Fachhochschule Köln
Cologne University of Applied Sciences

Exploration von Textkorpora

Topic Models als Grundlage der Interaktion

Masterthesis

zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

vorgelegt an der
Fachhochschule Köln
Fakultät für Informatik und Ingenieurswissenschaften

im Studiengang
Medieninformatik

von

Lennart Renker

Erster Prüfer: Prof. Dr. Kristian Fischer
Zweiter Prüfer: Prof. Dr. Gerhard Hartmann

Köln, den 24. Juli 2015

Inhaltsverzeichnis

Abstract

The internet offers a seemingly infinite amount of information. A central issue of our days is to make use of this information. To formulate efficient search queries a user must have good domain knowledge. Often this is not the case, wherefore a lot of time has to be invested to get an overview of the topic in question. In those situations a user ends up in an exploratory search process, in which he has to outline the individual topics step by step. By now machine learning algorithms are frequently used in data organization but stay invisible to the user for most of the time. The interactive use of these methods could optimize search processes by connecting the human ability to judge with machine processing powers used on large data sources. Topic models are algorithms that find latent topics in unstructured text corpora and are relatively good to interpret. Their use is promising in exploratory search processes, in which a user has to gain new domain knowledge quickly.

It appears that many researches use topic models mostly to generate static visualizations of latent text structures. Sensemaking is an essential part of exploratory search processes but it is still only used to a small extent in order to justify algorithmic novelties and bring them into a larger context. Therefore the assumption is derived, that the use of sensemaking models and user centered concepts for exploratory search processes could lead to the development of new ways to interact with topic models and to generate a framework for publications of the correlating research fields.

To proof this hypothesis, first an overview of fundamental methods of text preprocessing and stochastic topic modeling is given. Afterwards, the most essential cognitive models are described, which form the basis of sensemaking. Subsequently the ways of interaction in problem-solving-processes are discussed. New scientific publications will be evaluated, which address the exploratory search under the aspect of sensemaking, while using topic models or similar techniques. By joining the individual observations, general limitations of the current research field are derived and partially solved.

The evaluation was able to confirm the hypothesis. Limitations like the absence of methods for restructuring and comparing data were exposed and resolved in a conceptual manner. Furthermore the externalization and usage of schemes as a direct implementation of the Pirolli et al. sensemaking model [1] are put in prospect.

The observations and concepts showed a general necessity to stronger integrate sensemaking in exploratory search environments. The individual structure, comparison and externalization of schemes and previous knowledge seem therefore to have an important benefit for complex analyses. The present thesis can serve as a fundament for further research which strives for integrating schemes in exploratory search processes with the use of topic models.

Zusammenfassung

Das Internet birgt schier endlose Informationen. Ein zentrales Problem besteht heutzutage darin diese auch zugänglich zu machen. Es ist ein fundamentales Domänenwissen erfordert, um in einer Volltextsuche die korrekten Suchanfragen zu formulieren. Das ist jedoch oftmals nicht vorhanden, sodass viel Zeit aufgewandt werden muss, um einen Überblick des behandelten Themas zu erhalten. In solchen Situationen findet sich ein Nutzer in einem explorativen Suchvorgang, in dem er sich schrittweise an ein Thema heranarbeiten muss. Für die Organisation von Daten werden mittlerweile ganz selbstverständlich Verfahren des Machine Learnings verwendet. In den meisten Fällen bleiben sie allerdings für den Anwender unsichtbar. Die interaktive Verwendung in explorativen Suchprozessen könnte die menschliche Urteilskraft enger mit der maschinellen Verarbeitung großer Datenmengen verbinden. Topic Models sind ebensolche Verfahren. Sie finden in einem Textkorpus verborgene Themen, die sich relativ gut von Menschen interpretieren lassen und sind daher vielversprechend für die Anwendung in explorativen Suchprozessen. Nutzer können damit beim Verstehen unbekannter Quellen unterstützt werden.

Bei der Betrachtung entsprechender Forschungsarbeiten fiel auf, dass Topic Models vorwiegend zur Erzeugung statischer Visualisierungen verwendet werden. Das Sensemaking ist ein wesentlicher Bestandteil der explorativen Suche und wird dennoch nur in sehr geringem Umfang genutzt, um algorithmische Neuerungen zu begründen und in einen umfassenden Kontext zu setzen. Daraus leitet sich die Vermutung ab, dass die Verwendung von Modellen des Sensemakings und die nutzerzentrierte Konzeption von explorativen Suchen, neue Funktionen für die Interaktion mit Topic Models hervorbringen und einen Kontext für entsprechende Forschungsarbeiten bieten können.

Zur Behandlung dieser Hypothese wird im ersten Schritt ein Überblick der Textvorverarbeitung und des stochastischen Topic Modelings gegeben. Daraufhin werden Modelle der Kognitionspsychologie erläutert, auf denen das Sensemaking beruht. Anschließend werden Interaktionsvorgänge in Problemlösungsprozessen behandelt. Es werden neun Forschungsarbeiten evaluiert, die die explorative Suche unter dem Aspekt des Sensemaking thematisieren und dabei Topic Models oder damit verwandte Verfahren verwenden. Aus der Zusammenführung der Einzelerkenntnisse werden bestehende Beschränkungen des Forschungsfeldes abgeleitet und exemplarisch behandelt.

Die Hypothese konnte in der Evaluation bestätigt werden. Dabei wurden Beschränkungen, wie etwa die fehlende Strukturier- und Vergleichbarkeit von Daten, aufgedeckt und konzeptionell gelöst. Darüber hinaus wurde mit der Erzeugung und Verwendung von Schemata, die Umsetzung des Sensemaking-Modells von Pirolli et al. [1] in Aussicht gestellt.

Die Beobachtungen und Konzepte zeigen die generelle Notwendigkeit auf, das Sensemaking stärker in explorative Analyseprozesse zu integrieren. Die individuelle Strukturierung, Vergleichbarkeit und Externalisierung von Vorstellungen und Wissen scheinen demnach bislang unterschätzte Aspekte komplexer Analyseszenarien zu sein. Die vorliegende Arbeit kann als Fundament für weitere Auseinandersetzungen dienen, die sich der Konzeption eines Frameworks und der Integration von Schemata in explorative Suchprozesse auf der Basis von Topic Models widmen.

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ES Exploratory Search

HCI Human Computer Interaction

ID Interaction Design

IIR Interactive Information Retrieval

IS Information Science

IR Information Retrieval

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

LSI Latent Semantic Indexing

NLP Natural Language Processing

OCR Optical Character Recognition

PAM Pachinko Allocation Model

PCA Principle Componence Analysis

pLSI probablistic Latent Semantic Indexing

SVD Singular Value Decomposition

VSM Vector Space Model

1 Einführung

Der tägliche Gebrauch von immer größer werdenden Datenmengen erschwert das Auffinden konkreter Informationen, als auch das Identifizieren von generellen Inhalten. Ein Nutzer der mit einer neuen Menge von Dokumenten konfrontiert wird möchte in der Regel erst mal verstehen worum es darin eigentlich geht, um anschließend konkrete Dokumente aufzufinden. Dieses Vorgehen wird als Exploration, oder auch Exploratory Search (ES) bezeichnet, da der Nutzer sich schrittweise in das Unbekannte vorarbeitet, um neue Informationen zu entdecken, zu sammeln und sich in Form von Wissen anzueignen. Stellt man sich eine solche Szenerie ganz ohne digitale Datenverarbeitung vor, so würde ein Nutzer vor einem großen Stapel von losen Blättern sitzen die er einzeln durchlesen müsste. Glücklicherweise gehört das der Vergangenheit an und mittlerweile können Suchmaschinen Textdokumente jeglicher Form nach Schlagworten durchforsten. Wenn nun ein konkretes Ziel vorliegt, etwa das Finden eines Textes zu einem ganz bestimmten Sachverhalt, so kann nach allen Begriffen gesucht werden, die einem gerade dazu einfallen. Das Ergebnis ist eine kleine Auswahl von Dokumenten aus der Menge aller Quelle, die gerade zu den gegebenen Schlagworten passen. Wovon die anderen Dokumente handeln und ob sie ebenfalls relevante Informationen beinhalten, bleibt ungewiss.

In dieser Arbeit wird das Erfassen umfassender und komplexer Inhalte behandelt, die sich nur schwer mit heute gängigen booleschen Operatoren der Volltextsuche durchführen lassen. Stattdessen werden stochastische Verfahren verwendet, die aus einer Menge unstrukturierter Texte eine Menge von Themen erfassen und diese in gewichteter Form den Quellen zuweisen. Solche Verfahren werden als Topic Models bezeichnet und finden bereits eine weite Anwendung. Jeder Mensch der Online-Dienste verwendet, wird bereits in Form von Empfehlungsdiensten (*Recommender Systems*) darauf gestoßen sein. Etwa bei Amazone, YouTube oder der Internet Movie Database (IMDb) schlagen sie weiterführende Inhalte vor, die zum bisherigen Nutzerverhalten passen. Topic Models sind bei weitem nicht die einzigen Empfehlungsdienste und nehmen dabei auch nur einen Teil des Prozesses ein, jedoch geben diese Beispiele eine Vorstellung der Möglichkeiten entsprechender Methoden. hier wird nun nicht das indirekte Wechselspiel zwischen Mensch und Maschine, sondern die direkte Interaktion mit stochastischen Verfahren zur Exploration unbekannter und unstrukturierter Textkorpora behandelt. Die direkte Konfrontation mit

der Ausgabe eines Topic Models, die üblicherweise aus Histogrammen und Matrizen besteht, birgt eine bereits aktiv in der Wissenschaft behandelte Herausforderung. Wie kann sich also ein Nutzer einen Sinn über eine große Menge von Relationen zwischen Themen, Termen und Dokumenten bilden? Diese Frage bedarf einer interdisziplinären Betrachtung wie sie die Medieninformatik bietet und vereint Aspekte der Human Computer Interaction (HCI), des Interactive Information Retrieval (IIR) und der Kognitionspsychologie. Eine genauere Erläuterung erfolgt in Abschnitt 1.2.

1.1 Motivation

Durch das exponentielle Wachstum der Datennutzung und -speicherung wächst die Nachfrage nach Methoden der automatischen Datenaufbereitung. Oftmals ist der Dateninhalt nicht bekannt oder das Datenvolumen schlichtweg zu groß, als dass eine manuelle Verarbeitung erfolgen könnte. Als hilfreich hat sich bewährt große Mengen von Daten in Gruppen einzuteilen, die sich in vorab bestimmten oder unbestimmten Attributen ähneln. Das *Clustering* (dt. Gruppieren) findet im *Machine Learning*, der Information Retrieval (IR), der Bildverarbeitung und vielen weiteren Bereichen bereits seit Jahren eine intensive Anwendung. Die LDA ist ein Topic Model, das zum Clustern verwendet werden kann und für die Anwendung auf umfangreiche Datenbestände entwickelt wurde. Die LDA muss nicht mit vorab definierten Testdaten trainiert werden und wird daher den *unsupervised learnings*-Verfahren zugeordnet. Topic Models identifizieren in einem Datenbestand wiederkehrenden Strukturen und beschreiben einzelne Dateneinheiten, etwa ein Textdokument oder ein Bild, anhand einer Gewichtung dieser Strukturen. Durch den historischen Ursprung von Topic Models in der Textanalyse werden die Strukturen als Themen bezeichnet. Die Relationen lassen sich für Suchanfragen, *Recommender Systems* oder die direkte ES verwenden. Der Störanteil zu dem es bei autonomen Clustering-Verfahren kommt ist höher als bei trainierten Systemen, in der ES lassen sich falsch bestimmte Themen durch ein menschliches Urteil schnell begleichen.

Topic Models haben ihren Ursprung in der Latent Semantic Analysis (LSA) und dem Latent Semantic Indexing (LSI), das bereits 1998 von Papadimitriou et al. effektiv für die Bestimmung von Relationen zwischen Themen und Dokumenten angewandt wurde [2]. Das LSI erfolgt durch die Anwendung der Singular Value Decomposition (SVD) auf, als Matrizen behandelte, Tabellen der Termfrequenzen in allen Dokumenten und basiert auf der Annahme, dass Wörter mit ähnlicher Bedeutung in einem gemeinsamen Kontext auftreten. Daraus entwickelte sich die probablistic Latent Semantic Indexing (pLSI), die nicht mehr auf Operationen der linearen Algebra beruht, sondern auf statistischen Modellen. Blei et al. optimierten diese zur LDA, welche in dieser Arbeit hauptsächlich

betrachtet wird. Eine Erläuterung erfolgt in Abschnitt 2.1.2. Die automatische Analyse zum Zweck der thematischen Gruppierung und Relationsbestimmung wird seit Jahren intensiv verwendet. Weitere Anwendungsgebiete umfassen die Dokumentklassifikation, die Zusammenfassung von Texten, das Natural Language Processing (NLP), *Recommender Systems* und weitere Problemstellungen des IR. Als konkretes Beispiel kann der Video-On-Demand Service *Hulu* genannt werden, der Topic Models durch das Benutzerverhalten trainiert, um personalisierte Videoempfehlungen zu generieren[1].

Mittlerweile wurde die LDA mehrfach erweitert und auf Datenströme, visuelle Daten und genetische Informationen angewandt [3]. Die direkte Verwendung von Topic Models durch Endnutzern wird rege erforscht und besonders seit dem Jahr 2014 findet vermehrt eine interdisziplinäre Behandlung aus der Verbindung des IR und der HCI statt. Indikatoren dafür sind unter anderem die Fortschritte entsprechender ES-Tools, die in Abschnitt 3.2 behandelt werden.

Chaney und Blei beschreiben Topic Models in [4] als *unsupervised machine learning methods* und *high-level statistical tools*, für deren Verständnis ein Nutzer zuerst statistische Aussagen in numerischer Form analysieren müsste. Sie entwickeln daraufhin ein Navigationswerkzeug zur Erkundung von Dokumenten, mit dem bedeutende Muster in dem Dokumentkorpus identifiziert werden sollen. Chaney et al. sehen ein Problem bestehender Arbeiten der Topic Model-Visualisierung in der Fokussierung auf einzelne Aspekte. Die Spanne von der Übersicht eines gesamten Dokumentkorpus bis hin zu Details einzelner Dokumente, würde nicht abgedeckt werden [4].

Die Arbeit von Chaney et al. war ein Anstoß für Forschungen zu Endnutzer-tauglichen Anwendung der LDA. Viele Versuche bestanden aus einer Kombination von Funktionen die bereits von Chaney et al. vorgestellt wurden. Jedoch wird seit kurzer Zeit die Vereinigung von menschlicher Sinnstiftung und statistischen Topic Models als wissensgenerierender Prozess behandelt. hu et al. versuchen die Parametrisierung der LDA zu vereinfachen, um sie Nutzern ohne entsprechende Fachkenntnisse besser zugänglich zu machen [5]. Sacha et al. verbinden die Informationsverarbeitungsprozesse des Menschen und des Computers und leiten daraus ein umfassendes Modell der Wissensgenerierung ab. Sie beschränkten sich dabei nicht auf Topic Models, sondern generalisieren über die Verbindung von Datamining und Informationsvisualisierung [6]. In der folgenden Ausarbeitung soll ein ähnlicher Weg eingeschlagen werden und speziell nach Möglichkeiten gesucht werden, die Wissensgewinnung in der ES über Datenmodelle der LDA zu bereichern.

David Blei[2] beschreibt die offenen Fragen der Anwendung von Topic Models in einem

[1] `http://tech.hulu.com/blog/2011/09/19/recommendation-system/` , 16.12.2014

[2] David M. Blei ist einer der führenden Forscher im Bereich des Topic Modelings und hat unter anderem die weit verbreitete Latent Dirichlet Allocation (LDA) entwickelt.

einführenden Artikel über die Statistik des Topic Modelings wie folgt:

> *„Another promising future direction for topic modeling is to develop new methods of interacting with and visualizing topics and corpora. Topic models provide new exploratory structure in large collections — how can we best exploit that structure to aid in discovery and exploration? [...] A further problem is how to best display a document with a topic model. At the document level, topic models provide potentially useful information about the structure of the document. Combined with effective topic labels, this structure could help readers identify the most interesting parts of the document. Moreover, the hidden topic proportions implicitly connect each document to the other documents [...]. How can we best display these connections? What is an effective interface to the whole corpus and its inferred topic structure? These are user interface questions, and they are essential to topic modeling. Topic modeling algorithms show much promise for uncovering meaningful thematic structure in large collections of documents. But making this structure useful requires careful attention to information visualization and the corresponding user interfaces.“* [3]

Demnach müssen für die Strukturen von Topic Models neue Formen der Interaktion und Visualisierung gefunden werde. Daher setzt er sich in aktuellen Arbeiten für die interdisziplinäre Behandlung von Topic Model im Bereich der HCI und der *Digital humanities*[3] ein. Das umfasst die Verteilung von Themen in Textkorpora und in einzelnen Dokumenten, als auch sich davon ableitende Verbindungen. Als ein konkretes Beispiel nennt Blei die Darstellung von Relationen und das Bezeichnen von Themen.

Im klassischen IR wird meistens von einem Nutzer ausgegangen, der sich über seine Ziele im Klaren ist. Dazu im Gegensatz steht die Motivation der ES, die einem Nutzer das für ihn unbekannte erfassbar machen möchte. Die folgenden potentiellen Anwendungen von Topic Models zeigen deren vielfältige Einsetzbarkeit auf:

- Bei der Einarbeitung in ein Sachthema, wie es im Studium und auch im Beruf häufig notwendig ist, wird in der Regel mit mehreren Quellen gearbeitet. Digitalisierte Bücher gewinnen an Popularität und werden nicht selten mit Webressourcen oder wissenschaftlichen Publikationen verbunden. Mit einem Topic Model ließe sich der Inhalt der Quellen schnell in seine thematischen Einheiten separieren, sodass sich der lernende auf die relevanten Abschnitte konzentrieren und problemlos zwischen den Quellen wechseln kann.

- Notizsysteme wie *Evernote* oder *Microsoft OneNote* erfreuen sich steigender Beliebtheit. Mittels Topic Model könnten lose Notizen zu Sinneinheiten zusammengeführt

[3] `http://goo.gl/NDP611` , 8.01.2015

werden, sodass sich wiederkehrende Gedankengänge identifizieren und präzisieren lassen.

- Größere Unternehmen verwenden viel Aufwand auf die Verwaltung eigener Patente. Für Projekte und für die Formulierung neuer Patente muss auf bestehende Einträge zurückgegriffen werden. Topic Model können die Verwaltungssysteme ergänzen und Ingenieuren helfen, bestehende Arbeiten im Rahmen neuer Projekte zu finden.

- Experimentell wurden Topic Models bereits in der Literaturwissenschaft verwendet um wiederkehrende Motive und Themen zu identifizieren. Beispielsweise konnte Jeff Drouin einen Bedeutungswandel der Liebe in Marcel Prousts hauptwerk *À la recherche du temps perdu*[4] beobachten. Ein weiteres Beispiel sind die exemplarischen Datenbanken der Webversion von *Serendip*. Dort kann man selbst nach Mustern in Shakespeares hauptwerk suchen[5].

1.2 Positionierung im wissenschaftlichen Feld

Topic Models ermöglichen die explorative Datenanalyse unstrukturierter Textquellen mit dem Ziel der Sinnstiftung (engl. *Sensemaking*) und die explorative Suche nach relevanten Inhalten. Das Sensemaking bezeichnet die Bedeutungsfindung aus Erfahrungen und aufgenommenen Informationen und wird in der Human Computer Interaction (HCI) als auch der *Information Science* behandelt. In die HCI wurde es von Russel et al. als effizientes Arbeiten mit externen Informationen durch die Externalisierung von Wissensrepräsentationen eingeführt [7]. Pirolli et al. beschreiben das Ziel der daran anschließenden Forschung als das Finden von Methoden, Wissen extern zu Formen und ein Verständnis des interaktiven Sensemaking-Prozesses zu erlangen. Die Verbindung der visuellen Datenanalyse und der Wissensgenerierung wurde bereits durch verschiedene Modelle beschrieben. Aufgrund der wachsenden technologischen Möglichkeiten entwickelt sich dieser Prozess stetig weiter. Die meisten Erkenntnisse der menschlichen Informationsaufnahme und -verarbeitung entstammen der Kognitionspsychologie, deren Verwendung zu umfassenderen Modellen führt. [1, 6]

Auf Seiten der maschinellen Datenverarbeitung werden dem Information Retrieval (IR) entstammende Methoden verwendet. Das IR kann als die Wissenschaft, Technik und Praxis des Suchen und Finden von Informationen verstanden werden. Stock erkennt in der IR eine Schnittmenge aus der Information Science (IS) und der Informatik [8]. Manning hingegen sieht den Ursprung in der NLP, von der eine Wegentwicklung durch die Anwendung

[4] `http://www.proustarchive.org/?q=node/35` , 9.01.2015
[5] `http://vep.cs.wisc.edu/serendip/` , 11.12.2014

vorwiegend statistischer Methoden stattfand [9]. Das IR umfasst verschiedenste Methoden zur Identifikation und Auffindung von, für einen Nutzer relevanten Informationen und Dokumenten. Topic Models sind stochastische Modelle zur Auffindung von *Themen* in Dokumentkollektionen. Auf die Resultate eines Topic Models lassen sich algebraische Methoden des VSM anwenden um Gemeinsamkeiten zwischen einzelnen Elementen zu bestimmen. Oftmals beschränkt sich die Anwendung auf die Bestimmung von Relationen zwischen Dokumenten. Des Weiteren werden simple NLP Verfahren verwendet, die durch Vorverarbeitung der Textquellen die Ergebnisse des Topic Models optimieren sollen. Dabei wird darauf geachtet, möglichst Sprachunabhängig zu operieren. Da in der IR bestehende Informationen zurückgewonnen werden, wird sie von der *Knowledge Discovery* und dem Sensemaking unterschieden, bei denen völlig neue Erkenntnisse und Bedeutungen gesammelt werden sollen. Ingwersen formulierte wesentliche Problemstellungen der IR, die für die Exploration und das technologische Fundament der Sinnstiftung äußerst Relevant sind. Dazu gehört die Beschreibung des Inhalts eines Mediums (*Aboutness*), die Vergleichbarkeit von thematischen Informationen und die Relevanz von Inhalten für den Nutzer [10][6].

Besonders der letzte Punkt verlangt wegen seiner Abhängigkeit vom jeweiligen Nutzer die Perspektive der HCI. Die HCI und das IR finden eine Verbindung in dem Interactive Information Retrieval (IIR). Ingwersen beschreibt es als *"the interactive communication processes that occur during the retrieval of information by involving all the major participants in information retrieval (IR), i.e. the user, the intermediary, and the IR system."* [10]. Alternative Bezeichnungen überschneiden sich sehr stark, sodass sich weitere Begriffe wie *human-computer Information retrieval*[7] finden, die sich in ihrer Definition zu gering unterscheiden als sie hier zu unterscheiden, sodass im Folgenden der Ausdruck Interactive Information Retrieval (IIR) verwendet wird.

Dörk et al. behandeln den Informationsfindungsprozess unter der Metapher des *Information Flaneur*, der durch eine Datenstadt flaniert und sich versucht ein Bild von ihr zu machen. Dabei spielen Orientierung, Serendipizität und die Prozesse der Exploratory Search eine wichtige Rolle, mit denen sich bewusste und unbewusste Vorgehensweisen und Ziele der Informationssuche behandeln lassen. Ihre Forschung ordnen Dörk et al. der Kognitionspsychologie, den Bibliothekswissenschaften, der HCI und der Informationsvisualisierung zu. [11]

Eine auf die Anwendung explorativer Datenanalyse in den humanwissenschaften ausgeweitete Definition gibt Blei mit dem Begriff *Digital humanities*[8] und in ähnlicher Form

[6] Kapitel 3.1 Essential Issues of IR

[7] `http://www.asis.org/Bulletin/Jun-06/marchionini.html` , 12.02.2015

[8] Siehe Abschnitt 1.1.
`goo.gl/NDP61l` , 8.01.2015

auch Rogers, der den Begriff *Digital Methods* prägte [12].

