ZfdG

Zeitschrift für digitale Geisteswissenschaften

Artikel aus:
Zeitschrift für digitale Geisteswissenschaften

Titel:
Turing Test für das Topic Modeling. Von Menschen und Maschinen erstellte inhaltliche Analysen der Korrespondenz von Leo von Thun-Hohenstein im Vergleich

Autor/in:
Peter Andorfer

Kontakt:
peterandorfer@oeaw.ac.at

Institution:
Österreichische Akademie der Wissenschaften (OAW), Austrian Centre for Digital Humanities (ACDH)

GND:
1043833846

ORCID:
0000-0002-9575-9372

DOI des Artikels:
10.17175/2017_002

Nachweis im OPAC der Herzog August Bibliothek:
882673483

Erstveröffentlichung:

Letzte Überprüfung aller Verweise:
03.04.2017

GND-Verschlagwortung:
Automatische Klassifikation | Computergestütztes Verfahren | Mustererkennung |

Zitierweise:
Peter Andorfer: Turing Test für das Topic Modeling. Von Menschen und Maschinen erstellte inhaltliche Analysen der Korrespondenz von Leo von Thun-Hohenstein im Vergleich. In: Zeitschrift für digitale Geisteswissenschaften. 2017. PDF Format ohne Paginierung. Als text/html abrufbar unter DOI: 10.17175/2017_002.

Peter Andorfer

Turing Test für das Topic Modeling. Von Menschen und Maschinen erstellte inhaltliche Analysen der Korrespondenz von Leo von Thun-Hohenstein im Vergleich

Abstracts

Wer ist schlauer? Mensch oder Maschine? Die Antwort auf diese Frage wird seit 1950 mit Alan Turing und dem von ihm konzipierten Turing Test verknüpft. Daran anknüpfend vergleicht vorliegender Aufsatz inhaltliche Analysen eines historischen Briefwechsels, die einmal ›vom Menschen‹ mittels ›close reading‹ und anschließender Vergabe von Schlagworten und einmal ›von der Maschine‹ mittels Topic Modeling erzeugt wurden. Neben der konkreten Evaluierung des Topic Modeling Verfahrens wirft dieser Aufsatz auch die Frage auf, ob und wieweit es möglich und vertretbar ist, Methoden einzusetzen, die ohne tiefere Kenntnisse von Wahrscheinlichkeitsberechnungen und Statistik kaum noch gänzlich verstanden werden können.

Who is cleverer, man or machine? Since the 1950s, the answer to this question has been linked to Alan Turing and the Turing test he devised. This paper builds upon this foundation with its comparison of two analyses of a historical collection of correspondence: one created by humans using ›close reading‹ and the application of subject terms, and one generated by machines with the help of topic modeling. In addition to concrete evaluation of the topic modeling process, this paper investigates whether, and, if so, to what extent, it is feasible and justifiable to use methods that can hardly be understood without in-depth knowledge of probability calculations and statistics.

1. Topic Modeling, Turing Test und Fragestellung

Topic Modeling beschreibt ein Verfahren, das aus einer abgeschlossenen Textmenge eine vorher festgelegte Anzahl von Topics extrahiert. Sehr verkürzend und vereinfachend formuliert werden in mehreren Wiederholungen die Wörter eines jeden Dokuments einem bestimmten Topic zugewiesen, wobei die Zuweisung anhand statistischer Häufigkeiten und den daraus berechneten Wahrscheinlichkeiten erfolgt.

Bei den so generierten Topics oder Themen handelt es sich um Listen von Wörtern und deren jeweiligen Häufigkeiten, die in einem Topic vermehrt anzutreffen sind und so ein Topic konstituieren.

Das Thema Topic Modeling wurde im Kontext der Digital Humanities in den letzten Jahren bereits breit diskutiert. Einen guten Einstieg in die statistisch-mathematischen Hintergründe des Verfahrens bietet der Aufsatz von David Blei *Probabilistic Topic Models* aus dem Jahr 2012.[1] Ebenfalls 2012 erschien eine Ausgabe des *Journal of Digital Humanities*, welche in Gänze dem

[1] Blei 2012.

Thema Topic Modeling gewidmet war.[2] Neben Artikeln eher genereller und einführender Natur[3] finden sich darin auch konkrete Fallbeispiele, in denen geisteswissenschaftliche Fragestellungen mit Hilfe des Topic Modelings mehr oder weniger erfolgreich bearbeitet wurden.[4] Topic Modeling wird auch in einer Fülle von Blogposts vorgestellt. Die Bandbreite der Inhalte reicht dabei von umfassenden Einführungen,[5] über detaillierte Tutorials[6] bis hin zur Beantwortung der ›wahrlich essenziellen Frage‹, ob man mit Hilfe von Topic Modeling die pikanteren Passsagen in *Fifty Shades of Grey* [7] identifizieren kann, ohne deswegen gleich das ganze Buch lesen zu müssen.[8] Von besonderer Bedeutung für diesen Aufsatz ist jedoch die 2014 von Matthew L. Jockers veröffentlichte Monographie *Text analysis with R for students of literature*,[9] eine sehr stark methodisch-praktisch orientierte Ergänzung der vom selben Autor 2013 publizierten Studie *Macroanalysis: Digital Methods and Literary History*,[10] führt Jockers in *Text analysis* doch Schritt für Schritt oder besser Codezeile für Codezeile vor, wie man unter Verwendung der Programmiersprache R, Topic Modeling auf einen Textkorpus anwenden kann. Weite Teile des für diesen Artikel verwendeten Codes wurden aus *Text analysis* übernommen.

Aufgrund der Fülle vorhandener Einführungen zum Thema Topic Modeling wird hier auf weitere einführende und erläuternde Ausführungen verzichtet. Vielmehr wird hier der Versuch unternommen, zu testen, ob Topic Modeling als Werkzeug für die inhaltliche Erschließung großer Textmengen ähnlich brauchbare Ergebnisse liefert, wie dies bei einer Erschließung durch Menschen der Fall ist, welche sämtliche Texte des Korpus lesen und diese mit einem oder mehreren Schlagworten versehen. Verkürzt gesagt handelt es sich hierbei also um einen Wettstreit zwischen Mensch und Maschine, frei nach dem von Alan Turing 1950 vorgestelltem, sogenannten Turing Test.[11] Während dabei aber ein Mensch versucht zu erkennen, ob er mit seinesgleichen oder mit einer Maschine kommuniziert, steht hier dagegen die eben aufgeworfene Frage im Vordergrund, ob eine Maschine respektive ein von einem Computer angewendeter Algorithmus den menschlichen Arbeitseinsatz im Bereich der Texterschließung ersetzen oder wenigstens ergänzen oder erleichtern kann.

Anhand dieser Fallstudie soll zudem demonstriert werden, dass es prinzipiell möglich ist, Methoden oder Verfahren einzusetzen, deren mathematisch-statistischer Hintergrund nicht zur Gänze durchdrungen wurde.[12]

[2] Journal of Digital Humanities 2012.
[3] Brett 2013.
[4] Rhody 2012; Schmidt 2012.
[5] Weingart 2012.
[6] Graham et al. 2012.
[7] James 2012.
[8] Andorfer 2015a.
[9] Jockers 2014.
[10] Jockers 2013.
[11] Turing 1950.
[12] Sämtliche, für das Verfassen dieses Aufsatzes geschriebene Materialien sind im GitHub-Repository https://github.com/csae8092/topicModeling https://github.com/csae8092/topicModeling veröffentlicht und können eingesehen und vor allem auch nachgenutzt werden. Da dieser Text in einer Online-Zeitschrift und somit in digitaler Form erscheinen wird, ist es außerdem möglich, direkt auf die jeweils im Text erwähnten Ressourcen (Code, Bilder, Textdaten) zu verlinken, weshalb von einer tatsächlichen Einbindung dieser Materialien in den Text, beispielsweise in Form von ausführlichen Code-Listings, oder der Abbildung von 115 Wortwolken verzichtet wird.

