der Methodenkenntnis der Statistik zu profitieren.

Diese Dissertation ist im Rahmen des **Dortmund Center für Datenbasierte Medienanalyse (DoCMA)** entstanden. DoCMA ist ein virtuelles Zentrum an der TU Dortmund, an dem Methodenwissenschaftler*innen aus Informatik und Statistik zusammen mit dem Lehrstuhl für wirtschaftspolitischen Journalismus arbeiten. Anhand aktueller Forschungsfragen aus der Journalistik entwickeln beteiligte Methodenwissenschafler*innen zielgerichtet Software und Methoden, die durch die Projektpartner*innen sofort in der Praxis erprobt werden können. Die enge Zusammenarbeit ermöglicht einen kontinuierlichen Austausch der Fachgebiete. Die Methodenwissenschaft erledigt so nicht „bloß“ Auftragsforschung, sondern kann in enger Zusammenarbeit mit der Fachwissenschaft benötigte Werkzeuge entwickeln oder vorhandene anpassen.

Nach einer Einführung in das Textmining und das Methodenbündel der (automatisierten) Inhaltsanalyse (Kapitel 2) und die in dieser Arbeit angewendeten Methoden (Kapitel 3), stellt diese Arbeit drei Themenbereiche vor. In Kapitel 4 wird das R-Paket tosca (Koppers u. a., 2021) vorgestellt. Dieses Paket ist im Rahmen von DoCMA durchgeführten Projekten entstanden und enthält für die Programmiersprache R (R Core

Team, 2023) Funktionalität, die bei automatisierten Inhaltsanalysen immer wieder benötigt wird.

Kapitel 5 beschäftigt sich mit der Problematik des Ziehens von Stichproben zur Validierung von Subkorpora. Da das Herausfiltern der relevanten Texte aus einer Sammlung von Texten in der Regel nicht im ersten Versuch erfolgreich ist, ist der Aufwand hoch, anhand von Stichproben die Qualität der Auswahl zu bestimmen - zumal dies durch menschliche Kodierer*innen geschehen muss. Das Kapitel stellt ein Verfahren vor, mit dem der Aufwand bei mehrmaliger Wiederholung reduziert werden kann.

Abschließend wird in Kapitel 6 ein grundsätzliches Problem beim Einsatz von Themenmodellen betrachtet. Themenmodelle clustern Texte oder Teile davon in einzelne „Themen“, was die automatisierte Auswertung großer Textmengen erleichtert. Oft besteht zwischen dem vom Algorithmus bestimmten „Thema“ und dem was die Fachwissenschaft unter einem Thema versteht ein Unterschied. Ziel einer Modellwahl ist es hierbei, die Parameter des Modells so zu wählen, dass die resultierenden Themen optimal im Sinne des Anwenders sind. In dieser Arbeit kommt das Modell Latent Dirichlet Allocation (LDA, Blei, Ng u. a., 2003) zum Einsatz. Die vorliegende Arbeit untersucht daher schlussendlich in Kapitel 6, inwiefern sich die Topic Coherence (Mimno u. a., 2011) als Kriterium zur Modellwahl eignet.

2 Textmining in den Digital Humanities

Durch die stetig wachsende Menge an verfügbarem Text und der gleichzeitigen Entwicklung von Methoden zur Analyse solcher Textkorpora (Textsammlungen) in der Statistik und der Informatik, ist auch in den Geistes- und Sozialwissenschaften ein neues Forschungsfeld entstanden. Während das Datenmanagement auch in diesen Fachrichtungen schon länger computergestützt abläuft, entwickelt sich der Einsatz von Computern zum Textmining erst jetzt. Für dieses Feld wird der Begriff der *Digital Humanities* verwendet, wobei „Humanities“ als Zusammenfassung von Geistes- und Sozialwissenschaften betrachtet wird (Lemke und Wiedemann, 2016).

Dieses Kapitel soll in eben dieses Wissenschaftsfeld, in dem auch diese Arbeit entstanden ist, einführen. Nach einer allgemeinen Einführung in das Textmining in Abschnitt 2.1, wird in Abschnitt 2.2 das Werkzeug der Inhaltsanalyse und die Entwicklung der Digital Humanities beschrieben. Es folgt die Beschreibung des Vorhabens für diese Arbeit. Abschließend wird in Abschnitt 2.3 auf das Datenmanagement im Projekt eingegangen und der in dieser Arbeit verwendete Korpus der Süddeutschen Zeitung vorgestellt.

2.1 Grundlagen des Textmining

Texte sind in der Regel für einen menschlichen Leser oder eine Leserin gedacht und im Vergleich zu den Datensätzen, mit denen Statistiker*innen in der Regel arbeiten, unstrukturiert. Für geübte Leser*innen ist es normalerweise leicht den Inhalt eines Textes zu erfassen, auch wenn verschiedene Leser*innen den gleichen Text nicht immer gleich interpretieren. So kann die im Text enthaltene Information einfach genutzt werden. Für eine computergestützte Auswertung eines Textes im Rahmen des

Textminings muss, je nach Ziel der Auswertung, einige Vorverarbeitung durchgeführt werden, um die Information für einen Computer „verständlich“ zu machen. Dieser Aufwand lohnt sich insbesondere dann, wenn große Mengen an Text vorliegen, die von einem Menschen nicht mehr in vertretbarer Zeit gelesen und ausgewertet werden können.

Wer sich mit Textmining beschäftigt, stößt auf verschiedene Begriffe, die nicht einheitlich benutzt werden. „Information Retrieval“ beschreibt den Prozess aus Textsammlungen Fragestellungen zu beantworten, während „information extraction“ das Extrahieren von Information beschreibt (zum Beispiel Themenverläufe über die Zeit), die dann anhand von Forschungsfragen weiter untersucht werden können (Miner u. a., 2012, S. 5). Ein weiterer Begriff, der im Umfeld von Textmining benutzt wird, ist „Natural Language Processing“ (NLP). Während klassisches Textmining direkt mit dem Text beziehungsweise einzelnen Wörtern arbeitet, versucht NLP die Bedeutung des Texts zu extrahieren (Kao und Poteet, 2007). Dabei werden nicht nur die Worte, sondern zum Beispiel auch die Syntax verwendet. Methoden wie Themenmodelle, die nicht versuchen die Struktur von Text zu verstehen, werden in der Regel eher dem Textmining und nicht NLP zugeordnet. Die CRAN Task View, die Paketübersichten für Themenfelder zur Verfügung stellt, nutzt den Begriff Natural Language Processing allerdings als Oberbegriff für die Sammlung von Paketen, die sich mit der Analyse von Text beschäftigen (`https://CRAN.R-project.org/view=NaturalLanguageProcessing`). In dieser Arbeit wird Textmining als Oberbegriff verwendet, der auch NLP einschließt.

Je nach Fragestellung können Textsammlungen unterschiedlich verarbeitet werden. Allen Anwendungsszenarien geht voraus, dass die Texte möglichst einheitlich in einen Datensatz zusammengefasst werden müssen. Da es kein einheitliches Format für Textsammlungen gibt, kann der Aufwand hierfür je nach Quelle sehr unterschiedlich sein. Strukturierte Plaintext-Formate, wie XML oder JSON, sind meist gut zu verarbeiten. Bei HTML-Dateien, die in der Regel die Grundlage von Web-Inhalten bilden, müssen relevanter Text und zugehörige Metadaten zuerst identifiziert werden. Dies gilt ebenso für unstrukturierte Plaintext-Formate oder Text-Dateien der großen Office-Programme (odt, docx, pdf). Insbesondere bei den Plaintext-Formaten birgt das Encoding der Dateien eine zusätzliche Fehlerquelle. Encoding bezeichnet dabei die Zuordnung von

Zeichen auf ein Zahlensystem. Neben den beiden üblichen Formaten latin1 (ISO 8859-1 `https://www.iso.org/standard/28245.html`, genutzt von Windows) und das auf Unicode basierende UTF-8 (genutzt von allen anderen gängigen Betriebsystemen) gibt es eine Vielzahl weiterer Encodings, insbesondere die der ISO-Norm 8859. Diese werden genutzt, um außerhalb von UTF-8 verschiedene Sprachräume abzudecken. Das führt dazu, dass alle nicht ASCII-Zeichen (Sonderzeichen und Nicht-Standard-Buchstaben) nicht korrekt dargestellt werden.

Für viele Textmining-Aufgaben genügt es, Häufigkeiten der einzelnen Wörter als Variablen zu verwenden und die Struktur des Textes dabei zu ignorieren (siehe hierzu die Beschreibung der bag-of-words-Annahme in Abschnitt 3.3.1). Gerade bei der Klassifikation oder dem Clustern von Texten wird so vorgegangen. Nachdem die Texte in ihre einzelnen Wörter zerlegt wurden, können die Häufigkeiten der einzelnen Types (zur Begriffserklärung siehe Abschnitt 3.3.1) als Variablen verwendet werden. Somit steht das weite Feld des maschinellen Lernens offen. Als Besonderheit des Textminings seien hier die Themenmodelle genannt, von denen die Latent Dirichlet Allocation (Blei, Ng u. a., 2003) einen zentralen Punkt in dieser Arbeit einnimmt. Auf die Entstehung der Themenmodelle wird in Abschnitt 3.3.2 genauer eingegangen.

Das Verwenden der Worthäufigkeiten als Modell-Variablen kann, je nach Anwendung, verschiedentlich verbessert werden. Um die Zahl der Variablen zu verringern können Flexionen von Wörtern zusammengefasst werden. Dazu gibt es zwei Vorgehensweisen. Das *Stemming* führt ein Wort nach sprachspezifischen Regeln auf einen Wortstamm zurück. Dafür ist der Porter-Stemmer (M.f. Porter, 1980; Martin Porter und Boulton, 2018) gebräuchlich. Dieses Vorgehen funktioniert mit englischen Texten deutlich besser als mit deutschen Texten, da im ersten Fall Flektieren verschiedener Wörter häufiger den gleichen Regeln folgt. Eine andere Möglichkeit ist das lexikonbasierte *Lemmatizing*, das Wörter auf ihre Grundform zurückführt. Dazu wird allerdings ein Wörterbuch benötigt, das diese Zuordnungen enthält. Solche Lexika existieren für verschiedene Sprachen; für die deutsche Sprache bietet das „Wissenschaftszentrum Berlin für Sozialforschung“ einen Lemmatizer an, der auf dem TIGER Korpus (Brants u. a., 2004) der Universität Stuttgart basiert (`https://github.com/WZBSocialScienceCenter/germalemma`). Für manche Untersuchungen kann es nützlich sein, nur bestimmte Wortarten (zum

Beispiel nur Nomen) in das Modell aufzunehmen. Auch die Grammatik eines Satzes enthält Informationen, die genutzt werden können, wie zum Beispiel Verneinungen. Um solche Informationen aus Texten extrahieren zu können, wurden *Part-of-Speech* (POS) Tagger entwickelt. Ein Beispiel dafür ist der TreeTagger (Schmid, 1994), welcher aus Trigrammen, also aus jeweils drei aufeinanderfolgenden Wörtern, die Wortform mithilfe von Entscheidungsbäumen schätzt.

Für spezielle Probleme bei der Analyse von Texten gibt es weitere Methoden um Information aus Texten zu extrahieren. Um Personen und Organisationen zu identifizieren kann *Named-Entity-Recognition* verwendet werden. Ein Beispiel ist das OpenNLP Framework, welches in R über das gleichnamige Paket angebunden ist (Hornik, 2019). So werden die Entitäten auch als gleich erkannt, wenn verschiedene Beschreibungen verwendet werden („Die Bundeskanzlerin“, „Angela Merkel“). Die Stimmung von Texten ist zum Beispiel bei Kundenrezensionen und Einträgen in Diskussionsforen von Interesse. Für die *Sentimentanalyse* existieren Tools wie Sentistrength (`http://sentistrength.wlv.ac.uk/`, Thelwall u. a., 2010), die dabei helfen, Stimmung von Texten zu messen. Dazu wurden Methoden des maschinellen Lernens verwendet, um einen Klassifikator für Sentiment zu generieren. Die Ergebnisse solcher Analysen müssen in der Regel für neue Quellen angepasst werden und können nur sehr begrenzt übernommen werden, da die verwendete Sprache und insbesondere das Sentiment in verschiedenen Textarten (Zeitungsartikel, Disskussionsforeneintrag, Kundenbewertung, ...) unterschiedlich ist.

Im Textmining kommen auch neuronale Netze zum Einsatz. Will man man die Konzepte hinter den Wörtern lernen, bietet sich word2vec (Mikolov u. a., 2013; Rong, 2014) an. Mithilfe eines dreischichtigen neuronalen Netzes kann das Konzept eines Wortes gelernt werden, indem es als Vektor des gesamten Vokabulars dargestellt wird. Die daraus resultierenden Vektorrepräsentationen können nun zum Beispiel subtrahiert werden, um die Konzepte zweier Wörter zu vergleichen. Das Standard-Beispiel (`https://ronxin.github.io/wevi/`) ist für diesen Anwendungsfall ein Datensatz, aus dem gelernt werden kann, dass sich das Wort „king“ zu „queen“ genauso verhält, wie „man“ zu „woman“: king − queen = man − woman. Das Verstehen von Konzepten hinter Wörtern ist auch für Suchmaschinen interessant. Einer Forschergruppe bei

Google entwickelte word2vec und hält auch das Patent (`https://patents.google.com/patent/US9037464B1/en`). Auch andere große Konzerne sind in diesem Bereich des Textmining aktiv: Der Autor des R-Pakets `lda` (Chang, 2015), welches eine zentrale Rolle in dieser Arbeit einnimmt, arbeitete zwischenzeitlich für Facebook (`https://www.linkedin.com/in/jonathan-chang-858242a1`).

Für die automatische Analyse von großen Textmengen bietet das Textmining verschiedene Werkzeuge, die je nach Fragestellung verwendet werden können. Die in dieser Arbeit vorgestellten Methoden sollen Geistes- und Sozialwissenschaftler*innen dabei helfen, manuelle mit automatisierten Auswertungen im Rahmen einer Inhaltsanalyse zu verbinden.

2.2 Journalistischer Hintergrund und Fragestellung

> „Sie werden nun fragen: Wo ist das Material für die Inangriffnahme solcher Arbeiten. Dies Material sind ja die Zeitungen selbst, und wir werden nun, deutlich gesprochen, ganz banausisch anzufangen haben damit, zu messen, mit der Schere und dem Zirkel, wie sich der Inhalt der Zeitungen in quantitativer Hinsicht verschoben hat im Lauf der letzten Generation, nicht am Letzten im Inseratenteil, im Feuilleton, zwischen Feuilleton und Leitartikel, zwischen Leitartikel und Nachricht, zwischen dem was überhaupt an Nachrichten gebracht wird und was heute nicht mehr gebracht wird.“ Max Weber (1911) Geschäftsbericht. Verhandlungen des ersten Deutschen Soziologentages vom 19. – 22.10.1910 in Frankfurt/M (in Weber, 2017, S. 106)

Mithilfe von Textsammlungen können viele Fragestellungen beantwortet werden. Die Unterschiede in der TTIP-Berichterstattung in verschiedenen Ländern, der Anteil an weiblichen Personen unter den erwähnten Expert*innen oder der zeitliche Verlauf der Bankenberichterstattung können so analysiert werden. Um zu solchen Fragestellungen Aussagen treffen zu können, wurde in der Kommunikationswissenschaft das Methodenbündel der **Inhaltsanalyse** entwickelt. Die Inhaltsanalyse ist eine standardisierte

Methode, mit der sich Inhalte aus Texten (beziehungsweise Medieninhalte allgemein) erfassen lassen (Rössler, 2017, S. 271). Sie ist ein Werkzeug zur systematischen Analyse von Mediensammlungen anhand von Forschungsfragen. Dabei können diese Sammlungen Textsammlungen sein, aber auch Video- oder Audiobeiträge können so analysiert werden. Die Inhaltsanalyse kann nicht nur manifeste Bedeutungen des Textes (Wird über Thema XY berichtet?), sondern auch latente Bedeutungen untersuchen (Hat der Autor eine positive Einstellung zum Thema?) (Brosius u. a., 2012, S. 139). Sind solche latenten Bedeutungen von Texten von Interesse, muss sichergestellt werden, dass die aus der Analyse resultierenden Ergebnisse nicht durch die Interpretation der Forschenden beeinflusst wird. Eine strukturierte Inhaltsanalyse hilft dabei, solche Fehlerquellen gering zu halten. Das Feld der Inhaltsanalyse hat sich in den vergangenen Jahren durch den vermehrten Einsatz computergestützter Methoden stark verändert.

Die klassische Inhaltsanalyse

Der Ablauf einer Inhaltsanalyse ist nicht genormt, besteht aber aus Blöcken (vergleiche Brosius u. a., 2012, S. 167ff), die immer wieder in ähnlicher Form auftreten. In einem ersten Schritt muss die für die aktuelle Fragestellung relevante Textsammlung – auch **Korpus** genannt – bestimmt werden. Dies kann zum Beispiel der Gesamtkorpus einer oder mehrerer Zeitungen sein. Dieser Korpus wird anschließend mithilfe von einfachen Filtern eingeschränkt, um nur die für die Fragestellung relevanten Texte zu behalten. Als Filter werden in der Regel Wortsuchen oder zeitliche Einschränkungen (Publikationsdatum) verwendet. Da diese Filter nicht absolut trennscharf sind, müssen sie bezüglich ihrer Güte bewertet und gegebenenfalls angepasst werden. Dies geschieht, indem eine zufällige Auswahl an Texten als Stichprobe gezogen und von menschlichen Kodierer*innen bezüglich ihrer Relevanz bewertet wird. Aus dieser bewerteten Stichprobe lässt sich anschließend die Güte des gefilterten **Subkorpus** bewerten (vergleiche Stryker u. a., 2006). In diesem Schritt sind in der Regel mehrere Iterationen notwendig, bis ein zufriedenstellendes Filterkriterium gefunden ist. Da für jeden Versuch neue Texte von menschlichen Kodierer*innen bewertet werden müssen, ist dieser Schritt ressourcen- und zeitintensiv.

Daran anschließend folgt die eigentliche Analyse der Texte, die im klassischen Fall wieder mithilfe von Kodierer*innen durchgeführt wird. Sowohl in diesem Schritt, als auch im vorangegangenen muss eine Definition dafür vorliegen, was genau kodiert werden soll. Schon die Festlegung der Relevanz der Texte im Vorschritt ist oft nicht so trivial, wie es auf den ersten Blick erscheint. Ob ein Text noch zum interessierenden Thema gehört, ist nicht immer eindeutig. Diese Eindeutigkeit muss aber sichergestellt werden, um objektive Ergebnisse zu erhalten. Die Anleitung für die Kodieraufgabe wird in Form eines Codebuchs festgehalten (Brosius u. a., 2012, S. 157). In der Regel ist es nicht möglich, diese Anleitung so zu formulieren, dass alle Kodierer*innen auf dasselbe objektive Ergebnis kommen. Aus diesem Grund wird mit dem Ausdruck der intersubjektiven Nachvollziehbarkeit gearbeitet (Rössler, 2017, S. 271). Um diese zu bestimmen, müssen wiederum Texte von Kodierer*innen bewertet werden. Die Maßzahlen zur Güte werden in Abschnitt 3.1 vorgestellt. Neben einer hohen **Reliabilität**, also der „Unabhängigkeit“ der Beurteilung eines Textes von der durchführenden Person, muss auch eine hohe **Validität** sichergestellt werden (Rössler, 2017, S. 281). Es muss also sichergestellt werden, dass mit den Kodierungen auch das gemessen wird, was im Rahmen der Forschungsfrage beabsichtigt wurde.

Computational Methods

Der sinnvolle Einsatz von Methoden des Textmining in der geistes- und sozialwissenschaftlichen Forschung wird viel diskutiert. Durch den Einsatz solcher Methoden werden Entscheidungen an „den Algorithmus“ abgegeben, die bei kleineren Datensätzen durch die Forschenden selbst entschieden werden können. Andererseits ermöglichen Textmining-Methoden erst die Analyse großer Textsammlungen. Auch die Arbeitsweise selbst ändert sich. Während die Forschenden beim *close reading* detailliert in die Arbeit mit dem Text einsteigen können, fallen Textmining-Methoden in den Bereich des *distant reading*: Die Texte werden automatisiert ausgewertet und die erhaltenen Ergebnisse weiterverarbeitet. Zwischen den Autoren der Texte und den lesenden Forschern stehen nun zusätzlich die Methoden des Textmining (Stulpe und Lemke, 2016). Diese vermeintliche Blackbox des Textmining führt bei einigen klassisch arbeitenden Wissenschaftler*innen zu Ablehnung dieser Arbeitsweise.

Andererseits gibt es in den vergangenen Jahren auch immer mehr Bestrebungen, Textmining für die Geistes- und Sozialwissenschaften zu erschließen. Das Bundesministerium für Bildung und Forschung (BMBF) stellte 2011 eine Förderlinie für Projekte auf, „die durch die Erforschung, Entwicklung und Anwendung moderner Informationstechnologien die Arbeit in den Geisteswissenschaften erleichtern oder verbessern wollen“ (`https://www.bmbf.de/foerderungen/bekanntmachung.php?B=643`). Projekte wie der Leipzig Corpus Miner (Wiedemann und Niekler, 2016) oder das R-Paket `quanteda` (Benoit u. a., 2018) stellen Tools zur Verfügung, die Forschende aus den Humanities unterstützen sollen. Auch wird eine verbesserte Ausbildung von Studierenden und Forschenden der Sozialwissenschaften im Bereich Textmining diskutiert (Puchinger, 2016).

Um die Akzeptanz von Textmining-Methoden in den Geistes- und Sozialwissenschaften zu stärken, ist die Einbindung in die Theorie der jeweiligen Fächer notwendig. Um die beiden Konzepte des Distant Reading und des Close Reading zu vereinen, schlagen Stulpe und Lemke (2016) den Begriff des *Blended Reading* vor. Unter dem Begriff verstehen die Autoren einen modularen Analyseprozess, der zum einen die Methoden des Textmining (distant reading) und zum anderen das Bearbeiten von Einzeltexten durch die Forschenden (close reading) beinhaltet. Beide Bereiche sollen dabei so verzahnt sein, dass sie sich gegenseitig ergänzen. Zusätzlich kann ein Mixed-Methods-Ansatz gewählt werden (Dumm und Niekler, 2016), in dem verschiedene Analyseverfahren (insbesondere qualitative und quantitative) angewendet werden, um so die resultierenden Ergebnisse zu validieren. Dies hilft dabei auszuschließen, dass die Ergebnisse der verwendeten Methodik geschuldet sind und nicht den zugrundeliegenden Texten.

Für einen sinnvollen Einsatz von Textmining in den Geistes- und Sozialwissenschaften müssen die Forschenden Einfluss auf das Forschungsdesign und damit auch auf die Textmining-Methoden nehmen können. Andersherum müssen zu hohe Erwartungen der Anwender*innen oft enttäuscht werden. Das Anwenden von Textmining ist keine „one-button-Lösung“, die sofort die gewünschten Forschungsergebnisse liefert. Auch hier muss eine geeignete Methode mit den passenden Parametereinstellungen gewählt werden. Nur so kann sichergestellt werden, dass es sich bei den verwendeten Verfahren nicht um eine Blackbox handelt (Wiedemann und Lemke, 2016).

Da der Einsatz von Textmining im Rahmen der Inhaltsanalyse ein vergleichsweise neues Feld ist, hat sich noch kein Goldstandard herausgebildet. Erste Vorschläge gibt es aber schon. Waldherr u. a. (2019) zeigen, wie man in den klassischen Ablauf einer Inhaltsanalyse die Methoden des Textminings integrieren kann und Maier u. a. (2018) schlagen ein Framework vor, mit dem Themenmodelle für geistes- und sozialwissenschaftliche Forschung verwendet und validiert werden können. Die Zielsetzung dieser Arbeit ist es, den Einsatz von Textmining im Rahmen der Inhaltsanalyse mithilfe von statstischen Methoden zu unterstützen und effizienter zu gestalten.

Zielsetzung dieser Arbeit

Diese Arbeit ist im Rahmen des Dortmund Center für datenbasierte Medien-Analyse (DoCMA) entstanden. Im Center arbeiten Forschende der Journalistik, Informatik und der Statistik zusammen an gemeinsamen Projekten. Während die inhaltliche Ausrichtung von Seiten der Journalistik festgelegt wird, ist die Statistik für die Auswahl und Entwicklung geeigneter Analysemethoden verantwortlich. Die Besonderheit dieser interdisziplinären Zusammenarbeit ist, dass beide Bereiche nicht getrennt voneinander bearbeitet werden, sondern ein ständiger Austausch zwischen Methodenentwicklung (Statistik) und Anwendung (Journalistik) etabliert wurde. So kann zum einen sichergestellt werden, dass die Methodenentwicklung nicht am Bedarf vorbei erfolgt und zum anderen kann durch das direkte Feedback der Anwender (*human-in-the-loop*) an jeder Stelle des Prozesses die Analyse der jeweiligen Fragestellung angereichert werden.

Ziel dieser Arbeit ist es, nicht direkt Methoden für (daten)-jounalistische Projekte zur Verfügung zu stellen, sondern Forschung über Journalismus zu unterstützen. Da diese Forschung oft mithilfe von Inhaltsanalysen erfolgt, soll genau dieser Prozess verbessert werden. Die drei Anwendungskapitel sind dabei wie folgt unterteilt:

In **Kapitel 4** wird das im Projekt erstellte R-Paket tosca (Tools for Statistical Content Analysis, (Koppers u. a., 2021)) vorgestellt. Der Schwerpunkt in diesem Kapitel liegt auf der Datenvorverarbeitung und -visualisierung. Zugleich wird die Pipeline zum Bilden von Subkorpora und das Anwenden der Latent Dirichlet Allocation (LDA) an einem Beispiel gezeigt.

Kapitel 5 beschäftigt sich mit dem zeit- und personenintensiven Prozess des Kodierens von zufällig gezogenen Texten. Da dieser Prozess mehrfach zur Validierung von Filtern oder Themenmodellen durchgeführt werden muss und in der Regel auch für die eigentliche Analyse Texte von menschlichen Kodierer*innen kodiert werden müssen, ist eine Reduktion der benötigten Texte wünschenswert. In diesem Kapitel wird gezeigt, wie bei der Berechnung von Precision und Recall bereits vorhandene gelabelte Texte genutzt werden können, um die Zahl der neu zu kodierenden Texte gering zu halten. Außerdem sollen auch neu zu ziehende Texte so gezogen werden, dass möglichst wenig Texte bearbeitet werden müssen.

Abschließend wird in **Kapitel 6** untersucht, welchen Einfluss die Parametereinstellungen der LDA auf die Güte der resultierenden Modelle hat. Insbesondere soll darauf geachtet werden, wie sich die einzelnen Themen bei einer unterschiedlichen Gesamtzahl von Themen im Modell verhalten.

2.3 Datenbasis und Datenmanagement

Das Dortmund Center für datenbasierte Medien-Analyse (DoCMA) hat sich darauf spezialisiert, große Volltextsammlungen von verschiedenen deutschen und internationalen Zeitungen zu erschließen und zu analysieren. Auch journalistische Online-Ressourcen nebst Nutzerkommentaren und Nachrichtenkanälen in sozialen Medien (zum Beispiel WhatsApp `https://www.whatsapp.com`) gehören zu der Sammlung. Für diese Arbeit wurde ein Korpus der Süddeutschen Zeitung verwendet, der im nächsten Abschnitt beschrieben wird. Um die aktuell mehr als 30 verschiedenen Korpora verwalten zu können, wurde im Rahmen des Forschungsdatenmanagements eine einheitliche Struktur für alle Ressourcen und Projekte verwendet.

Das Forschungsdatenmanagement beim DoCMA besteht aus verschiedenen Säulen. Die Daten werden auf einem zentralen Server mit Backup-System verwaltet. Dort wird strikt zwischen den Daten und einzelnen Projekten unterschieden. Beide Bereiche haben eine eigenständige Struktur, die es allen Beteiligten ermöglicht, schnell auf die benötigten Informationen zugreifen zu können. Die Daten werden dabei in

verschiedenen Vorverarbeitungsschritten vorgehalten, soweit die Lizenzbedingungen dies erlauben. Innerhalb der Projektstruktur wird darauf geachtet, dass jedes Projekt unter Verwendung der benötigten Daten ohne weitere Abhängigkeiten zu anderen Projekten reproduzierbar ist. Zentrale R-Funktionen, die auch über das Projekt hinaus nutzbar sind, sind im Paket tosca (Koppers u. a., 2021) veröffentlicht. Programmcode in Entwicklung und Einlesefunktionen, die nur im Projekt genutzt werden, befinden sich in einem Repository (`https://github.com/Docma-TU/tmT`). Das Paket tosca wird in Kapitel 4 vorgestellt.

Diese Arbeit ist als dynamisches Dokument entstanden. Aller Code zur Erzeugung von Tabellen und Grafiken ist in die LaTeX-Dokumente eingebunden. Aufwändigere Berechnungen und Simulationen sind in einzelne R-Skripte ausgelagert. Der Gesamte Code ist unter `https://github.com/lkoppers/Validierung_Inhaltsanalysen` abrufbar, dort befindet sich auch die tablellarische Übersicht, wie die einzelnen Skripte ineinander greifen. Die Liste der installierten R-Pakete ist im Repositorium enthalten. Die Analysen sind in der finalen Fassung mit der R-Version 4.3.2 unter Ubuntu 22.04 erzeugt worden, sind aber qualitativ über verschiedene R-Versionen, Betriebsysteme und Hardware stabil. Auf den verwendeten Rohdaten ist die Arbeit so vollständig reproduzierbar, einzig bei einzelnen konkreten LDA-Modellen wurde bei der Erzeugung der vorverarbeiteten Korpora auf ältere Softwareversionen zurückgegriffen, sodass hier vorverarbeitete Datensätze verwendet werden müssen. In der Arbeit wird an diesen Stellen darauf hingewiesen.

2.3.1 Süddeutsche Zeitung

Die Süddeutsche Zeitung, abgekürzt SZ, ist nach eigenen Angaben mit Millionen Leser*innen Deutschlands größte überregionale Qualitätstageszeitung (*Süddeutscher Verlag* 2022). Nach Informationen der „Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V.“ (`https://www.ivw.eu`) erreichte die Süddeutsche Zeitung im 3. Quartal 2020 die Zahl von 306 553 verkauften Exemplaren (Mo–Sa, ohne Sonntagszeitung). Auflagenstärker sind nur die BILD-Zeitung mit 1 269 181 Exemplaren (inkl. B.Z.) und die ZEIT mit 532 453 verkauften Exemplaren. Alle

Zahlen sind Durchschnittswerte für die einzelnen Ausgaben der jeweiligen Titel in dem Quartal.

Die erste Ausgabe der Süddeutschen Zeitung erschien am 6. Oktober 1945 als erste Zeitung nach dem zweiten Weltkrieg in Bayern. In ihrer ersten Ausgabe bekennt sie sich zu einem süddeutschen Bewusstsein und einem föderalen deutschen Staat:

> Die „Süddeutsche Zeitung“ wird, durch ihren Erscheinungsort München verpflichtet, aus dem süddeutschen, insbesondere bayerischen Geschichtsbewusstsein leben und in Ablehnung eines öden, undeutschen Zentralismus einen kräftigen, den besten Ueberlieferungen [sic] verbundenen Föderalismus vertreten.

Außerdem bekennt sie sich zu einer „geistigen und kulturellen Umgestaltung Europas“ und stellt sich gegen nationalsozialistische und preußisch-militärische Tendenzen. („Zum Geleit“ 1945).

Grundlage des in dieser Arbeit verwendeten Datensatzes sind Datenpakete der Süddeutschen Zeitung. Dabei wurden am 05.01.2015 die Jahrgänge 1994 bis 2014 angekauft, die Jahre 2015 bis 2017 jeweils zu Beginn der Folgejahre. Geliefert werden die Daten als XML-Dateien, mit jeweils einer Datei pro Tag. Laut Verlag sind in den Daten alle Artikel der gedruckten Ausgabe enthalten. Die gedruckte Ausgabe erscheint montags bis samstags. Zusätzlich erscheint am Samstag eine Wochenend-Beilage (SZ (am) Wochenende), die zusammen mit weiteren Beilagen (Literatur/Sport) ebenfalls in den Daten vorhanden ist. Das SZ Magazin, das von einer anderen Redaktion erstellt wird und der SZ am Freitag beiliegt, ist im Datensatz nicht enthalten. Das gleiche gilt für Artikel, die auf dem Online-Auftritt der SZ (`http://www.sueddeutsche.de`) erscheinen. Die XML-Dateien sind ISO-8859-1 kodiert (latin1), nicht-ASCII-Zeichen (Umlaute, Akzente, ...) aber ohnehin als HTML-Entity (ü = `&uuml;`) gespeichert. Jeder Text ist durch ein `<ARTICLE>`-Tag abgeschlossen, wobei diese Definition von Text alle Beiträge der Zeitung enthält, z.B. auch Karikaturen, die nur aus einer Bildunterschrift und einer Autorenzeile bestehen. Bilder sind in dem Datensatz nicht enthalten. Jeder `<ARTICLE>`-Tag enthält weitere Tags, die Metadaten enthalten.

Für diese Auswertung wurden die eindeutige Artikel-ID, das Datum, die Rubrik, die Dachzeile, der Titel, der Zwischentitel, der Untertitel, sowie die Autorenzeile ausgelesen. Ferner wurden die Seitenzahl, auf der der Text erschienen ist, sowie die im Datensatz angegebenen Anzahlen der Wörter und der Zeichen im Text gespeichert. Die verschiedenen Titel, sowie die Autorenzeile befinden sich im `TEXT`-Tag, der den jeweiligen Artikeltext enthält. Diese Metadaten wurden aus den Texten gezogen und waren damit nicht mehr Teil der eigentlichen Texte. Andere Formatierungen, wie Zwischentitel und Bildunterschriften wurden im Text belassen und die zugehörigen XML/HTML-Tags wurden entfernt. Einzig die Paragrafenstruktur (`<P>`) wurde verwendet, um die Texte in einem ersten Verarbeitungsschritt paragrafenweise abzuspeichern.

Da für alle Artikel ein **Datum** hinterlegt ist, kann die zeitliche Entwicklung der Anzahl der Artikel verglichen werden. Abbildung 2.1 zeigt sie in zwei Weisen: In der linken Grafik werden die veröffentlichten Artikel pro Monat dargestellt. Erkennbar sind im beobachteten Zeitraum immer wieder Perioden steigender Artikelzahlen, bei einem insgesamt sinkenden Trend. Die rechte Grafik zeigt die veröffentlichten Artikel pro Wochentag. Hier ist die erhöhte Artikel-Anzahl am Samstag zu sehen, die durch die Wochenendausgabe zu erklären ist. Die 22 Artikel, die an einem Sonntag veröffentlicht wurden, gehören alle zu einer Dokumentation des Papst-Besuchs am 10.09.2006 in München.

Zu jedem Text ist in den Daten eine **Rubrik** hinterlegt. Tabelle C.1 im Anhang listet alle Rubriken des Datensatzes auf, sortiert nach der Anzahl der Texte, die der jeweiligen Rubrik zugeordnet sind. Neben den Rubriken „Wirtschaft“, „Politik“, „Sport“ und „Nachrichten“, denen jeweils über 100 000 Texte zugeordnet sind, existieren auch 22 Rubriken mit weniger als 100 Texten. Bei der Rubrik „Seite Drei“ mit nur einem Text handelt es sich wahrscheinlich um einen Fehleintrag der Rubrik „Die Seite Drei“ mit 14 669 Einträgen. Die Rubriken „Sonstiges“ (173 Texte), „Vermischtes“ (15 Texte) sowie „unbekannt“ (10 Texte) scheinen sehr seltene Sammelrubriken für Texte zu sein, die nicht klar einzuordnen sind.

Betrachtet man den zeitlichen Verlauf der 109 Rubriken, sind einige weitere Auffälligkeiten in den Daten zu finden. Abbildung 2.2 zeigt den Anteil aller Rubriken an

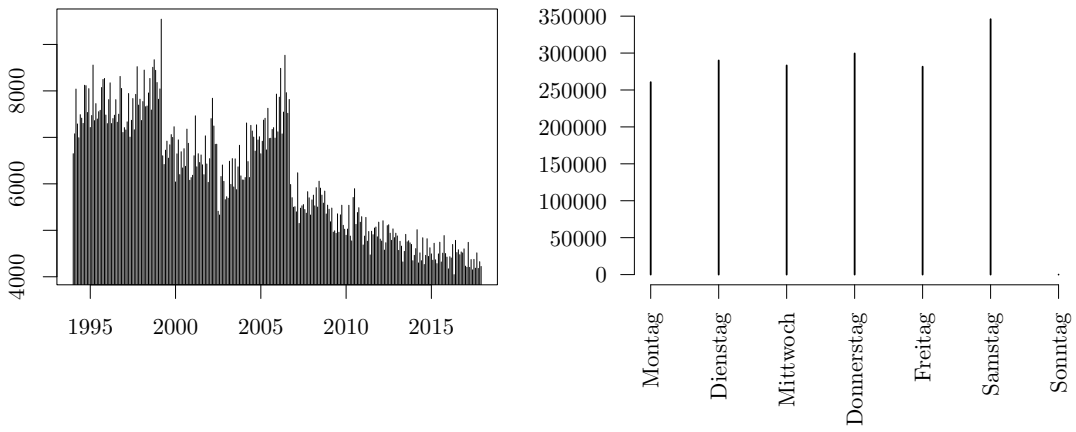

Abbildung 2.1: Zeitliche Verteilung der Anzahl der Artikel nach Monat (links) und Wochentag (rechts)

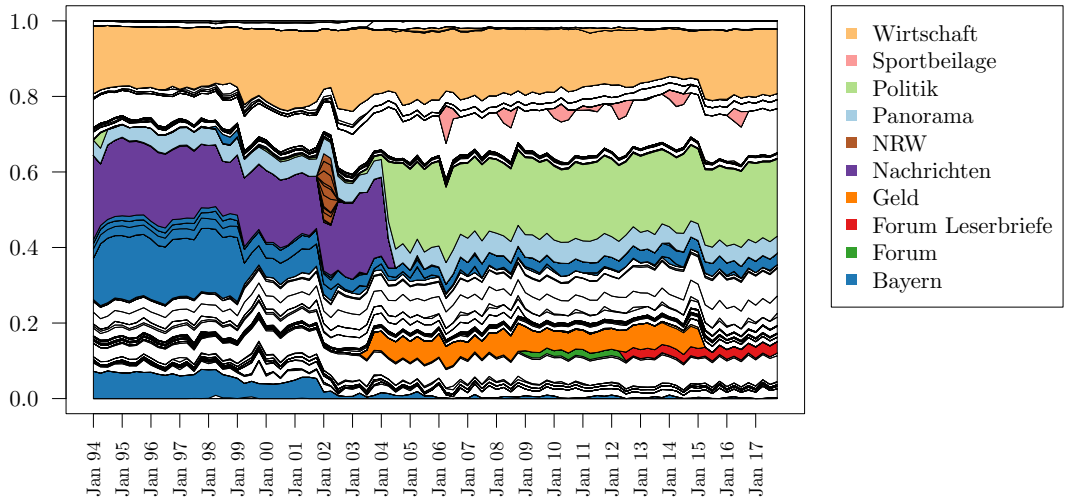

Abbildung 2.2: Anteile der Rubriken in der Süddeutschen Zeitung nach Quartalen. Farblich markiert sind einzelne Rubriken oder Rubrik-Gruppen

den in Quartalen zusammengefassten Zeitpunkten. Dabei sind einige Rubriken bzw. Rubrik-Gruppen farblich markiert. Die einzelnen Rubriken sind offensichtlich nicht alle über die gesamte Zeit gleichmäßig verwendet worden. Während die Rubrik „Wirtschaft“ über den gesamten Zeitraum existiert, ist die Rubrik „Geld“ nur von 2003 bis 2015 zu finden. Die Verläufe der beiden Rubriken legen nahe, dass „Geld“ außerhalb dieses Zeitraums der Rubrik „Wirtschaft“ zugeordnet wurde. Andere Rubriken, wie zum Beispiel „Forum“ und „Forum Leserbriefe“, wurden offensichtlich nur umbenannt. Einen weiteren Teil machen die auf einzelne Bundesländer bezogenen Rubriken aus. Während die Rubriken, die auf Grund ihres Namens Bayern oder München (der Sitz des Verlags) zugeordnet werden können, einen immer kleineren Anteil am Gesamtumfang der Zeitung einnehmen, wurden verschiedene Rubriken, die sich auf das Bundesland Nordrhein-Westfalen beziehen, nur im Jahr 2002 genutzt (Die Rubrik „NRW-Bayern“ wurde dabei NRW zugeordnet). Bei anderen Kategorien hat nicht nur ein Namenswechsel zu einem bestimmten Zeitpunkt stattgefunden. Die Rubrik „Nachrichten“ scheint durch „Politik“ abgelöst worden zu sein. Letztere existiert allerdings über den gesamten Zeitraum, wenn auch meist sehr schwach besetzt. Die Sportbeilage ist in dem Datensatz nur ab und zu enthalten, sie scheint nicht regelmäßig zu erscheinen.

Diese Beispiele zeigen, dass die Variable Rubrik nicht ausreicht, um Artikel thematisch zu sortieren. Neben dem unterschiedlichen Umgang mit den 109 Rubriken im zeitlichen Verlauf, sind diese auch zu grob gefasst, um aussagekräftig zu sein. Einige, wie zum Beispiel die Beilagen-Rubriken, geben auch eher die Organisationsstruktur, als die thematische Ausrichtung wider.

Betrachtet man die im Datensatz angegebene Zahl der **Wörter pro Artikel**, sieht man in Abbildung 2.3, dass die Texte in ihrer Länge über die Zeit nicht homogen sind. Während die Gesamtzahl der publizierten Wörter pro Monat über die Jahre schwankt, steigt die durchschnittliche Anzahl der Wörter pro Text von anfangs knapp über 300 auf etwa 450 Wörter pro Artikel. Für diese erste Analyse wurden die vom Verlag angegebenen Zahlen verwendet. Ein ähnliches Resultat ergibt eine Auswertung der Anzahl der Zeichen in den Texten, ebenfalls wie sie vom Verlag gezählt wurden, siehe dazu Abbildung B.1 im Anhang. Für die weiteren Auswertungen werden eigene Zählungen verwendet.

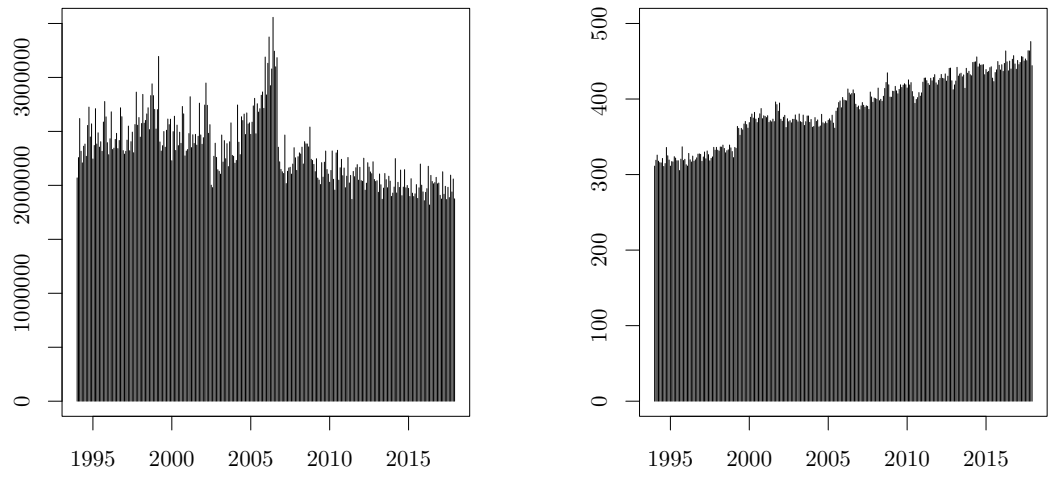

Abbildung 2.3: Zeitliche Verteilung der Anzahl der monatlich publizierten Wörter (links) und der durchschnittlichen Wörter pro Artikel und Monat (rechts)

3 Statistische Methoden

3.1 Intercoderreliabilität

Sollen im Rahmen einer Inhaltsanalyse Texte quantitativ ausgewertet werden, müssen diese in der Regel von menschlichen Kodierer*innen in Kategorien eingeteilt werden. Dies können im einfachsten Fall binäre Kategorien (Text ist relevant/nicht relevant) sein. Es sind aber auch mehrere (ordinale) Klassen (5 Sterne-Bewertung) oder Zählvariablen (Anzahl der im Text genannten Personen) möglich. Damit die Codier-Aufgabe für andere Personen nachvollziehbar ist, wird die Codieranweisung in einem Codebuch festgehalten. Trotz des Festlegens der Kriterien für die jeweilige Aufgabe, ist es oft so, dass bei anspruchsvolleren Aufgaben (Thema XY ist im Text behandelt worden) mehrere Kodierer*innen zu unterschiedlichen Entscheidungen kommen. Da diese Ergebnisse aber genutzt werden sollen, um objektive Aussagen treffen zu können, muss die Reliabilität zwischen den Kodierer*innen anhand einer Stichprobe, die von mehreren Kodierer*innen bearbeitet wurde, überprüft werden.