Es ist eine Vielzahl von Einflüssen verschiedener Disziplinen zu erkennen, die sich am besten mit einer Beschränkung auf spezielle Aspekte der HCI und der IIR verstehen lassen. Die darin behandelten Themen sind ebenfalls umfassend und reichen von der Modellierung der Wissensgenerierung bis hin zu der Erschließung neuer Anwendungsgebiete. In dieser Arbeit werden bestehende und potentielle Anwendungsbereiche als Motivation gesehen und die Prozesse der Informationsaufnahme als theoretische Grundlage verwendet, um den potentiellen Nutzen von Topic Models in der explorativen Suche und Datenanalyse, mit dem Ziel der Sinnstiftung zu analysieren.

1.3 Ziel und Vorgehen

In der Betrachtung bestehender Tools zur explorativen Analyse von Textkorpora mittels Topic Models, fiel eine Wiederholung grundlegender Funktionen auf. Das explorative Suchen wird momentan vorwiegend durch algorithmische Neuerungen behandelt, die nicht in ein umfassendes Konzept gefügt werden, sodass sich einzelne Arbeiten schwer miteinander vergleichen und vereinen lassen. Daraus ergibt sich die Vermutung einer gemeinsamen und grundlegenden Problematik in der Konzeption entsprechender Tools ab. Die gezielte Gegenüberstellung mehrerer Arbeiten kann ein solches Problem konkretisieren und eine mögliche Ursache identifizieren. Das Sensemaking ist ein wesentlicher Bestandteil der explorativen Suche und Analyse und wird von vielen Autoren entsprechender Publikationen als umfassendes Ziel beschrieben. Daraus leitet sich die Hypothese ab, dass Modelle des Sensemaking den Explorationsprozess über Strukturen von Topic Models verbessern und einen Kontext für entsprechende Forschungsarbeiten bieten können. Dafür dürfen Explorationsszenarien nicht mehr rein algorithmisch behandelt werden und müssen die menschlichen Informationsverarbeitungs-, Wissensaneignungs- und Problemlösungsprozesse integriert. Das Ziel dieser Arbeit ist es nicht, ein konkretes Modell oder ein umfassendes Framework zu schaffen, sondern Beschränkungen in bestehenden Forschungsarbeiten aufzudecken und deren Lösung in Aussicht zu stellen. Blei et al. beschreiben die Aussichten der Interaktion mit Topic Models wie folgt [3]:

> *„Topic modeling algorithms show much promise for uncovering meaningful thematic structure in large collections of documents. But making this structure useful requires careful attention to information visualization and the corresponding user interfaces.“*

Die Visualisierung und die Konzeption von Interfaces sind wichtige Aspekte in der Anwendung von Topic Models, jedoch muss der Behandlungsradius weiter gesteckt werden, um kognitive Prozesse zu integrieren.

In Abschnitt 2 werden Grundlegende Methoden und Modelle erläutert, die für den Vergleich bestehender Arbeiten in Abschnitt 3 verwendet werden. Einige beschriebene Methoden werden nicht direkt in die Spezifikation der Evaluation einfließen. Sie sind dennoch wichtig, um ein umfassende Verständnis zu erhalten und in Abschnitt 4 die gesammelten Erkenntnisse weiter auszuführen.

Das genauere Vorgehen sieht wie folgt aus. In Abschnitt 2.1 werden Verfahren der Textverarbeitung, der Stochastik und des VSM vorgestellt. Dazu gehören einfache und etablierte Methoden der NLP und des IR zur Vorverarbeitung von Texten, mit denen die Ergebnisse der LDA optimiert werden. Anschließend werden die Vorteile und Funktionsweisen von Topic Models im Allgemeinen und der LDA im speziellen beschrieben. Dabei wird nicht in die Details der Parametrisierung und der Dirichlet A-priori-Wahrscheinlichkeit eingegangen, durch die sich die LDA von der pLSI unterscheidet. Stattdessen wird die grundlegende Herleitung erläutert, die die Anwendung von Topic Models und die Interpretation der Ergebnisse ermöglicht. Für die LDA wurde sich hier entschieden, da sie als Verallgemeinerung der pLSI zu generell besseren Ergebnissen führt. Darüber hinaus sprechen die weite Verbreitung in der Praxis und die Anerkennung in der Forschung dafür. Nach der Beschreibung der Topic Models erfolgt eine kurze Erläuterung des VSM, das verwendet wird um die Ergebnisse eines Topic Models in Dokumentrelationen zu überführen.

In Abschnitt 2.2 werden Modelle des menschlichen sensorischen Registers, des Kurzzeitgedächtnis und des Langzeitgedächtnis beschrieben. Diese Modelle werden in der HCI und im Interaction Design (ID) häufig angewandt und helfen auch hier die Informationsaufnahme, die Wissensgenerierung und die Sinnstiftung zu verstehen. Das Aneignen neuer Kenntnisse wird anschließend in einem Überblick der verschiedenen Lerntypen genauer betrachtet, wobei besonders das bedeutungserzeugende Lernen im weiteren Verlauf eine wichtige Rolle spielt.

In Abschnitt 2.3 wird der Suchprozess als Teil der IIR beschrieben. Außerdem werden praktische Aspekte der Konzeption von Explorations-Tools behandelt. Konkret zählt dazu der Umgang mit verschiedenen Perspektiven auf vorliegende Daten und deren Aufteilung auf mehrere Ansichten. Da ein Topic Model Relationen zwischen den Elementen eines Textkorpus generiert, wird die Visualisierung von Relationen im Text Mining speziell behandelt. Abschließend wird überprüft wie sich die Ergebnisse eines Topic Models interpretieren lassen.

In Abschnitt 3 erfolgt eine Gegenüberstellung von neun Tools der explorativen Datenanalyse und Suche mit Hilfe von Topic Models. Dazu werden in Abschnitt 3.1 Spezifikationen für die Auswahl von Arbeiten und Kriterien der Evaluation bestimmt, die in die drei Gruppen *Algorithmus*, *Interface* und *Kognition* unterteilt werden. Diese wer-

den in jeweils gleicher Reihenfolge in Abschnitt 3.2 verwendet, um jedes Tool einzeln zu analysieren. In Abschnitt 3.3 erfolgt die Gegenüberstellung der einzelnen Tools und die Zusammenführung der erkannten Probleme. Die neuen Erkenntnisse werden verwendet um in Abschnitt 4 einen Ausblick auf mögliche Ausführungen zu bieten, die als potentielle Weiterentwicklungen zu verstehen sind und nicht als konkrete Lösungen. Eine Zusammenfassung und ein abschließendes Fazit wird in Abschnitt 5 gezogen. Die Bezeichnungen der verwendeten Akronyme, Literatur-, Tabellen- und Abbildungsreferenzen finden sich in den entsprechenden Verzeichnissen deren Ordnung dem Inhaltsverzeichnis zu entnehmen sind.

2 Methoden und Grundlagen

In diesem Kapitel werden Grundlagen der Textverarbeitung, des Topic Modeling, der Wissensaufnahme im Rahmen der Kognitionspsychologie und des IIR erläutert, die für die Behandlung von ES-Tools in Abschnitt 3 benötigt werden. Dabei wird besonders im Bereich der Kognitionspsychologie und des IIR über das zwingend notwendige hinaus gegangen, um den Blick für Möglichkeiten zu öffnen den momentanen Forschungsstand weiterzuentwickeln.

2.1 Text Analyse durch stochastische Verfahren

Topic Model sind stochastische Verfahren die auf Worthäufigkeiten basieren und keinerlei semantisches Wissen besitzen. Ein Vorteil von stochastischen Methoden in der Textanalyse ist, dass sie im Gegensatz zu grammatikalischen NLP-Verfahren gänzlich sprachunabhängig sind. Dadurch entstehende Schwächen können durch statistische Filter und sprachabhängige Vorverarbeitung ausgeglichen werden. Diese werden nun beschrieben und zu den Grundlagen des Topic Modelings und der LDA überführt, deren Anwendung zur Berechnung von Dokumentähnlichkeiten in Abschnitt 2.1.3 erläutert wird. Abschließend werden die Einschränkungen und in Aussicht stehende Entwicklungen von Topic Models beschrieben.

2.1.1 Textvorverarbeitung

Für die Anwendung von Topic Models auf unformatierte Texte ist es zweckmäßig ein *Preprocessing* (dt. Vorverarbeitung) der Quellen vorzunehmen. Einige Schritte wie die Überführung von Dokumenten in eine maschinell verarbeitbare Form sind voraussetzend, während andere Schritte zu der Optimierung der Ergebnisse führen. Es wird versucht möglichst viele nicht benötigte Informationen zu eliminieren. Der in *Instances* (dt. Einheiten) umgewandelte Text lässt sich maschinell effizient verarbeiten und ist nicht auf die menschliche Lesbarkeit ausgerichtet. Die Instances bleiben weiterhin mit den originalen Quellen verknüpft.

Digitale Texte werden heutzutage üblicherweise als PDF gespeichert, um die Portabilität und eine korrekte Darstellung zu gewährleisten. Mittels *Optical Character Recognition*

(OCR) lassen sich auch gerasterte Texte in Zeichenketten umformatieren. Die Zerteilung von Texten in einzelne Wörter wird Tokenisierung genannt und erfolgt für lateinische Sprachen, unter Beachtung von Spezialfällen, durch die einfache Trennung an den Positionen der Leerzeichen. Da ein Topic Model im Gegensatz zum NLP keine strukturelle Analyse vornimmt, können Sonderzeichen wie Punkte, Kommata und Anführungszeichen entfernt werden. Um Wörter möglichst weit zu vereinheitlichen werden alle Buchstaben in Kleinbuchstaben umgewandelt. Anschließend werden alle Wörter auf ihr *Lexem* reduziert. Damit ist eine Bedeutungseinheit gemeint, die von den verschiedenen Wortformen unabhängig ist. Grammatikalische Informationen, wie etwa zeitliche Relationen, gehen dadurch verloren. Etwa werden die Wörter *schreiben, schrieb* und *schrieben* auf *schreiben* zurückgeführt. Eine mögliche Umsetzung ist das Reduzieren eines Wortes auf seinen Wortstamm. Dieser Vorgang basiert auf festen grammatikalischen Regeln der im Text verwendeten Sprache. Ungenauigkeiten entstehen durch *Overstemming* und *Understemming*. Zum einen können Wörter zu stark reduziert werden, sodass zwei unterschiedliche Wörter auf den gleichen Stamm zurückgeführt werden. Andererseits können Wörter gleichen Lexems auf unterschiedliche Stämme zurückgeführt werden. Beispielsweise führt *schreiben, geschrieben, schrieben* zu den Wortstämmen *schreib, geschrieb, schrieb*. Diese Ungenauigkeiten halten sich allerdings in Grenzen, sodass sie in den meisten Anwendungen hingenommen werden. Alternativ kann per *Lemmatisierung* mit einer lexikalischen Referenz die Grundform eines Wortes bestimmt werden. Dadurch lassen sich die im Stemming auftretenden Ungenauigkeiten umgehen. Dafür muss jedoch eine sprachabhängige und umfangreiche Referenz vorliegen, die aufwändiger zu bestimmen ist als die Regeln eines Stemmingalgorithmus. [8]

Wörter von denen ausgegangen wird, dass sie wegen ihres häufigen Vorkommens keine relevante Bedeutung für einzelne Texte haben können, werden vor der weiteren Verarbeitung gelöscht. Diese Wörter werden *stop words* (dt. Stoppwörter) genannt und umfassen üblicherweise Artikel, Konjunktionen und Präpositionen[1]. Stop word-Listen umfassen etwa 200 Wörter, wobei der Umfang zwischen den Anwendungssprachen und den anschließend verwendeten Methoden variieren. Bei der Anwendung von stop word-Listen nach der Lexembildung muss darauf geachtet werden, dass diese ebenfalls dem Wortstamm entsprechen. Das Stemming als auch das Filtern von stop words ist Sprachabhängig. Low-level Verfahren wie sie hier beschrieben wurden sind einfach an einzelne Sprachen anpassbar und führen zu messbar besseren Ergebnissen. Sie finden daher eine weite Anwendung in der Praxis der Textanalyse. [13, 14, 9]

Für die weitere Verarbeitung werden den einzelnen Worten integer Indizes zugewiesen. Mit diesen *Features* kann effizienter Operiert werden als mit Zeichenketten. Aus ihnen

[1]Beispiel für *stop words*: der, und, in, von, nicht

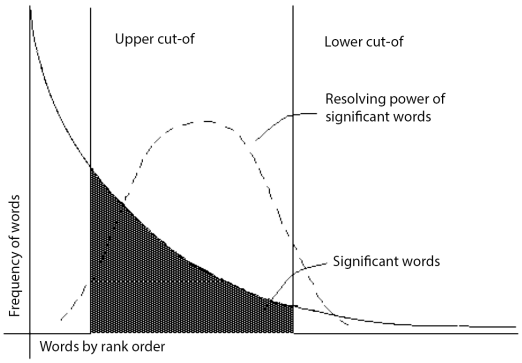


Abbildung 2.1: häufigkeit und Signifikanz von Wörtern in einem Dokument nach Luhn (1985). *Quelle: Stock(2007), S. 319*

können Featurebags oder Bag-of-Words (ungeordnete Mengen) generiert werden, welche die Frequenzen der Features im gesamten Datenkorpus als auch den einzelnen Dokumenten enthalten.

Luhn beschreibt eine zu den stop word-Listen alternative Filterung, die auf der Annahme beruht, dass hochfrequente und niederfrequente Wörter eine geringe Bedeutung für die Charakteristik eines Textes haben. Abbildung 2.1 zeigt das Verhältnis von Wortfrequenz und Wortrang in Form einer hyperbel. Die Frequenzen bezeichnen die Vorkommenshäufigkeit und der Rang die Sortierung der Wörter nach ihrer Frequenz. Die gestrichelt angedeutete Normalverteilung entspricht der *Resolving power of significant words*, eine von Luhn zugewiesenen Relevanz der Wörter. Daraus folgert er die beiden eingezeichneten *cut-offs*, von denen besonders der *upper cut-off* seine Berechtigung hat, da er zu einer starken Reduktion der Wortmenge und folglich einer geringeren Rechenlast führt. Der eigentliche Zweck ist allerdings die Optimierung der zu findenden Themen, die in Abschnitt 2.1.2 behandelt wird. Im Gegensatz zu stop word-Listen ist die statistische Filterung sprachunabhängig. [15, 8]

2.1.2 Topic Modeling mit der Latent Dirichlet Allocation

Für das Auffinden von Informationen ist die Volltextsuche noch immer die weit verbreitetste Methode. Grund ist die schnelle Umsetzbarkeit, sodass große Datenbanken effizient indiziert werden können und lokale Daten von Endnutzern ad-hoc durchsucht werden können. Mit der Volltextsuche werden gezielt Informationen in möglicherweise unbekannten und unformatierten Daten gesucht. Problematisch ist, dass die Sucheingabe nur exakte Ergebnisse findet. Für den Fall, dass ein Nutzer nur eine grobe Vorstellung seines Ziels hat, werden einfache semantische Verbindungen mit einbezogen und somit auch nach Synonymen gesucht. Des Weiteren gibt es das *approximate string matching* (dt. *unscharfe Suche* oder *fuzzy Suche*), bei dem auch ähnlich geschriebene, klingende und vertauschte

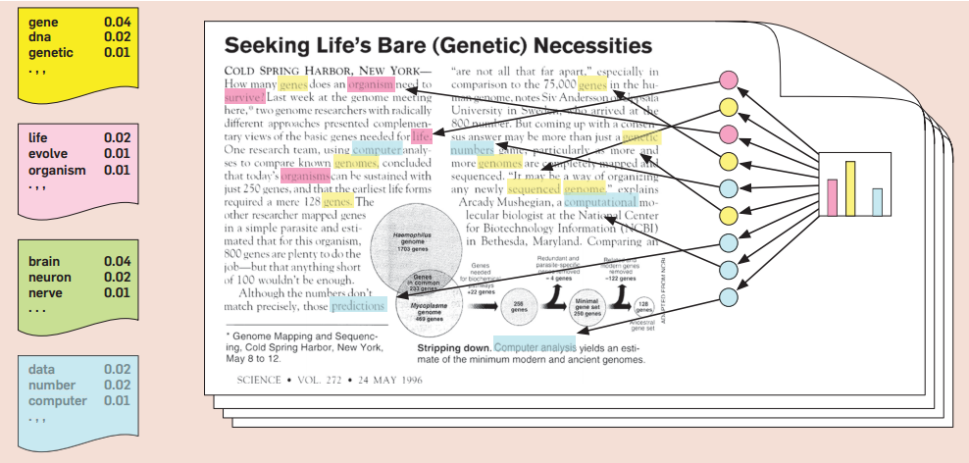


Abbildung 2.2: Verbildlichung der Funktionsweise der LDA. *Quelle: Blei (2012)*

Zeichenketten einbezogen werden. [14, 8]

Wenn der Suchende nur eine vage Vorstellung seines Zieles hat, oder gar erste eine Übersicht der vorliegenden Daten erhalten möchte, befindet er sich in einer ES-Situation, die sich nur schwer mit einer Volltextsuche lösen lässt. Ein besseres Verständnis lässt sich durch die Abstraktion konkreter Daten auf umfassende Inhaltsbereiche lösen. Das Identifizieren von Themen mittels stochastischer Modelle hat eben dieses Ziel. Topic Models finden abstrakte Themen in Dokumentkollektionen indem sie davon ausgehen, dass Terme die außergewöhnlich häufig in wenigen Dokumenten vorkommen ein spezielles Thema bilden. Die LDA wurde 2003 von Blei, Ng und Jordan beschrieben und ist noch immer das am weitesten verbreitete Topic Model. Abbildung 2.2 zeigt die Funktionsweise der LDA in vereinfachter Form. Darin sind vier Themen farblich gekennzeichnet. Auf der linken Seite sind die jeweils wichtigsten Begriffe und deren Wahrscheinlichkeiten aufgelistet. Die Pfeile und Färbungen im Text zeigen die verteilte Zuweisung von Wörtern zu den vier Themen. Das Diagramm auf der rechten Seite beschreibt die Vorkommensanteile der Themen im betrachteten Dokument. Thema *Grün* ist nicht zu sehen, da ihm keine Wörter des Dokuments zugewiesen sind.

Um die LDA besser zu begreifen wird ein, zugegebenermaßen etwas merkwürdig wirkender, Perspektivenwechsel zur Sicht des Modells gemacht. Dazu wird nun beschrieben, wie sich ein Modell die Erzeugung eines Dokumentes *vorstellt*. Zu Beginn wird vom Nutzer eine feste Zahl von Themen festgelegt und jedem Thema entsprechende Wörter mit zufälligen A-Priori-Wahrscheinlichkeiten zugewiesen. Die Wahrscheinlichkeiten einzelner Wörter und Themen wird im Folgenden auch als *Gewicht* bezeichnet. Beispielhaft definieren wir die zwei Themen *Genetics* und *Evolution* mit den folgenden Begriffen und Wahrscheinlichkeiten [2]:

[2]Das Beispiel ist, abgesehen von den Wahrscheinlichkeiten, [3] entnommen und zeigt ein echtes Resultat

1. *Genetics*: (0.25) human, (0.20) genome, (0.20) dna, (0.15) genetics, (0.10) genes, ...
2. *Evolution*: (0.30) evolution, (0.25) evolutionary, (0.20) species, (0.11) organism, (0.10) life, ...

Wenn nun ein Dokument erzeugt wird, das zu 90% aus *Genetics* und zu 10% aus *Evolution* besteht, so werden zufällig Wörter aus diesen beiden Mengen entsprechend ihrer Gewichtung gewählt. Der fertige Text könnte dann etwa heißen *„human dna evolution genome genes human life“*. Die Wahl der Wörter ist durch die Gewichtung bestimmt und erfolgt ansonsten zufällig anhand der *Dirichlet-Verteilung*[3], weshalb die Reihenfolge der Wörter irrelevant ist. Das einfache Zählen von Worten bezeichnet man als *bag of words*. Eine *bag* (dt. *Multimenge*) ist eine ungeordnete Menge in der Elemente mehrfach vorkommen können. Die Wortkette *„life genes human human evolution dna genome“* führt somit zu dem gleichen Ergebnis wie das vorherige Beispiel. [3, 16] Betrachten wir einen Korpus von D Dokumenten d mit einem Vokabular W der Wörter w und gehen davon aus, dass K Themen t darin enthalten sind. Die Zahl der Themen ist für die LDA und auch die meisten anderen Topic Models festzulegen. Optimale Ergebnisse sind zu erwarten, wenn die Zahl der zu findenden Themen den tatsächlich behandelten Themen entspricht. In der Regel ist die Zahl der tatsächlichen Themen jedoch nicht bekannt und es gibt auch keine generelle beste Wahl für K. Wenn K zu groß ist fällt die Aufteilung zu detailliert aus und wenn K zu klein gewählt wurde, kann es zu Vermengung von Inhalten in einem t kommen.[17] Nun wird für alle d jedem w ein zufälliges $t \in T$ zugewiesen. Dadurch entsteht bereits die Struktur der Themen- und Wortverteilung, die jedoch noch keine semantische Aussage besitzt. Für jedes Wort eines jeden Dokumentes werden die folgenden zwei Wahrscheinlichkeiten neu berechnet:

1. Die Bedingte Wahrscheinlichkeit $p(t|d)$, dass das Thema t in Dokument d vorkommt. Sie ergibt sich aus der Anzahl von Worten im Dokument d, die dem Thema t zugewiesen sind, im Verhältnis zu der gesamten Anzahl von Worten in d.
2. Die Bedingte Wahrscheinlichkeit $p(w|t)$, dass das Wort w im Thema t vorkommt. Sie ergibt sich aus der Zahl der Zuweisungen von w zum Thema t im gesamten Korpus, im Verhältnis zum gesamten Vorkommen des Wortes w.

Die Wahrscheinlichkeiten ergeben sich aus einfachen Aufzählungen von Worten, Dokumenten und Themen, die in entsprechende Universen, beziehungsweise Kontexte gesetzt werden. Die Berechnungen basieren auf der Annahme, dass alle Verteilungen, bis auf die

der LDA über 17.000 Artikel der Zeitschrift *Science*.

[3] Die Dirichlet-Verteilung gibt die Wahrscheinlichkeit des Auftretens einer Wahrscheinlichkeitsverteilung an.

vom aktuell betrachteten Wort w, korrekt sind. Durch das Produkt ergibt sich die neue Wahrscheinlichkeit $p(w|t)$, dass w zu t gehört:

$$p(w|t) = p(t|d) * p(w|t) \tag{2.1}$$

Diese Schritte werden über den ganzen Korpus wiederholt, jede Iteration ergibt eine bessere Näherung an eine optimale Verteilung der Themen, bis diese konvergieren. Der Vorgang kann aus Performanzgründen auch schon vor dem Erreichen eines statischen Zustandes angehalten werden, wenn ein ausreichend genaues Ergebnis zu erwarten ist. Die bedingten Wahrscheinlichkeiten $p(t|d)$ und $p(w|t)$ können als Matrizen ausgegeben werden:

$$\begin{aligned} \phi \in \mathbb{R}^{K \times W} \quad & \text{mit} \quad \phi_{i,j} = p(w_i|t_j) \\ \theta \in \mathbb{R}^{D \times K} \quad & \text{mit} \quad \theta_{i,j} = p(t_i|d_j) \\ \theta_i \in \mathbb{R}^{K} \quad & \text{als} \quad \text{Dokumentvektor} \end{aligned} \tag{2.2}$$

Alle Darstellungen erfolgen daher auf den beiden Gewichtungsmatrizen ϕ für die Relation Themen $\times$ Termen und θ für die Relation Dokument $\times$ Themen [16]:

$$\phi = \begin{pmatrix} p(w_1|t_1) & \cdots & p(w_1|t_i) \\ \vdots & \ddots & \vdots \\ p(w_j|t_1) & \cdots & p(w_i|t_j) \end{pmatrix} \qquad \theta = \begin{pmatrix} p(t_1|d_1) & \cdots & p(t_1|d_i) \\ \vdots & \ddots & \vdots \\ p(t_j|d_1) & \cdots & p(t_i|d_j) \end{pmatrix} \tag{2.3}$$

Der Inhalt eines Themas ergibt sich durch seine Termgewichtung, daher erfolgt dessen Interpretation in der Regel durch die Betrachtung der x Terme höchsten Ranges. Dabei ist zu beachten, dass das stochastische Modell keine tatsächliche Kenntnis über die inhaltliche Deutung hat und auch unpassende Begriffe einfließen können.