2. Korpus und Datenmaterial

Bei dem für diesen Feldversuch zu analysierenden Korpus handelt es sich um einen Teil der Korrespondenz von Leo von Thun-Hohenstein (1811–1788) aus der Zeit seiner Tätigkeit als Minister für Kultus und Unterricht (1849–1860). Das Korpus besteht aus etwa 520 Briefen. Hinzu kommen noch rund 250 andere Dokumente wie Protokolle, Konzepte, Memoranden oder Gesetzesentwürfe. Der Großteil dieses Materials befindet sich in Děčín, einer Zweigstelle des Tschechischen Gebietsarchives Litoměřice.[13] Abgesehen von 38 Dokumenten sind sämtliche Texte in deutscher Sprache verfasst. In einem vom Fonds zur Förderung der wissenschaftlichen Forschung (FWF) geförderten Projekt wird diese Korrespondenz in Form einer Online-Edition sukzessive publiziert. Dafür wurden die einzelnen Dokumente transkribiert und den Richtlinien der TEI entsprechend in XML kodiert.[14]

Von gewissem Vorteil für diesen Aufsatz ist es, dass es sich bei dem für die technische Umsetzung der Online-Edition verantwortliche Person auch um den Verfasser dieser Zeilen handelt. Damit geht einerseits eine rudimentäre Vertrautheit mit dem Korpus sowie den projektorientierten Arbeitsabläufen einher, andererseits ist es dadurch auch möglich, einige der für diesen Aufsatz geschriebenen Skripte bzw. Funktionen zur Analyse des Korpus auch gleich in die Online-Edition zu integrieren. Das entsprechende xQuery-Modul kann hier eingesehen werden. Die damit ›on-the-fly‹ generierten Informationen bilden auch die Basis für die nun folgende kurze Beschreibung des Textmaterials.[15]

Zum Zeitpunkt der Niederschrift dieses Aufsatzes umfasste der von der Projektleiterin Brigitte Mazohl freigegebene Bestand 81 Dokumente, verfasst von 48 unterschiedlichen Autoren und einer Autorin, sofern die Verfasser der Texte von den Editoren ermittelt werden konnten. Während sich diese Zahlen aber noch mit ziemlich großer Gewissheit feststellen lassen und im Zweifel durch einfaches Zählen der Dokumente auch nachgeprüft werden kann, bereitet die Frage nach der Anzahl der Wörter in den Dokumenten schon größeres Kopfzerbrechen, kommt es dabei doch auf den jeweils verwendeten Tokenizer an. Für die statistische Auswertung, wie sie auf der Webpage der Online-Edition eingesehen werden kann, wurde dafür die xQuery Funktion functx:word-count verwendet, welche Wortgrenzen entsprechend der Unicode Definition von »word characters« festlegt. Die Anzahl der ›Wörter‹ (tokens) der Transkripte dieser 81 Dokumente liegt dabei bei 119.577 ›Wörtern‹ und 17.944 distinkten Wortformen (types). Dies entspricht knapp 200 Din-A4 Seiten, beschrieben in Calibri mit einfachem Zeilenabstand und in der Schriftgröße 11 Punkt. Lässt man dieselbe Textmenge von LibreOffice zählen, so erhält man ein Ergebnis von 117.939 Wörtern, analysiert man hingegen den Text mit Hilfe von voyant-org, so werden 132.062 »words« und 17.380 »unique words« gezählt. Dabei gilt es aber zu beachten, dass darin auch die vereinzelten Anmerkungen der Editoren enthalten sind, welche jedoch aufgrund des verhältnismäßig geringen Umfanges nicht herausgefiltert wurden. Die Anzahl der Wörter in Anmerkungen pro Dokument liegt nämlich nur bei knapp 24 Wörtern. Bei einer durchschnittlichen

[13] Vgl. Aichner 2015.
[14] Zu den Editionsrichtlinien vgl. Aichner 2015.
[15] Vgl. Andorfer 2015b.

Dokumentlänge von knapp 1500 Wörtern machen die Anmerkungen somit gerade einmal 1,6 Prozent aus.

Vor dem Hintergrund einer Standardabweichung von 1693,6 lassen sich alleine aus der durchschnittlichen Textlänge von 1500 Wörtern aber keine weitergehenden Rückschlüsse auf den Umfang der einzelnen Dokumente ziehen. Sinnvoller erscheint hier schon eine Verteilung der Dokumente, gestaffelt nach ihrer Wortanzahl:

Tabelle 1: Wörter pro Dokument gestaffelt

Anzahl der Wörter	Anzahl der Dokumente
max 500	19
501-1000	26
1001-2000	18
2001-3000	9
3001-4000	3
4001-	6

Wie in der Tabelle zu sehen ist, umfassen 45 der 81 Dokumente weniger als 1000 Wörter. Ein Wert, der im Kontext der Datenvorbereitung für das Topic Modeling noch eine nicht unbedeutende Rolle spielen wird.

3. Mensch: Inhaltliche Erschließung durch Vergabe von Schlagwörtern

Die Korrespondenz von Leo von Thun-Hohensteins ist aber nicht nur aufgrund ihres Umfanges und der systematischen Strukturiertheit der Daten prädestiniert für den hier vorzunehmenden Wettstreit zwischen Mensch und Maschine. Das Korpus bzw. die einzelnen Dokumente darin wurden von den Projektmitarbeitern Tanja Kraler und Christof Aichner nämlich gleich in zweierlei Weise inhaltlich erschlossen. So wurde jedes Dokument sowohl mit einem knappen Regest beschrieben als auch mit einem oder mehreren (maximal neun) Schlagwörtern versehen. Insgesamt wurden so 299 Schlagwörter vergeben, womit auf ein Dokument im Schnitt 3,7 Schlagwörter kommen. Am häufigsten – 24 mal – begegnet man dem Schlagwort »Personalfragen«, gefolgt von »Kultus« (21), »Gymnasien« (12), »Katholische Kirche« (11) und »Personalvermittlung«, das noch in zehn von 81 Texten anzutreffen ist.[16]

Schon an diesem kleinen Beispiel lässt sich erkennen, dass die vergebenen Schlagwörter teils eng miteinander in Verbindung stehen. Wollte man diese Begriffe zu einer Ontologie zusammenführen, so könnte man beispielsweise »Personalvermittlung« als einen

[16] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/schlagworte-all.html.

spezifischeren Unterbegriff zu der weiter gefassten Bezeichnung »Personalfragen« beschreiben. Ähnliches ließe sich auch beim Begriff »Kultus« bewerkstelligen. Blättert man nämlich jene 21 Dokumente durch, die allesamt unter »Kultus« subsumiert werden, so findet man darin stets weitere Schlagwörter wie: »Katholische Kirche«, »Katholikenvereine«, »Bischofsversammlung«, »Griechisch-katholische Kirche«, »Juden«, »Kirchenbau«, »Konkordat«, »Konfessionen«, »Deutschkatholiken«, »Protestanten«, »Kirchenverfassung«, »Griechisch-orthodoxe Kirche« und »Evangelische Kirche«, also stets Begriffe, die um die Themengebiete Konfessionen und religiöse Einrichtungen kreisen. Einzig bei zwei aus 21 Dokumenten wurden keine religiös-konfessionell konnotierten Schlagwörter vergeben.[17]

Wie in Gesprächen mit den Editoren in Erfahrung gebracht werden konnte, erfolgte die Vergabe der Schlagwörter in unmittelbarem Anschluss an die Kodierung eines jeweiligen Dokumentes. Im Falle bis dahin im Korpus noch nicht aufgetretener Themen mussten somit von den Editoren stets neue Schlagwörter ge- oder erfunden werden, was zwangsläufig zu einem gewachsenen und kaum reglementierten Bestand an Schlagwörtern führt. Eine allfällige Ordnung, Strukturierung oder anderweitige Kuration der Schlagwörter ist bisher nicht erfolgt, wobei dies den Editoren angesichts der begrenzten Projektmittel nicht angelastet werden kann.

Da die Gretchenfrage des Topics Modelings aber genau die Frage nach der Anzahl der Topics im Korpus ist – immerhin handelt es sich dabei um so ziemlich den einzigen Parameter, welcher dem Algorithmus übergeben werden muss – ist für das hier durchzuführende Experiment zumindest eine rudimentäre Kuration der im Korpus anzutreffenden 115 distinkten Schlagwörter unumgänglich. Denn – soviel sei schon vorweggenommen – erstellt man ein Topic Model mit tatsächlich 115 Topics, so sind diese in ihrer Zusammensetzung sehr ähnlich und lassen sich nur im Ausnahmefall semantisch sinnvoll aufladen.[18]

Die ›rudimentäre Kuration‹ besteht allerdings bloß darin, nur jene Schlagwörter beizubehalten, die wenigstens zweimal vergeben wurden. Dadurch reduziert sich die Anzahl der Topics von 115 auf 53. Ein solcher Schritt, welcher auf den ersten Blick sehr willkürlich erscheinen mag, lässt sich aber insofern rechtfertigen, als – wie bereits oben angedeutet – eine Fülle von Schlagwörtern keine völlig neuen Themenfelder eröffnet, sondern im Gegenteil bereits von anderen Schlagwörtern grob umrissene Diskurse präzisiert und konkretisiert.

4. Maschine: Inhaltliche Erschließung durch Topic Modeling

4.1 Datenbeschaffung

[17] Vgl. http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=heufler-an-thun_1850-04-13_A3-XXI-D44.xml&searchword=qwertzy sowie http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=entwurf-wirkungskreis-ministerium-fuer-kultus-und-unterricht_-o.D._A3-XXI-D84.xml&searchword=qwertzy.

[18] Die Ergebnisse eines Topic Models mit 115 Topics können hier eingesehen werden: https://github.com/csae8092/topicModeling/tree/master/results/2000_115/wordclouds.

