Humboldt-Universität zu Berlin

Philosophische Fakultät

Institut für Geschichtswissenschaften

Erstgutachter: Prof. Dr. Rüdiger Hohls

Zweitgutachter: Prof. Dr. Alexander Nützenadel

Masterarbeit

Topic Modeling für Historiker

Vorgelegt von:

Florian Müller

E-Mail: mullerfl@hu-berlin.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Einleitung

Das Digitale, der Umgang mit elektronischen Medien wie auch die Arbeit mit einem elektronischen Gerät zur Datenverarbeitung, ist nicht nur aus dem heutigen Alltag eines jeden Menschen kaum noch wegzudenken, sondern ist auch ein fester Bestandteil der geisteswissenschaftlichen Forschung geworden. „Eine stetig wachsende Zahl von Quellen liegt inzwischen in digitaler Form vor, Informationen über Archiv- und Bibliotheksbestände sind deutlich leichter erreichbar als früher und die wissenschaftliche Kommunikation, das Exzerpieren, Ordnen, Dokumentieren, Schreiben wie auch das Publizieren erfolgt zunehmend digital."[1] Die Vernetzung der Bibliotheken und Archive lässt die Welt der Forschung immer enger zusammenrücken. Statt in ein Archiv zu reisen, wird das Archiv digital zugesendet. Das Digitale gleicht vielen Menschen als ein neuer Gott, der die irdischen Grenzen zu sprengen vermag,[2] andere fürchten nicht absehbare diabolische Konsequenzen einer stetig voranschreitenden Globalisierung.[3]

Die Bemühungen, Maschinen der elektronischen Datenverarbeitung zur geisteswissenschaftlichen Forschung zu nutzen, sind älter als die heute dafür übliche Bezeichnung „Digital Humanities" bzw. „digitalen Geisteswissenschaften" an sich. Ist die heutige Bezeichnung auf marketingtechnische Überlegungen Anfang der 2000er Jahre zurückzuführen,[4] lassen sich die ersten Arbeiten mit Computern und digitalen Daten in der geisteswissenschaftlichen Forschung bereits bis in die 40er und 50er Jahre des letzten Jahrhunderts zurückverfolgen.[5]

In den Geschichtswissenschaften firmierten die Pioniere der elektronischen Datenverarbeitung bis zum Anfang des neuen Jahrtausends unter dem Namen der historischen Fachinformatik, ehe sie sich, wahrscheinlich mit Blick auf interdisziplinäre Synergieeffekte, unter dem Dach der Digital Humanities sammelten.[6] Das Ziel der digitalen geisteswissenschaftlichen Disziplinen ist die „Entwicklungen und Verfahren der Informatik und Informationswissenschaft auf ihre Verwendbarkeit in den Geisteswissenschaften zu prüfen oder zu adaptieren und anzupassen."[7] Traditionell handelt es sich bei den Anwendungen meist um Hilfsmittel zur Erschließung, Codierung und Auswertung von Textkorpora. In den vergangenen Jahren haben aber auch Verfahren zur Visualisierung von Informationen eine steigende Bedeutung erfahren.[8]

Da sich viele Werkzeuge der digitalen Geisteswissenschaften häufig nicht ohne weitere Anpassung für die historische Forschung anwenden lassen, haben sich unter dem Dach der Digital Humanities die

[1] Rüdiger Hohls, Digital Humanities und digitale Geschichtswissenschaften, in: Rüdiger Hohls, Thomas Meyer und Wilfried Enderle, et al. (Hrsg.), Clio-Guide. Ein Handbuch zu digitalen Ressourcen für die Geschichtswissenschaften, 2018, A.1-1 - A.1-34, hier S. 3.
[2] Vgl. Joris van Eijnatten, Toine Pieters und Jaap Verheul, Big Data for Global History: The Transformative Promise of Digital Humanities, in: BMGN - Low Countries Historical Review 128, 2013, Nr: 4, S. 55–77, hier S. 56, online verfügbar unter: http://www.bmgn-lchr.nl/index.php/bmgn/article/view/9350.
[3] Vgl. Hohls, Digital Humanities und digitale Geschichtswissenschaften, S. 1.
[4] Vgl. Kathleen Fitzpatrick, The Humanities, Done Digitally, in: Matthew K. Gold (Hrsg.), Debates in the digital humanities, Minneapolis, 2012, S. 12–15, hier S. 12–13.
[5] Vgl. R. Busa, The annals of humanities computing: The index Thomisticus: Computers and the Humanities, in: Comput Hum 14, 1980, Nr: 2, S. 83–90, hier S. 84. online verfügbar unter: doi.org/10.1007/BF02403798.
[6] Vgl. Jörg Wettlaufer, Neue Erkenntnisse durch digitalisierte Geschichtswissenschaft(en)? Zur hermeneutischen Reichweite aktueller digitaler Methoden in informationszentrierten Fächern, S. 1.
[7] Hohls, Digital Humanities und digitale Geschichtswissenschaften, S. 3–4.
[8] Vgl. Rafael C. Alvarado, The Digital Humanities Situation, in: Matthew K. Gold (Hrsg.), Debates in the digital humanities, Minneapolis, 2012, S. 50–55, hier S. 51.

Digital Histories bzw. digitalen Geschichtswissenschaften als eigenständige Spezialisierung formiert, „die den spezifischen Methoden und Anforderungen der Forschung in den historischen Wissenschaften entspricht."[9] Die digitalen Historiker befinden sich hierbei innerhalb der historischen Diszplin zwischen den Polen „offensiver Ablehnung grundsätzlicher digitaler Arbeitstechniken und IT-gestützter Forschungsmethoden einerseits und der überschwänglichen Verkündung eines neuen Zeitalters andererseits."[10] . Die Euphoriker ersehnen sich in vielen Fällen eine quantitative Wende herbei, mit der sich Ergebnisse der historischen Forschung im Speziellen und geisteswissenschaftlichen Forschung im Allgemeinen auf Modelle und Nachweisbarkeit stützen lassen.[11] Skeptiker verweisen hingegen, durchaus begründet, auf die sehr überschaubare Liste der Erfolge, die die Verwendung von digitalen Werkzeugen bisher vorweisen kann.[12] Bestärkt werden Skepsis und abwartende Haltung dadurch, dass die Forderung nach mehr Objektivität in den Geschichtswissenschaften ein wiederkehrendes Phänomen ist. Mittels adaptierter Werkzeuge anderer Disziplinen sollten dabei die Arbeit mit historischen Materialien und ihre Auswertung auf eine zahlengestützte, valide, vielleicht gar modellgestützte Grundlage gestellt werden. Doch sowohl die Bemühungen der Gesellschaftsgeschichte mit den Werkzeugen der Soziologie[13] als auch die der Cliometrie mit den Werkzeugen der Wirtschaftswissenschaften[14] müssen als gescheitert angesehen werden. Ein Ruf nach mehr Objektivität mittels eines Digital Turns muss also fast schon zwangsläufig „zu einer Abwehrhaltung bei den anderen Vertretern der Profession [führen]."[15] Diese stehen dabei zumeist weniger der Quantifizierung der Quellen, sondern vielmehr der reinen datengestützten Analyse dergleichen ohne weitere hermeneutische Betrachtung kritisch gegenüber.[16] Einer Unterstützung bei der Erschließung der stetig wachsenden Menge an Texten und historischen Quellenmaterial ist man dagegen durchaus positiv aufgeschlossen.

Der aktuell von vielen Vertretern der Profession getragene Kompromiss der *hybriden Methoden der Digital Histories* sieht vor, dass, besonders mit Blick auf die traditionellen Mitglieder der historischen Profession, die Interpretation weiter Aufgabe des klassischen Historikers bleibt und die von den Digital Histories bereitgestellten Werkzeuge die hermeneutische Arbeit unterstützen wird, diese aber nicht ersetzen soll.[17] Dies widerstrebt aber vielen digitalen Forschern in Zeiten ökonomisch orientierter Reputationssysteme. Sie befürchten, dass eine erfolgreiche Etablierung einer Anwendung im

[9] Wettlaufer, Neue Erkenntnisse durch digitalisierte Geschichtswissenschaft(en)? Zur hermeneutischen Reichweite aktueller digitaler Methoden in informationszentrierten Fächern, S. 2.
[10] Thomas Meyer, Digitale Werkzeuge, in: Rüdiger Hohls, Thomas Meyer und Wilfried Enderle, et al. (Hrsg.), Clio-Guide. Ein Handbuch zu digitalen Ressourcen für die Geschichtswissenschaften, 2018, A.2-1 - A.2-45, hier S. 1.
[11] Vgl. Armand Marie Leroi, Cicero zählen: Algorithmus oder Kritik? Plädoyer für eine universelle Kulturtheorie., in: Süddeutsche Zeitung, Nr. 54 vom 06. 3. 2015.
[12] Vgl. Hohls, Digital Humanities und digitale Geschichtswissenschaften, S. 10–11.
[13] Vgl. Chris Lorenz, Wozu noch Theorie in der Geschichte?: Über das ambivalente Verhältnis zwischen Gesellschaftsgeschichte und Modernisierungstheorie, in: Volker Depkat, Matthias Müller und Andreas Urs Sommer (Hrsg.), Wozu Geschichte(n)? Geschichtswissenschaft und Geschichtsphilosophie im Widerstreit, Stuttgart, 2004, S. 117–143, hier S. 124.
[14] Vgl. Gerold Ambrosius, Werner Plumpe und Richard Tilly, Wirtschaftsgeschichte als interdisziplinäres Fach, in: Gerold Ambrosius, Dietmar Petzina und Werner Plumpe (Hrsg.), Moderne Wirtschaftsgeschichte. Eine Einführung für Historiker und Ökonomen, München, 2006, S. 9–38, hier S. 25–26.
[15] Hohls, Digital Humanities und digitale Geschichtswissenschaften, S. 10.
[16] Vgl. Wettlaufer, Neue Erkenntnisse durch digitalisierte Geschichtswissenschaft(en)? Zur hermeneutischen Reichweite aktueller digitaler Methoden in informationszentrierten Fächern, S. 3 und; van Eijnatten et al., Big Data for Global History: The Transformative Promise of Digital Humanities, S. 75–76.
[17] Vgl. Wettlaufer, Neue Erkenntnisse durch digitalisierte Geschichtswissenschaft(en)? Zur hermeneutischen Reichweite aktueller digitaler Methoden in informationszentrierten Fächern, S. 4.

alltäglichen Forschungseinsatz zu einer Loslösung von ihrem ursprünglichen Urheber führen würde, [18] wodurch dem Forscher keinerlei den Forschungsaufwand rechtfertigende Reputation verbleiben würde.[19] Sie schlagen daher Konzepte wie die Datenautorenschaft[20] vor oder fordern neue Formen der wissenschaftlichen Kritik.[21] Diese können aber nur bedingt als Lösung dienen. Die neuen Formen der Kritik werden zwar vermutlich im vergleichsweise kleinen Kreis der digitalen Geisteswissenschaften sich als Kommunikationsmedium etablieren, in diesen Kreisen aber auch verharren, da die Kritik von Algorithmen oder Datenerhebungen ein wichtiges disziplininternes Thema bleibt und für den traditionellen, geisteswissenschaftlichen Anwender i.d.R. schwer nachvollziehbar ist. Eine Datenautorenschaft hingegen bedeutet zunächst nur einen Mehrwert für den Urheber, da dieser die für ihn wichtige Reputation erhält, aber keinen zusätzlichen Mehrwert für den Anwender. Solche Forderungen wären legitim, wenn die traditionelle Forschung nicht ohne aufbereitete digitale Daten und Konzepte fortgesetzt werden könnte. Da aktuell aber eher die digitale Disziplin Werbung für ihre Arbeit machen muss, ist der Erfolg bei einem Mehraufwand für den Nutzer, der über das bisher bekannte Maß an Herkunftsnennung hinaus geht, egal wie klein und gerechtfertigt dieser auch ist, pessimistisch einzuschätzen. Im Zweifel könnte ein Mehraufwand für den Nutzer eine schlichte Nichtnutzung der angebotenen Daten bedeuten.

Schaut man in die Liste der bisher abgeschlossenen Projekte der digitalen Geisteswissenschaften, so handelt es sich in den besten Fällen um aufbereitete Datensammlungen,[22] Datenbankentools[23] oder linguistische Anwendungen, [24] die teils aus lizenzrechtlichen Gründen, teils aus eigenen urheberrechtlichen Interessen für externe Anwender zum größten Teil verschlossen bleiben oder nur als eine Software-as-a-Service Lösung angeboten werden. Eine Möglichkeit der Erweiterung der bestehenden Anwendungen um eigene Daten oder eigene Analyseschritte wird für projektfremde Forscher i.d.R. nicht angeboten. Dies verhindert jedoch eine Anpassung an andere Fragestellungen, die die ursprünglichen Entwickler nicht bedacht haben. Dadurch muss, statt auf bestehenden Strukturen aufbauen zu können, für jedes Projekt und jede Fragestellung eine neue Anwendung entwickelt werden. Die verschlossenen Quelltexte erschweren nicht nur die von den Vertretern der Disziplin gewünschte neue Form der Kritik zusätzlich, da so keine Aussage über die praktische Umsetzung von theoretischen Modellen möglich ist. Sie bedeuten aufgrund der ständigen Neuentwicklungen von Anwendungen für den klassisch arbeitenden, interessierten Anwender notwendigerweise eine kontinuierliche Einarbeitung in neue

[18] Vgl. Schmitt, Martin, HT 2018: Digital Humanities in der Analyse gespaltener Gesellschaften. Beispiele aus der Praxis. Erschienen am 07.12.2018. in: H-Soz-Kult (Hrsg.), online verfügbar unter: https://www.hsozkult.de/conferencereport/id/tagungsberichte-8009. Zuletzt geprüft am: 10.07.2019.

[19] Vgl. Müller, Andreas und Purschwitz, Anne, HT 2018: Forschungsdaten: Rechtliche Herausforderungen und wissenschaftliche Reputation. Forschungsdatenmanagement als Bestandteil einer neuen Wissenschaftskultur. Erschienen am 30.11.2018. in: H-Soz-Kult (Hrsg.), online verfügbar unter: https://www.hsozkult.de/conferencereport/id/tagungsberichte-7988. Zuletzt geprüft am: 10.07.2019.

[20] Vgl. ebenda.

[21] Vgl. Fiedler, Maik, HT 2018: Für Skeptiker und Enthusiasten: Was ist und zu welchem Ende nutzt das ›Digitale‹ in den Geschichtswissenschaften? Erschienen am 16.11.2018. in: H-Soz-Kult (Hrsg.), online verfügbar unter: https://www.hsozkult.de/conferencereport/id/tagungsberichte-7960. Zuletzt geprüft am: 10.07.2019.

[22] Staatsbibliothek zu Berlin Zeitungsabteilung, Zeitungsinformationssystem ZEFYS - Staatsbibliothek zu Berlin. Erschienen am 01.01.2018, online verfügbar unter: http://zefys.staatsbibliothek-berlin.de/ddr-presse. Zuletzt geprüft am: 11.08.2019.

[23] Berlin Brandenburgische Akademie der WIssenschaften, D*/DTA: DiaCollo, online verfügbar unter: http://kaskade.dwds.de/dstar/dta/diacollo/. Zuletzt geprüft am: 11.08.2019.

[24] CLARIN-D, WebLicht, online verfügbar unter: https://weblicht.sfs.uni-tuebingen.de/weblicht/. Zuletzt geprüft am: 11.08.2019.

Anwendungen. Eine positive Ausnahme bildet hierbei der Topic Explorer von DARIAH-DE, dessen Quellcode in einem öffentlichen GitRepository frei zugänglich ist.[25]

Es muss durch die Ausarbeitung ein Mehrwert für den Anwender geschaffen werden, der dem Bedarf nach Veröffentlichungen und zitierfähigen Arbeiten auf Seiten des digitalen Forschers Rechnung trägt, da diese wichtige Voraussetzungen für Anstellungen und weitere Projektbewilligungen im universitären Alltag geworden sind. In der Informatik wird die Konstruktion neuer Algorithmen i.d.R. mit einem Paper begleitet, in dem meist sehr kurz und knapp die theoretische Grundlage und die Funktionsweise der neuen Anwendung beschrieben werden. Die Ergebnisse werden dabei meist nur auf Plausibilität und auf die Auswirkungen verschiedener Parameter hin untersucht. Der Programmierer verfügt so über ein Dokument, dass der Anwender bei Verwendung der Anwendung zitieren kann und zugleich vom Anwender wahrgenommen, zumeist sogar für die Fundierung der eigenen wissenschaftlichen Arbeit gebraucht wird. In den Arbeiten der Informatik kann der Autor allerdings davon ausgehen, dass der Leser der Artikel über ein gewisses Maß an informationstechnischen Vorwissen verfügt. Die Digital Humanities befinden sich jedoch an einer Schnittstelle und müssen davon ausgehen, dass der interessierte Nutzer mit Konsolenanwendungen und Skriptsprachen noch nie in Berührung gekommen ist. Es ist daher anzunehmen, dass allein eine komplette mathematische Herleitung der zugrundeliegenden Berechnungsmodelle dem Nutzer in seinem Arbeitsalltag nicht helfen wird. Dies bedeutet nicht, dass keine theoretische Einordnung der Grundlagen erfolgen darf, sie soll aber die für die Anwendung wichtigsten Punkte zusammenfassen und bei Bedarf auf die weiterführenden Textstellen verweisen. Es wird zudem neben der Offenlegung und Erläuterung des zugrundeliegenden Modells besonders eine Gebrauchsanleitung benötigt, die es dem Nutzer ermöglicht, selber mithilfe der Anwendungen die von ihn gewünschten Daten generieren zu können.

Ein hoher Bedarf in der geisteswissenschaftlichen Forschung besteht aktuell in der Verarbeitung des stetig wachsenden Stroms potentieller historischer Quellen, da dieser schon jetzt mit den klassischen Methoden nur noch punktuell betrachtet werden kann.[26] Die für eine automatisierte Erfassung von Dokumenten notwendigen Algorithmen sind unter dem Begriff des Distant Readings bekannt.[27]

Das aktuell größte Potential, Distant Reading zu ermöglichen, bieten Topic Modeling Anwendungen: Hierbei handelt es sich um Anwendungen, die Dokumente nach ihrer inhaltlichen Nähe ordnen. Dies ermöglicht die Erfassung unbekannter Korpora, das Aufdecken der in ihnen versteckten Strukturen und das Suchen nach für den Forscher relevanten Dokumenten. Besonders für Forscher, deren Quellen oft nicht aus anglizistischen Regionen stammen, bieten diese Werkzeuge viel Potential, da die Modelle keinerlei sprachliches Vorwissen benötigen, sondern allein basierend auf der Datengrundlage ihre Berechnungen durchführen können. Im Idealfall wären Topic Modeling Anwendungen dann beispielsweise „zur Erkundung der gesamten Geschichte der New York Times [nutzbar]. Auf einer allgemeinen Ebene korrespondieren einige Themen vermutlich mit den Sektionen der Zeitung –

[25] DARIAH-DE, Topics Explorer, online verfügbar unter: https://dariah-de.github.io/TopicsExplorer/. Zuletzt geprüft am: 05.07.2019.
[26] Vgl. Matthew Wilkens, Canons, Close Reading, and the Evolution of Method, in: Matthew K. Gold (Hrsg.), Debates in the digital humanities, Minneapolis, 2012, S. 249–258, hier S. 251.
[27] Vgl. DARIAH-DE, Handbuch Digital Humanities: Anwendungen, Forschungsdaten und Projekte, 2015, S. 9, online verfügbar unter: https://handbuch.tib.eu/w/images/2/2c/DH-Handbuch.pdf. Zuletzt geprüft am: 04.07.2019.

Außenpolitik, Innenpolitik, Sport. Wir könnten nun in ein Thema von Interesse hineinzoomen, um verschiedene Aspekte aufzudecken – chinesische Außenpolitik, der Konflikt im Mittleren Osten, das US-amerikanische Verhältnis zu Russland. Wir könnten dann durch die Zeit navigieren, um Veränderungen aufzuzeigen wie z.B. die Veränderungen beim Konflikt im Mittleren Osten über die letzten 50 Jahre. Und bei dieser Erkundung könnten wir zu den relevanten Originalartikeln weitergeleitet werden."[28]

Leider fehlt bisher eine gute Übersicht über die Möglichkeiten und Grenzen dieser Programme. Im Clio Guide heißt es nur kurz, dass „sich das Thema zu einem sehr populären Forschungsfeld [entwickelt], Analysen von Twitter-Nachrichten oder Nutzertexten aus dem Netz sollen Aufschluss über Thementrends geben."[29] Im Anschluss folgt zwar noch die Nennung von zwei für diese Zwecke anwendbaren Werkzeugen, jedoch fehlen Anlaufpunkte für den interessierten Nutzer, wo er sich in die komplexe Materie des Topic Modelings einarbeiten könnte. Ohne eine Einarbeitung ist es aber absehbar, dass falsche Anforderungen und Erwartungen an die Anwendungen gestellt werden, die diese gar nicht in der Lage sind bereitzustellen.

Das Ziel dieser Arbeit ist es daher, diese bisher fehlende Einführung in die Arbeit mit Topic Modeling Anwendungen zu geben. Die Einführung verfolgt dabei zwei Ziele. Zum einen soll sie als Grundlage für einen Diskurs zur Arbeit mit Topic Modeling Anwendungen in den digitalen Geschichtswissenschaften dienen. Zum anderen soll die Arbeit es dem Leser ermöglichen, sich selbstständig in den Umgang mit dieser Art von Distant Reading einzuarbeiten. Es wird sich hierbei zu einem großen Teil auf die den wahrscheinlichkeitsbasierten Verfahren zuzuordnende Latente Dirichlet Allokation konzentriert, wie es auch beim Topic Explorer von DARIAH zum Einsatz kommt. Die Gründe für die Wahl dieses Verfahrens und die Unterschiede zu anderen Verfahren sollen dabei zunächst genau untersucht werden. Im Anschluss wird die Latente Dirichlet Allokation und der auf ihr aufbauende hierarchische Dirichlet Prozess praxisnah erläutert und jeweils eine Umsetzung vorgestellt. Bei den Umsetzungen wurde darauf geachtet, dass es sich um Anwendungen handelt, die möglichst wenig zusätzliche Software benötigen, auf allen bekannten Betriebssystemen verwendbar sind und unter einer Open Source Lizenz bereitgestellt werden. Die Wahl fiel auf zwei Anwendungen, die aus dem direkten Umfeld des Teams hinter der LDA stammen. Die Fragen, die die Analyse der Anwendungen leitet, ist, welche Daten der Anwender benötigt, um die gewünschten Ergebnisse zu generieren und welche Ergebnisse der Nutzer überhaupt von den Anwendungen erwarten darf?

Die Analyse der Anwendungen erfolgt dadurch unter zwei Gesichtspunkten. Sie ist sowohl an den Programmierer gerichtet und weist auf Besonderheiten und wichtige Elemente im Quelltext hin. Sie ist aber auch als eine Anleitung für geisteswissenschaftliche Anwender konzipiert und soll dem Nutzer helfen, die gewünschten Analysen starten und auswerten zu können.

In der Folge wird die Auswertung der Ergebnisse der Anwendungen näher betrachtet. Hierbei dient die parallel zur schriftlichen Arbeit selbsterstellte Anwendung TopModHis (**Top**ic **Mod**eling für **His**toriker) als Hilfsmittel, in dem die wichtigsten Operationen zur Ergebnisanalyse gebündelt wurden. Dies soll

[28] David M. Blei, Probabilisitc Topic Models, in: Communications of the ACM 55, 2012, Nr: 4, S. 77–84, hier S. 77, online verfügbar unter: https://cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext. Zuletzt geprüft am: 01.03.2019, eigene Übersetzung.
[29] Meyer, Digitale Werkzeuge, S. 37.

dem geisteswissenschaftlichen Anwender eine Reihe von repetitiven Schritten wie das Erstellen von Topwortlisten abnehmen und die Arbeit mit größeren Korporas beschleunigen. Daher soll der Umgang mit der Anwendung TopModHis in einem eigenen Kapitel ausführlich betrachtet werden.