Der Name *latent dirichlet allocation* entstammt daher, dass die Terme eines Dokuments mittels der *Dirichletverteilung* verschiedenen Themen zugewiesen werden (engl. *to allocate*). *Latent* (verborgen) sind alle Variablen der verwendeten Verteilung mit Ausnahme der Wörter, die die beobachteten Variablen sind. Topic Models wurden ursprünglich für die Anwendung auf Textkorpora entwickelt, daher hat sich der Begriff der *topics* bzw. Themen durchgesetzt. Mittlerweile werden die Methoden auch auf Bildmaterial (*spacial LDA*), Videos und andere Datentypen angewandt. [3]

2.1.3 Dokumentähnlichkeit

Für die Bestimmung der Ähnlichkeit zwischen Dokumenten hat sich das Vector Space Model (VSM) bewährt. Dazu wird ein Vektorraum mit K Dimensionen gebildet, in dem jede Dimension einem Thema entspricht. Für jedes Dokument kann die Gewichtung der einzelnen Themen verwendet werden um einen Vektor zu bilden, der in das VSM eingefügt wird. Der Grad der Ähnlichkeit zweier Dokumente wird durch die *Kosinus-Ähnlichkeit* bestimmt. Es ergibt sich aus dem zwischen zwei Vektoren eingeschlossenen Kosinus-Winkel:

$$\cos\left(\angle(\vec{a},\vec{b})\right) = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2}\cdot\sqrt{\sum_{i=1}^{n}(b_i)^2}} \tag{2.4}$$

Aus $cos(0) = 1$ folgt, dass ein hoher Kosinus-Wert auf eine Ähnlichkeit der Dokumente schließen lässt. Da ausschließlich der zwischen zwei Vektoren eingeschlossene Winkel beachtet wird, haben die Vektor-Beträge keinen Einfluss auf den Grad der Relation. Das VSM ist nicht der LDA zu eigen, sondern wurde zuvor in Verbindung mit dem pLSI und einfachen Termgewichtungen wie dem *Tf-idf-Maß* angewandt. Die LDA und die pLSI gruppieren einzelne Terme zu Themen, reduzieren die Dimensionalität des VSM und werden deshalb oftmals als eine Form der Principle Componence Analysis (PCA) beschrieben. Eine Erläutertung des Tf-idf-Maß findet sich im Anhang in Abschnitt 5.3 mit Formel 5.1. Abbildung 2.3 ist ein Beispiel von Manning et al. für ein einfaches VSM. Es zeigt den Vergleich von Dokumenten durch Terme und lässt sich direkt auf ein VSM für Themen übertragen. Daher wird nun davon ausgegangen, dass die beiden Achsen zwei Themen mit den Bezeichnungen *car* und *insurance* entsprechen. Die eingezeichneten Vektoren d_1, d_2 und d_3 repräsentieren drei Dokumente, die nach ihrer jeweiligen thematischen Gewichtung ausgerichtet sind. In d_1 wird hauptsächlich das Thema *car* und in d_3 das Thema *insurance* behandelt. Der vierte Vektor q ist eine Suchanfrage, die etwa *„car insurance“* lauten könnte und damit beide Themen zu gleichen Anteilen beinhaltet. Wie man sieht ergibt sich die größte Kosinus-Ähnlichkeit zwischen q und d_2, sodass auf die Suche q als Ergebnis das Dokument d_2 resultiert. [18, 9]

Formel 2.5 beschreibt die wiederholte Anwendung der Kosinus-Ähnlichkeit auf alle Spaltenvektoren von θ. Dadurch entsteht eine symmetrische Matrix, in der die erwarteten Relationsgrade zwischen allen Dokumenten erfasst sind.

$$\sigma_D = \begin{pmatrix} cos(\theta_i,\theta_i) & \cdots & cos(\theta_1,\theta_D) \\ \vdots & \ddots & \vdots \\ cos(\theta_D,\theta_1) & \cdots & cos(\theta_D,\theta_D) \end{pmatrix} \qquad mit \qquad 1 \le i \le D \tag{2.5}$$

$$cos(\theta_i,\theta_j) = cos(\theta_j,\theta_i)$$

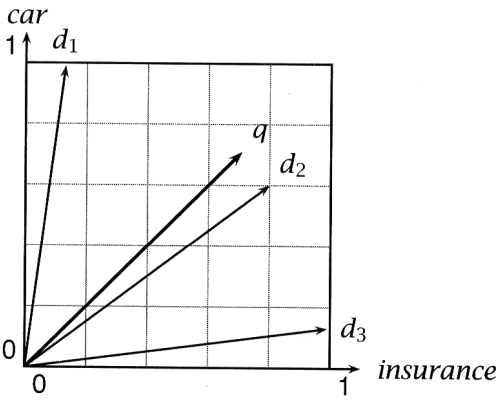


Abbildung 2.3: Ein zweidimensionaler Vektorraum mit der Einordnung dreier Dokumente und einer Suchanfrage q. *Quelle: Manning und Schütze (1999), S. 540*

2.1.4 Einschränkungen der Aussagekraft von Topic Models

Topic Models basieren auf verallgemeinernden Annahmen, die nicht der Realität entsprechen, die Ausgaben verfälschen und zu Informationsverlusten führen. Die Resultate von Topic Models können daher nicht als absolut richtig erfasst werden, sondern müssen mit Bedacht interpretiert werden. Die erste Annahme entsteht durch die Verwendung von *bag of words*, demnach die Reihenfolge von Worten unwichtig ist. Diese Annahme ist allerdings für fast keine Sprache korrekt. Zwischen *"Der Hund beißt den Mann"* und *"Der Mann beißt den Hund"* trifft ein Topic Model keine Unterscheidung. Es erkennt ausschließlich, dass der Inhalt von *Hund, beißt* und *Mann* handelt. Für konkrete Suchanfragen ist die Volltextsuche daher einem Topic Model in einigen Fällen überlegen[4]. Die gleiche Annahme wird für die Chronologie der Dokumente gestellt. Werden etwa die Kapitel eines Buches als einzelne Texte analysiert, so werden inhaltliche Bezüge nicht beachtet. Die dritte Annahme ist, dass die Anzahl der Themen bekannt ist. Abgesehen von der erheblich längeren Berechnungsdauer, führt eine hohe Zahl von Themen zu der Aufteilung von tatsächlichen Themen und einer folglich schwereren Interpretierbarkeit. Dieses Problem wird als *model checking problem* bezeichnet. Ein zu geringer Detailgrad ist dagegen noch problematischer, da mehrere Themen ineinander übergehen und daher neben der Interpretierbarkeit die Relationsbestimmung leidet. [19, 3].

Blei et al. sehen als Lösung einiger der genannten Probleme die interaktive Analyse von Topic Models. Sie beschreiben die Entwicklung neuer Visualisierungs- und Interaktionsmethoden als vielversprechende Forschungsrichtungen. Demnach erstellen Topic Models eine neue inhaltlicher Struktur, für die eine besondere Form der Exploration gefunden werden muss. Zur weiteren Forschung der interaktiven Analyse mit Topic Models stellen

[4] Testweise lässt sich über `www.google.com` nach zwei Begriffen suchen. Das Vertauschen führt zu anderen Resultaten.

sie die folgenden Fragen:

- Wie lassen sich die Beziehungen zwischen Dokumenten darstellen?
- Was macht ein effizientes Interface aus, mit dem sich der Dokumentkorpus und die Themenstruktur erkunden lassen?
- Themen werden üblicherweise durch die Ausgabe der wichtigsten Begriffe beschrieben. Gibt es effizientere Methoden Themen zu bezeichnen?
- Wie lassen sich nützliche Informationen auf Dokumentebene darstellen?[5]

Chang et al. analysierten die Ausgabe verschiedene Topic Models mit mehreren Themenzahlen auf ihren Detailgrad und der damit verbundenen Interpretierbarkeit. Es wurde eine Massenstudie über den Amazon-Dienst *Mechanical Turk* durchgeführt, bei der 10000 Wikipedia-Einträge und 8557 New York Times Artikel mit jeweils 50, 100 und 150 Themen durch das pLSI, der LDA und dem *Correlated Topic Model* analysiert wurden. Die Interpretierbarkeit wurde anhand von zwei Tests gemessen.

- *Word intrusion*: Die fünf am höchsten gewichteten Begriffe eines beliebigen Themas werden um ein Wort ergänzt. Dieses Wort wird zufällig aus einer Menge von Worten gewählt, die in anderen Themen hoch gewichtet und im gewählten Thema niedrig gewichtet sind. Die sechs Begriffe werden gemischt. Der Proband soll nun das nicht in die Gruppe gehörige Wort, den *intruder* (dt. Eindringling) erkennen.
- *Topic intrusion*: Zu der Überschrift und dem ersten Textabsatz eines zufällig gewählten Artikels werden vier Themen anhand von Schlagwortgruppen angezeigt. Drei der Themen sind die tatsächlich am höchsten gewichteten Themen des Artikels. Das vierte Thema wird zufällig aus einer Menge niedrig gewichteter Themen gewählt. Der Proband soll nun das unpassende Thema finden.

Chang et al. haben dabei festgestellt, dass die LDA am stärksten der menschlichen Interpretation von Themen entspricht. Die Interpretierbarkeit unterscheidet sich dabei stark von mathematischen Maßen, die bislang für die Bestimmung der Effizienz und Exaktheit von Topic Models verwendet wurden. Außerdem haben Chang et al. festgestellt, dass Modelle mit einer hohen Zahl von Themen zu einer Trennung von Inhalten führt, die von Menschen schlecht verstanden werden. [19]

[5] Das Topic Model könnte beispielsweise auf interessante Bereiche des Textes verweisen.

2.2 Wissensaneignung und Problemlösung

Die explorative Suche und Analyse von Daten ist ein Prozess des *Sensemaking* über unbekannte Informationen. Der Suchende hat dabei keine konkrete Vorstellung von den betrachteten Inhalten (der Domäne) und ist nicht in der Lage die Zielfindung genau zu planen. Während der Erkundung werden neue Informationen gesammelt, die wiederum zu der Anpassung der zuvor gestellten Ziele führen. Es handelt sich daher bei der explorativen Suche um einen sehr dynamischen Prozess, dessen unterstützende Automatismen sich dem Zustand des Nutzers anpassen müssen und ihm zugleich möglichst viele Freiheiten lassen. Das Sensemaking in der explorativen Analyse wird nun als Teil der HCI erläutert, die selbst auf viele Erkenntnissen der Psychologie Bezug nimmt. Die Kognition (lat. *erkennen, erfahren, kennenlernen*) beschreibt die Informationsumgestaltung und -aneignung. Modelle der Kognitionspsychologie, also der Funktionsweisen des menschlichen Gedächtnisses, geben einen tieferen Einblick in die Informationsakkumulation, -verarbeitung und -speicherung. Die Lernpsychologie erklärt zudem das Erlernen neuen Wissens in der Exploration, auf die die Modelle von explorativen Suchprozessen aufbauen. Auch für die Informationsvisualisierung und das IIR sind Aspekte der kurzzeitigen Informationsaufnahme und -verarbeitung relevant, wie sie auch aus der Gestaltpsychologie bekannt sind.

2.2.1 Das Gedächtnis als 3-Speicher-Modell

Das 3-Speicher-Modell wurde 1968 von Atkinson und Shiffrin vorgestellt und unterteilt das Gedächtnis in drei unabhängige Teile unterschiedlicher Speicherdauer und Funktion. Die einzelnen Teile sind das sensorische Register, das Arbeitsgedächtnis (ursprünglich Kurzzeitgedächtnis) und das Langzeitgedächtnis. Das sensorische Register hat die kürzeste Speicherdauer und ist die erste Stufe der Filterung von sensorischen Informationen (Sinneseindrücken). Das Arbeitsgedächtnis verbindet neue Erlebnisse mit bestehendem Wissen, wobei die neuen Erlebnisse aus dem sensorischen Register und das bestehende Wissen aus dem Langzeitgedächtnis stammen. Das Langzeitgedächtnis speichert Informationen als Wissensstrukturen. [20]

Das 3-Speicher-Modell wurde mittlerweile weiterentwickelt, sodass nicht mehr von drei unabhängigen Gedächtnisteilen, sondern von differenzierten und ineinandergreifenden Strukturen ausgegangen wird. Etwa haben Baddley et al. die Unterscheidung zwischen Kurzzeit- und Arbeitsgedächtnis eingeführt, demnach das Kurzzeitgedächtnis für die Informationsspeicherung und das Arbeitsgedächtnis für die Verarbeitung und die *Organisation* des Gedächtnisses zuständig ist [21]. Abbildung 2.4 zeigt die Aktionen des Arbeitsgedächtnisses. Das sensorische Register ist in der Abbildung nicht eingezeichnet und sollte dort noch vor der *Außenwelt* stehen.

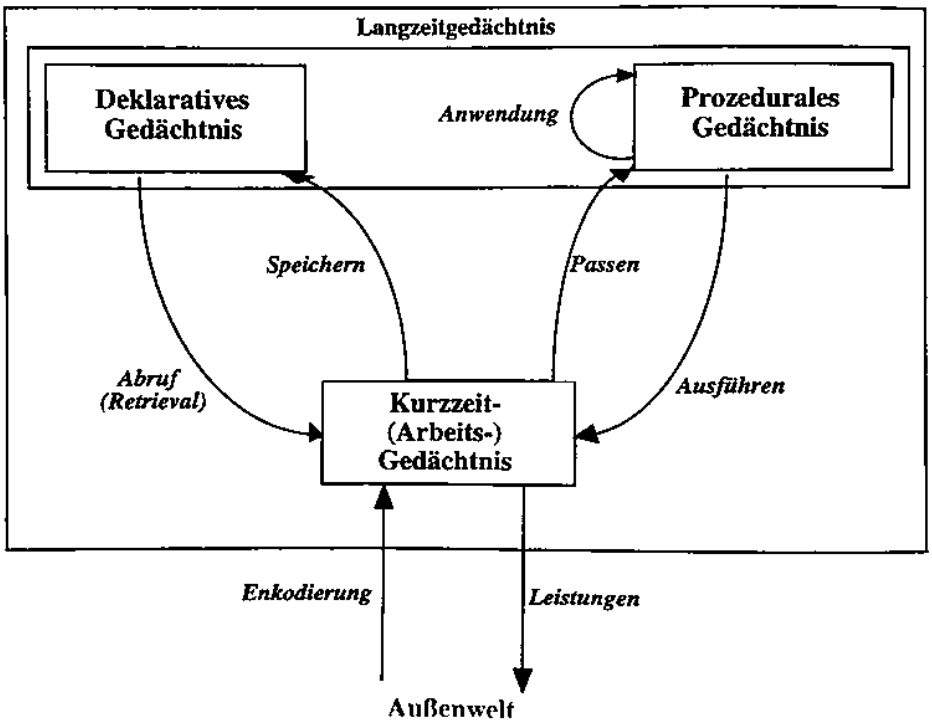


Abbildung 2.4: Zusammenspiel der Gedächtniseinheiten nach Anderson (1983). *Quelle: Seel (2003), S.42*

Das sensorische Register, auch ikonisches Gedächtnis oder Ultrakurzzeitgedächtnis genannt, speichert visuelle Informationen für weniger als eine Sekunde. Für diese kurze Zeitspanne werden visuelle Reize in großem Umfang behalten und gehen nach etwa $400ms$ rapide verloren. Der evolutionäre Zweck wird mit der Überbrückung des Lidschlusses argumentiert [22]. Die Effizienz des sensorischen Registers wird an dem Phänomen der Fehler-Suchbilder deutlich, deren Unterschiede sich viel einfacher finden lassen, wenn beide Bilder übereinandergelegt werden. Ähnliche Methoden können für den Vergleich von visualisierten Informationen verwendet werden.

Besonders prägnante Informationen des sensorischen Registers gelangen in das Arbeitsgedächtnis. Dadurch lässt sich die Aufmerksamkeit eines Rezipienten lenken, indem etwa in einem Text besonders wichtige Wörter markiert oder in größerer Schrift gesetzt werden. Andere Elemente werden erst durch eine kontextuelle Analyse im Arbeitsgedächtnis aktiv wahrgenommen. Die Aufgabe des Arbeitsgedächtnisses ist das Behalten von neuen Informationen, zu denen passende Wissensstrukturen des Langzeitgedächtnisses gesucht werden. Neue Informationen werden demnach anhand bekanntem Wissen organisiert, gefiltert und auf eine langfristige Speicherung vorbereitet. Dieser Effekt wird ebenfalls in Abschnitt 2.2.4 im Kontext des Sensemaking beschrieben. Neben Informationsverarbeitungsprozessen ist das Arbeitsgedächtnis auch für Problemlösungsprozesse zuständig. Bei der Lösung von Problemen müssen die Aufgabenstellung, Ziele und Inhalte erinnert wer-

den, während Aktionen geplant und durchgeführt werden. Etwa für einen Telefonanruf darf eine gemerkte Telefonnummer beim Wählen nicht vergessen werden. Externe Hilfsmittel, wie ein Stift und ein Notizblock, können das Arbeitsgedächtnis entlasten und den Problemlösungsprozess beschleunigen.[6] Während Atkinson und Shiffrin von einer Merkleistung des Kurzzeitgedächtnisses von 20 bis 30 Sekunden ausgingen, hat Georg Miller diese auf 7 ± 2 Einzelinformationen über eine Zeitspanne von bis zu 15 Sekunden bestimmt. Der Umfang der Einzelinformationen hat dabei weitaus weniger Einfluss auf die Merkbarkeit als die Anzahl unabhängiger Informationen. Demnach werden zusammengehörige Informationen effektiv Gruppiert, was Miller als *chunking* (dt. Bündelung) bezeichnet [23]. Die Merkdauer hängt von der zwischen Aufnahme und Wiedergabe liegenden Interferenz ab, wobei Passivität und Entspannung die Ablenkung lindern. In alltäglich Situationen, die üblicherweise kein völlig entspannendes Umfeld bieten, wird mittlerweile von einer Merkleistung von nur 3 ± 1 Einzelinformationen ausgegangen. [22]

Zur Lösung von Problemen und zur Erlernung neuen Wissens werden Informationen des Kurzzeitgedächtnisses mit Strukturen des Langzeitgedächtnisses verglichen. Abbildung 2.4 zeigt wie sich dessen Aufgabe in die Speicherung von prozeduralem und deklarativem Wissen aufteilt. Der prozedurale Teil speichert konkrete Erfahrungen und Fähigkeiten die nicht bewusst ausgeführt werden und wird deshalb auch *implizites Gedächtnis* genannt. Die Abstraktion von implizitem Wissen generiert explizites Wissen. Dieses wird in umfassendere Bezüge bereits bestehenden expliziten Wissens gesetzt und im deklarativen Gedächtnis gespeichert. Durch das entstehende semantische Netz ergeben sich die Bezeichnungen des *expliziten, semantischen* oder deklarativen Gedächtnis. Es enthält das Wissen über die Welt in Form von Fakten, Ideen, Meinungen und Konzepten und lässt sich als gewichtetes Netz atomarer Einzelinformationen verstehen. Die Zahl der Verbindungen und deren Stärke definieren die Informationsdichte und die Fähigkeit diese abzurufen. Man könnte meinen, dass für die explorative Datenanalyse das explizite Gedächtnis am stärksten beansprucht wird, doch dem gegenüber steht das Modell der *Schemata*. Damit sind abstrakte *Vorstellungen* des impliziten Gedächtnisses bezeichnet, die auf Eindrücke des Kurzzeitgedächtnisses gelegt werden um aus ihnen eine Sinneinheit zu bilden. Demnach werden neue Informationen am effizientesten Verarbeitet, wenn sie sich in bestehende Vorstellungen fügen. Die Wahl der Schemata erfolgt in den meisten Problemlösungsprozessen unbewusst, doch je komplexer ein Problem wird, umso expliziter muss über den Lösungsweg nachgedacht werden.

Es lässt sich zusammenfassen, dass neues Wissen nach bestehendem Wissen geordnet und gefiltert wird. Die Beurteilung ob etwas sinnvoll ist oder nicht und für eine weitere

[6] Diese selbstverständlich wirkenden Aussagen werden erst in der späteren Übertragung auf komplexe Analyseprozesse ihren Sinn entfalten.

Verarbeitung relevant ist, erfolgt vorwiegend implizit. Dieser Vorgang erfolgt sehr schnell, wodurch ständig neue Sinneseindrücke in das Arbeitsgedächtnis gelangen. Dort können nur wenige Einzelinformationen parallel gespeichert werden. Für kurze als auch für lange Zeitabstände funktioniert das Gedächtnis effizienter mit in sich schlüssigen Sinneinheiten. Die Merkmale der einzelnen Elemente des das 3-Speicher-Modells sind in Tabelle 2.1 als Übersicht aufgelistet.

	Sensorisches Register	**Kurzzeitgedächtnis**	**Langzeitgedächtnis**	
			Deklarativ	**Prozedual**
Kapazität	Sehr groß (bis 16.000 Bit)	7 ± 2 Elemente	Sehr groß	
Dauer	250 Millisekunden	$1 - 4$ Minuten	Lang	
Format	Visuelle Merkmale	Phonemisch	Semantisch	Assoziativ

Tabelle 2.1: Vergleich der Merkmale des Mehrspeichermodells *Quelle: Seel (2003), S. 43; Sokolowski (2011), S. 173*

2.2.2 Lernprozesse

Das Erwerben neuen Wissens wird in der Lernpsychologie behandelt. Es werden drei Formen kognitiver Lernprozesse unterschieden:

1. *Bedeutungserzeugendes Lernen*: Das Enkodieren und Speichern von Informationen und die semantische Einordnung in Vorwissen.

2. *Inferenzielles Lernen*: Das Schlussfolgern über bestehendes Wissen, mit dem Ziel logische Regel und Prinzipien zu identifizieren.

3. *Metakognitives Lernen*: Die Selbstreflexion und -regulierung im Bezug zu Lernstrategien und Wissensmanagement.