Ein großer Reiz, den Topic Modeling auf (digitale) Geisteswissenschaftler ausübt, liegt an den geringen Ansprüchen, die das Verfahren an das zu verarbeitende Datenmaterial stellt. Ist man ausschließlich an den generierten Topics interessiert, genügt schon eine in einzelne Dokumente unterteilte Textmenge, frei von Metadaten jeglicher Art und Weise. Aber auch wenn man den Verlauf, das Vorkommen oder die Abwesenheit von Topics im Korpus verfolgen möchte, ist nicht viel mehr als eine Art Titel des jeweiligen Dokuments notwendig, wobei hierfür in all jenen Fällen, in denen das Korpus aus einer Ansammlung einzelner Dateien (z.B. .txt, .xml) besteht, schon der Dateiname ausreicht. Dies, so zumindest die persönliche Erfahrung, verleitet dazu, Topic Modeling einfach einmal auszuprobieren, um zu sehen, was dabei herauskommt.

Die Online-Edition der Korrespondenz von Leo von Thun-Hohenstein basiert auf der xml-Datenbank eXist-db, die mit einer integrierten RESTful-API ausgeliefert wird, welche einen einfachen und schnellen Zugriff auf die in der Datenbank gespeicherten Dokumente erlaubt.

Die Möglichkeiten, die bereits veröffentlichten Dateien, welche unter der URL http://thun-korrespondenz.uibk.ac.at:8080/exist/rest/db/files/thun/xml/ aufgerufen werden können, in einen Topic Modeling-Workflow einzubinden, sind vielfältig. Für dieses Projekt wurde ein Python Skript geschrieben,[19] welches die einzelnen XML-Dokumente in einem eigenen Verzeichnis auf der lokalen Festplatte speichert. Wie anhand des Skripts zu erkennen ist, werden aber nicht die XML-Dateien gespeichert, sondern nur der von allen Tags befreite Text der Transkripte (dies betrifft nun auch die von den Editoren gemachten Anmerkungen). Dasselbe Skript sorgt auch für eine durchgängige Kleinschreibung des gesamten Textes.

Es wäre auch möglich gewesen, den Vorgang der Datenbeschaffung direkt in das R-Skript zu integrieren, mit dessen Hilfe das Topic Model und die Auswertung realisiert wird,[20] wodurch der gesamte Work-Flow, also die Datenbeschaffung, Aufbereitung, Modellierung und Analyse bzw. Visualisierung mit nur einem Knopfdruck hätte erfolgen können. Auf ein solches Vorgehen wurde hier aber verzichtet, da für diesen Artikel mehrere verschiedene Modelle erzeugt wurden, die zu prozessierenden Daten jedoch stets unverändert blieben. Aus diesem Grund wurden die Daten nur einmal heruntergeladen, vorbereitet und in einem Verzeichnis auf dem lokalen Rechner gespeichert, worauf das R-Skript zugreifen kann.

4.2 Datenaufbereitung

Wie bereits angemerkt, hält sich der Aufwand für die Datenaufbereitung bei dem hier geplanten Topic Modeling-Verfahren in Grenzen. So müssen die Texte der zu analysierenden Thun-Korrespondenz, die ja bereits als einzelne Dokumente im txt-Format in einem lokalen Verzeichnis liegen, nur noch in R eingelesen und in den R-Datentyp »data frame« transformiert werden. Bei einem data frame handelt es sich um eine Matrix, deren Werte – im Unterschied zu dem R-Datentyp »matrix« – nicht alle vom selben Datentyp sein müssen.

[19] https://github.com/csae8092/topicModeling/blob/master/python/getXMLfromThunRegExCleaned.py.
[20] https://github.com/csae8092/topicModeling/blob/master/R/TopicModel_txt.R.

Allerdings wird in der gesamten Literatur zum Thema Topic Modeling weitgehend einstimmig darauf hingewiesen, dass die Qualität des Modells und somit auch die Qualität der einzelnen Topics stark von der Anzahl der Dokumente abhängt. Die Faustregel lautet: je weniger Dokumente, je schlechter das Modell.[21] Versteht man unter ›Dokument‹ nun einen für sich alleinstehenden Text wie etwa einen Roman, einen Aufsatz, einen Abstract, einen Lexikonartikel oder auch einen Brief, so würde dies für das hier zu bestreitende Experiment bedeuten, dass das Topic Model aus nur 81 Dokumenten berechnet werden müsste. Auch ohne tiefere Kenntnisse in Statistik sollte klar sein, dass diese Zahl tendenziell zu niedrig sein dürfte, um einigermaßen verlässliche Ergebnisse erzielen zu können. Des Weiteren sei an dieser Stelle auf die bereits weiter oben präsentierte Übersicht hinsichtlich der Länge bzw. der Wortanzahl der einzelnen Dokumente verwiesen (Tabelle 1), geht aus dieser doch deutlich hervor, dass die einzelnen Briefe von höchst unterschiedlichem Umfang sind. So umfasst das kürzeste Dokument, ein Schreiben Joseph Jelačičs an Caroline Thun vom 30. März 1850,[22] gerade einmal 93 ›Wörter‹ während der längste Text, ein Gesuch niederösterreichischer Grundherren an den Ministerrat,[23] 9543 ›Wörter‹ zählt.

Um sowohl das Problem der unterschiedlichen Längen, als auch jenes der geringen Anzahl der Dokumente in den Griff zu bekommen, müssen die einzelnen Texte des Korpus für das Topic Modeling-Verfahren in kleinere Einheiten unterteilt werden. Dafür bieten sich nun wenigstens zweierlei Herangehensweisen an:

Einerseits könnte ein Text entlang einer allfällig gegebenen Binnenstruktur geteilt werden. Im Falle der Texte der Thun-Korrespondenz böten sich die einzelnen Absätze in den Briefen an, insbesondere deshalb, da diese auch entsprechend der Empfehlungen der TEI kodiert wurden. Aus mehreren Überlegungen wurde davon aber Abstand genommen. So wurden von den Editoren der Briefe neben den Absätzen im eigentlichen Brieftext auch die Gruß- und Verabschiedungsformeln als Absätze ausgezeichnet, ebenso wie die meist am Briefbeginn oder -ende anzutreffenden Datierungen, ohne aber diese verschiedenen Arten von Absätzen näher zu typisieren. Eine Summe aller Absätze würde daher viele sehr kurze und inhaltlich mäßig relevante mit längeren und inhaltlich sehr wohl relevanten Textteilen kombinieren. Aber auch wenn die weniger bedeutungsvollen Absätze ausgesondert werden könnten, was mit ein wenig Datenmodellierung einigermaßen gut zu bewerkstelligen wäre, so bestünde auf Ebene der Absätze immer noch das Problem unterschiedlicher Textlängen. Dieses ließe sich jedoch auf ähnliche Art und Weise in den Griff bekommen, wie dies auch auf der Ebene der gesamten Texte erfolgt ist.

Andererseits können die Texte auch einfach nach einer bestimmten Anzahl von Wörtern, beispielsweise nach jedem zweihundertsten oder jedem zweitausendsten Wort geteilt werden. Die Vorteile einer solchen Normalisierung bestehen sowohl in der Einfachheit der technischen Realisierung als auch in den daraus resultierenden gleichlangen Dokumenten, sieht man

[21] Vgl. Tang et al. 2014.
[22] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=jelacic-an-caroline-thun_1850-03-30_A3-XXI-D37.xml.
[23] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=gesuch-niederoesterreichischer-grundherren-an-ministerrat_1850-05-14_A3-XXI-D52.xml.

einmal von der Größe des letzten Textteils eines jeden Dokuments ab. Die Nachtteile wiederum liegen in einer gewissen Willkür, in der die Auswahl der Wortanzahl, nach welcher der Text gebrochen werden soll. Außerdem besteht die begründete Gefahr, dass diese künstlich herbeigeführten Bruchlinien thematisch homogene Passagen wie beispielsweise Absätze – sofern diese bewusst gesetzt wurden – trennen können.

Für den Turing Test wurde in weiterer Folge mit zwei unterschiedlichen Datensets gearbeitet, welche mit Hilfe der Funktion »makeFlexTextChunks«[24] aus den 81 Dokumenten der Thun-Korrespondenz erstellt wurden. »makeFlexTextChunks« basiert dabei weitgehend auf einer gleichnamigen, von Jockers geschriebenen Funktion.[25] Die Funktion übernimmt als Parameter einen Text und einen Wert, der festlegt, nach wie vielen Wörtern der übergebene Text geteilt werden soll.