Eine einfache prozentuale Übereinstimmung der Kodierer*innen kann zu Fehlschlüssen führen: In einem binären Beispiel, in dem ein Kodierer 99 von 100 Texten der ersten Kategorie und nur einen Text der zweiten Kategorie zugeordnet hat, die zweite Person aber alle Texte der ersten Kategorie, läge die prozentuale Übereinstimmung bei 99 %. Da es in diesem Fall keine Übereinstimmung in der zweiten Klasse gibt, sollte ein geeignetes Maß diesen Fall mit Null bewerten. Krippendorffs α (Krippendorff, 1970; Krippendorff, 2011) erfüllt diese Bedingung. Die Maßzahl berechnet sich aus dem Quotienten der beobachteten Unterschiede der Kodierer*innen und der unter zufälliger Klassenwahl erwarteten Unterschiede.

$$\alpha = 1 - \frac{D_0}{D_e} = 1 - \frac{\text{beobachtete Unterschiede}}{\text{erwartete Unterschiede}}$$

Dies führt dazu, dass bei Übereinkunft aller Bewertungen der Bruch zu Null wird und $\alpha = 1$ gilt. Ist die Zahl der Unterschiede so groß wie unter Zufall erwartet, ist der Bruch eins und $\alpha = 0$. Auch negative Werte sind möglich, wenn mehr Unterschiede auftreten, als unter Zufall erwartet werden. Im Vergleich zu anderen gebräuchlichen Maßzahlen in diesem Bereich, wie Kendalls W (Kendall und Smith, 1939, nur für ordinale Daten, keine fehlenden Werte) oder Cohens κ ((Cohen, 1960), nur für nominale oder ordinale Daten, nur zwei Kodierer*innen), kann Krippendorffs α auch für mehr als zwei Kodierer*innen, für verschiedene Skalenniveaus und beim Vorliegen von fehlenden Werten angewendet werden. Durch die Toleranz von fehlenden Werten können auch Beobachtungen in die Berechnung aufgenommen werden, die nicht von allen Kodierer*innen bearbeitet wurden.

Sei n_{uc} die Zahl der Kodierer*innen, die für die interessierende Kodiervariable Beobachtung $u = 1, \ldots, U$ der Ausprägung $c = 1, \ldots, C$ zugeordnet haben. Dabei bezeichnet $n_{u.} = \sum_c n_{uc}$ die Anzahl der für Beobachtung u abgegebenen Kodierungen. Bei den Zeilensummen bezeichnet $n_{.c} = \sum_{u|n_{u.} \geq 2} n_{uc}$ allerdings nur die Summe der Kodierungen mit Ausprägung c, der Beobachtungen, für die mehr als ein Votum abgegeben wurde. So werden nur Beobachtungen berücksichtigt, die von mindestens zwei Kodierer*innen bewertet wurden. Analog dazu ist die Gesamtsumme $n_{..} = \sum_{u|n_{u.} \geq 2} n_{u.}$ definiert.

Krippendorffs α berechnet sich nun dadurch, dass die Distanzen zu den einzelnen Kodierungen berechnet und in Relation zu den unter Zufall erwarteten Distanzen gesetzt werden

$$\alpha = 1 - (n_{..} - 1)\frac{\sum_u \frac{1}{n_{u.}-1} \sum_c \sum_{k>c} n_{uc} n_{uk} \delta^2_{ck\,\text{metric}}}{\sum_c \sum_{k>c} n_{.c} n_{.k} \delta^2_{ck\,\text{metric}}}. \tag{3.1}$$

Je nach Skalenniveau werden passende Distanzmaße $\delta^2_{ck\,\text{metric}}$ verwendet. Dabei ist δ^2_{metric} die symmetrische Matrix mit C Zeilen und Spalten, die die Distanz zwischen zwei Ausprägungen c und k angibt. Üblicherweise liegen nominal-, ordinal- oder intervall-

skalierte Daten vor, sodass eins der folgenden Distanzmaße verwendet werden kann. Auf die Standardisierung der folgenden Terme kann dabei verzichtet werden, da diese das α nicht beeinflusst.

$$\delta^2_{ck\,\text{nominal}} = \begin{cases} 0 & \text{iff } c = k \\ 1 & \text{iff } c \neq k \end{cases} \tag{3.2}$$

$$\delta^2_{ck\,\text{ordinal}} = \left(\sum_{g=c}^{k} n_g - \frac{n_c + n_k}{2} \right)^2 \tag{3.3}$$

$$\delta^2_{ck\,\text{intervall}} = (c - k)^2 \tag{3.4}$$

Für Intervall-skalierte Daten ist die Matrix δ^2_{metric} als die in den verwendeten Daten auftretenden Ausprägungen anzusehen, da eine vollständige Aufstellung bei stetigen Daten selbstverständlich nicht möglich ist.

3.2 Precision und Recall

Für die viele journalistische Fragestellungen ist nicht der gesamte Korpus von Bedeutung. Vielmehr sollen möglichst saubere Subkorpora erstellt werden, die dann nur noch die für die Fragestellung relevanten Texte enthalten. Unter der Bedingung, dass die Relevanz eines Textes durch menschliche Kodierer*innen bewerten werden (siehe Abschnitt 3.1), werden aus dem Korpus Stichproben gezogen, um mit geeigneten Schätzern die Qualität eines Subkorpus zu bewerten. Zwei geeignete Maßzahlen hierzu sind Precision und Recall. Während Precision den Anteil der relevanten Texte innerhalb des Subkorpus angibt, ist Recall die Maßzahl für den Anteil der relevanten Artikel, die durch den Subkorpus abgedeckt werden. Zieht man nun aus einer Grundgesamtheit zufällig Texte, lassen sich Precision und Recall für einen Subkorpus wie

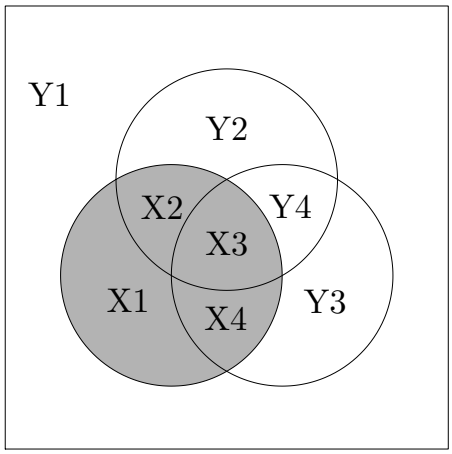

Abbildung 3.1: Beispiel für drei Subkorpora. Der interessierende Subkorpus (`X`) hat Überschneidungen zu den anderen beiden Subkorpora. Zur Berechnung von Precision und Recall werden die Schnittmengen `X1` bis `X4` zur Berechnung der Anteile im Subkorpus verwendet. Die Schnittmengen `Y1` bis `Y4` liegen außerhalb des Subkorpus und werden nur im Nenner des Recall-Schätzers verwendet.

folgt berechnen:

$$\widehat{\text{Precision}} = \frac{\text{Anzahl der als relevant gelabelten Texte im Subkorpus}}{\text{Anzahl der gelabelten Texte im Subkorpus}}$$
$$\widehat{\text{Recall}} = \frac{\text{Anzahl der als relevant gelabelten Texte im Subkorpus}}{\text{Anzahl aller als relevant gelabelten Texte}}.$$

Werden verschiedene Subkorpora bezüglich der beiden Maßzahlen verglichen, können die Überschneidungen der Subkorpora als Venn-Diagramm visualisiert werden (siehe Abbildung 3.1). Das Diagramm verdeutlicht, dass gelabelte Daten, die aus einem Subkorpus gezogen wurden, zum Beispiel um einen Schätzer für die Precision zu erhalten, zur Berechnung für die Maßzahlen anderer Subkorpora verwendet werden können. Da in den einzelnen Schnittmengen nicht zwingend gleiche Auswahlwahrscheinlichkeiten vorliegen (je nachdem aus welchen Grundgesamtheiten die Texte gezogen wurden), müssen die Schätzer für Precision und Recall so angepasst werden, dass sie die unterschiedliche Gewichtung der Schnittmengen berücksichtigen.

Sei dazu $\pi_i \in [0,1]$ der Anteil der für die journalistische Fragestellung relevanten Texte in Schnittmenge i und $w_i \in [0,1]$ der Anteil der Texte im Gesamtkorpus, die in die Schnittmenge i fallen. Die Schnittmengen, die zu dem interessierenden Subkorpus gehören, werden durch die Indexmenge X gekennzeichnet, die übrigen befinden sich in der Indexmenge Y (siehe dazu auch Abbildung 3.1). Der Anteil der Texte in der Indexmenge X, die sich in der Schnittmenge i befinden, werden mit $\tilde{w}_i \in [0,1]$ bezeichnet. Die Schätzer für Precision und Recall können nun als gewichtete Schätzer aus den π_i berechnet werden:

$$\widehat{\text{Precision}} = \sum_{i \in X} \tilde{w}_i \hat{\pi}_i \tag{3.5}$$

$$\widehat{\text{Recall}} = \frac{\sum_{i \in X} w_i \hat{\pi}_i}{\sum_{i \in X \cup Y} w_i \hat{\pi}_i} \tag{3.6}$$

Die Berechnung über die Schnittmengen bietet weitere Vorteile. Texte, die aus unterschiedlichen Teilmengen, z.B. unterschiedlichen Subkorpora gezogen wurden, und dementsprechend unterschiedliche Auswahlwahrscheinlichkeiten haben, je nach dem, in wie vielen Subkorpora der jeweilige Text enthalten ist, können kombiniert werden, da die Auswahlwahrscheinlichkeit für alle Texte innerhalb einer Schnittmenge gleich ist. Ein weiterer Vorteil zeigt sich bei der Berechnung des Recall. Da in vielen Anwendungsfällen davon auszugehen ist, dass die meisten relevanten Texte in mindestens einem der Subkorpora enthalten sind, können die Anteile der relevanten Texte außerhalb des interessierenden Subkorpus besser geschätzt werden, da die relevanten Texte durch die Teilmengen, die sich nicht im interessierenden Subkorpus befinden, besser von der großen Masse der nicht interessierenden Texte getrennt werden können. Während Stryker u. a. (2006) hier noch das Finden einer Obermenge durch eine möglichst allgemeine Wortliste vorsehen, um den Anteil der relevanten Texte außerhalb des interessierenden Subkorpus zu finden, kann hier einfach die Grundgesamtheit der verschiedenen Filterregeln kombiniert werden.

Die Varianz des Precision-Schätzers lässt sich direkt aus den $\hat{\pi}_i$ berechnen. Dabei ist n_i die Zahl der in Schnittmenge i codierter Texte. Die Berechnung der Varianz für den Recall-Schätzers lässt sich aufgrund des Quotienten nicht direkt berechnen, hier kann aber eine Näherung berechnet werden. Die Herleitung hierzu befindet sich im

Anhang in Abschnitt A.1.

$$\widehat{\mathrm{Var}}\left(\widehat{\mathrm{Precision}}\right) = \sum_{i \in X} \tilde{w}_i^2 \frac{1}{n_i} \hat{\pi}_i \left(1 - \hat{\pi}_i\right) \tag{3.7}$$

$$\widehat{\mathrm{Var}}\left(\widehat{Recall}\right) \approx \left(\frac{\sum_{i \in Y} w_i \hat{\pi}_i}{\left(\sum_{i \in X \cup Y} w_i \hat{\pi}_i\right)^2}\right)^2 \sum_{i \in X} w_i^2 \frac{1}{n_i} \hat{\pi}_i \left(1 - \hat{\pi}_i\right) \tag{3.8}$$

$$+ \left(\frac{\sum_{i \in X} w_i \hat{\pi}_i}{\left(\sum_{i \in X \cup Y} w_i \hat{\pi}_i\right)^2}\right)^2 \sum_{i \in Y} w_i^2 \frac{1}{n_i} \hat{\pi}_i \left(1 - \hat{\pi}_i\right) \tag{3.9}$$

3.2.1 Ziehen von weiteren Stichproben

Soll die Varianz der Schätzer für Precision und Recall verringert werden, muss die Zahl der gelabelten Texte erhöht werden. Da das Kodieren von Texten zeitaufwendig ist, kann oft nur eine begrenzte Menge von m weiteren Texten gelabelt werden. Statt die neuen Texte zufällig aus dem Korpus zu ziehen, ist es sinnvoll, besonders in den Schnittmengen Texte zu ziehen, in denen die erwartete Varianzreduktion des Schätzers am größten ist. Für die Berechnung der Precision werden nur Texte benötigt, die aus dem interessierenden Subkorpus stammen. Zur Berechnung des Recalls hingegen werden alle Schnittmengen im Korpus benötigt. Um nun eine optimierte Aufteilung der zu ziehenden Texte festzulegen ist es sinnvoll, den Recall für die Berechnung der optimalen Aufteilung zu Grunde zu legen. So werden potentiell alle Schnittmengen in die Optimierung mit einbezogen. Im Folgenden soll nun die Varianz des Recall-Schätzers minimiert werden. Für m neue Texte, die auf die Schnittmengen aufgeteilt werden sollen, sodass $\sum_i m_i = m$, kann für jede Schnittmenge i die Anzahl der zu ziehenden Texte wie folgt berechnet werden. Dabei sei I die Indexmenge für die von Null verschiedenen m_i: $I = m_i : m_i > 0$.

$$m_i \stackrel{!}{=} \begin{cases} \frac{\sqrt{c_i}\left(\sum_j n_j + m\right)}{\sum_I \sqrt{c_j}} - n_i & m_i \neq 0 \\ 0 \end{cases} \tag{3.10}$$

Dabei ist $c_i = b_i w_i^2 \hat{\pi}_i(1-\hat{\pi}_i)$ die gewichtete Varianzkomponente aus der Varianz für den Recall und es gilt $b_i = \left(\frac{\sum_{i \in Y} w_i \hat{\pi}_i}{\left(\sum_{i \in X \cup Y} w_i \hat{\pi}_i\right)^2}\right)^2$ wenn die Schnittmenge i im interessierenden Subkorpus liegt und andernfalls $b_i = \left(\frac{\sum_{i \in X} w_i \hat{\pi}_i}{\left(\sum_{i \in X \cup Y} w_i \hat{\pi}_i\right)^2}\right)^2$. In die Summe der c_j im Nenner gehen nur diejenigen c_j ein, für die m_j nicht Null ist. Um zu entscheiden welche m_i Null sind, müssen alle möglichen Kombinationen aus Null und nicht-Null Schnittmengen bezüglich ihrer zu erwartenden Varianzminimierung berechnet werden. Kombinationen in denen mindestens ein m_i negativ ist, werden ausgeschlossen. Der Beweis zu dieser Formel befindet sich im Anhang in Abschnitt A.2.

Da bei mehreren Subkorpora die Zahl der Schnittmengen und damit die Zahl der möglichen Kombinationen sehr groß werden kann, kann auch ein Ansatz genutzt werden, der in Simulationen das Optimum oder zumindest einen sehr nahen Wert gefunden hat (siehe dazu Kapitel 5): Ausgehend von der Aufteilung, in der alle m_i ungleich Null sind, wird iterativ immer das m_i auf Null gesetzt, das den kleinsten Wert besitzt. Dieses Vorgehen wird so lange wiederholt, bis keines der m_i mehr negativ ist. Existieren in der ursprünglichen Aufteilung keine negativen Werte, werden diese unverändert verwendet.

Für die praktische Nutzung werden optimale Werte so auf ganze Zahlen gerundet, dass in der Summe die m zu ziehenden Texte aufgeteilt werden. Als Online-Variante kann so auch, statt direkt eine Aufteilung für m Texte zu berechnen, immer nur die Schnittmenge für den nächsten Text berechnet werden ($m = 1$). Dieser gelabelte Text kann dann im nächsten Schritt in die Berechnung für den nächsten Text aufgenommen werden. Insbesondere bei diesem Vorgehen, aber auch bei den anderen Strategien, kann die Schnittmenge auch mit den auf Summe Eins normierten m_i als Auswahlwahrscheinlichkeit gezogen werden. So ist der Algorithmus nicht deterministisch und für menschliche Kodierer*innen ist es schwerer vorherzusagen aus welcher Schnittmenge der nächste Text gezogen wird. Dies vermindert das Risiko einer Verzerrung durch menschliche Kodierer*innen.

3.3 Textmining

Die Daten mit denen Statistiker*innen in der Regel arbeiten, liegen zumeist mehr oder weniger sortiert als Datensatz vor. Einzelne Variablen enthalten Zahlen oder Kategoriensysteme, die dann mit der Vielzahl der statistischen Methoden ausgewertet werden können. Im Textmining existieren solche Variablen häufig auch: Verschlagwortungen, Datumsangaben oder Textlängen können als klassische Variablen aufgefasst werden. Die hauptsächliche Information soll allerdings aus den im Datensatz enthaltenen Texten extrahiert werden. In diesem Abschnitt werden Methoden vorgestellt, mit denen die in dieser Arbeit behandelten Fragestellungen untersucht werden können.

3.3.1 Vorverarbeitung

Um einen Text mit statistischen Methoden auswerten zu können, muss dieser vorverarbeitet werden. Je nachdem was für eine Auswertung folgen soll, stehen verschiedene Methoden zur Verfügung (siehe auch Abschnitt 2.1). Für die Themenmodelle in dieser Arbeit wird nur eine sehr einfache Vorverarbeitung gewählt. Dabei wird die **bag-of-words**-Annahme (Miner u. a., 2012, S. 45) getätigt. Dieser Ansatz geht davon aus, dass die Information eines Texts in den Wörtern an sich und nicht in ihrer Position im Text enthalten ist. Die Wortfolge „Hund beißt Mann“ ist demnach gleichbedeutend mit der Wortfolge „Mann beißt Hund“. Diese zugegebenermaßen vereinfachende Annahme führt in der Praxis der Themenmodelle allerdings zu zufriedenstellenden Ergebnissen. Für das Verständnis einzelner Sätze ist sie aber nicht geeignet, wie das Beispiel zeigt. Inzwischen wird das Konzept des bag-of-words auch in anderen Bereichen, wie zum Beispiel der Bilderkennung, verwendet (Fei-Fei und Perona, 2005). Auch werden neben den eigentlichen Worten auch weitere Metadaten, wie Datumsangaben oder Verschlagwortung den Modellen zugefügt. Der Vorteil der bag-of-words-Annahme ist, dass Texte so als ein Vektor von Worthäufigkeiten angesehen werden können. Dieses **vector-space-model** kann nun genutzt werden um Texte anhand ihrer enthaltenen Worthäufigkeiten zu beschreiben.

Um einen Text auf Wortebene analysieren zu können, muss zuerst geklärt werden, wie genau ein Text in einzelne Wörter aufgeteilt wird. In einem ersten Schritt werden dazu alle Zeichen aus dem Text entfernt, die nicht zu einem Wort gehören. In der Regel sind dies Satzzeichen und Zahlen, die alleinstehend keine Aussagekraft haben und so für eine bag-of-words-Analyse nicht nützlich sind. Insbesondere in der deutschen Sprache ist dabei darauf zu achten, wie mit Bindestrich-Wörtern umgegangen wird. In dieser Analyse wird der Bindestrich entfernt und beide Wortteile als ein Wort betrachtet. Großbuchstaben werden in Kleinbuchstaben umgewandelt, um so Wörter am Satzanfang nicht anders zu behandeln als Wörter im Satz. Dies führt allerdings auch dazu, dass einige Nomen nicht mehr von Verben unterschieden werden können (Der Rasen, die Autos rasen). Da die im Projekt verwendeten Datenquellen unterschiedlich mit deutschen Umlauten umgehen, werden ä, ö, ü und ß in ae, oe, ue und ss umgewandelt. Andere nicht-ASCII-Zeichen werden nicht verändert. Als Wort wird nun jede Zeichenkette verstanden, die durch ein leerzeichenartiges Zeichen (alle Arten von Leerzeichen, Tab, ...) getrennt ist. Dies führt dazu, dass Entitäten (Personen-, Organisationsnamen, etc.) nicht als ein Wort aufgefasst werden. Für diese Analyse wurde darauf verzichtet, die Wörter durch Lemmatisierung oder Stemming zusammenzufassen, da die Stämme schwer zu interpretieren sind und auch Wörter zusammengefasst werden, die keine inhaltliche Verbindung haben (vergleiche auch Maier u. a., 2018). Besonders häufige Wörter, die einigermaßen gleichmäßig in Texten enthalten sind (der, die, das, und, oder, ...) und gleichzeitig wenig Information enthalten, werden abschließend aus den Texten entfernt. Diese Stoppwörter können mit Hilfe von Maßzahlen wie tf-idf, einer Maßzahl, die sich aus der Multiplikation der term frequency, also der relativen Häufigkeit des Wortes im Korpus mit der inverse document frequency, also dem Inversen des Anteils der Dokumente in denen das Wort enthalten ist, identifiziert werden, oder es können etablierte Listen (zum Beispiel Martin Porter und Boulton, 2018) verwendet werden, wie in diesem Fall.

Für die weiteren Methoden werden folgende Begriffe definiert. Sei $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ eine Sammlung von M Dokumenten. Jedes Dokument $\mathbf{w}_i = \{w_1, w_2, \ldots, w_{N_i}\}$ besteht wiederum aus N_i Worten w_j. An dieser Stelle muss der Begriff „Wort“ etwas genauer spezifiziert werden. Zum einen beschreibt der Begriff ein spezifisches Wort an einer

spezifischen Stelle in einem Text. Hier wird auch der Begriff „Token“ verwendet. Zum anderen beschreibt „Wort“ auch ein Wort aus dem verwendeten Vokabular. Hier wird auch der Begriff „Type“ verwendet. Ein Token ist also ein bestimmtes Wort an einer bestimmten Stelle, ein Type ein Wort, das in Texten an verschiedenen Stellen verwendet werden kann. Das Vokabular $\mathbf{V} = \{v_1, v_2, \ldots, v_V\}$ besteht aus V Wörtern (Types), die die Grundlage für die Token w_j sind. Für die Realisierung eines bestimmten Wortes wird die Notation w^i verwendet, wobei $w^i = 1$ ist, wenn die Realisierung dem i-ten Wort des Vokabulars entspricht, und ansonsten Null ist. Die analoge Schreibweise gilt für ein bestimmtes Thema z^i.

3.3.2 Themenmodelle

Ein Ziel im Textmining ist es, Textsammlungen thematisch zu unterteilen. Eine weit verbreitete Methode sind Themenmodelle (Topic Models). Die meisten Modelle gehen dabei von der bag-of-words-Annahme aus, also davon, dass die Reihenfolge von Wörtern in einem Text keine Bedeutung hat. Nach einiger Vorverarbeitung (siehe Abschnitt 3.3.1) können die Texte im Korpus als Worthäufigkeitstabellen oder als Dokument-Wort-Matrix dargestellt werden. Erste Ansätze, wie die auf einer Singulärwertzerlegung der Dokument-Wort-Matrix basierende Latent Semantic Analysis (LSA, auch latent semantic indexing (LSI), Deerwester u. a., 1990), stammen aus den 90er Jahren. Hier können die einzelnen Faktoren der Singulärwertzerlegung interpretiert werden. Die Weiterentwicklung probabilistic LSA (pLSA, auch pLSI, Hofmann, 1999) nimmt für jedes einzelne Wort eine latente Klassenvariable an. Diese als Themen interpretierbaren Klassen bilden ein Mischungsmodell über multinomial verteilte Zufallsvariablen. Auf der Ebene des Korpus wird allerdings noch keine Verteilungsannahme über die Themenverteilungen getroffen, was dazu führt, dass die Zahl der Parameter linear mit der Zahl der Dokumente im Trainingsdatensatz steigt (vgl. Blei, Ng u. a., 2003). Erst mit der Latent Dirichlet Allocation (Blei, Ng u. a., 2003) wird ein vollständiges generatives Modell eingführt, das nun durch Dirichlet Prior ergänzt wird. Während die einfache LDA unüberwacht lernt und keine weiteren Metadaten der Texte beachtet, gibt es einige Erweiterungen, die aus dem unüberwachten Modell ein überwachtes machen. Hier sei zum Beispiel auf die supervised LDA (Mcauliffe und Blei, 2008)

und die labeled LDA (Ramage u. a., 2009) verwiesen, die Klassifikationsvariablen in das Modell aufnehmen, im Fall von supervised LDA eine Klasse pro Text, oder auch mehrere Klassen wie bei der labeled LDA. Andere Ansätze versuchen die LDA in ihren Modellanahmen näher an die Realität zu bringen, indem sie Korrelationsstrukturen zwischen den Themen zulassen (Correlated Topic Model, CTM, Blei und Lafferty, 2005) oder zeitliche Strukturen in den Themenverläufen modellieren (Blei und Lafferty, 2006). Auch an der Verbindung mit anderen Methoden, wie neuronalen Netzen wird gearbeitet (Wan u. a., 2012).

Latent Dirichlet Allocation

Das generative Modell der Latent Dirichlet Allocation (LDA, Blei, Ng u. a., 2003) ist ein hierarchisches Bayesianisches Modell und beschreibt den Prozess der Entstehung eines Dokuments. Es zeichnet sich dadurch aus, dass es für die Verteilung der Themen und die der Wörter als apriori-Verteilung Dirichlet-Verteilungen annimmt. Aus diesen Annahmen und den Daten kann die aposteriori-Verteilung bestimmt werden. Wie bereits erwähnt, werden hierbei vereinfachte Annahmen getroffen. Aufgrund der bag-of-words-Annahme wird die Syntax von Texten nicht beachtet. Ein Dokument entsteht nach diesem Modell in drei Schritten:

1. Ziehe die Länge des Dokuments (Anzahl der Wörter) $N \sim \text{Poison}(\xi)$
2. Ziehe die Themenverteilung des Dokuments aus der apriori-Verteilung $\theta \sim \text{Dir}(\alpha)$
3. Für jedes Wort w_n im Dokument:
    a) Ziehe ein Thema z_n aus der Themenverteilung des Dokuments $z_n \sim \text{Multinomial}(\theta)$
    b) Ziehe aus dem jeweiligen Thema ein Wort w_n aus der Wortverteilung $P(w_n|z_n, \beta)$ des Themas z_n

Jeder Text im Korpus ist also eine multinominal verteilte Mischung aus latenten Themen, wobei jedes Thema sich über eine Multinominalverteilung aller Wörter

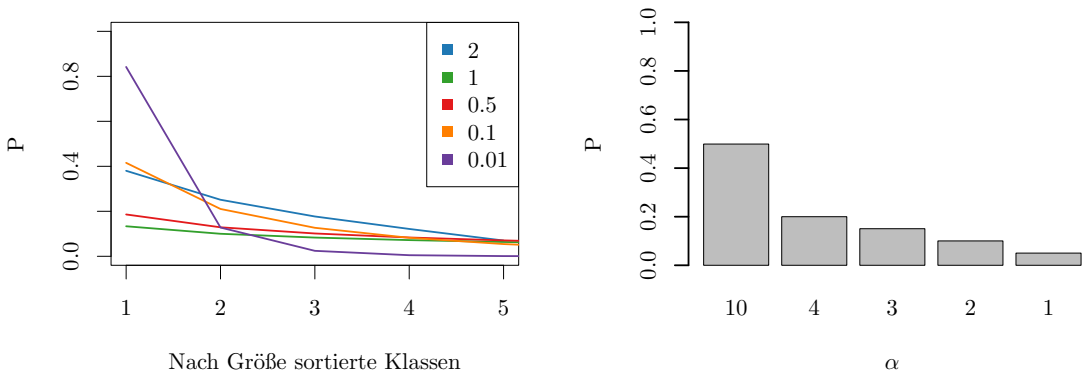

Abbildung 3.2: Verhalten der Dirichlet-Verteilung bei verschiedenen α-Parametern. In der linken Grafik wurden für eine fünfdimensionale Dirichlet-Verteilung jeweils alle α_i auf den gleichen Wert gesetzt. Abgebildet ist die Durchschnittliche Wahrscheinlichkeitsmasse auf der größten, zweitgrößten, ..., Kategorie. Für kleine Werte von α konzentriert sich die Wahrscheinlichkeitsmasse pro Realisation auf wenige Klassen. In der rechten Grafik ist eine durchschnittliche Wahrscheinlichkeitsverteilung für Dirichlet-verteilte Zufallszahlen mit dem Parameter $\alpha = (10, 4, 3, 2, 1)$ abgebildet. Die relative Größen der Werte für α entsprechen den relativen Anteilen an der Wahrscheinlichkeitsmasse. Simuliert wurden alle Werte mit 10 000 Simulationen.

definiert. Die Themenverteilung im zweiten Schritt wird aus einer Dirichlet-Verteilung mit Parametervektor $\alpha = \alpha_1, \ldots, \alpha_k;\ \alpha_i > 0\, \forall i$:

$$P(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma\left(\alpha_i\right)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1} \tag{3.11}$$

Dabei beschreibt Γ die Gamma-Funktion. Damit ist θ für jedes Dokument die Realisierung einer Dirichlet-Verteilung und damit ein Wahrscheinlichkeitsvektor der Länge k, wobei k der Anzahl der Themen im Modell entspricht. Dadurch ergibt sich auch, dass sich die Summe über die Themenwahrscheinlichkeiten zu Eins addiert: $\sum_{i=1}^{k} \theta_i = 1$. Der Parametervektor α legt hierbei fest wie viel Wahrscheinlichkeitsmasse im Mittel

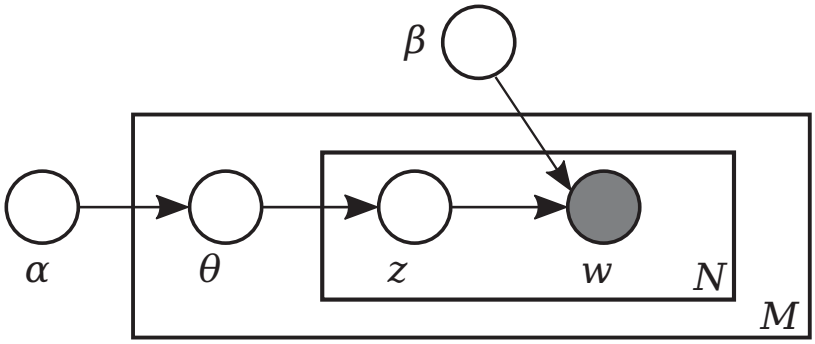

Abbildung 3.3: Grafische Darstellung der LDA. Die äußere Box stellt die M Dokumente des Korpus dar, die innere Box ein Dokument mit N Wörtern. Die Parameter α und β werden auf Korpus-Ebene definiert, der Parameter θ, der die Themen-Mischung pro Dokument festlegt, auf Dokumentenebene. Er wird aus einer Dirichlet-Verteilung mit Parameter α gezogen. Auf der Dokumentenebene wird für jedes der N Wörter ein Thema z multinomial mit Parameter θ gezogen. Für jedes so generierte Thema wird mit der zugehörigen Reihe der β-Matrix multinomial ein Wort aus der Wortverteilung des Themas gezogen.

auf die einzelnen Klassen entfällt (Verhältnis der α_i zueinander) und wie stark die Wahrscheinlichkeitsmasse für einzelne Realisierungen auf wenige Klassen konzentriert ist. Abbildung 3.2 zeigt diese Zusammenhänge. Für kleine Werte von α wird die Wahrscheinlichkeitsmasse auf wenige Klassen konzentriert, die Wahrscheinlichkeit für die Klasse j definiert sich durch $\alpha_j / \sum \alpha_i$. Die Dirichlet-Verteilung ist die konjugierte a-priori-Verteilung der Multinomialverteilung. Aus einer Realisierung einer Themenverteilung θ können nun multinomial Themen gezogen werden. Da jedes Dokument eine eigene Realisierung von θ erhält, ist die Mischung der Themen für jedes Dokument unterschiedlich. Für diese gezogenen Themen können nun Wörter aus der jeweiligen Wortverteilung des Themas gezogen werden. Auch die Wortverteilungen in einem Thema sind multinomialverteilt. Die notwendigen Parametervektoren für die Wortverteilungen sind in β in einer $k \times V$ Matrix zeilenweise zusammengefasst. Damit ist $\beta_{ij} = P(w_j = 1 | z_i = 1)$ die Wahrscheinlichkeit für das Wort w_j im Thema z_i. Abbildung 3.3 verdeutlicht das zugrunde liegende Modell noch einmal grafisch.

Die Wahrscheinlichkeit für ein Dokument aus der gemeinsamen Verteilung der The-

menverteilung θ kann mit latenten Themen $\mathbf{z}$ und Wörtern $\mathbf{w}$ dargestellt werden:

$$P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = P(\theta|\alpha) \prod_{n=1}^{N} P(z_n|\theta) P(w_n|z_n, \beta). \tag{3.12}$$

Die Wahrscheinlichkeit für die Wörter eines Dokument w ergibt sich als Randverteilung dieser gemeinsamen Verteilung durch herausintegrieren der Themenverteilung θ und der Themen z_n:

$$P(\mathbf{w}|\alpha, \beta) = \int P(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} P(z_n|\theta) P(w_n|z_n, \beta) \right) d\theta. \tag{3.13}$$

Die Wahrscheinlichkeit für den Gesamtkorpus ergibt sich abschließend aus dem Produkt der Dokumentenwahrscheinlichkeiten:

$$P(D|\alpha, \beta) = \prod_{d=1}^{M} \int P(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} P(z_n|\theta_d) P(w_{dn}|z_{dn}, \beta) \right) d\theta_d. \tag{3.14}$$

Gibbs Sampler

In der Praxis müssen für einen bestehenden Korpus die Themen-Zuordnungen z_i der einzelnen Token geschätzt werden. Da die Likelihood nicht analytisch lösbar ist, müssen andere approximative Schätzverfahren verwendet werden. Blei, Ng u. a. (2003) verwenden dazu einen Variational Bayes Algorithmus, in dieser Arbeit wird ein Gibbs Sampler (Griffiths und Steyvers, 2004) verwendet, der im R-Paket lda implementiert ist. Eine Übersicht über verwendete Verfahren liefern Asuncion u. a. (2009), dort wird auch die Ähnlichkeit der einzelnen Verfahren insbesondere unter der Optimierung der Hyperparameter beschrieben.

Für die Anwendung des Gibbs Sampler nach Griffiths und Steyvers (2004) wird der oben beschriebenen LDA ein Dirichlet-Prior für die Wortverteilung in den Themen hinzugefügt. Der Gibbs Sampler benötigt die Verteilungen der Zufallsvariablen bedingt auf alle anderen. Für die LDA ergeben sich folgende Wahrscheinlichkeitsverteilungen:

$$w_{dn}|z_{dn}, \beta_{z_{dn}} \sim \text{Multinomial}\left(\beta_{z_{dn}}\right) \tag{3.15}$$

$$\beta \sim \text{Dirichlet}\left(\eta\right) \tag{3.16}$$

$$z_{dn}|\theta^d \sim \text{Multinomial}\left(\theta^d\right) \tag{3.17}$$

$$\theta \sim \text{Dirichlet}\left(\alpha\right). \tag{3.18}$$

Da bei der Initialisierung keine Informationen zu den einzelnen Themen vorliegen, werden hier nicht informative symmetrische Dirichlet-Prior verwendet. Die initiale Themenzuordnung der Token im Korpus erfolgt zufällig. Für optimierte Initialisierungen vergleiche Maier u. a. (2018).

Die gemeinsame Verteilung $P\left(w, z\right)$ lässt sich als $P\left(w|z\right) P\left(z\right)$ schreiben. Beide Wahrscheinlichkeiten sind jeweils nur von einem Hyperparameter abhängig:

$$P(w|z) = \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^k \prod_{j=1}^{k} \frac{\prod_w \Gamma(n_{j,w} + \beta)}{\Gamma(n_j + V\beta)} \tag{3.19}$$

$$P(z) = \left(\frac{\Gamma(k\alpha)}{\Gamma(\alpha)^k}\right)^M \prod_{d=1}^{M} \frac{\prod_j \Gamma(n_{j,d} + \alpha)}{\Gamma(n_d + k\alpha)} \tag{3.20}$$

Das j ist dabei der Index für das jeweilige Thema.

Die Markow-Kette iteriert über die einzelnen Token im Korpus. Die Zuordnung des Token w_i zu Thema j wird zufällig mit folgender Wahrscheinlichkeit bestimmt:

$$P\left(z_i = j|\mathbf{z}_{-i}, \mathbf{w}\right) \propto \frac{n_{-i,j}^{w_i} + \eta}{n_{-i,j}^{\cdot} + V\eta} \frac{n_{-i,j}^{d_i} + \alpha}{n_{-i}^{d_i} + k\alpha}. \tag{3.21}$$

Die Hyperparameter α und η stammen aus den Dirichlet-Prior der Themenverteilung beziehungsweise der Wortverteilung. Der erste Bruch beschreibt damit einen Schätzer

für die Wahrscheinlichkeit von w_i im Thema j, der zweite Bruch die Wahrschinlichkeit des Themas in dem zu w_i gehörigen Dokument.

In dieser Arbeit wird jeweils der letzte Status der Markow-Kette für die weiteren Berechnungen verwendet.

Repräsentative Wörter

Ein Thema in einem Themenmodell ist definiert durch seine Wahrscheinlichkeitsverteilung über alle Token. Für die Anwendung muss eine übersichtlichere Darstellung gewählt werden, mit der zum Beispiel das konkrete Thema bestimmt werden kann. Eine Darstellung der wahrscheinlichsten Token ist in der Regel nicht zielführend, da dies meist inhaltsarme Wörter sind, die in vielen Themen hohe Wahrscheinlichkeiten aufweisen. Als Stellvertreter für ein Thema sollen daher repräsentative Wörter gewählt werden. In dieser Arbeit werden dazu die top.topic.words (TTW) aus dem lda-Paket genutzt. Für ein Token w in einem Thema k' wird ein Score berechnet, der die Wahrscheinlichkeit $\beta_{w,k'}$ mit der Wahrscheinlichkeit des Token in den anderen Themen gewichtet:

$$\mathrm{TTW}(w, k') = \beta_{w,k'}(\log(\beta_{w,k'}) - \frac{1}{k}\sum_{i=1}^{k}\log(\beta_{w,i})). \tag{3.22}$$

Token, die für das Thema nicht spezifisch sind, erhalten so einen niedrigeren Score. Die Token mit den höchsten Scores können als Repräsentanten des Themas genutzt werden.

3.3.3 Modellwahl bei Latent Dirichlet Allocation

Um Latent Dirichlet Allocation auf einen Textkorpus anwenden zu können, müssen die drei Parameter α, η und k des Modells gewählt werden. Da ein reiner Maximum-Likelihood-Ansatz oft zu Ergebnissen führt, die der Anwender nicht als optimal ansieht,

schlagen Chang u. a., 2009 eine Validierung durch menschliche Kodierer*innen vor. Die von ihnen vorgeschlagenen Intruder Words beziehungsweise Intruder Topics werden in Abschnitt 4.3.3 vorgestellt. Durch den Einsatz von menschlichen Kodierer*innen und die damit verbundenen kapazitären Limitierungen muss die Zahl der möglichen Modelle im Vorfeld eingeschränkt werden. In dieser Arbeit wird dafür die Topic Coherence verwendet.

Topic Coherence

Die Topic Coherence (Mimno u. a., 2011) ist eine Maßzahl zur Themenvalidierung. Die Kohärenz eines Themas wird dabei anhand des paarweisen gemeinsamen Auftretens von häufigen Wörtern innerhalb des Themas gemessen. Für ein Thema z werden die häufigsten M Wörter paarweises auf ihr gemeinsames Auftreten in einem Text überprüft. Gezählt wird die Anzahl der Texte $D\left(v_m^{(z)}, v_l^{(z)}\right)$ in denen beide Wörter dem Thema z zugeordnet wurden. Ein mehrfaches Auftreten von Wörtern in einem Text wird dabei nicht beachtet. Die Häufigkeit des paarweisen Auftretens wird mit der Häufigkeit eines der beiden Wörter gewichtet. Die Topic Coherence C definiert sich durch die Summe der logarithmierten gewichteten Häufigkeiten:

$$C\left(z; V^{(z)}\right) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \left(\frac{D\left(v_m^{(z)}, v_l^{(z)}\right) + 1}{D\left(v_l^{(z)}\right)} \right). \tag{3.23}$$

Um einen Logarithmus über Null zu vermeiden, wird ein Pseudocount eingeführt. Dadurch sind in der Theorie positive Werte der Maßzahl möglich. In der Praxis ist die Maßzahl negativ. Bei einem besonders kohärenten Thema sollten die wichtigen Wörter in den zugeordneten Texten möglichst häufig zusammen auftauchen. Ein hoher Wert C ist also ein Zeichen für ein kohärentes Thema. Neben dem Vergleich einzelner Themen können auch ganze Modelle verglichen werden, indem die berechnete Topic Coherence für alle Themen eines Modells geeignet zusammengefasst werden. Näheres dazu in Kapitel 6.

In dieser Arbeit werden zusätzlich zwei Abwandlungen dieser Maßzahl betrachtet. Zum einen werden statt den häufigsten Wörtern eines Themas auch die repräsentativsten Wörter, wie im vorherigen Abschnitt beschrieben, verwendet, zum anderen wird die Topic Coherence so abgewandelt, dass das paarweise Auftreten eines Wortes durch die mittlere Häufigkeit beider Wörter gewichtet wird und so eine symmetrische Maßzahl entsteht:

$$C\left(t; V^{(t)}\right) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \left(\frac{D\left(v_m^{(t)}, v_l^{(t)}\right) + 1}{0.5 \cdot \left(D\left(v_l^{(t)}\right) + D\left(v_m^{(t)}\right)\right)} \right). \tag{3.24}$$

4 Tools for statistical content analysis (tosca)

Um die in Abschnitt 2.3 vorgestellten Korpora maschinell auswerten zu können, müssen diese vorverarbeitet werden. Je nach weiterer Analyse soll es möglich sein, zu jeder Zeit der Analyse auf verschiedene vorverarbeitete Texte zugreifen zu können, damit sowohl eine maschinelle als auch eine Auswertung durch einen Menschen (human-in-the-loop) möglich ist. In R gibt es eine Fülle von Paketen, die sich mit Textmining beschäftigen. Die CRAN Task View NLP[1] (natural language processing) gibt einen Überblick über die Vielfalt der Pakete. Als zentrale Pakete seien hier insbesondere tm (Feinerer u. a., 2008), das in den Sozialwissenschaften verbreitete quanteda (Benoit u. a., 2018) und tidytext (Silge und Robinson, 2016), das Textmining-Funktionalität für das „tidyverse“ zur Verfügung stellt, genannt.