Wie bereits in Abschnitt 2.2.1 erläutert Teilt sich das menschliche Langzeitgedächtnis in explizites und implizites Wissen. Prinz beschreibt die darin enthaltenen Wissensstrukturen als Netze einzelner Eindrücke, die eine Weltsicht ergeben, aus der sich die individuelle Interpretationsfähigkeit ableitet. Besonders starke Vernetzungen bilden Bedeutungseinheiten, die als Schemata bezeichnet werden [24]. Schemata beeinflussen das handeln durch die *aufmerksamkeitssteuernde Funktion*, die *Integrationsfunktion* und die *Inferenzfunktion*. Diese psychologische Ansicht findet sich auch im Machine Learning und dem IIR wieder, etwa in der in Abschnitt 4 verwendeten Inferenz eines trainierten Topic Models. Das Langzeitgedächtnis bildet sich als stetiger Prozess neu, indem bestehende Netze um neue Elemente und neue Verbindungen ergänzt werden. Je stärker ein Netz ist, umso schwerer lässt es sich umstrukturieren. Ebenso schwer fällt das erlernen völlig neuen

Wissens. Anderson beschreibt dies folgendermaßen: „Without a schema to which an event can be assimilated, learning is slow and uncertain“. [25, 26, 24]

Beim *inferenziellen Lernen* werden bestehende Strukturen bewusst analysiert und reorganisiert. Dadurch kommt es zur *auslöschenden Assimilation*, dem entfernen bestehender Wissenseinheiten. Es wird allerdings kein ungenutztes Wissen verdrängt, sondern bestehende Wissensbereiche zusammengefügt oder für eine höhere Detaillierung aufgespalten.[7] Im ineretiellen Lernprozess wird bestehendes Wissens gesammelt, progressiv Differenzieren, sequentiell Organisieren und integrierend Verbunden. Vereinfachend sind damit das Strukturieren vom Allgemeinen zum Detail, das Verbinden von Kernbegriffen verschiedener Bereiche und das Verbinden von Begriffen innerhalb eines Bereiches zum Auffinden von Ähnlichkeiten und Unterschieden gemeint.

Das *metakognitive Lernen* bezeichnet das Beobachten der eigenen Denkweise und das Organisieren eigener Lern- und Denkprozesse zum Zweck der Effektivitätssteigerung, die sich aus folgenden Elementen zusammensetzt [25]:

1. *Metakognitives Wissen*: Das Wissen über den eigenen Wissensbestand.

2. *Metakognitive Fertigkeiten*: Das Wissen über das eigene Verhalten in dem Kontext ähnlichen Problemlösungsszenarien.

3. *Metakognitive Erfahrungen*: Die Fähigkeit eine Situation bezüglich emotionaler Reaktionen und der eigenen Motivation einschätzen zu können.

2.2.3 Explorative Suche als Problemlösungsprozess

Eine explorative Analyse hat das Ziel einen unbekannten Datenbestand inhaltlich erfassen zu können. Die Charakteristik einer explorativen Suche ist, gegenüber einer *normalen* Suche, dass der Nutzer kein genaues Wissen über die vorliegende Domäne hat. In diesem Fall ist das Problem nicht, die richtigen Wörter zu finden, sondern nicht die richtigen Konzepte für eine Suchanfrage zu kennen. Qu et al. führen ein Beispiel auf:

> Ying ist eine Studentin aus China, die einer Freundin zum Geburtstag eine Flasche Wein schenken möchte. Wein ist in der chinesischen Kultur nicht weit verbreitet, daher möchte Ying mehr darüber erfahren. In einer Suchmaschine sucht sie nach "Weinünd findet einige Webseiten, die in das Thema einführen. Darin entdeckt sie weitere interessante Themen, wie die Herstellung und den Anbau in verschiedenen Regionen. Ying macht eine Liste von interessanten Subthemen und sucht unabhängig nach jedem einzelnen Thema. Nun hat sie

[7] In der Kognitionspsychologie werden diese Vorgänge als *korrelative* und *deriative Subsumption* bezeichnet.

> etwas Wissen über das Thema "Wein"gesammelt und beginnt nach einem Wein für ihre Freundin zu suchen. Einige kommen in Frage, sodass Ying nach jedem einzeln sucht und schließlich einen passenden findet.

Qu et al. wollen an diesem Beispiel zeigen, dass es in der explorativen Suche nicht ein einzelnes *nugget* an Informationen gibt, dass den Nutzer befriedigt. Vielmehr werden *nuggets* verschiedener Themen zusammengefügt. Dafür ist das Wissen über die Struktur des Themas wichtig, also keine Einzelinformation sondern ein Verständnis über das generelle Thema [27]. In diesem Punkt geht die explorative Suche in die explorative Analyse über. Der Unterschied von explorativer Suche und Analyse liegt in der Zielsetzung. In der Suche soll etwas gefunden werden, von dem eine mehr oder weniger konkrete Vorstellung besteht. In der Exploration soll das Verständnis von einem grob umrissenen Thema, oder eine vorliegende Datensammlung erweitert werden.

In der Exploration werden mehrere Informations- und Themenstränge voneinander unabhängig verfolgt, die im Verlaufe der Analyse zusammenfügt werden. Durch das stets neu angeeignete Wissen ändert sich das Vorgehen häufig und spontan. Dabei kann sich ein Nutzer schnell in den Informationen verirren, wodurch er verunsichert und demotiviert wird. Um dem entgegenzuwirken zergliedert Ingwersen den Suchprozess in einzelne *Microsystems*, in denen jeweils unterschiedliche Nutzerziele verfolgt werden. Tabelle 2.2 ist eine Zusammenfassung dieser Microsystems und zeigt das jeweilige Nutzerverhalten und die vom unterstützenden System verwendbaren Informationen. Nach Ingwersen et al. gehen Prozessschritte fließend ineinander über, da beim Erreichen eines Teilziels direkt eine neue Suche angestoßen wird. [10][8]

Die Bearbeitung eines jeden Teilziels kann als eigener Problemlösungsprozess verstanden werden. Sokolowski beschreibt ein Experiment über die Dokumentation von Problemlösungsprozessen, in denen Probanden eine Problemstellung bekamen, während deren Lösung sie ihre Gedanken laut aussprechen sollten. Der Prozess fiel individuell sehr verschieden aus, sodass keine wiederkehrenden Lösungsmuster identifiziert werden konnten. Die Probanden haben das Lösen einer Aufgabe selbst in keinem Fall aktiv ausgesprochen, sondern nur Teillösungen und Vorhaben geäußert. Daraus wurde gefolgert, dass das tatsächliche Lösen von Problemen nicht bewusst erfolgt. Die von den Probanden verbalisierten Teillösungen lassen sich als Ergebnisse von Iterationen der folgenden Schritte auffassen [22]:

1. In der Vorbereitung wird das Problem formuliert und in einen persönlichen Bezug gesetzt.

2. Die Lösung erfolgt unbewusst.

[8] Kapitel 5 und 6.

	Zustand und Verhalten des Nutzers	**Vom System verwendbare Informationen**
Vor der Suche	Fehlendes Wissen verursacht ein Problem, dass der Nutzer durch Anfragen des Systems beheben möchte	Situation, Ziele, Vorwissen, Erwartungen
Während der Suche	Formulierung der Suchstrategie, Selektion der Quellen, aktive Suche, Evaluation der Resultate, neu Formulierung der Suche	Frage des Suchenden, Suchverhalten, Systeminteraktion, Bewertung der Resultate
Nach Erhalt der Ergebnisse	Evaluation der endgültigen Resultate, Verwendung der Informationen	Zufriedenheit des Nutzers, Erfüllung der Nutzerziele

Tabelle 2.2: Zustand und Ziele mit dazugehörigen Informationen, die zur Optimierung des Suchprozesses verwendet werden können. *Quelle: Ingwersen (1992), S. 86*

3. Wurde eine Lösung gefunden, tritt ein belohnendes Gefühl des Verstehens und der Erleichterung ein; die gegebenen Informationen werden nun als zu Eigen gemacht empfunden.

4. Die Einordnung wird evaluiert.

Während der Ausführung des Experiments fiel auf, dass das bewusste Reflektieren von Gedankengängen den Lösungsprozess erheblich verlangsamte. Ein *Monitoring* der Introspektive sollte demnach nicht mehr Kognitionsleistung benötigen, als dadurch Vorteile gewonnen werden. Der Nutzen einer Externalisierung hängt stark von der Komplexität der Problemstellung und der kognitiven Leistung des Nutzers ab. Eine technische Unterstützung, ähnlich eines Notizblockes, müsste sehr Frei gestaltet sein, sodass eine individuelle Balance zwischen Auslagerung und innerer Verarbeitung gefunden wird. [22]

2.2.4 Sensemaking

Sensemaking bezeichnet das Ableiten eines Sinnes aus Erlebnissen. Diese Erlebnisse können entweder direkt und körperlich erfahren werden (beispielsweise in einer Unfallsituation), oder indirekt in Form von Informationen aufgenommen werden. Der Begriff wurde 1976 von Brenda Dervin eingeführt, der auch im Deutschen verwendet wird. Nach ihrer Beschreibung entsteht Sensemaking in einer Situation, in der eine Wissenslücke überbrückt werden muss, sodass ein erahntes Ziel erreicht werden kann. Um die Lücke zu überbrücken muss aus neuem Wissen ein Sinn abgeleitet werden [28]. Russell et al. beschreiben das Sensemakings als *„a process of searching for a representation and encoding*

data in that representation to answer task-specific questions“ [7][9] Aus der Lernpsychologie stammt das *kognitiv-konstruktivistische Erklärungsmodell*, demnach jeder Lernende entsprechend seinem inneren Modell der Welt Informationen aufnimmt und verarbeitet [25]. Nach Klein et al. wird das Sensemaking durch Eindrücke ausgelöst, die nicht zum inneren Welten-Modell passen. Auf der Grundlage des bestehenden Welten-Modells werden neue Informationen geordnet und in Sinneinheiten aufgeteilt. Die Art der Verarbeitung wird somit von jedem Menschen unterschiedlich ausgeführt. Den gleichen Informationen wird daher von verschieden geprägten Menschen, unter Umständen unterschiedliche *Sinne* zugewiesen.

Klein et al. beschreiben das Sensemaking als Wechselspiel zwischen Eindrücken und Vorstellungen. Den Eindrücken wird versucht durch bestehendes Wissen einen Sinn zu geben und wenn dies nicht klappt, wird versucht das bestehende Wissen an die neuen Eindrücke anzupassen, um eine neue Erklärung zu finden. Dabei steht die Plausibilität und der Kontext im Fokus und nicht die Genauigkeit einzelner Informationen. Da Eindrücke erst gesammelt und in ihrer Menge betrachtet und geordnet werden müssen, wird das Sensemaking als retrospektiver Prozess bezeichnet. Das bedeutet nicht, dass es ein langwieriger Vorgang ist. Vielmehr erfolgt es so schnell, dass äußerst komplexe und gut dokumentierte Fälle großer Ereignisse verwendet werden müssen, um das Sensemaking zu beobachten.[10] Durch die retrospektive Verarbeitung müssen für komplexe Situationen sehr viele Informationen behalten werden. An dieser Stelle bietet sich die Möglichkeit der maschinellen Unterstützung. Ein weiterer Punkt ist die Orientierung im Prozess, der durch eine fehlende Linearität erschwert wird. Auch dort kann ein Computer die kognitive Last reduzieren. Im Sensemaking strömen üblicherweise sehr viele Informationen auf eine Person ein, von denen nur solche wahrgenommen werden die besonders hervorgehoben sind. Klein et al. bemerken, dass die Unterstützung der Hypothesenbildung vermieden werden sollte, da somit der intellektuelle Anspruch gemindert wird, was den Nutzer demotivieren kann und bei der Lösung komplexer Probleme zu schlechten Ergebnissen führt. Daher muss ein Kompromiss zwischen Entlastung und Forderung des Nutzers gefunden werden. [29, 30]

Pirolli et al. zergliedern den Sensemaking-Prozess in Wissensstrukturen steigender Komplexität, deren Bildung einen äquivalent hohen Aufwand benötigt. Die einzelnen Stufen werden durch loops verschiedener Aufgaben und unterschiedlichen Umfängen verknüpft. Das Modell ist in Abbildung 2.5 dargestellt[11]. Von links nach rechts steigt die Komplexität

[9] Zitiert nach Qu et al. [27].

[10] Da die Theorie des Sensemaking nicht zwischen direkt erlebten Eindrücken und der bewussten Informationsaufnahme unterscheidet, werden zur Analyse des Sensemakingprozesses, Vorfälle wie der Absturz eines Flugzeuges verwendet.

[11] Die Autoren von [1] sind der Kognitionspsychologe Peter Pirolli und der Research Scientist Daniel M. Russell. Das Modell von Pirolli et al. ist daher eine Weiterentwicklung der Arbeit [7] von Russell, das

und mit jedem Schritt findet eine erneute Filterung und Anordnung der Informationen statt. Die jeweils neu gebildeten Strukturen können weiter ausgeführt werden, oder dazu dienen die darunterliegenden Ansammlungen zu evaluieren. Ebenso wie Klein et al. sehen Pirolli et al. den Prozess als nicht-linear. Eine in Schritt 13 (auf Abbildung 2.5) gefundene Hypothese kann demnach zur Verwerfung darunterliegenden Strukturen führen, sodass in den *1. External Data Sources* nach neuen Informationen gesucht werden muss. Es werden zwei wesentliche Schleifen erfasst. Dazu gehört der *foraging loop* (foreaging auf dt. hamstern) bei dem neue Informationen gefunden, gefiltert und in ein bestehendes Framework eingeordnet werden. Der *Ort* der Datensammlung wird in dem Modell als *Shoebox* bezeichnet. Die zweite Schleife ist der *sensemaking loop*, der die iterative Entwicklung von Schemata und der Ableitung von Hypothesen umfasst. [1]

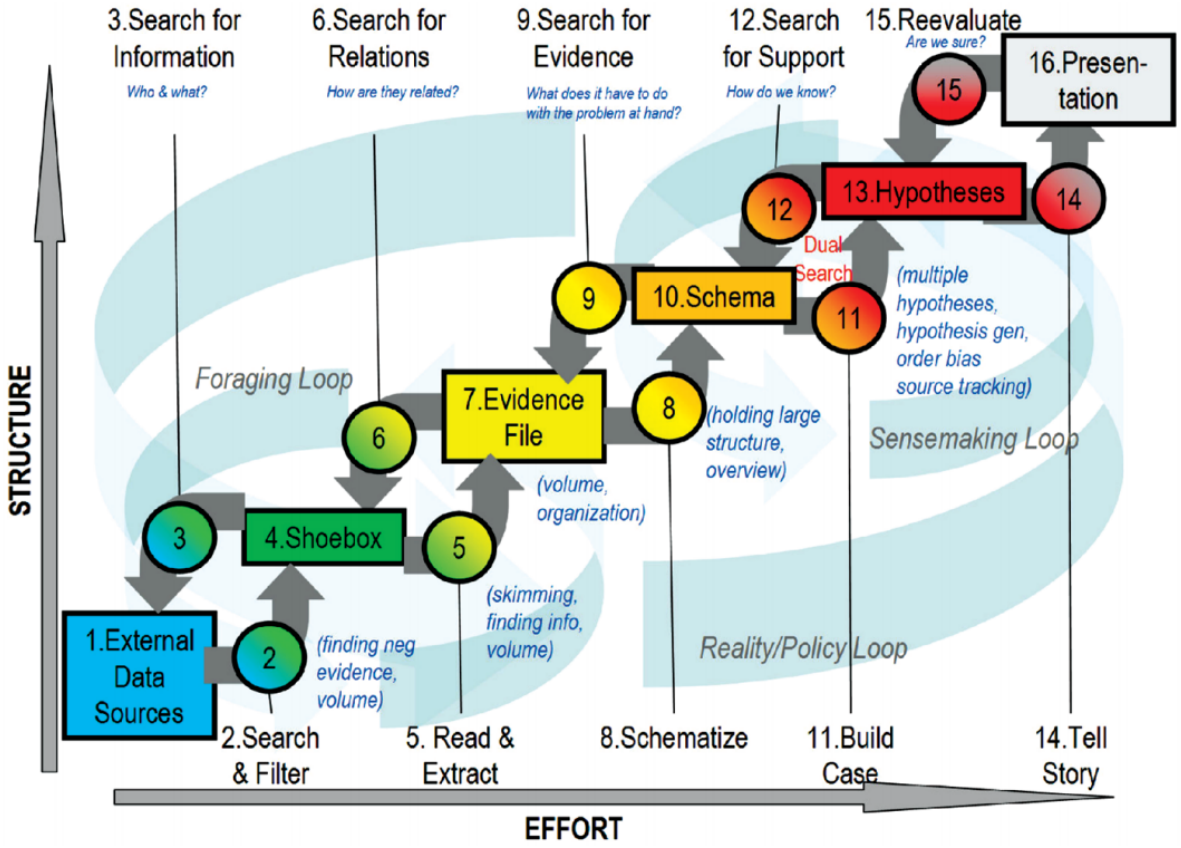


Abbildung 2.5: Modell des Sensemaking-Prozesses in der explorativen Analyse. *Quelle: Pirolli et al (2011)*

Klein et al. haben darauf hingewiesen, dass die Hypothesenbildung dem Nutzer nicht abgenommen werden soll, da er sonst entmutigt wird komplexe Probleme zu lösen [29]. Nach Pirolli et al. sind die *Shoebox*, die *Evidence File* und das *Schema* die Einheiten, deren Bildung in einer explorativen Suche aktiv maschinell unterstützt werden können [1].

um kognitionspsychologische Aspekte ergänzt wurden.

Auch wenn der Aufwand mit der Komplexität der Strukturen wächst, ist nach der Aussage von Klein et al. davon auszugehen, dass der Grad der Automatismen relativ zur Komplexität abnehmen sollte. Somit könnten etwa Informationen automatisch in der Shoebox akkumuliert werden, ohne dem Nutzer die Aufgabe abzunehmen, daraus eigene Schemata abzuleiten. Diese Erkenntnisse und Fragen werden in Abschnitt 4 wieder aufgegriffen.

2.3 Interaktives Information Retrieval

Es werden nun praktische Übertragungen der vorgestellten Modelle der Kognitionspsychologie und der Sinnstiftung vorgestellt. Dazu werden die einzelnen Aufgaben des Sinnstiftungsprozesses auf verschiedene Ansichten unterteilt, sodass sie optimal unterstützt werden können. Diese Aufteilung wird um generelle Richtlinien der Konzeption von Interfaces zur visuellen Analyse ergänzt. Anschließend werden Methoden und Designrichtlinien der Visualisierung und Interaktion mit relationalen Daten erläutert.

2.3.1 Die Verwendung mehrerer Datenansichten

Die Merkleistung des Arbeitsgedächtnis ist, wie in Abschnitt 2.2.1 beschrieben, zeitlich und inhaltlich sehr beschränkt. Besonders gut arbeitet das menschliche Gedächtnis mit abgeschlossenen Sinneinheiten. Die explorative Suche verläuft jedoch nicht linear sondern in parallelen Themensträngen, zwischen denen gesprungen wird, um sie am Ende zu einem Konzept zusammenzufügen. Sich mehrere Themen parallel zu merken fällt dem Gedächtnis sehr schwer, wodurch die Orientierung in der Exploration schnell verloren geht und ein Nutzer schnell vergisst, was er bereits behandelt hat. Shrinivasan et al. entwickelten mit Bezug auf den information foreaging und sensemaking loop ein Framework, das den Prozess der analytischen Schlussfolgerung über interaktive Visualisierungen unterstützen soll. Sie beobachteten, dass in der Praxis nur das Sammeln und Organisieren von Informationen unterstützt wird und nicht die Konstruktion eines mentalen Modelles. Demnach wird der sensemaking loop noch nicht ausreichend behandelt. In Abbildung 2.5 ist der sensemaking loop auf der rechten Seite zu sehen, wo er das Bilden von Schemata und das Ableiten von Hypothesen umfasst. Klein et al. sind der Meinung, dass die Unterstützung der Hypothesenbildung einen Nutzer bei der Lösung komplexer Probleme demotiviert [29]. Den Schritt der Hypothesenbildung gehen Shrinivasan ebenfalls nicht und begrenzen ihr Framework auf die Entwicklung von Schemata. Im Ganzen besteht dieses aus einer *Data View* in der mit visualisierten Darstellungen interagiert wird, einer *Navigation View* die aus einem automatischen Aktivitätsprotokoll besteht und einer *Knowledge View* in der gesammelte Informationen festgehalten werden. Laut Shrinivasan et al. kann mit Visualisierungen direkt agiert werden, etwa durch Selektion, oder durch die Veränderung

des *interaction Interfaces*, das aus Spezifikationen besteht die einzelnen Aktionen visuelle Attribute zuweist. [31]

Die Navigation View bietet eine interaktive Übersicht verschiedener handlungsstränge, bei der das einfache Wechseln zwischen Zuständen im Vordergrund steht. Außerdem sollte es nach Shrinivasan et al. möglich sein, von jedem beliebigen vergangenen Zustand eine neue Suchrichtung einzuschlagen, von dem ein Verlaufsprotokoll parallel zum ursprünglichen Protokoll geführt wird. Shrinivasan verwenden dazu eine Baumstruktur, in der beliebige Zustände hervorgehoben werden können und somit eine Organisation des Suchprozesses ermöglicht wird. Bei einer automatischen Protokollierung sollte der Übersicht halber darauf geachtet werden, dass nicht zu viele Informationen aufgezeichnet werden. Neben direkten Aktionen, wie etwa einer Selektion, sollen auch Manipulationen des interaction Interfaces aufgezeichnet werden.

Die Knowledge View ist ein dynamischer Arbeitsraum in dem Attribute und Elemente verschiedener Ebene und Ansichten in Verbindung gestellt werden. Einzelne Knoten lassen sich miteinander verbinden oder in Gruppen zusammenfügen. Die Tiefe der Verschachtelung ist dabei dem Nutzer überlassen, sodass sich auch Gruppen von Gruppen bilden lassen und eine möglichst hohe Flexibilität beibehalten wird. So wie in der Navigation View können einzelne Elemente hervorgehoben und mit einem eigenen *Degree Of Interest* (Relevanz-Wert) gewichtet werden. Die Schemen der Knowledge View dienen der Entlastung des Arbeitsgedächtnisses, der Bildung von komplexen mentalen Modellen und der Präsentation gesammelter Informationen. Shrinivasan et al. sehen jedoch nicht vor, die Schemen wiederum auf die Daten anzuwenden. Filter müssten somit manuell entworfen werden. Abschließend erfassen Shrinivasan et al. fünf Anforderungen an interaktive Visualisierungsumgebungen [31]:

1. Analyseartefakte wie Beweise, Hypothesen und Zuweisungen können ausgedrückt und festgehalten werden
2. Analyseartefakte können organisiert werden
3. Der Explorationsprozess ist Überprüfbar, mehrdimensional und organisierbar
4. Analyseartefakte und Visualisierungen lassen sich in Verbindung stellen
5. Ergebnisse lassen sich aufbereiten und können mit anderen Geteilt werden

Generellere Regeln zur Erstellung von *multi-view interfaces* der visuellen Datenanalyse haben Wang et al. aus der Analyse mehrerer Tools abgeleitet [32]. Tabelle 2.3 listet einen Ausschnitt der Regeln und deren positiven als auch negativen Einflüsse auf. Der *Lern*aufwand steigt mit der Komplexität eines Interfaces, sodass die *Vielfalt* zu einer

Regel	Zusammenfassung	Positive Einflüsse	Negative Einflüsse
Vielfalt	Verwende mehrere Ansichten, wenn es eine hohe Vielfalt an Attributen, Modellen, Benutzerprofilen oder Abstraktionslevel gibt	Erinnerung	Lernen, Berechnung, Platzbedarf
Komplementarität	Verwende mehrerer Ansichten, wenn diese Zusammenhänge oder Unterschiede hervorstellen	Erinnerung, Vergleichbarkeit, Kontextwechsel	Lernen, Berechnung, Platzbedarf
Zerlegung	Trenne komplexe Daten in verständliche Teile und unterstützte den Verstehensprozess durch dimensionsübergreifende Interaktionsmöglichkeiten	Erinnerung, Vergleichbarkeit	Lernen, Berechnung, Platzbedarf
Sparsamkeit	Verwende möglichst wenige Ansichten	Lernen, Berechnung, Platzbedarf	Erinnerung, Vergleichbarkeit, Kontextwechsel
Ressourcenoptimierung	Vergleiche den Zeit- und Platzbedarf einer Ansicht mit dessen Nutzen		
Selbstverständlichkeit	Verwende hinweise der Reize um auf Verbindungen von Ansichten hinzuweisen	Lernen, Vergleichbarkeit	Berechnung
Konsistenz	Verwende einheitliche Layouts und Zustände in den verschiedenen Ansichten	Lernen, Vergleichbarkeit	Berechnung
Aufmerksamkeitslenkung	Lenke die Aufmerksamkeit des Betrachters auf die wesentlichen Ansichten	Erinnerung, Kontextwechsel	Berechnung

Tabelle 2.3: Regeln zum Erstellen von Interfaces der visuellen Analyse mit mehreren Ansichten. *Quelle: Wang et al. 2000*

schwerer verständlichen Ansicht führt und die Regel der *Sparsamkeit* entsprechend zu einfach erlernbaren Arbeitsumgebungen führt. Andersherum verhält es sich mit der *Erinnerungsleistung*, die durch eine höhere Anzahl von Ansichten entlastet wird, da weniger Variationen und Charakteristiken der Daten gemerkt werden müssen. Die Möglichkeit, Zustände in einer speziellen Ansicht wie der Knowledge View temporär zu speichern werden von Wang et al. nicht bedacht. Die *Vergleichbarkeit* von Daten bedingt Möglichkeit über sie schlussfolgern und urteilen zu können. So lassen sich Hypothesen nur effektiv evaluieren, wenn alternative Lösungen miteinander verglichen werden können. Der *Kontextwechsel* bezeichnet die zu erbringende Gedächtnisleistung, die benötigt wird damit ein Nutzer sich in einer neuen Ansicht orientieren kann. Klare Unterscheidungen und offensichtliche Verbindungen von Ansichten erleichtern demnach den Kontextwechsel.