Das erste Datenset resultiert auf dem Textteilungsparameter 2000 – jeder Text wird nach 2000 Wörtern geteilt – und umfasst 111 Texteinheiten, -teile, chunks oder Dokumente, um mit letzterem Begriff in der Terminologie des Topic Modelings zu bleiben.[26] Die Wahl eines Textteilungsparameters in dieser Größenordnung erfolgte mit dem Hintergedanken, die Mehrheit der Texte des Thun-Korpus nicht aufsplitten zu müssen, überlange Texte aber dennoch normalisieren zu können. Bei diesem Datenset kann nun davon ausgegangen werden, dass inhaltlich zusammengehörige Passagen innerhalb eines Texts nicht oder nur in sehr wenigen Fällen getrennt wurden. Bei diesem Datenset muss aber auch davon ausgegangen werden, dass die geringe Anzahl von chunks oder Dokumenten sich negativ auf die Qualität des Topic Models auswirkt.

Das zweite Datenset hingegen wurde mit dem Textteilungsparameter 200 erstellt und setzt sich aus 634 Dokumenten zusammen.[27] Hier darf davon ausgegangen werden, dass die einzelnen Dokumente hinsichtlich ihrer jeweiligen Textlänge einheitlicher gestaltet sind, als dies beim vorigen Set der Fall ist. Außerdem darf gehofft werden, dass die Qualität des Topic Models besser ausfallen wird. Allerdings muss auch in Kauf genommen werden, dass Themenblöcke in den einzelnen Texten häufiger getrennt wurden als beim ersten Datenset. Zu bedenken gilt es außerdem – und darüber wird gegen Ende dieses Artikels noch zu sprechen sein –, dass eine höhere Anzahl an Dokumenten die Erstellung und Erfassung visualisierter Ergebnisse des Topic Modeling-Vorgangs erschweren.

4.3 Topic Modeling

Was das Verhältnis des Arbeitsaufwandes für die Datenaufbereitung gegenüber der Datenverarbeitung in Form von Topic Modeling betrifft, so kann dieses unter anderem anhand der dafür notwendigen Codezeilen abgeschätzt werden. Ausgehend von der Situation, dass die zu prozessierenden Daten bereits auf der lokalen Festplatte und im gewünschten Format

[24] https://github.com/csae8092/topicModeling/blob/master/R/code/TopicModel_externalFunctions.R.
[25] Jockers 2014, S. 138.
[26] Vgl. https://github.com/csae8092/topicModeling/tree/master/results/2000_53.
[27] Vgl. https://github.com/csae8092/topicModeling/tree/master/results/200_53.

vorliegen, sind für die Datenaufbereitung rund 30 Zeilen Code nötig, wobei sich diese Zahl durch Verwendung kompakterer Ausdrücke aber noch reduzieren ließe. Die für die Erstellung des auf diesen aufbereiteten Daten basierenden Topic Models notwendigen Schritte umfassen hingegen gerade einmal sechs Zeilen.

Mit dieser Gegenüberstellung soll zum Ausdruck gebracht werden, dass sich der Großteil der fürs Topic Modeling zu erbringenden Eigenleistung auf die vorausgehende Datenmodellierung erstreckt. Dass die eigentliche Erstellung des Topic Models selbst dann ohne weitere große Mühen erfolgen kann, ist jedoch weniger einer möglichen Trivialität dieses Vorganges geschuldet, als vielmehr den dafür existierenden Werkzeugen, Paketen oder Bibliotheken zu danken.

Konkret wurde zum Erstellen der Topic Models für diesen Artikel das R-package »mallet« verwendet, ein »wrapper around the Java machine learning tool MALLET«, geschrieben und gewartet von David Mimno.[28] Während es sich bei der Java Version von Mallet aber um ein umfassendes Natural Language Processing Toolkit handelt,[29] erschöpft sich das gleichnamige R-Paket in seiner Topic Modeling-Funktionalität.

Mallet ermöglicht es, eine Instanz eines Topic Models zu erstellen. Dieser Instanz müssen in Form von Parametern die zu analysierenden Dokumente und deren ›Titel‹ oder Identifikatoren (z. B. Dateinamen) übergeben werden. Außerdem kann festgelegt werden, ob der Text hinsichtlich Groß-Kleinschreibung normalisiert werden soll, wie der Text in einzelne Wörter unterteilt wird (Tokenizer), und es kann eine Liste mit Wörtern übergeben werden, welche bei der Erstellung des Topic Models nicht berücksichtigt werden sollen.

```
mallet.instances lt- mallet.import(documents$id, documents$text, "./R/stoplist.csv", FALSE)
```

Anschließend muss ein Trainingsobjekt erstellt werden, welchem als Parameter auch die Anzahl der zu generierenden Topics übergeben wird.

```
topic.model lt- MalletLDA(num.topics=53)
```

In dieses Trainingsobjekt werden danach die konkreten Daten in Form der zuvor erstellten Instanz geladen.

```
topic.model$loadDocuments(mallet.instances)>
```

Jockers folgend besteht nun die Möglichkeit, »to tweak the optimiziation hyperparameters«, sprich die Anzahl der »burn-in iterations« and »iterations between optimization« festzulegen, deren Standardwerte bei 200 und 50 liegen.

[28] Mimno 2013.
[29] MALLET 2013.

```
topic.model$setAlphaOptimization(40, 80)
```

In einer Fußnote dazu notiert Jockers: »The ramifications of resetting these values is beyond the scope of this chapter«[30] und verweist auf einen Aufsatz von Wallach, Mimno und McCallum.[31] Für die Erstellung der Topic Models für diesen Artikel wurden die von Matthew Jockers verwendeten Parameter übernommen, ohne aber die daraus folgenden Konsequenzen verstehen oder wenigstens abschätzen zu können. Ein Blick in das von Jockers empfohlene Paper macht rasch deutlich, dass die darin verhandelten Überlegungen ein tieferes mathematisch-statistisches Verständnis erfordern, dessen adequate Nachvollziehbarkeit hier nur punktuell angestrebt wird.

```
topic.model$train(400)
```

Dieser Befehl startet den Topic Modeling-Vorgang und führt diesen 400 Mal durch. Wie Jockers anmerkt, sollte mit jedem Durchlauf die Qualität des Models verbessert werden, seine eigenen Versuche zeigen jedoch, dass ab einer bestimmten Anzahl von Iterationen die Ergebnisse wieder an Qualität verlieren.[32]

4.4 Analyse durch Visualisierung

Nach Ausführung des letztgenannten -Befehls soll, das so erzeugte und in dem R-Objekt gespeicherte Topic Model der Thun-Korrespondenz zu analyisiert werden. Vor dem Hintergrund des hier durchzuführenden Experiments gilt es in erster Linie zwei Fragestellungen zu beantworten: Erstens geht es darum zu überprüfen, ob den von der Maschine generierten Topics sinnvollerweise auch ein Thema, eine Bedeutung eingeschrieben oder zugewiesen werden kann. Zweitens muss es möglich sein zu überprüfen, in welchen Texten bzw. Textabschnitten welche Topics wie stark vertreten sind.

Für die Beantwortung der ersten Frage bedarf es einer Aufstellung, die darüber Auskunft gibt, welche Wörter wie oft in jedem einzelnen Topic vorkommen. Eine solche Aufstellung erzeugt die Funktion , welche das trainierte Topic Model als Parameter übernimmt und eine Matrix auswirft, worin die Reihen die Topics, die Spalten die Wörter aus dem gesamten Wortschatz des Datenmaterials benennen und deren Felder die Häufigkeit der Wörter pro Topic beinhalten. Bei 53 Topics und einer Anzahl von 17.173 prozessierten distinkten Wörtern (exklusive der Stoppwörter) ergibt das eine Matrix von 910.169 Feldern.

Anhand dieser Matrix ließen sich nun die einzelnen Topics anhand der darin am häufigsten anzutreffenden Wörter beschreiben. Die Funktion erleichtert dieses Unterfangen jedoch, indem sie die häufigsten Wörter eines Topics zurückliefert, wobei die Anzahl der ausgegebenen Wörter und das jeweilige Topic durch die Übergabe entsprechender Parameter frei auswählbar

[30] Jockers 2014, S. 146.
[31] Wallach et al. 2009.
[32] Jockers 2014, S. 147.

sind. Da die Funktion aber nicht nur die Wörter selbst, sondern auch deren Häufigkeit präsentiert, ist es ein Leichtes, mit diesen Daten für jedes Topic im Model eine Wortwolke der n-häufigsten Wörter zu gestalten. Vor allem auch, weil es für R diverse Pakete für die Erstellung von Wortwolken gibt. Dazu zählt auch das hier verwendete package »wordcloud«. Mittels einfacher Iteration über die Anzahl der Topics wird so von jedem einzelnen Topic eine Wortwolke mit den 150 am häufigsten darin vorkommenden Wörtern erstellt und im .png-Format auf der lokalen Festplatte gespeichert. (Abbildung 1–3)

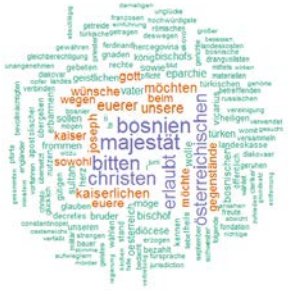

Abb. 1: Beispiel von in Form von Wortwolken visualisierten Topics. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/4.png.