Im Anschluss an die Betrachtung der Anwendungen werden zwei Szenarien vorgestellt, die die jeweiligen Stärken der beiden Topic Modeling Anwendungen hervorheben und die Aufgabenfelder noch einmal verdeutlichen sollen. Die Ergebnisse werden dabei bewusst nur bis zu dem Punkt betrachtet, bis zu dem die Ergebnisse auch ohne eine historische Interpretation eingeordnet und bewertet werden können. Die darüber hinausführende historische Interpretation der Ergebnisse wäre, ganz dem Prinzip des hybriden Ansatzes folgend, Aufgabe von Vertretern der historischen Forschung.

Topic Modeling Ansätze im Vergleich

Topic Modeling Anwendungen werden allgemein in distanzbasierte und wahrscheinlichkeitsbasierte Anwendungen unterteilt. In Ausarbeitungen zu distanzbasierten Algorithmen wird dabei häufig die überlegene Skalierbarkeit dieser Anwendungen bei besonders großen Datenmengen betont.[30] Woher diese Vorteile kommen und welche Vorteile dagegen wahrscheinlichkeitsbasierte Anwendungen für den Einsatz in der historischen Forschung haben, ist Gegenstand dieses Kapitels.

Distanzbasierte Algorithmen

Distanzbasierte Anwendungen basieren auf der Idee, dass Wörter und die sie enthaltenen Dokumente verschiedene Eigenschaften teilen und bei anderen Eigenschaften Unterschiede aufweisen. Die Unterschiede werden mittels Distanzmaßen gemessen. Diese sind dabei: [31]

1) Immer positiv
2) Nur dann null, wenn die beiden Messpunkte gleich sind
3) Unabhängig von der Richtung der Messung
4) Der kürzeste Pfad zwischen den beiden Messpunkten (Dreiecksungleichung)

Zur Veranschaulichung dieser Eigenschaften sei das bekannteste und meist verwendete Maß für eine Distanz in der Mathematik, die euklidische Norm, betrachtet. Hier werden die Differenzen in den einzelnen Dimensionen durch Subtraktion ermittelt, diese quadriert und aufsummiert und anschließend die Wurzel gebildet.[32] Die Punkte P(1,-3,5) und Q(15,5,5) hätten also eine Distanz von ca. 16 Maßeinheiten, zusammengesetzt aus

$$\sqrt{(15-1)^2+(5-(-3))^2+(5-5)^2}=\sqrt{14^2+8^2+0^2}=\sqrt{260}=16{,}12\ldots$$

[30] Vgl. Boris Lorbeer und Ana Kosareva et al., Variations on the Clustering Algorithm BIRCH, in: Big Data Research 11, 2018, S. 44–53, hier S. 44. online verfügbar unter: doi.org/10.1016/j.bdr.2017.09.002.; Tomas Mikolov und Kai Chen et al., Efficient Estimation of Word Representations in Vector Space, 1/16/2013, S. 2, online verfügbar unter: http://arxiv.org/pdf/1301.3781v3. Zuletzt geprüft am: 11.07.2019.

[31] Vgl. Jurij Leskovec, Anand Rajaraman und Jeffrey D. Ullman, Mining of massive datasets, Cambridge, 2015, 2nd ed., 1st repr, S. 87.

[32] Vgl. Gerhard Jank und Hubertus Theodorus Jongen, Höhere Mathematik für Maschinenbauer: Skript zur Vorlesung, 1999, S. 55.

Durch das Quadrieren der Differenzen erhalten wir im reellen Zahlenraum immer positive Werte und die Richtung der Subtraktion wird egal (Eigenschaften 1 und 3). Da ferner nur positive Werte aufaddiert werden, kann die Summe nur null ergeben, wenn alle Summanden der Gleichung null sind (Eigenschaft 2). Für einen vollständigen Beweis der vierten Eigenschaft sei auf die einschlägige Literatur verwiesen, die Eigenschaft erschließt sich aber daraus, dass es laut Definition von Vektoren nur einen Vektor geben kann, der die direkte Verschiebung von P nach Q erfüllen kann.[33]

Die distanzbasierten Anwendungen werden in der Praxis häufiger verwendet als die entsprechenden wahrscheinlichkeitsbasierten Pendants.[34] Dies liegt sicherlich z.T. in der performanten Überlegenheit einiger Vertreter dieser Kategorie gegenüber den Vertretern der wahrscheinlichkeitsbasierten Modelle begründet, kann aber auch der Komplexität der wahrscheinlichkeitsbasierten Algorithmen geschuldet sein, bei denen die Verfahren bei aller Algebra durchaus auch einmal auf einer gesunden Prise Intuition beruhen können.[35] Es werden zwei Arten von distanzbasierten Verfahren unterschieden: die partitionierenden Verfahren und die hierarchischen Verfahren.[36]

Bei den partitionierenden Verfahren wird, ausgehend von einer Startverteilung, iterativ versucht, die Abstände zwischen den einzelnen Gruppen über die Verschiebung von Datenpunkten zu optimieren. Die Abstände zwischen den Gruppen werden entweder über ihren Durchschnitt gemessen oder über das Element, das am zentralsten in den jeweiligen Gruppen gelegen ist.[37] Ein häufig eingesetztes Verfahren ist das KMEANS Verfahren, bei dem zunächst X zufällige Punkte für X zuvor festgelegte Gruppen ausgewählt werden. Im Anschluss wird iterativ ein neuer Punkt aus der Menge an verfügbaren Punkten gezogen und dem Cluster zugeordnet, dessen Mittelwert dem Datenpunkt am nächsten ist.[38] Der Nachteil an der Grundvariante dieser Algorithmen ist, dass alle Datenpunkte in die zuvor festgelegten Cluster einsortiert werden müssen und es keine Möglichkeit gibt, neue Cluster zu bilden.

Hierarchische Algorithmen dagegen versuchen nicht, das beste Cluster für einen Datenpunkt zu finden, sondern vereinen und spalten mit jeder Iteration die bereits vorhandenen Cluster. Dies hat zumeist den Nachteil, dass bereits vor dem Clustern alle Datenpunkte bekannt sein müssen, damit der Algorithmus dann ausgehend von einer Startverteilung eine Optimierung der Cluster und ihrer Zahl vornehmen kann.[39] Zudem kann es passieren, dass zueinander gehörende Begriffe durch eine ungünstige Ziehung von weiteren Begriffen unterschiedlichen Themen zugeordnet werden oder Cluster entstehen, die eigentlich getrennt werden müssten.

---

[33] vgl. ebenda, S. 52.
[34] Vgl. Tian Zhang, Raghu Ramakrishnan und Miron Livny, BIRCH: A New Data Clustering Algorithm and Its Applications, in: Data Mining and Knowledge Discovery 1, 1997, Nr: 2, S. 141–182, hier S. 144. online verfügbar unter: doi.org/10.1023/A:1009783824328.
[35] Vgl. Michael I. Jordan und Zoubin Ghahramani et al., An Introduction to Variational Methodsfor Graphical Models, in: Machine Learning 37, 1999, Nr: 2, S. 183–233, hier S. 185. online verfügbar unter: doi.org/10.1023/A:1007665907178.
[36] Vgl. Zhang et al., BIRCH, S. 144.
[37] Vgl. ebenda, S. 144.
[38] Vgl. James MacQueen und others, Some methods for classification and analysis of multivariate observations, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA, 1967, S. 281–297, hier S. 283.
[39] Vgl. Zhang et al., BIRCH, S. 144.

Die beschriebenen Nachteile der Grundalgorithmen sind in Weiterentwicklungen mittlerweile z.T. gänzlich aufgehoben worden. Bei den partitionierenden Algorithmen kann man sich z.B. einen Schwellwert vorstellen, der festlegt, ob die Ähnlichkeit von einem Datenpunkt zu dem nächstliegenden Thema groß genug ist. Sollte der Schwellwert unterschritten werden, so eröffnet der betrachtete Datenpunkt ein neues Cluster.[40] Um hierarchische Modelle für unbekannte Daten zu öffnen, muss dagegen durchaus auf Methoden aus den wahrscheinlichkeitsbasierten Algorithmen zurückgegriffen werden.[41]

Normen wie die eingangs erläuterte euklidische Norm können nicht ohne eine weitere Vorverarbeitung für Dokumente angewendet werden, da Buchstaben und Buchstabenketten zunächst keinen Zahlen entsprechen und daher nicht in einem n-dimensionalen reellen Vektorraum darstellbar sind. Die meisten Clusteralgorithmen arbeiten jedoch in einem euklidischen Raum,[42] weshalb es Strategien und Algorithmen bedarf, mit denen die Wörter und Dokumente in Vektoren eines euklidischen Raumes umgewandelt werden können. Ein einfacher Ansatz hierfür ist das Bestimmen der Distanzen zwischen allen im Korpus vorkommenden Begriffen. Jeder Begriff stellt eine von n - Dimension dar und ähnliche Begriffe haben eine ähnliche Distanz zu den anderen Begriffen. Im Rahmen von Data Analysis wurden hierzu die Hamming-Distanz, die Edit- bzw. Levenstein- Distanz und die Jaccard- Distanz entwickelt. Diese Messen, grob vereinfacht, beim Vergleich zweier Zeichenketten die Zahl an gemeinsamen Zeichen und setzen diese auf unterschiedliche Art mit der Zahl an Unterschieden in ein Verhältnis. Dies ermöglicht es, ohne größere Vorverarbeitung eine Distanz zwischen zwei Wörtern oder ganzen Dokumenten angeben zu können.[43] Diese sehr einfachen Vergleiche erlauben eine hohe Geschwindigkeit bei der Vektorisierung von Dokumenten. Der Nachteil dieser drei Alternativmaße ist jedoch, dass sie keinerlei Informationen über Zusammenhänge von Wörtern oder von ganzen Wortketten berücksichtigen. So sind das Tier und das Tor nach dem Prinzip dieser Maße meist näher verwandt als das Dampf- und das Segelschiff oder das Dampfschiff und die Dampfschiffrundfahrt.

Um solche familiären Informationen bei der Überführung von Dokumente in Vektoren eines euklidischen Raumes erhalten zu können, haben sich in den letzten Jahren verschiedene Verfahren entwickelt, die mithilfe von beobachteten Wahrscheinlichkeiten die Nähe von Wörtern zueinander bestimmen. Der GenEx Algorithmus versucht hierbei beispielsweise über das Extrahieren von Schlüsselwörtern eine Bedeutungsmatrix zu erstellen.[44] Schlüsselwörter definieren die Autoren als 5 bis 15 Begriffe, die die wichtigsten Themen eines Dokumentes erfassen.[45]

Noch populärer ist das von Google Mitarbeitern entwickelte Word2Vec Verfahren.[46] Dieses Verfahren erstellt eine Vektorrepräsentation von Dokumenten auf zwei unterschiedliche Arten. Bei dem Continous Bag of Words (CBOW) Verfahren wird von n Begriffen vor dem Begriff und den nächsten n Begriffen,

[40] Vgl. Mark Last, Maxim Stoliar und Menahem Friedman, Clustering-Based Classification of Document Streams with Active Learning, in: Abraham Kandel, Horst Bunke und Mark Last (Hrsg.), Data mining in time series and streaming databases, Singapore, 2018, S. 92–117, hier S. 102.
[41] Vgl. Zhang et al., BIRCH, S. 145.
[42] Vgl. ebenda, S. 145.
[43] Vgl. Leskovec et al., Mining of massive datasets, S. 88–91.
[44] Vgl. Peter D. Turney, Learning Algorithms for Keyphrase Extraction, in: Information Retrieval 2, 2000, Nr: 4, S. 303–336, hier S. 305. online verfügbar unter: doi.org/10.1023/A:1009976227802.
[45] Vgl. ebenda, S. 303.
[46] Google Inc., Google Code Archive - Word2Vec, online verfügbar unter: https://code.google.com/archive/p/word2vec/. Zuletzt geprüft am: 17.07.2019; Tomas Mikolov, Word2Vec. Erschienen am 17.07.2017, online verfügbar unter: https://github.com/tmikolov/word2vec. Zuletzt geprüft am: 28.02.2019.

die in einem Satz dem Begriff folgen, die Wahrscheinlichkeit aller Wörter bestimmt, die diese Kombination zulassen. Bei dem Continous Skip-gram Modell hingegen geht man von einem gegebenen Wort aus und berechnet die Wahrscheinlichkeiten der verschiedenen n Wörter vor dem Wort und die möglichen n Wörter nach dem Wort.[47]

Ein faszinierender Nebeneffekt der Repräsentation nach Word2Vec ist, dass sie mathematische Begriffsoperationen zulassen. Nimmt man den Begriff [König], subtrahiert den Begriff [Mann] und addiert den Begriff [Frau], so erhält man einen Vektor, der stark in der Nähe des Begriffs [Königin] ist.[48] Ähnlich verhält es sich mit der Paarung von Ländern und Hauptstädten.[49]

Wahrscheinlichkeitsbasierte Algorithmen

Die wahrscheinlichkeitsbasierten Topic Modeling Verfahren versuchen über Wahrscheinlichkeitsmaße Dokumente und Wörter verschiedenen Themen zuzuordnen.[50] Bei dieser Art von Modellen wird davon ausgegangen, dass die Reihenfolge der Begriffe in einem Dokument und die Reihenfolge der Dokumente in einem Korpus keine Aussagekraft haben über die Themenzugehörigkeit (Bag-of-Words Assumption). Dies ermöglicht es, im Gegensatz zu den distanzbasierten Verfahren, sehr schlanke und vor allem einfache Vektorrepräsentationen nutzen zu können, in denen nur die Häufigkeiten von Begriffen pro Dokument gezählt werden und keinerlei Informationen über Wahrscheinlichkeitsverteilungen in den Vektoren enthalten sind.[51] Ein recht altes Beispiel für wahrscheinlichkeitsbasierte Algorithmen ist COBWEB, bei dem iterativ ausgehend von einer Wurzel an jedem Knoten in einer Baumhierarchie das Programm vier Auswahlmöglichkeiten hat:

1. Zu einem anderen Knoten gehen
2. Einen neuen Knoten erstellen
3. Zwei Knoten vereinen, um das Element dort dann einzufügen
4. Einen Knoten aufteilen

Die Option mit der höchsten Wahrscheinlichkeit wird dann vom Programm ausgewählt. Das Speichern der Baumstruktur und aller zu den Elementen gehörenden Werte macht das Programm sehr ressourcenintensiv und ungeeignet für große Datenmengen.[52]

Die Menge an Daten zu begrenzen, die die wahrscheinlichkeitsbasierten Modelle erzeugen, war in den folgenden Jahrzehnten eines der Hauptanliegen in dieser Forschungsrichtung. Das latente semantische Indizieren (LSI) war hierbei ein großer Fortschritt, zumal es erste linguistische Merkmale erkennbar machte, jedoch fehlte dem Modell eine statistische Beschreibung von Dokumenten. Mit dem unter den Namen Aspekte Modell oder wahrscheinlichkeitsbasierten LSI (pLSI) bekannten Konzept erfolgte ein weiterer wichtiger Meilenstein auf dem Weg zu den heute üblichen Verfahren. Wörter werden in diesem Modell als Stichproben eines Mischmodells angesehen, dessen Bestandteile mulitnomial verteilt sind.

[47] Vgl. Mikolov et al., Efficient Estimation of Word Representations in Vector Space, S. 4.
[48] Vgl. ebenda, S. 2.
[49] Vgl. ebenda, S. 5.
[50] Vgl. Zhang et al., BIRCH, S. 143.
[51] Vgl. David M. Blei, Andrew Y. Ng und Michael I. Jordan, Latent Dirichlet Allocation, in: Journal of Machine Learning Research 3, 2003, S. 993–1022, hier S. 994, online verfügbar unter: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf. Zuletzt geprüft am: 01.03.2019.
[52] Vgl. Zhang et al., BIRCH, S. 143.

Die einzelnen Bestandteile des Mischmodells können als Themen angesehen werden, aus denen wiederum die einzelnen Wörter stammen. Dies hebt den Ansatz von den distanzbasierten Modellen ab, da dadurch mehrere Themen einem Dokument zugeordnet werden können. Da allerdings das Modell auf eine statistische Beschreibung der Dokumente verzichtet, müssen diese durch eine Reihe von Nummern (den Themen) repräsentiert werden. Mit einer steigenden Zahl an Themen steigt damit auch die Länge dieser Liste. Zudem können die Themen nur Dokumenten zugeordnet werden, für die das Modell trainiert wurde. Diese Nachteile wurden bei der Entwicklung der Latenten Dirichlet Allokation berücksichtigt, die aktuell als Standard bei den wahrscheinlichkeitsbasierten Algorithmen angesehen werden kann und im folgenden Kapitel ausführlich betrachtet werden soll.[53]

Zusammenfassender Vergleich

Distanzbasierte Modelle funktionieren für große Daten und vor allem Echtzeit Daten schneller und sind anpassungsfähiger bei dem Hinzufügen eines neuen Themas. Sie benötigen jedoch eine längere Bearbeitungszeit bei der Erstellung der einzelnen Vektoren, da diese bereits alle notwendigen Informationen über Zusammenhänge von Begriffen beinhalten müssen,[54] bevor der eigentliche Clusteralgorithmus angewendet werden kann. Das Bereithalten aller Informationen bei dem Hinzufügen neuer Datenpunkte zum Modell benötigt außerdem eine große Menge an Arbeitsspeicher, dafür sind die Endergebnisse sehr schlank

Wahrscheinlichkeitsbasierte Anwendungen andererseits kommen in ihrem Vektorabbild ohne Informationen bezüglich statistischer Zusammenhänge aus. Dies ermöglicht zum einen eine vergleichsweise geringe Nutzung von Arbeitsspeicher, führt aber zu großen Datenmengen bei den Ergebnissen, da in diesen alle statistischen Zusammenhänge abgelegt werden müssen. Für jede neue Modellberechnung müssen die alten Modelldaten von der Festplatte geladen werden und die neuen, großen Ergebnisdateien wieder auf die Festplatte geschrieben werden.

Da in den Geschichtswissenschaften vor allem „ruhende“ Daten, sogenannte Data Warehouses, mit wenig veränderlichen Themenclustern die Regel sind, konzentriert sich die Arbeit auf Topic Modeling mit wahrscheinlichkeitsbasierten Modellen. Die historische Forschung erfolgt i.d.R. nicht auf Hochleistungsrechnern mit einem großen Arbeitsspeicher, sondern auf normalen Bürorechnern. Für diese ist ein großer Festplattenspeicher günstiger zu beschaffen als ein großer Arbeitsspeicher und zudem machen die ruhenden Daten eine ständige Neukallibrierung der Modelle unnötig. Daher sprechen sowohl pragmatische als auch ökonomische Gründe für den Einsatz wahrscheinlichkeitsbasierter Topic Modeling Anwendungen in der historischen Forschung.

[53] Vgl. Blei et al., Latent Dirichlet Allocation, S. 994.
[54] Vgl. Last et al., Clustering-Based Classification of Document Streams with Active Learning, S. 100.

Die Latente Dirichlet Allokation und der Hierarchische Dirichlet Prozess im Detail

Der Fokus dieser Arbeit liegt auf zwei Anwendungen, die auf dem Prinzip der wahrscheinlichkeitsbasierten Modelle beruhen. Im Gegensatz zu Vektoren der abstandsbasierten Anwendungen enthalten daher die hier betrachteten Vektoren der Dokumente noch keinerlei Informationen über Wahrscheinlichkeiten. Es handelt sich um Unigramme.[55] Dies bedeutet in der Praxis, dass die Vektorisierung zunächst deutlich schneller verläuft als bei anderen Algorithmen wie den zuvor vorgestellten Word2Vec oder GenEx Algorithmen. Der Geschwindigkeitsvorteil geht allerdings im Anschluss wieder verloren, wenn auf der Grundlage der Dirichlet Verteilung Zusammenhänge zwischen Wörtern und der sie enthaltenen Dokumente nachträglich berechnet werden müssen.

Dieser Abschnitt beginnt mit einer praxisorientierten Einführung in den theoretischen Grundaufbau der beiden Modelle. Praxisorientiert bedeutet dabei, dass auf die Herleitung der einzelnen Formeln so weit wie möglich verzichtet wird und stattdessen vor allem die Bedeutung der verwendeten Formeln und Variablen für die Arbeit mit den Anwendungen erläutert wird. Ziel ist es, einem Anwender ohne tiefgehende mathematische und statistische Vorkenntnisse die Möglichkeit zu geben, die Ergebnisse der Anwendungen interpretieren zu können, die Quelltexte nachvollziehen zu können und mit den unterschiedlichen Ergebnissen arbeiten zu können.

Da der HDP (Hierarchical Dirichlet Process) zu großen Teilen auf der LDA (Latent Dirichlet Allocation) beruht, wird zunächst die LDA erklärt und darauf aufbauend der HDP. Im Anschluss an die theoretische Vorbetrachtung erfolgt jeweils eine Analyse der Anwendung, wie sie in den GitRepositories[56] der Autoren zu finden sind. Anhand von Quelltextanalysen und Testläufen sollen die verschiedenen Einstellungsmöglichkeiten und Funktionsweisen der beiden Programme festgehalten werden, damit diese Dokumentation dem Leser als Nachschlagewerk für die eigene Verwendung dieser Programme oder den Programmen zugrundliegenden Algorithmen in anderen Implementationen dienen kann.

Da einige Wörter häufiger im Sprachgebrauch vorkommen als andere Wörter, bietet es sich an, an verschiedenen Stellen der Analyse von Korpora nur auf die Begriffe zu schauen. Die Menge der Begriffe setzt sich dabei aus der Menge der Wörter zusammen, wobei jedes Element der Menge Wörter genau ein einziges Mal in der Menge der Begriffe vorkommt. Die Liste {Auto, Auto, Hund, Katze, Haus, Haus, Haus} setzt sich daher aus 7 Wörtern und den 4 Begriffen {Auto, Hund, Katze, Haus} zusammen.

[55] Vgl. Blei et al., Latent Dirichlet Allocation, S. 995.

[56] Blei, David M., blei-lab/lda-c. Erschienen am 09.06.2016, online verfügbar unter: https://github.com/blei-lab/lda-c. Zuletzt geprüft am: 14.07.2019; Wang, Chong, blei-lab/hdp, 2010. Erschienen am 21.02.2017, online verfügbar unter: https://github.com/blei-lab/hdp. Zuletzt geprüft am: 14.07.2019.