2.3.2 Relationen im Text Mining

Visualisierungen können das Auffinden von wichtigen Informationen und Muster in großen Textkollektionen vereinfachen, die sich nur schwer durch die direkte Betrachtung einzelner Texte finden lassen. Die Abstraktion der Daten wird durch den Retrieval-Algorithmus bestimmt, in diesem Fall der LDA. Die LDA gibt gewichtete Relationen zwischen allen Dokumenten, Themen und Worten aus. Dadurch wird ein hochdimensionaler Raum gebildet, der auf eine zweidimensionale Abbildungsebene projiziert werden muss. Durch die Dimensionsreduktion gehen zwangsläufig Informationen verloren, sodass es schwer ist *eine* optimale Darstellungsart zu finden.

Es werden explizite und implizite Relationen unterschieden. Die expliziten Relationen

sind konkret definiert, wie zum Beispiel die Nutzerverbindung in sozialen Netzen, während implizite sich durch Ähnlichkeiten zwischen Attributen ergeben, etwa den gemeinsamen Interessen von Nutzern. Nach Dörk et al. werden für explizite Relationen meistens *node-link Diagramme* verwendet und für implizite Relationen das *multidimensional scaling* [33]. Node-link Diagramme sind durch Kanten verbundene Knoten, die üblicherweise für die Darstellung von Graphen verwendet werden und daher auch häufig selbst als Graphen bezeichnet werden. Im multidimensional scaling werden Elemente entsprechend ihrer Ähnlichkeit in einem zwei- oder dreidimensionalen Raum positioniert. Dadurch entstehen Cluster von sich ähnelnden Elementen, wobei auch nebeneinanderliegende Cluster eine hohe Gemeinsamkeit aufweisen. Die entstehende *Karte* kann vom Nutzer räumlich erkundet werden und lässt sich mit einer Volltextsuche verbinden, indem die Resultate auf der Karte hervorgehoben werden. Eine weit verbreitete Umsetzung von node-link Diagrammen ist der *force directed graph*, der im weiteren Verlauf dieser Arbeit wiederholt erwähnt werden wird. [34, 33]

Richtlinie	**Beschreibung**
Detail auf Anfrage	Das schnelle erfassen von Details in einer globalen Ansicht
Datenübersicht	Globale Ansicht der Daten, um die Orientierung zu unterstützen
Verlinkungen	Typ und Gewicht von Verbindungen anzeigen. Das Sichtbarmachen von Verbindungen in verschiedenen Datensets unterstützt das Erkennen von Relationen
Einstiegspunkt	Einen einfachen, möglichst visuellen Einstiegspunkt bieten, um neuen Nutzern einen schnellen Zugang zu ermöglichen
Verschiedene Suchformen	Mehrere Komplexitätsstufen der Suchanfragen anbieten, um verschiedene Anwendergruppen und Arbeitsweisen zu unterstützen
Filtern	Unwichtige Daten vor der Visualisierung entfernen. Außerdem können Filter von Benutzern als einfache Suchanfragen verwendet werden
Facettensuche	Das Suchen und Filtern über weitere Eigenschaften als Entitätsrelationen ermöglichen, um unbekannte Relationen aufzudecken
Verlauf	Aufzeichnen der Aktionen zur späteren Ansicht und Wiederherstellung vorangegangener Stadien.
Zusammenfügen von Daten	Daten verschiedener Quellen zusammenfügen können
Visuelle Präsentation	Verwenden bekannter Darstellungsformen in Form von Piktogrammen
Skalierung der Anwendung	Verarbeitbarkeit unterschiedlich großer Datenmengen

Tabelle 2.4: Designrichtlinien für interaktive Visualisierungen in der explorativen Suche. *Quelle: Dadzei et al. 2011*

Landesberger et al. haben 2011 die momentan aktuellste Übersicht der Analysemethoden über Graphenvisualisierungen erstellt und erfassen eine Vielzahl von Visualisierungs- und Interaktionstechniken. Die einfachsten Interaktionen mit einem node-link Diagramm sind das räumliche Verschieben und Zoomen, für die lokale Analyse werden sogenannte *Magic Lenses* verwendet. Dabei handelt es sich um lokale Filter, die als Mauszeiger-Werkzeuge angewandt werden wie etwa die *fisheye view*, die eine temporäre Verzerrung des Diagramms erzeugt, durch die weiter außen liegende Knoten beiseitegeschoben wer-

den und den Knoten an der Position des Mauszeigers mehr Platz gibt. Die *Local edge lens* blendet alle Kanten aus, die nicht auf einen Knoten in der Nähe des Mauszeigers zeigen. Neben der Verwendung von Werkzeugen können auch globale Filter verwendet werden, etwa indem charakteristische Knoten hervorgehoben werden. Auch der Vergleich mehrerer Graphen-Visualisierungen kann unterstütz werden. Dazu werden im *graph matching* zwei oder mehr Graphen vereint und anschließend deren strukturelle Gemeinsamkeiten und Unterschiede herausgestellt. Es gibt eine Vielzahl weiterer Methoden, welche die Analyse von Graphen erleichtern, wie etwa die Berechnung des kürzesten Pfades zwischen zwei Knoten mittels Djikstra-Algorithmus. [35]

Die durch die LDA generierten Relationen können auch als Adjazenzmatrix eines Graphen verstanden werden. Da die Verbindungen nicht gerichtet sind, entsteht eine symmetrische Matrix, die als Matrixdiagramm dargestellt werden kann. Das ist eine direkte Visualisierung der Matrix in Form einer Tabelle. Matrixdarstellungen haben gegenüber Node-link Diagrammen den Vorteil, keine Kantenüberlappungen aufzuweisen. Die Ordnung ist zwar Linear und auf einen einzigen Vergleichsfaktor beschränkt, doch können die Zeilen der Matrix beliebig umgeordnet werden, sodass sich unterschiedliche Charakteristiken des Datensatzes identifizieren lassen. Die verschiedenen Anordnungen miteinander in Verbindung zu bringen kann unter Umständen kompliziert sein und wird durch das Markieren einiger Einträge gelöst. Ein weiterer Nachteil dieser Darstellungsform ist, dass sich mehrstufige Verbindungen zwischen Knoten nur schwer verfolgen lassen. [35]

Allgemeinere Designrichtlinien für Anwendungen der visuellen Analyse über relationale Daten haben Dadzie et al. aus mehr als 70 Arbeiten der visuellen Analyse über relationale Daten abgeleitet. In Tabelle 2.4 ist ein Auszug zu sehen. Neben der Regel *„Overiew first, zoom and filter, then details-on-demand“* ist vor allem die individuelle Anpassung des Analyseprozesses wichtig. Nutzer mit individuellen Zielen und unterschiedlichem Vorwissen können bereits beim Einstieg in die Daten unterstützt werden. In der Analyse sollen sich Filter der visuellen Spezifikationen als auch der Datenselektion möglichst frei definieren lassen. Die Beschreibungen in der Tabelle sind selbsterklärend und werden im weiteren Verlauf der Arbeit häufiger referenziert. [36]

3 Evaluation bestehender Arbeiten

In Abschnitt 2.3.1 wurden Shrinivasan et al. zitiert [31] die ein Problem darin sehen, dass die meisten Tools nur den foreaging loop und nicht den sensemaking loop des Sensemakingprozesses nach Pirolli et al. [1] unterstützen. Da Shrinivasan et al. nicht auf die algorithmische Sicht eingegangen sind, soll nun betrachtet werden wie die Sinnstiftung speziell in Anwendungen des Topic Modelings unterstützt wird. Das Ziel ist dabei nicht die Formulierung konkreter Regeln, wie es Wang et al. [32] und Dadzie et al. [36] gemacht haben. Stattdessen soll identifiziert werden in welchem Umfang die Möglichkeiten von Topic Models genutzt werden, ob es sich um reine Visualisierungen handelt, in welchem Grade interagiert werden kann und ob der foreaging loop oder sogar der sensemaking loop aktiv unterstützt werden. Dazu wird in Abschnitt 3.1.1 die Auswahl der betrachteten Arbeiten argumentiert und in Abschnitt 3.1.2 die Indikatoren der Evaluation definiert. Die Analyse der einzelnen Arbeiten erfolgt in Abschnitt 3.2 deren Ergebnisse anschließend zusammengeführt werden.

3.1 Spezifikationen

Die Indikatoren der folgenden Evaluation entsprechen den in Abschnitt 2 beschriebenen Grundlagen und erfassen Aspekte des verwendeten Algorithmus, des Interfaces und des Sensemakings. In der Beschreibung der Algorithmen werden die Parametrisierbarkeit und die Form der Implementierung beleuchtet. Der Algorithmus selbst wird nur beschriebe, wenn er sich von der LDA unterscheidet, oder wenn er eine spezielle Erweiterung der LDA ist.

3.1.1 Auswahl der Arbeiten

Für die Evaluation wurden 9 Anwendungen und Publikationen ausgewählt, die sich mit dem Sensemaking über unstrukturierte Dokumentkollektionen beschäftigen und dabei Topic Models als automatische Analyseverfahren, oder damit eng verwandte Verfahren verwenden. Bei der Auswahl wurde darauf geachtet, dass ein auf Endnutzer ausgerichtetes Konzept verfolgt wird. Es werden aber auch solche Ansätze beachtet, die Einzelaspekte besonders gut behandeln.

Vorab lässt sich feststellen, dass fast alle Arbeiten ein Topic Model als Datengrundlage verwenden und lediglich in *Jigsaw* andere Clustering-Verfahren angewandt werden. Die gewählten Tools befinden sich in unterschiedlichen Entwicklungsstadien. *Serendip, D-Vita, Qiqqa, Carrot*^{2}*, Hiérarchie* und *Jigsaw* gehören zu den fortgeschrittenen, während für *iVizClustering, TopicViz* und *Tiara* noch keine Prototypen existiert, weshalb sie anhand der Publikationen analysiert werden.

3.1.2 Evaluationskriterien

Eine Evaluation ist die Bestimmung von Werten nach bekannten Maßstäben mit dem Zweck der Verbesserung und Vergleichbarkeit von Systemen. Die Qualität der Messung basiert auf ihrer Validität und Reliabilität und muss reproduzierbar sein. Es dürfen nicht nur einzelne Komponenten, sondern deren Zusammenspiel mit dem Ziel der Erfüllung einer Hauptfunktionalität evaluiert werden. Lange Zeit wurde im IIR nur die algorithmische Effizienz gemessen und wenig Aufmerksamkeit auf den menschlichen Nutzen gelegt. Kriterien der explorativen Suche sind generell schwer zu bestimmen, da kein exakt prüfbares Ziel definiert werden kann. In vielen wissenschaftlichen Publikationen, wird die Evaluation der präsentierten Anwendung gänzlich ausgelassen und auf zukünftige Forschungsarbeiten verwiesen. Dadurch lassen sich mehrere Arbeiten nur schwer vergleichen und das Fortschreiten der Forschung wird behindert. In Abschnitt 2.1.4 wurde die Untersuchung von Chang et al. vorgestellt [19], die die Unterschiede zwischen Interpretation und *search performance measures* aufgezeigt hat. Demnach sollte sich eine Evaluation nicht alleine auf die Genauigkeit der retrieval-Methoden konzentrieren, sondern die Verbindung von Algorithmus, Repräsentation, Interaktion und Sinnstiftung beobachtet werden. Qu et al. zufolge ist die Forschung von explorativen Systemen an einem Punkt, an dem die technologischen Aspekte weitaus ausgereifter sind als das Verständnis vom Benutzungsprozess. Daher empfehlen sie die *user centered evaluation*, bei der die Benutzerprozesse vor das Systemdesign gestellt werden. Sie begründen ihre Aussage damit, dass zu Beginn der HCI-Forschung sehr viele und umfangreiche summative Evaluationen durchgeführt wurden. Damit sind Evaluationen fertiger Produkte gemeint, die vorwiegend auf quantitativen Indikatoren basieren. Stattdessen sollten Tools der explorativen Suche formativ Evaluiert werden. Das ist eine Evaluation im Entwicklungsprozess, die verstärkt auf dem menschlichen Urteil als Indikator beruht, wodurch komplexe Vorgänge besser verstanden werden sollen. Um eine formelle Vergleichbarkeit einer Evaluation zu erhalten empfehlen Qu et al. die Verwendung von Modellen der Wissensaufnahme und -verarbeitung, der Informationssammlung und des Sensemakings. Ebendiese Modelle wurden in Abschnitt 2 beschrieben.

Bevor gesondert auf das Interface und das Sensemaking eingegangen wird, wird jede

Publikation mit den Zielen der Autoren und verwendeten Funktionen eingeleitet. Dazu werden die darunterliegenden Daten, die Anwendung des Topic Models und die Programmstruktur beschrieben, in der sich die Elemente vereinen.

Interface

Im Rahmen des Interfaces werden die Visualisierung von Daten, die Repräsentation relationaler Daten, der Umgang mit mehreren Ansichten und die jeweils verwendeten Interaktionskonzepte behandelt. Die Aufteilung in verschiedene Datenansichten richtet sich nach den Regeln von Wang et al., die in Tabelle 2.3 aufgeführt sind. Die Regeln der *Komplementarität* und *Zerlegung* fallen besonders ins Gewicht. Entsprechend wird evaluiert ob die Ansichten redundant sind oder sich ergänzen und in welcher Form sie visuell und auf Datenebene miteinander verbunden sind. Zum Teil werden ebenfalls low-level Interaktionen wie das Selektieren, Verschieben und Organisieren von Elementen beschrieben, um sie anschließend in einen umfassenderen Kontext zu setzen. Umfassende Interaktionen betreffen den Verbund der Ansichten und weitere Charakteristiken der Navigationsstruktur. Dazu zählen die Kosten einzelner Navigationsschritte, die Umkehrbarkeit von Aktionen, der Wechsel zwischen Ansichten und die Navigation zwischen Abstraktionsebenen und Datentypen. In geringerem Umfang werden auch Regeln der Gestaltpsychologie einbezogen um festzustellen wie mit Ähnlichkeiten, Gruppierungen und Hervorhebungen umgegangen wird. Da Topic Models relationale Daten produzieren, werden die Regeln von Dadzei et al. aus Tabelle 2.4 verwendet, um den Umgang mit Relationen zu analysieren. Konkret gestellte Fragen sind unter anderem:

- Wie werden Funktionen, Aufgaben und Datenebenen in verschiedene Ansichten aufgeteilt und kommt es dabei zu Redundanzen?
- Wie sind Ansichten miteinander verbunden?
- Wie lässt sich zwischen Ansichten und Arbeitsbereichen wechseln?
- Wie ist der Datenzoom umgesetzt: wie schnell und feinstufig lässt sich zwischen Abstraktionsebenen wechseln?
- Wie werden Relationen visualisiert?
- Wie werden Themen erklärt und bezeichnet?
- Wird der Einstieg in die Analyse unterstützt?
- Wird ein sensorischer (visueller) Informationsoverload verhindert?

Sensemaking

Für die Evaluation der kognitiven Entlastung werden die verschiedenen Gedächtnistypen und ihre Charakteristiken beachtet, wie sie in Abschnitt 2.2 beschrieben wurden. Besonders wichtig ist die Unterstützung der Orientierung im prozeduralen sowie im semantischen Sinne, die in der Evaluation auch als zeitliche und räumliche Dimension bezeichnet werden. Dazu zählen die Protokollierung von Aktionen und die Möglichkeit, Wissen zu externalisieren. Wichtig ist neben der ausschließlichen Darstellung auch die Integration in den weiterführenden Arbeitsprozess. So ist herauszustellen ob über ein Protokoll auf vergangene Zustände zugegriffen werden kann, ob diese mit aktuellen oder ebenfalls vergangenen Zuständen verglichen werden können und ob die chronologische Linearität eines Protokolls verlassen werden kann, um von vergangenen Zuständen aus alternative Wege einzuschlagen. Für die explorative Suche ist es besonders wichtig, dass der Nutzer seinen Aktionsweg rekapitulieren kann, um die Herleitung der Daten verstehen und bereits gegangene Pfade identifizieren zu können. Eine Externalisierung erfolgt durch das bewusste Festhalten und dem individuellen Umstrukturieren von Resultaten. Besonders positiv ist eine hohe Dynamik in der Erzeugung von Wissensstrukturen, etwa wenn der Nutzer eigene Elemente und Relationen erstellen kann. Auch dort wird beachtet, ob diese Strukturen mit weiteren Datenbefunden verglichen werden können. Es werden auch direkte Aspekte des ID beachtet, da eine Externalisierung nur durch schnelle Modifizierung, und ein Protokoll nur durch einen direkten Zugriff effizient in den Arbeitsablauf integriert werden können. Die angebotenen Funktionen werden auf ihre Dynamik und Individualisierbarkeit hin überprüft. Die Unterschiede der Merkleistungen schwanken von Vorwissen und Zustand des Nutzers erheblich, daher wird analysiert ob durch das Monitoring, also die Externalisierung, eine Behinderung der Effizienz auftritt. Für die Schlussfolgerung von Hypothesen ist die direkte Vergleichbarkeit von Daten wichtig. Daher soll festgestellt werden ob und zu welchem Grad, Daten gegenübergestellt werden können. Dazu zählt unter anderem, ob sich der Nutzer Bezüge über Umwege herleiten muss, oder ob numerische und direkte visuelle Vergleiche geboten werden. Es werden also die drei Ebenen Datenschicht, Visualisierung und Interaktion im Hinblick auf das Sensemaking und die kognitive Entlastung evaluiert. Die folgenden Leitfragen fassen das nochmal zusammen:

- Wie wird die zeitliche und räumliche Orientierung unterstützt?
- Wie wird mit kurzzeitigem (inhaltlichem) Informationsoverload umgegangen und können Informationen mit dem Zweck der Entlastung des Arbeitsgedächtnisses externalisiert werden?
- In welchem Umfang lassen sich Daten vergleichen?

- Können eigene Schemata gebildet werden, etwa indem Daten strukturiert und um eigene Eingaben ergänzt werden?
- Sind solche Schemata weiterhin mit den Daten verknüpft? Wird nicht bloß ein visuelles sondern ein die Daten umfassendes Konzept verfolgt?

3.2 Analyse der einzelnen Arbeiten

Es werden nun neun Arbeiten im Einzelnen analysiert, die das gemeinsame Ziel verfolgen den Sinnstiftungsprozesses in einer explorativen Suche durch zu unterstützen. Der Fokus der Arbeiten liegt unterschiedlich stark auf der explorativen Suche oder der explorativen Analyse. Eine weitere Unterscheidung ist der Entwicklungszustand der Tools, die zum einen für die tatsächliche Nutzung wurden und zum anderen ausschließlich als proof-of-concept implementiert wurden. Auch befinden sich viele Anwendungen der Natur von Forschungsarbeiten gemäß in einem Entwurfsprozess oder behandeln spezielle Teilaspekte. Dadurch wird der Wert der Arbeiten nicht gemindert, doch kann und soll bei der Evaluation nicht auf fehlende Details eingegangen werden. Stattdessen werden grundlegende Fehler aber auch positive Eigenschaften hervorgehoben und verpasste Potentiale herausgestellt. Wie in Abschnitt 3.1.2 beschrieben, handelt es sich um eine nutzerzentrierte und formative Evaluation, die den Variationen einzelner Betrachtungen mehr Raum gibt als es eine quantitative Evaluation kann, ohne dabei willkürlich zu werden. Jedes Tool wird erst mit seinen individuellen Zielen, Techniken und Funktionen vorgestellt und daraufhin für die Bereiche *Interface* und *Sensemaking* genauer betrachtet.

3.2.1 Serendip

Serendip ist ein Tool zum Explorieren von Textkorpora beliebigen Formates mittels LDA. Ziel der Autoren war die effektive Verbindung verschiedener Abstraktionslevel, über die in mehreren *Views* interagiert werden kann [37][1]

Interface

Serendip wird als Desktop-Applikation über einen lokalen Webserver ausgeführt und im Webbrowser verwendet. Die drei Hauptansichten *CorpusViewer*, *TextViewer* und *RankViewer* sind in Abbildung 3.1 zu sehen. Der *CorpusViewer* auf der linken Seite hat als Kernfunktionalität die Visualisierung der *Topic-Document-Matrix*, die eine direkte Ausgabe der verwendeten LDA ist. Auf der horizontalen Achse sind die Themen aufgelistet

[1] Die Publikation ist sehr aktuell und ein Prototyp von Serendip wurde erst im Dezember 2014 veröffentlicht `http://vep.cs.wisc.edu/serendip/` , 11.12.2014

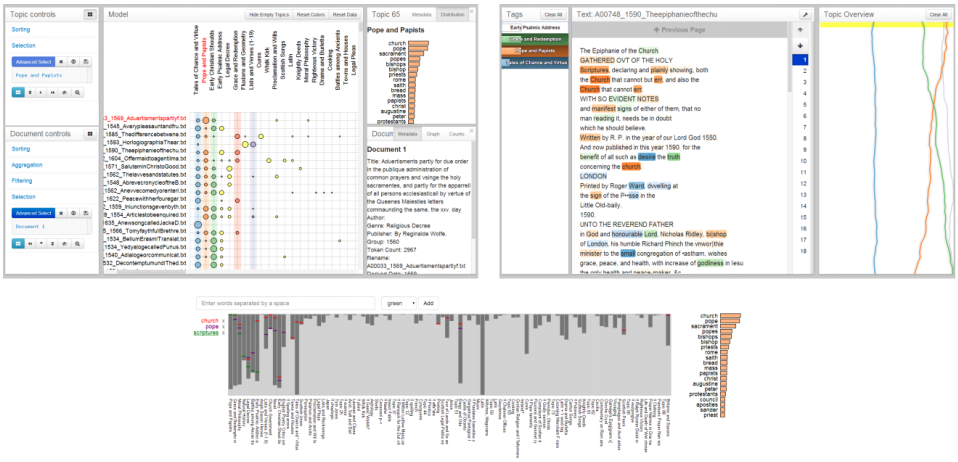


Abbildung 3.1: Die drei Ansichten von *Serendip*: *CorpusViewer*, *TextViewer* und *RankViewer* Quelle: *Alexander et al. 2014*

und auf der vertikalen Achse die Dokumente. Jeder Schnittpunkt bezeichnet eine Relation, wobei die jeweilige Gewichtung durch den Kreisumfang encodiert ist. Im linken Teil sind Werkzeuge zur visuellen Filterung der Themen und Dokumente zu sehen. Die Visualisierung selbst lehnt sich an *Terminte* von Chuang et al. an [38]. Die Tabelle lässt sich auf beiden Achsen global oder bezüglich eines einzelnen Elements sortieren. Etwa können Themen nach ihrer generellen Häufigkeit angeordnet werden, oder nach der Gewichtung in einem speziellen Dokument. Für die Filterung werden Metadaten verwendet, die von der LDA nicht erfasst werden und daher anderweitig generiert worden müssen. Nach Eigenschaften der Relationen kann hingegen nicht gefiltert werden. Es kann lediglich eine Umsortierung der Matrix nach Maximalwerten oder Median der Themen vorgenommen werden. Die Anordnung der Elemente lässt sich auch manuell per *drag-and-drop* anpassen. Die Interpretation einzelner Themen erfolgt auch in Serendip über die Auflistung der am höchsten gewichteten Terme.