Abb. 2: Beispiel von in Form von Wortwolken visualisierten Topics. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/14.png.

Abb. 3: Beispiel von in Form von Wortwolken visualisierten Topics. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/20.png.

Wortwolken waren und sind im Umfeld der Digital Humanities gehäuft anzutreffen. Nicht zuletzt aufgrund dieses teils inflationären (Stichwort: Wordl) und teils wenig reflektierten Gebrauchs (Stichwort: Tokenizer) kann die Verwendung von Wortwolken kritisch gesehen werden. Gerade im Falle von Topic Modeling scheinen Wortwolken aber ein probates Mittel zu sein um auszutesten, ob den generierten Topics Sinn unterstellt werden kann, und falls ja, um abzuschätzen, welcher Themenkomplex von einem Topic beschrieben werden kann.

Was die Frage nach der Lokalisierung der Topics bzw. die Frage nach der Häufigkeit jedes einzelnen Topics innerhalb des gesamten Korpus betrifft, so erweist sich hier die Funktion als äußerst hilfreich. Diese liefert eine Matrix zurück, deren Reihen die Dokumente und deren Spalten die Topics bilden. Die Felder dieser Matrix beinhalten wiederum einen Wert, der über den Grad der Wahrscheinlichkeit Auskunft gibt, mit welcher ein Topic in einem Dokument anzutreffen ist. Bei 634 Dokumenten (das sind die 81 Texte der Korrespondenz, gesplittet nach jedem 200. Wort) und 115 Topics ergibt das somit eine Matrix mit 634 Reihen, 115 Spalten und 72.910 Feldern. Diese Matrix kann nun (u.a.) mit Hilfe des R packages »heatmap.2« in Form einer Heatmap visualisiert werden. (Abbildung 4–5)

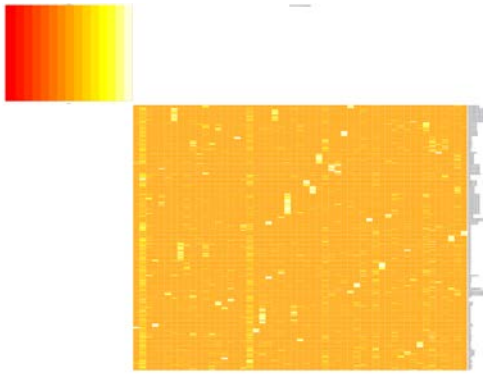

Abb. 4: Topic Model Thun Korrespondenz, 634 Dokumente und 53 Topics (chunksize 200). © Peter Andorfer, 2015, hochauflösende Datei unter https://github.com/csae8092/topicModeling/blob/master/results/200_53/heatmap.png abrufbar.

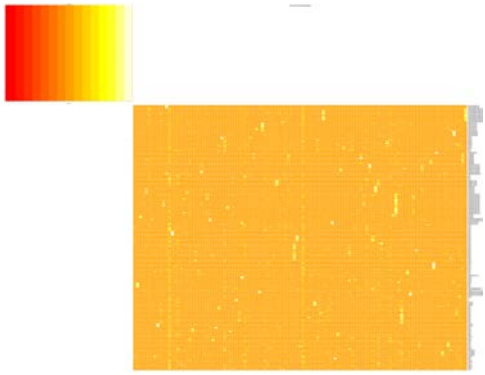

Abb. 5: Topic Model Thun Korrespondenz, 111 Texte und 115 Topics (chunksize 2000). © Peter Andorfer, 2015, hochauflösende Datei unter https://github.com/csae8092/topicModeling/blob/master/results/200_115/heatmap.png abrufbar.

5. Maschine gegen Mensch

Wenn im Titel dieses Artikels ein Turing Test angekündigt wurde, dann muss ein solcher natürlich auch durchgeführt werden, selbst wenn bereits aus dem Untertitel einigermaßen deutlich geworden sein dürfte, dass die Bezeichnung Turing Test hier eher als Chiffre für einen weiter gefassten Vergleich menschlicher und maschineller Verfahren zur inhaltlichen Erschließung großer Textmengen steht als für den konkreten Turing Test selbst.

5.1 Menschliche Schlagwörter versus maschinelle Labels

Ein solch konkreter Test kann in der Form stattfinden, dass man einer Reihe von Personen entweder die Liste der von den Editoren erstellten Schlagwörter vorlegt oder die Liste maschinell erzeugter Schlagwörter (Labels). Die Versuchskandidaten müssen dann entscheiden, ob sie es mit einem Produkt menschlicher oder künstlicher Intelligenz zu tun haben. Klarerweise darf es sich bei den maschinellen Schlagwörtern dabei natürlich nicht um die von einem Menschen auf jeweils ein Wort verdichteten Interpretationen der automatisch generierten Topics handeln, vielmehr müssen diese direkt von der Maschine erzeugt werden. Eine solche Liste könnte etwa aus den am häufigsten verwendetem Wort eines jeden Topics bestehen, die mit der Mallet Funktion sehr einfach erstellt werden kann. Allein aber aus einem direkten Vergleich von zehn weitgehend arbiträr ausgewählten Schlagwörtern aus der Liste von automatisch generierten Labels mit zehn vom Menschen definierten Schlagwörtern, wird offenkundig, dass die Maschine in diesem Turing Test – ohne weitere menschlichen Eingriffe – keine allzu großen Gewinnchancen hat.

Tabelle 2: Schlagwörter und Labels

Mensch[33]	Maschine[34]
Personalfragen	lernen
Kultus	klasse
Gymnasium	kaiserin
Volksschulen	ausgesprochen
Nationalitätenfrage	berechtigen
Bischofsversammlung	hätte
Siebenbürgen	geistlichen
Sprachanfrage	gemeinden
Verwaltung	schulen
Universität	wahlen

Die Entscheidung, ob es sich um Mensch oder Maschine handelt, kann bei diesem Beispiel bereits auf formaler Ebene getroffen werden, ohne überhaupt auf die Semantik dieser Wörter eingehen zu müssen. Während es sich bei den von Menschen vergebenen Schlagwörtern ausschließlich um Substantive handelt, befinden sich unter den maschinell erzeugten Labels auch andere Wortarten, beispielsweise die Verbform »hätten«, welches noch dazu in einer flektierten Form und nicht im Infinitiv anzutreffen ist. Hinzukommt – und dies ist vermutlich noch augenfälliger – die konsequente Kleinschreibung der Labels.

Diese formalen Probleme ließen sich – auch mit dem vorhandenen Datenmaterial – jedoch lösen. So könnte etwa auf die Normalisierung in Form von ausschließlicher Kleinschreibung verzichtet werden und basierend auf der Unterscheidung zwischen groß- und kleingeschriebenen Wörtern ausschließlich jene Wörter in das Datenmaterial für das Topic Modeling aufgenommen werden, welche mit einem Großbuchstaben beginnen, in der Annahme, dass es sich dabei weitgehend um Substantive handelt. Aber selbst wenn man nur die Substantive in den Labels berücksichtigen würde und diese auch konsequent großgeschrieben wären, könnten menschliche und maschinelle Schlagwörter, sofern sie einen Korpus historischer Texte beschreiben, anhand historischer Schreibweisen, welche in den maschinellen Schlagwörtern angetroffen werden, unterschieden werden. Die Maschine kann für die Generierung von Labels bekanntlich nur auf den konkret im Korpus vorhandenen Wortschatz zurückgreifen. Es sei denn, und dies sei nur als Idee am Rande notiert, man würde versuchen, diesen Wortschatz, vielleicht aber auch nur die aus diesem Wortschatz generierten Labels mit einer Ressource zu verknüpfen, welche in der Lage ist, Wortbedeutungen zu kennen und zu abstrahieren, wozu etwa GermNet[35] herangezogen werden könnte.

[33] Für eine Liste aller von den Editoren vergebenen Schlagwörtern samt deren Häufigkeit vgl. http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/schlagworte-all.html.
[34] Die maschinell generierte Liste von Schlagwörtern (Labels) kann hier abgerufen werden: https://github.com/csae8092/topicModeling/blob/master/results/200_53/topicLabes.csv.
[35] GermaNet 2009; Hamp / Feldweg 1997; Henrich / Hinrichs 2010.

Aber auch hinsichtlich der semantisch-inhaltlichen Qualität von menschlich erstellten Schlagwörtern und maschinellen Labels können relativ rasch Unterschiede ausgemacht werden. Denn während die Schlagwörter häufig abstrakte Konzepte bezeichnen, finden sich unter den Labels häufig deren konkrete Ausprägungen. In der obigen Tabelle kann man dies am Schlagwort »Verwaltung« und dem Label »gemeinden« festmachen. Aber auch hier könnte GermaNet helfen, die maschinellen Labels zu ›vermenschlichen‹.