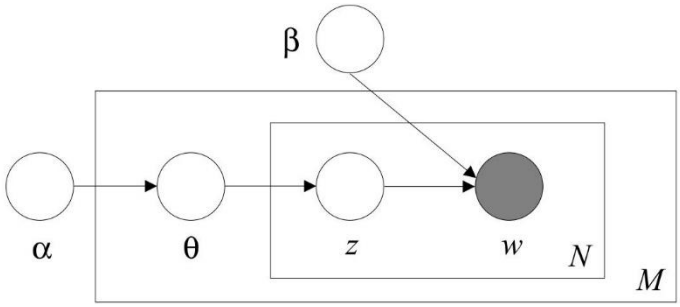

Abbildung 1: LDA Modell[57]

Die Abbildung 1 illustriert das grundlegende Konzept des LDA Algorithmus. Die Rechtecke sollen hierbei Teile unterschiedlicher Mengen darstellen. Das innere Rechteck steht für ein einzelnes Wort, dass sich mit N anderen Wörtern in einem von insgesamt M Dokumenten (äußeres Rechteck) eines Korpus befindet. Der graue Punkt bedeutet, dass Nutzer und Anwendung nur die Wörter in den Dokumenten sehen können, die Themen z und alle anderen Parameter dagegen versteckt sind (hidden variables). α und β sind Korpusparameter und werden wie die Anzahl an Themen im Korpus als gegeben vorausgesetzt.[58]

β ist eine Matrix, in der die Wahrscheinlichkeiten aller im Korpus vorkommenden Begriffe für jedes der Themen gespeichert wird. Der Themengestaltungsparameter θ hingegen setzt sich aus zufällig gezogenen Werten zusammen, die aus einer Dirichlet Verteilung Dir(α) stammen. Das α steuert bei dieser Art der Verteilung, wie wahrscheinlich ein beobachteter Verteilungstyp ist. Kleine α's befürworten Verteilungen mit einzelnen, extremen Werten, große α's hingegen sprechen für sehr gleichmäßige Verteilungen.[59] Anhand dieses α wird in θ abgelegt, wie wahrscheinlich die Zuordnung eines Dokuments zu einer Verteilung verglichen mit anderen Verteilungen im Korpus ist. z wiederum gibt daraus folgend die Wahrscheinlichkeit an, dass ein Wort w, wenn es in einem bestimmten Dokument ist, zu einem der verschiedenen Themen gehört. Bei einer Anzahl von k Themen und M Dokumenten im Korpus hat der Vektor θ eine Dimension von k und die Matrix z eine Dimension von k x M. Die Dimension von β ist bei einer Anzahl von V Begriffen und k Themen im Korpus k x V[60]

Modellberechnung

Ziel der LDA Anwendung ist es, beliebige Dokumente und Worte eines Korpus einer festen Anzahl von Themen zuordnen zu können. Dazu wird zunächst auf Grundlage eines Trainingskorpus, bei dem es sich i.d.R. um eine Stichprobe aus dem Gesamtkorpus handelt, allein anhand von Worthäufigkeiten und

[57] Blei et al., Latent Dirichlet Allocation, S. 997.
[58] Vgl. ebenda, S. 997.
[59] Bol'shev, L. N., Dirichlet distribution. Erschienen am 07.02.2011, online verfügbar unter: https://www.encyclopediaofmath.org/index.php/Dirichlet_distribution. Zuletzt geprüft am: 13.07.2019; Wikipedia, Dirichlet-Verteilung. Erschienen am 18.03.2018. in: Wikipedia (Hrsg.), online verfügbar unter: https://de.wikipedia.org/w/index.php?oldid=175153858. Zuletzt geprüft am: 13.07.2019.
[60] Vgl. Blei et al., Latent Dirichlet Allocation, S. 996.

der vorgegeben Zahl an Themen ein Modell trainiert, dessen Parameter die beobachteten Verteilungen bestmöglich abbilden kann. Die A-Posteriori Verteilung, d.h. die nach der Ziehung der Stichproben beobachtete Verteilung der versteckten Variablen für ein Dokument des Korpus lautet[61]

$$p(\theta, z|\boldsymbol{w}, \alpha, \beta) = \frac{p(\theta, z, \boldsymbol{w}|\alpha, \beta)}{p(\boldsymbol{w}|\alpha, \beta)}$$

wobei **w** in dieser Formel **ein Dokument** im Gesamtkorpus darstellt! Aus dieser Verteilung lassen sich unter den bisher gemachten Bedingungen jedoch keine Rückschlüsse auf die dahinterliegenden versteckten Variablen ableiten, da für die Bestimmung der Wahrscheinlichkeit eines Dokuments im Modell (Nenner) eine normale Berechnung nicht möglich ist. Daher müssen für die Umsetzung des Modells in die Praxis eine Anpassung und eine Schätzung erfolgen.[62]

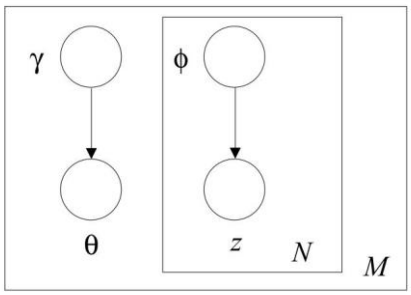

Abbildung 2: Anpassungen LDA Modell[63]

Die Anpassung am Modell wird in Abbildung 2 gezeigt. Die Verbindungen zwischen α, θ und z wurden aufgehoben und stattdessen wurden die zwei Parameter γ und ϕ eingefügt. Diese beiden Parameter sollen dabei für jedes Dokument so bestimmt werden, dass die Divergenz zwischen Modell und beobachteten Wert minimal wird. Zur Bestimmung dieser optimalen Werte schlagen die Autoren eine Abwandlung des EM Algorithmus (Expectations Maximization Algorithmus) vor.

Das Ziel des EM Algorithmus ist es, von einer Menge beobachtbarer Werte auf eine Menge nicht sichtbarer Werte schließen zu können. Die beobachtbaren Werte dieser Menge werden dabei als „unvollständige Daten" bezeichnet und die versteckten Daten als vollständige Daten. Der Algorithmus beruht auf der Annahme, dass der beobachtbare Wert y und der versteckte Wert x über einen Parameter ϕ miteinander verknüpft sind. Die iterative Bestimmung eines optimalen Wertes für ϕ ist dan das Ziel des EM-Algorithmus. Die einzelne Iteration besteht hierbei immer aus dem E-Schritt (E-Step) und dem M-Schritt (M-Step), woher sich auch der Name des Algorithmus herleitet. Das E bedeutet Erwartung bzw. expectation und das M Maximierung bzw. maximization.[64] Es wird also zunächst für den versteckten Wert der Erwartungswert in Abhängigkeit zum Parameter ϕ gebildet und dann auf Grundlage dieses Erwartungswertes ein Optimum für den versteckten Wert. Der Erwartungswert berechnet

[61] Vgl. ebenda, S. 1003.
[62] Vgl. ebenda, S. 1003.
[63] ebenda, S. 1003.
[64] Vgl. Arthur P. Dempster, Nan M. Laird und Donald B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, in: Journal of the Royal Statistical Society: Series B (Methodological) 39, 1977, Nr: 1, S. 1–22, hier S. 1, online verfügbar unter: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

sich dabei nach dem zugrundeliegenden statistischen Verteilungsmodell. Unter einem Optimum wird die Maximierung der Wahrscheinlichkeit für die gemachte Beobachtung verstanden.

Die von Blei et al. vorgeschlagene Abwandlung sieht vor, dass die optimalen Werte für ϕ und γ im E-Step über das Verfahren

(1) initialize $\phi_{ni}^0 := 1/k$ for all i and n
(2) initialize $\gamma_i := \alpha_i + N/k$ for all i
(3) **repeat**
(4) **for** $n = 1$ **to** N
(5) **for** $i = 1$ **to** k
(6) $\phi_{ni}^{t+1} := \beta_{iw_n} \exp(\Psi(\gamma_i^t))$
(7) normalize ϕ_n^{t+1} to sum to 1.
(8) $\gamma^{t+1} := \alpha + \sum_{n=1}^{N} \phi_n^{t+1}$
(9) **until** convergence

Abbildung 3: Pseudocode zur Ermittlung von ϕ und γ[65]

bestimmt werden, bevor im M-Step nach optimierten Werten für α und β gesucht wird.[66] β wird dabei über die Formel

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni}^* w_{dn}^j$$

berechnet. α dagegen wird über die Newton-Raphson Methode berechnet.[67]

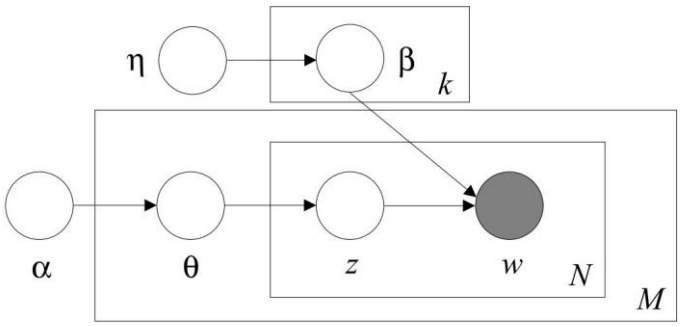

Abbildung 4: Finales LDA Modell[68]

Die LDA in der bisherigen Form setzt voraus, dass neben den bei der Modellerstellung berücksichtigten Begriffen keine weiteren Begriffe existieren. Würde man versuchen, Dokumente mit neuen Begriffen einem Thema zuordnen zu wollen, würde dies die LDA nicht zulassen.[69] Da diese Annahme in der Praxis nicht vertretbar ist, wurde das Modell um den Smoothing Parameter η erweitert, damit eine Optimierung auch bei unbekannten Begriffen geschehen kann. Das finale, geglättete Modell kann man

[65] Blei et al., Latent Dirichlet Allocation, S. 1005.
[66] Vgl. ebenda, S. 1005.
[67] Vgl. ebenda, S. 1006.
[68] ebenda, S. 1006.
[69] Vgl. ebenda, S. 1006.

Abbildung 4 entnehmen. β ist nun abhängig von η und für jedes Thema individuell zu bestimmen. Praktisch wird nur die Formel zur Optimierung von β um eine Addition mit η erweitert:

$$\beta_{ij} \propto \eta + \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni}^{*} w_{dn}^{j}$$

Die praktische Umsetzung in LDA-C

Die in den folgenden Kapiteln praktischen Anwendungen von LDA verwenden ein von David Blei selber bereitgestelltes Programm. Das Programm wurde in der Sprache C geschrieben und der Quellcode inklusive eines kleinen Beispielskorpus ist im GitHub Projekt von David Blei zu finden.[70] Auch wenn Blei in späteren Aufsätzen lieber auf die R-Implementierung von LDA verweist,[71] wird für diese Arbeit die wahrscheinlich ältere C Umsetzung bevorzugt. Dies liegt vor allem daran, dass der Einsatz an unterschiedlichen Programmiersprachen auf diese Art reduziert werden kann. Nach der erfolgreichen Kompilierung einer C Anwendung ist diese nicht mehr auf weitere Anwendungen und sich ändernde Standards angewiesen. Lediglich das Ende einer Betriebssystemarchitektur könnte hier zu Problemen führen. Da es sich durch die in den Programmen genutzten Bibliotheken aber um betriebssystemunabhängige Anwendungen handelt, ist diese Gefahr recht gering, da mit wenig Aufwand die Anwendung auf andere Systeme übertragbar ist.[72] Ein anderer Vorteil der Verwendung einer Hochsprache ist die höhere Geschwindigkeit bei datenintensiven und rechenintensiven Anwendungen.[73]

Von der Verwendung der im Projektordner mitgelieferten topics.py zum Anzeigen der Topwörter der einzelnen Themen wird aus mehreren Gründen abgeraten. Zum einen verwendet das Skript noch den im Jahr 2019 auslaufenden Standard Python2 und zum anderen erfolgt die Ausgabe der Themen direkt im Ausgabestream des Anwendungsfenster, der dann erst in eine Datei umgeleitet werden muss. Da die Themen in diesem Stream untereinander angeordnet sind, können die Ergebnisse nicht in andere Programme importiert werden. Im GitRepository zu dieser Arbeit[74] befindet sich eine überarbeitete Version des Skripts, das in den aktuellen Python3 Standard umgeschrieben wurde, die Ergebnisse in eine angegebene Datei schreibt und ein Ausgabeformat bereitstellt, dass einen direkten Import in anderen Anwendungen ermöglicht. Noch mehr wird allerdings die Verwendung der Topwortlisten Funktion von TopModHis empfohlen, da diese Funktion bei großen Datenmengen performanter als das Pythonskript ist und keine funktionierende Python Umgebung voraussetzt.

Der Projektordner von LDA-C beinhaltet acht Header-Dateien. Bis auf die Datei „lda.h“ beinhalten diese Dateien aber nur die Definitionen der zur Berechnung erforderlichen Funktionen, geordnet nach dem jeweiligen Einsatzzweck. Funktionen zum Einlesen der Dokumente befinden sich also in der Datei „lda-data.h“, Funktionen zur Alpha-Optimierung in der Datei „lda-alpha.h“. Besonderheiten bilden die Dateien

[70] Blei, blei-lab/lda-c.
[71] Vgl. Blei, Probabilisitc Topic Models, S. 82.
[72] Die einfachste Art ist hier natürlich die Kompilierung des Quellcodes auf dem anderen Zielsystem.
[73] Vgl. Mikolov et al., Efficient Estimation of Word Representations in Vector Space, S. 11.
[74] Müller, Florian, GitRepository TopModHis, online verfügbar unter: https://github.com/ImdWf/TopModHis. Zuletzt geprüft am: 14.08.2019.

„cokus.h“, in der die Bytegröße von α festgelegt wird und „utils.h“, in der u.a. die Ableitungsfunktionen der Gamma-Funktion hinterlegt sind.

Von Interesse sind die Definitionen der Strukturen, da diese Aufschluss über die Umsetzung des theoretischen Modells in die Praxis geben. Blei definiert vier Strukturen:

1. Document
2. Corpus
3. lda_model
4. lda_suffstats (Suffizienz der LDA, d.h. die Umwandlung der gezogenen Stichproben in das Verteilungsmodell)

Die Dokumente und das Korpus sind dabei naheliegend aufgebaut. Ein Dokument besteht aus verschiedenen Wörtern (technisch korrekt wäre die Bezeichnung Begriffe/ terms, aber es wird der Name Words verwendet), die jeweils verschieden oft in diesem Dokument vorkommen (counts). Das Dokument enthält eine bestimmte Anzahl an Begriffen (length) und eine bestimmte Menge an Wörtern insgesamt (total). Das Korpus besteht nun aus beliebig vielen Dokumenten und hat eine bestimmte Anzahl an Begriffen (num_terms) und eine bestimmte Zahl an Dokumenten (num_docs).

Die Anzahl an Begriffen wird auch im LDA Model hinterlegt, ferner wird hier der Ausgangswert der Variable Alpha, die Anzahl der Themen und die logarithmierte Wahrscheinlichkeit der Wörter (log_prob_w) hinterlegt. In Kombination mit den beiden hinterlegten Werten für die Zahl der Themen und die Zahl an Begriffen entspricht log_prob_w dem β aus dem theoretischen Modell.

Die LDA_suffstats bestehen aus einem zweidimensionalen Array zur Klassifikation der Wörter (class_word), einem eindimensionalen Array der absoluten Klassifikation (class_total), einem anderen Alpha-Wert und der Anzahl der Dokumente. Wieder lassen sich anhand des gespeicherten Wertes, dieses Mal für die Zahl der Dokumente, Rückschlüsse auf die Verknüpfung mit dem theoretischen Modell machen. Da class_total von der Zahl der Dokumente abhängt, handelt es sich hierbei um θ, während class_word z darstellt.

Anwendungsstart

Die für C und C++ Anwendungen typische Datei main.[c/cpp], in der der Startpunkt, die Funktion main(), enthalten ist, findet man im LDA-C Projekt nicht. Stattdessen befindet sich die Hauptfunktion in der Datei „lda-estimate.c“. In der Hauptfunktion wird zunächst ein leerer Pointer auf ein Korpus Objekt initialisiert und für einen eventuell benötigten zufälligen Einstiegspunkt die aktuelle Zeit in eine Zahl umgewandelt. Im Anschluss wird überprüft, ob der Funktion mit dem Aufruf der Anwendungen weitere Parameter (Argumente) übergeben wurden. Falls dies der Fall ist (argc >1), wird bestimmt, ob mit diesem Aufruf ein Modell trainiert („est“ ≙ estimate) oder ein bestehendes Modell getestet werden soll („inf“ ≙ inference). Je nach Art des Aufrufs versucht die Anwendung nun die jeweils notwendigen Informationen aus den übergegebenen Parametern zu gewinnen. Es ist zu betonen, dass es hierbei keinerlei Fehlerbehandlung gibt. Wurden z.B. zu wenige Parameter übergeben, so bricht das Programm einfach die Durchführung ab. Lediglich, wenn keine Parameter übergeben werden, erscheint ein

Zweizeiler, indem die Reihenfolge der zu übergebenden Parameter für die beiden Arten der Anwendung aufgelistet wird.

```
$ ../lda-c/lda.exe
usage : lda est [initial alpha] [k] [settings] [data] [random/seeded/manual=filename/*] [directory]
        lda inf [settings] [model] [data] [name]
```

Abbildung 5 "Hilfe" von LDA-C

Bei beiden Arten der Anwendung muss eine Datei mit Einstellungen übergeben werden. Stellt man die beiden mitgelieferten Vorschläge „settings.txt“ und „inf-settings.txt“ gegenüber (Abbildung 6) so fällt auf, dass sie sich lediglich in der Anzahl an maximal vorgeschlagenen Iterationen bei der Bestimmung der optimalen Varianz unterscheiden. Während bei dem Modelltraining („est“) maximal 20 Iterationen vorgesehen sind, sind es bei den Modelltests („inf“) notfalls unendlich Iterationen.

Interessanter in der Datei ist allerdings der letzte Punkt. Mit „alpha estimate“ wird der Anwendung mitgeteilt, ob der Startwert von α im Sinne einer besseren Konvergenz angepasst werden soll oder ob der Startwert über alle Iterationen hinweg konstant gehalten werden soll. Da, wie noch weiter unten gezeigt werden soll, die Bestimmung eines optimalen α eine zeitintensive Aufgabe ist und es oftmals nicht das eine optimale α gibt, bietet es sich gerade bei größeren Datensätzen an, nach der Bestimmung eines optimalen α die Option in „alpha fixed“ umzuändern.

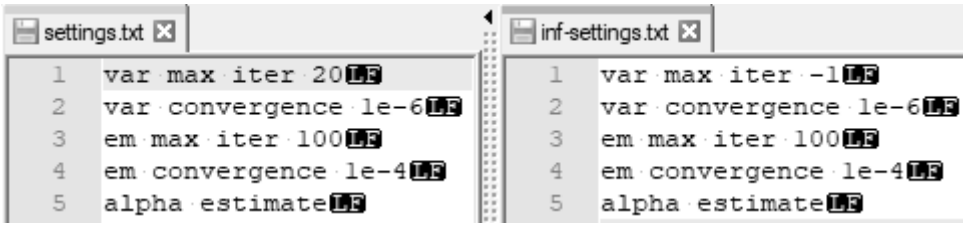

Abbildung 6 Settings im Vergleich

Die weiteren an die Anwendung zu übergebenden Parameter bedürfen keiner tieferen Betrachtung, da es sich um Pfade zu Dateien oder dem Zielordner für die Ergebnisse handelt. Das kleine k steht für die Anzahl an Topics, für die das Modell trainiert werden soll. Lediglich auf die unterschiedlichen Arten der Initialisierung des Modells im Schätz (Est)- Modus muss tiefer eingegangen werden. LDA verfügt über vier unterschiedliche Arten zur Initialisierung von θ und z. Die dazugehörigen Funktionen befinden sich in der Datei „lda-model.c“:

- seeded ≙ corpus_initialize_ss
- random ≙ random_initialize_ss
- manual=[filename] ≙ manual_initialize_ss
- * / beliebige Zeichenkette ≙ keine Spezielle initialisierung, alle Werte werden mit 0 initialisiert

Der Modus random füllt θ und z Thema für Thema mit zufälligen Werten und lässt sich schlecht nachverfolgen, da diese Startwerte sich nicht wieder bei späteren Durchläufen laden lassen. Dagegen handelt es sich bei der Option „seeded“ um einen gut nachvollziehbaren Initialisierungsmodus. Jedes Thema wird mit einem zufällig ausgewählten Dokument initialisiert. Alle ausgewählten Dokumente

werden in der Kommandoausgabe ausgegeben, wodurch man bei späteren Versuchen die gleiche Dokumentenkombination im Modus manual= verwenden kann.

```
$ ./lda-c/lda.exe est 0.1 10 lda-c/settings.txt Daten/Ergebnis_1969.txt seeded example_lda
reading data from Daten/Ergebnis_1969.txt
number of docs    : 5785
number of terms   : 96388
initialized with document 5592
initialized with document 2368
initialized with document 5142
initialized with document 1033
initialized with document 2142
initialized with document 1964
initialized with document 1538
initialized with document 3023
initialized with document 4859
initialized with document 1907
**** em iteration 1 ****
document 0
document 1000
document 2000
document 3000
document 4000
document 5000
alpha maximization : -40490060.59709   -410735.12450
alpha maximization : -14545875.95982   -410285.21799
alpha maximization : -5018217.85274   -409051.01085
alpha maximization : -1530230.71280   -405615.45507
alpha maximization : -265058.10335   -395744.18927
alpha maximization : 180370.14775   -366455.15152
alpha maximization : 322206.24104   -287106.44073
alpha maximization : 355677.31801   -142660.62028
alpha maximization : 359445.88690   -26020.78743
alpha maximization : 359529.11981   -703.28381
alpha maximization : 359529.17610   -0.49241
alpha maximization : 359529.17610   -0.00000
new alpha = 0.10546
```

Abbildung 7 Beispiel eines gültigen Starts von LDA

Für die manuelle Initialisierung der Themen wird eine separat erstellte Datei benötigt. Der Inhalt der Datei enthält lediglich die einzelnen Startdokumente Zeile für Zeile aufgelistet und könnte wie folgt aussehen:

22677

22628

11122

1376

21323

5610

Es ist unbedingt zu beachten, dass der Dateiname ohne ein trennende Leerzeichen an die Anweisung „manual=“ gehängt werden muss. Ein gültiger Aufruf wäre also manual=start\ seeds.txt. Man beachte die Kombination \[Leerzeichen], die notwendig ist, da ein Leerzeichen im Dateinamen ist. Andernfalls würde das Programm nur nach der Datei „start“ suchen. Wie viele Dokumente pro Thema zur Initialisierung genutzt werden, kann leider nicht über die settings.txt gesteuert werden, sondern muss in der Header – Datei „lda-model.h“ geändert werden. Die Konstante NUM_INIT ist dort standardmäßig

auf 1 gestellt. Nach einer Änderung dieses Wertes muss die Anwendung neu kompiliert werden, damit die Änderung wirksam wird.

Dauer

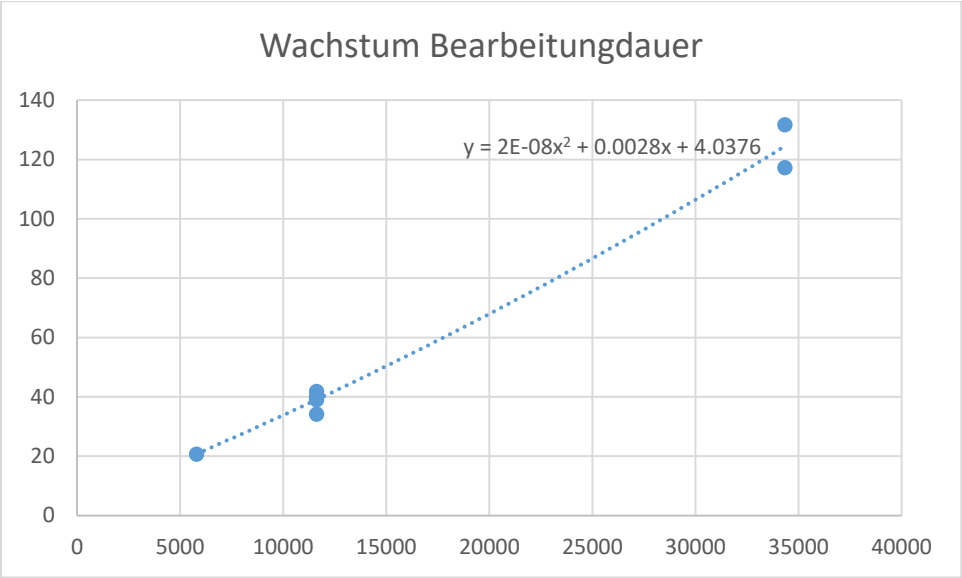

Abbildung 8 Bearbeitungsdauer LDA Zahl Dokumente/ Bearbeitungszeit in Minuten

Die Dauer der Ausführung ist schwach quadratisch, mehr linear von der zur Kalkulation genutzten Zahl an Dokumenten abhängig (Abbildung 8). Jedoch kann die Bearbeitungszeit selbst bei gleichbleibender Zahl von Dokumenten erheblich in Abhängigkeit von der Wahl des Startwertes für α schwanken. Dauerte die Berechnung des Modells auf dem System des Autors[75] für einen Jahrgang (ca. 5785 Dokumente) mit α Optimierung und einen Startwert von 0,1 für α im Durchschnitt 60 min, verkürzte sich die Berechnungszeit auf ca. 21 min bei der Wahl eines näher an einem Optimum liegenden Startwertes. Bei linearer Zunahme der Bearbeitungsdauer je Dokument benötigt die Berechnung eines Modells für den Startwert 0,1 genauso lange wie die Berechnung mit der dreifachen Zahl an Dokumenten für ein optimiertes α, in diesem Fall 0,01277. Es empfiehlt sich also, mit einer kleinen Stichprobe die Größe von α zu bestimmen, bevor man die großen Stichproben für das Modelltraining verwendet.

```
$ ./lda-c/lda.exe inf lda-c/inf-settings.txt example_lda/010 Daten/Ergebnis_1969.txt example_lda_inf
reading data from Daten/Ergebnis_1969.txt
number of docs    : 5785
number of terms   : 96388
loading example_lda/010.other
loading example_lda/010.beta
document 100
document 200
document 300
document 400
document 500
document 600
document 700
document 800
document 900
document 1000
```

Abbildung 9 Beispiel gültiger Aufruf LDA Inferenz

[75] 1TB HDD mit 7200 RpM; 12 GB DD3 RAM 600 MHZ Taktung, Intel i7 980 Extreme Edition

Ergebnisse

Den Abläufen der theoretischen Vorbetrachtung folgend erstellt LDA-C unterschiedliche Ergebnisdateien, die im folgenden Abschnitt analysiert werden. Zunächst werden die Ergebnisse des Estimate Modus betrachtet und im Anschluss die Ergebnisse des Inference Modus.