Die Idee des R-Pakets tosca (Tools for statistical content analysis, Koppers u. a., 2021) ist es, nicht ein weiteres zentrales Textmining Paket zu etablieren, sondern vielmehr Werkzeuge zu bündeln, die bei einer Inhaltsanalyse typischerweise verwendet werden können. Dieses Paket ist im Rahmen der DoCMA-Kooperation in enger Zusammenarbeit von Statistikern und Journalist*innen entstanden. Neben der Vorverarbeitung der Korpora sind dies insbesondere die Visualisierung zur deskriptiven Auswertung und das Anwenden und Validieren von LDA-Modellen. In diesem Bereich fehlt den etablierten Paketen die Möglichkeit einfache grafische Auswertungen vorzunehmen. Zur Validierung von LDA-Modellen gibt es bereits Ansätze. Mit dem Paket topicmodels (Grün und Hornik, 2011) kann beispielsweise die Perplexity des Modells berechnet werden, das Paket lda (Chang, 2015) erstellt unterschiedlich gewichtete Topword-Listen und LDAvis (Sievert und Shirley, 2015) visualisiert zweidimensional die Größe der Themen und ihre Distanzen zueinander. Zusätzlich werden in diesem Paket auch die Worthäufigkeiten im Thema beziehungsweise im Korpus visualisiert. Diesen Methoden

[1] `https://CRAN.R-project.org/view=NaturalLanguageProcessing`

soll tosca weitere Analysemöglichkeiten hinzufügen, indem insbesondere der Themenverlauf über die Zeit auf verschiedene Weisen visualisiert werden kann. Zusätzlich bietet das Paket eine Implementierung von Changs „intruder topics“ und „intruder words“ (Chang u. a., 2009).

Um für alle Methoden und die menschlichen Kodierer*innen die passende Form der Texte vorhalten zu können, werden die Korpora in einer einheitlichen Form abgelegt, sodass der gleiche Datensatz auch in verschiedenen Stufen der Vorverarbeitung immer die gleiche Struktur aufweist. Dabei werden insbesondere für grundlegende Vorverarbeitungsschritte von Texten die Funktionen des Pakets tm (Feinerer u. a., 2008) verwendet. Als LDA-Implementierung wird das lda-Paket von Chang (2015) genutzt. In diesem Abschnitt wird die Grundfunktionalität von tosca vorgestellt, die in der Arbeit weiter verwendeten Subkorpora erstellt und vorverarbeitet. Eine Liste aller Funktionen im Paket bietet die Tabelle C.3 im Anhang.

4.1 Vorverarbeitung der Datensätze

Um Textcorpora mit tosca bearbeiten zu können, müssen sie als S3 `textmeta`-Objekt vorliegen. Ein `textmeta`-Objekt ist als Liste mit den folgenden drei Elementen definiert.

`meta` ist ein `data frame`, das die Metadaten zu den Texten enthält. Neben den notwendigen Variablen `id` (Eindeutiger Character-String), `date` (`date`-Objekt) und `title` (beliebiger Character-String), können beliebige andere Variablen ergänzt werden.

`text` ist eine Liste, deren Listeneinträge die einzelnen Texte repräsentieren. Die einzelnen Listenelemente sind mit den IDs der Texte benannt. Dabei ist es nicht zwingend notwendig, dass für jeden Metadaten Eintrag ein Text-Eintrag besteht, oder umgekehrt. Die einzelnen Texte können aus Character-Strings oder Character-Vektoren bestehen, je nach Vorverarbeitungsschritt.

metamult kann optional Metadaten enthalten, die als Multilabel vorliegen, z.B. eine Verschlagwortung, in der die Schlagwörter pro Text variieren können.

Die Struktur wird noch einmal an einem kleinen Beispiel-Datensatz deutlich, der aus drei Texten der Süddeutschen Zeitung besteht.

```
## List of 3
##  $ meta    :'data.frame': 3 obs. of  11 variables:
##   ..$ id           : chr [1:3] "A42695576" "A42695691" "A49189693"
##   ..$ date         : Date[1:3], format: "2008-09-15" "2008-09-15" ...
##   ..$ rubrik       : chr [1:3] "Politik" "Wirtschaft" "Geld"
##   ..$ page         : int [1:3] 1 17 26
##   ..$ AnzChar      : int [1:3] 911 4857 3071
##   ..$ AnzWoerter   : int [1:3] 122 673 453
##   ..$ dachzeile    : chr [1:3] NA "Dramatische R"| __truncated__ NA
##   ..$ title        : chr [1:3] "Letzte Chance"| __truncated__ "Letzte Chance"| __truncated__ "Letzte Chance"| __truncated__
##   ..$ zwischentitel: chr [1:3] NA NA NA
##   ..$ untertitel   : chr [1:3] "US-Regierung "| __truncated__ "Angeschlagene"| __truncated__ "Im  Sommer ve"| __truncated__
##   ..$ author       : chr [1:3] "dpa" "Von  Nikolaus Piper" "Von  Charlotte Theile"
##  $ text    :List of 3
##   ..$ A42695576: chr [1:2] "" "New York   Am"| __truncated__
##   ..$ A42695691: chr [1:11] "" "" "New York US-F"| __truncated__ "Die Verhandlu"| __truncated__ ...
##   ..$ A49189693: chr [1:8] "" "" "Muenchen     "| __truncated__ "Wer Lehman-Ze"| __truncated__ ...
##  $ metamult: NULL
##  - attr(*, "class")= chr "textmeta"
```

Wie generell bei der Analyse von Daten ist es bei der Analyse von Textkorpora notwendig, in einem ersten Schritt die Qualität der Daten zu beurteilen. Da die hier vorliegenden Daten direkt vom Verlag kommen, wäre eine gute Qualität im Sinne von Vollständigkeit, einheitlicher Formatierung und einheitlichem Encoding zu erwarten. Die Erfahrung von DoCMA mit verschiedenen Verlagshäusern zeigt jedoch, dass die Datenbanken häufig nicht so gepflegt werden, wie es für eine automatische Analyse wünschenswert ist. Neben einer uneinheitlichen Kategorisierung oder Verschlagwortung, insbesondere über längere Zeiträume, ist insbesondere die Existenz von Dubletten ein Problem, da einzelne Texte so in der Auswertung höher gewichtet werden. In Zeitungs-Korpora entstehen Dubletten oft durch verschiedene Ausgaben, bei denen Texte auf Grund von Lokalausgaben, oder Früh- und Spätausgaben, mehrfach in die Datenbanken einfließen. Der in tosca implementierte erste Schritt ist das Identifizieren von Duplikaten. Da im vorliegenden Datensatz keine ID mehrfach vergeben ist, kann die Suche nach Duplikaten direkt über die Funktion `duplist` erfolgen.

```
SZduplist <- duplist(SZ, paragraph = TRUE)
```

Die Funktion durchsucht den Korpus nach Text-Duplikaten. In einem zweiten Schritt werden innerhalb der Text-Duplikate Texte mit gleichen Metadaten (mit Ausnahme der ID) gesucht. Insgesamt existieren im Korpus 1 760 921 Einträge. Davon existieren 1 585 364 nur einmal. Bei 175 557 Einträgen existieren mindestens zwei identische Texte. Insgesamt gibt es 48 652 einzelne Texte, die mehrfach im Datensatz enthalten sind. In dieser Menge enthalten sind auch die 45 963 Versionen, bei denen nicht nur die Texte, sondern auch die Metadaten (mit Ausnahme der ID) mehrfach im Korpus vorhanden sind. Nimmt man jeden Text nur einmal in den Korpus auf, verbleiben 1 634 016 Texte im Korpus.

Zur weiteren Untersuchung der Duplikate hilft ein erster Blick auf die Verteilung der Duplikate über die Zeit. Abbildung 4.1 zeigt den Anteil der Duplikate über die Zeit. Auffällig dabei ist, dass zwischen September 2002 und September 2007 der Anteil an Duplikaten deutlich zunimmt. Betrachtet man die absoluten Zahlen abzüglich der Duplikate (untere Grafik in Abbildung 4.1), stellt man fest, dass die Zahl der Artikel ohne Duplikate kaum schwankt. Dies deutet darauf hin, dass in dieser Zeit Texte doppelt abgespeichert wurden, die außerhalb dieses Zeitraums anders behandelt wurden. Der Vergleich der Duplikate innerhalb und außerhalb des Zeitraums zeigt neben der unterschiedlichen Häufigkeit keine weiteren Auffälligkeiten. Insgesamt fällt auf, dass bei sehr vielen Duplikaten alle Artikel am gleichen Tag veröffentlicht wurden. Es gibt bei 94.99 % der Duplikate nur Texte vom gleichen Tag. Bei den verbleibenden 2437 Artikeln sind 997 kürzer als 50 Wörter und damit keine Artikel im eigentlichen Sinne, sondern Bildunterschriften, Infozeilen und Ähnliches. Damit lässt sich rechtfertigen, nur einen Text im Korpus zu belassen und das Veröffentlichungsdatum dabei nicht zu berücksichtigen. Dieses Fallbeispiel zeigt deutlich, dass das Dublettenproblem im Gegensatz zu den Aussagen von Lemke und Wiedemann (2016) auch relevant sein kann, wenn die Daten direkt vom Verlag kommen.

Neben Duplikaten enthält der Korpus Einträge, die nur sehr kurz sind, weil sie keine Zeitungsartikel im eigentlichen Sinne sind. In diese Kategorie fallen zum Beispiel die Bildunterschriften von alleinstehenden Bildern (zum Beispiel Karikaturen) und sehr kurze Ankündigungen (Fernsehprogramm, Verweis auf einen Artikel im Blatt). Während der Anteil der Duplikate gleichmäßig über die unterschiedlich langen Texte

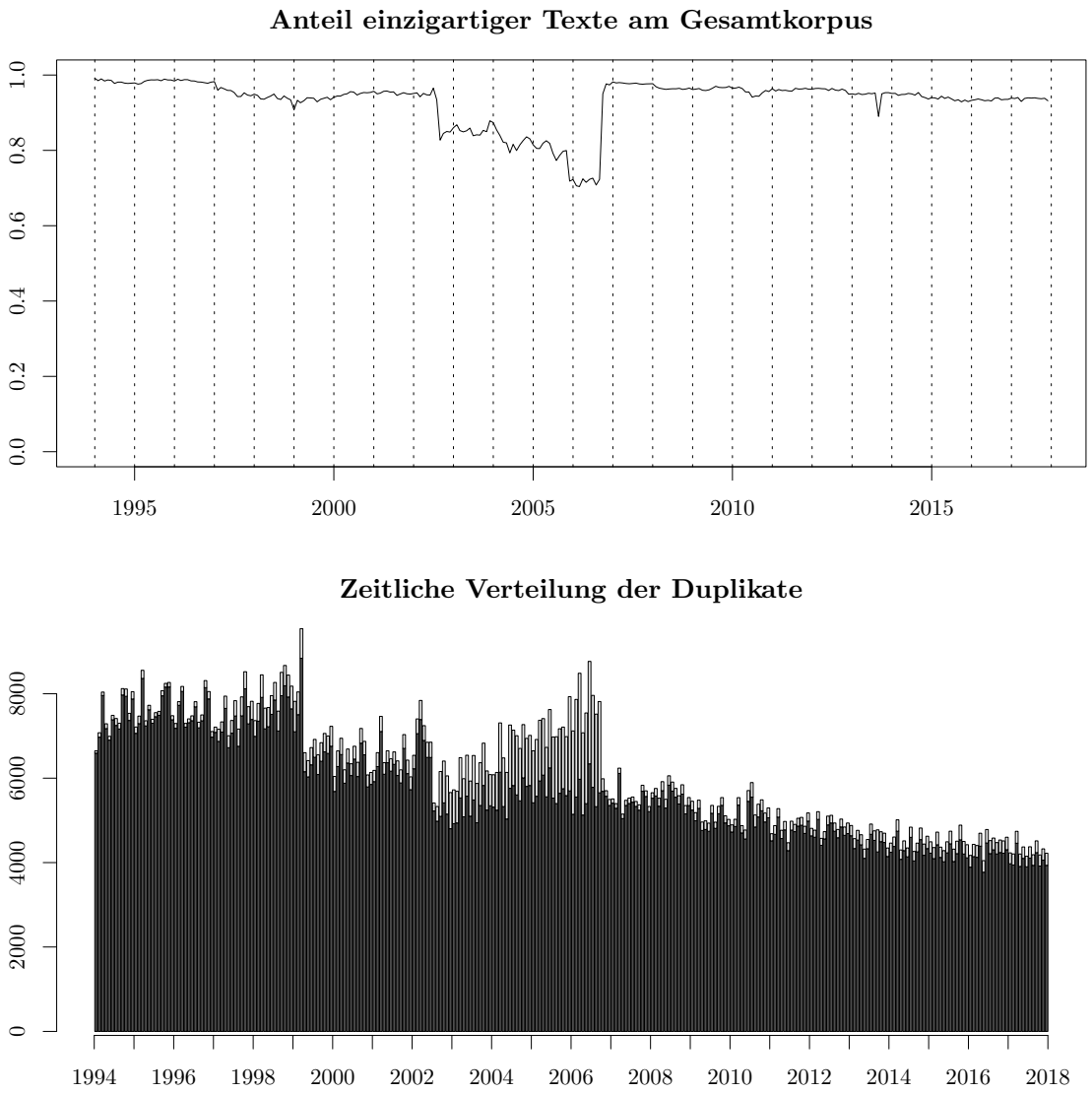

Abbildung 4.1: Zeitliche Verteilung der Duplikate. Die obere Grafik zeigt den Anteil der einzigartigen Texte über die Zeit inklusive jeweils eines Textes der Duplikate. In der unteren Grafik sind wie in Abbildung 2.1 die Artikel pro Monat abgetragen, wobei der helle Balken Texte beschreibt, die bereits in den dunklen Balken enthalten sind (Duplikate). Bei einem Duplikat wird also eine Kopie als einzigartiger Text gewertet.

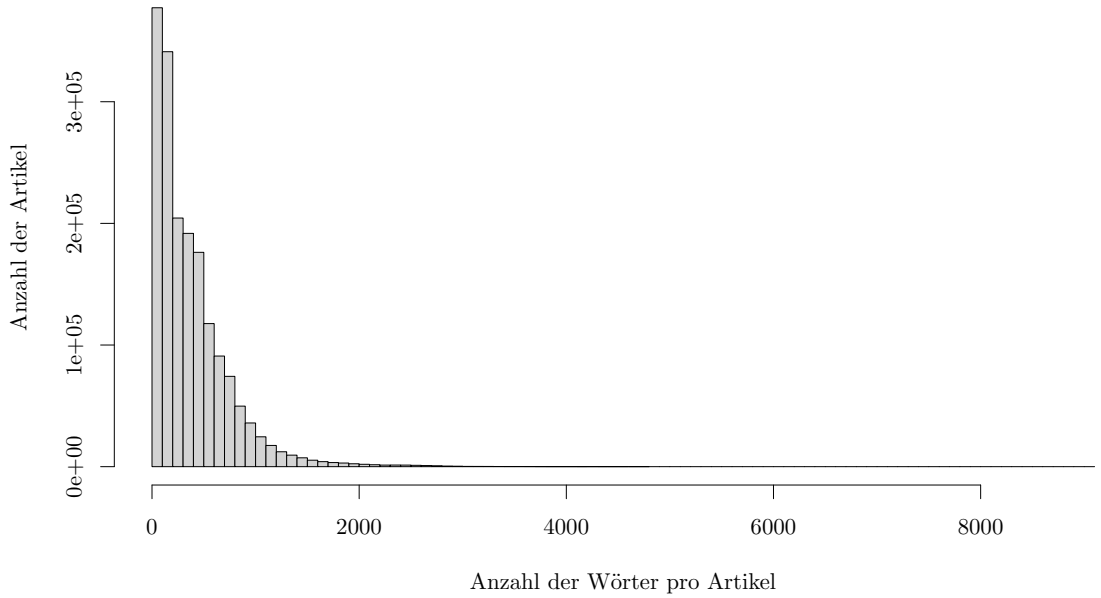

Abbildung 4.2: Histogramm über die Textlängen in der SZ

verteilt ist (siehe Abbildung B.2 im Anhang), ist es sinnvoll, sehr lange und sehr kurze Texte genauer zu betrachten um die Datenqualität insgesamt zu verbessern.

Abbildung 4.2 zeigt die Verteilung der Textlängen. Während die meisten Artikel deutlich weniger als 2000 Wörter lang sind (im Median sind es 276 Wörter pro Artikel), gibt es einige sehr lange Texte. Tabelle 4.1 listet alle Artikel auf, die mehr als 5500 Wörter lang sind. Unter den zwölf Texten sind drei Texte, die ausschließlich Wahlergebnisse enthalten (Nummer 1, 5 und 8) und Texte mit der Überschrift „Aktuelles in Zahlen“. Artikel mit dieser Überschrift sind auch in den weiteren Artikeln mit vielen Wörtern sehr stark vertreten und enthalten Sportergebnisse in Aufzählungsform. Insgesamt existieren mit dieser Überschrift 7883 Artikel. Diese und die Wahlergebnisse werden aus dem Datensatz entfernt, da sie keinen Text im eigentlichen Sinne enthalten, durch ihre Größe aber in weiteren Analysen ein starkes Gewicht bekommen können. Die übrigen langen Artikel sind oft Interviews, aber auch der Koalitionsvertrag der Bundesregierung von 1998 ist dabei. Diese Texte werden im Datensatz belassen.

	Datum	Titel	Anzahl der Wörter
1	1994-10-18	Ergebnisse aus den Wahlkreisen	9080
2	1996-05-06	„Ein Stueck Welt ist kaputtgegangen“	6812
3	2015-11-21	„Ich haette alles gemacht“	6786
4	2015-04-25	Hoffnung (Fortsetzung von Seite 11)	6469
5	1998-09-29		6283
6	1998-10-21	„Aufbruch und Erneuerung der Vertrag“	5706
7	2015-02-02	AKTUELLES IN ZAHLEN	5594
8	1998-09-29		5579
9	2015-02-09	AKTUELLES IN ZAHLEN	5566
10	2009-01-26	Aktuelles in Zahlen	5555
11	2016-02-13	Eure Feigheit kotzt mich an (Fortsetzung)	5547
12	2015-01-26	AKTUELLES IN ZAHLEN	5514

Tabelle 4.1: Liste der längsten Artikel im SZ Korpus. Die beiden Texte ohne Titel enthalten die auf zwei Texte aufgeteilten Wahlergebnisse der Bundestagswahl 1998.

Der so bereinigte Datensatz enthält nun 1 628 031 Artikel. Da das Entfernen von zu kurzen Texten damit begründet ist, dass sie nicht ausreichend Information für die spätere Analyse enthalten, ist es sinnvoll, zuerst die Texte vorzuverarbeiten und erst im Anschluss besonders kurze Texte zu entfernen. `tosca` bietet mit `cleanTexts` einen Wrapper für zentrale Textverarbeitungsschritte.

```
SZclean <- cleanTexts(object=SZ, sw = "de", paragraph = TRUE,
                      lowercase = TRUE, rmPunctuation = TRUE,
                      rmNumbers = TRUE, checkUTF8 = TRUE, ucp=TRUE)
```

Das Vorgehen wird hier exemplarisch an einem Beispieltext aus dem Korpus gezeigt. Der Text „Letzte Chance für Lehman-Bank“ mit der Unterüberschrift „US-Regierung will Kollaps des Unternehmens verhindern“ ist am 15.09.2008 in der Süddeutschen Zeitung erschienen. Beim Einlesen der Texte wurden bereits HTML-Tags (z.B. `<B> </B>`) entfernt und die ursprünglich als HTML-entities geschriebenen deutschen Umlaute in ihre ASCII-Umschreibungen überführt (ä → ae). Andere HTML-Entities von nicht-ASCII-Zeichen wurden in ihre UTF-8 Zeichen umgewandelt. Die abweichende

Verarbeitung der deutschen Umlaute beruht darauf, dass im Projekt auch mit Daten gearbeitet wird, bei denen die deutschen Umlaute bereits in den Rohdaten umgeschrieben sind und so die anderen Korpora daran angeglichen wurden. Die für den menschlichen Betrachter verwendete Version des Textes sieht damit wie folgt aus:

New York Amerikanische Grossbanken und die US–Regierung haben am Wochenende um die Rettung der schwer angeschlagenen Investmentbank Lehman Brothers gekaempft. Sie befuerchten, ein Kollaps des Traditionshauses koennte die Finanzbranche weltweit stark belasten. Medienberichten zufolge soll die Bank gespalten werden. Als moegliche Kaeufer gelten die Bank of America und die britische Barclays Bank. Der urspruenglich geplante Verkauf der gesamten Bank an einen Konkurrenten erwies sich als schwierig, weil niemand im Alleingang und ohne Hilfe Washingtons Lehman uebernehmen wollte. Finanzminister Henry Paulson hatte aber klargemacht, dass aus der Staatskasse diesmal kein Geld zu erwarten sei. Auch die Investmentbank Merrill Lynch, der groesste US–Versicherer AIG und die Sparkasse Washington Mutual stehen unter Druck. (Wirtschaft)

Um die Texte mit Textmining-Methoden besser verarbeiten zu können, werden die Texte mit `cleanTexts` verarbeitet. Dazu werden alle Großbuchstaben in ihre jeweiligen Kleinbuchstaben umgewandelt. Zahlen und Sonderzeichen werden entfernt. In einem weiteren Schritt werden Stoppwörter entfernt. Hier wird die Snowball-Liste (Martin Porter und Boulton, 2018) verwendet, wie sie im `tm`-Paket implementiert ist. Die Liste wurde in `cleanTexts` um die neue Rechtschreibung („daß“ und „dass“) und um umschriebene Umlaute ergänzt. Die gesamte Liste befindet sich im Anhang in Tabelle C.2. Der Beispieltext sieht nach dieser Vorverarbeitung wie folgt aus:

new york amerikanische grossbanken usregierung wochenende rettung schwer angeschlagenen investmentbank lehman brothers gekaempft befuerchten kollaps traditionshauses

finanzbranche weltweit stark belasten medienberichten zufolge bank gespalten moegliche kaeufer gelten bank of america britische barclays bank urspruenglich geplante verkauf gesamten bank konkurrenten erwies schwierig niemand alleingang hilfe washingtons lehman uebernehmen finanzminister henry paulson klargemacht staatskasse diesmal geld erwarten sei investmentbank merrill lynch groesste usversicherer aig sparkasse washington mutual stehen druck wirtschaft

Dies ist die zweite Version der Texte mit denen in dieser Arbeit gearbeitet wird. Gespeichert wird diese Textversion in einem an den Leerzeichen tokenisierten Zustand, sodass jedes Token einen eigenen Eintrag im Character-Vektor enthält. Aufgrund der bag-of-words-Annahme z.B. der LDA ist diese Darstellung für viele weitere Analysen gleichbedeutend mit der sortierten Liste der Wörter:

aig alleingang america amerikanische angeschlagenen bank bank bank bank barclays befuerchten belasten britische brothers diesmal druck erwarten erwies finanzbranche finanzminister gekaempft geld gelten geplante gesamten gespalten groesste grossbanken henry hilfe investmentbank investmentbank kaeufer klargemacht kollaps konkurrenten lehman lehman lynch medienberichten merrill moegliche mutual new niemand of paulson rettung schwer schwierig sei sparkasse staatskasse stark stehen traditionshauses uebernehmen urspruenglich usregierung usversicherer verkauf washington washingtons weltweit wirtschaft wochenende york zufolge

Die nur leicht vorverarbeiteten Rohtexte und die nicht sortierten bereinigten Texte bilden die Grundlage für alle weiteren Analysen. Weitere Vorverarbeitung beschränkt sich dabei auf das Filtern von Texten um kleinere und spezifischere Korpora zu erzeugen und die interne Darstellung in R. In Abbildung 4.3 ist der Effekt der Stoppwort-Entfernung

gut zu erkennen. In der linken Grafik ist das Histogramm über die Textlängen für Texte mit weniger als 1500 Wörtern, so wie sie im Datensatz angegeben sind, dargestellt, in der rechten Grafik der gleiche Wertebereich für die vorverarbeiteten Texte. Durch die feste Grenze am rechten Rand werden in der rechten Grafik mehr Texte betrachtet, da diese unverarbeitet zum Teil länger als 1500 Wörter lang sind. Das Histogramm über die gesamten Textlängen des bereinigten Datensatzes ist in Abbildung B.3 im Anhang zu finden.

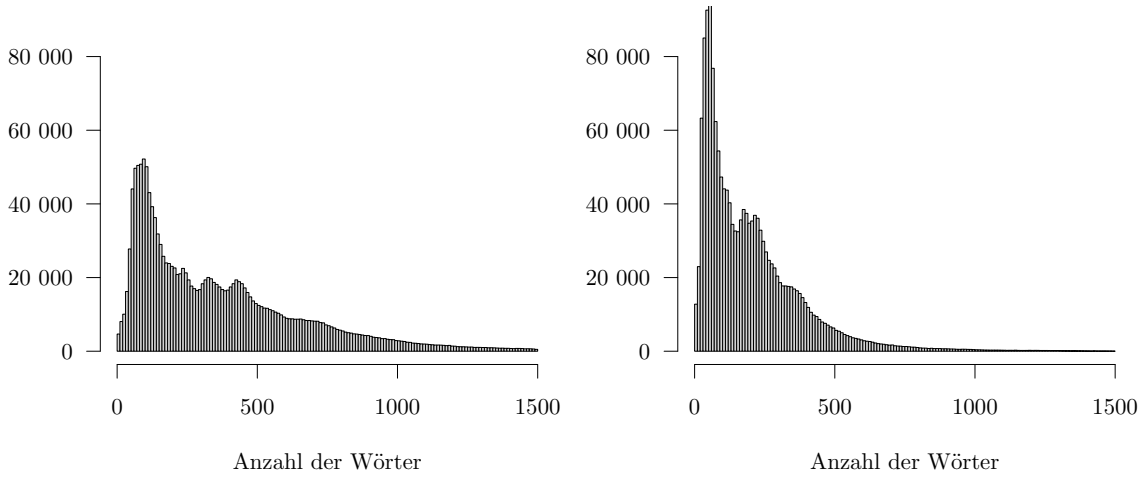

Abbildung 4.3: Histogramm über die Textlängen vor (links) und nach (rechts) der Vorverarbeitung. Die Daten sind am rechten Rand trunkiert, sodass nur Texte mit jeweils weniger als 1500 Wörtern betrachtet werden.

Um kurze Texte zu filtern, die keine wirklichen Artikel darstellen, wurden händisch einige Beispielartikel angesehen. Die Grenze für die Anzahl der Token wurde daraufhin auf zehn gesetzt. Damit verbleiben im Korpus nur Texte, die in der vorverarbeiteten Form aus mehr als zehn Token bestehen. Dieser Korpus aus 1 615 152 Texten soll für alle weiteren Analysen als Grundgesamtheit dienen.

4.2 Generieren von Subkorpora

In der Regel ist für eine journalistische Fragestellung nicht der gesamte Korpus von Interesse, sondern vielmehr die Gesamtheit der Texte zu einem bestimmten Thema. Dabei ist der ursprüngliche Korpus nur noch dafür relevant, das Thema von Interesse in Relation zur Gesamtberichterstattung zu stellen. Das Paket tosca bietet Funktionen zur zeitlichen und inhaltlichen Einschränkung von Korpora, letztere mit Hilfe von Wortlisten.

Die Grundlage der Analysen für diese Arbeit ist ein Subkorpus über die Bankenberichterstattung. Hier sind besonders die Bankenkrise von 2008 und ihre Auswirkungen von Interesse. Um auch die Berichterstattung vor diesem Ereignis in den Subkorpus aufzunehmen, werden in einem ersten Schritt alle Artikel der Jahre 2008 bis 2013 mit Hilfe von `filterDate` ausgewählt. Dieser Korpus dient als neue Grundgesamtheit und enthält 348 807 Texte. Um die für die Bankenberichterstattung relevanten Texte zu filtern, werden zwei verschiedene Wortsuchen mit Hilfe von `filterWord` durchgeführt: Der erste Subkorpus enthält alle Texte, in denen die Zeichenfolge „bank“ oder „sparkasse“ enthalten ist. So werden auch die Token „banken“ oder „commerzbank“ gefunden, aber natürlich auch „datenbank“ oder „trainerbank“. Der zweite Subkorpus wird deutlich weiter eingeschränkt. Hier werden nur Texte aufgenommen, in denen mindestens dreimal die Zeichenfolge „bank“ enthalten ist, oder mindestens einmal „sparkasse“ oder die Zeichenkettenkombinationen „bank“ und „kredit“ oder „bank“ und „institut“. Eine ausführliche Analyse der Qualität der Subkorpora ist in Kapitel 5 zu finden. Der für diesen Schritt notwendige R-Code lässt sich auf drei Funktionsaufrufe beschränken:

```
SZbankGG <- filterDate(SZ, s.date=as.Date("2008-01-01"),
                           e.date=as.Date("2013-12-31"))
SZbank   <- filterWord(SZbankGG, search=c("bank", "sparkasse"),
                       ignore.case=TRUE)
SZbank2  <- filterWord(SZbankGG, search=list(
  data.frame(pattern=c("bank"), word="pattern", count=3),
```

```
  data.frame(pattern=c("sparkasse"), word="pattern", count=1),
  data.frame(pattern=c("bank", "kredit"), word="pattern", count=1),
  data.frame(pattern=c("bank", "institut"), word="pattern", count=1)
  ), ignore.case=TRUE)
```

Der zeitlich eingeschränkte Grundgesamtheits-Korpus besteht damit aus 348 807 Texten, der weiter gefasste Banken-Korpus aus 52 329 Texten und der eingeschränktere Banken-Korpus aus 28 765 Texten. In der Regel interessiert nicht nur die Berichterstattung zu einem Thema, sondern auch der Anteil des Themas an der Gesamtberichterstattung. Mit der Funktion plotScot kann der Anteil eines Subkorpus am Gesamtkorpus dargestellt werden. In Abbildung 4.4 ist der Anteil des großen Banken-Korpus am Gesamtkorpus und der Anteil des kleinen Banken-Korpus am großen Banken-Korpus abgebildet. Für beide Grafiken wurden die Texte auf Monatsbasis aggregiert. Deutlich zu sehen ist der Anstieg im Anteil an der Berichterstattung aufgrund der Bankenkrise im September 2008. Auch danach bleibt dieser Anteil auf einem höheren Niveau.

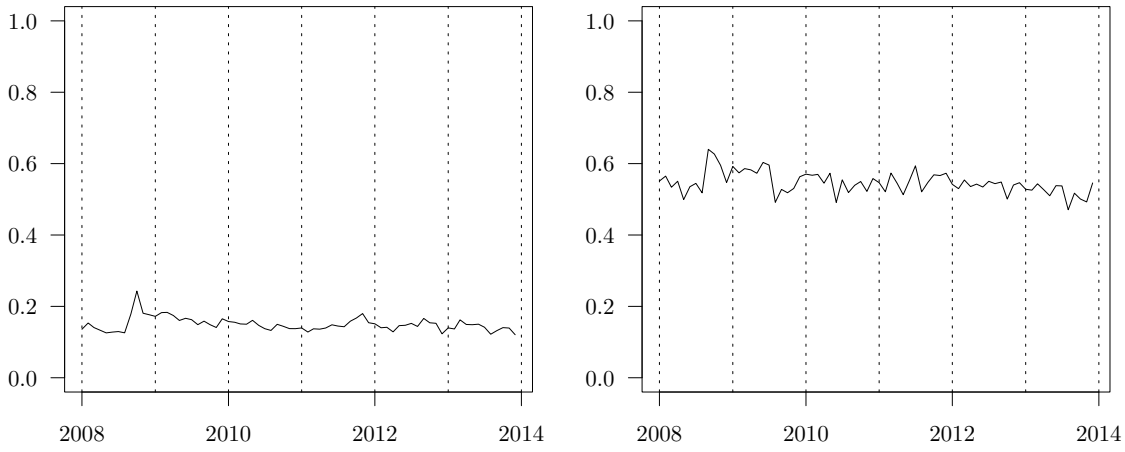

Abbildung 4.4: Die linke Grafik zeigt den Anteil des großen Banken-Korpus an der Gesamtberichterstattung der SZ über die Zeit. In der rechten Grafik ist der Anteil des kleinen am großen Banken-Korpus angegeben. Für beide Grafiken wurde die Berichteratattung monatsweise aggregiert.

Der mittlere Anteil an der Gesamtberichterstattung in den ersten acht Monaten in 2008 liegt bei 13.4 %, während der mittlere Wert in den Jahren 2009 bis 2014 bei 15.17 % liegt.

4.3 Latent Dirichlet Allocation

In diesem Abschnitt werden die Analyse-Möglichkeiten von tosca in Bezug auf LDA vorgestellt. Eine Betrachtung der Möglichkeiten zur Modelloptimierung ist in Kapitel 6 zu finden. Um die Rechenzeit zu verkürzen und die Qualität des Modells zu verbessern, werden die Wörter identifiziert, die mehr als fünfmal im Korpus enthalten sind. Hierbei hilft die Funktion `makeWordlist`, die ein Wrapper für die `table` Funktion ist. `makeWordlist` verarbeitet den Korpus in mehreren Teilen (Default 100 000 Texte pro Verarbeitungsschritt) und benötigt so deutlich weniger Arbeitsspeicher als `table`. Anschließend wird mit Hilfe des nun reduzierten Wörterbuchs mit der Funktion `LDAprep` ein Korpus gebildet, der den Voraussetzungen für `lda.collapsed.gibbs.sampler` aus dem lda-Paket entspricht. Die einzelnen Texte werden dadurch als Listenelemente dargestellt. Die einzelnen Token sind durch die (nullindizierte) Position im Wörterbuch gekennzeichnet.

```
Bank2Wordlist <- makeWordlist(text=SZbank2$text)
words5 <- Bank2Wordlist$words[Bank2Wordlist$wordtable>5]
SZbank2LDA <- LDAprep(SZbank2$text, vocab=words5)
```

Der so vorverarbeitete Datensatz kann nun an den Gibbs-Sampler übergeben werden. Zur Präsentation der Funktionalität wird hier eine LDA mit 30 Themen berechnet. Für die anderen Parameter werden die Standardeinstellungen verwendet. Die Markow-Kette läuft dabei mit einer Burnin-Phase von 70 und anschließend noch weitere 200 Iterationen. Die Hyperparameter α und η werden beide auf $1/k$ gesetzt.

```
LDAresult <- LDAgen(documents=SZbank2LDA, K=30, vocab=words5)
```

Das daraus resultierende Modell dient als Grundlage für alle weiteren Analysen in diesem Kapitel. Ein automatisch von `LDAgen` mitgeliefertes Ergebnis ist die Repräsentation der einzelnen Themen durch die Topwortlisten. Tabelle C.4 im Anhang zeigt die zehn relevantesten Wörter pro Thema. Neben einigen Stoppwort-Themen, also Themen, deren relevanten Wörter aus nicht informativen Wörtern bestehen, ist es bei den meisten Themen für menschliche Kodierer*innen möglich, eine passende Überschrift zu finden. Tabelle 4.2 zeigt die vom Autor gewählten Themen-Überschriften. Neben verschiedenen Themen, die der Bankenberichterstattung zuzuordnen sind, ist auch ein Fußball-Thema (Thema 18) und ein IT-Thema (Thema 13) enthalten. Im IT-Thema ist die „Datenbank“ das 16. Wort der Topwort-Liste. Diese Themen könnten für weitere Analysen als Filter genutzt werden, um so für die Fragestellung nicht relevante Texte aus dem Korpus herauszufiltern.

1	Börse	16	Stopwords
2	Stopwords	17	Banken / Kunden
3	Bundesfinanzen	18	Fußball
4	Landesbanken?	19	Ratingagenturen / Finanzprodukte
5	Rohstoffe / Entwicklung	20	US-Regierung
6	Krisenbanken -firmen	21	Stopwords
7	Justiz	22	Deutsche? Firmen
8	US-Banken	23	Bankbilanzen
9	Wirtschaftslage	24	GB-Banken
10	Landesbanken	25	Deutsche Bank
11	EZB	26	Finanzprodukte
12	SZ Stopwords	27	Stopwords
13	IT	28	Denkschulen
14	Finanz-Stopwords	29	Russland / China
15	Griechenland	30	Kunst

Tabelle 4.2: Übersicht über die 30 Themen der LDA

4.3.1 Visualisierung von Themen-Verläufen

Im klassischen LDA-Modell wird die zeitliche Komponente des Korpus nicht modelliert. Trotzdem ist eine zeitliche Betrachtung der Themenverläufe von Interesse. Da alle

hier genutzten Texte mit einem Datum versehen sind, können auch die Themen im zeitlichen Verlauf betrachtet werden. Dazu werden die den Themen zugeordneten Token jeweils in Quartalen zusammengezählt und als relativer Anteil mit der Funktion plotArea als Sediment-Plot dargestellt. Abbildung 4.5 zeigt den zeitlichen Verlauf der Themen. Die Themen sind dabei nach ihrer Größe sortiert. Beschriftet sind die Themen, die in mindestens einem Quartal drei Prozent oder mehr am Gesamtkorpus ausmachen.

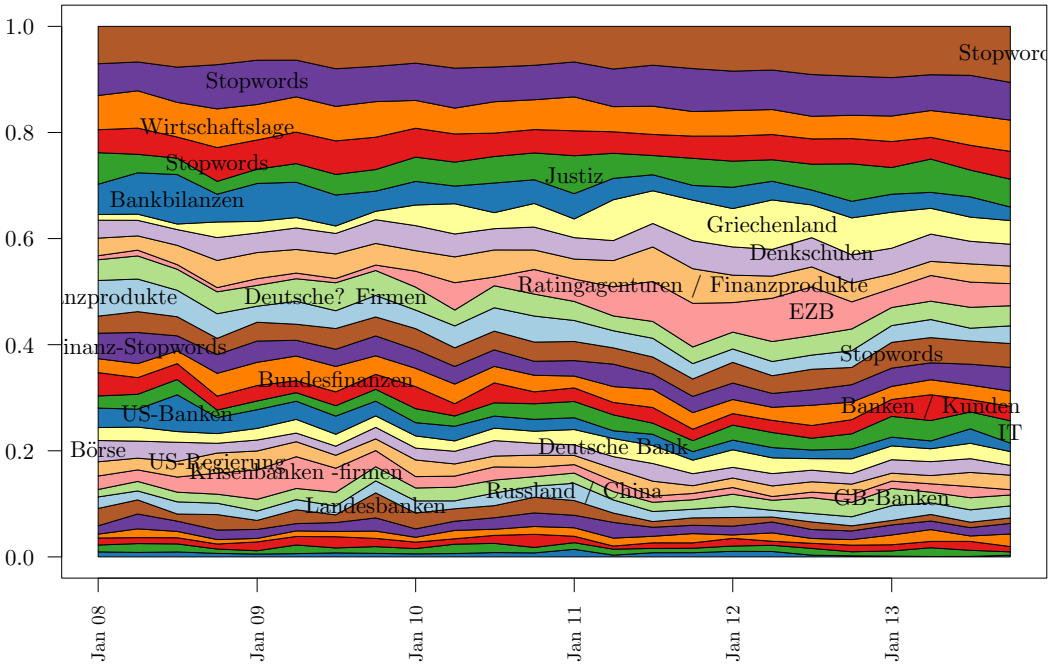

Abbildung 4.5: Relativer Anteil der Themen am kleinen Banken-Korpus im zeitlichen Verlauf. Die Daten sind auf Quartalsbasis zusammengefasst und nach Größe sortiert.

Typisch für so ein Themenmodell ist, dass die größten Themen in der Regel wenig Aussagekraft haben. In diesem Fall bestehen von den vier größten Themen drei Themen

aus Stoppwörtern. Für das vierte Thema sind allgemeine Begriffe zur Beschreibung der Wirtschaftslage besonders repräsentativ (siehe Tabelle C.4 im Anhang). Andere Themen wie zum Beispiel „Griechenland“, nehmen im Verlauf der Zeit einen größeren Teil der Gesamtberichterstattung ein oder sind zu einem Zeitpunkt besonders relevant, wie das „US-Banken“ Thema im dritten Quartal 2008.

Neben der Gesamtübersicht über die Themenanteile im Verlauf der Zeit können mit tosca auch einzelne Themen-Verläufe betrachtet werden. Die Funktion `plottopic` ermöglicht eine einfache Betrachtung eines oder mehrerer Themenverläufe. Abbildung 4.6 zeigt beispielhaft den zeitlichen Verlauf des Themas „Landesbanken“ in absoluten und relativen Werten. Auffällig ist, dass die Höhe der einzelnen Peaks in der absoluten und relativen Darstellung nicht gleich ausgeprägt sind. Als Beispiel sei hier der Beginn der Bankenkrise (4. Quartal 2008) genannt, in dem das Thema zwar absolut stark vertreten ist (der dritthöchste Peak des beobachteten Zeitraums), relativ aber nur an 6. Stelle liegt. In Abbildung 4.4 konnte bereits ein Anstieg der zum Banken-Korpus gehörenden Texte in diesem Zeitraum beobachtet werden, der alleine einen absoluten Anstieg der Token in diesem Zeitraum erwarten lässt. Der Anteil des Themas am Banken-Subkorpus steigt in dieser Zeit ebenfalls, es gibt aber mehrere Zeitpunkte, in denen das Thema einen höheren Anteil der Banken-Berichterstattung ausmacht.

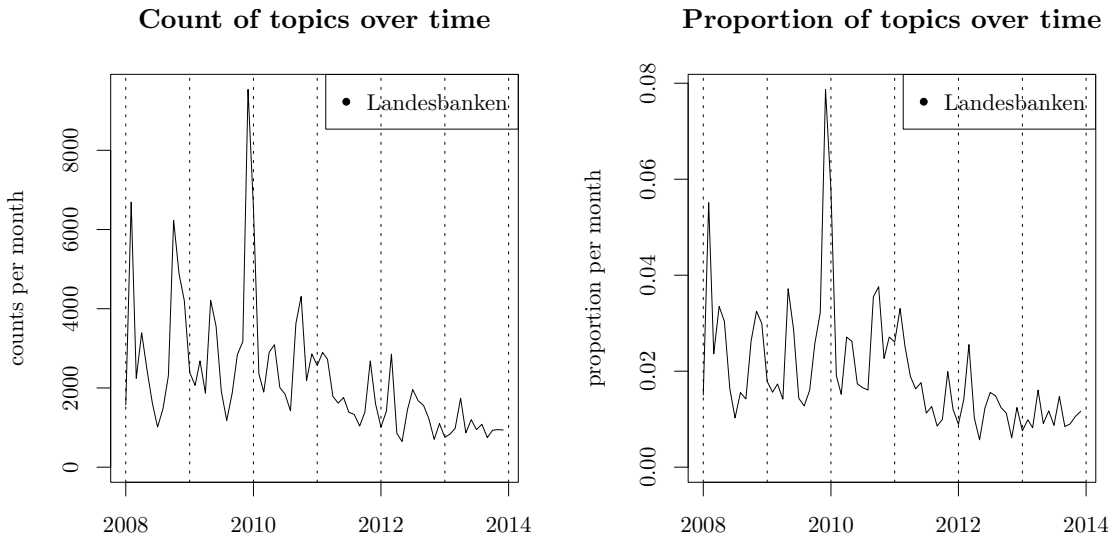

Abbildung 4.6: Absolute (links) und relative (rechts) Häufigkeit des Themas „Landesbanken“ pro Monat. Der relative Anteil bezieht sich auf den Subkorpus. Gezählt werden jeweils die dem Thema zugeordneten Token.