5.2 Semantisch aufladbare Topics

Die oben vorgenommene implizite Gleichstellung von »Verwaltung« und »gemeinden« ist ohne Kenntnisse weiterer Wörter, die dieses Topic konstituieren, milde ausgedrückt gewagt. Dass sie im konkreten Fall jedoch nicht gänzlich aus der Luft gegriffen ist, zeigt die entsprechende Wortwolke (Abbildung 6):

Abb. 6: Wortwolke zum 28. Topic »gemeinden«. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/28.png.

Tabelle 3 zeigt eine Übersicht über die 30 häufigsten Wörter dieses Topics. Es gilt jedoch zu beachten, dass diese Wortliste in einem anderen Topic Modeling-Durchlauf als die Wortwolke erstellt wurde.Tabelle 3 zeigt eine Übersicht über die 30 häufigsten Wörter dieses Topics. Es gilt jedoch zu beachten, dass diese Wortliste in einem anderen Topic Modeling-Durchlauf als die Wortwolke erstellt wurde.

Tabelle 3: Die 30 häufigsten Wörter des 30. Topics

words	weights
gemeinden	0.031447141
gemeinde	0.014458827
alten	0.010211749
liblin	0.010211749

ortschaften	0.009362333
richter	0.008512917
seit	0.007663501
unsere	0.007663501
steuer	0.007663501
k	0.006814086
endlich	0.006814086
müßten	0.006814086
verhältnis	0.006814086
kleinen	0.006814086
fl	0.006814086
tag	0.005115254
trotz	0.005115254
dorfrichter	0.005115254
ehemaligen	0.004265839
geschäfte	0.004265839
neue	0.004265839
früher	0.004265839
mittelgroßen	0.004265839
grundbesitzer	0.004265839
obliegenheiten	0.004265839

Sucht man in dem Korpus nach Dokumenten, in denen dieses Topic häufig vorkommt – was anhand der als Heatmap visualisierten Topic-Dokument-Matrix gut möglich ist – findet man rasch das Dokument *Gemeindemitglieder von Liblin an den Ministerrat Liblin, 6. September 1850*,[36] ein Dokument, das von den Editoren – den Menschen also – mit den Schlagwörtern »Gemeindeverwaltung«, »Gemeindeordnung« und »Neoabsolutismus« belegt wurde.

Anhand der Heatmap lassen sich jedoch auch noch andere Dokumente ausfindig machen, in denen das Topic »gemeinden« gehäuft anzutreffen ist. So etwa in einem Textteil aus dem Dokument *Konzept eines Briefes von Leo Thun ohne Adresssat.* [37] Im Gegensatz zu dem zuvor genannten Dokument wurde dieses »Konzept« von den Editoren jedoch nicht mit »Gemeindeverwaltung« verschlagwortet, sondern mit »Volksschulen«, »Volksschullehrer« und »Gehaltsfragen«. Hier ein Auszug aus der entsprechenden Passage:

[36] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/results.html?ref=gemeindemitglieder-von-liblin-an-ministerium_1850_09-06_A3-XXI-D72.xml&searchword=Wurmbrand.
[37] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=thun-oA-konzept_1849_A3-XXI-D2.xml.

> »[...] um Geschäfte ihre gut zu besorgen, - und werden zu können, was sie geworden sind, - er aber darbt in bitterer Noth; er der fremde Kinder erzogen hat, weiß vielleicht jetzt nicht, wie er seine eigenen erhalten und ernähren soll! Denn selbst das Schulgeld und die sonstigen Giebigkeiten, die er bisher bezogen hatte, werden ihm jetzt oft verweigert, seit in den Zeiten allgemeiner Unordnung, die wir im vorigen Jahr erlebt haben, auch die Meinung ausgesprengt wurde, das Schulgeld müsse aufhören, denn der Staat müsse die Schullehrer bestohlen. Nichts ist verderblicher für die gegenwärtige Lage der Schullehrer geworden als die Verbreitung dieses Gedankens. Der Schullehrer arbeitet zunächst nur für seine Gemeinde; es ist also das Natürlichste, daß hauptsächlich sie ihn bezahle, und immer und überall wird das wohl so sein. Die Gemeinde könnte überdies wenigstens der dringenden Noth gleich abhelfen, während die Regierung es nicht kann, ehe sie durch Gesetze dazu ermächtigt ist.«[38]

Aus der Lektüre dieses Auszuges geht deutlich hervor, dass hier ›Gemeinde‹ nicht im Kontext von »Gemeindeverwaltung« oder »Gemeindeordnung« gebraucht wird, sondern als lokale Bezugsgröße dient. Gleichzeitig können aber in der Wortwolke dieses Topics eine Reihe weiterer Wörter gefunden werden, welche durchaus auf die zuvor erwähnten Verwaltungskontexte verweisen wie etwa »richter«, »dorfrichter«, »grundbesitzer« oder »steuer«. Die dominierenden Begriffe dieses Topics sind jedoch »gemeinde« und »gemeinden«, weshalb dieses Topic auch für die obige Passage als dominant ausgegeben wird, kommt darin »Gemeinde« doch gleich zweimal vor. Außerdem finden sich in diesem Abschnitt auch noch die Wörter »seit«, »besolden« und »Geschäft« wieder. Eine ähnliches Beispiel stellt das Dokument *Ein bosnischer katholischer Priester an Joseph Strossmayer* [39] dar.

Ohne die Probe aufs Exempel für jedes Topic durchzuführen, darf wohl davon ausgegangen werden, dass bei einer Anzahl von 53 generierten Topics diese meist mehre Themenbereiche umfassen. Eine Beobachtung, die in der einschlägigen Literatur intensiv diskutiert wird.[40] Folgt man der von Jordan Boyd-Graber und anderen vorgestellten »Categories of Poor Quality Topics«, dann wäre jenes »gemeinden« -Topic wohl am ehesten ein »mixed and chained topic«,[41] welches außerdem noch mit den erschwerenden Bedingungen zu kämpfen hat, sowohl sehr allgemeine (»gemeinde«) als auch sehr spezifische (»liblin«) Worte (»General and specific words«[42]) zu beinhalten.

Resümierend kann also festgehalten werden, dass die semantische Aufladung von mittels Topic Modeling generierten Wortlisten möglich und zulässig ist, sofern die Interpretationen nicht zu eng gefasst sind. Eine gezielte Suche nach sehr konkreten Themengebieten ist mit diesen automatisch genierten Topics jedoch nicht mit jener hohen Präzision möglich, wie manche Diskurse um das Topic Modeling gelegentlich versprechen. Festgehalten werden muss aber auch, dass der Akt der Interpretation der maschinell zusammengestellten Wortlisten

[38] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=thun-oA-konzept_1849_A3-XXI-D2.xml.
[39] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=priester-an-strossmayer_1850-06-16_A3-XXI-D57.xml.
[40] Vgl. dazu Jockers 2014, S. 144.
[41] Boyd-Graber et al. 2015, S. 17.
[42] Boyd-Graber et al. 2014, S. 16.

zeitaufwendig ist und trotz aller nicht menschlicher Vorarbeiten letztendlich wieder subjektiv und individuell gefärbte Themen/Topics produziert.

Die für den Wettstreit Mensch-Maschine entscheidende Frage, ob die Maschine generell im Stande ist, ›sinnvolle‹, sprich semantisch aufladbare Topics oder Wortlisten zu generieren, kann insgesamt also bejaht werden. Denn auch wenn die Maschine einem Topic niemals selbst Sinn und Bedeutung einschreibt, so ist sie dennoch in der Lage, Wortlisten zu produzieren, die vom Menschen als sinnvoll bewertet werden. Dies wurde an anderer Stelle bereits mittels »word-intrusion«- und »topic-intrusion«-Tests belegt[43] und dies zeigen auch die meisten aus dem Korpus der Thun-Korrespondenz generierten Topics (Abbildung 7)

Abb.7: Drei Beispiele ›kohärenter‹, sprich leicht interpretierbarer Topics. Diese Topics basieren auf nur 111 Dokumenten (chunksize 2000). © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/2000_53/wordclouds/1.png, https://github.com/csae8092/topicModeling/blob/master/results/2000_53/wordclouds/48.png, https://github.com/csae8092/topicModeling/blob/master/results/2000_53/wordclouds/46.png.

Auffallend ist in diesem Kontext außerdem die Beobachtung, dass die Anzahl der Dokumente (634 oder 111) auf die Interpretierbarkeit der daraus generierten Topics keinen erkennbaren Einfluss genommen haben dürfte.[44]

5.3 Ordnung, Strukturierung und Orientierung von und in Korpora

Doch welchen Beitrag können diese mehrheitlich ›sinnvollen‹ Topics zur Ordnung und Strukturierung und besseren Orientierung in den jeweiligen Textmassen leisten?