Die wesentlichen drei Ergebnisdateien im Estimate Modus sind die drei „final" Dateien mit den Endungen *beta*, *gamma* und *other*. In der *gamma* Datei sind wie in Abbildung 2 illustriert die verschiedenen Verteilungsparameter der Themen pro Dokument abgelegt. Die Verteilung der Wörter pro Thema, d.h. β, in Form der logarithmischen Wahrscheinlichkeit der Wörter liegt dagegen in der Datei mit der Endung *beta*. Diese wird z.B. dazu benötigt, um die Topwörter pro Thema anzeigen zu können. Zu betonen ist, dass, da es sich um Werte zwischen 0 und 1 handelt, der entsprechende Logarithmus negativ ist. Je näher die Werte gegen 0 gehen, desto „kleiner" werden sie. Werte von -500 sprechen dabei für eine geringere Zugehörigkeit als -1. Deshalb ist auf eine absteigende Sortierung zu achten.

In der final Datei mit der Endung *other* sind Angaben wie die Zahl an Begriffen, an Themen und der Endwert von α notiert. Die Datei „word-assignment.dat" enthält Dokument für Dokument, welchem Thema welcher Begriff im Dokument zugeordnet wurde. Die Zeile

082 4395:03 21552:05 25502:05 29756:05 […]

müsste hierbei folgendermaßen gelesen werden: Das Dokument hat 82 Begriffe insgesamt, Begriff 4359 wurde Thema 3 zugeordnet, Begriff 21552 dagegen wie auch die folgenden Begriffe Thema 5. Es ist zu beachten, dass Thema 5 bei sechs Themen das höchste Thema ist, da bei 0 angefangen wird zu zählen.

Verwendet man den Inferenzen Modus, so erstellt LDA-C zwei Dateien mit der Endung „.dat". Einfach zu interpretieren ist die Datei mit dem Suffix *–gamma.dat.* Jede Zeile in der Datei stellt ein Dokument dar. Pro Zeile werden für jedes Thema die Wertungen abgebildet.

15.61243409 2.57731371 6.59596291 0.01276999 32.22616204 54.05197723

Bei sechs Themen ist das hier beispielhaft abgedruckte Dokument sehr stark dem sechsten Thema zuzuordnen, aber auch das fünfte Thema ist sehr stark vertreten. Dass es sich bei dem Dokument dagegen um ein Element des vierten Themas handeln könnte, kann als unwahrscheinlich angesehen werden.

In der Datei mit dem Suffix *–lda-lhood* werden die maximierten Likelihoods für jedes Dokument am Ende der Optimierung abgelegt. Ohne weitere Informationen zu einem Dokument ist die Aussagekraft dieser Werte leider recht gering. Große negative Werte sprechen i.d.R. für ein großes und langes Dokument, während kleine Werte eher für kurze Dokumente sprechen. Dies erklärt sich daraus, dass der Likelihood Wert eines Dokumentes u.a. durch ein Aufsummieren der Likelihood Werte aller im Dokument vorkommenden Begriffe gebildet wird. Um die verschiedenen Werte in ein vergleichbares Verhältnis setzten zu können, muss der jeweilige Likelihood Werte durch die Zahl an Wörtern pro Dokument geteilt werden. Diese Art von Likelihoods wird im weiteren Verlauf der Arbeit als durchschnittliche Likelihoods bezeichnet.

Da LDA-C mit Zufallsvariablen arbeitet, ist es zu befürchten, dass Ergebnisse nicht reproduzierbar sind und von anderen Forschern bei Benutzung der gleichen Daten nicht nachvollzogen werden können. Diese Befürchtung ist jedoch unbegründet, da LDA-C mit pseudozufälligen Werten arbeitet. Ausgehend von einem zeitabhängigen Startwert folgen die gezogenen Werte immer dem gleichen Muster. Zieht man also beliebig oft einhundert Werte von dem gleichen Startpunkt aus, so erscheinen diese einhundert Werte zwar zufällig verteilt, sind aber in jedem Fall für jeden Versuch identisch. Für LDA-C bedeutet dies, dass, wenn im Modus *seeded* alle Startdokumente notiert wurden, zu einem späteren Zeitpunkt durch das manuelle Aufrufen der Startpunkte (Modus *manual*) das gleiche Modell noch einmal erstellt wird, sofern auch die anderen Bedingungen wie die Wahl des α's gleich bleiben.

Der hierarchische Dirichlet Prozess (HDP) und die hierarchische Dirichlet Allokation (hLDA)

Der bisher betrachtete LDA Algorithmus hat den Nachteil, dass die Zahl der Themen vorgegeben werden muss. Möchte man einen Korpus jedoch in seiner Tiefe erkunden, muss diese Beschränkung aufgehoben werden.

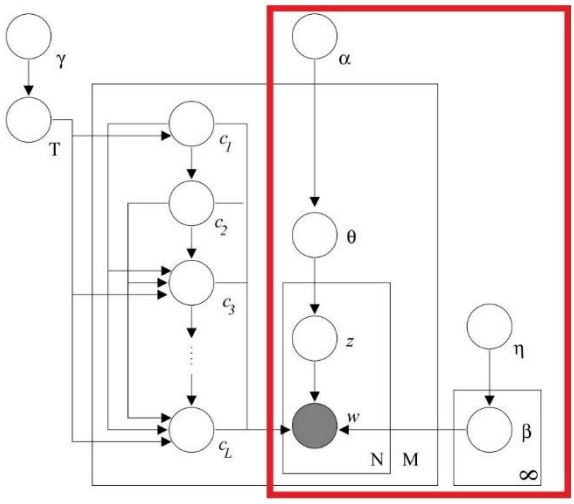

Abbildung 10: Aufbau hLDA[76]

Abbildung 10 zeigt die für diesen Zweck von Blei et al. vorgeschlagenen Modifikationen am LDA – Modell. Mit dem Rahmen wurde dabei der ursprüngliche Teil des geglätteten LDA- Modells markiert, um die Erweiterung besser sichtbar zu machen. Das ursprüngliche LDA Modell bleibt also bis auf eine unscheinbare Veränderung unangetastet und wird nur durch einen neuen Mechanismus unterstützt. Die Veränderung innerhalb des Modells trägt dabei der gewünschten Auflösung der Themenrestriktion Rechnung, weshalb die Dimension von β nun unendlich groß werden kann.

[76] David M. Blei und Michael I. Jordan et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, Proceedings of the 16th International Conference on Neural Information Processing Systems, S. 17–24, hier S. 21.

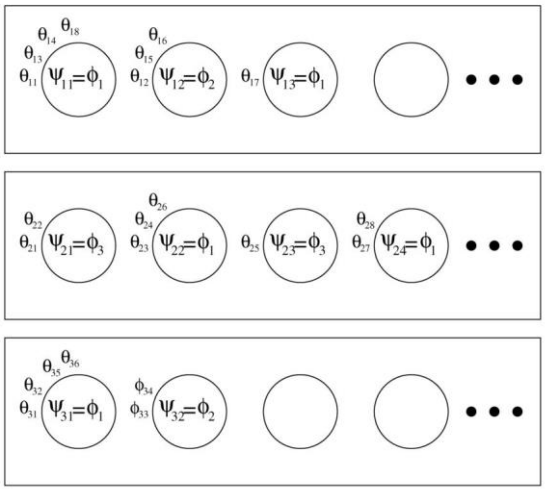

Abbildung 11: Platzverteilung in den unterschiedlichen Restaurants[77]

Die Erweiterung des hLDA Modells basiert auf der Idee des verschachtelten Chinese Restaurant Prozesses (hCRP). Der Name wurde von den chinesischen Restaurants in San Francisco Anfang der achtziger Jahre des letzten Jahrhunderts inspiriert, welche eine unendliche Zahl an Sitzmöglichkeiten zu haben schienen.[78] Der Prozess ist weit verbreitet in der nichtparametrischen Statistik, da er es möglich macht, statistische Modelle zu definieren, in denen die Beobachtungen aus einer unbekannten Zahl von Grundmengen stammen.[79] Nichtparametrisch bedeutet hierbei, dass die Zahl der Cluster offen ist.[80] Zunächst wird angenommen, dass es ein Franchise an chinesischen Restaurants gibt, die quer über die Stadt verteilt sind und die gleiche Speisekarte anbieten. Der erste Kunde (θ), der sich in einem der Restaurants an einen freien Tisch (ψ) setzt, bestellt sich ein Gericht (ϕ). Dabei dürfen verschiedene Tische in den unterschiedlichen Restaurants das gleiche Gericht essen.[81] Alle weiteren Kunden, die nun das Restaurant betreten, haben entweder die Wahl, sich an einen der bereits besetzten Tische zu setzen oder einen neuen Tisch zu besetzen, abhängig von einem Wahrscheinlichkeitswert.[82] Eine mögliche Verteilung von Gästen und Gerichten zeigt Abbildung 11. Um die Restaurants mit den unterschiedlich besetzten Tischen nun miteinander zu verbinden, wird die Geschichte um Touristen erweitert, die für eine kulinarische Entdeckungsreise in die Stadt kommen. Am ersten Abend betreten diese Touristen das Hauptrestaurant der Kette und entscheiden sich nach dem obigen Muster für einen Tisch. An diesem Tisch erhalten sie dann nicht nur ihr Gericht, sondern erfahren das Restaurant, an dem sie den nächsten Abend essen werden. Jedem Tisch im Hauptrestaurant ist dabei genau ein

[77] Yee Whye Teh und Michael I. Jordan et al., Hierarchical Dirichlet Processes, in: Journal of the American Statistical Association 101, 2006, Nr: 476, S. 1566–1581, hier S. 1571. online verfügbar unter: doi.org/10.1198/016214506000000302, online verfügbar unter: http://www.cs.columbia.edu/~blei/papers/TehJordanBealBlei2006.pdf. Zuletzt geprüft am: 22.06.2019.

[78] Vgl. Blei et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, S. 18.

[79] Vgl. David M. Blei, Thomas L. Griffiths und Michael I. Jordan, The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, in: Journal of the ACM 57, 2010, Nr: 2, S. 1–30, hier S. 8. online verfügbar unter: doi.org/10.1145/1667053.1667056.

[80] Vgl. Yee W. Teh und Michael I. Jordan et al., Sharing clusters among related groups: Hierarchical Dirichlet processes, Advances in neural information processing systems, Cambridge, London, 2003, S. 1385–1392, hier S. 1385–1386.

[81] Vgl. Teh et al., Hierarchical Dirichlet Processes, S. 1571.

[82] Vgl. Teh et al., Sharing clusters among related groups: Hierarchical Dirichlet processes, S. 1388.

folgendes Restaurant zugeordnet. Nach x Tagen Urlaub hat also jeder der Touristen in x verschiedenen Restaurants gegessen.[83] Abbildung 12 zeigt dieses Szenario für fünf Touristen und drei Tagen Urlaub.

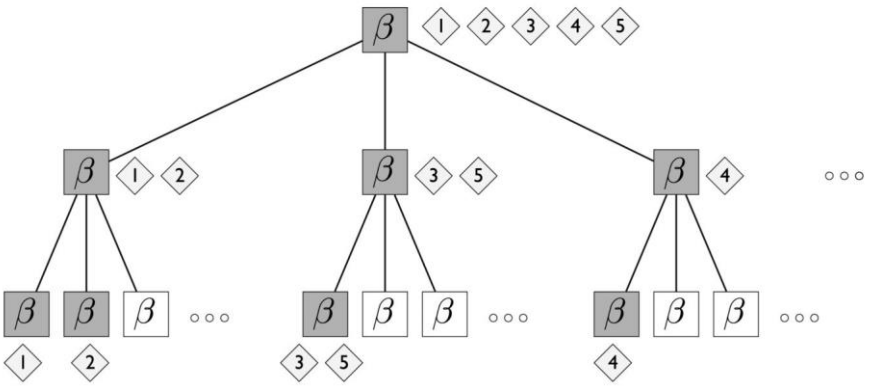

Abbildung 12: Verteilung von fünf Touristen über drei Tage[84]

Modellberechnung

Zu den bisher versteckten Dokumentenparameter θ, α und η kommen in diesem Modell (Abbildung 10) nun L versteckte Paramater c_l hinzu. Diese c's stehen für die einzelnen Ebenen, denen ein Dokument zugeordnet werden kann und drücken die Hierarchie aus, d.h. die c's bauen mit fortlaufendem Index aufeinander auf. Jedes der c's wird mit einem Topic nach der Wahrscheinlichkeit $p(\beta|\eta)$ assoziiert. Das T steht für die möglichen Pfade, die durch den Algorithmus erstellt werden können. Da dieses aber nicht sichtbar ist, müssen die Informationen über die Rückverfolgung der Pfade von den einzelnen c's gewonnen werden.[85]

Die zugrundeliegende Strategie hinter dem Schaubild erklärt sich dabei wie folgt:[86]

1. Wir haben ein Restaurant c_1 als Startpunkt
2. Jeder neue Punkt auf unserem Pfad wird mit einem Thema aus dem vorhergehenden Punkt über die Gleichungen:

   $$p(Nehme\ ein\ bereits\ ausgewählt es\ Thema\ |bereits\ betrachtete\ Dokumente) = \frac{m_i}{\gamma + m - 1}$$

   $$p(Wähle\ ein\ neues\ Thema\ |bereits\ betrachtete\ Dokumente) = \frac{\gamma}{\gamma + m - 1}$$

   bestimmt. Dieser Schritt wird solange wiederholt, bis die gewünschte Dimension L erreicht wurde. Die Dimension kann dabei vorgegeben werden oder durch Verfahren wie das Stick-Breaking Verfahren erzeugt werden. Auf dieses Verfahren wird im Anschluss noch genauer eingegangen.

[83] Vgl. Blei et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, S. 19.
[84] Blei et al., The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, S. 9.
[85] Vgl. Blei et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, S. 20–21.
[86] Vgl. ebenda, S. 20.

3. Über die Dirichletverteilung Dir(α) wird für jeden Punkt c_l nun eine Themenproportion gezogen und im Vektor θ hinterlegt
4. Für jedes Wort ziehe einen Punkt auf dem Pfad aus dem multinominalverteilten θ. Ziehe im Anschluss ein Wort aus dem Thema, dass dem Punkt zugewiesen wurde.

Um die Tiefe des Baumes nicht unendlich groß werden zu lassen, wird meistens zur Ermittlung der Tiefe das sogenannte Stick-Breaking Verfahren vorgeschlagen. Hierbei werden auf Grundlage einer Beta-Verteilung Stücke gebildet, die einen Wert zwischen 0 und 1 annehmen. Jedes Stück vermindert die mögliche Größe des nachfolgenden Stückes. Wenn die Summe aller gebildeten Stücke schließlich den Wert 1 erreichen, können keine weiteren Stücke mehr gebildet werden. Dies stellt die Abbruch-bedingung des Prozesses dar. Die Variable, in der die Größe der Stücke gespeichert wird, heißt in der Regel Pi.[87]

Für die eigentliche Ermittlung der Daten schlagen die Autoren nun ein zweiteiliges Gibbs Sampling vor.[88] Zunächst werden die Werte für die Zuordnung von Themen und Wörtern aus der bekannten LDA Verteilung über eine Markov Kette bestimmt. Dabei wird ausgehend von einer zufälligen Initialisierung für jede Iteration die Werte für die Themen z neu bestimmt. Ziel ist es, dass die Werte der so ermittelten Verteilung mit jeder Iteration näher an die beobachtete Verteilung kommen.[89] Im zweiten Teil des gleichen Iterationsschrittes werden die Werte des Chinese Restaurant Process (CRP) für jedes Dokument blockweise bestimmt. Dieser Vorgang wird solange durchgeführt, bis die A-Priori Verteilung stark genug der A-Posteriori Verteilung gleicht.[90]

Praktische Umsetzung

In dieser Arbeit wird als praktische Anwendung die von Chong Wang in C++ geschriebene Umsetzung HDP-faster verwendet. Diese ist, wie bereits LDA-C, im GitRepository der Bleilabs zu finden. Da es sich um eine C++ Umsetzung handelt, ist der Ansatz deutlich objektorientierter, als es C zulässt. Dies bedeutet aber auch, dass nun nicht mehr die Strukturen in einer einzigen Datei gebündelt definiert werden, sondern, dass die Klassen in vier unterschiedlichen Header Dateien definiert werden. Da in den beiden Dateien „stirln.h“ und „utils.h“ keine Klassen definiert werden, ist eine genauere Betrachtung nicht notwendig und der Fokus verdichtet sich auf die beiden Header-Dateien „corpus.h“ und „state.h“.

Wie es der Name schon vermuten lässt, werden in der Datei „corpus.h“ alle für das Korpus notwendigen Klassen definiert. Dies ist neben der Klasse Corpus noch die Klasse Document. Ein Dokument besitzt die Eigenschaften:

- Id
- Words
- Counts
- Length
- Total

[87] Vgl. Teh et al., Hierarchical Dirichlet Processes, S. 1568.
[88] Vgl. Blei et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, S. 19.
[89] Vgl. Thomas L. Griffiths und Mark Steyvers, A probabilistic approach to semantic representation, Proceedings of the annual meeting of the cognitive science society, 2002, hier S. 4.
[90] Vgl. Blei et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, S. 21.

Wie bereits bei LDA wäre die Bezeichnung *Terms* anstelle von *Words* genauer, da es sich wieder um die ID der einzelnen Begriffe handelt. In *Counts* wird abgelegt, wie häufig jeder Begriff in dem Dokument vorkommt, *Length* definiert die Gesamtzahl an Begriffen und *Total* die Gesamtzahl an Wörtern. Für die objektorientierte Programmierung in C++ unüblich ist die Tatsache, dass alle Eigenschaften öffentlich sind.[91] Normalerweise werden in C++ die Eigenschaften eines Objektes mit der Eigenschaft privat versehen. Nachvollziehbar ist dieser Schritt in der Hinsicht, dass dies die Masse an Funktionen erheblich reduziert und lediglich die Initialisierung eines Document-Objektes neben den Eigenschaften definiert werden muss. Ein Dokument kann dabei als leere Instanz initialisiert werden oder unter Angabe der Zahl der im Dokument enthaltenen Begriffe. Interessanterweise wird die ID eines Dokuments nicht während der Initialisierung direkt vergeben.

Zum Einsatz kommen die Dokumente in der Klasse *Corpus*, die ebenfalls in der Datei „corpus.h“ definiert wird. Wie bereits bei den Dokumenten sind auch bei der *Corpus* Klasse alle Eigenschaften publik. Abgelegt werden in *Corpus* Informationen über die Zahl der Dokumente, die Größe des Vokabulars (≙ Zahl der Begriffe) und die Gesamtzahl an Wörtern. Um den Bedarf an Arbeitsspeicher gering zu halten, werden von den Dokumenten nur die Zeiger zu den einzelnen Variablen in einem Vektor hinterlegt.

Das Herzstück von HDP ist die Datei „state.h“, da hier die verschiedenen Pfade zusammengeführt werden und alle Klassen definiert werden, die für die folgenden Berechnungen notwendig sind. Auch die Definition von Konstanten erfolgt in dieser Datei. Besonderes Interesse gebührt dabei dem Schwellenwert für neue Themen *TOPIC_THRESHOLD* der Standardmäßig auf 1,95 eingestellt ist. Um eine Veränderung dieses Wertes wirksam werden zu lassen, ist eine komplette Neukompilierung der Anwendung notwendig. Als Anwender würde man sich wünschen, dass man solche Werte in einer Konfigurationsdatei abändern könnte.

Im Vergleich zu LDA-C sind die einzelnen Modellvariablen nicht in Strukturen definiert, sondern es werden die Ergebnisse in verschiedenen Vektoren abgelegt und anschließend in Klassen der verschiedenen Ebenen hinterlegt. Die tiefste Ebene ist dabei die *WordInfo,* in der die ID des Begriffs (*word_*), die Anzahl an Vorkommen dieses Begriffs (*count_*) und die Themenzuordnung des Begriffs (*topic_assignment_*) hinterlegt werden. Die nächsthöhere Ebene bilden die Statistiken der einzelnen Dokumente (*DocState*). Diese speichern dabei kaum andere Information als die bereits in der Klasse *Document* gespeicherten Informationen ID (*doc_id_*), vorkommende Begriffe (*words_*) und Länge des Dokuments (*doc_length_*). Diese Informationen werden hierbei sogar aus den *Document* Objekten des Korpus gewonnen. Die einzige Erklärung für diese auf dem ersten Blick unnötige Dopplung der Informationen versteckt sich im Schlüsselwort *const*, mit der das *Document* Objekt im Konstruktor aufgerufen wird. Dieses Schlüsselwort lässt vermuten, dass es sich bei *DocState* um eine „Arbeitskopie“ handelt, um die Daten im Korpus nicht zu verfälschen, da ein Aufruf mit dem Zusatz *const* explizit einen rein lesenden Aufruf von Variablen verspricht.

Alle Hyperparameter wie η, γ, β und α werden in der Klasse *HDPState* hinterlegt. Auch die bereits gebildeten Bruchstücke π (*pi_*) und die verbleibende Menge (*pi_left_*) werden in dieser Klasse

[91] Vgl. Torsten T. Will, C++: Das umfassende Handbuch, Bonn, 2018, 1. Auflage, S. 273.

gespeichert. Funktionen, die diese Werte verändern könnten, werden in dieser Klasse aber nicht definiert. Sie dient lediglich als Container für diese wichtigen Werte.

Alle Fäden werden schlussendlich in der Klasse *HDP* zusammengeführt. Interessanterweise werden die Statistiken der Dokumente als ein zweidimensionales Array initialisiert, um welche Dimensionen es sich hierbei handelt, lässt sich aus der reinen Definition jedoch leider nicht ableiten. Für die späteren Ergebnisse interessant ist der Vektor *table_counts_by_topic_doc_*, da aus diesem später die einzelnen Ebenen abgeleitet werden.

Anwendungsstart

```
$ ./hdp/hdp-faster/hdp.exe

C++ implementation of Gibbs sampling for hierarchical Dirichlet process, a much faster version.
Authors: Chong Wang, chongw@cs.princeton.edu, Computer Science Department, Princeton University.
usage:
      hdp               [options]
      --help:           print help information.
      --verbose:        print running information.

      control parameters:
      --directory:       the saving directory, required.
      --random_seed:     the random seed, default from the current time.
      --max_iter:        the max number of iterations, default 100 (-1 means infinite).
      --max_time:        the max time allowed (in seconds), default 1800 (-1 means infinite).
      --save_lag:        the saving point, default 5.

      data parameters:
      --train_data:      the training data file/pattern, in lda-c format.

      model parameters:
      --eta:             the topic Dirichlet parameter, default 0.05.
      --gamma:           the first-level concentration parameter in hdp, default 1.0.
      --alpha:           the second-level concentration parameter in hdp, default 1.0.
      --gamma_a:        shape for 1st-level concentration parameter, default 1.0.
      --gamma_b:        scale for 1st-level concentration parameter, default 1.0.
      --alpha_a:        shape for 2nd-level concentration parameter, default 1.0.
      --alpha_b:        scale for 2nd-level concentration parameter, default 1.0.
      --sample_hyper:   sample 1st and 2nd-level concentration parameter, default false

      test only parameters:
      --test_data:       the test data file/pattern, in lda-c format.
      --model_prefix:    the model_prefix.
*****************************************************************************************************
```

Abbildung 13 Hilfe bei Anwendungsstart von HDP-faster

Der Start der HDP Anwendung muss über eine Konsole oder eine BATCH Datei erfolgen, da für den Start der Anwendung eine Reihe von Parametern übergeben werden, die als Argumente an den Funktionsaufruf angehängt werden müssen. Im Vergleich zu LDA-C ist die Dokumentation über die verschiedenen Parameter wesentlich besser, d.h. ruft man hdp.exe einzeln oder mit dem Zusatz *—help* auf, so erscheint eine lange Liste an möglichen Parametern, die der Funktion übergeben werden können, gefolgt von einer kurzen Beschreibung über die Bedeutung der jeweiligen Option (Abbildung 13). Es fehlt jedoch eine Beschreibung der unterschiedlichen Arten des Funktionsaufrufs für das Training und die Tests.