Der Verlauf einzelner Themen über die Zeit ermöglicht bereits einen Erkenntnisgewinn. Möchte man einzelne Themen noch genauer untersuchen, kann das Auftauchen einzelner Wörter in einem Thema beobachtet werden. Das Paket `tosca` bietet auch hier passende Funktionen. Hier sollen die Funktionen `plotTopicWord` und `plotWordpt` vorgestellt und auf ihre Unterschiede hingewiesen werden. Dazu werden die Funktionen auf vier ausgewählte Themen angewendet. Als Suchwort wird „euro“ verwendet. Ausgegeben werden soll jeweils das Ergebnis der Suche mit absoluten und relativen Werten.

```
topicselect <- c(6,8,11,14)
par(mfrow=c(2,2), mar=c(2,3,3,1), oma=c(0,0,0,0))
plotTopicWord(object = SZbank2, docs = SZbank2LDA, ldaresult = result,
           ldaID, select = topicselect, wordlist = "euro",
           rel=FALSE, tnames = tnames[topicselect], ylab="", xlab="")
```

```
plotWordpt(object = SZbank2, docs = SZbank2LDA, ldaresult = result,
           ldaID, select = topicselect, wordlist = "euro",
           rel=FALSE, tnames = tnames[topicselect], ylab="", xlab="")
plotTopicWord(object = SZbank2, docs = SZbank2LDA, ldaresult = result,
           ldaID, select = topicselect, wordlist = "euro",
           rel=TRUE, tnames = tnames[topicselect], ylab="", xlab="")
plotWordpt(object = SZbank2, docs = SZbank2LDA, ldaresult = result,
           ldaID, select = topicselect, wordlist = "euro",
           rel=TRUE, tnames = tnames[topicselect], ylab="", xlab="")
```

Abbildung 4.7 zeigt das Ergebnis des obigen Codes. Die beiden oberen Grafiken zeigen exakt die gleichen Kurven. Hier wird jeweils die absolute Häufigkeit des Suchwortes „euro“ pro Monat und Thema gezeigt. Auffällig ist die hohe Häufigkeit im Thema „Krisenbanken -firmen“ zwischen dem dritten Quartal 2008 und Ende 2009. Bei der Betrachtung aller vier Grafiken fallen die beiden Peaks des Themas „EZB“ in 2011 und 2012 auf. Diese Peaks finden sich in den relativen Daten nur bei `plotTopicWord` (linkes unteres Panel). Bei `plotWordpt` (Panel rechts unten) sind sie nicht zu finden. Dafür hat das Thema „Krisenbanken -firmen“ in den späteren Jahren deutlich höhere relative Werte als die anderen drei Themen. Der Unterschied in den beiden Darstellungen ist die jeweilige Grundgesamtheit, die zur Berechnung der relativen Werte herangezogen wird. Die Funktion `plotTopicWord` betrachtet für einen bestimmten Zeitraum die Verteilung des Suchwortes über die Themen. Hier addieren sich also pro Zeitraum die Anteile der Themen zu Eins. Wird ein Thema, wie zum Beispiel „Krisenbanken -firmen“ insgesamt größer (vgl. dazu den Sedimentplot in Abbildung 4.5), steigt für dieses Thema in der Regel auch der Anteil an einem Wort im Korpus. Im Gegensatz dazu normiert `plotWordpt` bezüglich der Anzahl der Wörter, die einem Thema zugeordnet sind. Hier kann also eine Veränderung der relativen Häufigkeit eines Wortes in einem Thema beobachtet werden. Während mit `plotTopicWord` die These „In der Bankenkrise wird das Wort ‚Euro‘ häufiger in Zusammenhang mit dem Thema ‚Krisenbanken -firmen‘ verwendet“ betrachtet werden kann, hilft `plotWordpt` bei Thesen wie zum Beispiel „Im Thema ‚Krisenbanken -firmen‘ ist das Wort ‚Euro‘ im Laufe der Zeit wichtiger geworden“.

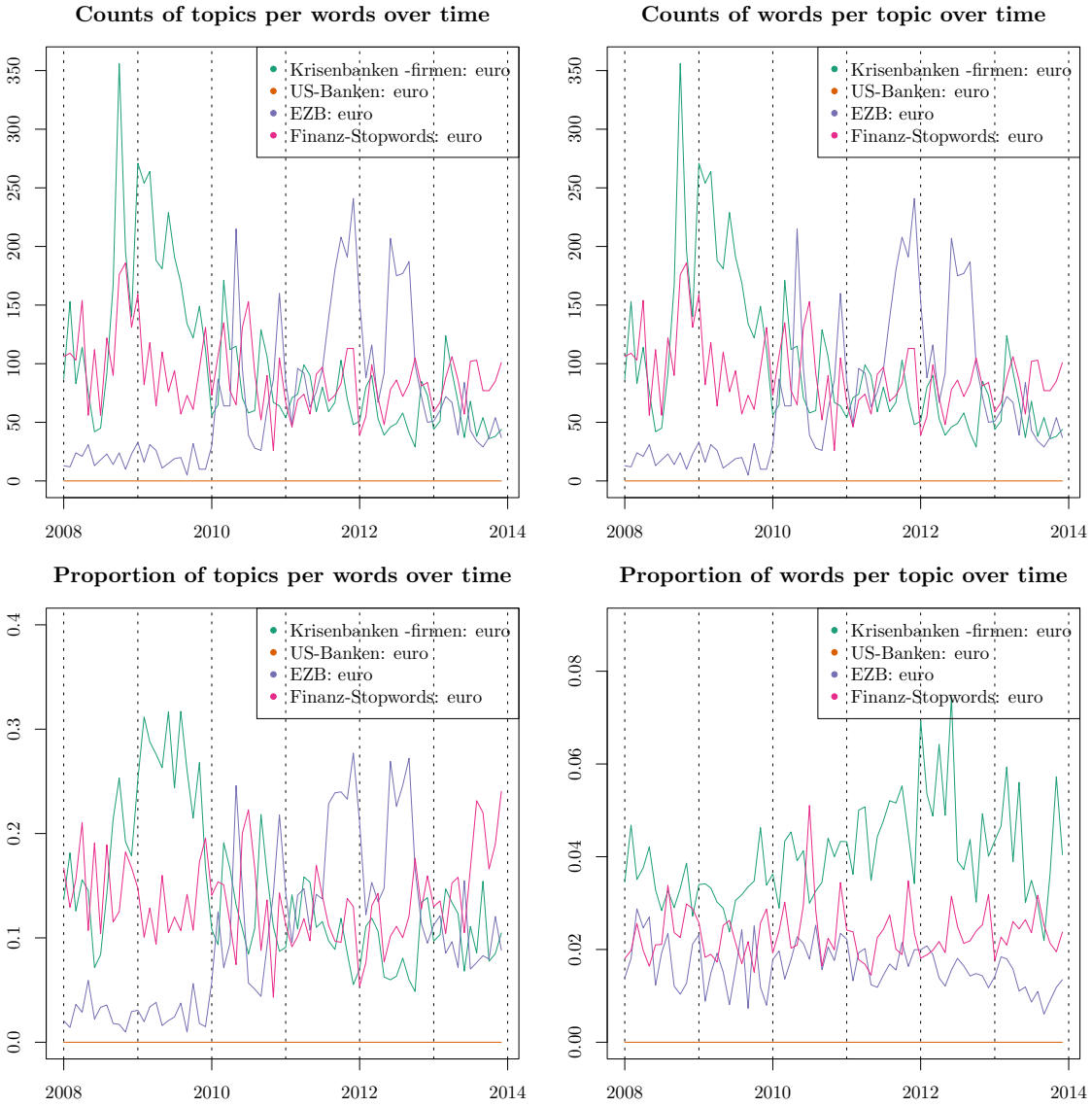

Abbildung 4.7: Worthäufigkeiten mit den Funktionen `plotTopicWord` und `plotWordpt`. Gezeigt wird das Auftauchen des Wortes „euro“ in vier ausgewählten Themen. Während die absoluten Häufigkeiten über die Zeit noch mit beiden Funktionen zum gleichen Ergebnis führen, sind relative Werte auf Grund der unterschiedlichen Grundgesamtheiten verschieden.

4.3.2 Themen-Cluster

Da die LDA ein unüberwachtes Verfahren ist, ist die Ähnlichkeitsstruktur der Themen unbekannt. Die Ähnlichkeit einzelner Themen ist allerdings von Interesse. Aus Sicht der Modelloptimierung können ähnliche Themen darauf hindeuten, dass die Anzahl der gewählten Themen zu hoch ist und mehrere Themen eigentlich zusammen ein Thema bilden sollten. Aber auch bei „wirklichen“ Themen ist eine Ähnlichkeitsstruktur für die Analyse von Bedeutung, da sie dabei hilft verwandte Themen(-gruppen) zu identifizieren.

Für einen schnellen Überblick über die Themen-Struktur ist in tosca eine hierarchische Clusteranalyse eingebunden, die mit Hilfe eines Dendogramms die Ähnlichkeitsstruktur der Themen innerhalb einer LDA oder auch zwischen verschiedenen LDA-Modellen visualisiert. Um die Ähnlichkeit zwischen zwei Themen zu bestimmen, wird der Abstand zwischen den beiden Wortverteilungen gemessen. Als Distanzmaß wird die Hellinger-Distanz (Oosterhoff und Zwet, 2012) verwendet, die für den Vergleich von zwei Wahrscheinlichkeitsverteilungen geeignet ist. Zur Erstellung des Diagramms wird das average-linkage-Verfahren verwendet. Abbildung 4.8 zeigt das Dendogramm für die in diesem Kapitel betrachetete LDA.

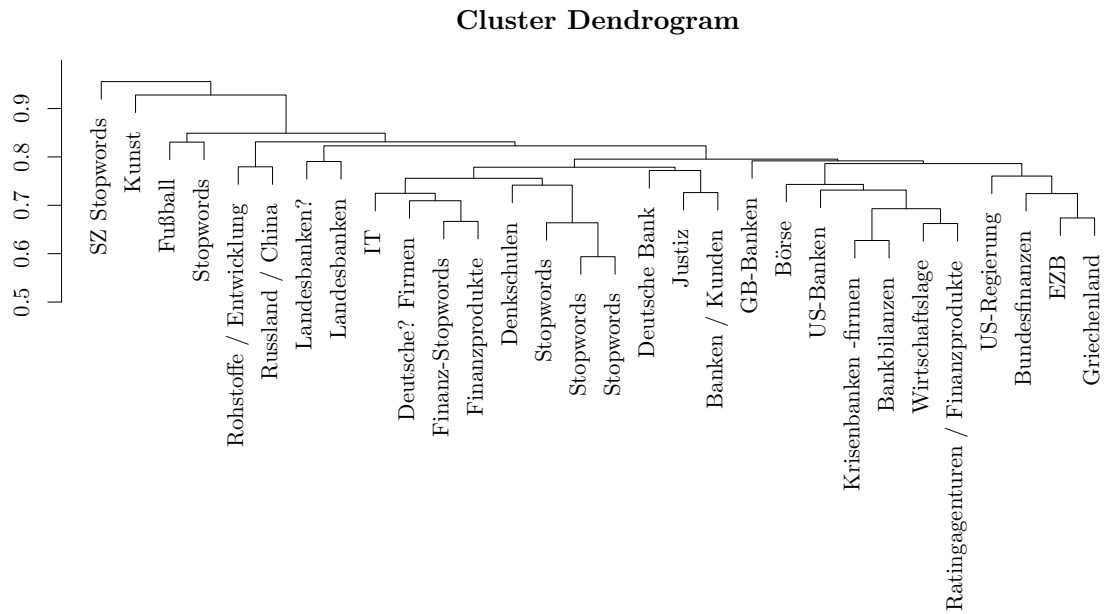

Abbildung 4.8: Dendrogramm über die Themen der LDA. Die Ähnlichkeit der Wortverteilungen wurde mit Hilfe der Hellinger-Distanz bestimmt. Das Dendrogramm wurde mit dem average-linkage hierarchischen Clustern erstellt.

Typisch für solche Dendrogramme ist, dass die ersten miteinander verbundenen Themen Stoppwort-Themen sind. Mit einer Ähnlichkeit von 0.59 werden auch hier zwei Stoppwort-Themen zuerst verbunden (mittig in der Grafik). Ein drittes Stoppwort-Thema kommt später hinzu. Auffällig ist außerdem, dass die beiden Landesbanken-Themen zuerst miteinander verbunden werden. Ob diese ein gemeinsames Thema bilden sollten, oder aber zwei Aspekte eines größeren Themas beleuchten, könnte durch die Analyse von repräsentativen Texten für beide Themen untersucht werden. Die vier Themen, die zuletzt dem Dendrogramm zugefügt werden, enthalten zwei Stoppwort-Themen, von denen eines aus für die Zeitung spezifischen Wörtern besteht. Die anderen beiden Themen sind Kunst und Fußball, für die eine große Distanz zu den anderen Themen plausibel ist.

4.3.3 Themen-Validierung

Bei der Validierung von Themenmodellen müssen zwei Ebenen beachtet werden. Zum einen kann das Modell im Ganzen mit seinen Parametern optimiert werden. Hierauf wird weiter in Kapitel 6 eingegangen. Zum anderen können auch die Themen innerhalb eines Modells validiert werden. Natürlich sind beide Aufgaben nicht unabhängig voneinander, da eine gute Interpretierbarkeit der Themen auch ein Auswahlkriterium für die Modellwahl sein sollte. In tosca gibt es mehrere Hilfsmittel, um die Themen einer LDA qualitativ oder quantitativ zu bewerten. Für die qualitative Beurteilung stehen insbesondere die Funktionen `topTexts` und `topicsInText` zur Verfügung. Die Funktion `topTexts` gibt für ausgewählte Themen die Texte mit dem höchsten Anteil dieser Themen heraus. Besonders kurze Texte können dabei ignoriert werden. Die so gewählten Texte können für einen menschlichen Betrachter als repräsentative Texte für das jeweilige Thema angesehen werden. Diese Texte helfen dabei, Themenüberschriften zu finden oder den Inhalt des Themas besser zu fassen.

Themen in Texten visualisieren

Mit `topicsInText` kann der umgekehrte Weg beschritten werden. Hier werden die durch das Modell geschätzten Themen in einem Text farblich markiert. In Abbildung 4.9 wurde die Funktion auf den Text „Letzte Chance fuer Lehman Brothers“ vom 15.09.2008 angewendet. Abbildung 4.10 zeigt den gleichen Artikel, nur dass hier die Wörter des Textes alphabetisch nach Themen sortiert wurden. Der Artikel stammt von dem Tag, an dem die US-Bank „Lehman Brothers“ Insolvenz anmeldete, ein Schlüsselereignis in der Bankenkrise von 2008. Insgesamt ordnet das Modell die Token im Text acht verschiedenen Themen zu. Dabei werden vier Themen („Bundesfinanzen“, „GB-Banken“, „Landesbanken“ und „Russland / China“) weniger als zehn Token zugeordnet. Themen mit sehr wenigen Worten sind oft Themen, die menschliche Kodierer*innen nicht dem Text zugeordnet hätten und bei denen nur einzelne Token, nicht aber die Aussage der entsprechenden Textstellen passen. Als Beispiel kann hier das Token „kontrolle“ gelten, das vom Modell als einziges Wort dem Thema „Russland / China“ zugeordnet wird. Aus dem Thema „Bundesfinanzen“ stammen auch nur sieben

Token. In dem Text äußert sich allerdings „Bundesfinanzminister Peer Steinbrueck“ zu der Thematik in den USA, was menschliche Kodierer*innen zwar in der Regel nicht als „Bundesfinanzen“ kodiert hätten, aber sehr wohl nachvollziehbar ist. An dieser Stelle wird deutlich, dass Themenmodelle zwar sehr hilfreich für Inhaltsanalysen sein können, aber nicht exakt wie menschliche Kodierer*innen arbeiten. Die vier häufigsten Themen („US-Banken“, „Ratingagenturen / Finanzprodukte“, „US-Regierung“ und „Bankbilanzen“) sind auch für menschliche Kodierer*innen erkennbar.

Intruder Words

Die beiden vorgestellten Funktionen geben bereits Möglichkeiten zur qualitativen Analyse eines Themen-Modells. Um Themen auch quantitativ bewerten zu können, sind in tosca die Funktionen `intruderWords` und `intruderTopics` implementiert. Bei diesen von Chang u. a. (2009) vorgestellten Verfahren wird ein Intruder (deutsch: Eindringling) in eine Wort- beziehungsweise Themenliste eingefügt, der nicht zu den anderen Listenelementen passt. Je nachdem, wie zuverlässig der Intruder gefunden wird, kann eine Aussage über die Eindeutigkeit des Themas getroffen werden.

```
intruderWords(beta=result$topics, byScore = TRUE,
    numTopwords = 30L, numIntruder = 1L, numOutwords = 5L,
    noTopic = TRUE, printSolution = FALSE, oldResult = NULL)
```

Für die Intruder Words werden Kodierer*innen Wortlisten vorgelegt. Diese Wortlisten enthalten die vier ersten Wörter der Topwort-Liste des jeweiligen Themas. Das fünfte Wort ist ein für das Thema nicht repräsentatives Wort. Um sicherzustellen, dass dieser Intruder kein seltenes Wort ist und er allein durch seine Seltenheit in der Textsammlung entdeckt wird, wird er aus den repräsentativen Wörtern der anderen Themen gezogen. In diesem Beispiel werden die Top 30 Wörter der anderen Themen verwendet. Enthalten diese Themen Wörter, die in den Top 30 des aktuellen Themas enthalten sind, werden diese aus der Auswahl im Vorfeld entfernt. Alle fünf Wörter werden in zufälliger Reihenfolge ausgegeben. Für das hier verwendete Modell wurde 10 Kodierer*innen für alle dreißig Themen jeweils eine Wortliste vorgelegt. Die Kodierer*innen bekamen die

Abbildung 4.9: Visualisierung der Themen in einem Text mit Hilfe von `topicsInText`. Farblich markiert sind die Themenzuordnungen der LDA. Schwarze Wörter sind bereits bei der Vorverarbeitung entfernt worden und werden somit im Modell nicht beachtet. Im oberen Bereich sind die Topwortlisten der vorkommenden Themen in der Reihenfolge der Anzahl der zugeordneten Wörter abgebildet.

Abbildung 4.10: Visualisierung der Themen in einem Text mit Hilfe von `topicsInText`. Farblich markiert sind die Themenzuordnungen der LDA. Die Wörter sind dabei alphabetisch nach Themen geordnet. Im oberen Bereich sind die Topwortlisten der vorkommenden Themen in der Reihenfolge der Anzahl der zugeordneten Wörter abgebildet.

Finanz-Stopwords	0.8	Fußball	0.2
Stopwords	0.8	Ratingagenturen / Finanzprodukte	0.2
Krisenbanken -firmen	0.7	GB-Banken	0.2
SZ Stopwords	0.6	Stopwords	0.2
Finanzprodukte	0.6	Russland / China	0.2
Stopwords	0.5	Börse	0.1
Rohstoffe / Entwicklung	0.5	Wirtschaftslage	0.1
Banken / Kunden	0.4	EZB	0.1
Deutsche? Firmen	0.4	Bankbilanzen	0.1
Denkschulen	0.4	Deutsche Bank	0.1
Bundesfinanzen	0.3	Kunst	0.1
Landesbanken?	0.3	US-Banken	0
Justiz	0.3	Landesbanken	0
Stopwords	0.3	Griechenland	0
IT	0.2	US-Regierung	0

Tabelle 4.3: Anteil der Kodierer*innen, die den Intruder in der jeweiligen Topwortliste nicht gefunden haben, pro Thema.

Information, dass es sich um genau einen Intruder handelt und dass das Themenmodell auf einem Banken-Korpus aufgestellt wurde. Bei Unsicherheit sollten sie zufällig aus den für sie plausiblen Wörtern wählen.

Anzahl der Fehler	5	6	7	8	10	18
Anzahl der Kodierer*innen	1	1	2	3	2	1

Tabelle 4.4: Anzahl der falsch kodierten Intruder pro Kodierer*in.

Tabelle 4.3 zeigt die Anteile der Kodierer*innen, die den Intruder nicht gefunden haben. Bei insgesamt 10 Themen haben nicht mehr als 10 % der Kodierer*innen den falschen Intruder angegeben, bei 20 waren es nicht mehr als 30 %. Insgesamt 2 Themen haben mit einer Fehlerquote von 80 % exakt das Ergebnis erreicht, das unter Zufall zu erwarten ist. Beide Themen wurden als Stoppwort-Themen identifiziert. Allerdings existieren auch Stoppwort-Themen, bei denen der Intruder gut gefunden werden konnte. Auch wenn Themen inhaltlich nicht gut interpretierbar sind, ist es manchmal möglich, anhand der Wortart ein Muster in dem Thema zu entdecken und so einen Intruder zu identifizieren. Es muss ebenfalls beachtet werden, dass die

Kodierer*innen unterschiedlich viel Wissen über die Inhalte des Banken-Korpus haben. Abgekürzte Banknamen und Namen von Vorstandsvorsitzenden konnten nicht von allen Kodierer*innen zweifelfrei zugeordnet werden. Die Häufigkeitstabelle 4.4 zeigt die Anzahl der unerkannten Intruder von den Kodierer*innen. Deutlich zu erkennen ist, dass ein Kodierer eine besonders hohe Zahl an unerkannten Intrudern erzeugt hat. Dies mag auf Probleme bei der Beschreibung der Kodieraufgabe zurückzuführen sein. Die drei Personen mit der meisten Erfahrung mit LDA und Intruder Words haben mit fünf, sechs und sieben Fehlern die wenigsten falsch erkannten Intruder, sodass hier ein Trainingseffekt angenommen werden darf. Die Funktion `intruderWords` ist sehr flexibel und unterstützt auch andere Kodieraufgaben. So kann die Zahl der Intruder flexibel gehalten werden, damit die Kodierer*innen nicht wissen, wie viele Intruder enthalten sind. Weiterhin gibt es die Möglichkeit, Themen von den Kodierer*innen als Stoppwort-Themen markieren zu lassen.

Intruder Topics

Während bei `intruderWords` die Auffindbarkeit von nicht für das Thema repäsentativen Wörtern überprüft wird, arbeitet `intruderTopics` mit der Identifizierbarkeit von Themen in Texten. Bei dieser Methode wird dem Kodierer ein Text aus dem Korpus zusammen mit einer Anzahl an Themen, repräsentiert durch die Topwort-Listen, vorgelegt. Die Aufgabe hier ist es Themen zu finden, die nicht zu dem Text gehören. Wie bereits bei `topicsInText` festgestellt, ordnet die LDA einzelne Token in den Texten auch Themen zu, die durch einen menschlichen Kodierer in der Regel nicht erkannt werden. Um diesen Effekt zu berücksichtigen, ist es sinnvoll nur Themen zu wählen, die mit ausreichend Token im Text vertreten sind. Für diese Analyse wurde ein Minimum von zehn Wörtern gewählt. Außerdem wurden die sechs erkannten Stoppwort-Themen aus der Themen-Auswahl entfernt.

```
intruderTopics(text=SZbankData$text, beta=result$topics,
      theta=result$document_sums, id=ldaID, minWords = 10,
      numIntruder=1, numOuttopics=4, byScore=TRUE,
      stopTopics = c(2, 12, 14, 16, 21, 27))
```

Für die Validierung der Themen wurden vom Autor 20 Texte bearbeitet. Dabei wurden 6 Intruder Topics falsch angegeben, was eine Fehlerquote von 30 % ergibt. Für eine richtige Anwendung ist die Zahl der verwendeten Texte zu gering und die Fehlerquote zu hoch. Die Themen sind also in dieser Form eher nicht gut den Texten zuzuordnen.

Weitere Funktionen von tosca

Über die an diesem Fallbeispiel präsentierten Funktionen von tosca hinaus, bietet das Paket noch weitere Funktionalitäten. Dies sind unter anderem kleinere Hilfsfunktionen, wie zum Beispiel `mergeTextmeta` und `mergeLDA` zum Zusammenfassen verschiedener Korpora beziehungsweise LDA-Themen. Letztere können so später geclustert werden. Auch im Bereich der Visualisierung existieren noch weitere Möglichkeiten. So können mit `plotFreq` Worthäufigkeiten über die Zeit visualisiert werden oder Themenverläufe mit `plotHeat` als Heatmap dargestellt werden. Während Einlesefunktionen für Quellen, die nur im Projekt DoCMA vorliegen nicht in das Paket integriert sind, enthält tosca Einlesefunktionen für allgemein zugängliche Quellen. Alle Funktionen haben gemeinsam, dass sie direkt ein Textmeta-Objekt erzeugen. Die Funktion `readTextmeta` liest CSV-Dateien ein. Außerdem wurden einige spezialisierte Funktionen erstellt: `readWhatsApp` zum Einlesen des Datenexport des Messengers WhatsApp, `readWiki` für Wikipedia-Artikel (`https://www.wikipedia.org/`) und `readWikinews` für Wikinews (`https://www.wikinews.org/`). Als weiterer großer Bereich wird das optimierte Ziehen von Texten aus verschiedenen Subkorpora behandelt. So kann die Qualität von Subkorpora effizient bewertet werden. Dieser Bereich wird im nächsten Kapitel vorgestellt.

5 Effektive Qualitätsbewertung von Subkorpora

In Abschnitt 3.2 wurden die Methoden zur Schätzung von Precision und Recall von Subkorpora vorgestellt. In diesem Kapitel sollen diese Methoden zuerst anhand der Banken-Korpora (siehe Abschnitt 4.2) präsentiert werden. Anschließend wird das Verhalten der verbesserten Sampling-Strategie für verschiedene Szenarien simuliert.

5.1 Der Banken-Korpus

Die in Abschnitt 3.2 beschriebenen Methoden zur Bestimmung von Precision und Recall zielen auf die Situation ab, in der aus verschiedenen Untermengen einer Grundgesamtheit jeweils zufällig Stichproben gezogen wurden und die daraus resultierenden Schätzer kombiniert werden sollen. Im Bereich der Inhaltsanalyse ist dies ein häufig auftretendes Problem, da der Filter für den interessierenden Subkorpus zumeist erst dann sinnvoll gewählt werden kann, wenn bereits Texte bezüglich ihrer Relevanz gelabelt wurden. In diesem Abschnitt wird das Vorgehen anhand einer Subkorpus-Auswahl zum Thema Banken präsentiert. Dabei werden die beiden Subkorpora aus Abschnitt 4.2 in einem ersten Schritt verwendet. Im zweiten Schritt werden die für den ersten Teil erhobenen Daten für die Bewertung eines neuen durch LDA generierten Datensatzes genutzt.

5.1.1 Bewertung der Wortfilter-Subkorpora

In diesem Abschnitt sollen die in Abschnitt 4.2 generierten Subkorpora bezüglich Precision und Recall bewertet werden. Für diese Bewertung muss eine Regel aufgestellt werden, die festlegt, welche Texte relevant sind. Hier werden zwei verschiedene Regeln

verwendet. Im ersten Fall sollen alle Texte als relevant gelten, die sich allgemein mit Banken-Berichterstattung beschäftigen. Im zweiten Fall interessieren nur die Texte, die sich mit US-amerikanischen Banken beschäftigen. Als Kodieranleitung werden folgende Regeln aufgestellt. Dabei entstammen die kursiven Bereiche einer präzisierten Definition der Fragestellung die sich durch einen Pretest als notwendig herausstellte.

Allgemeine Bankenberichterstattung
Als relevant werden alle Texte angesehen, die zumindest in einem Teil des Textes über Banken berichten, also in denen Banken Haupt- oder Nebenberichterstattungsgegenstand *(mehr als zwei Sätze)* sind. Dies beinhaltet beispielsweise Berichterstattung über die Banken- und Finanzkrise und damit auch die Staatsschuldenkrise, nicht jedoch Texte, die gar nicht über Banken berichten, oder in denen Banken nur als Randaspekt erwähnt werden, wie zum Beispiel eine Firmeninsolvenz. Ebenso nicht relevant ist eine standardisierte Finanz-Berichterstattung, wie zum Beispiel tabellarische Börsenkurse von Banken. *Auch allgemeine Berichte über Märkte, in denen Banken aktiv sind (Immobilien / Aktien), sind nicht relevant, solange nicht explizit über Banken berichtet wird. Tochterfirmen von Banken sind nicht relevant, wenn diese nicht im Bankgeschäft sind und im Text nicht weiter auf die Bank eingegangen wird. Finanzdienstleister wie MLP oder AWD sind nicht relevant. Das Rating einer Agentur ist nicht relevant, ein Text über Ratingagenturen schon. Ebenfalls nicht relevant sind Berichte über Bewegungen, die aus der Finanzkrise entstanden sind, wenn die Texte ansonsten keine Verbindung zu Banken aufweisen. Berichte über Bankraub sind nicht relevant, Berichterstattung über Fehlverhalten von Banken schon.*

Berichterstattung über US-Banken
Für die US-Banken-Berichterstattung sind nur Texte relevant, die mindestens in einem Abschnitt über US-Banken oder Ratingagenturen berichten *(als Haupt- oder Nebenberichterstattungsgegenstand)*. Hier reicht es nicht aus, dass eine amerikanische Ratingagentur erwähnt wird, die ihr Rating für eine Institution gesenkt hat. *Texte in dieser Kategorie gehören automatisch auch zur ersten Kategorie.*

Um die Eindeutigkeit dieser Kodieranleitung zu überprüfen, wurden 50 Texte von zwei Kodierer*innen gelabelt. Der Wert für Krippendorffs α für die allgemeine Bankenbe-

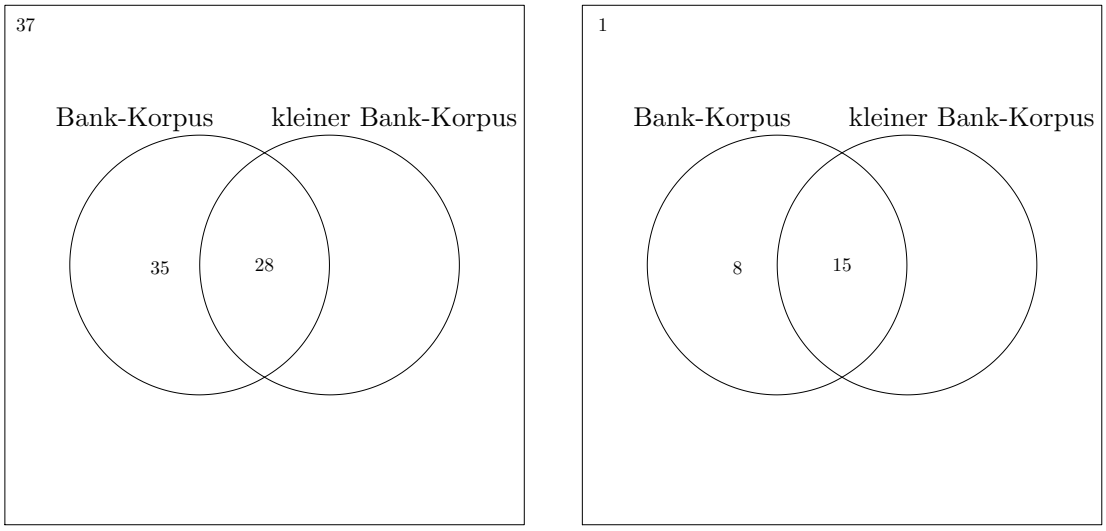

Abbildung 5.1: Venn-Diagramme der Interkoder-Stichproben. Die linke Grafik zeigt die Verteilung der Stichproben-Texte auf die einzelnen Schnittmengen, die rechte Grafik zeigt analog die Verteilung der relevanten Texte.

richterstattung lag im ersten Versuch bei 0.58. Nachdem das Kodierbuch um die oben hervorgehobenen Passagen ergänzt wurde, konnte anhand einer neuen Stichprobe ein α von 0.61 erreicht werden. Für das zweite Kriterium waren in diesen Stichproben zu wenig relevante Texte enthalten, sodass kein sinnvoller Wert bestimmt werden konnte. Alle Kodierungen der beiden Kodierer*innen aus diesen Stichproben wurden danach miteinander verglichen und bei Unstimmigkeiten wurde der „wahre“ Wert in diesem Durchgang festgelegt. In Abbildung 5.1 zeigen die Venn-Diagramme die Verteilung der Stichproben auf die einzelnen Schnittmengen und die als relevant kodierten Texte.

Mit diesen Informationen können die gewichteten Schätzer der beiden Subkorpora für Precision und Recall berechnet werden. Tabelle 5.1 zeigt die Schätzer für Precision und Recall für beide Subkorpora inklusive der Varianzschätzer für die Schätzer der beiden Maßzahlen. Wie zu erwarten war, ist die Precision im kleineren Banken-Korpus größer, während der weiter gefasste Banken-Korpus einen größeren Recall hat.

	Precision	sd	Recall	sd
Bank-Korpus	0.40	0.06087	0.72	0.20042
kleiner Bank-Korpus	0.54	0.09425	0.53	0.15626

Tabelle 5.1: Precision und Recall der allgemeinen Bankenberichterstattung, inklusive des jeweiligen Schätzers der Standardabweichung (sd) für die beiden Subkorpora auf Basis der Interkoder-Stichproben.

5.1.2 Bewertung eines neuen Subkorpus

Nachdem in einem ersten Schritt für die beiden durch Wortfilter generierten Subkorpora Precision und Recall berechnet wurden, soll nun ein weiterer Subkorpus betrachtet werden. Dieser wurde mit Hilfe eines Themen-Modells erzeugt. Auf dem großen Banken-Korpus wurde eine LDA mit 100 Themen gerechnet. Zwei dieser Themen können direkt mit unserer zweiten Definition der relevanten Texte in Verbindung gebracht werden: „US-Banken“ und „Krise (US)“. Tabelle C.5 im Anhang zeigt die Topwortlisten für diese beiden Themen. Für den Subkorpus werden dabei alle Texte ausgewählt, für die das Modell mindestens zehn Wörter aus den beiden Themen zugeordnet hat. Abbildung 5.2 zeigt die Verteilung der Texte auf die drei Korpora. Da die beiden kleineren Subkorpora Teilmengen des großen Banken-Korpus sind, sind einige Schnittmengen nicht besetzt.

Mit den bereits erhobenen Stichproben sind bereits alle nicht leeren Schnittmengen besetzt, wie in der obere Hälfte der Abbildung 5.3 zu sehen ist. Da aber insbesondere die beiden Schnittmengen, die zum US-Banken-Korpus gehören, sehr schwach besetzt sind, wurden weitere 50 Texte gezogen, davon 20 aus dem US-Banken-Korpus und 30 aus dem Banken-Korpus. Dies stellt sicher, dass der menschliche Kodierer nicht durch die Zugehörigkeit der Texte beeinflusst wird. Diese zusätzlichen Informationen sind in der unteren Hälfte der Abbildung 5.3 integriert. Betrachtet wird hier weiterhin das erste Kriterium, also eine allgemeine Bankenberichterstattung.

Mit diesen Daten können analog zum ersten Beispiel Schätzer für Precision und Recall berechnet werden. Tabelle 5.2 zeigt die Schätzer, die sich ergeben, wenn nur die ersten beiden Stichproben verwendet werden. In Tabelle 5.3 sind analog dazu die

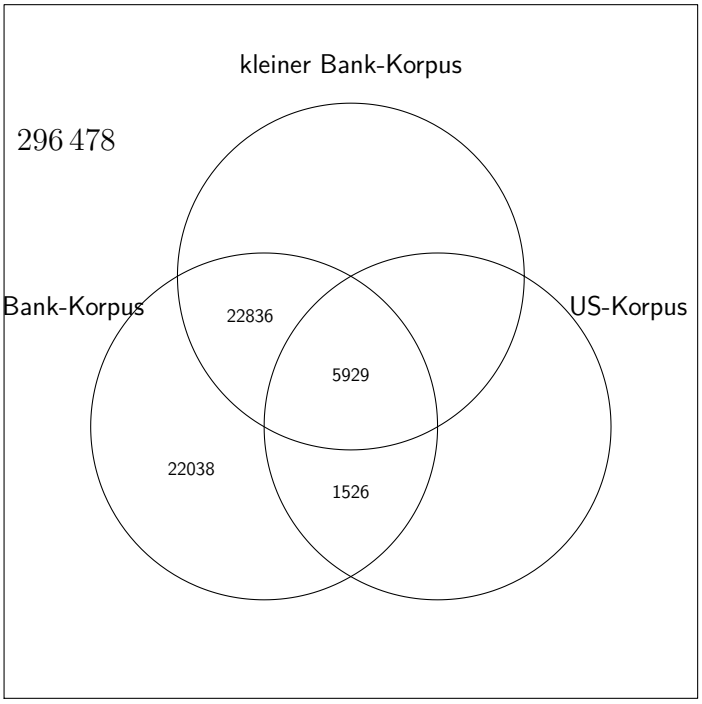

Abbildung 5.2: Venn-Diagramm für die drei betracheten Subkorpora. Da die beiden kleineren Subkorpora Teilmengen des großen Banken-Korpus sind, sind einige Schnittmengen nicht besetzt. Die äußere Menge bezeichnet dabei den bereinigten Datensatz, der als Grundgesamtheit in Abschnitt 4.2 erstellt wurde.

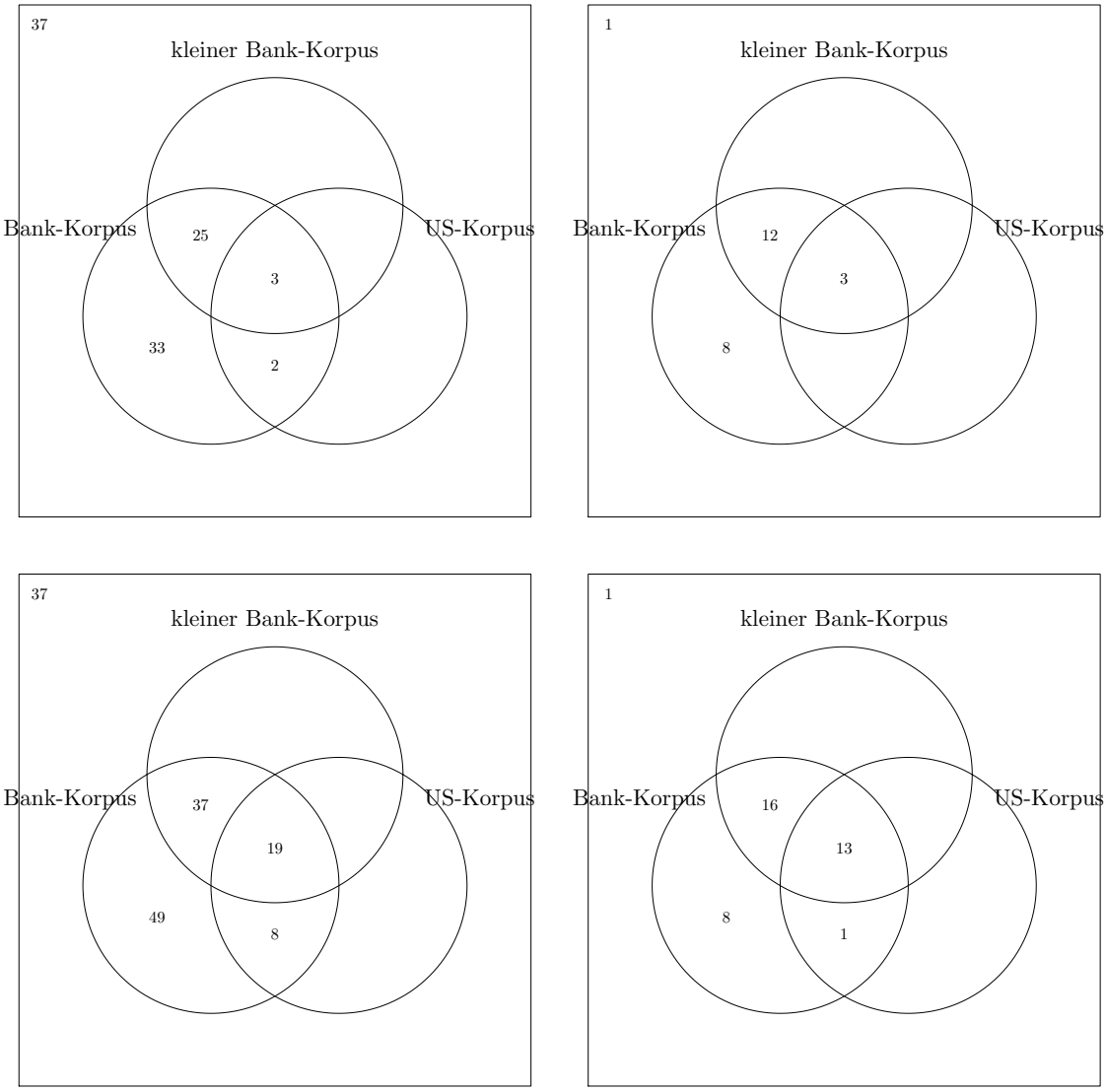

Abbildung 5.3: Venn-Diagramme der Stichproben. Die linken Grafiken zeigen die Verteilung der Stichproben-Texte auf die einzelnen Schnittmengen, die rechten analog die Verteilung der relevanten Texte. Bei den oberen Grafiken wurden nur die ersten beiden Stichproben verwendet, in den unteren Grafiken die zusätzliche Stichprobe für den US-Banken-Korpus.

Schätzer enthalten, die die Informationen aus der letzten Stichprobe enthalten. Die Varianzschätzer für die Recall-Schätzer der drei Subkorpora sind sogar bei der größeren Stichprobe höher. Dies liegt daran, dass in der kleineren Stichprobe die π_i für zwei Schnittmengen 1 beziehungsweise 0 geschätzt werden und die Varianzen hier mit 0 geschätzt werden. Im folgenden Simulationsteil wird auf dieses Problem eingegangen.

	Precision	sd	Recall	sd
Bank-Korpus	0.42	0.0537439	0.74	0.19366
kleiner Bank-Korpus	0.59	0.0793246	0.56	0.15273
US-Bank-Korpus	0.80	0.0000000	0.20	0.05437

Tabelle 5.2: Precision und Recall der allgemeinen Bankenberichterstattung, inklusive des jeweiligen Schätzers der Standardabweichung (sd) für die drei Subkorpora auf Basis der ersten beiden Interkoder-Stichproben.

	Precision	sd	Recall	sd
Bank-Korpus	0.34	0.0437646	0.69	0.21331
kleiner Bank-Korpus	0.48	0.0682920	0.54	0.17172
US-Bank-Korpus	0.57	0.0881233	0.17	0.05677

Tabelle 5.3: Precision und Recall der allgemeinen Bankenberichterstattung, inklusive des jeweiligen Schätzers der Standardabweichung (sd) für die drei Subkorpora auf Basis der drei Stichproben.

Will man nun weitere Texte kodieren um die Schätzungen zu verbessern, kann die in Abschnitt 3.2.1 vorgestellte Strategie verwendet werden. In diesem Beispiel sollen 100 weitere Texte gezogen und gelabelt werden. Die Tabellen C.9 und C.10 im Anhang zeigen für die zwei beziehungsweise drei Stichproben, die nach Formel 3.10 aus Abschnitt 3.2.1 berechneten Kandidaten für eine optimale Aufteilung. Verzeichnet sind alle 31 möglichen Kombinationen bei 5 nicht-leeren Schnittmengen. Einige Kombinationen enthalten negative Werte für die Zahl der zu ziehenden Texte. Da dies auf die Anwendung bezogen keine sinnvolle Lösung ist, werden diese Kombinationen ignoriert. Für die übrigen kann die Varianz des Recall-Schätzers unter Beibehaltung der Schätzwerte für die π_i berechnet werden. Die Kombination mit der kleinsten Varianz wird als optimale Lösung festgelegt. Anhand der beiden Tabellen ist ersichtlich, dass der vereinfachte Algorithmus mit der Version anzufangen, in der für alle Schnittmengen die Zahl der zu

ziehenden Texte berechnet wird und dann immer den größten negativen Wert auf null zu setzen, nur im zweiten Fall zum Optimum geführt hat. Da nur ganze Texte gezogen werden können, muss das Ergebnis noch so gerundet werden, dass sich die Gesamtzahl von 100 Texten nicht ändert. Im ersten Fall werden damit im Idealfall 94 Texte aus der äußeren Menge und 6 Texte aus der Schnittmenge der beiden Banken-Korpora gezogen. Der schnellere Algorithmus hätte 69 Texte aus der äußeren Menge, 6 Texte aus der Schnittmenge von großem Banken-Korpus und US-Banken-Korpus und 25 Texte aus der Schnittmenge der drei Subkorpora gezogen. Im Fall der zusätzlichen Texte aus dem US-Banken-Korpus werden 84 Texte aus der äußeren Menge gezogen und 16 aus der Schnittmenge, die zu allen drei Subkorpora gehört. Hier wird zum einen deutlich, dass die Aufteilung variiert, je nachdem wie viel Information in den einzelnen Schnittmengen vorliegt und zum anderen, dass der schnelle Algorithmus nicht zwingend das globale Minimum findet. In den folgenden Simulationen soll nun betrachtet werden, in wie weit sich die einzelnen Strategien unterscheiden.