[43] Vgl. Chang et al. 2009.
[44] Topics bei 634 Dokumenten: https://github.com/csae8092/topicModeling/tree/master/results/200_53/wordclouds und Topics bei 81 Dokumenten: https://github.com/csae8092/topicModeling/tree/master/results/2000_53/wordclouds.

Wie schon in Abschnitt 4 angemerkt, folgte die Schlagwortvergabe im Zuge der Transkription der Texte keinen vorgegebenen und somit heute noch einsehbaren und nachvollziehbaren Richtlinien. Die Konsequenz daraus ist eine nicht strukturierte, unterschiedlich granulare und teils auch redundante Liste von Schlagwörtern. Diese Eigenschaften treffen aber auch auf die maschinell erzeugten Topics zu, weshalb hier Mensch und Maschine gleichauf liegen, auch wenn man fairerweise ins Feld führen muss, dass sich mittels eines korrigierenden Arbeitsschrittes diese flache, ungeordnete und redundante Liste (menschlich erstellter) Schlagwörter relativ rasch in eine in sich schlüssige und ausdifferenzierte Ontologie transformieren ließe. Entsprechende Überlegungen wurden seitens der Editoren bereits formuliert und dürften, vorausgesetzt die dafür notwendigen Ressourcen können aufgetrieben werden, in nicht allzu ferner Zukunft realisiert werden. Ob und wie eine solche Strukturierung von Themen auch maschinell zu realisieren wäre, ist unklar. Möglicherweise könnten auch hier Ressourcen wie GermNet helfen.

Mögen Mensch und Maschine hinsichtlich der (Un-)Ordnung der von ihnen generierten Topics noch gleichauf liegen, so basieren die maschinell erzeugten Topics jedoch nicht auf stets subjektiven und individuellen Lesarten und Interpretationen einzelner Texte, wie dies bei den von Menschen vergebenen Schlagwörtern unausweichlich der Fall ist. Topic Modeling funktioniert frei von persönlichem Vorwissen oder speziellen Forschungsinteressen, weshalb die generierten Topics auch nicht entsprechend gefärbt sind und somit einen unvoreingenommenen und nicht bereits von allfälligen Vorannahmen gefilterten Zugriff auf das Korpus ermöglichen. Geht man etwa die von den Editoren vergebenen Schlagwörter der Thun-Korrespondenz durch, lassen sich rasch die Themenblöcke ›Bildung‹, ›Verwaltung‹, ›Politik‹ und ›Kirche‹ als die dominierenden Inhalte des Korpus bestimmen. Dies ist ein Eindruck, der über weite Strecken auch anhand der Durchsicht der maschinell erzeugten Topics bestätigt werden kann. Allerdings generiert die Maschine auch Wortlisten, deren mögliche Interpretationen kaum Entsprechung in einem der menschlichen Schlagwörter findet. Eine solche Wortliste zeigt nachfolgende Abbildung 8.

Abb. 8: Wortwolke zum 33. Topic. Vgl. https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/33.png.

Dieses Topic könnte mit »Geld, Finanzen, Ausgaben« überschrieben werden. Ein Themenbereich, welcher von den 115 vergebenen Schlagwörtern in dieser Form nicht abgedeckt wird. Thematisch am nächsten liegt hier nur noch das im Korpus zweimal anzutreffende Schlagwort »Gehaltsfragen«. In diesen beiden mit »Gehaltsfragen« überschriebenen Dokumenten[45] ist das Topic 33 (Geld/Finanzen/Ausgaben) aber nicht sehr stark ausgeprägt, wie ein Blick auf die visualisierte Topic-Dokument-Matrix zeigt.

Ungleich markanter tritt Topic 33 aber in dem Dokument *Entwurf zur Neuregelung der Kompetenzen des Ministeriums für Kultus und Unterricht* [46] in Erscheinung. Es handelt sich um ein Dokument, dem die Editoren die Schlagwörter »Ministerium für Kultus und Unterricht«, »Verwaltung« und »Kultur« zugewiesen haben.

Im Gegensatz zum Menschen ist die Maschine außerdem im Stande, die Gewichtung der Topics in den Dokumenten systematisch in Zahlen zu beschreiben. Die derart dokumentierte thematische Verteilung ist somit nicht nur ebenfalls frei von jeglicher menschlicher Subjektivität, sondern kann auch sehr gut visualisiert werden. Die Maschine ist also im Stande, Themen aus großen Textmengen zu extrahieren, diese Themen in den Dokumenten zu lokalisieren und diese Informationen auch in einer einzigen Abbildung zu präsentieren. Menschen können so etwas prinzipiell weniger gut.

5.4 Das Ergebnis des Turing Tests

Wenn es darum geht konkrete Nutzungspotentiale von Topic Modeling zu skizzieren, so sei hier auf den Schluss von David Mimnos Paper *Computational Historiography* verwiesen.[47] Für den hier vorliegenden Artikel hingegen soll darüber hinaus aber vor allem der im vorigen Abschnitt zuletzt genannte Aspekt betont werden, insbesondere vor dem Hintergrund des hier angestellten Vergleichs zwischen Mensch und Maschine, denn wie gezeigt werden konnte, kann die Maschine sehr passabel Themen identifizieren und in den Texten lokalisieren. Was die Maschine weniger gut kann, sind Interpretation, Verschlagwortung und semantische Aufladung dieser Themen. Was aber hoffentlich ebenfalls deutlich geworden ist, ist, dass dieser Akt der Interpretation der Topics gar nicht immer notwendig ist. So etwa dann, wenn es darum geht, Texte auf inhaltlicher Ebene ordnen zu können. Dafür genügt es zu erkennen, in welchen Texten Themen ähnlich gewichtet sind, und dies kann die Maschine zweifelsfrei besser und objektiver als jeder Mensch.

Nur der Vollständigkeit halber sei außerdem noch darauf hingewiesen, dass Topic Modeling ohne großen Aufwand durchgeführt werden kann. Die entsprechenden Tools und die

[45] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=thun-oA-konzept_1849_A3-XXI-D2.xml&searchword=qwertzy; http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/show.html?ref=thun-friedrich-an-thun_1849-10-05_A3-XXI-D7.xml&searchword=qwertzy.
[46] http://thun-korrespondenz.uibk.ac.at:8080/exist/apps/Thun-Collection/pages/results.html?ref=entwurf-wirkungskreis-ministerium-fuer-kultus-und-unterricht_-o.D._A3-XXI-D84.xml&searchword=qwertzy.
[47] Mimno 2012.

entsprechenden Tutorials sind vorhanden und auch der methodisch-theoretische Kontext ist breit erforscht.

6. Diskussion und Ausblick

Dass Topic Modeling ohne großen Aufwand durchzuführen ist, wie eben noch angeführt, ist meist aber nur die halbe Wahrheit und trifft in erster Linie fast ausschließlich auf den eigentlichen Vorgang des Topic Modelings zu, nämlich auf das Konfigurieren der wenigen vorhandenen Parameter und das Einspeisen von Daten. Auf den mit der Datenaufbereitung einhergehenden Aufwand wurde schon verwiesen. Doch auch dieser bewegt sich angesichts der Genügsamkeit der gängigen Topic Modeling Tools, was das verwertbare Datenmaterial betrifft, in überschaubarem Rahmen. Allerdings wäre es sicherlich lohnenswert auszutesten, welche Ergebnisse zu erzielen wären, wenn die Texte desselben Korpus etwa in lemmatisierter Form und mit POS-Tags versehen vorliegen würden. Wie kohärent und ›sinnvoll‹ wären etwa Topics, die nur aus Texten von Adjektiven und Nomen bzw. deren Lemmata generiert werden würden?[48]

Auszutesten wäre auch, ob sich eine Einbindung einer Ressource wie GermaNet in einen Topic Modeling Workflow positiv auswirken könnte, sei es nun im Sinne einer (semantischen) Normalisierung des Ausgangsmaterials oder der Topics oder der automatisch generierten Labels.

Der größte Arbeitsaufwand scheint jedoch mit der Interpretation und Evaluation der mit Topic Modeling erzielten Ergebnisse verbunden zu sein. Vor allem dann, wenn man die genauen internen Abläufe dieser Technik nur rudimentär zu begreifen in der Lage ist und daher ein als sinnvoll und brauchbar erachtetes Model nur im trial and error-Verfahren erschaffen kann. Inwieweit es sich hierbei dann aber noch um ein ›objektives‹ oder ›unvoreingenommenes‹ Modell handelt, wie dies im vorigen Abschnitt ja noch behauptet wurde, ist fraglich.