Der *TextViewer*, rechts in Abbildung 3.1, zeigt den Inhalt eines einzelnen Dokuments. Daneben ist der Verlauf der prägnantesten Themen in dem Dokument zu sehen. Durch die Auswahl eines Themas werden dazu gehörige Wörter im Text farblich hervorgehoben. Der Verlauf auf der rechten Seite entspricht immer dem ganzen Dokument. Daher wird keine Verbindung zwischen der aktuellen Position im Text und dem Verlaufsgraphen der Themen hergestellt. Es ist auch nicht möglich direkt über die Verlaufsansicht zu einer bestimmten Stelle im Text zu wechseln. Der *RankViewer*, unten in Abbildung 3.1, zeigt den Umfang aller Themen in Form von grauen Balken an. Darin kann nach einzelnen Worten gesucht werden, deren Ränge in jedem Thema durch Striche auf den grauen Balken gekennzeichnet werden. Durch die Selektion eines Balkens werden die Terme des dazugehörigen Themas aufgelistet.

Jede Ansicht hat einen eigenen Datentypen als Schwerpunkt und stellt zugleich den Bezug zu den anderen Ansichten her. Der *CorpusViewer* visualisiert die Dokument-Themen-Matrix, mit der Absicht die Verteilung von Themen im ganzen Korpus zu erkennen. Der *TextViewer* zeigt ein einzelnes Dokument und stellt auch dort den Zusammenhang zwischen Termen und Themen her. Der *RankViewer* gibt eine Übersicht der Termverteilung in den Themen und ist die einzige Ansicht, die auf die direkte Verbindung zu Dokumenten verzichtet. Die Verbindung der Daten eines Topic Models erfolgt über Themen. Entsprechend sind die drei Ansichten über die Selektion und das hervorheben von Themen miteinander verbunden. Nur im *RankViewer* fehlt eine Farbliche Markierung, was die Übertragung von Informationen zu den anderen Ansichten merklich beeinflusst. Die Verknüpfung der Ansichten beschränkt sich leider darauf und es werden keine Filter oder Modifikationen übernommen. Das Löschen, Umbenennen oder Umordnen von Themen in der *Topic-Document-Matrix* hat keinen Einfluss auf den Rest des Programmes, es handelt sich damit um rein visuelle Veränderungen, die nicht auf den Datenbestand übertragen werden.

Die Matrix-, Rank- und DocumentView stellen jeweils unterschiedliche Abstraktionen der Daten dar. Zwischen ihnen kann über Selektionen einzelner Elemente einfach navigiert werden. Dadurch erfolgt jedoch ein Wechsel des Arbeitsbereiches, der eine kontextuelle Anpassung verlangt. Des Weiteren besteht keine Verbindung vom RankView zu einer anderen Ansicht, sodass der Workflow dort unterbrochen wird. *Serendip* ist kein Recommender-System, es gibt keine direkten Vorschläge für weiterführende Dokumente und Themen. Diese Informationen werden stattdessen detailliert in der Matrix dargestellt, so wie sie vom Topic Model erzeugt werden. Es erfolgt also keine Abstraktion von Informationen in visueller Hinsicht und der Nutzer ist bei der Erfassung von Charakteristiken auf sein eigenes Urteil gestellt.

Die initiale Benennung der Themen ist eine einfache Durchnummerierung, die vom Nutzer selbst angepasst werden muss. Der Such-Einstieg erfolgt über die *Topic-Document-Matrix* die entweder nach Themen sortiert wird, falls der vorliegende Datenbestand völlig unbekannt ist. Alternativ kann die Matrix nach einem bekannten Dokument sortiert werden, von dem sich anschließend weiter fortbewegt werden kann. Bei der Verwendung einer hohen Themenzahl[2] erschwert die fehlende Benennung der Themen die Orientierung.

Sensemaking

Da die Daten nicht abstrahiert oder vorgefiltert werden, können die vielen Informationen der *Topic-Document-Matrix*, die nicht bezeichneten Themen und die einzelnen Verläufe in

[2]Etwa im auf der Website gegebenen Beispiel, der Analyse von Shakespeares Werken, über 50 Themen `http://vep-test.cs.wisc.edu/serendip/corpus:ShakespeareChunkedOptimized_50/matrix` , 11.12.2014

der *Topic Overview* auf den ersten Blick überfordern. Nach der Erlernung der Interaktionsmöglichkeiten, insbesondere dem schnellen Umordnen in der Matrix und dem Highlighten in der *Topic Overview*, kann effektiv mit den Daten gearbeitet werden. Für eine effiziente Nutzung von *Serendip* wird daher ein längerer Gebrauch vorausgesetzt. Eine Reduktion der Informationslast erfolgt nur durch die Lernkurve des Nutzers und wird nicht durch die Applikation unterstützt.

Da es nur drei Ansichten gibt ist die Orientierung in den Daten leicht zu behalten. Um vorübergehend relevante Themen festzuhalten lassen sich diese farblich markieren. Die Verbindungen zwischen den Ansichten sind nicht immer klar hergestellt, da eine Umsortierung in der Topic-Document-Matrix keinen Einfluss auf die anderen Ansichten hat. Im *DocumentViewer* wird der Text und der Themenverlauf eines Textes angezeigt, ohne diese zu verknüpfen. Die *Topic Overview* zeigt den Verlauf im gesamten Dokument und es lässt sich nicht darauf schließen, an welcher Stelle man sich momentan befindet. Die fehlende Verbindung der beiden Ansichten erschwert die Orientierung der Positionsbestimmung und der Themenrelevanz im vorliegenden Textabschnitte erheblich. Eine Dokumentation des Aktionshergangs findet nicht statt.

Durch die direkte Wiedergabe der Topic Model-Daten ist der visuelle Vergleich von einzelnen Themen und Dokumenten möglich. Elemente lassen sich nicht gruppieren, sodass eine direkte Gegenüberstellung mehrerer Elemente nicht möglich ist. Im *RankViewer* erfolgt eine Gegenüberstellung von Themen anhand des globalen Umfangs, doch es wird kein Bezug zum aktuell ausgewählten Dokument hergestellt.

Zusammenfassung

Serendip filtert die vom LDA erstellten Informationen nicht vor und bietet dadurch einen hohen Detailgrad mit vielen Analysemöglichkeiten, die allerdings eine hohe Merkleistung erfordern. Die Daten werden in drei komplementären Ansichten dargestellt, die von der Übersicht zum Detail linear verbunden sind. Durch den automatischen Wechsel und durch farbliche Kennzeichnungen wird die Aufmerksamkeit des Nutzers effektiv gelenkt. Direkte Vergleiche als auch die Externalisierung gefundener Informationen werden nicht unterstützt. Der Zoom über verschiedene Informationslevel und Datentypen ist in Serendip gut gelöst, allerdings beschränkt sich die Analyse auf das Filtern, Ordnen und Betrachten von Visualisierungen, ohne dass das Datenmodell modifiziert werden könnte.

3.2.2 Qiqqa

Qiqqa ist ein Programm[3], dass auf die Organisation von wissenschaftlichen Publikationen ausgerichtet ist. Dateien im PDF-Format können importiert und per Optical Character Recognition (OCR) in Text umgewandelt werden, auf den anschließend dem Topic Modeling verwandte Verfahren angewandt werden. Fehlende Metadaten können automatisch über Google-Scholar[4] bezogen werden.

Interface

Qiqqa besteht aus den drei Bereichen *Bibliothek, Expedition* und *Brainstorm*. Die Bibliothek listet alle importierten Dokumente auf. Dort können Filter über Metainformationen, Tags und Themen angewandt werden. Beliebige Teile der Bibliothek können in der Expedition mittels LSA analysiert werden. Die Form der Resultate der LSA gleichen denen der LDA, bloß dass keine stochastischen sondern algebraische Methoden verwendet werden. Die Relationen werden daher auf gleiche Weise beschrieben, aber auf verschiedene Weisen bestimmt.

Abbildung 3.2 zeigt die Expeditionsumgebung. Im Zentrum steht der Inhalt eines Dokumentes, das umringt ist von weiterführenden Inhalten. Darunter sind die im Dokument behandelten Themen als Piechart aufgeführt und daneben Dokumente aufgelistet, die eine hohe Ähnlichkeit aufweisen. Der Linke Rand zeigt mehrere farblich gekennzeichnete Themen mit dazugehörigen Dokumenten. Unter jedem Thema ist eine *Show me in Brainstorm*-Schaltfläche zu sehen, über welche die Brainstorm-Ansicht mit den Dokumenten des jeweiligen Themas geöffnet wird. Alternativ können Dokumente in der Bibliothek ausgewählt werden, um sie in der Brainstorm zu betrachten. Abbildung 3.3 zeigt zwei nebeneinander gestellte Brainstorm-Graphen. Darüber liegt die Vorschau eines Dokumentes die erscheint, wenn der Mauszeiger über einen Dokumentknoten gehalten wird. Die farbigen Rechtecke neben jedem Dokument zeigen die Gewichtung einzelner Themen. Der Nutzer soll damit schnell ausmachen können, in welchem Bezug zwei Elemente stehen. Über Tastenkürzel können verschiedene Eigenschaften eines Knoten expandiert werden. Dazu zählen die wichtigsten Themen, ähnliche Dokumente, das wichtigste Dokument zu den wichtigsten im Knoten enthaltenen Themen, Autoren, Tags und Zitationen.

Die Aufteilung der Programmstruktur erfolgt nicht anhand von Datentypen, sondern der darauf anwendbaren Funktionen. Die Arbeitsbereiche sind weitgehend eigenständig und über Weiterleitungen geringfügig miteinander verknüpft, es gibt nur wenige direkte Verbindung. Ein direkter Wechsel zwischen Brainstorm und Expedition ist nicht möglich, stattdessen muss ein Umweg über eine Detailansicht gegangen werden. Diese Detailan-

[3] *Qiqqa* wird kostenlos angeboten auf `http://www.qiqqa.com/` , 25.11.2014
[4] Google-Scholar: `http://scholar.google.com/` , 25.11.2014

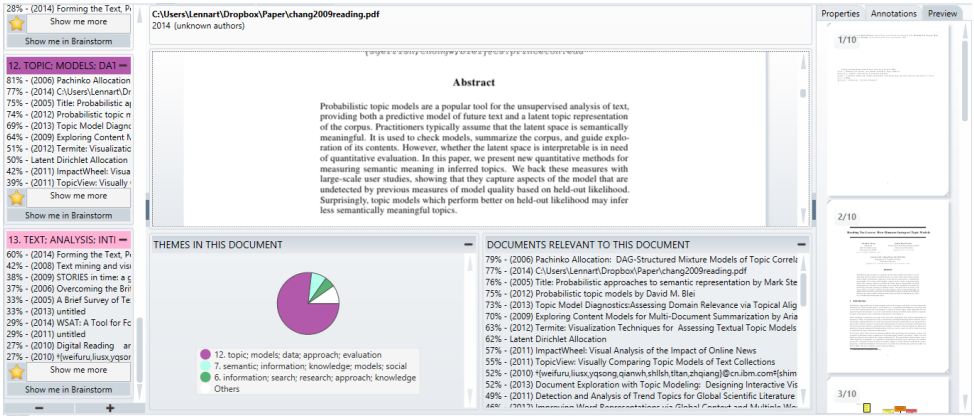


Abbildung 3.2: *Expedition*-Ansicht in *Qiqqa*. *Quelle: Screenshot der Anwendung Qiqqa v.67s*

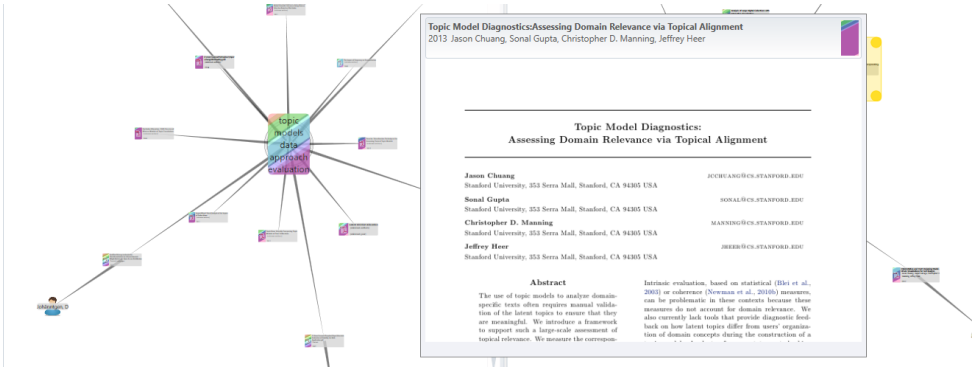


Abbildung 3.3: *Brainstorm*-Ansicht in *Qiqqa*. *Quelle: Screenshot der Anwendung Qiqqa v.67s*

sicht kann auch direkt von der Expedition erreicht werden, gibt jedoch außer Zitationen keine weiteren Informationen. Die Übergänge zwischen den Ansichten sind daher nicht fließend und können den Prozess der explorativen Analyse behindern, der ein schnelles Wechseln zwischen Detail, Übersicht und Externalisierung benötigt. Die Bibliothek gibt eine Übersicht in Form einer Liste, während die Expedition und Brainstorm mehrere Informationsebenen verbinden. Die Betrachtungsweise und die Interaktionsmöglichkeiten der beiden Arbeitsbereiche unterscheiden sich, obwohl sie auf den gleichen Daten operieren. In der Expedition wird das Betrachten einzelner Dokumente unterstützt, während im Brainstorm die Relationen von Dokumenten angezeigt werden. Die Darstellung ist konsequent und das Layout übersichtlich. Themen können klar durch ihre Färbung identifiziert werden. Zwischen den drei Arbeitsbereichen kann manuell schnell gewechselt werden, doch es bestehen nur wenige direkte Verbindungen. Das Maß der Themengewichtung ist in allen Ansichten klar verständlich, auf eine Gewichtung der Terme wird verzichtet. Um in der Brainstorm die Informationen einzelner Dokumente zu lesen muss visuell herangezoomt werden. Durch diesen Aufwand ist immer nur ein einzelnes Dokument zu sehen, wodurch

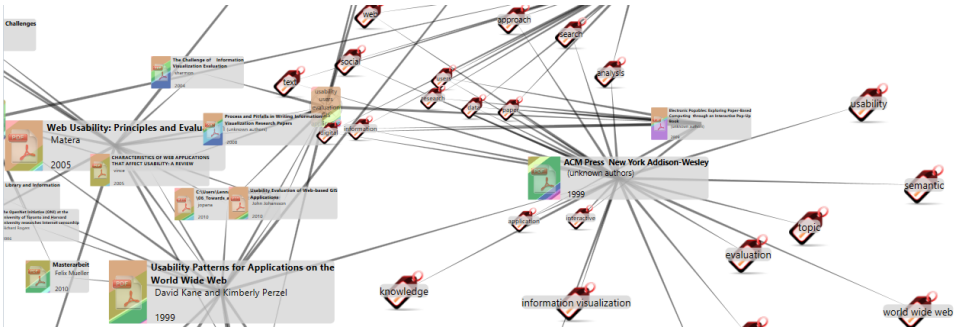


Abbildung 3.4: Durch das Auskappen von Tags wird die *Brainstorm* nach wenigen Schritten unübersichtlich. *Quelle: Screenshot der Anwendung Qiqqa v.67s*

der umfassende Bezug verloren geht. Die Informationsebenen werden in allen Arbeitsbereichen direkt in statischer Form nebeneinander gestellt. In der Exploration als auch der Brainstorm lassen sich genauere Informationen über die Inhalte einzelner Themen und Dokumente nicht in Erfahrung bringen.

In der Expedition werden Dokumentempfehlungen mit der entsprechenden Kosinusdistanz angegeben. Da keine Details über die Form der Relation gegeben werden, muss ein Nutzer den Vorschlägen vertrauen und könnte verleitet werden, den jeweils ersten Empfehlungen zu folgen. Inhaltliche Ähnlichkeiten müssen durch die direkte Betrachtung einzelner Dokumente gefunden werden. Auch die Gruppierung nach Themen[5] hilft dort nicht weiter, da sie ausschließlich die Relevanz der Dokumente zu den jeweiligen Themen anzeigt. In der Brainstorm sind Relationen als Kanten eines Graphen angegeben, ohne das Gewicht oder die Art der Relation darzustellen. Dafür ist die Themengewichtung eines jeden Elements farblich markiert, wie es auch in der Exploration erfolgt. Durch die vielen farbigen Flächen steigt der Detailgrad stark an, sodass es schwer fällt die gegebenen Informationen rasch zu verstehen und Ansichten übergreifend einzuordnen. Die Arbeitsbereiche eignen sich daher vor allem für die separate Nutzung.

Themen werden in jeder Ansicht durch die vier wichtigsten Terme beschrieben. In dem vor allem mit Listen gefüllten Arbeitsbereich der Expedition, lassen sich Themen ausreichend schnell erkennen, was in der Brainstorm durch eine Vielzahl von Elementen erschwert wird. Beispielsweise werden auch die Themen selbst durch ihre Relationen zu anderen Themen dargestellt, wie in Abbildung 3.3 zu sehen ist.

Der Einstieg in die Exploration erfolgt in *Qiqqa* über die Volltextsuche in der Bibliothek. Da der Fokus auf wissenschaftlichen Publikationen liegt, wird einem Benutzer die Suchdomäne einigermaßen bekannt sein, sodass er bereits eine Auswahl relevanter Artikel hat von denen er weiter vorgehen kann, oder er entsprechende Schlagworte formulieren kann. Alternativ können alle Dokumente der Bibliothek mittels LSA in Themen separiert

[5]In der Explorations-Ansicht auf der linken Seite.

werden. Ein Nutzer kann sich dann vom für ihn interessantesten Thema weiter vorarbeiten.

Sensemaking

Die präsentierten Informationen sind sehr detailliert und alle Zusammenhänge müssen selbstständig erfasst werden, die Resultate der LSA werden also direkt wiedergegeben. Eine algorithmische Unterstützung durch die Berechnung verschiedener Abstraktionsstufen erfolgt nicht. In Abbildung 3.4 wurden *AutoTags* dreier Dokumente verwendet. Durch die hohe Zahl an Elementen und die vielen Überschneidungen lassen sich nur schwer zentrale Zusammenhänge erschließen. Die Brainstorm ist demnach sehr schnell überladen. Das nutzen von Piktogrammen als Anlehnung an Objekte der analogen Welt werten Dadzie et al. [36] positiv, was sich im Falle der Brainstorm allerdings nicht bestätigt. Bei einer hohen Dichte von Objekten überlagern sich die Piktogramme und verstopfen vielmehr die Sicht als beim Verstehen zu helfen. Durch einzelne Aktionen können bereits viele weiterführende Knoten aufgeklappt werden, die manuell eliminiert werden müssen.

Es werden in keinem Arbeitsbereich Zustände festgehalten, sodass Aktionen nicht rückgängig gemacht werden können. In der Brainstorm können Elemente daher nur ausgeklappt, aber nicht wieder zusammengefaltet werden. Abbildung 3.4 zeigt deutlich, wie hinderlich diese Einschränkung ist. In der Expedition wird nicht angezeigt welche Dokumente zuvor betrachtet wurden und in welchem Bezug sie zu dem aktuellen Dokument stehen. Die chronologische Orientierung wird demnach überhaupt nicht unterstützt. Die Orientierung in den Daten ist in der Expedition nur über die Gruppierung der Themen möglich. In der Brainstorm können Detailinformationen durch ein hereinzoomen betrachtet werden. Damit ist dem Prinzip der *detail-on-demand* entsprochen. Es gibt keine Übersicht, sodass sich die Betrachtung von Details schwer in ein globales Konzept fügen lässt.

Alle Ansichten lassen sich nebeneinander stellen, wodurch das parallele Arbeiten in Expedition und Brainstorm ermöglicht wird. Entsprechende Bezüge muss ein Nutzer allerdings selbstständig herstellen. Zwei nebeneinandergestellte Brainstorm-Graphen lassen sich also nur visuell vergleichen, was äußerst schwer ist, da die Anordnung als force-directed-graph quasi zufällig erfolgt.

Der Nutzer kann eigene Elemente wie Tags oder Labels in die *Brainstorm*-Ansicht einfügen. Alle Elemente können zusätzlich zu den gegebenen Relationen frei verknüpft werden. Der Graph ist visuell durch die Verjüngung seiner Kanten gerichtet, die allerdings nicht in Beziehung zu den tatsächlichen Datenrelationen stehen und sich ausschließlich aus der Reihenfolge des Hinzufügens der Knoten ergeben. Die Modifizierung der Ansicht beschränkt sich auf visuelle Attribute. Elemente können verschoben, verknüpft, gelöscht und hinzugefügt werden, dies alles hat jedoch keinen Einfluss auf andere Ansichten, oder den

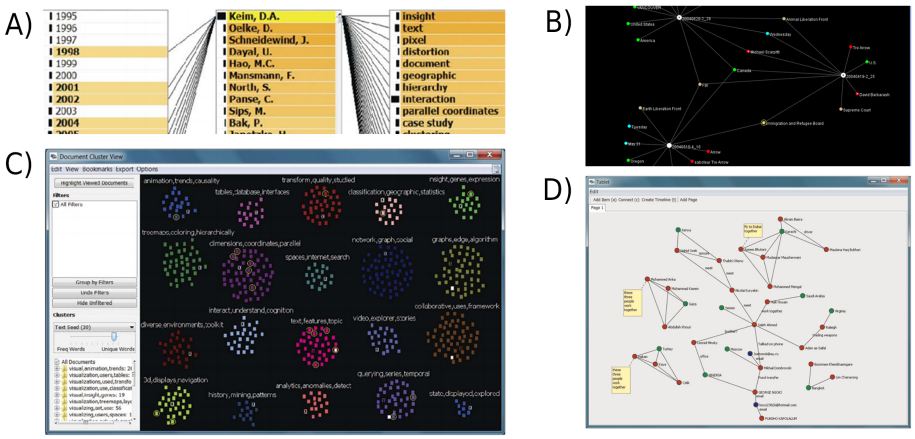


Abbildung 3.5: Ausschnitte einiger der Ansichten von *Jigsaw*: *(A) List View, (B) Graph View, (C) Document Cluster View* und *(D) Tablet. Quelle: Stasko et al. 2008; Görg et al. 2013*

Datenbestand selbst. Abgesehen von einer visuellen Anordnung besteht keine Möglichkeit Dokumente zu gruppieren, sodass eine schnelle Organisation der Daten nicht möglich ist.