	Precision	sd	Recall	sd
Bank-Korpus	0.03	0.0119432	1.00	0.00000
kleiner Bank-Korpus	0.05	0.0208226	0.89	0.09815
US-Bank-Korpus	0.23	0.0838330	1.00	0.00000

Tabelle 5.4: Precision und Recall der US-Bankenberichterstattung, inklusive des jeweiligen Schätzers der Standardabweichung (sd) für die drei Subkorpora auf Basis der drei Stichproben.

Der Subkorpus, der aus den beiden US-Banken-Themen der LDA gefiltert wurde, sollte sich besonders für das bisher nicht betrachtete zweite Kriterium eignen, das nur Texte als relevant ansieht, die über US-Banken berichten. Da bei allen drei Stichproben auch dieses Kriterium erhoben wurde, können analog zum ersten Kriterium Precision und Recall berechnet werden. Tabelle 5.4 gibt die Maßzahlen für die drei Subkorpora an. Erwartungsgemäß hat der US-Bank Korpus mit 23.49 % die höchste Precision, aber auch hier sind mehr als drei Viertel der Texte für die US-Bankenberichterstattung nicht relevant. Bei der Interpretation von LDA-generierten Themen muss deswegen berücksichtigt werden, dass auch solche Subkorpora nicht zwingend eine hohe Precision haben müssen.

5.2 Simulation verschiedener Sampling-Szenarien

In diesem Abschnitt sollen verschiedene auf den in Abschnitt 3.2 aufbauende Sampling-Strategien mit einer einfachen Zufallsauswahl verglichen werden. Die Strategien wurden bereits in Abschnitt 3.2.1 eingeführt. Für die **standard**-Methode werden die zu kodierenden Texte zufällig aus der Grundgesamtheit gezogen. Berechnet man alle aus der Formel 3.10 in Abschnitt 3.2.1 resultierenden Verteilungen der neu zu ziehenden Stichproben, kann man aus diesen diejenige wählen, die die Varianz des Recall-Schätzers minimiert. In **bestround** wird diese Aufteilung gewählt und auf ganze Zahlen gerundet, falls das Optimum nicht ganzzahlig ist. Da dieses Vorgehen bei vielen Schnittmengen recht rechenintensiv sein kann, wird außerdem **quick** betrachtet. Bei dieser Methode wird, wie in Abschnitt 3.2.1 beschrieben, mit der Verteilung begonnen, in der keine Schnittmenge auf Null gesetzt ist, und solange iterativ die kleinste negative Zahl auf Null gesetzt, bis alle zu ziehenden Stichproben positiv oder Null sind. Für die online-Variante werden noch zwei weitere Methoden betrachtet. Bei **best1** wird jeweils aus der Schnittmenge gezogen, die die größte Varianzreduktion verspricht. Da dies bei ungleichen Informationen über die Varianz in den Schnittmengen dazu führen kann, dass oft hintereinander aus der gleichen Schnittmenge gezogen wird, versucht **sample1** dies etwas abzuschwächen, indem die Entscheidung für die Schnittmenge zufällig gezogen wird. Die erwartete Varianzreduktion dient dabei als Gewicht für die jeweilige Schnittmenge.

5.2.1 Simulation des Anwendungsbeispiels

In Abschnitt 5.1.2 wurde am Beispiel mit drei Subkorpora gezeigt, wie 100 Texte mit der Strategie `bestround` auf die Schnittmengen aufgeteilt werden können, sodass die Varianz des Recall-Schätzers minimiert werden kann. In diesem Abschnitt sollen die verschiedenen Strategien genauer untersucht werden. Dafür wird angenommen, dass die aus den drei Stichproben berechneten Schätzer für die π_i die wahren Werte sind. Es sollen nun jeweils 10 000 Stichprobenziehungen simuliert werden. Gezogen wird zum einen zufällig aus dem gesamten Korpus (`standard`) und zum anderen nach

der in Abschnitt 5.1.2 berechneten optimalen Aufteilung (`bestround`). In Tabelle 5.5 sind die Werte für die π_i, die w_i, die erwartete Aufteilung unter Ziehung von 100 zufälligen Texten und die berechnete optimale Aufteilung angegeben. Auffällig ist, dass die Aufteilung in der äußeren Schnittmenge fast identisch ist und sich nur die Verteilung der Texte auf die übrigen Schnittmengen unterscheidet.

	F F F	T F F	T F T	T T F	T T T
π_i	0.03	0.16	0.12	0.43	0.68
sd	0.03	0.05	0.12	0.08	0.11
w_i	0.85	0.06	0.00	0.07	0.02
Zufallsauswahl	85.00	6.32	0.44	6.55	1.70
optimiert nach Schnittmengen	84.00	0.00	0.00	0.00	16.00

Tabelle 5.5: Werte für die π_i, die w_i, die erwartete Aufteilung unter Ziehung von 100 zufälligen Texten und die berechnete optimale Aufteilung für die einzelnen Schnittmengen im Anwendungsbeispiel. Die Spalten geben jeweils die Schnittmengen an, wobei die Spaltenüberschriften für die drei Subkorpora „großer Banken-Korpus“, „kleiner Banken-Korpus“ und „US-Banken-Korpus“ kennzeichnen, ob die Schnittmenge Teil des Korpus ist (T) oder nicht (F).

Abbildung 5.4 zeigt die Histogramme der berechneten Varianzen des Recall-Schätzers für die Simulation der beiden Verfahren. Da die verbesserte Strategie sicherstellt, dass aus den Schnittmengen gezogen wird, die die größte Varianzreduktion erwarten lassen, ist das Histogramm im Vergleich zur Zufallsauswahl rechts abgeschnitten. Es wird also sichergestellt, dass keine zufälligen Aufteilungen vorgenommen werden, die eine vergleichsweise hohe Varianz erzeugen.

5.2.2 Simulation weiterer Szenarien

In diesem Abschnitt soll untersucht werden, welche Sampling-Strategien für welche Szenarien geeignet sind. Dabei wird zuerst eine eher theoretische Simulation betrachtet, bevor verschiedene Anwendungsszenarien simuliert werden. Für alle Simulationen werden drei Subkorpora verwendet. Es existieren also, zusammen mit den Texten, die zu keinem Subkorpus gehören, insgesamt acht Schnittmengen (siehe Abbildung

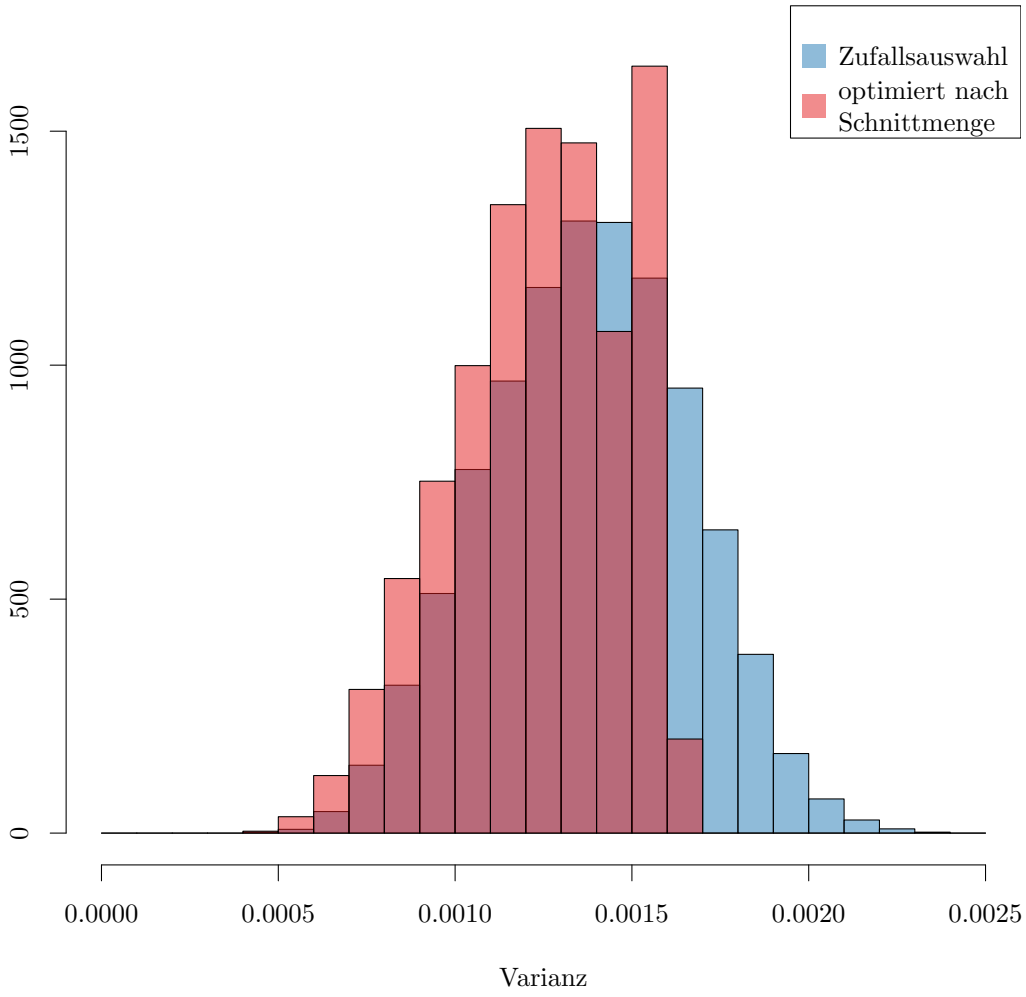

Abbildung 5.4: Histogramme der Varianzschätzer der Recall-Schätzer der beiden Sampling-Methoden im Anwendungsbeispiel.

3.1). Für alle Simulationen werden jeweils 100 000 Durchläufe durchgeführt. In jedem Durchlauf werden alle Sampling-Strategien (`standard`, `bestround`, `quick`, `best1` und `sample1`) durchgeführt. Dabei wird davon ausgegangen, dass in allen Schnittmengen ausreichend Texte vorhanden sind, sodass aus dieser auch gezogen werden kann. Die Endlichkeit der Schnittmengen wird nicht weiter betrachtet. Ein Subkorpus wird als der für die Berechnung des Recall relevante bestimmt. Wenn ein π_i nicht berechnet werden kann, weil noch keine Texte aus der Schnittmenge gezogen wurden, wird als Schätzer 0.5 angenommen. Bei Schätzern, die Null oder Eins sind, wird als Schätzer $\frac{1}{n+1}$ beziehungsweise $\frac{n}{n+1}$ gewählt.

Dirichlet-verteilte Schnittmengengrößen

Für die erste Simulation werden die simulierten w_i, also die Aufteilung der Texte im Korpus auf die Schnittmengen, aus einer Dirichlet-Verteilung mit Parameter $1/8$ gezogen. Der Anteil der relevanten Texte π_i wird aus einer Gleichverteilung über $[0, 1]$ gezogen. Es wurden jeweils zehn Beobachtungen aus jeder Schnittmenge gezogen, um einen Anfangsschätzer für π_i zu erhalten. Anschließend werden weitere 100 Texte nach der jeweiligen Strategie gezogen. Um die Texte den einzelnen Schnittmengen der Subkorpora zuzuordnen, wird ihnen jeweils eine ID aus drei Ziffern zugeordnet. Die Ziffer an der i-ten Stelle gibt dabei an, ob die Schnittmenge zum i-ten Subkorpus gehört (1) oder nicht (0). Zur Berechnung des Recall wird der erste Subkorpus, also alle Schnittmengen, deren ID mit einer eins beginnt, als der Subkorpus von Interesse bestimmt. Abbildung 5.5 zeigt die Beanplots (Kampstra, 2008) der Schätzungen der Standardabweichung der Schätzer für den Recall.

Rang	standard	bestround	quick	best1	sample1
1	32646	17058	17253	15927	17216
2	19216	20481	20459	19846	20100
3	16707	20889	20953	20791	20710
4	15776	20893	20757	21608	20975
5	15655	20679	20578	21828	20999

Tabelle 5.6: Ränge der absoluten Abstände zum wahren Wert für Dirichlet-verteilte Schnittmengengrößen bei 100 000 Wiederholungen.

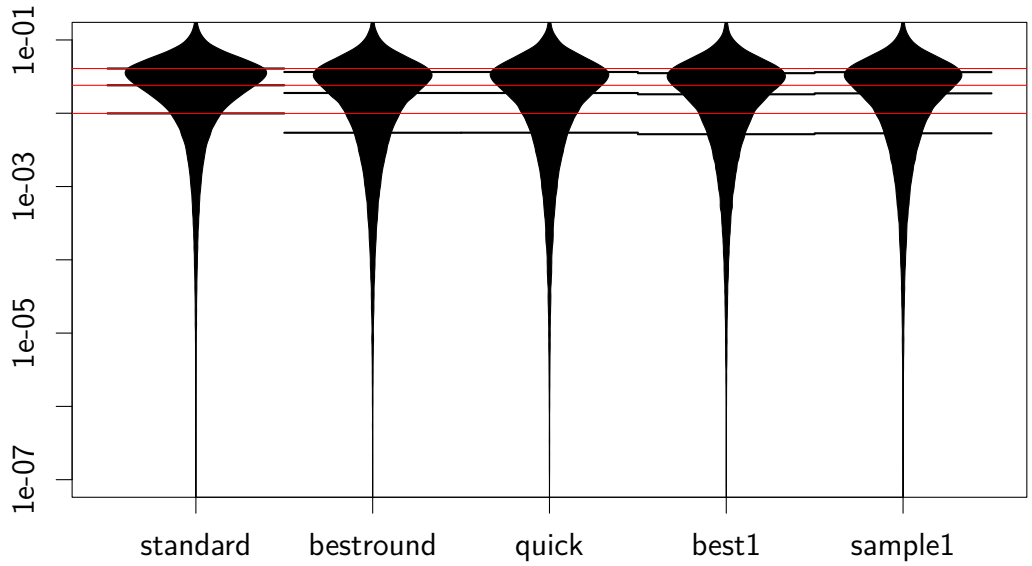

Abbildung 5.5: Trunkierte Beanplots der geschätzten Standardabweichungen der Recall-Schätzer für Dirichlet verteilte Schnittmengengrößen auf einer log-Skala. Die roten Linien markieren die Quartile der `standard`-Methode.

Dadurch, dass die Größen der einzelnen Schnittmengen alle aus der gleichen Dirichletverteilung und die Anteile für relevante Texte jeweils gleichverteilt gezogen werden, existieren für einzelne Schnittmengen keine besonderen Eigenschaften bezüglich dieser Parameter. Die Beanplots in Abbildung 5.5 ähneln sich durchaus. Betrachtet man die rot eingezeichneten Quartile der `standard`-Methode, also der Zufallsauswahl über den gesamten Korpus, sieht man, dass die 25 %-, 50 %- und 75 %-Quartile der anderen Methoden niedriger liegen. Im Mittel sind die Verbesserungen für die anderen Methoden im Vergleich zu `standard` 0.0046, beziehungsweise 0.0054 und 0.0046. Die mediane Verkleinerung der Standardabweichung im Vergleich zur Standardmethode beträgt demnach absolut 0.0054 und relativ 22.49 %. In Tabelle 5.6 sind für die fünf Methoden und die 100 000 Simulationen die Häufigkeiten der Ränge vermerkt. Dabei wurde jeweils die absolute Abweichung der Schätzer vom wahren Wert betrachtet. Bei gleichem Rang wurde für alle betroffenen Methoden der kleinste dazugehörige Rang vermerkt. In dieser Simulation ist die `standard`-Methode in knapp 1/3 der Fälle die beste Wahl. Die Methoden `bestround`, `quick` und `sample1` unterscheiden sich nicht wesentlich, `best1` schneidet etwas schlechter ab.

Anwendungsnahes Szenario

Die oben beschriebene Simulation ist auf die übliche Anwendung innerhalb einer Inhaltsanalyse als realitätsfern zu bezeichnen. Die Größe der Schnittmengen unterliegt nicht der gleichen Verteilung, die eine Austauschbarkeit der Schnittmengen voraussetzen würde. Die Schnittmenge mit den Texten, die in keinen Subkorpus aufgenommen wurden, ist in der Regel deutlich größer als die anderen Schnittmengen und enthält, unter geschickter Auswahl der Subkorpora, nur sehr wenige oder sogar keine relevanten Texte. Andersherum sollte in der Schnittmenge mit den Texten, die in allen Subkorpora enthalten sind, ein sehr hoher Anteil an relevanten Texten sein. Um diesen Aspekt zu berücksichtigen, wurden die Anteile der Schnittmengen am Gesamtkorpus w_i und die Anteile der relevanten Texte innerhalb der Schnittmengen π_i für die weiteren Simulationen realistischer simuliert. Die Werte für die einzelnen Schnittmengen sind in Abbildung 5.6 dargestellt. Eine tabellarische Aufzählung befindet sich im Anhang (Tabellen C.6 und C.7). Für jede Simulation wurden die wahren Werte gleichverteilt

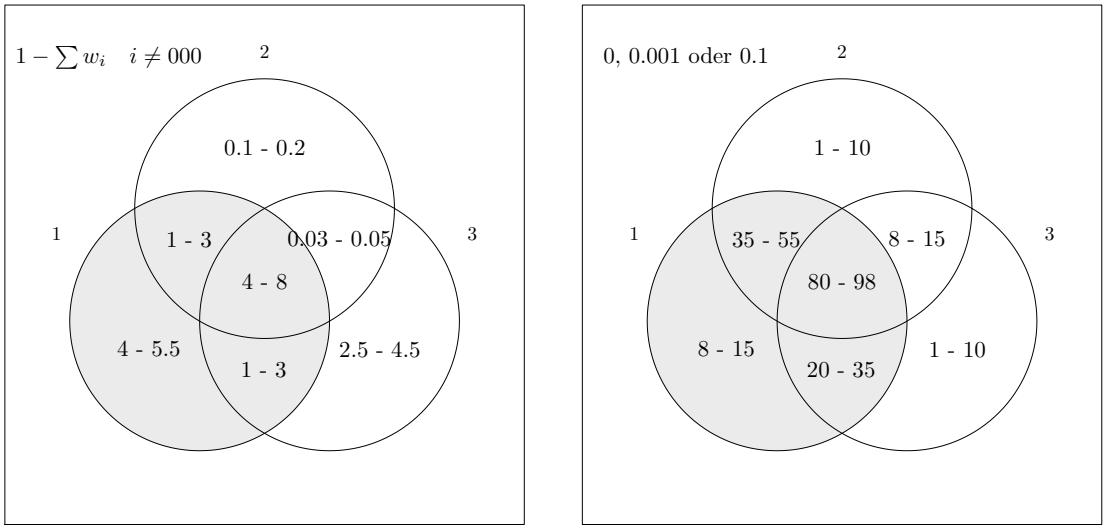

Abbildung 5.6: Wertebereiche für die Anteile w_i der Schnittmengen am Gesamtkorpus (links) und für den Anteil relevanter Texte π_i in den verschiedenen Schnittmengen (rechts). Der markierte Subkorpus ist der, für den die Maßzahlen berechnet werden sollen. Alle Angaben in Prozent.

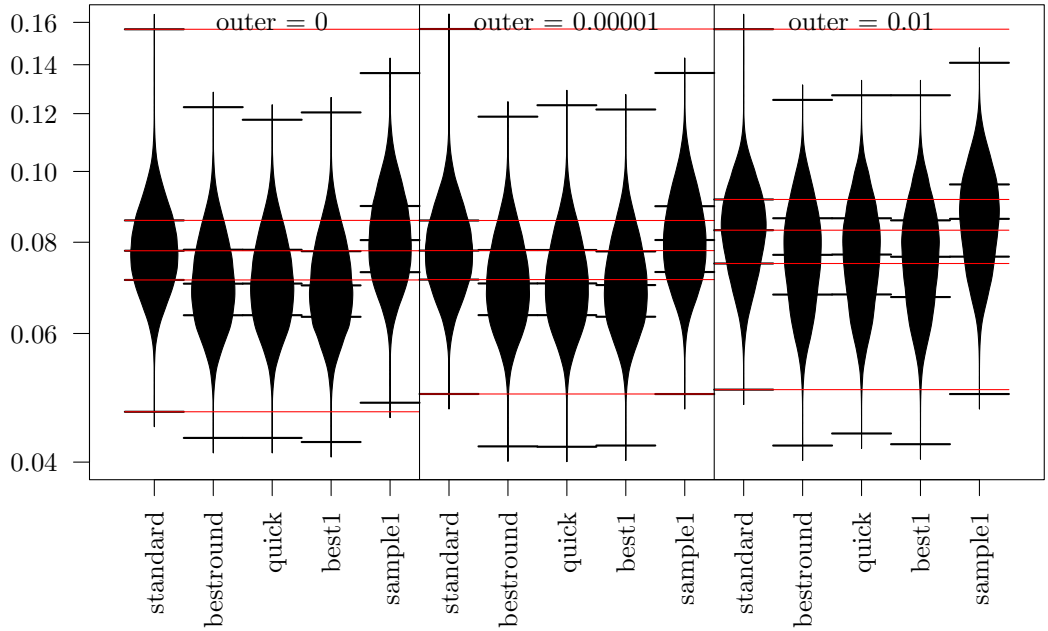

Abbildung 5.7: Trunkierte Beanplots der geschätzten Standardabweichungen der Recall-Schätzer für das anwendungsnahe Szenario auf einer log-Skala. Pro Subkorpus wurden 20 Texte im Vorfeld gelabelt, in der äußeren Menge 50. Weitere 100 Texte wurden nach der jeweiligen Strategie gezogen. Die roten Linien markieren die Quartile der `standard`-Methode.

aus dem jeweiligen Wertebereich gezogen. Einzig der Wert π_1, also den Anteil der relevanten Texte, die zu keinem Subkorpus gehören, wird nicht zufällig ausgewählt. Hier werden für die drei in der Abbildung 5.6 genannten Einstellung jeweils 100 000 Simulationen durchgeführt. Die Wahl der jeweiligen Bereiche ergibt sich aus den im Projekt untersuchten Fragestellungen und den dort auftretenden Werten. Zugleich wurde darauf geachtet, dass ein breites Spektrum an realistischen Szenarien abgedeckt wird.

Für diese Einstellungen werden nun verschiedene Simulationen betrachtet. Im ersten Fall werden im Vorfeld pro Subkorpus 20 Texte gelabelt und in der äußeren Menge 50, also der Menge deren Texte zu keinem Subkorpus gehören. Dies ist deutlich realistischer

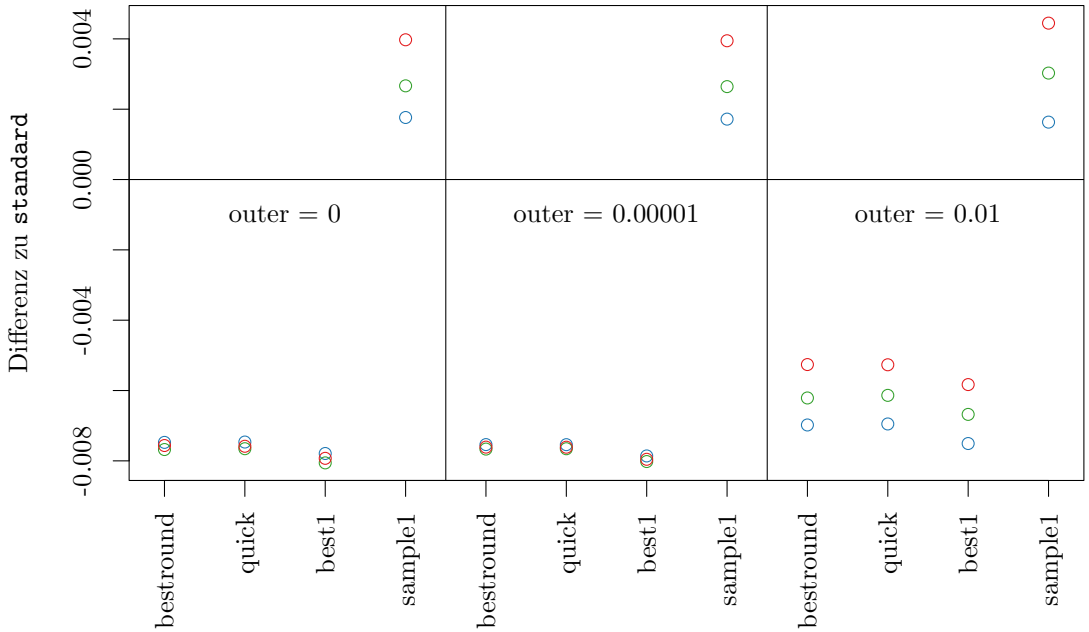

Abbildung 5.8: Differenzen der Quartile der geschätzten Standardabweichungen der Recall-Schätzer zur `standard`-Methode für das anwendungsnahe Szenario. Pro Subkorpus wurden 20 Texte im Vorfeld gelabelt, in der äußeren Menge 50. Weitere 100 Texte wurden nach der jeweiligen Strategie gezogen. Die Differenzen der 25 %-Quartile sind durch die blauen Punkte gekennzeichnet. Für die 50 %-Quartile sind die Punkte grün und für die 75 %-Quartile rot.

Rang	standard	bestround	quick	best1	sample1
1	6572	51246	51068	169139	43970
2	19114	103861	103637	49144	56148
3	36694	76894	76799	29770	55346
4	73062	40204	40385	27650	93842
5	164558	27795	28111	24297	50694

Tabelle 5.7: Ränge der absoluten Abstände zum wahren Wert für das anwendungsnahe Szenario zusammengefasst für die drei Einstellungen mit jeweils 100 000 Wiederholungen. Pro Subkorpus wurden 20 Texte im Vorfeld gelabelt, in der äußeren Menge 50. Weitere 100 Texte wurden nach der jeweiligen Strategie gezogen.

als das Vorgehen in der ersten Simulation, da oft bereits gelabelte Texte aus Vortests aus den einzelnen Subkorpora stammen. So ist die Auswahlwahrscheinlichkeit der Texte in den Schnittmengen größer, die zu mehreren Subkorpora gehören. Anschließend werden 100 Texte nach den jeweiligen Strategien gezogen und gelabelt. Abbildung 5.7 zeigt die Beanplots für die drei Szenarien mit unterschiedlichen Anteilen an relevanten Texten in der äußeren Menge. Sind in der äußeren Menge keine oder nur sehr wenige relevante Texte vorhanden, sind die drei alternativen Strategien deutlich überlegen. Hier liegen die Mediane unterhalb des 25 % Quartils der `standard`-Methode. Für höhere Anteile an relevanten Texten in der äußeren Menge sind die Unterschiede nicht so groß. Die `sample1` Strategie, die etwas mehr Variation in der Auswahl der Schnittmengen liefern soll, schneidet schlechter als die `standard`-Methode ab. In Abbildung 5.8 sind die Differenzen der Quartile zur jeweiligen `standard`-Methode abgetragen. Auch hier ist das schlechtere Abschneiden der `sampling1`-Methode deutlich zu sehen. Betrachtet man die Ränge der Methoden in den Simulationsdurchläufen (Tabelle 5.7), schneidet die `standard`-Methode erneut deutlich schlechter ab als die verbesserten Strategien.

Die Relevanz der Information aus der äußeren Schnittmenge

Für das Abschneiden der einzelnen Strategien sind insbesondere zwei Dinge besonders relevant: Das Vorwissen in den einzelnen Schnittmengen aufgrund von bereits durchgeführten Kodierungen und der Anteil der relevanten Texte in der äußeren Menge, die

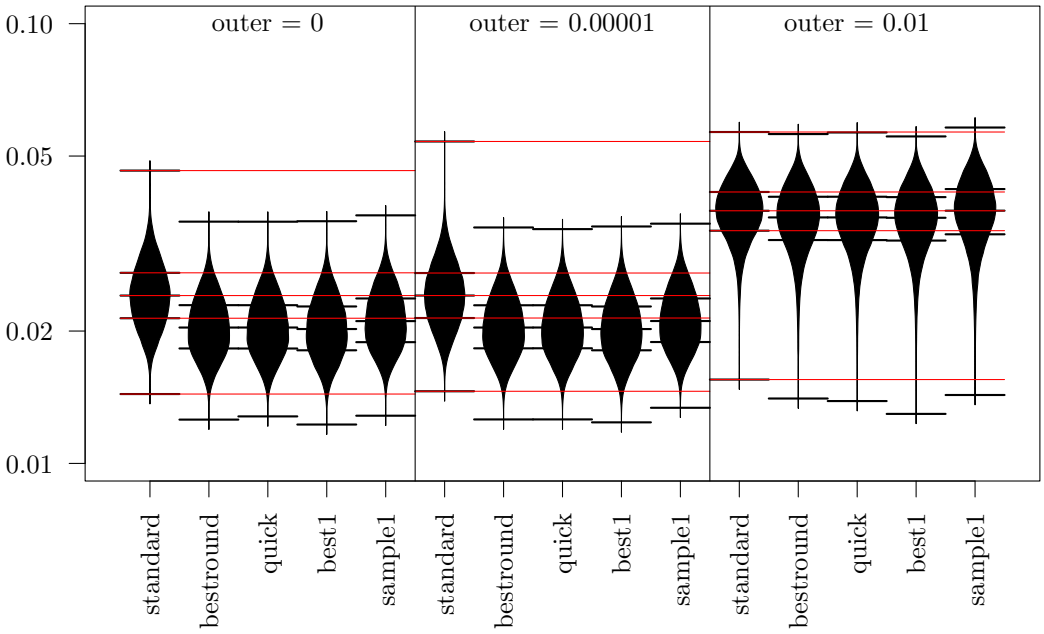

Abbildung 5.9: Trunkierte Beanplots der geschätzten Standardabweichungen der Recall-Schätzer für das anwendungsnahe Szenario auf einer log-Skala. Pro Subkorpus wurden 100 Texte im Vorfeld gelabelt, in der äußeren Menge 500. Weitere 200 Texte wurden nach der jeweiligen Strategie gezogen. Die roten Linien markieren die Quartile der `standard`-Methode.

Rang	standard	bestround	quick	best1	sample1
1	29510	65536	65572	65286	79849
2	38393	69491	69288	63720	60515
3	46570	63056	63055	64242	60351
4	60650	58978	58772	61347	57575
5	124877	42939	43313	45405	41710

Tabelle 5.8: Ränge der absoluten Abstände zum wahren Wert für das anwendungsnahe Szenario bei drei Einstellungen mit jeweils 100 000 Wiederholungen. Pro Subkorpus wurden 100 Texte im Vorfeld gelabelt, in der äußeren Menge 500. Weitere 200 Texte wurden nach der jeweiligen Strategie gezogen.

in der Regel deutlich größer als die übrigen ist. Auch wenn der Anteil der relevanten Texte in der äußeren Menge in der Regel sehr klein ist, da verschiedene Subkorpora so generiert werden, dass sie nach Möglichkeit alle relevanten Texte abdecken, können kleine Änderungen im Anteil schon größeren Einfluss auf den Recall-Schätzer haben, da die Schnittmenge den Großteil der Texte enthält. In dem hier betrachteten Simulationsbeispiel sind immer mindestens 75 % der Texte in dieser Menge. Bei vielen Kodieraufgaben kann man diese Texte oft deutlich leichter kodieren als die Texte der anderen Schnittmengen, da sie thematisch von der jeweiligen Fragestellung entfernt sind. Deshalb können hier in gleicher Zeit mehr Texte bearbeitet werden. Für die nächste Simulation wird ein solches Szenario betrachtet. In allen Subkorpora werden zu Beginn bereits 100 Texte ausgewertet, in der äußeren Menge sogar 500. Anschließend werden 200 Texte nach den jeweiligen Strategien gelabelt. In diesem Szenario existiert also bereits einiges Wissen aus gelabelten Texten und es können, im Vergleich zur vorherigen Simulation, noch doppelt so viele weitere Texte gezogen werden. Betrachtet man die Beanplots der Standardabweichungen für den Recall-Schätzer (Abbildung 5.9), so sieht man für die Einstellungen mit keinen oder sehr wenigen relevanten Artikeln in der äußeren Menge eine deutliche Verbesserung der Sampling-Methoden im Vergleich zum Standardvorgehen. Auch `sample1` schneidet hier deutlich besser ab. Erhöht man aber den Anteil der relevanten Texte in der äußeren Menge auf 1 %, so fällt die Verbesserung nicht mehr so deutlich aus. Das 75 %-Quartil der `sampling1`-Strategie ist sogar etwas größer als das der `standard`-Methode. Das Ranking der einzelnen Methoden (Tabelle 5.8) zeigt noch einmal, dass die `standard`-Methode nur in weniger als einem

Zehntel der Durchläufe auf dem ersten, dafür aber bei mehr als einem Drittel auf dem letzten Rang liegt. Während die anderen drei Methoden ungefähr gleich häufig auf dem ersten Rang platziert sind, verzeichnet `sample1` fast 14 000 Erstplatzierungen mehr als die zweitplatzierte Methode und liegt damit fast 80 000-mal auf dem ersten Platz. Reduziert man das Wissen in der äußeren Menge, indem man nur 100 Texte im Vorfeld labelt, fällt dieses Ergebnis in den Rängen noch deutlicher aus (siehe Tabelle C.8 im Anhang). Die Beanplots der Standardabweichungen liegen hier etwas näher beieinander, ohne dass sich die grundlegende Aussage verändert (siehe Abbildung B.4 im Anhang).

Wenig Vorinformation

In einem letzten Szenario soll betrachtet werden, wie die einzelnen Strategien sich verhalten, wenn wenig Vorinformation vorliegen. In der Praxis existieren für einzelne Subkorpora bereits Informationen aus Vortests. Es ist aber realistisch, dass für einzelne oder sogar alle Subkorpora keine Vortests gemacht wurden, sondern dirket mit dem Sampling nach einer der Strategien begonnen werden soll. Für diese Simulation wurden innerhalb der Schnittmengen keine Texte im Vorfeld gelabelt, einzig für die äußere Schnittmenge wurden 100 Texte gezogen, um für diese Menge schon etwas Vorinformationen zu haben. In jedem Simulationsschritt werden 500 Texte nach der jeweiligen Strategie gezogen. Abbildung 5.10 zeigt die Beanplots der Standardabweichungen für diese Simulation. Wie erwartet schneidet hier `best1` am besten ab. Da am Anfang für alle inneren Schnittmengen keine Informationen vorliegen, wird π_i mit 0.5 geschätzt. Der iterative Algorithmus kann diesen Wert nach jedem gelabelten Text anpassen und ist somit deutlich effizienter. Auch wenn `sample1` dies ebenfalls kann, schneidet sie im Vergleich zu `best1` erneut deutlich schlechter ab und ist eher mit den anderen Sampling-Strategien vergleichbar. Dies zeigen auch die Differenzen zu den Quartilen der `standard`-Methode in Abbildung 5.11. Betrachet man allerdings wieder das Ranking der Methoden in Tabelle 5.9, zeigt sich ein anderes Bild. Die `best1` Strategie ist in mehr als einem Drittel der Simulationen auf dem ersten Rang, auch `sample1` schneidet hier fast doppelt so gut wie die drittplatzierte Methode ab.

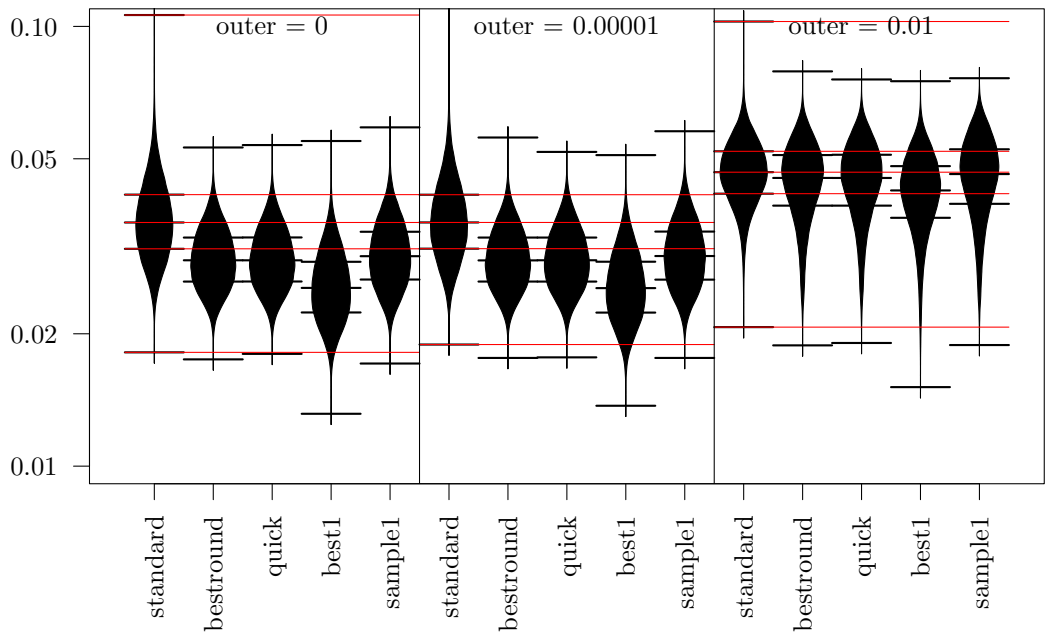

Abbildung 5.10: Trunkierte Beanplots der geschätzten Standardabweichungen der Recall-Schätzer für das anwendungsnahe Szenario auf einer log-Skala. Es wurden nur in der äußeren Menge im Vorfeld 100 Texte gelabelt, in den anderen Schnittmengen gibt es keine Vorinformation. Weitere 500 Texte wurden nach der jeweiligen Strategie gezogen. Die roten Linien markieren die Quartile der `standard`-Methode.

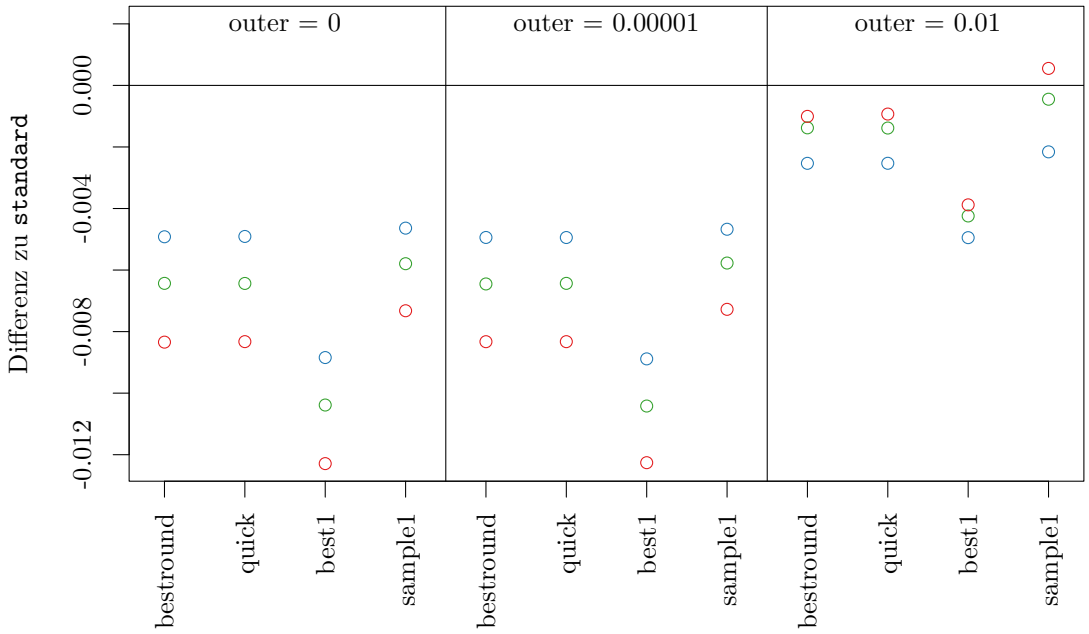

Abbildung 5.11: Differenzen der Quartile der geschätzten Standardabweichungen der Recall-Schätzer zur `standard`-Methode für das anwendungsnahe Szenario. Es wurden nur in der äußeren Menge im Vorfeld 100 Texte gelabelt, in den anderen Schnittmengen gibt es keine Vorinformation. Weitere 500 Texte wurden nach der jeweiligen Strategie gezogen. Die Differenzen der 25 %-Quartile sind durch die blauen Punkte gekennzeichnet. Für die 50 %-Quartile sind die Punkte grün und für die 75 %-Quartile rot.

Rang	standard	bestround	quick	best1	sample1
1	28577	40395	40262	115164	75604
2	38852	59547	58947	64474	78186
3	46480	72673	73210	48610	59027
4	60756	73277	73594	41450	50918
5	125335	54108	53987	30302	36265

Tabelle 5.9: Ränge der absoluten Abstände zum wahren Wert für das anwendungsnahe Szenario bei drei Einstellungen mit jeweils 100 000 Wiederholungen. Es wurden nur in der äußeren Menge im Vorfeld 100 Texte gelabelt, in den anderen Schnittmengen gibt es keine Vorinformation. Weitere 500 Texte wurden nach der jeweiligen Strategie gezogen. Die roten Linien repräsentieren die Quartile der `standard`-Methode.

Die Simulationen in diesem Abschnitt zeigen, dass die Sampling Strategien ohne Zufallskomponente der einfachen Zufallsstichprobe überlegen sind. Beim Ziehen mit Zufallskomponente ist diese Aussage nicht ganz so eindeutig. Die exakte Lösung `bestround` ist deutlich rechenaufwendiger als die Näherung `quick`, insbesondere bei vielen Schnittmengen. Die Ergebnisse beider Methoden sind allerdings sehr ähnlich, sodass im Folgenden nur noch `quick` betrachtet wird.

5.2.3 Betrachtung der Konfidenzintervalle

Im vorherigen Abschnitt wurde für eine gegebene Anzahl von zu ziehenden Texten verglichen, wie gut die einzelnen Sampling Methoden sind. Dieses Vorgehen entspricht durchaus der gängigen Praxis, in der meist vom Anwender eine zu bewältigende Zahl an zu kodierenden Texten festgelegt wird. Aus statistischer Sicht ist es hingegen sinnvoller, die Zahl der zu kodierenden Texte anhand einer akzeptierten Größe des Konfidenzintervalls des Schätzers zu bestimmen.