Um ein Training zu starten, benötigt hdp lediglich die Informationen *–directory* und *–train_data*. Wie bereits bei LDA-C müssen die Trainingsdaten (*--train_data*) bereits im vektorisierten LDA-C Format vorliegen. Mit *–directory* wird dem Programm mitgeteilt, in welchem Ordner die Ergebnisse abgelegt werden sollen. Gerade bei Probeläufen zur Bestimmung von geeigneten Parametern eines Modells

empfiehlt es sich, den Zusatz –*verbose* anzuhängen, da dieser Modus dem Nutzer mehr Informationen während des Trainings zur Verfügung stellt.

```
$ ./hdp/hdp-faster/hdp.exe --train_data Daten/Ergebnis_1969.txt --directory example-hdp
**********************************************************************************************************
Working directory: example-hdp.
Setting saved at example-hdp/settings.dat.
Reading training data from Daten/Ergebnis_1969.txt.
Reading data from Daten/Ergebnis_1969.txt.
number of docs   : 5785
number of terms  : 96388
number of total words : 924650
```

Abbildung 14 Erfolgreicher Start des Trainingsmodus in HDP-faster ohne Verbose Option

Neben der Möglichkeit, die Startwerte für η, α und γ vorzugeben, versteckt sich eine andere sehr interessante Funktion in der langen Auflistung. Fügt man im **Trainingsmodus** den Zusatz *--sample_hyper* hinzu, so werden mit dem Training für das Korpus die optimalen Werte von α und γ berechnet. Nur in diesem Fall kommen auch die Optionen *–gamma_a, --gamma_b, alpha_a* und *alpha_b* zum Einsatz.

Sowohl α als auch γ werden in diesem Modus pseudozufällig aus Gammaverteilungen gezogen, deren Verteilungsfunktion von den zuletzt genannten Parametern abhängen. In der Beschreibung von HDP-faster wird dabei aufgeführt, dass die Parameter mit der Endung *_a* Formparameter sind und die beiden Parameter mit der Endung *_b* Skalenparameter. Ein Skalenparameter beeinflusst im Wesentlichen, ob der Graph einer Funktion gestaucht oder gestreckt wird. Weniger intuitiv verhält es sich mit dem Formparameter, da dieser, im Falle der Gammaverteilung, den Graphen der Funktion wandern lässt, im Falle anderer Funktionen das Aussehen des Graphen gänzlich verändern kann. Um einen Eindruck über die Auswirkung unterschiedlicher Formparameter bei einer Gammaverteilung zu gewinnen, wurden in der Abbildung 15 die Kurven im Bereich von x = [0,10] für die Formparameter 1,2 und 5 geplottet. Man sieht, dass das Maximum der Kurve mit höheren Werten wandert.

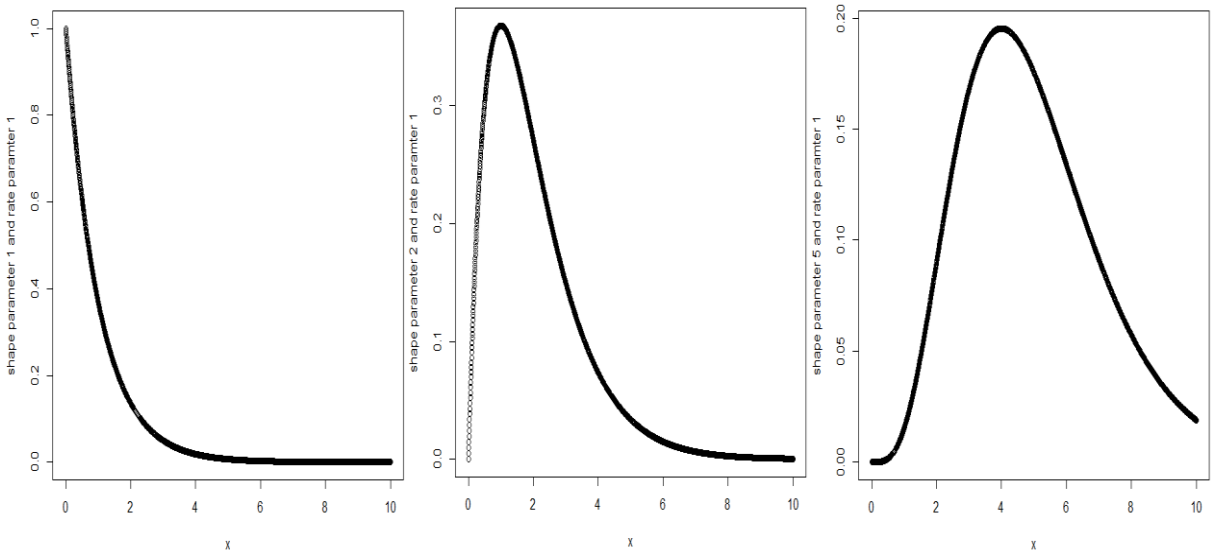

Abbildung 15: Unterschiedliche Kurven der Gammaverteilung für den Formparameter 1,2 und 5

Für die Nachverfolgbarkeit der Ergebnisse ist es wichtig zu betonen, dass alle Werte pseudozufällig ermittelt werden. Ausgehend von einem Startpunkt wird, in Abhängigkeit von diesem Startpunkt, immer eine feste Reihenfolge an Werten genommen. Möchte man also das Ergebnis eines Trainings noch einmal nachvollziehen, muss lediglich mithilfe der Option *–random_seed* der beim ersten Mal verwendete Startwert genutzt werden. Diesen findet man im Ergebnisordner des Durchlaufs in der Datei „settings.dat“.

```
$ ./hdp/hdp-faster/hdp.exe --test_data Daten/Ergebnis_1969.txt --model_prefix example-hdp/final --directory example-hdp
**********************************************************************************************
Working directory: example-hdp.
Reading data from Daten/Ergebnis_1969.txt.
number of docs   : 5785
number of terms  : 96388
number of total words : 924650
Loading model from prefix example-hdp/final...
```

Abbildung 16 Erfolgreicher Start eines Testlaufs

Um einen Testdurchlauf starten zu können, müssen die Parameter *–test_data* und *–model_prefix* dem Programm beim Aufruf übergeben werden. Bei den Testdaten handelt es sich um den Korpus, in dem nach Strukturen gesucht werden soll, bei dem Modellpräfix handelt es sich stattdessen um den Pfad zum zu testenden Modell inklusive des Modellnamens aber exklusive einer Dateiendung! Liegt das Modell daher im Verzeichnis mit dem Namen *Daten* und trägt den Namen *final*, so lautet die zu übergebende Zeichenkette „Daten/final“.

Ein möglicher Aufruf für einen Trainingsdurchlauf mit allen Einstellungsmöglichkeiten könnte also lauten:

> *hdp.exe --train_data "./Daten/Ergebnis_1969.txt" --sample_hyper --random_seed 1561581537 --gamma_b 0.001 --alpha_b 5.0 --directory "./Results_hdp" --max_iter 600 –max_time -1 --save_lag 1000 --alpha 0.05 --gamma 0.01 --eta 0.005 –verbose*

Für einen Testdurchlauf (Inferenz) würde der Aufruf hingegen lauten:

> *hdp.exe --test_data "./Daten/Ergebnis_1969.txt" –model_prefix „./Results_hdp/final“ --directory "./Results_hdp_test" --max_iter 600 --max_time -1 --save_lag 50 --verbose*

Bei dem Aufruf sind die Leerzeichen zwischen der Option und dem dazugehörigen Wert zu beachten!

Für beide Verfahren kann für ein geregeltes Ende entweder eine maximale Zahl an Iterationen (*--max_iter*) oder eine feste Dauer der Berechnung (*--max_time*) festgelegt werden. Möchte man das Programm unendlich lange laufen lassen, wählt man für beide Optionen den Wert -1. Dann empfiehlt es sich aber, mithilfe des Parameters *–save_lag* festzulegen, in welchen Abständen ein Abbild des Modells im Zielordner gespeichert werden soll.

```
$ ./hdp/hdp-faster/hdp.exe --test_data Daten/Ergebnis_1969.txt --model-prefix final --directory example-hdp
--model-prefix, unknown parameters, exit

C++ implementation of Gibbs sampling for hierarchical Dirichlet process, a much faster version.
Authors: Chong Wang, chongw@cs.princeton.edu, Computer Science Department, Princeton University.
usage:
      hdp               [options]
      --help:           print help information.
      --verbose:        print running information
```

Abbildung 17 Unknown Paramer. Abbruch eines Starts bei HDP-faster

Einflüsse der Parameter

α beeinflusst, wie stark sich die Themen unterscheiden. Wie bei der theoretischen Vorbetrachtung aufgezeigt, bedeuten große α's sehr homogene Verteilungen. Bei kleinen α's hingegen neigen die Verteilungen zu Extrema. Ziel bei der Modellerstellung ist es, ein α zu finden, dass klein genug ist, damit die Themen sich unterscheiden, dass aber auch groß genug ist, damit Themenbereiche nicht auseinandergerissen werden.

η beeinflusst, wie diffus die Wortverteilung ist. Ein größeres η sorgt dafür, dass die Wörter in den Themen vermischt und unnatürlich wirken. Ein kleineres η dagegen reduziert die Zahl an „themenfremden" Wörtern. Ganz abgestellt ($\eta = 0$) sollte η allerdings nicht werden, da ansonsten HDP-faster nicht mit unbekannten Begriffen arbeiten kann. η beeinflusst indirekt die Zahl an Themen, da mit steigender Größe von η mehr Begriffe miteinander in einzelnen Themen vermischt werden. Im Extremfall bedeutet dies, dass nur ein einzelnes Thema erstellt wird.

γ beeinflusst die Zahl an erstellten Themen maßgeblich. Je größer γ ist, desto wahrscheinlicher ist die Erstellung vieler Themen. Kleine γ's sorgen für eine flache Hierarchie mit wenigen, nahezu einheitlich großen Themen. Große γ's sorgen für eine große Hierarchie mit wenigen großen Topics und vielen kleinen Untertopics.

Dauer

Im Vergleich zu LDA-C gestaltet sich die Bestimmung der Dauer einer Modellberechnung schwierig, da das Programm keine vorgegebene Abbruchbedingung besitzt, d.h., dass kein mögliches Optimum existiert. Lediglich die Höchstzahl an Iterationen oder die maximale Dauer der Berechnung bestimmen, wie lange HDP-faster ein Modell berechnet. Wie aus dem Ende des letzten Abschnitts zudem hervorgegangen ist, bietet HDP-faster im Gegensatz zu LDA-C eine große Zahl an Einstellungs-möglichkeiten, die alle direkt oder indirekt Einfluss auf die Geschwindigkeit des Programms haben. Bedenkt man ferner, dass bei aktivierter Option „—sample_hyper" die Werte von α und γ variieren und dies wiederum Einfluss auf die Zahl an Themen hat, ist spätestens an dieser Stelle keine vergleichende Aussage über mögliche Berechnungsdauern mehr möglich, da die zufälligen Ziehungen der Werte zu unterschiedlichen Verläufen führen.

Lediglich für feste Standardwerte lässt sich ceteris paribus eine Aussage über die Auswirkung der Korpusgröße auf die Berechnungsdauer machen. Wie der Abbildung 18 entnommen werden kann, hängt die Bearbeitungsdauer für 100 Iterationen nahezu linear von der Zahl der Dokumente ab, wobei die Bearbeitungsdauer mit ca. sechszehn Minuten für 60550 Dokumente als sehr schnell angesehen werden kann.

Ergebnisse

Bei einem Trainingslauf werden fünf Dateien mit dem Namen *final* erstellt. Die größte und wichtigste Datei ist hierbei die Datei mit der Endung *.topics*, da in dieser die Häufigkeit jedes Wortes in einem Thema hinterlegt wird. In der Datei mit der Endung *.beta* wird die Anzahl der Dokumente festgehalten, die an einem dem jeweiligen Thema zugeordneten Tisch Platz genommen haben. Besonders mit der Option *–sample_hyper* ist die Datei mit der Endung *.info* interessant, da hier die finalen Werte von α und γ abgelegt werden. In der *.counts* Datei werden, dem Namen folgend, die Anzahl an Wörtern pro Thema abgelegt. Die Datei mit der Endung *.pi* schließlich macht den *StickBreaking* Prozess sichtbar, da in ihr die Größe der einzelnen Bruchstücke abgelegt werden. Der letzte Wert in dieser Datei ist die verbliebende Größe, die noch nicht einem Themenbruchstück zugeordnet wurde.

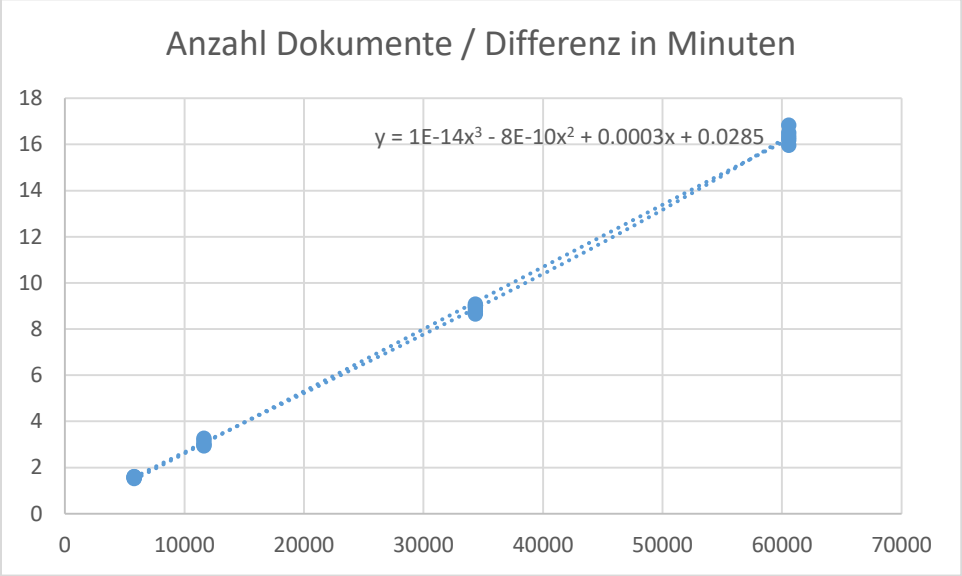

Abbildung 18 Dauer HDP-faster für 100 Iterationen bei festen Standardwerten

Nach dem Testlauf erstellt HDP-faster neben den Dateien des Trainingslaufs eine weitere Datei. In der Datei mit der Endung *.doc.states* werden für alle Dokumente die Themenzugehörigkeiten anhand der Worthäufigkeiten dokumentiert.

Bereits zu Beginn des Trainings-, aber auch des Testlaufs erstellt das Programm eine weitere Datei. In der Datei *train.log* bzw. ****-test.log* werden die Informationen abgelegt, die bei aktivierter Verbose-Option auch im Kommandofenster ausgegeben werden. Im Trainingsmodus kommt zudem die Datei *settings.dat* hinzu. Diese ist besonders interessant, wenn man ein Ergebnis nachvollziehen möchte, da in dieser Datei der Startseed hinterlegt wurde.

Robustheit und Nachvollziehbarkeit

Auch bei den Ergebnissen von HDP-faster kann es von Interesse sein, die Erstellung eines Modells noch einmal nachzuvollziehen. Wieder kommt es dem Nutzer dabei zu Gute, dass es sich bei den zufällig gezogenen Werten von HDP-faster um pseudozufällige Zahlen handelt. Anders als bei LDA-C hängt die Berechnung nicht von den Startpunkten der Topics ab, sondern von einem einzelnen Seed. Dieser wird praktischerweise in der Datei „settings.dat“ im Ergebnisordner abgelegt und kann so bei Bedarf ermittelt werden.

Topic Modeling für Historiker (TopModHis)

Die parallel zur Arbeit entstandene Anwendung TopModHis vereint zwei Funktionen. Zum einen stellt die Anwendung alle notwendigen Operationen bereit, um die von LDA-C und HDP-faster erstellten Modelldateien bearbeiten und einsehen zu können. Zum anderen bietet es Funktionen zur Verarbeitung der Testergebnisse an. Das Programm funktioniert für LDA-C wie auch HDP-faster gleichermaßen. Es handelt sich um eine menügesteuerte Konsolenanwendung, die aber auch automatisierte Aufrufe mittels Parameterübergabe zulässt.

Allgemeines

Das Programm wurde in der Sprache C++ entwickelt und kompiliert auf allen bekannten Betriebssystemen. Bei der Erstellung wurde darauf geachtet, dass nur betriebssystemunabhängige Bibliotheken eingebunden wurden. Die auf GitHub[92] zu findende ausführbare Datei wurde unter Windows kompiliert. Die Anwendung besteht aus einer Header und einer Cpp Datei, wobei in der Cpp Datei nur der Programmstart (main), das Menü und die verschiedenen Abkürzungen zum automatisierten Programmstart zu finden sind. Um TopModHis für beide Typen von Modellen nutzbar zu machen, wurde mit einem Template gearbeitet, wodurch sämtliche Funktionen, die TopModHis direkt betreffen, in der Header Datei definiert und initialisiert werden müssen.

```
No filepath given
Please type in the filepath to the .beta or .topics file
Path: final.beta
Reading Model at final.beta
read it
The model includes 6 topics
The model includes 762588 terms
The minimum got set to -758.265.
The maximum got set to 0.
Please choose:
1 = delete topic
2 = add topic
3 = append another model
4 = save topic to edit in external file
5 = show a range of values of a topic
6 = edit single value in a topic
7 = Print Topics
8 = Print assignment document to topics
9 = Print Dates
10 = exit and save
0 = exit
Your choice:
```

Abbildung 19 Startansicht TopModHis

Anwendungsstart

TopModHis kann über drei verschiedene Arten aufgerufen werden. Die einfachste Form des Starts ist der Doppelklick auf die ausführbare Datei oder der Aufruf über die Konsole. TopModHis erbittet dann einen Pfad zur .beta oder .topics Datei und lädt diese dann zur weiteren Verarbeitung. Es ist zu beachten, dass die Anwendung auch unter Windows akzeptiert, dass einzelne Pfadelemente mit einem Slash (/) abgetrennt werden und nicht, wie bei Windows eigentlich üblich, mit einem Backslash (\). Die Pfadangabe kann relativ erfolgen, wobei nur der Doppelpunkt (../) und nicht der einfache Punkt (./) für die Pfadangabe akzeptiert werden.

[92] Müller, GitRepository TopModHis.

Möchte man direkt den Pfad zur zu verarbeitenden Modelldatei angeben, so hängt man diesen bei dem Aufruf über die Konsole einfach an. Möchte man auf die Navigation in der Konsole verzichten, so kann man auch eine Verknüpfung zur Anwendung erstellen und in dieser die gewünschten Informationen unter „Eigenschaften“ -> „Ziel“ an den Pfad der Anwendung anhängen.

Geht man beispielsweise davon aus, dass die „final.beta“ Datei im Ordner „Ergebnisse“ liegt und der Aufruf einen Ordner darüber erfolgt, so wäre ein direkter Aufruf über

tmh.exe Ergebnisse/final.beta

möglich. Unter Linux und MacOS ist ein führender Punkt notwendig. Der Aufruf lautet hier

./tmh.exe Ergebnisse/final.beta

Möchte man keine Änderungen an einem Modell vornehmen, sondern nur die Begriffe pro Thema oder die Hierarchie ermitteln, so können diese Operationen ohne Menü direkt aufgerufen werden, indem man ähnlich wie zuvor die notwendigen Informationen dem Programmaufruf anhängt. Alle modellrelevanten Informationen befinden sich bei den folgenden beispielhaften Aufrufen wieder im Unterordner „Ergebnisse“ und alle korpusrelevanten Informationen auf der gleichen Ebene wie der Funktionsaufruf. *vocab.txt* steht dabei für das Vokabular, in dem pro Zeile ein Begriff aufgeführt wird. Die Zeilennummer entspricht hierbei später der ID im Modell. *doc_id.txt* enthält eine geordnete Liste der Dokumentennamen, wobei jede Zeile einem Dokumentennamen entspricht. In dieser Struktur lauten die direkten Aufrufe:

- Für das direkte Ausgeben der Topbegriffe pro Thema:

  HDP: tmh.exe Ergebnisse/final-test.topics 25 vocab.txt

  LDA: tmh.exe Ergebnisse/final.beta 25 vocab.txt

Die „25“ im Aufruf zur Ausgabe der Topbegriffe steht für die Menge an Begriffen, die gedruckt werden sollen.

- Für das Ermitteln der Hierarchie und der Themen geordnet nach Dokumenten:

  HDP: TopModHis.exe Ergebnisse/final-test.topics Ergebnisse/final-test.doc.states

  LDA: TopModHis.exe Ergebnisse/final.beta Ergebnisse/final-gamma.dat

- Für das Ermitteln der Zeit:

  HDP: tmh.exe Ergebnisse/final.topics Ergebnisse/final.doc.states Doc_id.txt

  LDA: tmh.exe Ergebnisse/final.beta Ergebnisse/final-gamma.dat Doc_id.txt

- Für das Erstellen eines Modells aller vereinten Hierarchieelemente, die ein bestimmtes Thema enthalten:

  HDP: tmh.exe Ergebnisse/final.topics Ergebnisse/final.doc.states Doc_id.txt 6

Es gilt zu beachten, dass das Ermitteln einer Hierarchie nur für HDP Modelle sinnvoll ist (.topics Datei), während alle anderen Funktionen auch für LDA Modelle zur Verfügung stehen. Die Zahl 6 bei der Erstellung der vereinten Hierarchieelemente steht für das Thema, das in allen zu berücksichtigenden Hierarchieelementen enthalten sein muss. TopModHis extrahiert diese Themen, addiert alle Themen für das entsprechende Hierarchieelement auf und gibt ein neues Modell aus.

Für die Weiterentwicklung der Anwendung wurde ein Debug-Modus integriert. Dieser ermöglicht es, langwierige Ladezeiten zu vermeiden und direkt auf bestimmte Funktionen zuzugreifen. Die verschiedenen Testszenarien müssen in der Funktion DebugMode() am Ende der Header Datei konfiguriert werden. Die einzelnen Bausteine, die bereits für die Erstellung eines Szenarios bereitstehen, beginnen mit dem Präfix *Debug.* Eine Auflistung über die vorhandenen Bausteine findet man in der Klassendefinition der Klasse Modeleditor unter dem Punkt Debug. Der Debug Modus kann auf zwei Arten gestartet werden. Wenn man dem Programm beim Start statt des Pfades für die Modelldatei „42“ oder „42,42“ übergibt, startet TopModHis den Debug Modus entweder im HDP Format (42) oder im LDA Format (42,42). Zudem kann der Debug Modus im Hauptmenü über die nicht aufgeführte Option 42 gestartet werden.

```
Reading Model at 42
You are in the Debug Mode of TopModHis

Please select the routine you want to check:
1: Doc_Names_to_Date
2: Print_Topic_in_dates
3: PrintHDP
4: Print Topics
0: Exit

Your choice: 1
We need a document with filenames. Please tell the path to a testfile:
Filepath: ../Daten/Dokumente_sortiert_all_Unix.txt
Using Date Format 7.
```

Abbildung 20 Debug Modus

Abbildung 20 zeigt einen Aufruf des Debug Modus im HDP Format (ersichtlich an der Zeile: „Reading Model at 42“). Interessant ist hierbei die Zeile „Using Date Format 7“, die auf eine weitere Besonderheit von TopModHis hinweist. Um dem Anwender bei der Zuweisung von Datumszahlen zu seinen Dokumenten größtmögliche Freiheit zu lassen, wurde eine breite Zahl an Datumsformaten im Programm berücksichtigt. Im konkreten Fall weist TopModHis darauf hin, dass es Jahreszahlen und Kalenderwochen als Datumsformat gefunden hat. Die Ermittlung und Zuordnung der Themen zu Datumsangaben erfolgt hierbei über die Dokumentenbezeichnungen selber. Es wurde darauf geachtet, dass die Vorgaben für die Formatierung sehr gering sind. Das Programm unterscheidet zwischen der amerikanischen und der europäischen Datumsformatierung, akzeptiert Jahrestage und Kalenderwochen und schreibt die Position des Datums im Dokumentennamen im Grunde nicht vor. Dennoch sind, um die unterschiedlichsten Formate unterscheiden zu können, einige Regeln bei der Benennung der Dokumente zu beachten. Um diese Regeln zu verstehen, lohnt es sich, den Mechanismus hinter der Datumerkennung zu erläutern.