Zusammenfassung

Die Aufteilung der Daten in Arbeitsbereiche ist gut und klar getrennt, wobei die Verbindung nicht fließend ist und somit der Wechsel der Datenbetrachtungen und auch der Abstraktionsebenen erschwert wird. Durch den Zwang, Arbeitsbereiche wechseln zu müssen, wird ein umfassendes Verständnis der Daten behindert. Auch der Aktionsverlauf wird nicht festgehalten, sodass sich Aktionen nicht umkehren lassen und nicht ersichtlich wird, welche Suchpfade bereits begangen wurden. Die fehlende Umkehrbarkeit führt dazu, dass der Brainstorm-Bereich nicht gut auf große Datenmengen skaliert werden kann. Die Art der Beziehung von Dokumenten lässt sich nicht im Detail erfassen, weshalb der Nutzer auf das Kosinusmaß vertrauen muss. Mehrere Dokumente, oder Elemente anderen Typs lassen sich nicht miteinander vergleichen.

Der foreaging loop wird insoweit unterstützt, dass sich Daten in der Brainstorm ansammeln lassen. Das Einfügen von selbst bezeichneten Labels und das freie Verbinden von Elementen kann als eine einfache Form der Schemabildung gesehen werden. Die node-link Graphen der Brainstorm sind jedoch rein visuell und haben keinen weiteren Einfluss auf die Daten und können auch nicht als Filter verwendet werden.

3.2.3 Jigsaw

Jigsaw ist ein Tool zur Unterstützung des Sensemakings in großen Textkollektionen, mit dem Schwerpunkt auf unformatierten Berichten. Die Umgebung kann allerdings auch auf

längere Dokumente, Bücher oder Webseiten angewandt werden. Besonders interessant macht Jigsaw, dass die Autoren Stasko et al. sich direkt auf die von Pirolli et al. definierten foreaging und sensemaking loops beziehen [39, 1]. Die Evaluation erfolgt auf Basis der Publikationen [40, 39] von Stasko et al. und der Betrachtung des Tools[6] in der Version 0.53. Da es sich um ein Betastadium handelt konnten einige Funktionen nicht getestet werden[7].

Jigsaw verwendet kein Topic Model und durchsucht stattdessen Dokumente auf Schlagworte vordefinierten Typs. Die Typen sind vom verwendeten Algorithmus abhängig und entsprechen in der Regel Angaben über Zeit, Ort, Personen oder Organisationen. Passende Begriffe werden als Entitäten erfasst und anhand gemeinsamen Vorkommen in Relation gesetzt. Darüber lassen sich Dokumente Clustern, wobei die Cluster als Themen interpretiert werden können. Die Datenstruktur unterscheidet sich von einem Topic Model insofern, dass neben den Relationen zusätzlich Informationen über den Typ der Entitäten vorliegen. Davon abgesehen ist die Gemeinsamkeit der Datenstrukturen sehr stark, da in beiden Fällen aus der Frequenz des gemeinsamen Auftretens von Termen, Cluster gebildet werden. Alternativ kann dieses Verfahren auch auf Metadaten, oder direkt auf die Texte angewandt werden. Dadurch entsteht eine hohe Dimensionalität, die reduziert wird, in dem alle Terme die in weniger als drei Dokumenten vorkommen ausgeschlossen werden. Anschließend wird über eine SVD die LSA durchgeführt.

Interface

Jigsaw bietet eine Vielzahl von Ansichten, wovon Abbildung 3.5 die vier wichtigsten zeigt. Die restlichen Funktionen befasst sich mit der Analyse von Wortfolgen und inhaltlichen Stimmungen, die an sich zwar sehr interessant sind, aber auf von Topic Models sehr verschiedenen Verfahren basieren. Der Grad des Details steigt von C über B nach A hin an. In C werden die Dokumente in einer Übersicht zusammengefasst. Jeder Punkt entspricht einem Dokument und jede Gruppe einem Cluster. In B werden die Relationen und deren Entitäten erkundet und in A lassen sich detaillierte Relationsgewichtungen der Entitäten ausmachen. Die Ansichten sind über das hervorheben beliebiger Selektionen miteinander verbunden, was den Kriterien der *Komplementarität* und *Aufmerksamkeitslenkung* entspricht. Dem Nutzer ist es demnach möglich über die verschiedenen Abstraktionsebenen hinweg Gemeinsamkeiten der Elemente und Cluster auszumachen. Beispielsweise lässt sich in C nicht die Art der Verbindung zweier Dokumente unterschiedlicher Cluster erfassen. Über B ist das problemlos möglich. Andersherum kann ein Element in A oder B selektiert werden, um in C zu sehen welchen Clustern es zugeordnet ist. *Jigsaw* erfüllt, mit

[6] Kostenlos zum Download angeboten auf `http://www.cc.gatech.edu/gvu/ii/jigsaw/`, 02.11.2014
[7] Es kam besonders beim Clustern größerer Dokumentkollektionen zu häufigen Programmabstürzen.

Ausnahme der grundlegenden Designprinzipien von *Sparsamkeit* und *Konsistenz*[8], soweit die von Wang et al. erstellten Kriterien [32].

Die Exploration beginnt in Jigsaw mit der Berechnung von Clustern. Die Zahl der Cluster wird vom Nutzer festgelegt, wobei der optimale Wert, wie es auch bei Topic Models der Fall ist, vorab nicht bekannt ist. Jigsaw ist darauf angelegt mehrfach zu clustern, wobei die verschiedenen Ergebnisse anschließend miteinander verglichen werden können. Anschließend erfolgt der Einstieg in die Analyse über Ansicht C und ist somit visuell, wodurch ungeübten Benutzern ein schneller Zugang ermöglicht wird [36].

Mit Ansicht C ist eine globale Datenübersicht gegeben die jederzeit die Einordnung von Detailinformationen ermöglicht. Das Prinzip der *Details auf Anfrage* wird durch die Aufteilung der Ansichten bestätigt. Des Weiteren lassen sich in Ansicht B Elemente durch einfache Aktionen beliebig aufklappen und wieder zusammenfalten. Im Gegensatz zu Topic Models ist es durch das Clustern von Entitäten möglich, die Relationen in Ansicht B zu bezeichnen. Die Relation zwischen zwei Dokumenten wird über mindestens eine Entität beschrieben, deren Typ sich über die farbliche Kodierung bestimmen lässt. Im Gegensatz zu diesem Verfahren arbeiten Topic Models rein stochastisch, wodurch eine Relationsbezeichnung in dieser Form nicht möglich ist.

Die Bezeichnung der Cluster erfolgt durch die Entitäten mit der höchsten Frequenz. Bei der Analyse von Dokumenten eines speziellen Themenbereiches kommt es dadurch zu Mehrfachbenennungen. Etwa ergaben sich die folgenden Bezeichnungen bei der Analyse von wissenschaftlichen Arbeiten zur Informationsvisualisierung [40]:

- visual; animation; trends
- visualization; users; tables
- visualizations; used; transformation
- visualization; use; classification
- ...

Die meisten Themen fangen mit einer Form von *visual* an und werden von *use* gefolgt. Zwischen Themen zu differenzieren wird auf Grund der feinen Unterscheidungen entsprechend erschwert.

Sensemaking

Die Autoren von *Jigsaw* überlassen dem Nutzer bewusst eine hohe Menge von Detailinformationen und unterstützten ihn nur durch deren Gruppieren und nicht durch ein

[8] Ansichten stehen unverbunden nebeneinander und verfolgen kein einheitliches Layout, was dem konzeptionellen Zustand von *Jigsaw* zu schulden ist.

automatisches Filtern. Durch die farbliche Markierung und örtliche Zusammenfassung ergeben sich Sinneinheiten, die trotz eines hohen Detailgrades gut erfasst werden können. Die Informationslast von Ansicht *B* kann individuell gestaltet werden, indem ein Graph visuell nach einzelnen Typen gefiltert wird, ohne die selektierten Daten selbst zu beeinflussen. Alternativ kann die Informationsdichte temporär reduziert werden, indem nur bestimmte Entitätstypen angezeigt werden. Außerdem können alle Knoten eines Graphen einfach zusammengeklappt werden, um momentan nicht benötigte Relationen und Knoten auszublenden.

Die Zustandsfunktion ist in der betrachteten Version von *Jigsaw* noch nicht vollständig implementiert, jedoch ist ersichtlich, dass ein schnelles Wechseln zwischen vorangegangen Zuständen angestrebt wird. Es wird kein automatisches Aktionsprotokoll geführt. Dafür hat der Nutzer die Möglichkeit, die Zustände jeder beliebigen Ansicht im *Tablet* festzuhalten, dass in Ansicht *D* der Abbildung 3.5 zu sehen ist. Zum Festhalten von Zuständen auf dem *Tablet*, werden Thumbnails der jeweiligen Ansicht erstellt. Dadurch lassen sich die Inhalte der festgehaltenen Zustände schnell erfassen. Durch einen Klick auf das Thumbnail wird der vergangene Zustand wieder hergestellt. Die Verbindung zu tatsächlichen Datenzuständen erhebt das Tablet von einer reinen Ablage zu einem aktiven Instrument. Neben der zeitlichen Orientierung gelingt auch die räumliche Orientierung, da sich jede Aktion einer Auswirkung auf alle anderen Ansichten hat.

Das *Tablet* fungiert wie ein komplexer Notizblock. In den ersten Versionen von Jigsaw wurde es noch als *Shoebox* bezeichnet und war damit, entsprechend des Sensemaking-Prozesses von Pirolli et al. [1], auf den Zweck beschränkt Informationen temporär zu speichern [41]. Mittlerweile wurde daraus ein komplexer allzweck-Arbeitsbereich, in dem Informationen weiterhin ausgelagert werden können, aber sich darüber hinaus auch eigene Schemata bilden lassen, wie der Graph in Ansicht *B* zeigt. Zustände und Entitäten können ohne großen Aufwand dort abgelegt, umbenannt, vernetzt und gruppiert werden. Des Weiteren können eigene Label verwendet werden um einzelne Objekte genauer zu beschreiben. Damit erfüllt Jigsaw die Aufgaben des foreaging loops und einen Teil des sensemaking loops. Die Hypothesenbildung und die umfassende Anwendung gebildeter Schemata werden derweil noch nicht unterstützt.

Daten lassen sich visuell vergleichen, indem eine beliebige Ansicht in ihrem momentanen Zustand dupliziert wird. Somit können zwei verschiedene Pfade verfolgt werden. Der Vergleich über Ansichten hinweg wird durch ein globales Hervorheben von Selektionen vereinfacht. Auch lassen sich die Ergebnisse mehrerer Clustering-Vorgänge vergleichen. Der Vergleich von Daten beschränkt sich vorwiegend auf die visuelle Form.

Zusammenfassung

In der Beurteilung und Schlussfolgerung wird der Nutzer nur geringfügig unterstützt. Jigsaw bietet viele Möglichkeiten Daten zu vergleichen, diese Vergleiche müssen aber manuell und rein visuell durchgeführt werden. Bei großen Dokumentkorpora kann diese Aufgabe sehr schwer sein. Die Autoren von Jigsaw, Stasko et al. haben sich direkt am sensemaking Prozess von Pirolli et al. orientiert und gezielt versucht den foreaging und den sensemaking loop zu unterstützen. Dazu haben sie mit dem *Tablet* einen Arbeitsbereich entworfen, in dem Informationen festgehalten und strukturiert werden können. Dadurch wird die *Shoebox* mit der Schemabildung verbunden und das Tablet wird im ersten Schritt von einer einfachen Ablage in ein aktives Werkzeug umgewandelt. Die im Tablet gebildeten Strukturen lassen sich nicht wieder rückwirkend auf die Daten anwenden, weshalb der sensemakig loop auch in Jigsaw noch nicht geschlossen wird.

3.2.4 iVisClustering

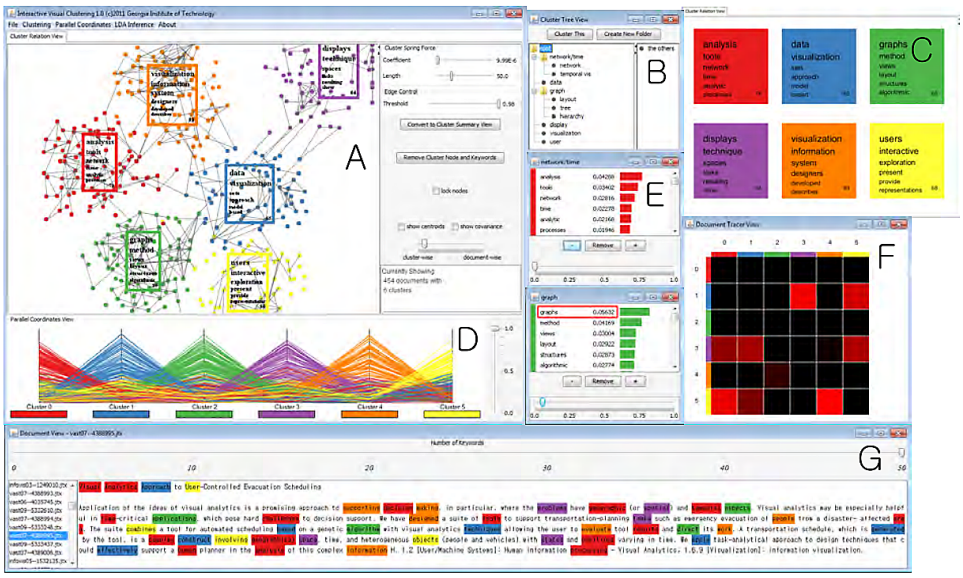


Abbildung 3.6: Ansichten von *iVisClustering*: (A) *Cluster Relation View* (B) *Cluster Tree View* (C) *Cluster Relation View* (D) *Parallel Coordinates View* (E) *Term-Weight View* (F) *Document Tracer View* (G) *Document View*. *Quelle: Lee et al. 2012*

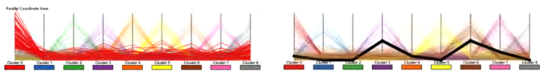


Abbildung 3.7: Die als *X-ray mode* Hervorhebung der Themengewichtung eines Themas (links) und eines einzelnen Dokuments (rechts). *Quelle: Lee et al. 2012*

iVisClustering ist ein Tool zur Sinnstiftung in umfangreichen Textkollektionen unter Verwendung von Clustering. Der Fokus der Autoren Lee et al. liegt auf der optimalen Interaktion mit dem Algorithmus durch dessen direkte Parametrisierung und damit der Optimierung der Clusterergebnisse. Als Algorithmus wird die LDA verwendet, deren bekannten Defiziten, wie der Erzeugung verallgemeinernder Themen und *junk-topics*[9], entgegen gewirkt werden soll [42]. Als Datengrundlage werden unformatierte Texte verwendet, die mit den üblichen Methoden vorverarbeitet werden, um irrelevante Terme zu filtern und die Dimensionalität zu reduzieren. Die folgende Evaluation erfolgt auf Basis der Publikation [43] von Lee et al. [10], ein Prototyp liegt nicht vor.

Interface

iVisClustering besteht aus acht Ansichten, die nach Funktionalität separiert sind, dabei jedoch partiell redundant sind. Abbildung 3.6 zeigt die gesamte Anwendung mit der Bezeichnung der einzelnen Fenster, die in der folgenden Beschreibung verwendet werden. A plottet alle Dokumente in einem force-directed Graph. Jedem Punkt wird die Farbe des dominantesten Themas zugewiesen. Über den Clustern zentriert, wird das jeweilige Thema durch dessen wichtigsten Terme beschrieben, die nochmals in C wiederholt werden. B zeigt eine hierarchische Ordnung der Themen, auf die erst später eingegangen werden soll. Im Parallel-Coordinate-Plot in Ansicht D entspricht jede Achse einem Thema und jede Linie einem Dokument. Durch die Überlagerung einer Vielzahl von Dokumenten lassen sich generelle Trends zwischen Themen ausmachen. Im Beispiel der Abbildung sind die Themen ungewöhnlich gleichmäßig ausgeprägt und klar voneinander separiert. Das ist sicherlich ein optimaler Fall, dem eine längere Optimierung des Topic Models vorausgegangen sein muss. Bei genauerem Hinsehen fallen dennoch einige Trends auf, wie etwa, dass das Thema Blau häufig mit Thema Gelb auftritt. Die Nähe der Cluster in Ansicht A bestätigt diese Ausprägung. F erfasst die Unterschiede die durch die Veränderung der LDA-Parameter entstehen und G zeigt den Inhalt eines Dokumentes mit der Färbung einzelner Wörter nach dem jeweils dominantesten Thema.

Die Redundanz der Ansichten ist nicht negativ, da die gleichen Informationen in unterschiedlichen Kontexten dargestellt werden. So haben die Darstellungen der Terme in A, E und C jeweils andere Funktionen. A verbindet Terme mit den Relationen der Themen, C ist eine Legende und vereinfacht das Selektieren von Themen und E zeigt die genaue Ordnung und Gewichtung von Termen innerhalb eines Themas. Das gleiche gilt für Themen, die in D als Trends angezeigt werden und dann in A auf einzelne Verbindungen überprüft werden können. Außerdem hat D eine höhere Dimensionalität und kann Verbindungen

[9] Themen die keine sinnhafte Gesamtheit bilden.
[10] Lee hat zuvor bereits an *Jigsaw* partizipiert. [39]

aufzeigen, die durch die Projektion auf zwei Dimensionen in A verloren gegangen sind.

A und G haben eigene visuelle Filter, die sich nicht auf andere Ansichten auswirken. In A wird ein Threshold für die Gewichtung von Kanten gesetzt und in G eines für die einzufärbenden Themen. A bildet das Interaktionszentrum, da sich dort die Parameter des Topic Model eingestellt und einzelne Dokumente selektiert werden. Die Selektion definiert den Inhalt von D, E und G. In B werden die Themencluster in einer Baumstruktur angezeigt. Die Clusterbildung wird per *drag-and-drop* parametrisiert, sodass einzelne Cluster gelöscht, zusammengefügt oder Untergruppen erstellt werden können. Die Anzahl der Cluster definiert die Granularität der Themen- und Dokumentseparation. Alle Ansichten sind somit partiell über die Aktionen in A, B und C vernetzt, wodurch die Aktion über mehrere Abstraktionsebenen im Sinne der *Zerlegung* erfolgt. Den Filtern fehlt es jedoch an *Konsistenz* da sie nicht global wirken, sodass etwa eine Themenselektion in C keine Auswirkung auf A oder G hat.

In A werden die einzelnen Dokumente in einem force-directed Graph entsprechend ihres Themenvektors positioniert. Die Themen wirken mit einer Kraft relativ zu ihrer dokumentspezifischen Gewichtung. Die Kanten zwischen den Knoten sind nicht bezeichnet, die Zugehörigkeit ist allein der Position und Färbung zu entnehmen. Jedes Dokument d_i wird einem Thema j zugewiesen, das sich aus $j = arg\,max_{j \in T} \theta_{ij}$ mit T als Menge aller Themen und θ_i als Themenvektor von d_i ergibt. Durch die eindeutige farbliche Markierung ist keine explizite Bezeichnung der Themen nötig. Problematisch könnte dieses Vorgehen bei höheren Themenzahlen werden. Beziehungen zwischen Themen zeigen sich entweder durch hohe Vernetzungen im Graphen von A, oder durch Ausprägungen in D. Eine beinahe exakte Aufteilung wie sie in Abbildung 3.6 zu sehen ist, spricht für eine gute Wahl der Themenzahl. Wenn die Zahl der Cluster nicht der tatsächlichen Themenzahl entspricht ist davon auszugehen, dass es zu stärkeren Vermengungen in der Themen kommt. Für einen Nutzer ist D daher ein guter Indikator für die Bestimmung der Themenzahl.

Der Einstieg in die Analyse erfolgt über die Definition einer beliebigen Anzahl von Themen. Anschließend werden A und D betrachtet und die Themenzahl optimiert. Daraufhin kann mit der Erkundung einzelner Dokumente und Themen begonnen werden.

Sensemaking

Informationen lassen sich durch die farbliche Markierung über alle Ansichten und Dimensionen leicht verfolgen. In A und G können Thresholds festgelegt werden, die bestimmen welche Kanten angezeigt, beziehungsweise welche Wörter eingefärbt werden. Der Nutzer kann diesbezüglich den Detailgrad selbst bestimmen. Die Fäden in D sollen im besten Fall differenzierbare Stränge ergeben. Wird ein Dokument in A oder ein ganzes Thema in C ausgewählt, so wird dieses in D hervorgehoben. Abbildung 3.7 zeigt diese Funktion,

die von den Autoren *x-ray mode* genannt wird.

Navigationsschritte werden von Lee et al. in [43] nicht behandelt, daher ist nicht davon auszugehen, dass eine *Re-&Undo*-Funktion vorgesehen ist. Dem Nutzer wird kein Verlauf angeboten und auch keine Möglichkeit gegeben weitere Schritte zu planen, Stadien festzuhalten, oder eigene Konzepte zu externalisieren. Ansicht F ist eine Art einschrittiges Protokoll für die Änderungen, die durch die Modifikation des Topic Models entstehen. So zeigt Abbildung 3.6, dass durch die letzte Modifikation viele Dokumente die zuvor Thema 1 zugewiesen waren, nun zu Thema 3 gehören.

Es lassen sich keine Zustände festhalten, daher kann bei einer Veränderung der Topic Model-Parameter kein Vergleich mit der vorherigen Verteilung gemacht werden. Dokumente lassen sich in D vergleichen indem sie abwechseln selektiert werden, die Selektion mehrerer Dokumente wird nicht unterstützt. Ansicht D bietet die beste Möglichkeit um Elemente eines Datenmodells zu vergleichen. Der Parallel-Coordinate-Plot unterliegt auch bei der Veränderung des Datenmodelles nicht so starken Schwankungen wie ein force-directed Graph, weshalb er auch für den Vergleich zwischen mehreren Dantemodellen verwendet werden könnte. Entsprechend ist A weniger für den Vergleich von modifizierten Datenmodellen verwendbar.

Der Fokus von Lee et al. lag darauf, das Sensemaking durch die direkte Interaktion mit einem Topic Model zu unterstützen. Das Externalisieren von Informationen oder das Festhalten von Zuständen wurde daher nicht angestrebt. Dafür werden neue Methoden vorgestellt, die die Charakteristik der LDA nutzen, um direkte Manipulationen vorzunehmen. So kann ein Nutzer in Ansicht E die Gewichtung einzelner Terme eines Themas anpasst. Das modifizierte Topic Model wird dann wiederum auf alle Dokumente angewandt. Da das Topic Model bereits berechnet wurde, lässt es sich schnell auf die gleichen (oder auch auf andere) Dokumente anwenden. Dieser Vorgang wird *Inference* (dt. Schlussfolgerung) genannt[11]. Des Weiteren können Themen über Ansicht B unterteilt, zusammengeführt und hierarchisch gegliedert werden. Lee et al. gehen jedoch in ihrer Beschreibung weder auf die Umsetzung noch auf die Effizienz ein, daher können diese Absichten nicht weiter beachtet werden.

Zusammenfassung

Lee et al. beschreiben mehrere Funktionen zur direkten Modifikation eines Topic Models und gehen dabei über die reine Filterung der Resultate hinaus. Durch die bewusst gesetzte Beschränkung der Arbeit wurde das Sensemaking im Sinne von Pirolli et al. nicht verfolgt. Durchweg positiv an der Arbeit ist die Verknüpfung verschiedener Perspektiven, sodass ein einfacher Wechsel zwischen Detail und Trend ermöglicht wird.

[11] Eine genauere Beschreibung der Inference erfolgt in Abschnitt 4.4.