Für diese Analyse werden zwei Einstellungen aus dem vorherigen Abschnitt erneut betrachtet. Dies ist zum einen die realitätsnahe Simulation mit 20 (beziehungsweise 50 in der äußeren Menge) im Vorfeld gelabelten Texten pro Subkorpus und 100 Texten nach der jeweiligen Strategie. Außerdem wird erneut die Einstellung mit wenig Vorinformationen (nur 100 Texte in der äußeren Menge im Vorfeld, dafür 500 nach

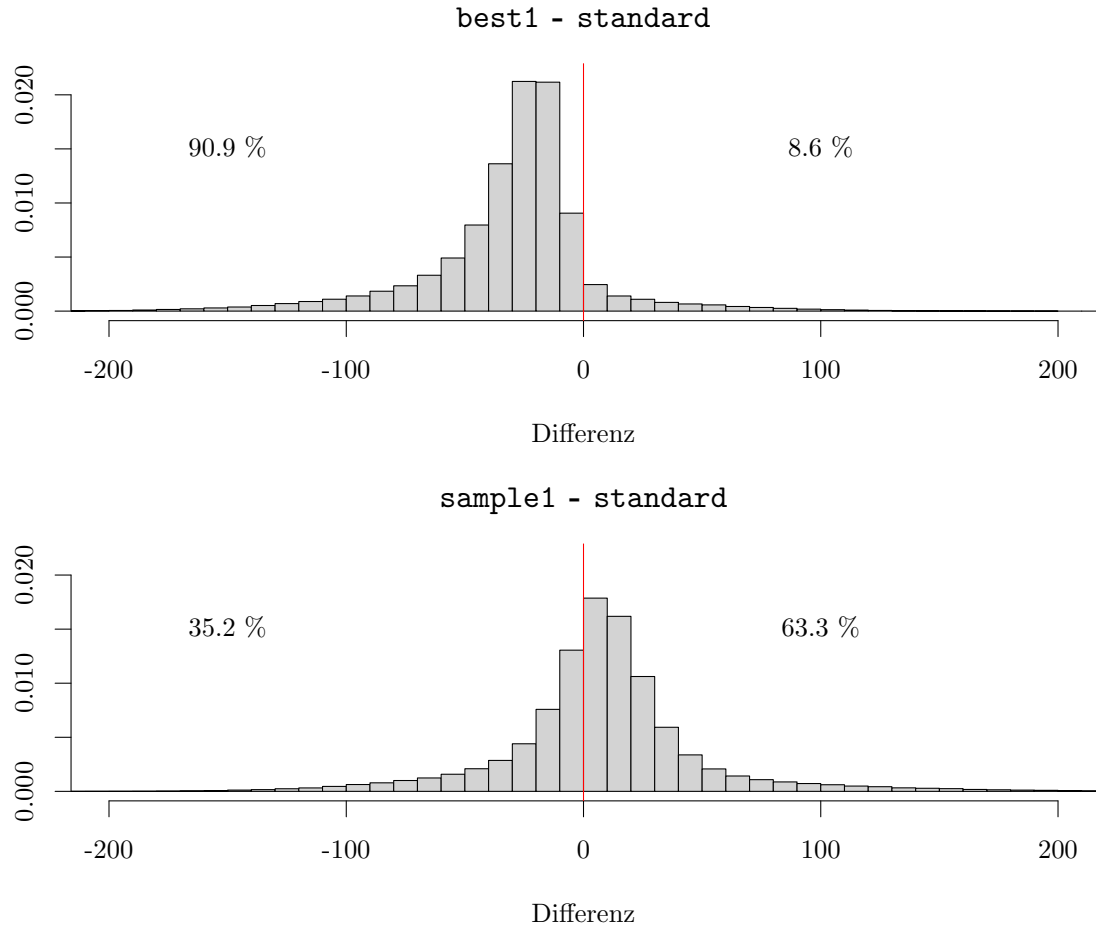

Abbildung 5.12: Trunkierte Histogramme der Differenzen zur `standard`-Methode in der Anzahl der benötigten Texte im Szenario mit 20 gelabelten Texten pro Subkorpus und 50 in der äußeren Schnittmenge. Bei positiven Werten benötigt die `standard`-Methode weniger Texte. Die Prozentangaben addieren sich nicht zu 100, weil die Fälle gibt, in denen beide Methoden gleich viele Texte benötigen.

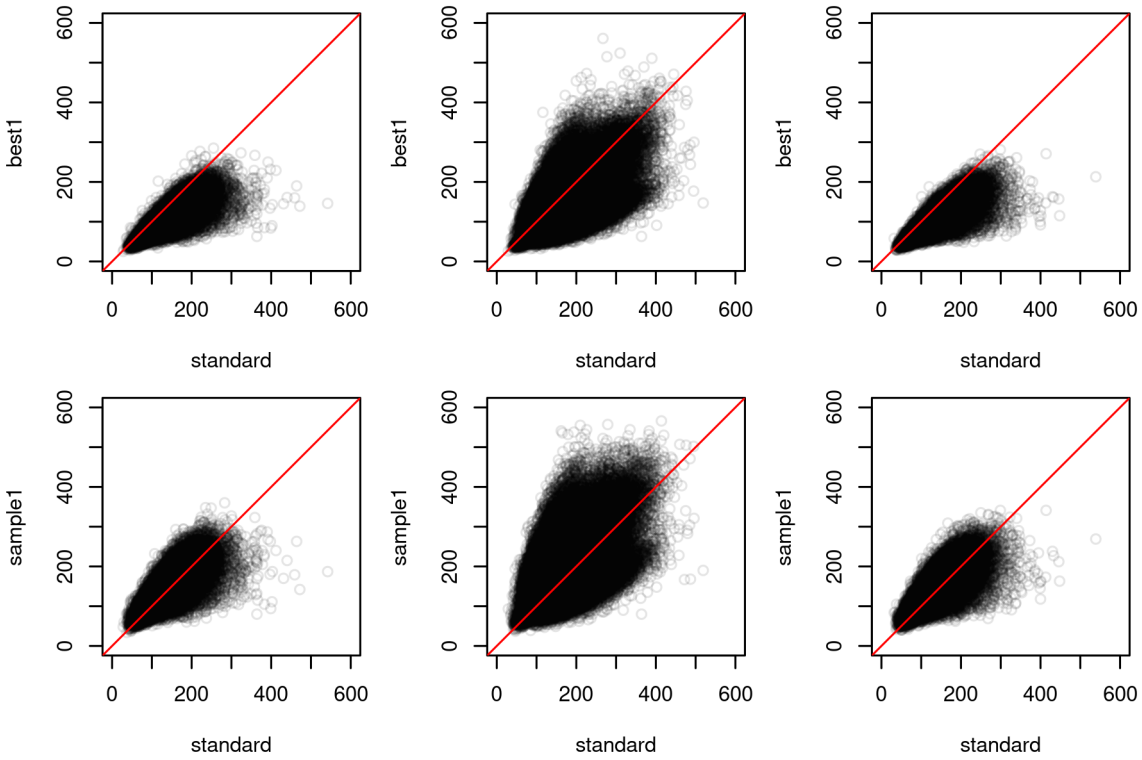

Abbildung 5.13: Scatterplots der benötigten zusätzlichen Texte der drei Methoden für ein Konfidenzintervall von 2 Prozentpunkten beim Recall-Schätzer im Szenario mit 20 gelabelten Texten pro Subkorpus und 50 in der äußeren Schnittmenge. In den drei Spalten sind jeweils die Simulationen mit 0 %, 0.001 % und 1 % relevanten Texte in der äußeren Schnittmenge abgebildet.

den Strategien) verwendet. Da jetzt nach jeder Iteration überprüft werden soll, ob die Länge des Konfidenzintervalls bereits unter 2 Prozentpunkte gesunken ist, werden hier nur die iterativen Verfahren `best1` und `sample1` mit der `standard`-Methode verglichen. In 100 000 Simulationen wird für alle Methoden die Zahl der benötigten Texte festgehalten. Die im Vorfeld gelabelten Texte werden dabei nicht einbezogen. Abbildung 5.12 zeigt die Ergebnisse der ersten Simulation. Die beiden Grafiken zeigen für die beiden Methoden die Histogramme der Differenzen der benötigten Texte zur `standard`-Methode. Während die `best1`-Methode in 90.9 % der Fälle weniger Texte als die `standard`-Methode benötigt, sind es bei `sample1` nur 35.2 %. In diesem Szenario würde das Verwenden einer Zufallsauswahl, um zu verhindern, dass lange aus der gleichen Schnittmenge gezogen wird, meistens mehr Texte benötigen als eine einfache Zufallsauswahl. Im Median werden in diesem Szenario mit `best1` 24 Texte weniger und mit `sample1` 8 Texte mehr als die `standard`-Methode benötigt. In Abbildung 5.13 sind die bei `best1` und `sample1` benötigten Texte gegen die von der `standard`-Methode benötigten Texte aufgetragen. Hierbei werden nicht die Texte betrachtet, die bereits initial ausgewertet wurden. Deutlich zu erkennen ist, dass für die Vergleiche mit `best1` deutlich mehr Punkte unterhalb der Winkelhalbierenden liegen und damit im Vergleich zur `standard`-Methode weniger Texte benötigt werden als mit `sample1`.

Das Ergebnis sieht anders aus, wenn man das Szenario mit wenig Vorinformation betrachtet (siehe Abbildung B.5 und B.6 im Anhang). Hier benötigen 95 % der `best1` Durchläufe weniger Texte als die einfache Zufallsauswahl und 82 % der `sample1` Durchläufe. In diesem Szenario bringen also beide Methoden einen Vorteil gegenüber der einfachen Zufallsauswahl. Im Median werden hier mit `best1` 93 Texte weniger als bei der `standard`-Methode benötigt. Bei `sample1` sind es 60 Texte.

5.2.4 Bewertung der verschiedenen Sampling-Methoden

Die Inhaltsanalyse von Texten ist ein iterativer Prozess. Um die richtigen Texte für die Analyse auszuwählen, ist das Überprüfen anhand kleinerer Stichproben notwendig. Die in diesem Kapitel vorgestellten sampling-Methoden können diese bereits vorhandenen

Informationen durch einen gewichteten Recall-Schätzer verwenden. Die angestellten Simulationen zeigen deutlich die Vorteile einer geschickteren Auswahl der zu kodierenden Texte gegenüber der einfachen Zufallsauswahl. Mit den beiden Strategien `bestround` und `quick` kann eine fest gewählte Anzahl an Texten auf die Schnittmengen verteilt werden. Da die Aufteilung bei `quick` stets identisch oder sehr ähnlich zu der optimalen Berechnung bei `bestround` ist, lohnt sich der Rechenaufwand für die exakte Lösung in der Praxis nicht. Die Rechenzeit potenziert sich bei der Berechnung der optimalen Lösung, während sie bei der Näherung linear wächst. Bereits bei der Verwendung von mehr als drei Subkorpora ist dies anhand der Rechenzeit deutlich zu spüren.

Der Ansatz, direkt die Aufteilung für m Texte zu wählen, ist nur dann sinnvoll, wenn eine iterative Lösung nicht umsetzbar ist. Dies kann zum Beispiel der Fall sein, wenn mehrere Kodierer*innen unabhängig voneinander ihre Texte kodieren sollen. Wenn es möglich ist, die Information eines gelabelten Texts direkt für die Auswahl des nächsten Texts zu verwenden, ist diese Methode vorzuziehen. Die Strategie `best1`, bei der iterativ immer aus der Schnittmenge mit der größten zu erwarteten Varianzreduktion gezogen wird, liefert über alle Simulationen die besten Ergebnisse. Sind die Varianzen der Schätzer innerhalb der Schnittmengen sehr unterschiedlich, kann es hier allerdings passieren, dass häufig hintereinander aus der gleichen Schnittmenge gezogen wird und ein menschlicher Kodierer das System erkennt und sich dadurch beeinflussen lässt. Man kann versuchen diesem damit entgegenzusteuern, indem nicht aus der Schnittmenge mit der größten erwarteten Varianzreduktion gezogen wird, sondern zufällig eine Schnittmenge gezogen wird, wobei die erwartete Varianzverbesserung dabei als Gewicht dient. Dies führt allerdings bei einigen Anwendungsszenarien dazu, dass die Ergebnisse schlechter werden als unter einer einfachen Zufallsstichprobe. Um diesem Effekt entgegenzuwirken wäre eine abgeschwächte Version möglich, bei der die Gewichte hin zu den Schnittmengen mit der größten erwartbaren Varianzreduktion verändert werden. Die Methoden `quick`, `best1` und `sample1` stehen als Funktion im R-Paket tosca zur Verfügung.

Die Analysen in diesem Kapitel beschränkten sich alle auf den Recall-Schätzer eines Subkorpus. Dies ist aus Sicht des Anwenders durchaus sinnvoll, da in der Regel der Recall des aktuellen Subkorpus berechnet wird und nur die Daten aus vorhergegangenen

Stichprobenziehungen mit anderen Subkorpora genutzt werden sollen. Soll der Recall von verschiedenen Subkorpora direkt verglichen werden, könnten die hier vorgestellten Verfahren in dem Sinne erweitert werden, dass nicht nur der Recall eines Subkorpus betrachtet wird, sondern die Werte von mehreren oder allen Subkorpora. Die Berechnung des Recalls bildet nur eine der gewünschten Eigenschaften der Schnittmengen ab und muss bei der Bewertung dieser um die Precision ergänzt werden. Da die Precision nicht über alle Schnittmengen eines Korpus berechnet wird, sondern nur aufgrund der zum interessierenden Subkorpus gehörenden, eignet sich diese Maßzahl nicht zur Optimierung der Zufallsauswahl. Um sie dennoch zu berücksichtigen, könnte statt des Recalls der F1-Score, also das harmonische Mittel zwischen Recall und Precision (Rijsbergen, 1979, Chinchor, 1992) verwendet werden.

6 Topic Coherence als Kriterium für die Modellwahl

In den vorangegangenen Kapiteln wurden bereits Themenmodelle an verschiedenen Stellen verwendet. Da die hier verwendete Latent Dirichlet Allocation, wie die meisten Themenmodelle zu den unüberwachten Verfahren gehört und die Ergebnisse stark von gewählten Parametern abhängen, muss das „richtige“ Modell gewählt werden. In Abschnitt 4.3.3 wurde bereits die Implementierung der Intruder Topics und der Intruder Words vorgestellt. Diese Verfahren haben den Nachteil, dass sie nicht automatisiert auf viele Modelle angewendet werden können. In diesem Kapitel wird die Topic Coherence für eine Vorauswahl der Modelle verwendet. Dazu werden im ersten Abschnitt zusammenfassende Maßzahlen auf Modellebene verglichen. Im zweiten Abschnitt wird das Verhalten der Topic Coherence auf Themenebene untersucht und abschließend auf einen synthetischen Datensatz angewendet.

6.1 Vergleich der Topic Coherence Varianten

In diesem Abschnitt werden die verschiedenen Varianten der Topic Coherence anhand von LDA-Modellen auf dem durch den umfangreicheren Wortfilter kleinen Banken-Korpus aus Abschnitt 4.2 verglichen. Dazu werden zuerst die gerechneten Modelle vorgestellt.

6.1.1 Die verwendeten Modelle

Für die Optimierung der LDA stehen drei Parameter zur Verfügung. Mit der Anzahl der Themen k kann zum einen darauf eingegangen werden wie divers der Korpus ist und zum anderen wie spezifisch die einzelnen Themen sein sollen. Die Hyperparameter

α und η dienen dem Modell als Parameter für die Dirichlet-Verteilungen, die die Verteilung der Themen pro Text (α), beziehungsweise die Verteilung der Wörter pro Thema (η) definiert. Darüber hinaus sind die einzelnen Durchläufe aufgrund des Optimierungsverfahrens vom Wert des Zufallszahlengenerators abhängig.

Für den Parameter η wird hier die gängige Vorgehensweise (Maier u. a., 2018) gewählt, die mit $\eta = 1/k$ den Parameter abhängig von der Zahl der Themen wählt. Für α und k werden verschiedene Werte gewählt. Für den Hyperparameter der Themenverteilung auf Textebene wird $\alpha \in \{0.01, 0.1, 0.5, 1, 2\}$ gewählt, für die Anzahl der Themen werden die Werte $k \in \{5, 10, 20, 30, 50, 75, 100, 125, 150, 200\}$ verwendet. Für alle Kombinationen von α und k werden jeweils fünf Modelle gerechnet.

6.1.2 Berechnung der Topic Coherence

Neben der klassischen Definition der Topic Coherence (Mimno u. a., 2011) werden noch drei weitere Varianten berechnet. In der ersten Variante wird die Topic Coherence in einer symmetrischen Variante berechnet (wie in Abschnitt 3.3.3 vorgestellt), in der zweiten werden die gewichteten Topwörter wie in Abschnitt 3.3.2 beschrieben verwendet. Die dritte Variante verbindet diese beiden Ansätze, sodass die gewichteten Topwörter in der symmetrischen Formel verwendet werden. Um einen Wert für die Topic Coherence auf Modell-Ebene zu erhalten wurden verschiedene zusammenfassende Maßzahlen verwendet. Neben dem arithmetischen Mittel der Topic Coherence aller Themen in einem Modell wurde auch der Wert des oberen Quartils (75 %) und das arithmetische Mittel über die besten 50 % der Themen (im Sinne der Topic Coherence) berechnet. Abschließend wird auch noch der beste Wert der Topic Coherence in einem Modell als Repräsentant für das ganze Modell betrachtet.

Die Abbildung 6.1 zeigt für alle oben beschriebenen Modelle die Werte der Topic Coherence im symmetrischen Fall unter Verwendung der gewichteten Topwörter. Die Abbildungen B.7, B.8 und B.9 im Anhang zeigen jeweils die Ergebnisse für die anderen Varianten der Topic Coherence. Alle vier Grafiken zeigen ein ähnliches Ergebnis. Für die gewählten Parameter ergeben kleinere Werte für k unter Festhalten von α bessere Werte für die Topic Coherence. Bei festem k schneiden wiederum kleine Werte für

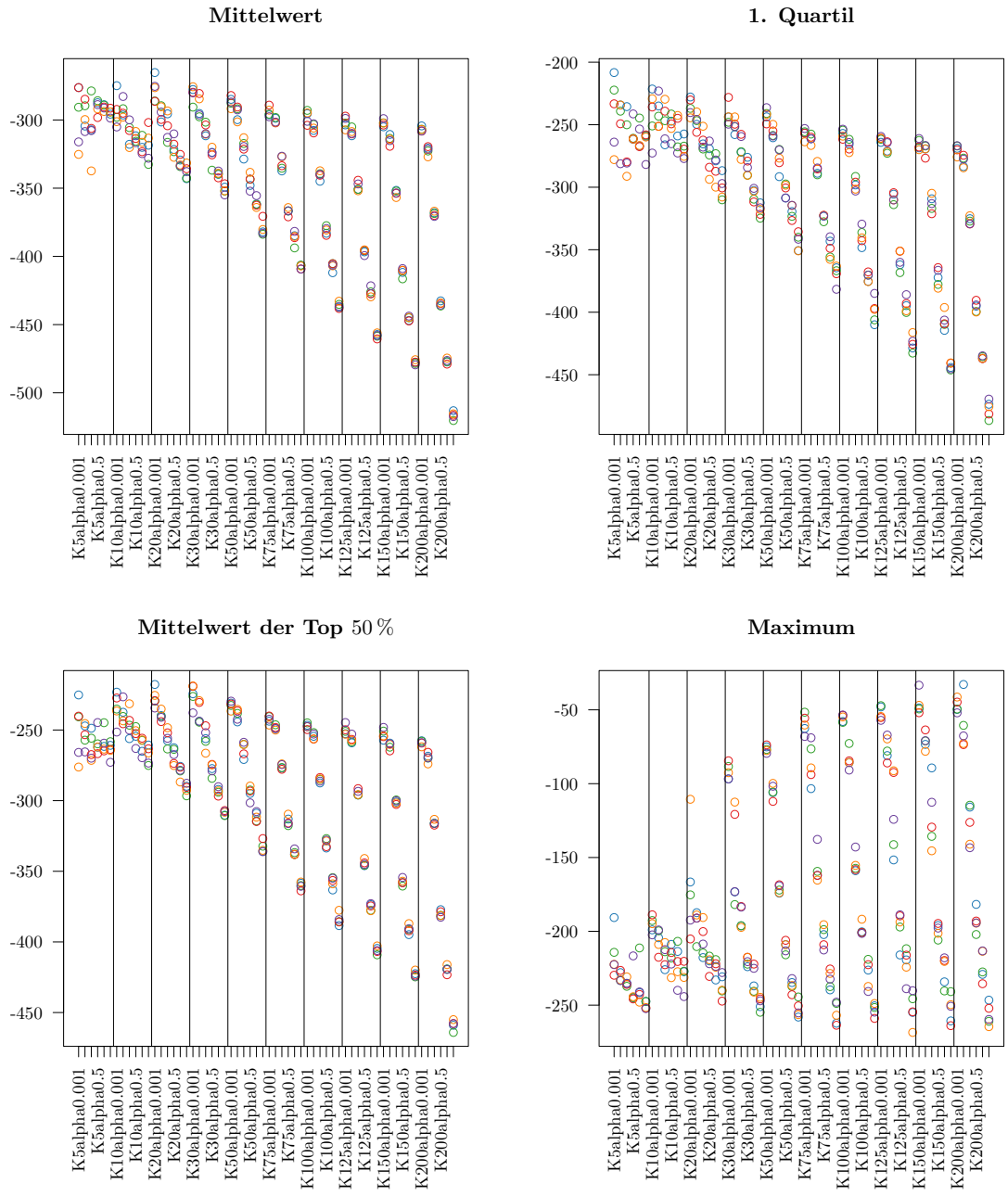

Abbildung 6.1: Verschiedene Maßzahlen für die symmetrische Topic Coherence unter Verwendung der Topwörter. Die jeweilige Maßzahl wurde über die Topic Coherencen der einzelnen Themen im Modell berechnet.

α besser ab. Je größer k gewählt wird, desto weiter streuen die Maßzahlen für die verschiedenen Werte von α. Diese Beobachtungen gelten für die beiden Mittelwerte (alle Themen und nur die besten 50 %) und das obere Quartil. Beim Maximum werden für größeres k bessere Werte erreicht, die übrigen Beobachtungen treffen auch hier zu. Dieses Verhalten der Topic Coherence konnte auch auf anderen Datensätzen reproduziert werden (zum Beispiel auf dem bioRχiv-Korpus der auf `www.biorxiv.org` veröffentlichten Abstracts). Für die Wahl der Parameter eignet sich die Topic Coherence in dieser Weise nicht, da es aus Anwendungssicht nicht sinnvoll ist immer das Modell mit den wenigsten Themen zu wählen.

Bei der Auswahl der Wiederholungen bei gleicher Parameterwahl kann die Topic Coherence allerdings eingesetzt werden. Die Varianten der Maßzahl bewerten die Modelle durchaus unterschiedlich und das Ergebnis hängt auch von der Art der Zusammenfassung der Topic Coherence Werte ab. Für diese Arbeit wird im Folgenden die Variante gewählt, bei der die gewichteten Topwörter verwendet werden, da diese im Projekt allgemein als aussagekräftiger im Vergleich zu den wahrscheinlichsten Wörtern angesehen werden. Zusätzlich wird die symmetrische Variante der Topic Coherence verwendet. Im folgenden Abschnitt wird die Topic Coherence nicht auf Modell- sondern auf der Themenebene untersucht und dabei eine alternative Modellwahl entwickelt.

6.2 Untersuchung der Eigenschaften auf Themenebene

6.2.1 Topic Coherence auf Themenebene

Für die Analyse eines Themenmodells sind nicht alle Themen des Modells gleich relevant. Auch bei einem im Vorfeld gut eingeschränkten Korpus sind immer Themen im Modell, die für die eigentliche Forschungsfrage nicht relevant sind. Dies sind zum einen die bereits erwähnten Stoppwort-Themen, die keine inhaltliche Information enthalten. Zusätzlich enthalten die Korpara auch immer „echte“ Themen, die für die Fragestellung nicht relevant sind. Bei der Wahl des Themenmodells kann es also sinnvoll sein nicht auf die Gesamtqualität des Modells zu achten, sondern vielmehr die Qualität des Modells anhand der für die Forschungsfrage relevanten Themen

zu bestimmen. Dazu soll in Abschnitt 6.2.2 in einem ersten Schritt das Vorgehen veranschaulicht werden. In Abschnitt 6.2.3 wird die Modellwahl am Banken-Korpus durchgeführt.

6.2.2 Vergleich von Themenmodellen mit fünf Themen

Die Modellauswahl anhand von für die Forschungsfrage relevanten Themen setzt die Identifizierung dieser Themen voraus. Je größer die Zahl der zu vergleichenden Modelle wird und je mehr Themen diese beinhalten, desto unrealistischer ist die Themenzuordnung für alle Themen. Hier kann das in Abschnitt 4.3.2 vorgestellte Ähnlichkeitsmaß helfen. In diesem Abschnitt wird das Vorgehen an einer überschaubaren Anzahl an Themen und Modellen vorgestellt. Dazu werden aus den in Abschnitt 6.1.1 beschriebenen Modellen die fünf Wiederholungen mit fünf Themen und $\alpha = 0.01$ verwendet. Dazu wurden alle 25 Themen vom Autor, ohne vorherige Betrachtung der Ähnlichkeitsstruktur, gelabelt. Die Themen wurden anhand der Topwortlisten betrachtet. Dabei wurde darauf geachtet, dass Themenüberschriften nicht so speziell gewählt wurden, dass diese nur auf die konkrete Wortliste passen.

Abbildung 6.2 zeigt das Dendrogramm der 25 Themen aus den fünf Modellen. Erkennbar bilden die vom Autor gleich gelabelten Themen auch gemeinsame Cluster. Dabei ist die Ähnlichkeit der fünf Stoppwortthemen untereinander am größten. Allerdings sind diese Themen auch am wenigsten interessant. Während alle Modelle jeweils ein Thema „Landesbanken“, „Bankenkrise“ und „Finanzmarkt“ enthalten, gibt es beim fünften Thema Unterschiede. Hier enthalten drei Modelle das Thema „Kunden“ und jeweils ein Modell das Thema „US-Banken“ beziehungsweise „Banken“. Die beiden Banken-Themen werden im Dendrogramm sogar erst mit dem Finanzmarkt-Cluster verbunden, bevor dieses Cluster mit dem Kunden-Cluster verbunden wird.

Die hier verwendete Hellinger-Distanz kann verwendet werden, um die Themen von verschiedenen Modellen zu clustern. Diese Eigenschaft wird im nächsten Kapitel für eine halbautomatisierte Modellwahl benötigt. Für eine Untersuchung verschiedener Maßzahlen für die Ähnlichkeit von Themenmodellen sei an dieser Stelle auf Rieger u. a., 2021 verwiesen. Soll aus diesen fünf Modellen eines ausgewählt werden, liegt es am

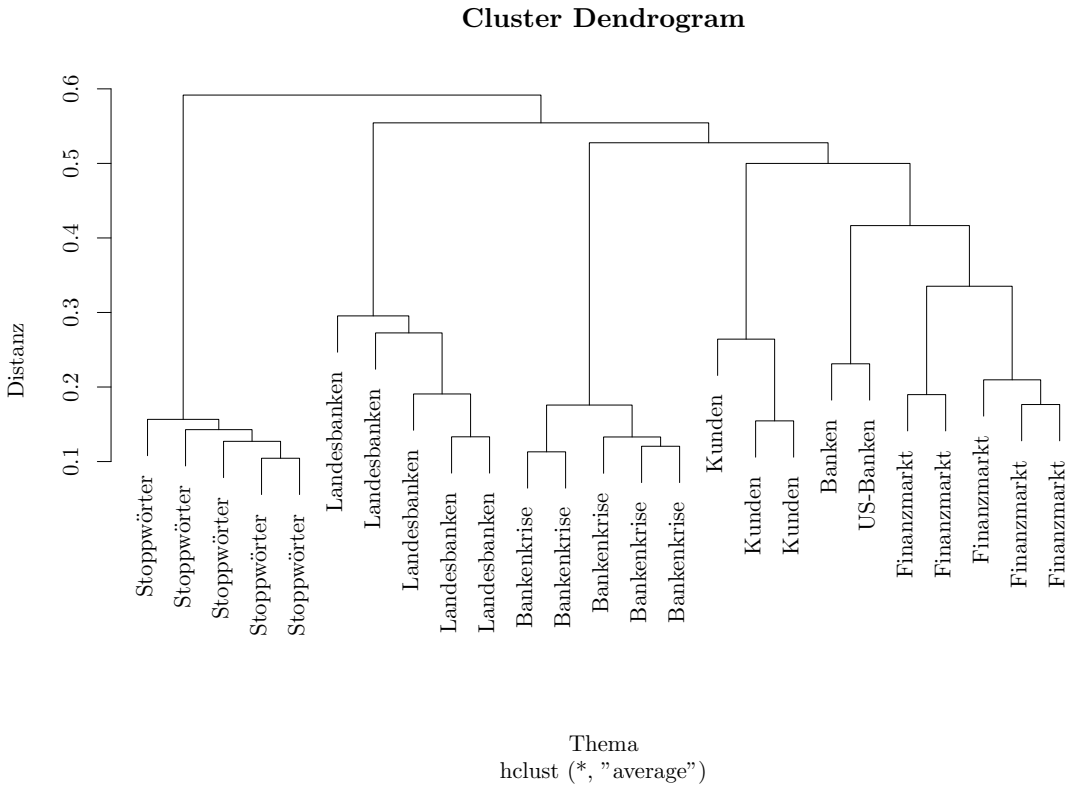

Abbildung 6.2: Dendrogramm einer hierarchischen Clusteranalyse mit average linkage der Themen in den fünf Modellen mit jeweils fünf Themen. Als Abstandsmaß wurde die Hellinger-Distanz verwendet.

konkreten Forschungsinteresse, welche Themen dabei betrachtet werden sollen. Tabelle 6.1 zeigt die Topic Coherence für jedes Thema in den fünf Modellen. Sortiert man diese Modelle nach der mittleren Topic Coherence (erste Spalte), so ist Modell `M3` den anderen überlegen, gefolgt von `M2`. Das Modell `M2` schneidet zu den nachfolgenden Modellen allerdings nur deswegen so gut ab, weil die Themen *Finanzmarkt* und *Stoppwörter* im Sinne der Maßzahl besonders gut sind. Da Stoppwort-Themen inhaltlich nicht relevant sind, sollten sie nicht in die Modellbewertung eingehen. Für eine Forschungsfrage, bei der die Bankenkrise betrachtet werden sollte, kann auch das Modell `M4` interessant sein, da es unter allen anderen Modellen die kleinste Topic Coherence im zugehörigen Thema besitzt. Während in diesem Beispiel mit wenigen Modellen und wenigen Themen alle Themen gelabelt werden konnten, ist dies für Modelle mit mehr Themen und/oder mehr Wiederholungen nicht mehr praktikabel. Hier können Ähnlichkeitsmaße helfen, eine halbautomatische Pipeline zu erstellen.

	Mittelwert	Banken	Bankenkrise	Finanzmarkt	Kunden	Landesbanken	Stoppwörter
M3	-284.81		-287.03	-226.46	-284.30	-249.24	-377.02
M2	-289.61		-368.07	-239.17	-299.12	-308.40	-233.29
M4	-299.70		-232.88	-451.02	-269.12	-234.26	-311.24
M1	-304.45	-431.38	-279.25	-227.97		-234.21	-349.46
M5	-308.52	-233.33	-386.57	-282.00		-359.48	-281.22

Tabelle 6.1: Topic Coherence der Themen in den fünf Modellen mit jeweils fünf Themen. Die erste Spalte zeigt das Gesamtmittel des Modells.

6.2.3 Modellwahl anhand von ausgewählten Themen

Soll die Modellwahl anhand bestimmter Themen in einem Umfeld mit deutlich mehr Themen erfolgen, ist ein angepasstes Vorgehen erforderlich, da eine Benennung aller Themen nicht mehr praktikabel ist. Auch hier ist aufgrund der Ergebnisse aus Abschnitt 6.1.2 darauf zu achten, dass nur Modelle mit gleicher Anzahl an Themen und gleichem α verglichen werden. Damit nicht alle Themen von menschlichen Kodierer*innen gelabelt werden müssen, werden alle Themen aus allen Modellen nach Topic Coherence geordnet in eine Tabelle geschrieben. So kann in der Tabelle nach relevanten Themen gesucht werden. Hier können verschiedene Vorgehen kombiniert werden. Da die Themen

nach der Topic Coherence sortiert sind, kann unter den besten Themen im Sinne der Topic Coherence gezielt nach interessanten Themen gesucht werden. Gleichzeitig kann aber auch, zum Beispiel über eine Wortsuche, in allen Themen nach speziellen Themen gesucht werden. Auch ein Blick auf für die Themen repräsentative Texte ist so möglich. Sind mehrere Themen relevant, was für viele Forschungsfragen der Fall ist, ist es sehr wahrscheinlich, dass die gefundenen „idealen“ Themen nicht aus einem einzigen Modell stammen und so nicht direkt genutzt werden können. Diese Themen können nun in einem zweiten automatisierten Schritt dazu verwendet werden in allen Modellen möglichst ähnliche Themen zu finden. Ausgewählt wird abschließend das Modell in dem alle relevanten Themen zusammen eine möglichst große Topic Coherence aufweisen.

Im folgenden werden die fünf Modelle mit 100 Themen und $\alpha = 0.01$ verwendet. Aus der nach Topic Coherence sortierten Liste der Themen wurde zu den drei Themen „Europäische Zentralbank“ (EZB), „Landesbanken“ und „US-Banken“ ein passendes Thema mit hoher Topic Coherence ausgewählt. Abbildung 6.3 zeigt die Distanzen dieser drei Themen zu allen anderen Themen. Zu allen drei Themen können in allen Modellen Themen gefunden werden, die eine gewisse Ähnlichkeit aufweisen. Betrachtet man die ähnlichen Themen für die „US-Banken“ in Tabelle 6.2 (die Tabellen zu den anderen beiden Themen sind C.11 und C.12 im Anhang) kann schnell überprüft werden, dass jedes der Modelle zwei ähnliche US-Banken-Thema enthält. Dabei ist ein Thema dem ausgewählten ähnlich, während das zweite Thema die Token *goldmann* und *sachs* enthält. Hier muss inhaltlich überprüft werden welches Thema das gewünschte ist, oder ob beide Themen zur Modellwahl verwendet werden sollen. Im letzteren Fall ist darauf zu achten, wie mit ähnlichen Themen umgegangen werden soll, da es bei ähnlichen Themen geschehen kann, dass in einem Modell ein Thema zu beiden Themen besonders ähnlich ist. Je nach inhaltlicher Entscheidung kann dieses Thema bei der Modellwahl dann einfach oder doppelt gewertet werden. Ein weiteres Problem zeigt sich bei Thema 33 aus Durchlauf 187. Hier ist ein Obama-Thema mit einer Distanz von 0.77 in der Nähe zu dem ausgewählten US-Banken-Thema. In Modellen in denen kein US-Banken-Thema existiert, könnte dieses (andere) Thema zugeordnet werden und die Modellwahl beeinflussen, da das Obama-Thema eine deutlich höhere Topic

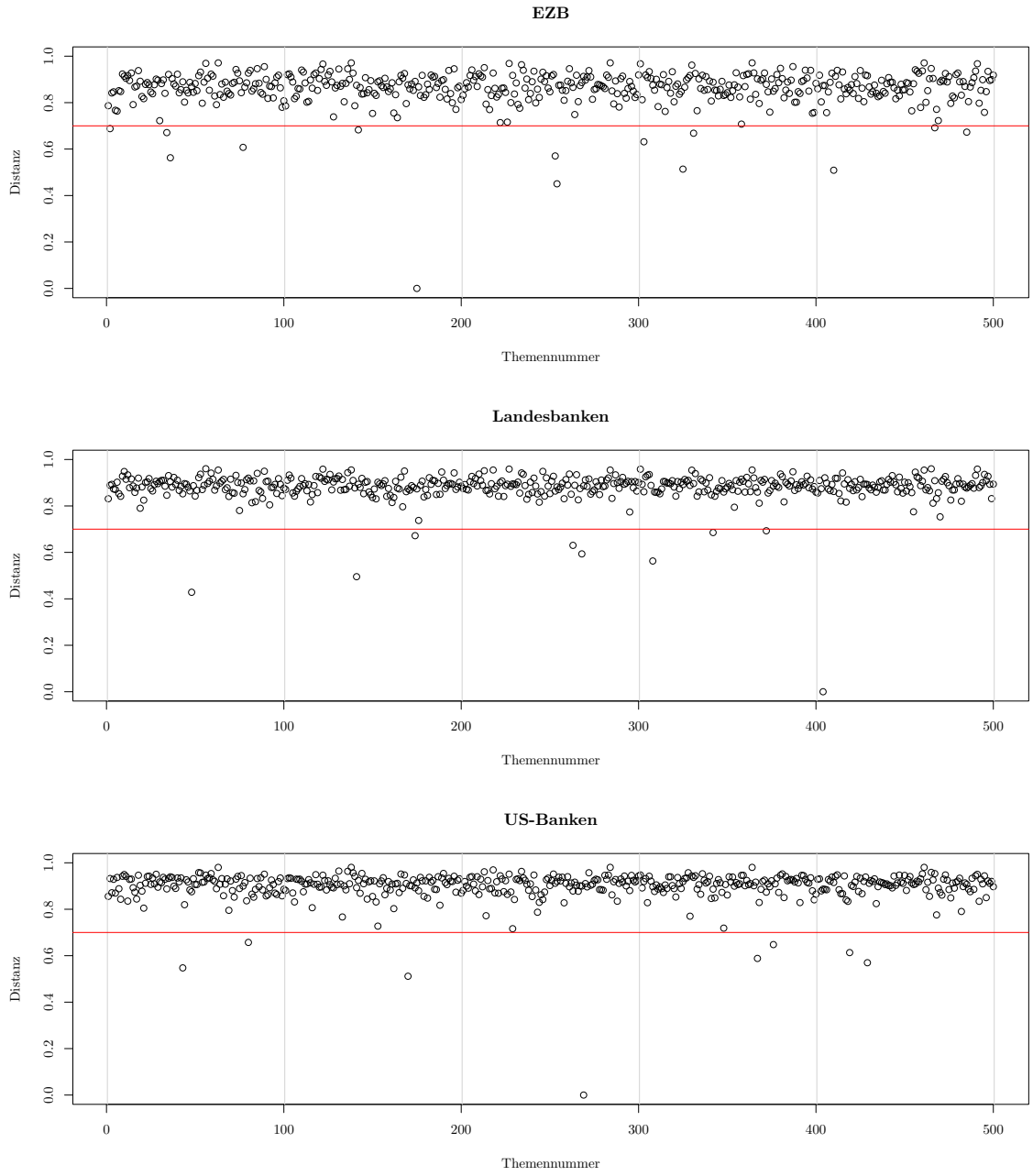

Abbildung 6.3: Distanzen aller Themen der fünf betrachteten Modelle zu den drei ausgewählten Themen. Die einzelnen Modelle sind durch die senkrechten Striche getrennt. Die Linie bei 0.7 dient als Lesehilfe.

Coherence als alle US-Banken-Themen besitzt. An dieser Stelle ist bei der Modellwahl eine Überprüfung durch Fachwissenschaftler*innen notwendig.

	Coherence	Distanz	Wort 1	Wort 2	Wort 3	Wort 4
d 188 t 69	-252.40	0.00	dollar	new	york	bank
d 187 t 70	-236.60	0.51	dollar	bank	morgan	milliarden
d 186 t 43	-240.20	0.55	dollar	goldman	bank	morgan
d 190 t 29	-289.30	0.57	new	dollar	goldman	york
d 189 t 67	-260.40	0.59	dollar	new	york	bank
d 190 t 19	-255.50	0.61	dollar	milliarden	bank	morgan
d 189 t 76	-275.70	0.65	dollar	milliarden	aig	regierung
d 186 t 80	-320.50	0.66	new	dollar	york	street
d 188 t 29	-296.60	0.72	goldman	new	sachs	street
d 189 t 48	-307.40	0.72	bank	milliarden	quartal	euro
d 187 t 53	-333.70	0.73	new	goldman	dollar	street
d 187 t 33	-216.40	0.77	obama	dollar	new	usa
d 188 t 14	-289.40	0.77	banken	dollar	milliarden	kredite
d 189 t 29	-335.00	0.77	dollar	millionen	firma	unternehmen
d 190 t 68	-320.10	0.78	milliarden	euro	bank	banken
d 188 t 43	-343.90	0.79	bank	dresdner	commerzbank	milliarden
d 190 t 82	-300.80	0.79	bank	deutsche	postbank	dresdner
d 186 t 69	-302.80	0.80	euro	millionen	milliarden	bank

Tabelle 6.2: Zum US-Banken-Thema ähnliche Themen t für die fünf untersuchten Durchläufe d 186 bis 190. Die erste Zeile enthält das ausgewählte Thema, die übrigen ähnliche Themen. Gezeigt werden jeweils vier für das Thema repräsentative Wörter.

Für die drei ausgewählten Themen wird nun wieder für jedes Modell ein passendes Thema automatisch zugeordnet. Tabelle 6.3 zeigt die Topic Coherence Werte für die drei Themen in den fünf verglichenen Modellen. Betrachtet man den Mittelwert aus den drei Werten für jedes Modell, ist hier der Durchlauf 187 der beste im Sinne der Maßzahl. Nur das Landesbanken-Thema hat hier nicht die beste Topic Coherence. Dieses Vorgehen kann dazu beitragen im Rahmen einer Inhaltsanalyse eine Modellwahl vorzunehmen und sowohl die Stärken der automatisierten Auswahl, wie auch der Bewertung der Themen durch Kodierer*innen zu nutzen. Im letzten Abschnitt soll nun das Verhalten der Topic Coherence auf einem simulierten Datensatz betrachtet werden.

	Mittelwert	EZB	Landesbanken	US-Banken
d 187	-210.35	-202.17	-192.34	-236.55
d 188	-218.31	-214.06	-188.41	-252.45
d 190	-228.20	-208.03	-187.24	-289.34
d 189	-237.16	-257.32	-193.78	-260.37
d 186	-237.18	-241.23	-230.14	-240.18

Tabelle 6.3: Topic Coherence der Themen in den fünf Modellen mit jeweils 100 Themen. Die erste Spalte zeigt das Mittel der ausgewählten drei Themen.

6.3 Untersuchung der Topic Coherence auf einem simulierten Korpus

In Abschnitt 6.1.2 wurde gezeigt, dass die Topic Coherence in der Anwendung nicht geeignet ist die Wahl der Parameter bei der LDA zu unterstützen. Modelle mit wenigen Themen und kleinem α führten grundsätzlich zu höhere Werte in der Topic Coherence. Abschließend soll nun überprüft werden, wie sich die Topic Coherence in einer kontrollierten Situation in einem simulierten Datensatz verhält.

Als Vorbild für den simulierten Datensatz wird der oben verwendete Banken-Korpus verwendet. Die Anzahl der Dokumente im Korpus, die einzelnen Dokumentlängen und die Anzahl der Token im Vokabular werden identisch zum Banken-Korpus gewählt. Mit diesen Daten wird die LDA als generierendes Modell genutzt. Dabei werden 30 Themen und $\alpha = 0.1$ gewählt. Die Themenverteilungen werden dirichletverteilt gezogen. Anschließend wird für jedes Dokument eine Themenverteilung gezogen, aus der anschließend für jedes Token ein Thema und aus der zugehörigen Themenverteilung das Wort gezogen wird.

Der so generierte Datensatz unterscheidet sich von realen Korpora in der Hinsicht, dass er die Modellannahmen der LDA einhält, während in realen Texten die Unabhängigkeit der Wortwahrscheinlichkeit gegeben das Thema nicht gegeben ist. Für diesen Datensatz werden die in Abschnitt 6.1.2 beschriebenen Modelle und die zugehörige Topic Coherence berechnet. Abbildung 6.4 zeigt die zusammenfassenden Maßzahlen der jeweils fünf Modelle für die verschiedenen Parametereinstellungen. Deutlich zu

sehen ist, dass die Mittelwerte der Topic Coherence für $k = 30$ die besten Ergebnisse bringen. Obwohl der Datensatz mit $\alpha = 0.1$ erzeugt wurde, geben aber auch hier kleinere Werte für α bessere Werte in der Topic Coherence.

Abschließend kann festgehalten werden, dass die Topic Coherence sich auch im simulierten Fall nicht so verhält wie es gewünscht ist. Zur Parameterwahl ist sie also nicht zu empfehlen. Eine Bewertung einzelner Themen unter gleicher Parameterwahl wie in Abschnitt 6.2.3 gezeigt, ist mit ihr aber durchaus möglich. Eine andere Möglichkeit zur Auswahl einzelner Modelle auf Basis der gesamten LDA mit allen Themen bietet Rieger, 2020.