TopModHis ermittelt das Datum aus einer Reihe von Dokumentennamen, indem nach den ersten maximal drei aufeinanderfolgenden, durch **ein** beliebiges nichtnumerisches Trennzeichen getrennten Zahlen in jedem Namen gesucht wird. Dies erlaubt Dokumentennamen wie *Name01-01-2000* oder *35#2000#Name#Laufziffer*, aber nicht *Laufziffer-Name-01-01-2000*, da im letzten Fall mit der Laufziffer bereits eine Zahl im Dokumentennamen vor dem Datum steht, die TopModHis fälschlicherweise als Datum interpretiert.

Bei der Analyse der verschiedenen gefunden Datumsangaben wird nach dem Maximalwert von jedem der drei Werte gesucht und diese Maximalwerte danach nach ihrer Wertigkeit geordnet. Wird nur eine Zahl gefunden, so muss es sich für TopModHis um eine Jahreszahl handeln. Bei zwei gefundenen Zahlen wird die größte der beiden gefundenen Zahlen als Jahreszahl angenommen. Für die kleinere Zahl versucht TopModHis zu unterscheiden, ob es sich um Kalenderwochen, Monate oder Tage im Jahr handeln kann. Die einfache Logik ist hierbei, dass ein Jahr maximal 12 Monate, 53 Wochen besitzt oder 366 Tage besitzt. Wenn der Maximalwert der zweiten gefundenen Zahl also kleiner gleich 12 ist, nimmt TopModHis an, dass es sich um Monate handeln muss. Für Zahlen, die größer als zwölf aber kleiner als 54 sind, geht TopModHis von Kalenderwochen aus. Für Zahlen im Bereich von 54 bis 366 wiederum geht TopModHis von Tagen im Jahr aus. Sollte die zweite Zahl keinen der Kriterien entsprechen, so wird der Wert nicht weiter berücksichtigt und stattdessen die Jahreszahl als einziger Bestandteil der Datumsangabe angenommen. Wie der Schilderung entnommen werden kann, ist vom Nutzer darauf zu achten, dass TopModHis das gewünschte Datumsformat auch erkennen kann. Besonders bei den zweiteiligen Datumsangaben ist es empfehlenswert, Werte zu haben, die klar im Bereich des jeweiligen Formates liegen.

Für die Fälle, dass TopModHis das Format Monat und Jahr bzw. nur Jahr erkennt, werden die nicht erkannten Bestandteile mit einer 1 aufgefüllt. Wird also nur das Jahr 1969 erkannt, hinterlegt TopModHis den 01.01.1969 als vollständiges Datum. Wird der Dezember 1969 als Datum erkannt, wird der 01.12.1969 im System als Datum abgelegt.

Ergebnisse

TopModHis erstellt unterschiedliche Dokumente in Abhängigkeit von den gewählten Optionen. Um die Funktion der unterschiedlichen Dokumente und ihren Inhalt besser verstehen zu können, folgt an dieser Stelle eine kurze Übersicht über die einzelnen Dokumente. Dabei wird sowohl auf den Inhalt als auch auf die Möglichkeiten eingegangen, wie die Datei erzeugt werden können.

Modell Datei [.beta/.topics]

Bei den Modelldateien bestimmt der Nutzer selber, wie er die Datei benennen möchte, jedoch wird automatisch in Abhängigkeit des Modus von TopModHis die Endung .topics im HDP Modus oder .beta im LDA Modus angehängt. Abgelegt werden die neuen Modelldateien immer in dem Ordner, in dem auch das Modell liegt, mit dem die Anwendung initialisiert wurde. Der Aufbau folgt dem jeweiligen Vorgaben der beiden Topic Modeling Anwendungen. Um eine solche Datei erstellen zu können, muss im Hauptmenü die Option 10 ausgewählt werden und das Anlegen einer neuen Modelldatei gewünscht werden (Überschreiben? -> Nein).

Datei für einzelne Topics

Bei den Dateien für die exportierten Themen kann der Nutzer Name und Dateiendung selber bestimmen. Der einzige Unterschied zwischen der Datei für einzelne Topics und einer Modell Datei besteht darin, dass die Zeilen transponiert wurden. Da die meisten Texteditoren Schwierigkeiten haben, unendlich lange Zeilen darzustellen, mit unendlich vielen Zeilen hingegen gut zurechtkommen, hat sich diese Umwandlung für eine bessere Bearbeitung angeboten. Die Erzeugung solcher Dateien geschieht über die Option 4 des Hauptmenüs.

Final_Cluster.txt

Die Final_Cluster.txt Datei beinhaltet im Grunde die Informationen, die die meisten Topic Modeling Anwendungen anzeigen. Nach den jeweiligen Themen sortiert stehen untereinander die wichtigsten Begriffe jedes Themas. Die Anzahl der Begriffe pro Thema wurden dabei vorher vom Nutzer festgelegt. Es wurde bei der Gestaltung dieser Datei darauf geachtet, dass der Import in anderen Anwendungen wie z.B. R oder Excel besonders einfach ist. Dazu wurde eine feste Spaltengröße von 50 Einheiten gewählt, die Themen nebeneinander angeordnet und als Trennzeichen das Leerzeichen gewählt. So können die Dateien wie CSV Dateien importiert werden. Um final_cluster.txt erstellen zu können, kann man entweder den weiter oben beschriebenen parametergestützten Start nutzen oder im Hauptmenü die Option 7 auswählen.

TopicsinDocuments.txt

Jede Zeile dieser Datei entspricht einem Dokument und in jeder Zeile reihen sich der Auftretens-häufigkeiten nach absteigend sortiert die Themen, die dem Dokument zugeordnet werden konnten. Die Ausgabe der Datei erfolgt parallel zur Ausgabe der TopicLevelIndex.txt und hilft, die Dokumente zu finden, die einem zu untersuchenden Thema zugeordnet wurden, da die Zeilennummern der Zeilen, in denen das gewünschte Topic enthalten ist, den Ids der Dokumente entspricht, die das Topic enthalten.

TopicLevelIndex.txt

Aufbauend auf den Informationen, die der Nutzer in TopicinDocuments.txt einsehen kann, entwickelt TopModHis eine Hierarchie der Themen. Die Hierarchie bildet sich dabei wie folgt:

Die erste Ebene der Hierarchie wird aus dem jeweils ersten Topic der Topics gebildet, da diese das jeweils größte Topic darstellen. Dies folgt der Logik aus der theoretischen Vorbetrachtung, dass es ein Restaurant / Topic geben muss, dass von nahezu allen Dokumenten angesteuert wird. In den weiter unten betrachteten Szenarien gab es zwar in der Regel ein Topic, das diesem Restaurant nahekam, aber es gab auch verschiedene andere Topics, die als Einstiegsrestaurant dienten. Im nächsten Schritt müssen alle Themen ermittelt werden, die in den Dokumenten das nächst kleinere Restaurant darstellen, d.h. dem Startrestaurant folgen. Dieser Vorgang wird im Anschluss solange wiederholt, bis alle Topics in allen Dokumenten erfasst wurden. Ein Knoten auf einer Ebene stellt nun die Kombination der verschiedenen Themen dar. Sei Thema 0 also der Startpunkt, dann ist Thema 0 auch Element der ersten Ebene. Seien nun die Themen 10 und 11 Themen, die auf das Thema 0 folgen, dann sind mögliche Elemente der zweiten Ebene „0 10“ und „0 11“. Folgt auf Thema 10 nun in der nächsten Iteration Thema 20 und auf Thema 11 Thema 22, dann sind mögliche Elemente der dritten Ebene „0 10 20“ und „0 11 22“.

Betrachten wir für ein konkretes Beispiel:

14 35 55 31 34 5
14 55 50 31 34 17 0 54 53 91 58
14 35 55 50 31 34 17 5 16 22 19 11
14 35 55 50 31 34 17 5 0 7 16 22 46 130 80
14 35 55 50 31 34 0 13 2 19
14 50 31 34 5 0 13

Element der ersten Ebene ist eindeutig das Thema 14. Hierbei handelt es sich meist um Begriffe, die von fast allen Dokumenten geteilt werden. Ohne Vorverarbeitung handelt es sich bei diesen Begriffen meist um Funktionswörter, die bei den meisten Analysen über Stopwort Listen entfernt werden. [93] Auf dieses Thema folgen die Themen 35,50 und 55. Dies bedeutet, dass sich die Elemente der zweiten Ebene aus den Kombinationen „14 35“, „14 50“ und „14 55“ ergeben. Die Elemente der dritten Ebene bilden sich nun aus allen eindeutigen Kombinationen, die auf den Elementen der zweiten Ebene aufbauen. Im konkreten Fall bedeutet dies „14 35 55“, „14 55 50“, „14 50 31“. Es gilt zu betonen, dass diese Rangfolgen eindeutig sind. Folgt also in einem Hierarchieelement auf Thema 14 Thema 50 als nächstgrößeres Thema, so wird Thema 35 nicht zu einem späteren Zeitpunkt in der Kette als Element erscheinen! Eine Kombination 14 50 35 ist in dem betrachteten Beispiel nicht möglich.

Die eigentlichen Themen bilden sich bei HDP-faster nun aus der Verkettung der einzelnen Unterthemen. Es genügt also nicht wie bei LDA-C, die Topwörter der ermittelten einzelnen Themen zu betrachten, sondern erst die Verknüpfung der einzelnen Unterthemen miteinander lässt ein eindeutiges Thema entstehen. Würde man die Themen nur getrennt betrachten, so erscheinen zwar einige Themen bereits ohne die anderen Themen klar, die meisten Themen aber erscheinen als unlogisches Gemisch aus Topwörtern verschiedener Themen. Die Datei wird im Zuge der Hierarchieerstellung (Option 8) erstellt oder die Erstellung wird direkt angefordert (s. *Ermitteln einer Hierarchie*).

[Zahl]_[Dateiname].topics

Auf Grundlage der Themenhierarchie, wie sie in der Datei TopicLevelIndex.txt eingesehen werden kann, ist es möglich, sich die vereinten Hierarchieelemente als eigenständiges Modell ausgeben zu lassen. Da die Ausgabe aller Elemente eine lange Bearbeitungsdauer bedeutet und das Ergebnis aufgrund der Dateigröße kaum zu nutzen ist, wurde die Ausgabe auf Elemente beschränkt, die ein gewünschtes Thema enthalten. Damit der Nutzer im Anschluss weiß, welches Thema bei welcher Modelldatei im Fokus steht, wird die Zahl dem Dateinamen vorangestellt. Die Ausgabe dieser Datei erfolgt entweder im Zuge der Hierarchieerstellung (Option 8), kann aber auch direkt aufgerufen werden (s. *Erstellen vereinter Hierarchieelemente*).

DateTopic.txt

In dieser Datei werden die Vorkommen jedes Topics zu einem Zeitpunkt abgelegt. Das jeweilige Datum steht hierbei am Anfang einer Zeile und entspricht dem europäischen Standard. Im Anschluss folgt die

[93] Vgl. Blei et al., The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, S. 21.

Topic ID und die Häufigkeit des Auftretens des jeweiligen Themas. Die Erstellung der Datei kann über die Option 9 im Hauptmenü angefordert werden, direkt gestartet werden (s. *Ermitteln der Zeit*) oder im Zuge der Hierarchiestellung mit angefordert werden.

DateDocument.txt

In dieser Datei wird abgelegt, welches Dokument welchem Datum zugeordnet wurde. Die Erstellung der Datei erfolgt parallel zur DateTopic.txt. Beide Dateien werden entweder über einen parameter-gesteuerten Start (s. *Ermitteln der Zeit*), über die Option 9 im Hauptmenü oder zusammen mit der Hierarchie erstellt.

Laden eines leeren Modells

In einigen Situationen bietet es sich an, ein neues, leeres Topic zu erstellen. Hierfür übergibt man TopModHis einen gültigen Dateinamen (.topics oder .beta am Ende), für den es im entsprechenden Ordner keine Datei gibt. TopModHis lädt dann ein leeres Topic. In dieses können nun exportierte Topics oder ganze Modelle eingefügt werden und am Ende kann das Modell bei Bedarf über die Option 10 gespeichert werden. Es ist lediglich darauf hinzuweisen, dass an verschiedenen Stellen nicht alle Funktionen zur Verfügung stehen, da dafür eine korrekte Initialisierung des Modellmaximums und –minimums notwendig ist. In diesen Fällen bietet es sich an, zunächst ein Modell mit einem einzigen Topic zu speichern, zu laden und erst dann alle gewünschten Operationen durchzuführen.

```
Reading Model at empty.topics
The model includes 0 topics
The minimum got set to 500.
The maximum got set to 0.
Please choose:
1 = delete topic
2 = add topic
3 = append another model
4 = save topic to edit in external file
5 = show a range of values of a topic
6 = edit single value in a topic
7 = Print Topics
8 = Print assignment document to topics
9 = Print Dates
10 = exit and save
0 = exit
Your choice:
```

Abbildung 21 Ein leeres Topic, man beachte die Werte für Maximum und Minimum

Anwendung

Um die Möglichkeiten des Topic Modelings zu veranschaulichen, sollen zwei übliche Aufgaben betrachtet und Schritt für Schritt nachverfolgt werden. Zum einen sollen mithilfe eines trainierten Modells Dokumente Themenfeldern zugeordnet werden und zum anderen sollen die wichtigsten Themenfelder in einer Dokumentensammlung ermittelt werden. Der Fokus bei diesen Aufgaben ist die unterschiedliche Arbeitsweise von HDP in Form von HDP-faster und LDA in Form von LDA-C sowie die Funktionsweise der im Rahmen von TopModHis vorgestellten Erweiterungen. Als Datenquellen dienen Artikel der

Wochenzeitung DIE ZEIT der Jahre 1969 bis 1989.[94] Da es sich hierbei z.T. um Retrodigitalisate handelt, befinden sich im Korpus auch Wörter mit falscherkannten Zeichen. Diese wurden bewusst nicht geändert, da Fehler in der Zeichenerkennung ein häufiges auftretendes Phänomen bei der Arbeit mit historischen Texten darstellt und es daher für den historischen Forscher wichtig ist zu verstehen, wie die Programme mit diesen Fehlern umgehen können. Wird im Kontext des Arbeitskorpus von Wörtern gesprochen, so sind hiermit ausschließlich die Grundformen von Nomen gemeint. Alle weiteren Wortarten wurden so gut wie möglich mittels eines TreeTaggers[95] vom Korpus ausgeschlossen.

Die im Folgenden gemessenen Zeiten beziehen sich auf das Testsystem des Autors, bestehend aus einer normalen HDD mit 7200 Umdrehungen pro Minute, 12 GB Arbeitsspeicher und einem Intel I7 980 Extreme Edition Prozessor unter dem Betriebssystem Windows 10. Diese Zeiten können für andere Systemzusammensetzungen stark variieren und sollen lediglich als Orientierung dienen, mit welchen Bearbeitungszeiten bei solchen Aufgaben zu rechnen ist.

Szenario 1: Zuordnung von Dokumenten zu Topics

Bezieht man sich auf die Vorstellung der Autoren von LDA, liegt die große Stärke dieser Anwendung darin, Strukturen in Korpora aufzudecken und zu evaluieren.[96] Wie eine solche Evaluation aussehen könnte, soll im ersten Szenario des Anwendungskapitels aufgezeigt werden. Alle weiterführenden Berechnungen, die weder von LDA noch von einem der Werkzeuge aus TopModHis bereitgestellt werden, werden mithilfe von R durchgeführt. Die entsprechend erstellten Skripte finden sich unter Szenario1.R im Ordner „Results Szenario“ des Repository dieser Arbeit.[97]

Das Szenario

Betrachtet werden soll ein Korpus aus Artikeln der deutschen Wochenzeitung DIE ZEIT der Jahre 1969-1989. Aus früheren Analysen dieses Korpus ist bekannt, dass dieser insgesamt dreizehn Ressorts mit einem Umfang von mehr als 100.000 Wörtern enthält und die Artikel von sieben dieser dreizehn Ressorts sogar mehr als eine Million Wörter über die erfassten 20 Jahre umfassen. Die Größe wird dabei nach der absoluten Anzahl der Wörter pro Ressorts gemessen. Die sieben sehr großen Ressorts tragen die Bezeichnungen „die Zeit“, „Kultur“, „Wirtschaft“, „Politik“, „Gesellschaft“, „Lebensart“ und „Wissen“. Die nächstkleineren Ressorts heißen „Länderspiegel“, „Reisen“, „Unzugeordnet“, „Sport“, „Karriere“ und „Auto“. Während elf Ressorts Namen tragen, die auch in anderen Printmedien üblich sind, können die Kategorien „unzugeordnet“ und vor allem „die Zeit“ nicht mit Themen anderer Medien in ein Verhältnis gesetzt werden. Um die Inhalte dieser Ressorts übertragbar zu machen, ist eine Analyse der inneren Struktur dieser Themen erforderlich Handelt es sich schlicht um bestimmte Seiten wie z.B. die Titelseiten einer Ausgabe mit vermischten Inhalten? Oder haben diese Ressorts eine eigene, von den Inhalten der anderen Ressorts klar abgrenzbare Strukturen? Um dieser Frage nachzugehen, werden

[94] Die Daten wurden im Rahmen einer früheren Arbeit heruntergeladen und aufbereitet. Gerne wäre in dieser Arbeit eine Verlinkung zu konkret betrachteten Artikeln vorgenommen worden, jedoch hat DIE ZEIT ihr Onlinearchiv drastisch ausgedünnt, weshalb eine Reihe von früher öffentlich zugänglichen Artikeln nicht mehr einsehbar ist.

[95] Helmut Schmid, Improvements in Part-of-Speech Tagging with an Application to German, Stuttgart, 1995, online verfügbar unter: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf. Zuletzt geprüft am: 03.12.2015.

[96] Vgl. Blei, David M., Topic Modeling and Digital Humanities, 2012. Erschienen am 16.06.2019, online verfügbar unter: http://journalofdigitalhumanities.org/2-1/topic-modeling-and-digital-humanities-by-david-m-blei/.

[97] Müller, GitRepository TopModHis.

jeweils zwei Modelle mit 7 und 13 Themen erstellt, in der die beiden Ressorts als eigenes Thema angenommen werden und zwei Vergleichsmodelle mit 6 und 11 Themen, in denen davon ausgegangen wird, dass beide Ressorts keine Bedeutung für die Themenstruktur des Korpus haben. Bei der Zahl der Themen wurde berücksichtigt, dass „Unzugeordnet“ nur bei 13 Themen zur Menge aller Themen zählt, während „die Zeit“ sowohl bei den sieben Themen als auch bei den 13 Themen Teil der Themenmenge ist. Daher ist ein Modell nur ein Thema kleiner, während bei dem anderen Modell zwei Themen abgezogen wurden.

Für das Szenario werden drei Listen verwendet. Zwei der Listen werden jeweils zum Erstellen der zuvor beschriebenen Modelle verwendet und bilden Stichproben, die dritte Liste dagegen ist die Grundgesamtheit aller Artikel, an der die erstellten Modelle getestet werden sollen. Hintergrund für die unterschiedlichen Stichproben ist die Frage, ob kleinere, gezielt ausgewählte Stichproben einer zufällig ausgewählten, größeren Stichprobe überlegen sein können oder ob sich die zu erwartende längere Bearbeitungszeit von größeren Stichproben mit einem besseren Ergebnis begründen lässt.

Versuchsaufbau

Für die erste Trainingsstichprobe werden fünf Jahrgänge zufällig ausgewählt und analysiert. Mithilfe der Funktion randint(1969,1989) aus dem Python Paket random wurden dabei der Reihe nach die Werte 1973, 1971, 1975, 1987, 1978 gezogen. Da es sich bei vier der fünf Stichproben um zeitlich nahe beieinander liegende Zeiträume handelt, könnte es zu einer thematischen Verschiebung zu Themen der frühen siebziger Jahre kommen. Als Gegenentwurf zu der Zufallsstichprobe werden die Jahrgänge der Jahre 1974 und 1984 ausgewählt, die dem unteren und oberen Quartil der gesamten Zeitspanne entsprechen.

Als Qualitätsmaß werden die von LDA gemessenen durchschnittlichen Likelihoods der Modelle über sechs und elf Themen verwendet. Da einer der wesentlichen Aspekte des wahrscheinlichkeitsbasierten Topic Modelings die Maximierung der Likelihoods ist, kann davon ausgegangen werden, dass ein größerer ermittelter durchschnittlicher Likelihood Merkmal eines Modells höherer Qualität ist. Das höhere Likelihood steht dabei stellvertretend für eine bessere Anpassung des Modells an die Beobachtungen, d.h. die Inhalte der Artikel lassen sich besser den verschiedenen Strukturen zuordnen.

Um eine größtmögliche Vergleichbarkeit der Ergebnisse gewährleisten zu können, wird auf eine Alpha Optimierung während der Modellerstellung verzichtet. Um dabei die Berechnungszeit minimal zu halten, wird für den Einstiegspunkt von α der Wert 0.01277 gewählt, der bei den Probeläufen im Kapitel zuvor als einer der zeitoptimalen Werte für den zu betrachtenden Korpus ermittelt wurde. Da die Größen der Themen variieren, würde ein einheitlicher Startpunkt die Vergleichbarkeit nicht verbessern. Daher wird LDA erlaubt, die Startpunkte der Themen selber zu ziehen.

Durchführung

Grundvoraussetzung für die Durchführung des Experimentes ist die Existenz der Liste des Gesamtvokabulars. Ohne dieses Gesamtvokabular würden später die Zuordnungen von Begriffen und Themen nicht stimmen, da nicht alle Begriffe des Gesamtvokabulars in jedem Jahrgang vertreten sind. Das Erstellen der Wortliste über 20 Jahrgänge hat mit 38 Stunden und 15 Minuten fast zwei Tage

benötigt. Die resultierende LDA-C Format Textdatei hat eine Größe von knapp 120 MB und umfasst 114722 Dokumente sowie 762588 Begriffe. Ein schwacher Trost für den Anwender ist es, dass die Berechnung der beiden Stichproben durch die Nutzung des nun vorliegenden Gesamtvokabulars beschleunigt wird. Bei zwei Jahrgängen spart man etwa drei Minuten Bearbeitungszeit ein, bei fünf Jahrgängen sind es schon fast anderthalb Stunden. Das Erstellen der Stichprobe aus fünf Jahrgängen dauerte insgesamt 123 min und umfasst 26936 Dokumente. Die Berechnung der zweiten Stichprobe aus zwei Jahrgängen dauerte 21 min und umfasst 10814 Dokumente.