Unbefriedigend sind außerdem auch die gängigen Lösungen der Visualisierung und damit einhergehend der Nutzung, Analyse und Evaluation von Topic Models. Die für diesen Aufsatz gewählte Form der Darstellung der Topics in Form von Wortwolken und der Topic-Dokument-Matrix in Form einer Heatmap dürften tendenziell in die richtige Richtung weisen. So umschifft die Wortwolke das Problem der Benennung der Topics und die Heatmap erlaubt eine rasche Orientierung im Korpus. Durch die Veröffentlichung dieser (und weiterer) Daten dürfte auch die notwendige wissenschaftliche Transparenz und Nachvollziehbarkeit der Ergebnisse gewährleistet sein. Die Benutzerfreundlichkeit hält sich aber dennoch in Grenzen. Hier wäre ein Zusammenführen der einzelnen Komponenten in einer interaktiven HTML-Darstellung wünschenswert, deren Kern die Heatmap darstellt, von wo aus einerseits zu den Wortwolken und den Dokumenten verlinkt werden kann und deren Reihen und Spaltenanordnung andererseits frei modifizierbar sind.

[48] Vgl. dazu etwa Jockers 2014, S. 157.

Doch was ist nun mit der Frage nach der wissenschaftlichen Vertretbarkeit der Verwendung von Tools und Methoden, die nicht zur Gänze verstanden wurden? Dazu noch zwei abschließende Bemerkungen. Vorliegender Artikel ist ein Beispiel dafür, dass man auch mit solchen Methoden Ergebnisse und Resultate erzielen kann. Die Wissenschaftlichkeit dieser Resultate, vor allem die mehrmals geäußerte Behauptung, die generierten Topics wären ›sinnvoll‹, mag jedoch berechtigterweise in Frage gestellt werden. Denn wer weiß, ob für die Person eine Reihe von einzelnen Wörtern so viel Sinn ergibt wie für eine andere. Abgesehen davon, dass zu fragen ist, wie ›objektiv‹ und unvoreingenommen solche Topics noch sind, wenn deren Generierung auf relativ willkürlichem Herumspielen mit den Parametern von der Anzahl der Dokumente und der Anzahl der Topics basiert.

Nichtsdestotrotz eröffnen die mehr oder weniger ›objektiv‹ generierten Topics neue Perspektiven auf ein vielleicht vermeintlich schon als gut erforscht geglaubtes Textkorpus. Vielleicht bestätigen die Topics und ihre Verteilung Theorien oder stoßen neue Fragestellungen an, woraus sich anschließende tatsächlich neue und auch wissenschaftlich haltbare Erkenntnisse gewinnen lassen.

Bibliographische Angaben

Christof Aichner: Die Korrespondenz von Leo von Thun-Hohenstein: Eine Dokumentation. In: Thun-App, 2015. [online]

Peter Andorfer (2015a): The 15 Shades of Grey: oder die Suche nach dem Sex. In: Digital-Archiv. Blogbeitrag vom 15. März 2015. [online]

Peter Andorfer (2015b): Quantitative Analyse der Thun-Korrespondenz. In: Thun-App, 2015. [online]

David Blei: Probabilistic Topic Models. DOI: 10.1145/2133806.2133826 In: Communications of the ACM 55 (2012), H. 4, S. 77–84. [online]

Jordan Boyd-Graber / David Mimno / David Newman: Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements. In: Handbook of Mixed Membership Models and Their Applications. Hg. von Edoard M. Airoldi / David M. Blei / Stephen E. Fienberg / Elena A. Erosheva (= CRC Handbooks of Modern Statistical Methods). Boca Raton 2015, S. 225–254. [Nachweis im GBV]

Megan R. Brett: Topic Modeling: A Basic Introduction. [online] In: Journal of Digital Humanities 2 (2013), H. 1. [online]

Jonathan Chang / Sean Gerrish / Chong Wang / Jordan L. Boyd-Graber / David M. Blei: Reading Tea Leaves: How Humans Interpret Topic Models. PDF. [online] In: Advances in Neural Information Processing Systems 22. Hg. von Yoshua Bengio / Dale Schuurmans / John D. Lafferty / Christopher K. I. Williams / Jaron Culotta (NIPS 22, Vancouver, 07.–10.12.2009). Vancouver 2009. [online]

Shawn Graham / Scott Weingart / Ian Milligan: Getting Started with Topic Modeling and MALLET. Programming Historian. 2. September 2012. [online]

Birgit Hamp / Helmut Feldweg: GermaNet - a Lexical-Semantic Net for German. In: Proceedings of the ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. Madrid 1997, S. 9–15. [online]

Verena Henrich / Erhard Hinrichs: GernEdiT – The GermaNet Editing Tool. PDF [online] In: Proceedings of the 7 International Conference on International Language Resources and Evaluation (LREC 7, Valetta, 17.–23.05.2010). Paris 2010, S. 2228–2235. PDF. [online]

E. L. James: Fifty Shades of Grey: Roman. München, 2012. [Nachweis im GBV]

Matthew Lee Jockers: Macroanalysis: Digital Methods and Literary History. Urbana 2013. [Nachweis im GBV]

Matthew Lee Jockers: Text analysis with R for students of literature. Cham 2014. [Nachweis im GBV]

Journal of Digital Humanities 2 (2012), H. 1. Hg. Daniel J. Cohen / Joan Fragaszy Troyano / Sasha Hoffman / Jeri Wieringa / Elijah Meeks / Scott Weingart. Fairfax, VA 2012. [online]

David Mimno: Computational Historiography: Data Mining in a Century of Classics Journals. In: ACM journal on computing and cultural heritage 5 (2012), H. 1. PDF. [online]

David Mimno: Mallet: A Wrapper around the Java Machine Learning Tool MALLET (version 1.0). 2013. [online]

Lisa M. Rhody: Topic Modeling and Figurative Language. In: Journal of Digital Humanities 2 (2012), H. 1. [online]

Benjamin M. Schmidt: Words Alone: Dismantling Topic Models in the Humanities. In: Journal of Digital Humanities 2 (2012), H. 1. [online]

Jian Tang / Zhaoshi Meng / Xuanlong Nguyen / Qiaozhu Mei / Ming Zhang: Understanding the Limiting Factors of Topic Modeling via Posterior Contraction Analysis. In: Proceedings of The 31st International Conference on Machine Learning. Hg. Eric P. Xing / Tony Jebara. (ICML 2014, Beijing, 21–26.06.2014) Red Hook, NY. 2014. (= JMLR Workshop and Conference Proceedings, 32). [online]

Allan M. Turing: Computing Machinery and Intelligence. In: Mind LIX 236 (1950), S. 433–460. DOI: 10.1093/mind/LIX.236.433

MALLET: A Machine Learning for Language Toolkit. Hg. University of Massachusetts Amherst. 2013. [online]

GermaNet: a german wordnet. Hg. University of Tübingen. 10.12.2009. [online]

Hanna M. Wallach / David Mimno / Andrew McCallum: Rethinking LDA: Why Priors Matter. [online] In: Advances in Neural Information Processing Systems 22. Hg. von Yoshua Bengio / Dale Schuurmans / John D. Lafferty / Christopher K. I. Williams / Jaron Culotta (NIPS 23, Vancouver, 07.–10.12.2009). Vancouver 2009. [online]

Scott Weingart: Topic Modeling for Humanists: A Guided Tour. In: The Scottbot Irregular. Blogbeitrag vom 25. Juli 2012. [online]

Abbildungslegende und -nachweise

Abb. 1: Beispiel von in Form von Wortwolken visualisierten Topics. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/4.png.

Abb. 2: Beispiel von in Form von Wortwolken visualisierten Topics. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/14.png.

Abb. 3: Beispiel von in Form von Wortwolken visualisierten Topics. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/20.png.

Abb. 4: Topic Model Thun Korrespondenz, 634 Dokumente und 53 Topics (chunksize 200). © Peter Andorfer, 2015, hochauflösende Datei unter https://github.com/csae8092/topicModeling/blob/master/results/200_53/heatmap.png abrufbar.

Abb. 5: Topic Model Thun Korrespondenz, 111 Texte und 115 Topics (chunksize 2000). © Peter Andorfer, 2015, hochauflösende Datei unter https://github.com/csae8092/topicModeling/blob/master/results/200_115/heatmap.png abrufbar.

Abb. 6: Wortwolke zum 28. Topic »gemeinden«. © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/28.png.

Abb. 7: Drei Beispiele ›kohärenter‹, sprich leicht interpretierbarer Topics. Diese Topics basieren auf nur 111 Dokumenten (chunksize 2000). © Peter Andorfer, 2015: https://github.com/csae8092/topicModeling/blob/master/results/2000_53/wordclouds/1.png, https://github.com/csae8092/topicModeling/blob/master/results/2000_53/wordclouds/48.png, https://github.com/csae8092/topicModeling/blob/master/results/2000_53/wordclouds/46.png.

Abb. 8: Wortwolke zum 33. Topic. Vgl. https://github.com/csae8092/topicModeling/blob/master/results/200_53/wordclouds/33.png.