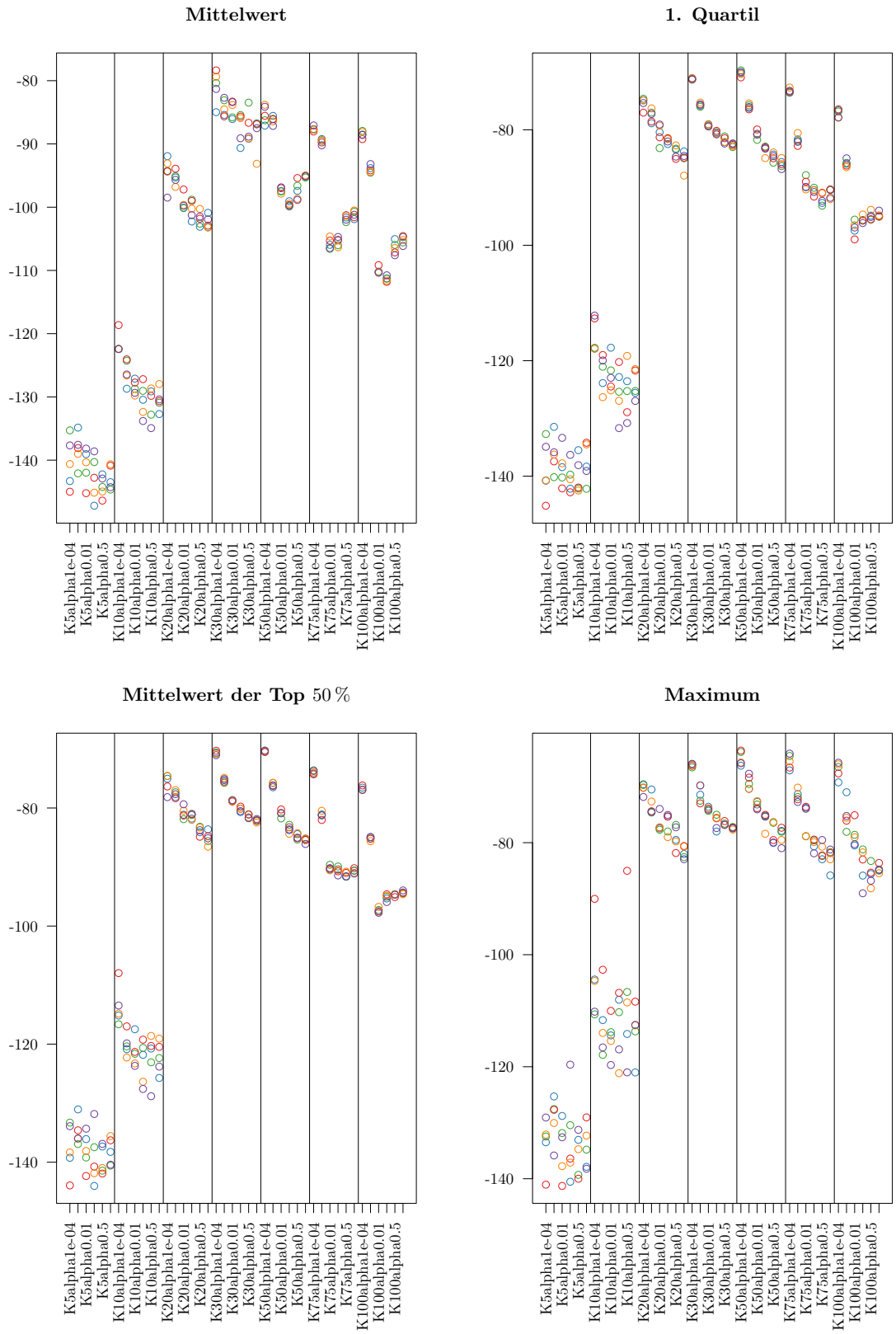

Abbildung 6.4: Verschiedene Maßzahlen für die symmetrische Topic Coherence unter Verwendung der Topwörter im synthetischen Datensatz. Die jeweilige Maßzahl wurde über die Topic Coherencen der einzelnen Themen im Modell berechnet.

7 Zusammenfassung und Ausblick

Diese Arbeit beschäftigt sich mit der Frage, wie geistes- und sozialwissenschaftliche Forschung mithilfe von Methoden aus den Digital Humanities verbessert werden kann. Dabei stehen drei Bereiche im Fokus: das eigens entwickelte R-Paket tosca, das Funktionen zur Analyse großer Textkorpora im Rahmen einer Inhaltsanalyse bündelt, eine angepasste Zufallsauswahl, die es verbessern soll, Subkorpora an Texten zu erstellen, und ein neuer Ansatz zur Bestimmung eines geeigneten Modells für die Auswahl von Themen. Dieses Kapitel fasst die Ergebnisse noch einmal zusammen und zeigt mögliche Ansatzpunkte für zukünftige Forschung auf.

7.1 tosca

Um Softwarepakete, wie tosca zu entwickeln, ist eine anwendungsbezogene Entwicklung Voraussetzung, wie sie im Rahmen von DoCMA stattfindet. Das im Rahmen von DoCMA entwickelte R-Paket tosca kann dabei helfen, im Rahmen einer Inhaltsanalyse häufig wiederkehrende Arbeitsschritte auch ohne ausgeprägte Programmierkenntnisse durchzuführen. Das Spektrum reicht dabei von der Datenvorverarbeitung und -visualisierung über das Berechnen von Themenmodellen bis hin zur Validierung der Ergebnisse. Dabei bietet tosca allerdings keine Softwarelösung, die durch eine grafische Benutzeroberfläche einfach zu bedienen wäre. Nutzer*innen müssen eine gewisse Bereitschaft zeigen sich mit Programmcode auseinanderzusetzen. Typische Arbeitsschritte können allerdings mithilfe von Code-Bausteinen zusammengefügt werden, sodass eine tiefere Kenntnis der Programmiersprache R nicht zwingend notwendig ist. Dieser Ansatz verlangt Nutzer*innen zwar einerseits eine längere Einarbeitungszeit ab, hat aber andererseits den Vorteil, dass das Programmpaket einfacher erweitert werden

kann, da der gesamte Bereich der Frontend-Entwicklung entfällt. Das R-Paket tosca wird laufend weiterentwickelt und an neue R-Versionen angepasst, sodass auch eine weitere Nutzung gewährleistet bleibt. Rapid-Prototyping erlaubt das einfache und schnelle Erweitern des Funktionsumfangs auch für Projekte außerhalb von DoCMA.

Auch wenn tosca bereits Funktionen enthält die textmeta-Objekte in Formate umwandeln, die in quanteda oder im tidyverse genutzt werden, so ist eine nähere Anbindung dort in Zukunft wünschenswert. Gerade im Bereich der statistischen Anwendung sowie in diesem Fall in der Wissenschaftskommunikation und im Wissenschafts-/Datenjournalismus werden die Pakete des tidyverse häufig genutzt. Insbesondere kann eine alternative Grafik-Ausgabe als ggplot2-Grafik (Wickham, 2016) ein lohnendes Zukunftsprojekt sein.

7.2 Qualitätsbewertung von Subkoropra

Das Erzeugen eines Subkorpus mit den für die Forschungsfrage relevanten Texten ist ein wichtiger Schritt im Rahmen einer Inhaltsanalyse, da vom erstellten Subkorpus die Qualität der weiteren Analysen abhängt. Um Precision und Recall zu bestimmen ist in der Regel die Bewertung einer Zufallsstichprobe durch menschliche Kodierer*innen notwendig. Dieser Schritt ist arbeitsintensiv, da häufig mehrere Anläufe und damit auch mehrere Anläufe benötigt werden um eine zufriedenstellende Filterregel zu erstellen. Im Rahmen einer statistischen Beratung ist die Frage „Wie viele Texte muss ich denn unbedingt kodieren um zuverlässige Ergebnisse zu bekommen?“ eine häufig gestellte. In dieser Arbeit wird ein angepasstes Vorgehen vorgestellt, das die Zahl der zu kodierenden Texte reduzieren kann.

Da der interessierende Subkorpus in der Regel um Größenordnungen kleiner als der Gesamtkorpus ist, werden häufig aus dem Subkorpus und dem übrigen Korpus getrennte Stichproben gezogen, die bei einer Veränderung des Subkorpus nicht mehr verwendbar sind. Der in dieser Arbeit vorgeschlagene gewichtete Schätzer zur Bestimmung des Recalls ermöglicht auch eine Verwendung der für andere Subkorpusgrößen gemachten Kodierungen indem für alle Schnittmengen gewichtete Einzelschätzer berechnet werden.

Daraus ergibt sich die Möglichkeit für alle Schnittmengen die Varianz des Schätzers zu berechnen. Eine weitere Zufallsauswahl kann so aus den Schnittmengen gezogen werden, dass die Varianz des Gesamtschätzers minimiert wird. Die verschiedenen denkbaren Vorgehen haben alle Vor- und Nachteile, sodass eine Auswahl der konkreten Situation angepasst werden sollte.

Insgesamt kann die Entscheidung auf drei Ebenen erfolgen:

- Die exakte Berechnung der idealen Aufteilung hat in der Praxis keinen Vorteil gegenüber der Näherung, weshalb grundsätzlich die Näherung verwendet werden soll.

- Die schrittweise Zufallsauswahl bietet insbesondere dort Vorteile wo noch wenig Wissen über die Schnittmengen durch vorhergegangene Kodierungen vorliegt. Der Nachteil dabei ist, dass der Vorgang des Kodierens häufig an mehrere Hilfkräfte ausgegeben wird und die Ausgabe einer Liste der zu kodierenden Texte üblich und logistisch einfacher ist. Auch wenn im Rahmen dieses Projekts bereits R-Code geschrieben wurde der so ein iteratives Vorgehen ermöglicht, kann dies durchaus als Hürde betrachtet werden, da die Fähigkeit der Benutzung statistischer Software nicht als vorhanden angesehen werden kann. Ein weiteres Problem des schrittweisen Vorgehens ist das vorhersehbare mehrmalige Ziehen aus der gleichen Schnittmenge, was zu einer Beeinflussung der Kodierer*innen führen kann („Der letzte Text war auch schon negativ und dieser vermutlich aus der gleichen Schnittmenge.“).

- Um zu verhindern, dass zu oft Texte aus der gleichen Schnittmenge hintereinander gezogen werden, kann statt der Auswahl der optimalen Schnittmenge eine gewichtete Zufallsauswahl vorgenommen werden. Dieses Verfahren verschlechtert die Effektivität der Zufallsauswahl deutlich.

Da die Zufallsauswahl der Schnittmengen die Zahl der zu kodierenden Texte deutlich erhöht, ist die Anwendung in der Praxis nicht wünschenswert. Eine andere Möglichkeit eine Beeinflussung der Kodierer*innen zu vermeiden ist die strikte Abschirmung dieser gegenüber Zwischenauswertungen. Dies ist dann praktikabel, wenn die Kodierer*innen

nur in dieser Aufgabe am Projekt beteiligt sind. Kodieren auch die beteiligten Wissenschaftler*innen, muss zumindest die statistische Auswertung von den Kodierer*innen getrennt werden. Unabhängig von der statistischen Auswertung kann die Zahl der zu kodierenden Texte natürlich dadurch klein gehalten werden, dass vor einer aufwändigen Überprüfung der Qualität der Subkorpus-Auswahl die gewählten Filterregeln im Lichte der Anwendungswissenschaft kritisch hinterfragt werden.

7.3 Modellwahl

Die Auswahl des „richtigen“ Themen-Modells ist ein weiterer Faktor der eine Inhaltsanalyse beeinflusst. Für die Latent Dirichlet Allocation müss neben der Anzahl der Themen auch die prior für die Dirichlet-Verteilungen festgelegt werden, die die Verteilung der Themen beziehungsweise die Verteilung der Wörter innerhalb eines Themas modellieren. Da das Modell stochastisch ist, führen unterschiedliche Durchläufe darüber hinaus nicht zu exakt gleichen Ergebnissen. Da die mathematische Optimierung oft nicht aus Sicht der Anwendung zur optimalen Lösung kommt, werden oft sehr zeitaufwändige manuelle Verfahren wie Intruder Words und Intruder Topics verwendet. In dieser Arbeit wurde die Topic Coherence auf ihre Fähigkeit hin überprüft eine Modellwahl zu vereinfachen.

Während im synthetischen Datensatz eine Parameterbestimmung mit Hilfe der Topic Coherence mit nur kleinen Abweichungen funktioniert, ist diese in realen Datensätzen nicht möglich. Hier werden grundsätzlich Modelle mit wenigen Themen und kleinem α bevorzugt. Dies ist aus Sicht der Anwendung nicht sinnvoll. Sinnvoll kann eine Modellwahl anhand der Topic Coherence bei der Wahl zwischen mehreren Läufen der gleichen Modellparameter sein. Dabei muss die Bewertung des Modells nicht über alle Themen geschehen, sondern kann auf die für die zu beantwortende Fragestellung relevante Themen beschränkt werden.

Da die Bestimmung der Themen bei vielen Replikationen und Modellen mit vielen Themen sehr aufwändig ist, kann die Topic Coherence genutzt werden die Benennung der Themen effektiver zu gestalten. Eine Sortierung der Themen verschiedener Modelle

anhand der Topic Coherence ermöglicht die einfache Identifizierung der relevanten Themen. Dabei ist zwar zu erwarten, dass für die Forschungsfrage nutzbare Themen eine gute Topic Coherence aufweisen, umgekehrt gilt dies allerdings nicht, da auch Stopwort-Themen in der Regel gut abschneiden. Damit nicht für alle zu vergleichenden Modelle die relevanten Themen manuell identifiziert werden müssen können mit Hilfe einer nach Topic Coherence geordnete Themenliste auch nur besonders gute Themen festgelegt werden. Ähnliche Themen in anderen Modell-Durchläufen können dann automatisch anhand von Ähnlichkeitsmaßen bestimmt werden. Abschließend kann wieder mit Hilfe der Topic Coherence das Modell gefunden werden, das insgesamt die höchste Topic Coherence für die relevanten Themen aufzeigt.

Die Auswahl nur eines Modells vereinfacht die weitere Analyse, da so im Folgenden von einer „Wahrheit“ ausgegangen wird. Dieser Ansatz bildet allerdings die in dem Modell enthaltene Unsicherheit über die wahre Themenverteilung nicht ab. Ein weiterführender Ansatz wäre die Betrachtung mehrerer Modelle. Der Grad der Übereinstimmung dieser Modelle kann dann als Maß für die Unsicherheit verwendet werden.

Literaturverzeichnis

Asuncion, Arthur, Max Welling, Padhraic Smyth und Yee Whye Teh.
„On Smoothing and Inference for Topic Models“.
In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*.
UAI '09.
Arlington, Virginia, United States: AUAI Press, 2009,
Seiten 27–34.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller und Akitaka Matsuo.
quanteda: An R package for the quantitative analysis of textual data.
30.
2018,
Seite 774.
DOI: 10.21105/joss.00774.

Blei, David M. und John D. Lafferty.
„Correlated Topic Models“.
In: *Proceedings of the 18th International Conference on Neural Information Processing Systems*.
NIPS'05.
Cambridge, MA, USA: MIT Press, 2005,
Seiten 147–154.

Blei, David M. und John D. Lafferty.
„Dynamic topic models“.
In: *Proceedings of the 23rd international conference on Machine learning*.
ICML '06.
New York, NY, USA: Association for Computing Machinery, Juni 2006,
Seiten 113–120.
DOI: 10.1145/1143844.1143859.

Blei, David M., Andrew Y. Ng und Michael I. Jordan.
„Latent dirichlet allocation“.
In: *J. Mach. Learn. Res.* 3 (März 2003), Seiten 993–1022.

Boczek, Karin und Lars Koppers.
„What's New about Whatsapp for News? A Mixed-Method Study on News Outlets' Strategies for Using WhatsApp“.
In: *Digital Journalism* 8.1 (2. Jan. 2020), Seiten 126–144.
DOI: 10.1080/21670811.2019.1692685.

Brants, Sabine, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith und Hans Uszkoreit.
„TIGER: Linguistic Interpretation of a German Corpus“. en.
In: *Research on Language and Computation* 2.4 (Dez. 2004), Seiten 597–620.
DOI: 10.1007/s11168-004-7431-3.

Brosius, Hans-Bernd, Alexander Haas und Friederike Koschel.
Methoden der empirischen Kommunikationsforschung: Eine Einführung. en.
6. Auflage.
Studienbücher zur Kommunikations- und Medienwissenschaft.
VS Verlag für Sozialwissenschaften, 2012.

Chang, Jonathan.
lda: Collapsed Gibbs Sampling Methods for Topic Models.
R package version 1.4.2.
2015.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber und David M. Blei.
„Reading Tea Leaves: How Humans Interpret Topic Models“.
In: *Advances in Neural Information Processing Systems 22*.
Herausgegeben von Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams und A. Culotta.
Curran Associates, Inc., 2009,
Seiten 288–296.

Chinchor, Nancy.
„MUC-4 Evaluation Metrics“.
In: *Proceedings of the 4th Conference on Message Understanding*.
MUC4 '92.
Stroudsburg, PA, USA: Association for Computational Linguistics, 1992,
Seiten 22–29.
DOI: 10.3115/1072064.1072067.

Cohen, Jacob.
„A Coefficient of Agreement for Nominal Scales“.
In: *Educational and Psychological Measurement* 20.1 (1. Apr. 1960), Seiten 37–46.
DOI: 10.1177/001316446002000104.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer und Richard Harshman.
„Indexing by Latent Semantic Analysis“.
In: *Journal of the American Society for Information Science; New York, N.Y.* 41.6 (1. Sep. 1990).

Dumm, Sebastian und Andreas Niekler.
„Methoden, Qualitätssicherung und Forschungsdesign“. de.
In: *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse*.
Herausgegeben von Matthias Lemke und Gregor Wiedemann.
Wiesbaden: Springer Fachmedien Wiesbaden, 2016,
Seiten 89–116.
DOI: 10.1007/978-3-658-07224-7_4.

Elandt-Johnson, Regina C. und Norman L. Johnson.
Survival Models and Data Analysis. en.
John Wiley & Sons, Ltd, 2014.
DOI: 10.1002/9781119011040.

Fei-Fei, Li und Pietro Perona.
„A Bayesian hierarchical model for learning natural scene categories - IEEE Conference Publication“.
In: *Conference on Computer Vision and Pattern Recognation*.
San Diego: IEEE Computer society, Juli 2005.
DOI: 10.1109/CVPR.2005.16.

Feinerer, Ingo, Kurt Hornik und David Meyer.
„Text Mining Infrastructure in R“.
In: *Journal of Statistical Software* 25.5 (März 2008), Seiten 1–54.

Griffiths, Thomas L. und Mark Steyvers.
„Finding scientific topics“.
In: *Proceedings of the National Academy of Sciences* 101 (suppl 1 6. Apr. 2004), Seiten 5228–5235.
DOI: 10.1073/pnas.0307752101.

Grün, Bettina und Kurt Hornik.
„topicmodels: An R Package for Fitting Topic Models“.
In: *Journal of Statistical Software* 40.13 (2011), Seiten 1–30.
DOI: 10.18637/jss.v040.i13.

Hofmann, Thomas.
„Probabilistic Latent Semantic Analysis“.
In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*.
UAI'99.
San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999,
Seiten 289–296.

Hornik, Kurt.
openNLP: Apache OpenNLP Tools Interface.
R package version 0.2-7.
2019.

Kampstra, Peter.
„Beanplot: A Boxplot Alternative for Visual Comparison of Distributions“.
In: *Journal of Statistical Software, Code Snippets* 28.1 (2008), Seiten 1–9.

Kao, Anne und Stephen R. Poteet.
„Overview“. en.
In: *Natural Language Processing and Text Mining*.
Herausgegeben von Anne Kao und Stephen R. Poteet.
London: Springer London, 2007,
Seiten 1–7.
DOI: 10.1007/978-1-84628-754-1_1.

Kendall, M. G. und B. Babington Smith.
„The Problem of m Rankings“.
In: *The Annals of Mathematical Statistics* 10.3 (Sep. 1939), Seiten 275–287.
DOI: 10.1214/aoms/1177732186.

Koppers, Lars, Jonas Rieger, Karin Boczek und Gerret von Nordheim.
tosca: Tools for Statistical Content Analysis.
R package version 0.3-2.
2021.
DOI: 10.5281/zenodo.3591068.

Krippendorff, Klaus.
„Bivariate Agreement Coefficients for Reliability of Data“.
In: *Sociological Methodology* 2 (1970), Seiten 139–150.
DOI: 10.2307/270787.

– „Computing Krippendorff's Alpha-Reliability“.
In: *Departmental Papers (ASC)* (25. Jan. 2011).

Kuhn, H. W. und A. W. Tucker.
„Nonlinear Programming“. EN.
In:
The Regents of the University of California, 1951.

Lemke, Matthias und Gregor Wiedemann.
„Einleitung Text Mining in den Sozialwissenschaften“. de.
In: *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse*.
Herausgegeben von Matthias Lemke und Gregor Wiedemann.
Wiesbaden: Springer Fachmedien Wiesbaden, 2016,
Seiten 1–13.
DOI: 10.1007/978-3-658-07224-7_1.

Maier, Daniel u. a.
„Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology“.
In: *Communication Methods and Measures* 12.2-3 (Apr. 2018), Seiten 93–118.
DOI: 10.1080/19312458.2018.1430754.

Mcauliffe, Jon D. und David M. Blei.
„Supervised Topic Models“.
In: *Advances in Neural Information Processing Systems 20*.
Herausgegeben von J. C. Platt, D. Koller, Y. Singer und S. T. Roweis.
Curran Associates, Inc., 2008,
Seiten 121–128.

Mikolov, Tomas, Kai Chen, Greg Corrado und Jeffrey Dean.
„Efficient Estimation of Word Representations in Vector Space“.
In: *arXiv:1301.3781 [cs]* (Jan. 2013). arXiv: 1301.3781.

Mimno, David, Hanna M. Wallach, Edmund Talley, Miriam Leenders und Andrew McCallum.
„Optimizing Semantic Coherence in Topic Models“.
In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
EMNLP ’11.
event-place: Edinburgh, United Kingdom.
Stroudsburg, PA, USA: Association for Computational Linguistics, 2011,
Seiten 262–272.

Miner, Gary, John Elder IV, Andrew Fast, Thomas Hill, Robert Nisbet und Dursun Delen.
Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications.
Saint Louis, UNITED STATES: Elsevier Science & Technology, 2012.

Nordheim, Gerret von, Karin Boczek und Lars Koppers.
„Sourcing the Sources“.
In: *Digital Journalism* 6.7 (Aug. 2018), Seiten 807–828.
DOI: 10.1080/21670811.2018.1490658.

Nordheim, Gerret von, Karin Boczek, Lars Koppers und Elena Erdmann.
„Digital Traces in Context| Reuniting a Divided Public? Tracing the TTIP Debate on Twitter and in Traditional Media“.
In: *International Journal of Communication* 12.0 (26. Jan. 2018), Seite 22.

Oosterhoff, J. und W. R. van Zwet.
„A Note on Contiguity and Hellinger Distance“.
In: *Selected Works of Willem van Zwet*.
Herausgegeben von Sara van de Geer und Marten Wegkamp.
Selected Works in Probability and Statistics.
New York, NY: Springer, 2012,
Seiten 63–72.
DOI: 10.1007/978-1-4614-1314-1_6.

Porter, M.f.
„An algorithm for suffix stripping“.
In: *Program* 14.3 (März 1980), Seiten 130–137.
DOI: 10.1108/eb046814.

Porter, Martin und Richard Boulton.
Snowball.
URL: http://snowballstem.org/ (besucht am 26. 10. 2018).

Puchinger, Carmen.
„Die Anwendung von Text Mining in den Sozialwissenschaften“. de.
In: *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse*.
Herausgegeben von Matthias Lemke und Gregor Wiedemann.
Wiesbaden: Springer Fachmedien Wiesbaden, 2016,
Seiten 117–136.
DOI: 10.1007/978-3-658-07224-7_5.

R Core Team.
R: A Language and Environment for Statistical Computing.
R Foundation for Statistical Computing. Vienna, Austria, 2023.

Ramage, Daniel, David Hall, Ramesh Nallapati und Christopher D. Manning.
„Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora“.
In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*.
EMNLP ’09.
Stroudsburg, PA, USA: Association for Computational Linguistics, 2009,
Seiten 248–256.

Rieger, Jonas.
„ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations“.
In: *Journal of Open Source Software* 5.51 (2020), Seite 2181.
DOI: 10.21105/joss.02181.

Rieger, Jonas, Carsten Jentsch und Jörg Rahnenführer.
„RollingLDA: An Update Algorithm of Latent Dirichlet Allocation to Construct Consistent Time Series from Textual Data“. en.
In: *Findings of the Association for Computational Linguistics: EMNLP 2021*.
Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021,
Seiten 2337–2347.
DOI: 10.18653/v1/2021.findings-emnlp.201.

Rijsbergen, C. J. Van.
Information Retrieval.
2nd.
Newton, MA, USA: Butterworth-Heinemann, 1979.

Rong, Xin.
„word2vec Parameter Learning Explained“.
In: *arXiv:1411.2738 [cs]* (Nov. 2014). arXiv: 1411.2738.

Rössler, Patrick.
Inhaltsanalyse.
3., völlig überarbeitete Auflage.
UTB ; 2671.
München: UVK Verlagsgesellschaft mbH, 2017.
292 Seiten.

Schmid, Helmut.
„Probabilistic Part-of-Speech Tagging Using Decision Trees“.
In: *International Conference on New Methods in Language Processing*.
Manchester, UK, 1994,
Seiten 44–49.

Sievert, Carson und Kenny Shirley.
LDAvis: Interactive Visualization of Topic Models.
R package version 0.3.2.
2015.

Silge, Julia und David Robinson.
„tidytext: Text Mining and Analysis Using Tidy Data Principles in R“.
In: *JOSS* 1.3 (2016).
DOI: 10.21105/joss.00037.

Stryker, Jo Ellen, Ricardo J. Wray, Robert C. Hornik und Itzik Yanovitzky.
„Validation of Database Search Terms for Content Analysis: The Case of Cancer News Coverage“.
In: *Journalism & Mass Communication Quarterly* 83.2 (1. Juni 2006), Seiten 413–430.
DOI: 10.1177/107769900608300212.

Stulpe, Alexander und Matthias Lemke.
„Blended Reading“. de.
In: *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse*.
Herausgegeben von Matthias Lemke und Gregor Wiedemann.
Wiesbaden: Springer Fachmedien Wiesbaden, 2016,
Seiten 17–61.
DOI: 10.1007/978-3-658-07224-7_2.

Süddeutscher Verlag.
URL: https://www.sueddeutscher-verlag.de/sueddeutsche-zeitung (besucht am 03. 08. 2022).

Thelwall, Mike, Kevan Buckley, Georgios Paltoglou, Di Cai und Arvid Kappas.
„Sentiment strength detection in short informal text“. en.
In: *Journal of the American Society for Information Science and Technology* 61.12 (2010), Seiten 2544–2558.
DOI: 10.1002/asi.21416.

Waldherr, Annie, Lars-Ole Wehden, Daniela Stoltenberg, Peter Miltner, Sophia Ostner und Barbara Pfetsch.
„Inductive Codebook Development for Content Analysis: Combining Automated and Manual Methods“. de.
In: *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 20.1 (Jan. 2019).
DOI: 10.17169/fqs-20.1.3058.

Wan, Li, Leo Zhu und Rob Fergus.
„A Hybrid Neural Network-Latent Topic Model“.
In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*.
Herausgegeben von Neil D. Lawrence und Mark Girolami.
Band 22.
Proceedings of Machine Learning Research.
La Palma, Canary Islands: PMLR, 21. Apr. 2012,
Seiten 1287–1294.

Weber, M.
Max Webers Vollstandige Schriften Zu Wissenschaftlichen Und Politischen Berufen.
Classics series.
Algora Publishing, 2017.

Wickham, Hadley.
ggplot2: Elegant Graphics for Data Analysis.
Springer-Verlag New York, 2016.

Wiedemann, Gregor und Matthias Lemke.
„Text Mining für die Analyse qualitativer Daten“. de.
In: *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse.*
Herausgegeben von Matthias Lemke und Gregor Wiedemann.
Wiesbaden: Springer Fachmedien Wiesbaden, 2016,
Seiten 397–419.
DOI: 10.1007/978-3-658-07224-7_15.

Wiedemann, Gregor und Andreas Niekler.
„Analyse qualitativer Daten mit dem „Leipzig Corpus Miner““. de.
In: *Text Mining in den Sozialwissenschaften: Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse.*
Herausgegeben von Matthias Lemke und Gregor Wiedemann.
Wiesbaden: Springer Fachmedien Wiesbaden, 2016,
Seiten 63–88.
DOI: 10.1007/978-3-658-07224-7_3.

„Zum Geleit“.
In: *Süddeutsche Zeitung* 1.1 (1945).

A Anhang zu den Sampling-Prozeduren

A.1 Varianzschätzer für den Recall-Schätzer

Der Recall-Schätzer kann als bivariate Funktion von X und Y angesehen werden, dabei beschreibt X die Verteilung der relevanten Texte in der interessierenden Menge und Y die Verteilung der relevanten Texte außerhalb der interessierenden Menge. Beide Variablen sind binomialverteilt. Durch das Umformen des Recall-Schätzers werden die beiden Summen so umgeschrieben, dass die genutzten Mengen disjunkt sind.

$$\begin{aligned}
\widehat{\text{Recall}} &= \frac{\sum_{i \in X} w_i \hat{\pi}_i}{\sum_{i \in X \cup Y} w_i \hat{\pi}_i} \\
&= \frac{\sum_{i \in X} w_i \hat{\pi}_i}{\sum_{i \in X} w_i \hat{\pi}_i + \sum_{i \in Y} w_i \hat{\pi}_i} \\
&= \frac{1}{1 + \frac{\sum_{i \in Y} w_i \hat{\pi}_i}{\sum_{i \in X} w_i \hat{\pi}_i}}
\end{aligned}$$

Die beiden Summen entsprechen nun den Erwartungswertschätzern $\bar{X} = \sum_{i \in X} w_i \hat{\pi}_i$ und $\bar{Y} = \sum_{i \in Y} w_i \hat{\pi}_i$. Der Schätzer kann als Funktion von $\tilde{X}$ und $\tilde{Y}$ dargestellt werden. Für eine Funktion dieser Form können allgemein die partiellen Ableitungen berechnet werden:

$$
\begin{aligned}
f\left(\tilde{X},\tilde{Y}\right) &= \frac{1}{1+\frac{\tilde{Y}}{\tilde{X}}} = \frac{1}{1+\frac{\sum_{i\in Y} w_i\hat{\pi}_i}{\sum_{i\in X} w_i\hat{\pi}_i}} \\
f'_{\tilde{X}}\left(\tilde{X},\tilde{Y}\right) &= \left(-\frac{\tilde{Y}}{\tilde{X}^2}\right)\cdot\left(-\frac{1}{\left(1+\frac{\tilde{Y}}{\tilde{X}}\right)^2}\right) = \frac{\tilde{Y}}{\left(\tilde{Y}+\tilde{X}\right)^2} \\
f'_{\tilde{Y}}\left(\tilde{X},\tilde{Y}\right) &= \frac{1}{\tilde{X}}\cdot\left(-\frac{1}{\left(1+\frac{\tilde{Y}}{\tilde{X}}\right)^2}\right) = -\frac{\tilde{X}}{\left(\tilde{Y}+\tilde{X}\right)^2}
\end{aligned}
$$

Da die einzelnen Schnittmengen unabhängig voneinander sind, sind $\tilde{X}$ und $\tilde{Y}$ unabhängig verteilt. Mit Hilfe einer Taylor-Entwicklung erster Ordnung von $f(X,Y)$ um $\theta = (\mathrm{E}(X), \mathrm{E}(Y))$ kann die Varianz abgeschätzt werden (Elandt-Johnson und Johnson, 2014, Seite 69ff). Es wird $\mathrm{E}\left(f\left(X,Y\right)\right) \approx f\left(\theta\right)$ genutzt.

$$
\begin{aligned}
& \mathrm{Var}\left(f\left(X,Y\right)\right) \\
= \quad & \mathrm{E}\left(\left(f\left(X,Y\right) - \mathrm{E}\left(f\left(X,Y\right)\right)\right)^2\right) \\
\approx \quad & \mathrm{E}\left(\left(f\left(\theta\right) + f'_X\left(\theta\right)\left(X-\theta_X\right) + f'_Y\left(\theta\right)\left(Y-\theta_Y\right) - f\left(\theta\right)\right)^2\right) \\
= \quad & \mathrm{E}\left(f'^2_X\left(\theta\right)\left(X-\theta_X\right)^2 + 2f'_X\left(\theta\right)\left(X-\theta_X\right)f'_Y\left(\theta\right)\left(Y-\theta_Y\right) + f'^2_Y\left(\theta\right)\left(Y-\theta_Y\right)^2\right) \\
= \quad & f'^2_X\left(\theta\right)\mathrm{Var}\left(X\right) + 2f'_X\left(\theta\right)f'_Y\left(\theta\right)\mathrm{Cov}\left(X,Y\right) + f'^2_Y\left(\theta\right)\mathrm{Var}\left(Y\right) \\
\overset{X,Y\text{unabh.}}{=} \quad & f'^2_X\left(\theta\right)\mathrm{Var}\left(X\right) + f'^2_Y\left(\theta\right)\mathrm{Var}\left(Y\right)
\end{aligned}
$$

Da X und Y unabhängig voneinander sind, entfällt der Kovarianz-Term. Der Varianzschätzer kann nun mit dieser Vorarbeit gebildet werden.

$$\begin{aligned}
& \operatorname{Var}\left(\widehat{Recall}\right) && \text{(A.1)}\\
= \; & \operatorname{Var}\left(\frac{1}{1+\frac{\sum_{i\in Y} w_i\hat{\pi}_i}{\sum_{i\in X} w_i\hat{\pi}_i}}\right) && \text{(A.2)}\\
\approx \; & \left(\frac{\bar{Y}}{\left(\bar{Y}+\bar{X}\right)^2}\right)^2 \operatorname{Var}(X) + \left(-\frac{\bar{X}}{\left(\bar{Y}+\bar{X}\right)^2}\right)^2 \operatorname{Var}(Y) && \text{(A.3)}\\
= \; & \left(\frac{\sum_{i\in Y} w_i\hat{\pi}_i}{\left(\sum_{i\in X\cup Y} w_i\hat{\pi}_i\right)^2}\right)^2 \sum_{i\in X} w_i^2 \frac{1}{n_i}\hat{\pi}_i\left(1-\hat{\pi}_i\right) && \text{(A.4)}\\
+ \; & \left(\frac{\sum_{i\in X} w_i\hat{\pi}_i}{\left(\sum_{i\in X\cup Y} w_i\hat{\pi}_i\right)^2}\right)^2 \sum_{i\in Y} w_i^2 \frac{1}{n_i}\hat{\pi}_i\left(1-\hat{\pi}_i\right) && \text{(A.5)}
\end{aligned}$$

A.2 Ziehen von neuen Stichproben

Das Ziel ist es m neu zu ziehende Texte so auf die einzelnen Schnittmengen aufzuteilen, dass die Varianz des Recall-Schätzers minimiert wird. Für die Schnittmenge i sollen m_i neue Texte gezogen und gelabelt werden. Dabei sollen alle m_i Null oder positiv sein und sich zu m addieren. Zur Berechnung der optimalen Aufteilung wird die Lagrange-Optimierung genutzt. Da die erste Bedingung eine Ungleichung ist, werden die Karush-Kuhn-Tucker-Bedingungen (Kuhn und Tucker, 1951) verwendet. Im ersten Schritt wird dazu die Lagrange-Funktion verwendet. Die Nebenbedingungen werden umgeschrieben zu $-m_i \leq 0$ (nicht-Negativität) und $\sum_i m_i - m = 0$ (Summe gleich m). Die Lagrange-Funktion lautet in diesem Fall:

$$\Lambda(m_i, \lambda, \mu) = \sum_i \frac{c_i}{n_i + m_i} + \lambda\left(\sum_i m_i - m\right) + \sum_i \mu_i \cdot (-m_i)\,.$$

Dabei sind n_i die bereits gezogenen und bewerteten Texte aus der Schnittmenge i, $c_i = b_i w_i^2 \pi_i(1-\pi_i)$ die gewichtete Varianzkomponente aus der Varianz für den Recall

und $b_i = \left(\frac{\sum_{i \in Y} w_i \hat{\pi}_i}{\left(\sum_{i \in X \cup Y} w_i \hat{\pi}_i \right)^2} \right)^2$ wenn die Schnittmenge i im interessierenden Subkorpus liegt und $b_i = \left(\frac{\sum_{i \in X} w_i \hat{\pi}_i}{\left(\sum_{i \in X \cup Y} w_i \hat{\pi}_i \right)^2} \right)^2$ sonst. Im nächsten Schritt wird die Ableitung der Lagrange-Funktion nach m_j Null gesetzt.

$$
\begin{aligned}
0 &\stackrel{!}{=} \frac{\partial}{\partial m_j} \sum_i \frac{c_i}{n_i + m_i} + \lambda \frac{\partial}{\partial m_j} \left(\sum_i m_i - m \right) + \sum_i \mu_i \frac{\partial}{\partial m_j} \cdot (-m_i) && \text{(A.6)} \\
&= \frac{\partial}{\partial m_j} \frac{c_j}{n_j + m_j} + \lambda - \mu_j && \text{(A.7)} \\
&= -\frac{c_j}{(n_j + m_j)^2} + \lambda - \mu_j && \text{(A.8)}
\end{aligned}
$$

Die Komplementaritätsbedingung für die Ungleichung $-m_i \leq 0$ fordert nun, dass für mögliche Minima unserer Zielfunktion jeweils μ_i oder m_i Null ist $(\forall i)$. Für den Fall $m_i = 0$ müssen keine weiteren Berechnungen angestellt werden. Für $\mu_i = 0$ kann nun die Gleichung nach m_j aufgelöst werden.

$$
\begin{aligned}
0 &\stackrel{!}{=} -\frac{c_j}{(n_j + m_j)^2} + \lambda && \text{(A.9)} \\
\Leftrightarrow \frac{\lambda}{c_j} &\stackrel{!}{=} \frac{1}{(n_j + m_j)^2} && \text{(A.10)} \\
\Leftrightarrow n_j + m_j &\stackrel{!}{=} \sqrt{\frac{c_j}{\lambda}} && \text{(A.11)} \\
\Leftrightarrow m_j &\stackrel{!}{=} \sqrt{\frac{c_j}{\lambda}} - n_j && \text{(A.12)}
\end{aligned}
$$

Durch Einsetzen in die Nebenbedingung $\sum_i m_i - m = 0$ ergibt sich ein Wert für $1/\sqrt{\lambda}$:

$$
\begin{aligned}
0 &\stackrel{!}{=} \sum_i \left(\frac{\sqrt{c_i}}{\sqrt{\lambda}} - n_i \right) - m && \text{(A.13)} \\
\Leftrightarrow 0 &\stackrel{!}{=} \frac{1}{\sqrt{\lambda}} \sum_i \sqrt{c_i} - \sum_i n_i - m && \text{(A.14)} \\
\Leftrightarrow \frac{\sum_i n_i + m}{\sum_i \sqrt{c_i}} &\stackrel{!}{=} \frac{1}{\sqrt{\lambda}} && \text{(A.15)} \\
& && \text{(A.16)}
\end{aligned}
$$

Das Einsetzen in die Gleichung für m_j ergibt dann die Berechnungsformel für den Fall, dass $m_j \neq 0$ ist.

$$m_j \stackrel{!}{=} \sqrt{\frac{c_j}{\lambda}} - n_j \tag{A.17}$$

$$= \frac{\sqrt{c_j}\,(\sum_i n_i + m)}{\sum_i \sqrt{c_i}} - n_j \tag{A.18}$$

Die möglichen Optima ergeben sich dementsprechend aus allen Kombinationen, in denen für die einzelnen Schnittmengen entweder die zu ziehenden Texte nach obiger Formel berechnet wurden, oder aus der Schnittmenge keine Texte gezogen werden ($m_i = 0$). Im Fall mit k Subkorpora ergibt dies bei 2^k Schnittmengen $2^{2^k} - 1$ mögliche Kombinationen, die bezüglich ihrer erwartbaren Varianz des Recall-Schätzers beurteilt werden müssen. Für drei Subkorpora sind dies schon 255 Kombinationen.

B Abbildungen

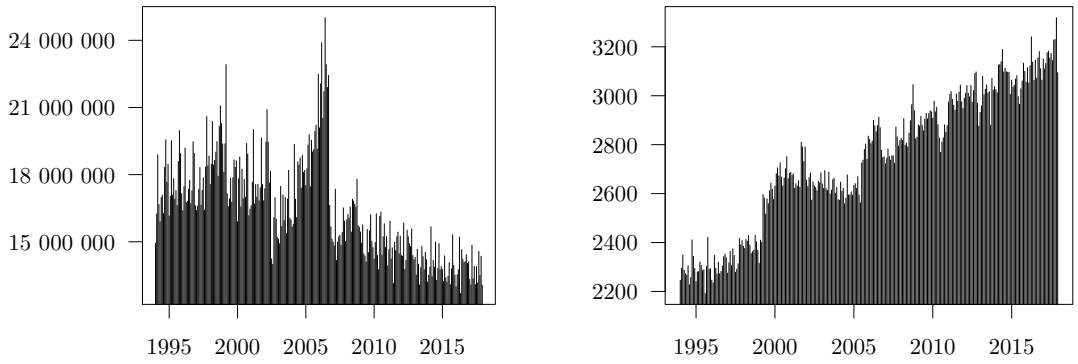

Abbildung B.1: Zeitliche Verteilung der Anzahl der monatlich publizierten Zeichen (links) und der durchschnittlichen Zeichen pro Artikel und Monat (rechts).

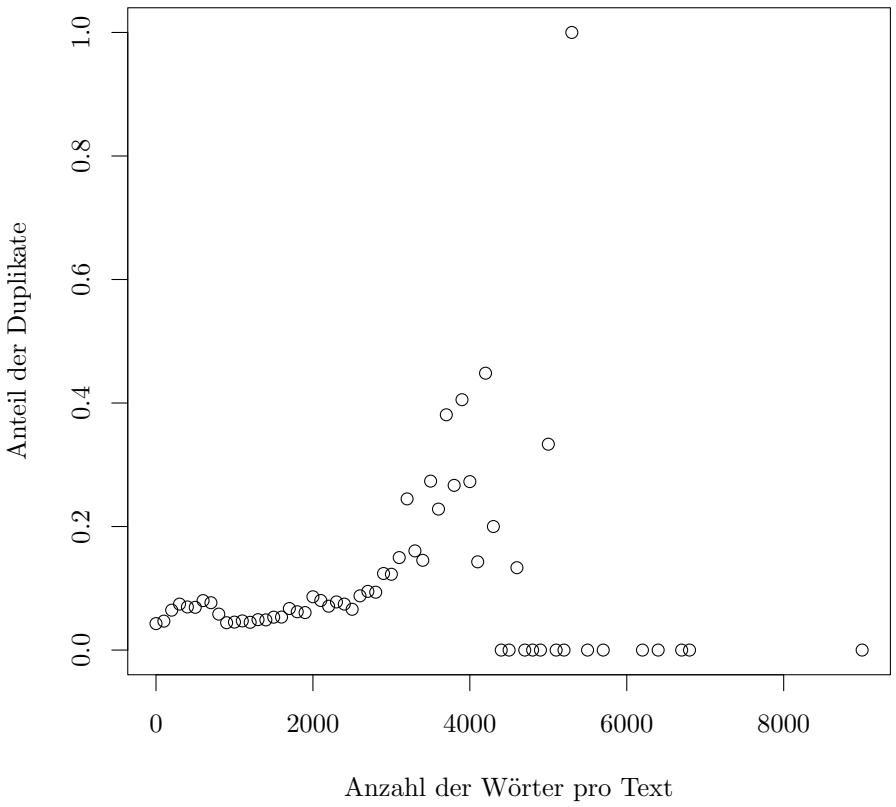

Abbildung B.2: Anteil der Duplikate an den Artikeln mit gleicher Wortanzahl. Die Anzahl der Wörter wird dabei auf ganze 100 abgerundet. Die Schwankungen in den hohen Kategorien können auf schwach besetzte Klassen zurückgeführt werden.