An dieser Stelle sei auf eine Besonderheit von LDA hingewiesen, die bei der Arbeit in Verbindung mit TopModHis keine weitere Bedeutung hat, bei der Verwendung des mitgelieferten Pythonskripts aber zu einem Fehler bei der Erstellung der Topwortlisten führen kann. Da LDA die Größe des Vokabulars selber berechnet, kann es passieren, dass die Größen des eigentlichen Vokabulars von dem durch LDA berechneten Werten abweichen. In diesem Fall verweisen Teile der Wortlisten auf nicht definierte Speicherplätze, die einen Fehler und Abbruch zur Folge haben. Um dem Anwender daher eine zeitaufwendige Fehleranalyse zu ersparen, sollte dieser wissen, dass LDA die maximale Größe des Vokabulars anhand der größten Wort ID festlegt, die LDA beim Laden des Korpus findet. Soll sichergestellt werden, dass die von LDA ermittelte Korpusgröße mit der wahren Korpusgröße übereinstimmen, so muss der Korpus ein Dokument enthalten, indem das Wort mit der größten Wort ID im Vokabular vorkommt. Eine solche Anpassung ist bei den Korpusgrößen in dieser Arbeit nicht weiter von Relevanz für das Ergebnis, kann aber bei kleineren Stichproben zu Verschiebungen führen. TopModHis ist gegenüber solch unterschiedlichen Größen unempfindlich. Unbekannte Begriff-IDs führen lediglich zu einer leeren Zeile im Enddokument, nicht aber zu einem Abbruch der Berechnungen oder einem Absturz der Anwendung.

Das Erstellen der beiden Modelle durch LDA mit sechs Themen ohne Alpha-Optimierung ging angenehm schnell. Für das Modell mit zwei Jahrgängen dauerte es nur sechszehn Iterationsschritte bzw. 06:51 min, bis das komplette Modell kalibriert war. Das Modell mit fünf Jahrgängen benötigte 17 Iterationsschritte bzw. 11:56 min. Vergleicht man dies mit den Zeiten eines einzelnen Jahrgangs mit Alpha-Optimierung, bei denen die Bearbeitungszeit zwischen 52 und 67 Minuten beträgt, zeigt sich als eine erste Erkenntnis, dass die Zeit, die vorab in die Optimierung des Alpha-Wertes investiert wird, für große Analysen sinnvoll investiert ist.

Das Training der beiden Modelle mit elf Themen dauerte erwartungsgemäß länger, blieb aber mit 12 bzw. 23 Minuten in einem annehmbaren Rahmen gemessen an den Größen der Stichproben. Der Test der Modelle am Gesamtkorpus erfolgte im Anschluss sehr zügig und dauerte selbst bei den großen Modellen keine fünf Minuten

Ergebnisse

Der Vergleich der Likelihoods der unterschiedlichen Stichprobengrößen fiel eindeutig zu Gunsten der mehr Dokumente umfassenden, zufällig gezogenen Stichprobe aus. Bei sechs Themen betrug die Differenz der durchschnittlichen Likelihoods bereits 2,85 Einheiten. Für elf Themen wuchs der Abstand sogar auf 2,94 Einheiten weiter an. Zumindest im betrachteten Fall ist also eine auf mehr Dokumente gestützte Modellerstellung einer gezielt ausgewählten Stichprobe vorzuziehen. Allgemein war die

Tendenz, dass eine steigende Zahl an Themen eine Maximierung der Likelihoods mit sich brachte. Dies lässt sich wahrscheinlich darauf zurückführen, dass mit steigender Themenzahl die Anzahl der Fehlallokationen abnimmt.

Über die verschiedenen Modelle hinweg lassen sich bei den Ressorts zwei unterschiedliche Verhalten beobachten. Sofern es ein Thema gibt, dass den Inhalten der Ressortdokumente entspricht, werden, unabhängig vom Umfang des Ressorts, zwischen 80 bis weit über 90 Prozent der Dokumente des Ressorts diesem LDA Thema zugeordnet. Ferner fallen die Zuteilungen zu anderen LDA Themen stark ab. Ist der Maximalwert beim Ressort Wirtschaft und sechs Themen 95,94 Prozent, so ist der Minimalwert bei 17,7 Prozent. Wir können bei solchen Mustern also von sicheren Zuweisungen und LDA Themen mit eigenen Strukturen ausgehen. Bei anderen Themen hingegen verteilen sich die Dokumente über die verschiedenen LDA Topics nahezu gleichmäßig oder sie werden LDA Themen zugeschrieben, die bereits eindeutig anderen Themen zugeordnet wurden. Ein gutes Beispiel hierfür sind im Modell für sechs Themen die Ressorts „Gesellschaft“ und „Wissen“. Der Maximalwert an Dokumenten aus dem Ressort „Wissen“ die einem der sechs LDA Themen zugeordnet wurden, liegt bei 60,36 Prozent. Der Minimalwert liegt dagegen bei 34,41 Prozent. Die Differenz zwischen Minimal- und Maximalwert liegt nur bei knapp 25 Prozent, die Differenz zwischen Median und Maximum beträgt sogar nur wenige Prozentpunkte. Bei dem Ressort „Gesellschaft“ ist die Differenz zwischen Minimum und Maximum zwar bei 41 Prozent, dieser Wert ist dennoch weit entfernt von der Differenz von 78 Prozent des Wirtschaftsressorts. Als grobe Faustregeln hat sich zur Bewertung der Qualität der einzelnen Ressortzuweisungen über alle Modellgrößen hinweg eine Mindestdifferenz zwischen Maximum und Minimum von über 50 Prozent als primäres Kriterium und ein Maximalwert von über 70 Prozent als sekundäres Kriterium bewährt. In einigen Fällen kam es vor, dass mehrere Ressorts den Kriterien entsprechend gleiche Topics als größte Übereinstimmung hatten. In diesem Fall hatte meist ein Ressort ein weiteres gültiges Topic, während für das andere Ressort nur ein LDA Topic in Frage kam. Sofern eine eindeutige Einteilung gewünscht ist, hat es sich meist bewährt, dann dem Ressort mit zwei gültigen Zuweisungen das zweite LDA Thema mit der etwas geringen statistischen Übereinstimmung zuzuteilen.

Tabelle 1 Beispiel für unterschiedliche Eindeutigkeiten bei den Themenzuordnungen

Ressort Wirtschaft: 18948 Documents total						
Totale	3354	6038	6631	7568	10564	18178
Relative Häufigkeiten	0.1770108	0.3186616	0.3499578	0.3994089	0.5575259	0.9593625
Topic Nr.	4	2	5	6	3	1
Ressort Wissen: 7584 Documents total						
Totale	2610	2947	4280	4509	4532	4578
Relative Häufigkeiten	0.3441456	0.3885812	0.564346	0.5945411	0.5975738	0.6036392
Topic Nr.	4	3	1	5	6	2

Betrachten wir nun die Dokumente des Ressorts „die Zeit“, so liegt die Differenz zwischen Maximum und Minimum bei sechs Themen bei knapp 33 Prozent. Daher ist bei dem Ressort davon auszugehen,

dass es sich hierbei um vermischte Themen zu handeln scheint. Als Themen können Wirtschaft, Politik und vor allem „Lebensart“ ausgemacht werden.

Fügt man dem Modell ein weiteres Ressort hinzu, so wird das Bild in fast allen bisher gemachten Beobachtungen bestätigt. Zwar wächst die Differenz zwischen Minimum und Maximum auf über 39 Prozent, das Maximum an sich sinkt aber zugleich auf unter 74 Prozent. Dieser Trend lässt sich mit steigender Modellgröße weiter fortschreiben. Bei dreizehn Themen schließlich ist das Maximum bei 55,24 Prozent mit einer Differenz von 29 Prozent. Vergleicht man dies mit anderen Ressorts wie z.B. dem Ressort „Kultur“ mit einem Maximum von 90,6 Prozent und einer Differenz von 77 Prozent, so kann die Abnahme der Maximalwerte nicht durch die Zunahme der Themen begründet werden. Bei dem Ressort „Unzugeordnet“ lassen sich ähnliche Werte messen wie bei dem Ressort „Die Zeit“. Auf Grundlage der Daten kann also nicht davon ausgegangen werden, dass es sich bei den Ressorts „Die Zeit“ und „Unzugeordnet“ um Ressorts mit eigener Themenstruktur handelt. Häufig ähneln die Inhalte der Dokumente aus dem Ressort „Die Zeit“ den Themen der Gebiete Lebensart, Politik und Kultur. Bei „Unzugeordnet“ handelt es sich dagegen meist um Beiträge aus den Bereichen „Karriere“, „Wirtschaft“ und „Lebensart“.

Zwei Hinweise seien an dieser Stelle noch angemerkt. Mit wachsender Themenzahl entwickelten sich zwei LDA Themen mit einem wirtschaftlichen Hintergrund. Dies führte dazu, dass bei dreizehn Themen für die Dokumente des Wirtschaftsressorts zwei Maximalwerte mit einer Größe unter 80 Prozent vorlagen, diese aber beide eine Differenz von über 50 Prozent zum Minimum hatten. Die Bedingung, dass der Abstand mehr als 50 Prozent zwischen Minimalwert und Maximalwert betragen muss, scheint für die Analyse also die wichtigere zu sein.

Tabelle 2 Vergleich Ressorts Sport für 11 und 13 Themen (die Nummern stehen für die ID im Vokabular)

Sport 1 11 Themen	Sport 2 11 Themen	Sport 13 Themen
92%	69.63%	93%
201353:Frau	294340:Jahr	585705:Spiel
316067:Kind	592276:Staat	294340:Jahr
294340:Jahr	200431:Frage	588534:Sport
387628:Mann	357442:Land	585918:Spieler
35166:Arzt	727049:Zeit	618342:Tag
399581:Mensch	476096:Politik	727049:Zeit
618342:Tag	457761:Partei	727944:Zeitung
421292:Mutter	99552:Bundesrepublik	387774:Mannschaft
727049:Zeit	703385:Welt	387628:Mann
585705:Spiel	227796:Gesellschaft	473943:Platz
588534:Sport	485417:Problem	402250:Meter
426586:Name	290122:Interesse	263793:Herr
29756:Arbeit	383146:Macht	407887:Minute
585918:Spieler	721876:Wort	160604:Erfolg

Für den historischen Anwender ist ferner eine zweite Beobachtung interessant. Bildet man nur elf Themen, so bildet sich das Ressort Sport noch nicht eigenständig aus, sondern geht eine Verbindung

mit anderen Ressorts ein. Das passendste LDA Topic bildet dabei das Topic, dass mehr dem Ressort Lebensart zugeschrieben werden kann, wobei Topwörter wie Spiel und Sport auf das Ressort Sport verweisen. Für weitere Analysen interessanter ist jedoch das zweite LDA Thema, auch wenn es knapp die Kriterien der Faustformel nicht erfüllt. Hierbei handelt es sich um Thema, dessen Vokabular mehr den Gebieten Außenpolitik, Macht und Interessenskonflikte zuzuordnen ist. Ohne einen Artikel gelesen zu haben, kann jedoch durch LDA festgestellt werden, dass es sich scheinbar in circa 70 Prozent der Artikel aus dem Ressort Sport um Artikel handelt, in denen ein Vokabular genutzt wird, das auch in Artikeln über politische Sachverhalte verwendet wird (vgl. Tabelle 2).

Szenario 2: Themenüberblick

In dem zweiten Anwendungsszenario soll die Arbeit mit HDP veranschaulicht werden. Inspiration für die zu bearbeitende Aufgabe bietet eine erweiterte Vision David Bleis bezüglich der Arbeit mit wahrscheinlichkeitsbasierten Topic Modeling Anwendungen, in der er Topic Modeling als eine Vorstufe des Distant Readings vermarktet.[98] Es soll daher untersucht werden, inwieweit Topic Modeling tatsächlich als ein Werkzeug des Distant Readings eingesetzt werden kann und Informationen über die Inhalte eines nicht erschlossenen Korpus mit dieser Anwendung gewonnen werden können. Als Testkorpus dient erneut der bereits im Szenario zuvor betrachtete Korpus aus 20 Jahrgängen der Wochenzeitung DIE ZEIT. Wie bereits bei Szenario 1 wurden alle weiterführenden Berechnungen über R durchgeführt. Das dabei erstellte R Skript findet sich unter Szenario2.R im Ordner „Results Szenario“ des Repository dieser Arbeit.

Versuchsaufbau

Wie bei dem vorherigen Versuch werden zwei unterschiedliche Stichprobengrößen verwendet. Da vor diesem Versuch auf keine Informationen bezüglich eventueller Parametergrößen zurückgegriffen werden kann und ferner keinerlei Informationen über potentielle Abbruchbedingungen bei der Berechnung der Modelle vorliegen, sollen diese Informationen zunächst für einen einzelnen Jahrgang mithilfe eines unendlich langen Probedurchlaufs ermittelt werden, bei dem auch die Hyperparameter α und γ angepasst werden. Auf Grundlage der gemachten Beobachtungen kann dann ein einheitlicher Versuchsaufbau für beide Modelle erfolgen. Wie bereits in Szenario 1 wird auf das Festlegen fester Startwerte verzichtet, da dies durch die unterschiedlich großen Stichprobengrößen keinen Mehrwert bezüglich der Vergleichbarkeit bedeuten würde. Nach erfolgten Training und Test werden die unterschiedlichen Modelle nach der Größe der gefundenen Themen geordnet und das Modell mit den meisten Themen als Referenz für die weiteren Untersuchungen genutzt.

Durchführung

Für die Vorbetrachtung wurde der Jahrgang 1979 gewählt, da dieser den Median des Korpus darstellt und möglichst genug thematische Verbindungen in die Vergangenheit beinhalte sollte, aber auch ausreichend Themen mit einer Verbindung zu späteren Ausgaben. Für 300 Iterationen benötigte HDP ca. 12 min. Danach wurde der unendliche Durchlauf abgebrochen, da sich die Likelihood Werte nicht mehr spürbar besserten, vielmehr war der Bestwert bereits bei Iteration 226 erreicht worden. Die Werte

[98] Vgl. Blei, Probabilisitc Topic Models, S. 77.

dieses Bestwertes wurden nun als die Startwerte für einen weiteren Probelauf festgelegt und wie bereits bei LDA wurde die Optimierung nun deaktiviert, um Rechenkapazitäten einzusparen. Ziel des neuen Testlaufes war es, zu untersuchen, wie HDP auf die festgelegten Werte reagiert und ob ein Wert für eine maximale Zahl von Iterationen ermittelt werden kann. Leider konnte HDP bei festen Hyperparametern nicht an die Werte des ersten Durchlaufs anknüpfen, dennoch waren die Werte für die ermittelte Paarung besser als die Werte für eine zweite Paarung in einem anderen Vergleichstestlauf, bei dem die Hyperparameter gewählt wurden, bei denen die durchschnittlichen Likelihoods erstmals in die Zielregion (>-9,3) gelangten.

Da sich die Auswahl eines geeigneten Endwertes auf Grundlage objektiver Einschätzungen als unmöglich erschien, wurde folgender Kompromiss gewählt: Für beide Stichprobengrößen wurde jeweils ein Modell berechnet, bei dem die Zeiten von den LDA Berechnungen genutzt wurden. Zusätzlich wurde für jede Stichprobengröße jeweils ein zweites Modell berechnet, bei dem die Zahl der maximalen Iterationen auf 500 festgesetzt wurde. Dadurch erhalten wir insgesamt vier Modelle. Ziel dieser beiden Ansätze war die Bewertung des Nutzens einer voraussichtlich deutlich längeren Bearbeitungszeit bei 500 Iterationen gegenüber den kürzeren Berechnungszeiten bei den zeitlichen Abbruchbedingungen. Um die Berechnungen nicht durch unnötige Festplattenaktivität zu verzögern, wurde in allen Durchläufen auf das Erstellen von Zwischenergebnissen verzichtet (*--save_lag -1*). Die Zahl an 500 Iterationen impliziert eine möglichst starke Annäherung an ein eventuelles Optimum. Da für die Vorbetrachtung bereits nach 226 Iterationen erreicht wurde, soll mit einer mehr als doppelt so großen Zahl an Iterationen sich das Modell möglichst nah an einen potentiell optimalen, festen Wert annähern.

HDP benötigte im Trainingsmodus für die Berechnung der 500 Iterationen im ersten Szenario 112 Minuten und im zweiten Szenario 53 Minuten. Da es sich von der Zahl der Dokumente her bei der ersten Stichprobe um eine deutlich größere Stichprobe handelt als bei der zweiten Stichprobe, erscheint der zusätzliche Zeitaufwand für die Menge an zusätzlichen Dokumenten gering. Tatsächlich kann den Log Dateien entnommen werden, dass die Iterationsdauer mehr von der Zahl der gefundenen Themen abhängig erscheint als von der Zahl der zu durchlaufenden Dokumente pro Iteration. Da für die erste Stichprobe mit 349 Themen im Vergleich zu 212 Themen für die zweite Stichprobe deutlich mehr Themen gefunden wurden, ist es anzunehmen, dass die längere Dauer der Berechnungen hauptsächlich mit der Zahl an Themen korreliert. Zieht man zusätzlich die Ergebnisse der zeitgesteuerten Durchläufe hinzu, lässt sich zudem eine Korrelation von gefundenen Themen und der Anzahl an Iterationen ausmachen. Je mehr Iterationen durchlaufen werden, desto größer wird die Zahl an Themen im Modell. Eine längere Berechnungsdauer hängt daher vor allem indirekt mit der Größe der Stichprobe zusammen, indem eine größere Stichprobe mehr zu berücksichtigende Themen impliziert.

Ergebnisse

Auch im Testbetrieb korreliert die Menge an gefundenen Themen sehr stark mit der Anzahl an Iterationen, die während des Tests durchlaufen werden. In beiden Fällen, in der die Zahl der Iterationen das Abbruchkriterium war, wurden über einhundert Themen mehr als bei den nach einem festen Zeitpunkt abbrechenden Modellen gefunden. Da der Testbetrieb im Gegensatz zum Testmodus bei LDA

nahezu vollständig dem Trainingsmodus gleicht, musste allerdings aus zeitökonomischen Gründen die Zahl an Iterationen für das Training auf 100 reduziert werden, da sich zunächst die bis zu 10fach größere Zahl an Dokumenten der Grundgesamtheit in der Bearbeitungszeit bemerkbar machte und im fortlaufenden Betrieb die Themenzahl auf über 600 Themen stieg.

Die Analyse der Auswertungen von TopModHis ergab je nach Szenario einen Hierarchiebaum mit einer Tiefe von bis zu 25 Elementen. Dies entspricht mehr als 100.000 unterschiedlichen Themenkombinationen, bestehend aus bis weit über 600 unterschiedlichen Einzelthemen, die HDP für die 20 Jahrgänge ermitteln konnte. Da diese Zahl an Themen nicht vollständig im Rahmen dieser Arbeit betrachtet werden kann, soll der Nutzen der Ergebnisse anhand eines einzelnen Beispiels veranschaulicht werden.

Unter Berücksichtigung der Vorgabe, das Modell mit der größten Themenzahl als Referenz zu wählen, wurde das Modell auf Grundlage von zwei Stichproben in Verbindung mit einer vorgegebenen Zahl an Iterationen ausgewählt. Als interessantes Thema fiel bei der Durchsicht der Topwortlisten Topic 6 auf, in dem Begriffe rund um Computer, Programmierer und andere digitale Themen zu finden sind. Drei Fragen sollen betrachtet werden:

1. Wie sehen die vereinten Elemente aus, in denen Thema 6 vorkommt?
2. Wie ist die Qualität der Themen, d.h. entsprechen die Dokumente den Erwartungen?
3. Gibt es ähnliche Topics in den anderen Modellen?

Da die Vereinigung aller Themen, die das betrachtete Topic 6 enthalten, zu einem Modell mit über 2500 Themen führt, kann nur eine Stichprobe der erstellten Themen betrachtet werden. Wie bereits bei der Ermittlung der Stichproben zur Modellerstellung werden fünf per Zufallsgenerator ermittelte Themen genutzt. Die gezogenen Themen werden über die Exportfunktion von TopModHis aus dem Modell exportiert und in einem leeren, neuerstellten Modell zusammengeführt, damit dann die Topwortlisten erstellt werden können. Bei der Betrachtung der Topwortlisten fällt auf, dass die Begriffe aus den oberen Ebenen die Ergebnisse dominieren. Zwar lässt sich eine Beeinflussung durch die kleineren Ebenen ausmachen, aber im direkten Vergleich der Listen wirken viele Begriffe redundant. Das reine Aufaddieren der Ergebnisse erlaubt also noch keine gute Aussage über den Inhalt der Themen. Hier müsste in Zukunft eine andere Strategie gefunden werden, wie z.B. die Normalisierung der Ergebnisse der einzelnen Unterthemen oder das simple Zählen aller Vorkommnisse von Begriffen über alle Unterthemen hinweg. Auch eine Gewichtung der Begriffe nach der Ebene des Untertopics wäre denkbar, könnte aber zu einer Überbetonung der kleineren Thermen führen und damit den Sachverhalt lediglich umkehren. Aktuell bietet das Nebeneinanderstellen der Topwörter der verschiedenen Einzelthemen eine bessere Übersicht über die in einem Dokument zu erwartenden Themen.

In einer Stichprobe von fünf per Zufallsgenerator gewählten Themen konnte jedem Thema ein Dokument zugeordnet werden. Dabei gehörte der Inhalt von drei Dokumenten tatsächlich eindeutig zum Themengebiet der Digitalisierung. Zu dem vierten Dokument wurde Topic 6 zugeordnet, da in einer Reihe von Zitaten auch ein Professor der Informatik zu Wort kam. Bei dem letzten Dokument handelte es sich um das Porträt eines Politikers, weshalb Begriffe wie Programm oder Entwicklung vorkamen, aber in ihrer politischen Bedeutung genutzt wurden.

Tabelle 3 Die ersten 10 Topwörter der 5 vereinten Topics

Topic_0	Topic_1	Topic_2	Topic_3	Topic_4
294340:Jahr	294340:Jahr	294340:Jahr	294340:Jahr	727049:Zeit
491245:Prozent	727049:Zeit	491245:Prozent	491245:Prozent	294340:Jahr
389111:Mark	491245:Prozent	727049:Zeit	389111:Mark	201353:Frau
727049:Zeit	389111:Mark	389111:Mark	405100:Million	399581:Mensch
405100:Million	357442:Land	399581:Mensch	200431:Frage	387628:Mann
200431:Frage	405100:Million	405100:Million	201353:Frau	357442:Land
99552:Bundesrepublik	399581:Mensch	200431:Frage	703385:Welt	316067:Kind
64859:Beispiel	200431:Frage	201353:Frau	727049:Zeit	362956:Leben
703385:Welt	703385:Welt	357442:Land	357442:Land	96418:Buch
592276:Staat	201353:Frau	703385:Welt	618342:Tag	618342:Tag

Tabelle 4 Topwörter Artikel Rangierbahnhof, ausgewählte Kategorien

Topic_94	Topic_13	Topic_79	Topic_6
294340:Jahr	389111:Mark	402250:Meter	110400:Computer
491245:Prozent	276158:Hotel	315936:Kilometer	391403:Maschine
389111:Mark	618342:Tag	619560:Tal	487485:Programm
405100:Million	541980:Schiff	547903:Schnee	116561:Datum
727049:Zeit	638464:Tourist	68115:Berg	503627:Rechner
657040:Unternehmen	196001:Flugzeug	450214:Ort	623353:Technik
99552:Bundesrepublik	287845:Insel	618342:Tag	116408:Datenverarbeitung
404862:Milliarde	718569:Woche	714589:Winter	80586:Bildschirm
357442:Land	481221:Preis	471361:Piste	248063:Hacker
219407:Geld	511436:Reise	276158:Hotel	150454:Elektronik

Tabelle 5 Topwörter Artikel Radartechnik 3 Tage Krieg, ausgewählte Kategorien

Topic_20	Topic_13	Topic_32	Topic_6
357442:Land	389111:Mark	19414:Amerikaner	110400:Computer
592276:Staat	276158:Hotel	579957:Sowjet	391403:Maschine
507716:Regierung	618342:Tag	103911:Carter	487485:Programm
294340:Jahr	541980:Schiff	691146:Waffe	116561:Datum
344487:Krieg	638464:Tourist	99552:Bundesrepublik	503627:Rechner
618342:Tag	196001:Flugzeug	100113:Bundeswehr	623353:Technik
718569:Woche	287845:Insel	332930:Kongress	116408:Datenverarbeitung
683502:Volk	718569:Woche	497993:Rakete	80586:Bildschirm
222257:General	481221:Preis	668999:Verhandlung	248063:Hacker
575864:Soldat	511436:Reise	646011:Truppe	150454:Elektronik

Besonders eindrucksvoll ist der Vergleich der verschiedenen Unterthemen mit den tatsächlichen Inhalten der Dokumente. Wurde in einem Artikel über den neuartigen Einsatz von modernen Luftüberwachungssystemen im Drei-Tage Krieg berichtet, so lassen sich in den unterschiedlichen

Unterthemen die Listen für Themen wie Krieg, Außenpolitik, Luftfahrt und ähnlichen Themengebiete finden. Wird dagegen über den neuen Rangierbahnhof bei Hamburg berichtet, lassen sich stärker Themen wie Deutschland, Handel, Verkehr aber auch Gebirge finden, da der neue Bahnhof die Nord-Süd Verbindung beschleunigt und für den deutschen Süden rund um München die Alpen ein häufiges Thema sind (vgl. Tabelle 4 und Tabelle 5).

Durchsucht man die anderen Modelle nach dem Wort Computer, kommen zwar Kategorien heraus, in denen auch Begriffe wie Hacker oder Computergeneration enthalten sind, die aber auch viel stärker als bei dem betrachteten Modell themenferne Begriffe enthalten. So befasst sich ein Thema eindeutig mit den Auseinandersetzungen von Schachgroßmeistern mit Computerprogrammen und ein anderes Thema enthält Begriffe von Reisen und Programmierung gleichermaßen. Dies bedeutet, dass bei einer Suche nach Computern und der Digitalisierung der Gesellschaft in den anderen Modellen zwar passende Themen ermittelt werden können, aber mit mehr unerwünschtem Beiwerk zu rechnen ist, da sich Themen überschneiden. Andererseits könnte man allein die festgestellten Verbindungen von Digitalisierung und (Schach-)Sport sowie von Digitalisierung und Verkehr zum Aufhänger für eine tiefergehende historische Analyse nutzen.

Anmerkung

Es muss betont werden, dass eine gezielte Suche nach Themen immer effektiver sein wird als das Trainieren von Topic Models, sofern der Gegenstand der Suche bereits bekannt ist. Möchte man jedoch einen unbekannten Satz von Daten erkunden, wird dem Nutzer mit HDP ein mächtiges Werkzeug für diese Zwecke zur Verfügung gestellt. LDA hingegen ist für das Auffinden bekannter Strukturen in unerschlossenen Datensätzen ein sehr nützliches Werkzeug. Daher wäre es für die Zukunft wünschenswert, wenn man die Modelle von HDP in LDA Modelle übertragen könnte, um in anderen Korpora nach den von HDP ermittelten Strukturen suchen zu können.

Fazit und Ausblick

Eine Begleiterscheinung der stetig voranschreitenden Digitalisierung der Gesellschaft ist eine mit klassischen Methoden des Lesens nicht mehr einsehbare Zahl an Publikationen. Mittels Methoden des Distant Readings, d.h. der automatisierten Auswertung digitaler Dokumente, sollen Computer dem interessierten Nutzer aus ihm unbekannten Korpora Informationen extrahieren und aufbereiten. Eine Möglichkeit dieser Formen von Algorithmen der Informationsaufbereitung bilden die Verfahren des Topic Modelings. Bei diesen Verfahren werden Dokumente in Abhängigkeit ihrer Inhalte in Gruppen einsortiert. Dokumente mit ähnlichen Inhalten finden sich hierbei in gleichen Gruppen wieder. Dadurch ermöglichen Topic Modeling Verfahren es dem Anwender, den Inhalt und die Zugehörigkeit von großen Mengen von Dokumenten besser überblicken zu können. Da auch in der historischen Forschung die Zahl an digital vorliegenden Quellen stetig wächst, war es das Ziel dieser Arbeit, das Potential einer Nutzung dieser Anwendungen in der historischen Forschung zu betrachten. In Ermangelung einer geeigneten Vorlage wurde hierzu zunächst ein Konzept erarbeitet, wie zwei Anwendungen aus dem Bereich der wahrscheinlichkeitsbasierten Topic Modeling Verfahren für die Arbeit im Bereich der Digital Histories vorgestellt werden können. Kern der Gedanken war die gegenwärtige Suche nach Formen

von Publikationen, die einen Mehrwert für traditionelle Vertreter des Faches bedeuten und zugleich der Anforderung nach Zitierfähigkeit und Reputation genügen. Als Vorbild wurden die in der Informatik üblichen Begleitpaper genommen, in denen die theoretischen Grundlagen der vorgestellten Algorithmen erläutert werden und wichtige Kennzahlen präsentiert werden. Anders als in den Papern der Informatik musste allerdings für diese Arbeit eine diversere Interessensgruppe mit unterschiedlichen fachlichen Hintergrund angenommen werden. Dabei wurden sowohl Anwender berücksichtigt, die eine bloße Einführung in die Verwendung der Anwendung wünschen, als auch Vertreter der digitalen geisteswissenschaftlichen Disziplinen, die sich für eine Diskussion der vorgestellten Anwendungen und Verfahren interessieren. Einen besonderen Stellenwert erhielt dabei die Präsentation der theoretischen Konzeption hinter den Anwendungen verbunden mit einer erstmals im Rahmen dieser Disziplin vorgenommenen Diskussion über alternative Ansätze. Dies soll als Grundlage für einen professionsinternen Diskurs genutzt werden und kann als ein Ansatz einer möglichen neuen Form der Algorithmuskritik verwendet werden.

Die theoretische Betrachtung der Algorithmen erfolgte unter einem praxisorientierten Gesichtspunkt. Erklärt wurden vor allem die Variablen und Verfahren, die auch in den Quelltexten der Anwendungen verwendet wurden. Die Anwendung LDA-C orientierte sich bei der Struktur der Quelltexte dabei deutlich stärker an dem theoretischen Modell, wohingegen für das Verständnis aller Parameter der HDP-faster Anwendung eine ganze Reihe von Veröffentlichungen gesichtet und zusammengetragen werden mussten. Am Ende konnte dem Leser aber eine detaillierte Beschreibung aller Möglichkeiten und der verschiedenen Auswirkungen von Parameteränderungen vorgestellt werden.

Zusätzlich zu der theoretischen Betrachtung der Verfahren und ihrer Anwendung wurde parallel zu dieser Arbeit eine Eigenentwicklung erstellt. Es handelt sich dabei um eine Art Werkzeugsammlung, die sowohl eine Ergebnisinterpretation erleichtern soll, indem häufig durchzuführende Arbeitsschritte automatisiert wurden, als auch eine Bearbeitung der meist sehr großen Modelle ermöglicht. Trotz der unterschiedlichen Ausgabeformate von LDA-C und HDP-faster wurde hierbei ein Werkzeug für beide Anwendungen gemeinsam bereitgestellt, womit der Nutzer auch bei unterschiedlichen Anwendungsszenarien sich nicht in seiner Arbeitsweise umstellen muss. Wie die Arbeit mit der Anwendung „**Top**ic **Mod**eling für **His**toriker“ (kurz TopModHis) erfolgt, wurde hierbei in Form einer Anleitung aufgezeigt.

In den beiden praktischen Szenarien wurden abschließend die für die beiden Anwendungen jeweils vorgesehenen Anwendungsfelder noch einmal verdeutlicht. Als Datengrundlage diente in beiden Fällen ein vom Autor selber erstellter Korpus aus Artikeln der deutschen Wochenzeitung DIE ZEIT aus den Jahren 1969 bis 1989. Mit LDA-C wurde die Beschaffenheit verschiedener Ressorts im Korpus untersucht, wobei auf Grundlage bekannter Ressortgrößen mehrere Modelle mit einer wechselnden Zahl von Themen erstellt wurden und die Auswirkungen der unterschiedlichen Themenzahl auf die Verteilung der den betrachteten Ressorts zugeteilten Dokumente analysiert wurde. Hierbei konnte gezeigt werden, dass die durch LDA-C gefundenen Gruppenzuordnungen den eigenen Erwartungen entsprechen. Ferner konnte der Vorteil genutzt werden, dass LDA im Vergleich zu distanzbasierten Anwendungen die Wahrscheinlichkeit aller Topiczuweisungen ausgibt, indem z.B. die politische Dimension in Artikeln des Sportressorts aufgezeigt werden konnte.

Anhand des gleichen Korpus wurde im Anschluss mittels HDP-faster eine Inhaltsanalyse des Korpus durchgeführt. Da das vollständige Ergebnis eine Hierarchie mit weit über 100.000 Themen beinhaltete, wurde die weiterführende Arbeit mit den Ergebnissen anhand eines einzelnen, prägnanten Themengebiets verdeutlicht. Hierbei handelte es sich um ein Gebiet, das sich aus Begriffen rund um die Digitalisierung zusammensetzte. Da auch dieses Gebiet noch rund 2500 Themen umfasste, wurden aus der Menge der über 2500 Themen eine Stichprobe aus fünf Themen gezogen und die Inhalte der dieser Themenkonstellation zugeordneten Artikel mit den einzelnen Unterthemen verglichen. Es konnte festgestellt werden, dass das Vokabular der einzelnen Themen bereits eine gute Übersicht über den zu erwartenden Artikelinhalt gibt.

Zudem wurde für beide Anwendungen untersucht, wie verschiedene Strategien der Modellerstellung aussehen könnten und wie eine Bewertung unterschiedlich erstellter Modelle vorgenommen werden kann. Bei beiden Anwendungen wurde hierzu als Trainingskorpus sowohl eine Stichprobe aus fünf zufällig gezogenen Jahrgängen der Zeitung wie auch eine selbstgewählte Stichprobe aus zwei Jahrgängen genutzt. Für LDA-C wurde dann als ein Vergleichswert zur Unterscheidung der Qualität von Modellen der durchschnittliche Likelihood verwendet.

Bei HDP-faster wurden ebenfalls die durchschnittlichen Likelihoods als ein Qualitätsmaß genutzt, in diesem Fall jedoch zur Wahl geeigneter Hyperparameter über alle Stichproben hinweg. Interessanterweise wurde am Ende bei der Anwendung des aus zwei Stichproben bestehenden Modells die größte Zahl an Themen gefunden.

Besonders die Analyse der Ergebnisse von HDP-faster benötigen eine Reihe von weiteren Analyseschritten und die Interpretation der Hierarchie konnte auch in dieser Arbeit nicht abschließend zusammengetragen werden. Die Informationen entsprechen aber im Grunde bereits jetzt der versprochenen Vision von Blei mit seiner sich stetig vertiefenden Themenhierarchien.

Neben einer besseren Analyse der Hierarchien sollte in folgenden Arbeiten unbedingt eine Diskussion über eine bessere Visualisierung der verschiedenen Ergebnisse besonders mit Blick auf statische Veröffentlichungen wie Bücher erfolgen. Aktuell müssen die Ergebnisse stark gekürzt werden, um dem Leser ein noch zu überblickendes Bild von den Ergebnissen geben zu können. Bei einer größeren Gewichtung der Ergebnisse des Anwendungsteils für die Aussage der Publikation wird diese Form der Präsentation aber mit hoher Wahrscheinlichkeit nicht mehr genügen. Eine Diskussion über zukünftige Formen der Präsentation und Ergebnisbereitstellung wäre daher sowohl für den digitalen Forscher im Rahmen seiner begleitenden Publikationen wichtig, aber auch für den klassisch arbeitenden Kollegen, um die Grundlagen seiner Interpretationen in geeigneter Weise darstellen zu können.

Des Weiteren wurde in dieser Arbeit in beiden Szenarien keine Optimierung der Hyperparameter beim Training der Modelle eingesetzt, sondern auf Werte von vorhergehenden Modelltrainings mit kleineren Stichproben des Korpus zurückgegriffen. Interessant für zukünftige Veröffentlichungen könnte es jedoch sein, ob der Hyperparameter α als ein Qualitätsmerkmal der Modelle genutzt werden könnte. Da besonders kleine α's für Verteilungen mit einzelnen Extremen stehen und große α's gleichmäßig verteilte Themengruppen bedeuten, könnte je nach Anforderung eine unterschiedliche Ausprägung von α für die Bevorzugung eines bestimmten Modells gegenüber anderen Modellen sprechen.

Zuletzt wäre eine Portierung von HDP Modellen in LDA Modelle wünschenswert, um den einzelnen Anforderungen der beiden vorgestellten Anwendungen gerecht werden zu können. Da die Modelle von HDP auch im Testmodus sich weiterentwickeln, können die ermittelten Modelle nicht an anderen Korpora getestet werden. Das LDA Modell ist hingegen nach dem Training nicht mehr veränderlich und kann auf unterschiedliche Sammlungen gleichermaßen angewendet werden. Mit einer Portierung der Modelle könnten daher die mittels HDP erstellten Strukturen durch LDA geprüft und an anderen Korpora getestet werden.

Literaturverzeichnis

Alvarado, Rafael C., The Digital Humanities Situation, in: Matthew K. Gold (Hrsg.), Debates in the digital humanities, Minneapolis, 2012, S. 50–55.

Ambrosius, Gerold, Plumpe, Werner und Tilly, Richard, Wirtschaftsgeschichte als interdisziplinäres Fach, in: Gerold Ambrosius, Dietmar Petzina und Werner Plumpe (Hrsg.), Moderne Wirtschaftsgeschichte. Eine Einführung für Historiker und Ökonomen, München, 2006, S. 9–38.

Berlin Brandenburgische Akademie der WIssenschaften, D*/DTA: DiaCollo, online verfügbar unter: http://kaskade.dwds.de/dstar/dta/diacollo/. Zuletzt geprüft am: 11.08.2019.

Blei, David M., Probabilisitc Topic Models, in: Communications of the ACM 55, 2012, Nr: 4, S. 77–84, online verfügbar unter: https://cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext. Zuletzt geprüft am: 01.03.2019.

Blei, David M., Topic Modeling and Digital Humanities, 2012. Erschienen am 16.06.2019, online verfügbar unter: http://journalofdigitalhumanities.org/2-1/topic-modeling-and-digital-humanities-by-david-m-blei/.

Blei, David M., blei-lab/lda-c. Erschienen am 09.06.2016, online verfügbar unter: https://github.com/blei-lab/lda-c. Zuletzt geprüft am: 14.07.2019.

Blei, David M., Griffiths, Thomas L. und Jordan, Michael I., The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, in: Journal of the ACM 57, 2010, Nr: 2, S. 1–30. online verfügbar unter: doi.org/10.1145/1667053.1667056.

Blei, David M., Jordan, Michael I. und Griffiths, Thomas L. et al., Hierarchical Topic Models and the Nested Chinese Restaurant Process, Proceedings of the 16th International Conference on Neural Information Processing Systems, S. 17–24, online verfügbar unter: http://www.cs.columbia.edu/~blei/papers/BleiGriffithsJordanTenenbaum2003.pdf. Zuletzt geprüft am: 28.02.2019.

Blei, David M., Ng, Andrew Y. und Jordan, Michael I., Latent Dirichlet Allocation, in: Journal of Machine Learning Research 3, 2003, S. 993–1022, online verfügbar unter: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf. Zuletzt geprüft am: 01.03.2019.

Bol'shev, L. N., Dirichlet distribution. Erschienen am 07.02.2011, online verfügbar unter: https://www.encyclopediaofmath.org/index.php/Dirichlet_distribution. Zuletzt geprüft am: 13.07.2019.

Busa, R., The annals of humanities computing: The index Thomisticus: Computers and the Humanities, in: Comput Hum 14, 1980, Nr: 2, S. 83–90. online verfügbar unter: doi.org/10.1007/BF02403798.

CLARIN-D, WebLicht, online verfügbar unter: https://weblicht.sfs.uni-tuebingen.de/weblicht/. Zuletzt geprüft am: 11.08.2019.

DARIAH-DE, Topics Explorer, online verfügbar unter: https://dariah-de.github.io/TopicsExplorer/. Zuletzt geprüft am: 05.07.2019.

DARIAH-DE, Handbuch Digital Humanities: Anwendungen, Forschungsdaten und Projekte, 2015, online verfügbar unter: https://handbuch.tib.eu/w/images/2/2c/DH-Handbuch.pdf. Zuletzt geprüft am: 04.07.2019.

Dempster, Arthur P., Laird, Nan M. und Rubin, Donald B., Maximum likelihood from incomplete data via the EM algorithm, in: Journal of the Royal Statistical Society: Series B (Methodological) 39, 1977, Nr: 1, S. 1–22, online verfügbar unter: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

Fiedler, Maik, HT 2018: Für Skeptiker und Enthusiasten: Was ist und zu welchem Ende nutzt das ›Digitale‹ in den Geschichtswissenschaften? Erschienen am 16.11.2018. in: H-Soz-Kult (Hrsg.), online verfügbar unter: https://www.hsozkult.de/conferencereport/id/tagungsberichte-7960. Zuletzt geprüft am: 10.07.2019.

Fitzpatrick, Kathleen, The Humanities, Done Digitally, in: Matthew K. Gold (Hrsg.), Debates in the digital humanities, Minneapolis, 2012, S. 12–15.

Google Inc., Google Code Archive - Word2Vec, online verfügbar unter: https://code.google.com/archive/p/word2vec/. Zuletzt geprüft am: 17.07.2019.

Griffiths, Thomas L. und Steyvers, Mark, A probabilistic approach to semantic representation, Proceedings of the annual meeting of the cognitive science society, 2002.

Hohls, Rüdiger, Digital Humanities und digitale Geschichtswissenschaften, in: Rüdiger Hohls, Thomas Meyer und Wilfried Enderle, et al. (Hrsg.), Clio-Guide. Ein Handbuch zu digitalen Ressourcen für die Geschichtswissenschaften, 2018, A.1-1 - A.1-34.

Jank, Gerhard und Jongen, Hubertus Theodorus, Höhere Mathematik für Maschinenbauer: Skript zur Vorlesung, 1999.

Jordan, Michael I., Ghahramani, Zoubin und Jaakkola, Tommi S. et al., An Introduction to Variational Methodsfor Graphical Models, in: Machine Learning 37, 1999, Nr: 2, S. 183–233. online verfügbar unter: doi.org/10.1023/A:1007665907178.

Last, Mark, Stoliar, Maxim und Friedman, Menahem, Clustering-Based Classification of Document Streams with Active Learning, in: Abraham Kandel, Horst Bunke und Mark Last (Hrsg.), Data mining in time series and streaming databases, Singapore, 2018, S. 92–117.

Leroi, Armand Marie, Cicero zählen: Algorithmus oder Kritik? Plädoyer für eine universelle Kulturtheorie., in: Süddeutsche Zeitung, Nr. 54 vom 06. 3. 2015.

Leskovec, Jurij, Rajaraman, Anand und Ullman, Jeffrey D., Mining of massive datasets, Cambridge, 2015.

Lorbeer, Boris, Kosareva, Ana und Deva, Bersant et al., Variations on the Clustering Algorithm BIRCH, in: Big Data Research 11, 2018, S. 44–53. online verfügbar unter: doi.org/10.1016/j.bdr.2017.09.002.

Lorenz, Chris, Wozu noch Theorie in der Geschichte?: Über das ambivalente Verhältnis zwischen Gesellschaftsgeschichte und Modernisierungstheorie, in: Volker Depkat, Matthias Müller und Andreas Urs Sommer (Hrsg.), Wozu Geschichte(n)? Geschichtswissenschaft und Geschichtsphilosophie im Widerstreit, Stuttgart, 2004, S. 117–143.

MacQueen, James und others, Some methods for classification and analysis of multivariate observations, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA, 1967, S. 281–297.

Meyer, Thomas, Digitale Werkzeuge, in: Rüdiger Hohls, Thomas Meyer und Wilfried Enderle, et al. (Hrsg.), Clio-Guide. Ein Handbuch zu digitalen Ressourcen für die Geschichtswissenschaften, 2018, A.2-1 - A.2-45.

Mikolov, Tomas, Chen, Kai und Corrado, Greg et al., Efficient Estimation of Word Representations in Vector Space, 1/16/2013, online verfügbar unter: http://arxiv.org/pdf/1301.3781v3. Zuletzt geprüft am: 11.07.2019.

Müller, Andreas und Purschwitz, Anne, HT 2018: Forschungsdaten: Rechtliche Herausforderungen und wissenschaftliche Reputation. Forschungsdatenmanagement als Bestandteil einer neuen Wissenschaftskultur. Erschienen am 30.11.2018. in: H-Soz-Kult (Hrsg.), online verfügbar unter: https://www.hsozkult.de/conferencereport/id/tagungsberichte-7988. Zuletzt geprüft am: 10.07.2019.

Müller, Florian, GitRepository TopModHis, online verfügbar unter: https://github.com/ImdWf/TopModHis. Zuletzt geprüft am: 14.08.2019.

Schmid, Helmut, Improvements in Part-of-Speech Tagging with an Application to German, Stuttgart, 1995, online verfügbar unter: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf. Zuletzt geprüft am: 03.12.2015.

Schmitt, Martin, HT 2018: Digital Humanities in der Analyse gespaltener Gesellschaften. Beispiele aus der Praxis. Erschienen am 07.12.2018. in: H-Soz-Kult (Hrsg.), online verfügbar unter: https://www.hsozkult.de/conferencereport/id/tagungsberichte-8009. Zuletzt geprüft am: 10.07.2019.

Staatsbibliothek zu Berlin Zeitungsabteilung, Zeitungsinformationssystem ZEFYS - Staatsbibliothek zu Berlin. Erschienen am 01.01.2018, online verfügbar unter: http://zefys.staatsbibliothek-berlin.de/ddr-presse. Zuletzt geprüft am: 11.08.2019.

Teh, Yee W., Jordan, Michael I. und Beal, Matthew J. et al., Sharing clusters among related groups: Hierarchical Dirichlet processes, Advances in neural information processing systems, Cambridge, London, 2003, S. 1385–1392, online verfügbar unter: http://papers.nips.cc/paper/2698-sharing-clusters-among-related-groups-hierarchical-dirichlet-processes.pdf. Zuletzt geprüft am: 22.06.2019.

Teh, Yee Whye, Jordan, Michael I. und Beal, Matthew J. et al., Hierarchical Dirichlet Processes, in: Journal of the American Statistical Association 101, 2006, Nr: 476, S. 1566–1581. online verfügbar unter: doi.org/10.1198/016214506000000302, online verfügbar unter: http://www.cs.columbia.edu/~blei/papers/TehJordanBealBlei2006.pdf. Zuletzt geprüft am: 22.06.2019.

Tomas Mikolov, Word2Vec. Erschienen am 17.07.2017, online verfügbar unter: https://github.com/tmikolov/word2vec. Zuletzt geprüft am: 28.02.2019.

Turney, Peter D., Learning Algorithms for Keyphrase Extraction, in: Information Retrieval 2, 2000, Nr: 4, S. 303–336. online verfügbar unter: doi.org/10.1023/A:1009976227802.

van Eijnatten, Joris, Pieters, Toine und Verheul, Jaap, Big Data for Global History: The Transformative Promise of Digital Humanities, in: BMGN - Low Countries Historical Review 128, 2013, Nr: 4, S. 55–77, online verfügbar unter: http://www.bmgn-lchr.nl/index.php/bmgn/article/view/9350.

Wang, Chong, blei-lab/hdp, 2010. Erschienen am 21.02.2017, online verfügbar unter: https://github.com/blei-lab/hdp. Zuletzt geprüft am: 14.07.2019.

Wettlaufer, Jörg, Neue Erkenntnisse durch digitalisierte Geschichtswissenschaft(en)? Zur hermeneutischen Reichweite aktueller digitaler Methoden in informationszentrierten Fächern.

Wikipedia, Dirichlet-Verteilung. Erschienen am 18.03.2018. in: Wikipedia (Hrsg.), online verfügbar unter: https://de.wikipedia.org/w/index.php?oldid=175153858. Zuletzt geprüft am: 13.07.2019.

Wilkens, Matthew, Canons, Close Reading, and the Evolution of Method, in: Matthew K. Gold (Hrsg.), Debates in the digital humanities, Minneapolis, 2012, S. 249–258.

Will, Torsten T., C++: Das umfassende Handbuch, Bonn, 2018.

Zhang, Tian, Ramakrishnan, Raghu und Livny, Miron, BIRCH: A New Data Clustering Algorithm and Its Applications, in: Data Mining and Knowledge Discovery 1, 1997, Nr: 2, S. 141–182. online verfügbar unter: doi.org/10.1023/A:1009783824328.