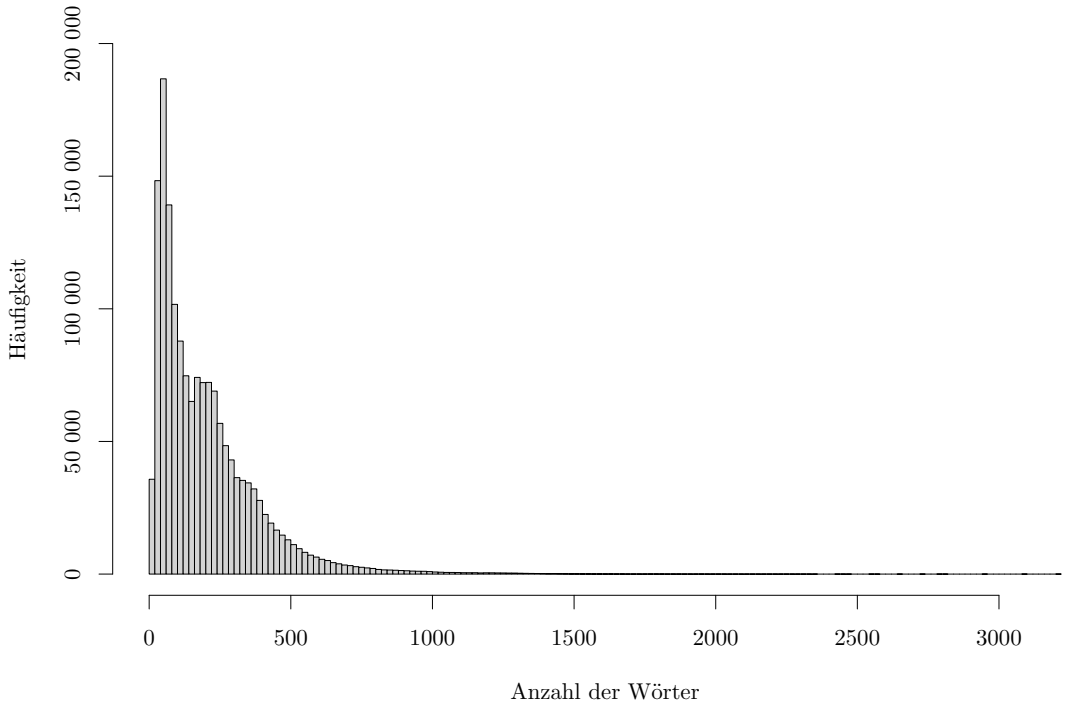

Abbildung B.3: Histogramm über die Textlängen nach der Vorverarbeitung.

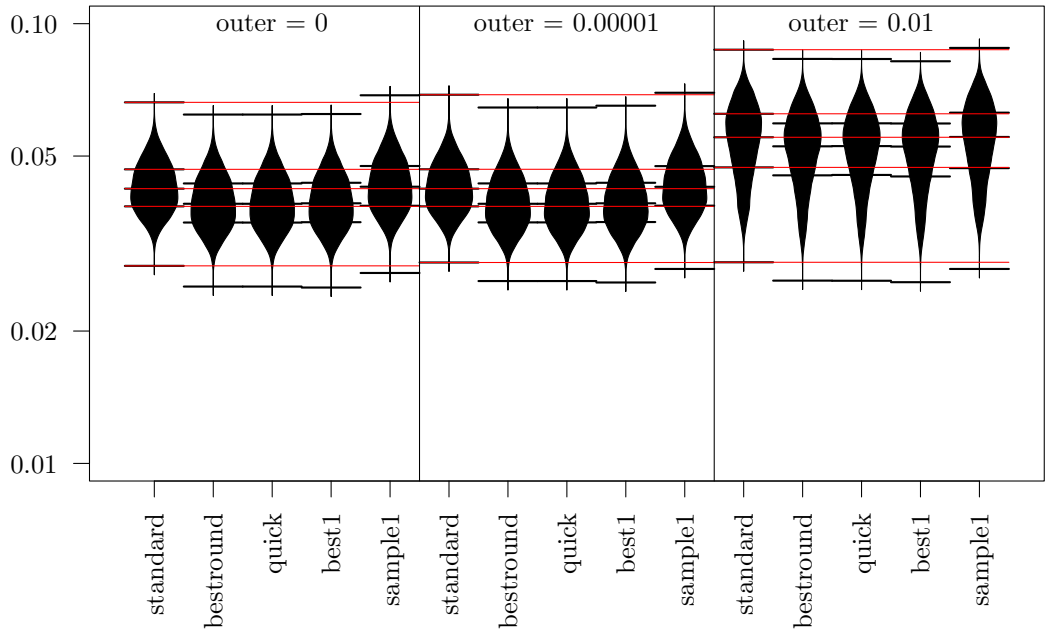

Abbildung B.4: Trunkierte Beanplots der geschätzten Standardabweichungen der Recall-Schätzer für das anwendungsnahe Szenario auf einer log-Skala. Pro Subkorpus wurden 100 Texte im Vorfeld gelabelt, in der äußeren Menge 100. Weitere 200 Texte wurden nach der jeweiligen Strategie gezogen. Die roten Linien markieren die Quartile der `standard`-Methode.

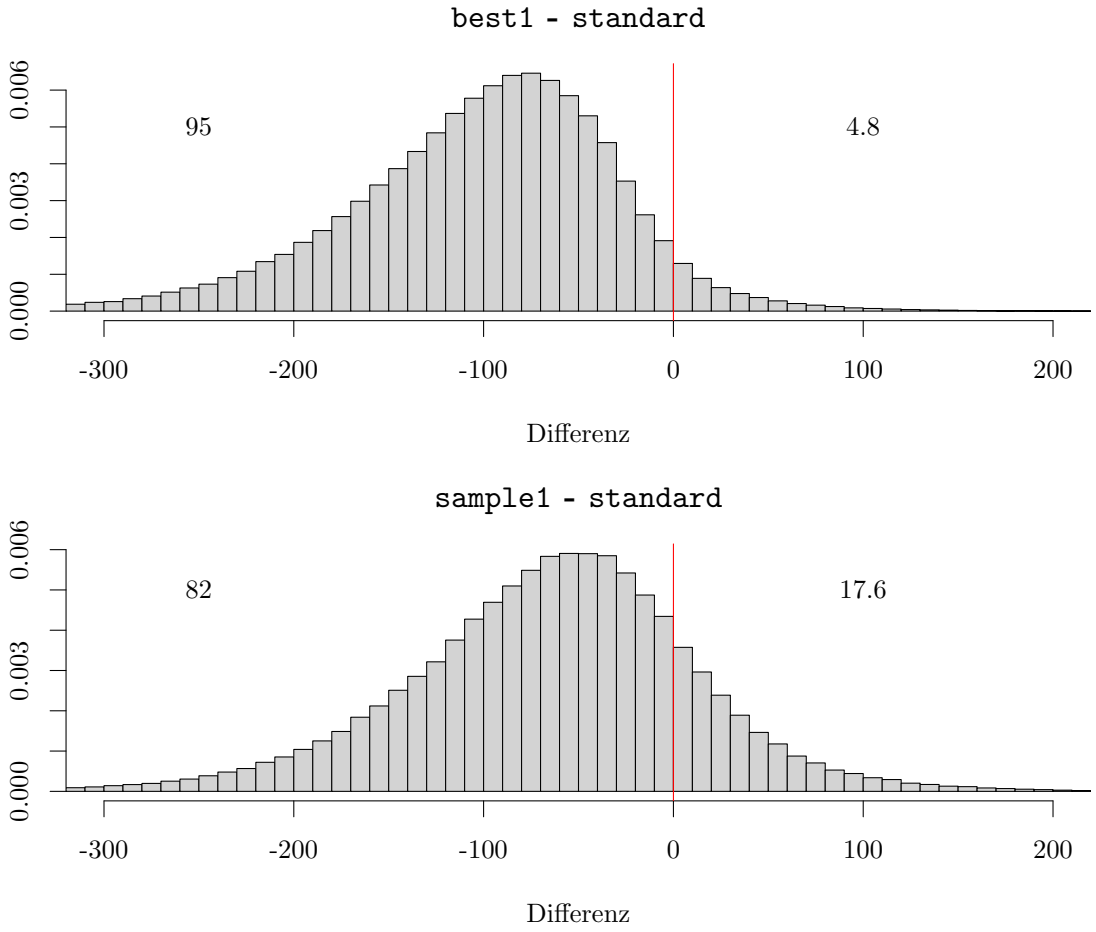

Abbildung B.5: Trunkierte Histogramme der Differenzen zur `standard`-Methode in der Anzahl der benötigten Texte im Szenario mit 100 in der äußeren Schnittmenge gelabelten Texten. Bei positiven Werten benötigt die `standard`-Methode weniger Texte. Die Prozentangaben addieren sich nicht zu 100, weil die Fälle in denen beide Methoden gleich viele Texte benötigen dazu kommen.

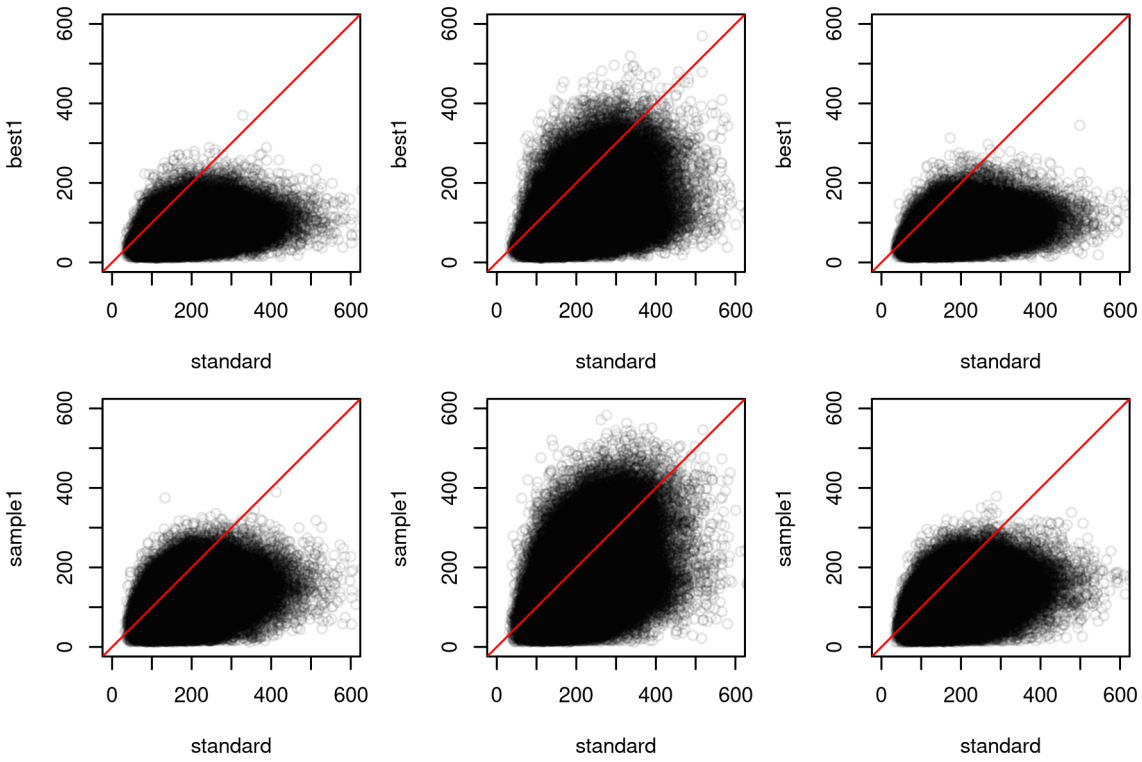

Abbildung B.6: Scatterplots der benötigten zusätzlichen Texte der drei Methoden für ein Konfidenzintervall von 2 Prozentpunkten beim Recall-Schätzer im Szenario mit 100 Texten in der äußeren Schnittmenge gelabelten Texten. In den drei Spalten sind jeweils die Simulationen mit 0 %, 0.001 % und 1 % relevante Texte in der äußeren Schnittmenge abgebildet.

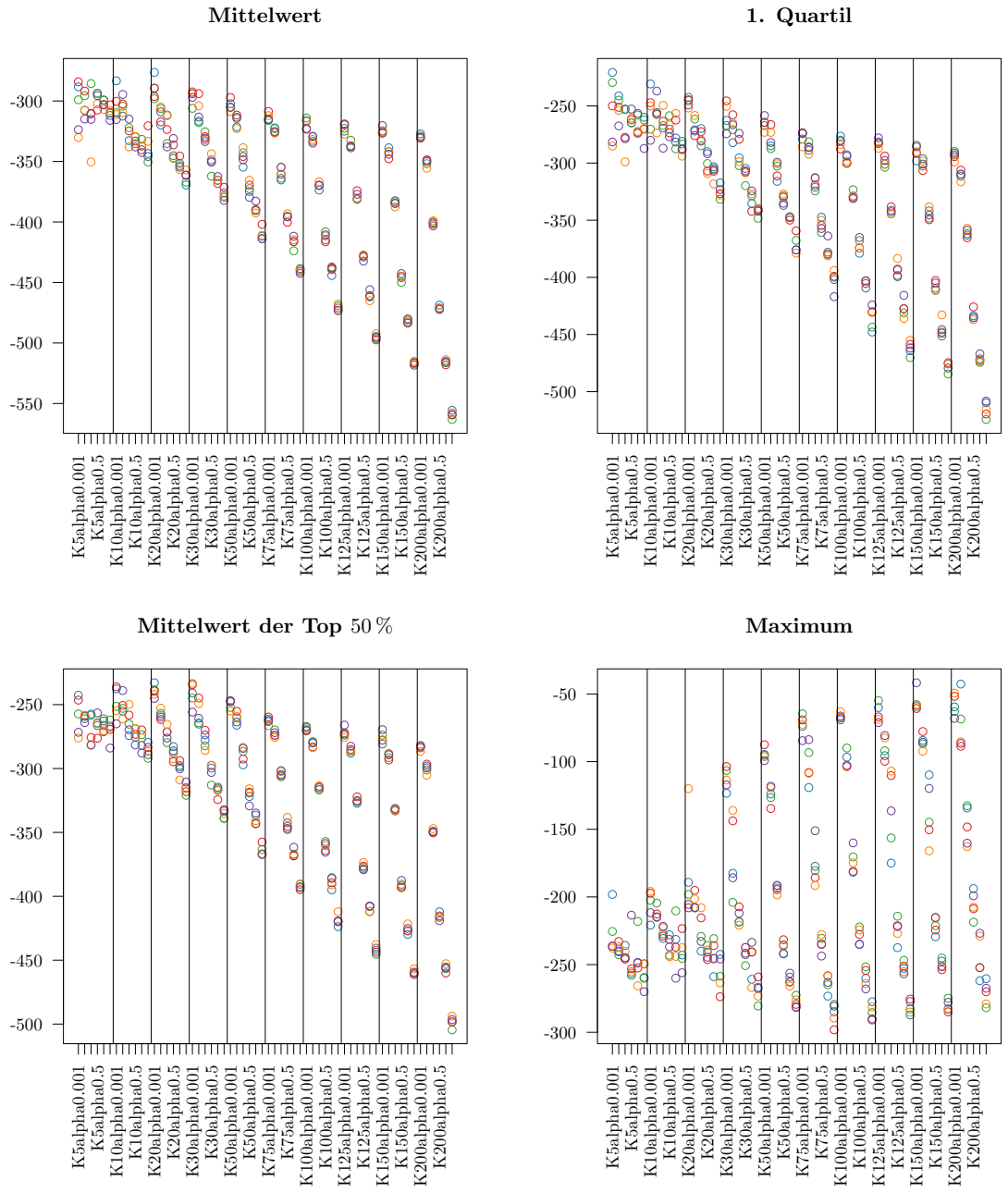

Abbildung B.7: Verschiedene Maßzahlen für die Topic Coherence unter Verwendung der Topwörter. Die jeweilige Maßzahl wurde über die Topic Coherencen der einzelnen Themen im Modell berechnet.

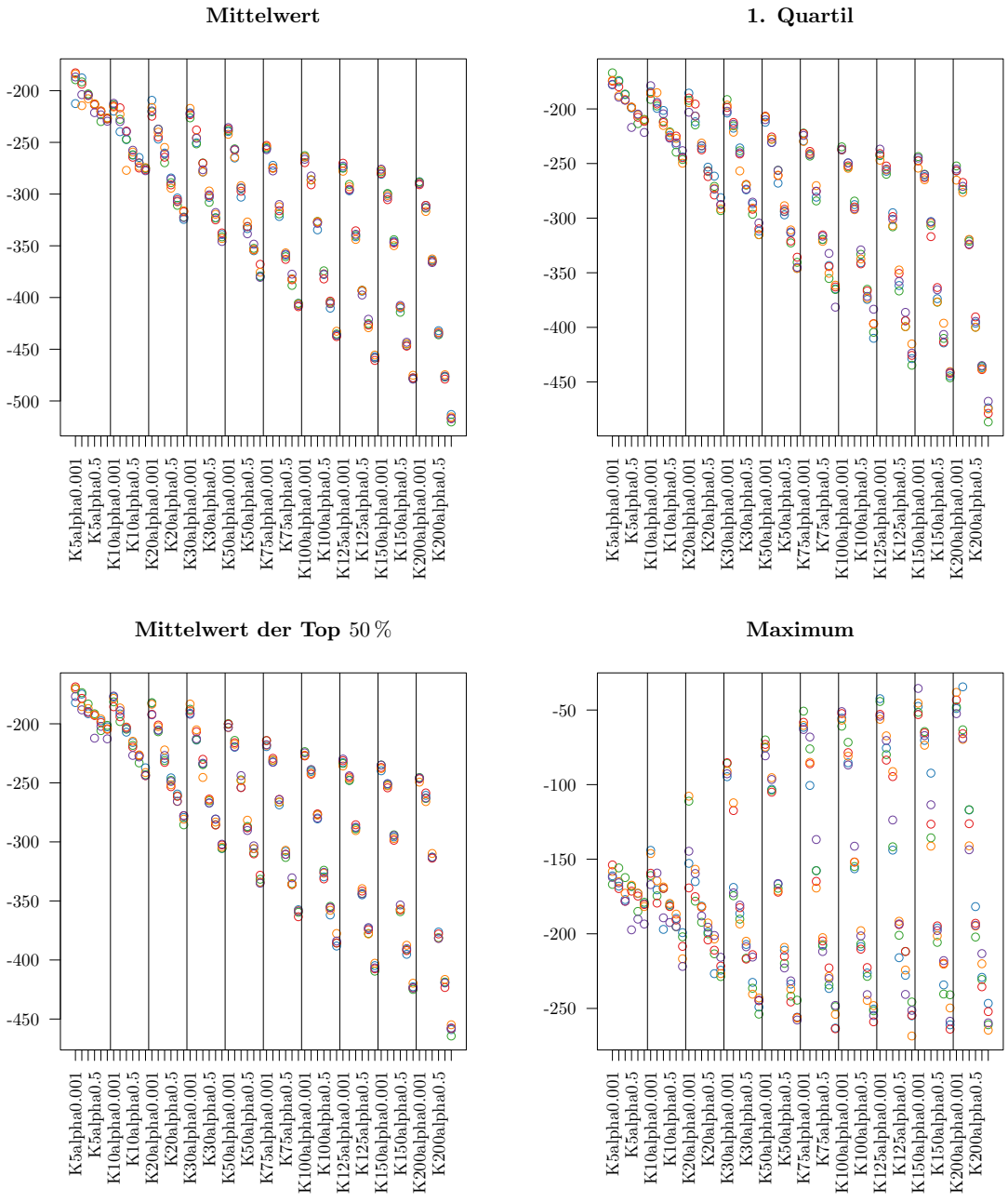

Abbildung B.8: Verschiedene Maßzahlen für die symmetrische Topic Coherence. Die jeweilige Maßzahl wurde über die Topic Coherencen der einzelnen Themen im Modell berechnet.

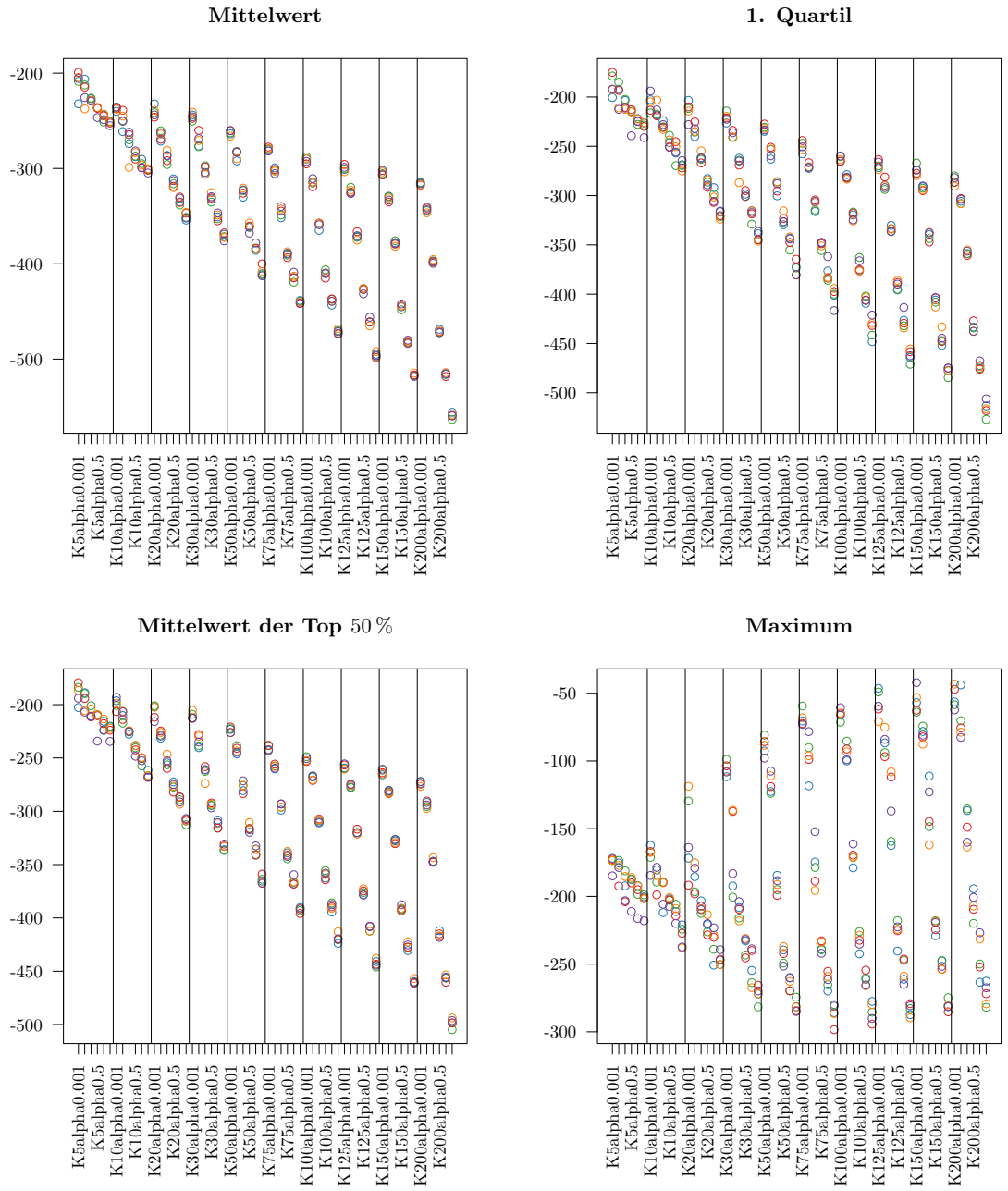

Abbildung B.9: Verschiedene Maßzahlen für die Topic Coherence. Die jeweilige Maßzahl wurde über die Topic Coherencen der einzelnen Themen im Modell berechnet.

C Tabellen

Rubrik	Anzahl	Rubrik	Anzahl
Wirtschaft	297639	NRW-Nordrhein-Westfalen	1462
Politik	181629	Hobby	1461
Sport	171499	Themen aus dem Ausland	1396
Nachrichten	159682	Computerseite	1369
Feuilleton	99573	NRW-Sport	1365
Muenchen	96654	Sachbuch	1357
Panorama	85247	NRW-Themen	1189
Meinungsseite	75990	NRW-Panorama	1118
Medien	66675	Roman	1068
Bayern	48078	Wirtschaft /Geld	868
Geld	43975	NRW-Wirtschaft	842
Beilage	33150	NRW-Feuilleton	828
Leserbriefe	32341	Buch Zwei	664
Themen des Tages	31382	Neue Technik	594
Muenchen/Bayern	30304	Anzeige	518
Literatur	27895	NRW-Kalender	470
Muenchner Kultur	20753	Mode	448
Wissen	18499	Geld Technik	385
Reise	17843	Serie	355
Mobiles Leben	15686	NRW-Report	297
Immobilien	14789	Boerse und Finanzen	270
Die Seite Drei	14669	Rock Jazz	250
SZ Wochenende	13937	Kunst Preise	209
Wissenschaft	12729	Kinder- und Jugendmedien	207

Sportbeilage	10010	Berlin-Service	182
Muenchner Sport	7747	Sonstiges	173
Muenchner Wirtschaft	7646	Zeitung in der Schule Muenchen	159
Forum Leserbriefe	7441	NRW-Bayern	141
Berlin-Seite	7048	Muenchen Nord	140
Filmseite	6607	Wirtschaftsreport	138
Schule und Hochschule	5675	NRW-Muenchen	131
Beruf und Karriere	5398	Jugend, Schule, Wirtschaft	113
Bildung und Beruf	5226	Sport in Bayern	96
SZ am Wochenende	3993	Wetter	83
Mietmarkt	3763	Gesundheit	77
Stil	3718	NRW-Bildung und Beruf	77
Themen aus Deutschland	3668	Sonderbeilage	70
Forum	3477	Lebenserfahrung	69
Letzte Seite	3406	Das politische Kinderbuch	65
Wirtschaft Beilage	3362	Video-Seite	63
Politisches Buch	3343	Buchjournal	32
Gesellschaft	3303	Berlin	29
Kinder- und Jugendliteratur	3273	Muenchner Stadtanzeiger	28
JETZT.DE	3230	Vermischtes	15
Kunstmarkt	3211	Muenchen Sued	14
Themen	3117	Bayern Kultur	11
Zeitung in der Schule	3001	unbekannt	10
Dokumentation	2776	Wirtschaft /Boerse und Finanzen	10
Berg- und Ski-Journal	2717	Hochschule	3
Literaturbeilage	2397	Alle Spiele, alle Tore	2
Region Muenchen	2263	Umwelt, Wissenschaft, Technik	2
Report	2238	jetzt.muenchen	1
Kinderseiten	2196	Seite Drei	1
Freizeit	2166	Wochenend und Freizeit	1
Schallplatte	1961		

Tabelle C.1: Häufigkeit der Rubriken in der Süddeutschen Zeitung

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
prozent	sagt	merkel	hsh	menschen
dax	mal	sagte	hamburg	land
aktien	ja	spd	aufsichtsrat	millionen
anleger	frau	steinbrueck	nonnenmacher	strom
punkte	menschen	berlin	nordbank	afrika
dollar	mann	cdu	wulff	wasser
haendler	leute	bundesregierung	vorstand	welt
euro	leben	fdp	siemens	dollar
minus	stadt	schaeuble	hamburger	energie
dow	immer	kanzlerin	millionen	oel
Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
euro	staatsanwaltschaft	dollar	prozent	bayernlb
bank	gericht	new	wirtschaft	landesbank
milliarden	bank	milliarden	jahr	sparkassen
hre	millionen	york	wachstum	hypo
kfw	worden	bank	rezession	adria
millionen	sei	goldman	krise	alpe
estate	wegen	morgan	konjunktur	westlb
ikb	richter	street	preise	bayern
hypo	fall	wall	inflation	seehofer
real	ermittlungen	sachs	unternehmen	csu
Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
ezb	tel	daten	euro	griechenland
griechenland	schaeffler	internet	prozent	regierung
euro	dr	computer	sagt	euro
zentralbank	gmbh	sagt	wer	bruessel
europaeischen	muenchen	facebook	immobilie	athen

europa	zeitung	menschen	zinsen	land
eurozone	conti	netz	immobilien	milliarden
europaeische	sueddeutsche	etwa	millionen	eu
staaten	h	wer	darlehen	iwf
staatsanleihen	a	informationen	wohnungen	europaeischen
Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
sz	kunden	fc	banken	obama
ja	banken	spieler	milliarden	usa
muessen	bank	hoeness	kredite	washington
banken	schweizer	trainer	geld	dollar
krise	schweiz	bayern	anleihen	fed
gibt	kunde	fussball	euro	praesident
geld	liechtenstein	millionen	institute	barack
warum	franken	spiel	bank	new
unsere	geld	klub	prozent	bernanke
waere	ubs	beim	kapital	iwf
Topic 21	Topic 22	Topic 23	Topic 24	Topic 25
schon	unternehmen	euro	london	bank
immer	sagt	bank	banken	ackermann
wer	firmen	milliarden	pfund	deutschen
ja	porsche	millionen	britischen	chef
mehr	mitarbeiter	prozent	britische	weber
waere	firma	commerzbank	londoner	jain
krise	frauen	dresdner	grossbritannien	deutsche
geld	deutschland	unternehmen	briten	josef
viele	vw	konzern	aufsicht	fitschen
vielleicht	viele	uebernahme	of	nachfolger
Topic 26	Topic 27	Topic 28	Topic 29	Topic 30
anleger	wurde	menschen	china	museum
sagt	heute	politik	russland	kunst
fonds	jahre	buch	dollar	sammlung
aktien	damals	gesellschaft	regierung	kunsthalle

prozent	mann	kapitalismus	peking	kuenstler
gold	geschichte	politischen	chinas	the
geld	new	demokratie	chinesischen	galerie
rendite	familie	politische	japan	kunstverein
investoren	spaeter	staat	land	kunstmuseum
zertifikate	film	buerger	chinesische	art

Tabelle C.4: Liste der Topwörter der Beispiel-LDA aus Kapitel 4

aber	dein	einem	ihm	könnte	selbst	weg
alle	deine	einen	ihn	machen	sich	weil
allem	deinem	einer	ihnen	man	sie	weiter
allen	deinen	eines	ihr	manche	sind	welche
aller	deiner	einig	ihre	manchem	so	welchem
alles	deines	einige	ihrem	manchen	solche	welchen
als	dem	einigem	ihren	mancher	solchem	welcher
also	demselben	einigen	ihrer	manches	solchen	welches
am	den	einiger	ihres	mein	solcher	wenn
an	denn	einiges	im	meine	solches	werde
ander	denselben	einmal	in	meinem	soll	werden
andere	der	er	indem	meinen	sollte	wie
anderem	derer	es	ins	meiner	sondern	wieder
anderen	derselbe	etwas	ist	meines	sonst	will
anderer	derselben	euch	jede	mich	über	wir
anderes	des	euer	jedem	mir	ueber	wird
anderm	desselben	eure	jeden	mit	um	wirst
andern	dessen	eurem	jeder	muss	und	wo
anderr	dich	euren	jedes	musste	uns	wollen
anders	die	eurer	jene	nach	unse	wollte
auch	dies	eures	jenem	nicht	unsem	wuerde
auf	diese	fuer	jenen	nichts	unsen	wuerden
aus	dieselbe	für	jener	noch	unser	würde
bei	dieselben	gegen	jenes	nun	unses	würden
bin	diesem	gewesen	jetzt	nur	unter	zu
bis	diesen	hab	kann	ob	viel	zum
bist	dieser	habe	kein	oder	vom	zur
da	dieses	haben	keine	ohne	von	zwar
damit	dir	hat	keinem	sehr	vor	zwischen
dann	doch	hatte	keinen	sein	waehrend	
das	dort	hatten	keiner	seine	während	
dass	du	hier	keines	seinem	war	
daß	durch	hin	koennen	seinen	waren	
dasselbe	ein	hinter	koennte	seiner	warst	
dazu	eine	ich	können	seines	was	

Tabelle C.2: Liste der verwendeten Stopwörter

Funktionsname	Kategorie	-	Funktionsname (Forts.)	Kategorie (Forts.)
readTextmeta	Daten einlesen		clusterTopics	Themenanalyse
readWhatsApp	Daten einlesen		LDAgen	Themenanalyse
readWiki	Daten einlesen		LDAprep	Themenanalyse
readWikinews	Daten einlesen		mergeLDA	Themenanalyse
as.corpus.textmeta	Textmeta-Objekte		removeXML	Themenanalyse
as.meta	Textmeta-Objekte		topicsInText	Themenanalyse
as.textmeta.corpus	Textmeta-Objekte		topTexts	Themenanalyse
mergeTextmeta	Textmeta-Objekte		topWords	Themenanalyse
showMeta	Textmeta-Objekte		plotArea	Visualisierung
showTexts	Textmeta-Objekte		plotFreq	Visualisierung
textmeta	Textmeta-Objekte		plotHeat	Visualisierung
tidy.textmeta	Textmeta-Objekte		plotScot	Visualisierung
cleanTexts	Vorverarbeitung		plotTopic	Visualisierung
deleteAndRenameDuplicates	Vorverarbeitung		plotTopicWord	Visualisierung
duplist	Vorverarbeitung		plotWordpt	Visualisierung
filterCount	Vorverarbeitung		plotWordSub	Visualisierung
filterDate	Vorverarbeitung		intruderTopics	Validierung
filterID	Vorverarbeitung		intruderWords	Validierung
filterWord	Vorverarbeitung		precision	Validierung
makeWordlist	Vorverarbeitung		sampling	Validierung
			topicCoherence	Validierung

Tabelle C.3: Liste der Funktionen im Paket tosca

US Banken	Krise (US)
dollar	krise
new	banken
york	geld
milliarden	finanzkrise
bank	usa
goldman	kredite
morgan	dollar
street	lehman
wall	rezession
sachs	regierung
jp	wirtschaft
lehman	amerikanischen
investmentbank	amerikanische
of	fed
citigroup	welt
sec	brothers
stanley	viele
yorker	pleite
america	heute
aig	mehr
usa	bernanke
amerikanischen	schulden
finanzkrise	staaten
brothers	notenbank
merrill	folgen
millionen	amerika
hedgefonds	new
lynch	immer
investoren	staat
buffett	jahre

Tabelle C.5: Liste der Topwörter der beiden verwendeten Themen der LDA aus Kapitel 5

Tabelle C.6: Wertebereiche für die Anteile w_i der Schnittmengen am Gesamtkorpus. Die ID für die Schnittmenge beschreibt in welchem Subkorpus sie enthalten ist.

Schnittmenge	Wertebereich für w_i
000	$1 - \sum w_i \quad i \neq 000$
001	zwischen 0.025 und 0.045
010	zwischen 0.001 und 0.002
011	zwischen 0.0003 und 0.0005
100	zwischen 0.04 und 0.055
101 und 110	zwischen 0.01 und 0.03
111	zwischen 0.04 und 0.08

Tabelle C.7: Werte für den Anteil relevanter Texte π_i in den verschiedenen Schnittmengen. Die ID für die Schnittmenge beschreibt in welchem Subkorpus sie enthalten ist.

Schnittmenge	Wertebereich für π
000	0, 0.01 oder 0.00001
001 und 010	zwischen 0.01 und 0.1
011 und 100	zwischen 0.08 und 0.15
101	zwischen 0.2 und 0.35
110	zwischen 0.35 und 0.55
111	zwischen 0.8 und 0.98

Rang	standard	bestround	quick	best1	sample1
1	17578	43060	43001	98381	123621
2	31145	88689	88365	98391	37833
3	44168	104919	105108	31213	30185
4	70598	37428	37460	35928	48022
5	136511	25904	26066	36087	60339

Tabelle C.8: Ränge der absoluten Abstände zum wahren Wert für das anwendungsnahe Szenario bei drei Einstellungen mit jeweils 100 000 Wiederholungen. Pro Subkorpus wurden 100 Texte im Vorfeld gelabelt, in der äußeren Menge 100. Weitere 200 Texte wurden nach der jeweiligen Strategie gezogen. Die roten Linien repräsentieren die Quartile der Standard-Methode.

	F F F	T F F	T F T	T T F	T T T	Varianz
1	76.05	-10.79	6.30	1.83	26.62	
2	0.00	8.63	13.56	25.29	52.53	0.00282
3	69.18	0.00	5.80	0.20	24.82	0.00124
4	0.00	0.00	14.66	28.86	56.47	0.00284
5	79.76	-10.06	0.00	2.71	27.60	
6	0.00	12.46	0.00	29.91	57.63	0.00281
7	73.05	0.00	0.00	1.11	25.84	0.00121
8	0.00	0.00	0.00	35.83	64.17	0.00283
9	77.24	-10.56	6.39	0.00	26.93	
10	0.00	17.97	17.05	0.00	64.98	0.00292
11	69.33	0.00	5.81	0.00	24.86	0.00125
12	0.00	0.00	20.98	0.00	79.02	0.00296
13	81.62	-9.70	0.00	0.00	28.08	
14	0.00	25.27	0.00	0.00	74.73	0.00291
15	73.93	0.00	0.00	0.00	26.07	0.00121
16	0.00	0.00	0.00	0.00	100.00	0.00296
17	93.71	-7.32	7.60	6.02	0.00	
18	0.00	28.97	21.16	49.86	0.00	0.00276
19	88.12	0.00	7.19	4.69	0.00	0.00107
20	0.00	0.00	28.01	71.99	0.00	0.00279
21	99.01	-6.28	0.00	7.27	0.00	
22	0.00	38.56	0.00	61.44	0.00	0.00274
23	93.93	0.00	0.00	6.07	0.00	0.00103
24	0.00	0.00	0.00	100.00	0.00	0.00278
25	98.45	-6.39	7.94	0.00	0.00	
26	0.00	65.27	34.73	0.00	0.00	0.00288
27	92.49	0.00	7.51	0.00	0.00	0.00108
28	0.00	0.00	100.00	0.00	0.00	0.00296
29	105.09	-5.09	0.00	0.00	0.00	
30	0.00	100.00	0.00	0.00	0.00	0.00287
31	100.00	0.00	0.00	0.00	0.00	0.00104

Tabelle C.9: Tabelle der Kandidaten für eine optimale Aufteilung von 100 zu ziehenden Texten im US-Banken Beispiel unter Berücksichtigung der ersten beiden Stichproben. Die Spalten geben jeweils die Schnittmengen an, wobei die Spaltenüberschriften für die drei Subkorpora „großer Banken-Korpus“, „kleiner Banken-Korpus“ und „US-Banken-Korpus“ kennzeichnen, ob die Schnittmenge Teil des Korpus ist (T) oder nicht (F). In der letzten Spalte sind die Varianzschätzer vermerkt, falls sie sinnvoll gebildet werden können.

	F F F	T F F	T F T	T T F	T T T	Varianz
1	106.04	-24.77	-0.40	-3.34	22.48	
2	0.00	-0.74	7.12	30.02	63.59	
3	90.34	0.00	-1.24	-7.03	17.93	
4	0.00	0.00	7.06	29.72	63.22	0.00282
5	105.80	-24.81	0.00	-3.40	22.41	
6	0.00	0.99	0.00	32.44	66.57	0.00283
7	89.53	0.00	0.00	-7.23	17.69	
8	0.00	0.00	0.00	32.88	67.12	0.00283
9	103.83	-25.14	-0.52	0.00	21.84	
10	0.00	9.18	10.24	0.00	80.58	0.00285
11	85.11	0.00	-1.52	0.00	16.41	
12	0.00	0.00	11.66	0.00	88.34	0.00286
13	103.47	-25.20	0.00	0.00	21.73	
14	0.00	12.96	0.00	0.00	87.04	0.00287
15	83.93	0.00	0.00	0.00	16.07	0.00125
16	0.00	0.00	0.00	0.00	100.00	0.00287
17	121.45	-22.15	0.41	0.29	0.00	
18	0.00	22.79	14.50	62.71	0.00	0.00309
19	104.26	0.00	-0.50	-3.76	0.00	
20	0.00	0.00	18.70	81.30	0.00	0.00310
21	121.75	-22.10	0.00	0.36	0.00	
22	0.00	28.86	0.00	71.14	0.00	0.00311
23	103.85	0.00	0.00	-3.85	0.00	
24	0.00	0.00	0.00	100.00	0.00	0.00312
25	121.69	-22.11	0.43	0.00	0.00	
26	0.00	70.54	29.46	0.00	0.00	0.00316
27	100.69	0.00	-0.69	0.00	0.00	
28	0.00	0.00	100.00	0.00	0.00	0.00319
29	122.05	-22.05	0.00	0.00	0.00	
30	0.00	100.00	0.00	0.00	0.00	0.00319
31	100.00	0.00	0.00	0.00	0.00	0.00135

Tabelle C.10: Tabelle der Kandidaten für eine optimale Aufteilung von 100 zu ziehenden Texten im US-Banken Beispiel unter Berücksichtigung der drei Stichproben. Die Spalten geben jeweils die Schnittmengen an, wobei die Spaltenüberschriften für die drei Subkorpora „großer Banken-Korpus“, „kleiner Banken-Korpus“ und „US-Banken-Korpus“ kennzeichnen, ob die Schnittmenge Teil des Korpus ist (T) oder nicht (F). In der letzten Spalte sind die Varianzschätzer vermerkt, falls sie sinnvoll gebildet werden können.

	Coherence	Distanz	Wort 1	Wort 2	Wort 3	Wort 4
d 187 t 75	-202.20	0.00	ezb	staatsanleihen	anleihen	griechenland
d 188 t 54	-214.10	0.45	euro	ezb	griechenland	milliarden
d 189 t 25	-257.30	0.51	staatsanleihen	spanien	italien	griechenland
d 190 t 10	-208.00	0.51	griechenland	euro	schulden	spanien
d 186 t 36	-241.20	0.56	ezb	zentralbank	staatsanleihen	draghi
d 188 t 53	-246.10	0.57	banken	griechenland	geld	schulden
d 186 t 77	-202.40	0.61	griechenland	milliarden	athen	euro
d 189 t 3	-230.60	0.63	griechenland	milliarden	banken	euro
d 186 t 34	-291.80	0.67	banken	geld	milliarden	kredite
d 189 t 31	-202.80	0.67	euro	griechenland	europa	staaten
d 190 t 85	-256.20	0.67	anleihen	prozent	anleger	sagt
d 187 t 42	-216.20	0.68	griechenland	euro	staaten	europa
d 186 t 2	-204.10	0.69	griechenland	europa	euro	deutschland
d 190 t 67	-262.20	0.69	ezb	euro	zentralbank	draghi

Tabelle C.11: Zum EZB Thema ähnliche Themen t für die fünf untersuchten Durchläufe d 186 bis 190. Die erste Zeile enthält das ausgewählte Thema, die übrigen ähnliche Themen. Gezeigt werden jeweils vier für das Thema repräsentative Wörter.

	Coherence	Distanz	Wort 1	Wort 2	Wort 3	Wort 4
d 190 t 4	-187.20	0.00	hypo	alpe	adria	oesterreich
d 186 t 48	-230.10	0.43	hypo	adria	alpe	bayernlb
d 187 t 41	-192.30	0.50	bayernlb	hypo	adria	alpe
d 189 t 8	-193.80	0.56	bayernlb	landesbank	hypo	alpe
d 188 t 68	-188.40	0.59	bayernlb	landesbank	hypo	adria
d 188 t 63	-269.70	0.63	staatsanwaltschaft	millionen	gribkowsky	ecclestone
d 187 t 74	-273.40	0.67	staatsanwaltschaft	bank	wegen	millionen
d 189 t 42	-258.20	0.69	staatsanwaltschaft	gribkowsky	gericht	ecclestone
d 189 t 72	-280.80	0.69	bank	staatsanwaltschaft	worden	hsh

Tabelle C.12: Zum Landesbanken Thema ähnliche Themen t für die fünf untersuchten Durchläufe d 186 bis 190. Die erste Zeile enthält das ausgewählte Thema, die übrigen ähnliche Themen. Gezeigt werden jeweils vier für das Thema repräsentative Wörter.

Eidesstattliche Erklärung

Hiermit erkläre ich, Lars Koppers, dass ich die vorliegende Arbeit mit dem Titel „Statistische Methoden zur Validierung von Inhaltsanalysen“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Dissertation ist bisher keiner anderen Fakultät vorgelegt worden. Ich erkläre, dass ich bisher kein Promotionsverfahren erfolglos beendet habe und dass keine Aberkennung eines bereits erworbenen Doktorgrads vorliegt.

Ort, Datum	Lars Koppers