3.2.5 D-Vita

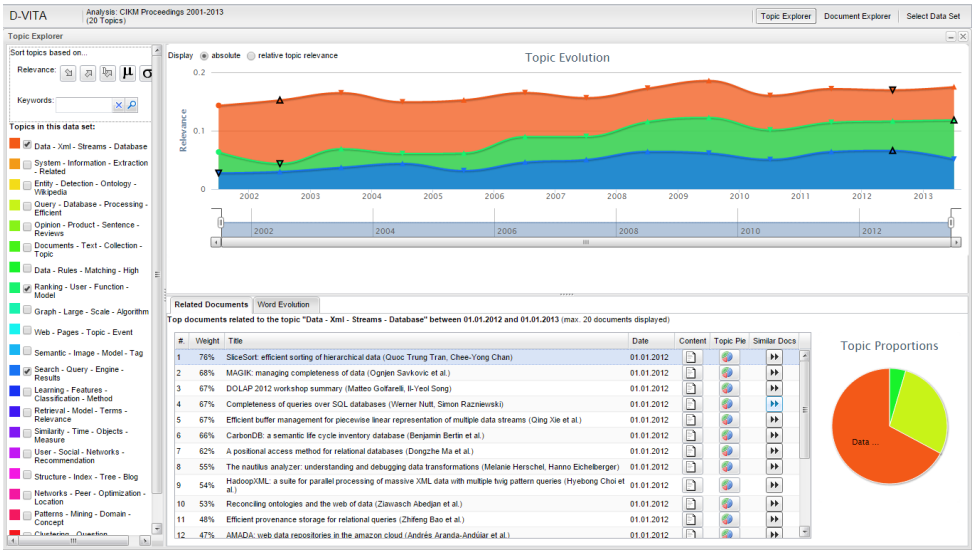


Abbildung 3.8: *D-Vita*: *Topic Explorer* mit Themen auf der linken Seite, dem zeitlichen Themenverlauf in der *Topic Evolution* und zum selektierten Thema passende Dokumente. *Quelle: Screenshot von http://merian.informatik.rwth-aachen.de:4444/DVita/ , 26.11.2014*

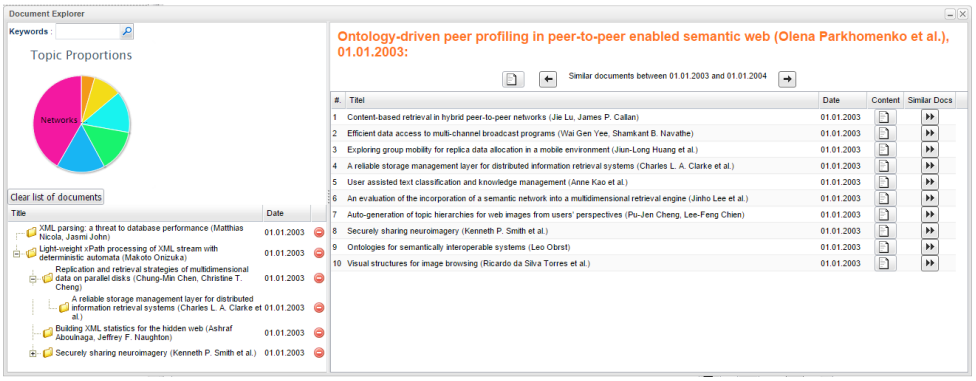


Abbildung 3.9: *D-Vita*: Dokumentgraph mit Themengewichtung, weiterführenden Dokumenten und auf der linken unteren Seite, dem Suchverlauf. *Quelle: Screenshot von http://merian.informatik.rwth-aachen.de:4444/DVita/ , 26.11.2014*

Günnemann et al. befassen sich bei *D-Vita* mit der visuellen Analyse von Resultaten der dynamischen LDA. Sie wollen die Extraktion von Wissen und das Verstehen zeitlicher Themenverläufe in unstrukturierten Textdaten unterstützen. Dazu entwickelten sie das Browser-Tool *D-Vita*. Es berechnet vorab ein Topic Model, das in dem Analyseprozess nicht mehr durch den Nutzer verändert werden kann [44]. Die folgenden Beobachtungen basieren auf der genannten Publikation und dem online frei zugänglichen Prototypen[12].

[12] `http://merian.informatik.rwth-aachen.de:4444/DVita/` , 26.11.2014

Da der Funktionsumfang von D-Vita begrenzt ist, wird auf eine Untergliederung dieses Abschnittes verzichtet.

Die Architektur der Anwendung wird von Günnemann et al., wohl in Anlehnung an das *Model-View-Controller*-Muster und das OSI-Modell, in *Data Layer, Application Layer* und *Presentation Layer* unterteilt. Die *Data Layer* besteht aus Datenbanken, die den Inhalt verschiedener Webressourcen und die Ergebnisse der darüber berechneten dynamischen LDA speichern. Die offline- und online-Komponenten die sich in einer Webapplikation zwangsläufig ergeben, werden in der *Application Layer* verbunden. Die Bezeichnung erfolgt aus der Perspektive des Servers, daher zählen zu den Offline-Komponenten die Vorverarbeitung der Quellen und die Anwendung der LDA. Darauf aufbauend werden die online-Komponenten angewandt. Diese bestehen aus einfachen Berechnungen der Dokument-, Themen- und Termstatistiken, die im Browser des Benutzers erfolgen und in der *Presentation Layer* visualisiert werden.

Abbildung 3.8 zeigt mit dem *Topic Explorer* den Hauptbestandteil von D-Vita. Auf der linken Seite sind die Themen aufgelistet, die selektiert und dann im nebenstehenden Stapelgraph dargestellt werden. Durch das Auswählen eines Themas im Graphen werden in der unteren Ansicht weiterführende Dokumente empfohlen. Abbildung 3.9 zeigt den *Document Explorer*, in dem die Themengewichtungen eines selektierten Dokuments anzeigt werden und weiterführende Dokumente aufgelistet sind. Über den Pfeil in der Spalte *Similar Docs* wird das ausgewählte Dokument im gleichen Fenster geladen. Die links-unten zu sehende Baumstruktur entspricht dem Vorgang der Exploration. Das Verlaufsprotokoll ist demnach nicht linear, sondern fügt neu begangene Dokumente in einer Stufe unter dem Dokument an, von dem aus es geöffnet wurde. Dadurch entsteht eine Ordnung die mehr über den Explorationsvorgang aussagt als eine chronologische Auflistung. Begangene Dokumente können zwar aus dem Protokoll gelöscht werden, jedoch lässt sich keine aktive Umsortierung vornehmen. Die Funktionen von D-Vita sind auf das Betrachten von Themengewichtungen in einem Dokument oder im zeitlichen Verlauf, und auf die Navigation über empfohlene Dokumente beschränkt. Der Topic Explorer und der Document Explorer sind bis auf das Verlaufsprotokoll redundant.

Die Anteile von Themen in einem Dokument werden als Piechart dargestellt. Durch das hoovern der Maus über einen der Teile werden die genauen Prozente des dazugehörigen Themas angezeigt. Die Relation zweier Dokumente wird nicht über die Kosinusdistanz hinausgehend angegeben und auch das Zustandekommen der Ähnlichkeiten wird nicht angezeigt. Alle Graphen werden animiert aufgebaut, was wegen der Flüchtigkeit visueller Reize für die Analyse unzweckmäßig ist. Veränderungen in der Visualisierung sollten stattdessen zu Gunsten der Vergleichbarkeit möglichst schnell aufgebaut werden [22]. Die Informationsdichte der Ansichten ist davon Abhängig wie viele Themen für die Anzei-

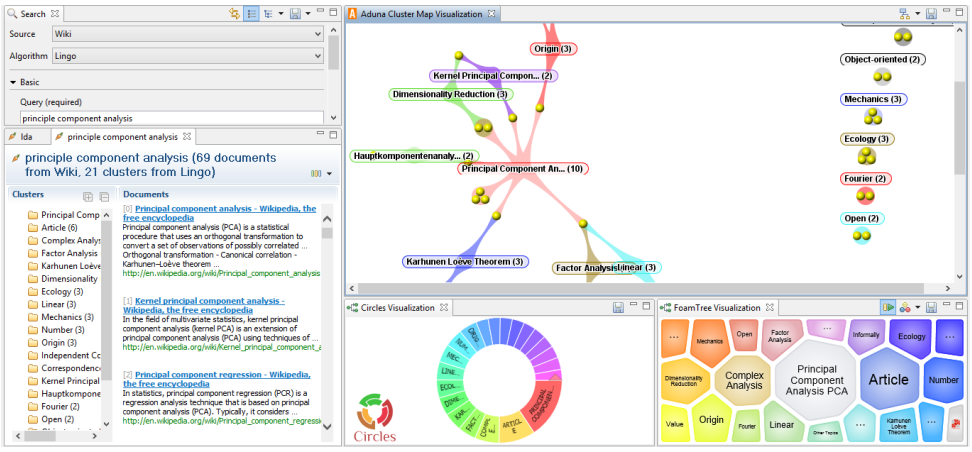


Abbildung 3.10: *Carrot*^{2}: Interface bestehend aus Suchanfragen, Clustern und Suchergebnissen(links), der *Aduna cluster map* (mitte) und Cluser-Visualisierungen (unten). Die gelben Punkte der Cluster Map repräsentieren einzelne Suchergebnisse, die örtlich und farblich in den Clustern gruppiert wurden. *Quelle: Screenshot der Carrot*^{2}*-Workbench , 01.12.2014*

ge selektiert wurden. Die Intervalle der Achsen der Graphen sind klar definiert, jedoch verändern sich diese Intervalle sobald die Selektion verändert wir. Der Vergleich von Darstellungen wird damit unnötig erschwert.

Zusammenfassung

D-Vita von Günnemann et al. ist ein webbasiertes Tool zur interaktiven Analyse von zeitlichen Themenverläufen. Durch die beschränkten Interaktions- und Modifikationsmöglichkeiten wurden die Einschränkungen von der Verwendung von Topic Models in Webanwendungen erkennbar. Ein positiver Aspekt von D-Vita ist automatische Organisation des Verlaufsprotokolls, wodurch die Orientierung im Explorationsprozess unterstützt wird.

3.2.6 Carrot2

Carrot^2 ist ein Open Source Framework für das Clustern von Suchergebnissen, für das ein Webinterface und eine *carrot*^{2}*-workbench* genannte Desktopanwendung entwickelt wurden. Für das Webinterface können verschiedene Suchmaschinen ausgewählt werden, unter Anderem *Microsoft Bing* und *Wikipedia*. *Carrot*^{2} kann auf beliebige Textdokumente angewandt werden, jedoch weisen die Autoren Weiss et al. darauf hin, dass es auf kurze Dokumente, wie etwa Suchergebnisse, ausgelegt und für Bücher oder mehrere tausend Dokumente nicht effizient genug ist. Auf die Suchumgebung kann online unter `http://carrot2.org/`[13] zugegriffen werden. Die Autoren Weiss et al. entwickelten einen

[13]Stand vom 01.12.2014

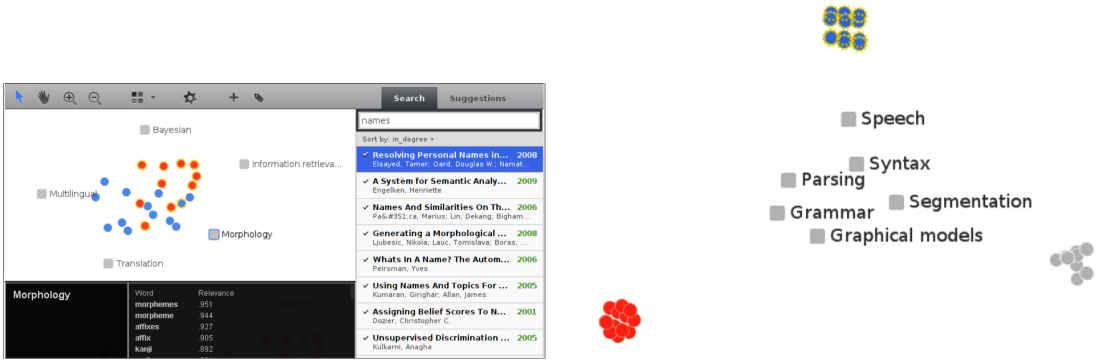
Cluster-Algorithmus, der auf der LSA basiert und daher zu den Topic Models gezählt werden kann. Carrot^2 wurde für den alltäglichen Gebrauch entwickelt und kann auf verschiedene Arten genutzt werden. Entweder mit einer einfachen Schlagwortsuche, oder in einer explorativen Suche, in der die Struktur der Daten betrachtet und die Parameter des Suchalgorithmus modifiziert werden. Dabei steht das IR und nicht die Analyse im Vordergrund. [45]

Interface

Die Desktopversion von Carrot^2 ist in Abbildung 3.10 zu sehen. Sie orientiert sich an dem Layout gängiger Dokumentexplorer. Auf der linken Seite werden Ordner aufgelistet, die den gefundenen Clustern entsprechen. Daneben werden die Inhalte des jeweils selektierten Clusters angezeigt. Dieser *Search*-Bereich wird um Visualisierungen der Cluster ergänzt. Die Elemente der *Circles Visualization* und der *Foam Tree Visualization* entsprechen im Volumen der Anzahl der ihnen zugeordneten Suchresultate. Es wird nicht auf die Hierarchie oder Relation der einzelnen Cluster geachtet, stattdessen werden diese einfach der Größe nach angeordnet. Eine Interaktion mit den Visualisierungen beschränkt sich auf die Selektion eines Themas, entsprechend redundant sind die Darstellungen. Interessanter ist die Ausgabe der Resultate als node-link Diagramm als *Aduna Cluster Map*, einem kommerziell von der Firma *Aduna* hergestellten *ready-to-use* Baustein.

Im Gegensatz zu den meisten Graphenvisualisierungen werden in der *Aduna Cluster Map* Elemente nach Themen gruppiert und Relationen farblich markiert. Durch die Gruppierung werden die einzelnen Kanten zwischen Elementen ausgelassen und die Informationslast somit reduziert. Die Gemeinsamkeit der Dokumente wird farblich gekennzeichnet und mit einem Titel versehen. Da es sich bei der Aduna Cluster Map um ein kommerzielles Produkt handelt, haben die Autoren Weiss et al. nicht erläutert wie der Graph generiert wird. Die Gruppen sind nicht verankert und können vom Nutzer beliebig verschoben werden, daher ist davon auszugehen, weshalb die Distanz der Knoten nicht zwangsläufig deren Ähnlichkeit entspricht. Durch das Selektieren eines Clusters werden alle dazugehörigen Gruppen und Dokumente hervorgehoben. Dieser visuelle Filter wird jedoch nicht auf die Suche übertragen.

Carrot^2 wurde ursprünglich für Informationswissenschaftler entworfen, mittlerweile haben Weiss et al. das Interface der Webanwendung auf die Benutzung durch Endnutzer zugeschnitten. In der Workbench gibt es noch sehr viel mehr Möglichkeiten den Cluster-Algorithmus zu parametrisieren. Dazu zählen die Menge der Cluster, die Berechnungsmethode der Themen-Bezeichnungen und die Gewichtung einzelner Terme. Diese Einstellungen erfolgen in dem separaten Arbeitsbereich *Tuning*. Um die Einflüsse der Parametrisierung zu erahnen ist ein erhebliches Wissen über die verwendeten Algorithmen



(a) Das Interface mit Sucheingabe, Detailansicht und Graphen mit verankerten Themen.

(b) Nach Autoren separierte und verankerte Dokumentcluster mit frei beweglichen Themen.

Abbildung 3.11: *TopicViz*: Interface des Programmes mit dust-and-magnet Graph. *Quelle: Eisenstein et al. 2012*

notwendig. Carrot^2 ist daher für verschiedene Suchformen und für Nutzer unterschiedlichen Wissens nutzbar. Eine schnelle Suche kann in klassischer Form als Keywordsearch durchgeführt werden und eine komplexere Analyse kann über die Analyse des Graphen und der Parametrisierung im Tuning-Bereich erfolgen.

Zusammenfassung

Carrot^2 ist auf kurze explorative Suchvorgänge ausgerichtet. Der Nutzer kann frei entscheiden ob er die Suchergebnisse direkt verwendet, sie in der Cluster Map analysiert, oder ob er die Parameter des Topic Models optimieren möchte. Darüber hinaus gibt es allerdings keine Interaktionsmöglichkeiten. Auch die Sinnstiftung wird nicht direkt unterstützt. Es ist dennoch interessant eine reduzierte und praktische Anwendung der explorativen Suche zu betrachten. Im Gegensatz zu klassischen Suchmaschinen werden dem Nutzer die Ergebnisse in Themen geordnet angeboten, sodass er schnell einen Überblick der enthaltenen und weiterführenden Themen machen kann. Vergangene Suchanfragen werden in einem Tab festgehalten. In Abbildung 3.10 ist etwa zu sehen, wie vom Suchbegriff *"lda"* zur *"principle component analysis"* gewechselt wurde. Nach der Erkundung des angrenzenden Themas kann die Suche im vorherigen Tab fortgesetzt werden.

3.2.7 TopicViz

Eisenstein et al. haben *TopicViz* mit dem Ziel entwickelt, die Sinnstiftung in der Exploration semantischer Strukturen in umfangreichen Textkollektionen zu unterstützten. Dazu werden Dokumente und Themen in einem interaktiven force-directed Graph dargestellt und mit einer Keyword-Suche kombiniert. Eisenstein et al. verwenden zwar ein Topic

Model, benennen allerdings nicht das spezifische Verfahren. Es existiert kein öffentlich zugänglicher Prototyp, daher basieren die folgenden Beobachtungen auf den Publikationen [46, 47] von Eisenstein et al.

Das Interface von TopicViz wie es in Abbildung 3.11 (a) zu sehen ist, scheint auf den ersten Blick sehr simpel. In einer Liste auf der rechten Seite werden Suchergebnisse wiedergegeben, in der Mitte befindet sich ein Graph mit ausgewählten Themen und Dokumenten und am unteren Rand werden Detailinformationen eines selektierten Elements angezeigt. Der force-directed Graph von TopicViz geht jedoch über die üblichen Interaktionsmöglichkeiten wie das Löschen, Erweitern und Verschieben einzelner Elemente hinaus. In Abbildung 3.11 sind mehrere Punkte und Quadrate zu sehen, die zum einen Dokumenten und zum anderen Themen entsprechen. Der Graph setzt auf dem *dust-and-magnet* Model von Yi et al. auf. Dabei üben die Themen (magnets) eine der jeweiligen Themengewichtung entsprechende, anziehende Kraft auf die Dokumente (dust) aus. Die Kraft wirkt allerdings nur wenn die Magneten bewegt werden [48]. Abbildung 3.11 (b) zeigt die Invertierung der Wirkungsweise. In diesem Fall sind die Dokumente fixiert (magnet) und die Themen orientieren sich an ihnen (dust). Die exemplarische Verteilung zeigt drei nach Autoren separierte Dokumentcluster mit dazwischen schwebenden Themen. Die Aufteilung ist noch nicht sehr stark ausgeprägt aber zeigt bereits, dass der blau markierte Autor verstärkt *Speech* und der rot markierte Autor eher die Themen *Grammar* und *Graphical models* behandelt hat.

In TopicViz ist es einem Nutzer möglich, die Aufteilung und Anordnung des Graphen selbst zu bestimmen. Mit jedem neuen Magneten wird eine neue Dimension aufgespannt, an der sich die Dust-Partikel orientieren. Dem Informationsverlust der durch die Projektion des hochdimensionalen Topic Models auf eine zweidimensionale Ebene erfolgt, wird damit partiell entgegen gewirkt. Die Charakteristiken der Themen und Dokumente können somit beliebig detailliert und in verschiedensten Kombinationen exploriert werden. [46, 47]

3.2.8 Tiara

Tiara ist eine von Wei et al. entwickelte Anwendung für die Analyse der zeitlichen Entwicklung von Themen in Textdokumenten [49]. Sie verwenden dazu eine dynamische Variante der LDA, wobei auch andere Clustering-Verfahren wie der *k-Means*-Algorithmus verwendet werden können. Wei et al. setzen sich in ihrer Arbeit hauptsächlich mit der Anwendung von Resultaten eines Topic Models auseinander und lassen dabei dessen Parametrisierung aus.

Abbildung 3.12 zeigt die Anwendung von Tiara auf mehr als 8000 E-Mails. Wei et al. vergleichen ihre Arbeit mit Carrot^2 (Abschnitt 3.2.6) und anderen auf Suchresulta-

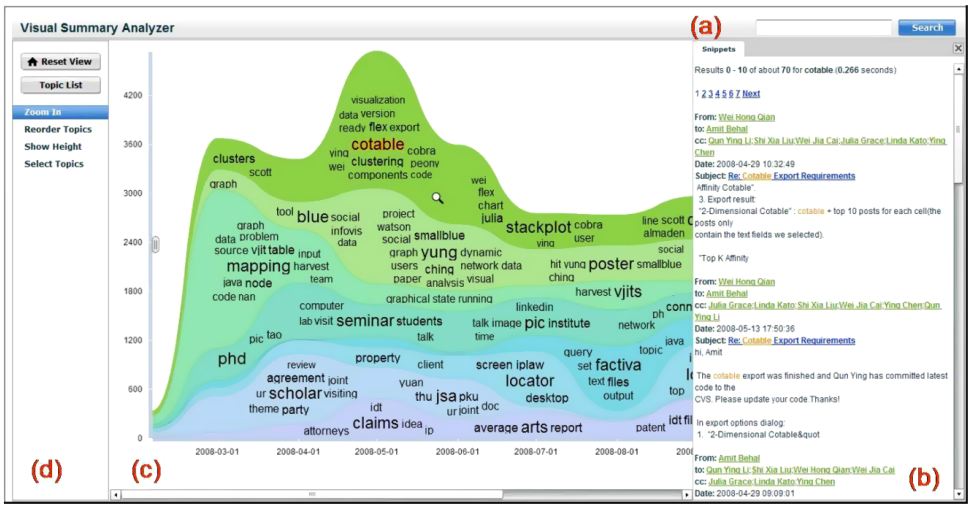


Abbildung 3.12: Interface von *Tiara* mit Visualisierung von zeitlichen Themenverläufen in mehr als 8000 E-Mails. Die Bestandteile sind (a) Sucheingabe, (b) Dokumentansicht, (c) Visualisierung der Relevanz einzelner Themen als *stacked graph* über einer Zeitachse und (d) Auswahl weiterer Darstellungsformen. *Quelle: Wei et al. 2010*

ten aufbauende Explorationsumgebungen, wobei der analytische Aspekt in Tiara stärker ausgeprägt ist als etwa bei Carrot[2].

Interface

Das Interface von Tiara ist auf Abbildung 3.12 zu sehen und besteht ausschließlich aus der Graphen-Ansicht *(c)* und der Liste relevanter Dokumente *(b)*. Der Suchprozess beginnt mit einer Volltextsuche im Bereich *(a)* über *Apache-Lucene*, durch die passende Dokumente identifiziert werden. Die Ergebnisse werden in *(b)* aufgelistet und in *(c)* visualisiert. Jeder Strang des Stacked-Graph wird durch Wörter des jeweiligen Themas beschrieben. Die Resultate können durch die Selektion von Metadaten in *(b)*, zum Beispiel einem speziellen Absender, oder durch die Auswahl eines Themas in *(c)* gefiltert werden. Die Volltextsuche kann auch mit Keywords der abgebildeten Terme verwendet werden, dazu muss der Nutzer lediglich auf einen der Terme klicken um eine Liste der passenden Dokumente zu erhalten.

Wei et al. verlassen sich bei der Relevanz der Terme nicht auf die *topic-term*-Matrix, sondern verwenden eine dem in Formel 5.1 beschriebenen *tf-idf-Maß* entlehnten Gewichtung. Bei der Berechnung von Topic Models kommt es häufig dazu, dass Terme in mehreren Themen hoch gewichtet werden. Ein Nutzer kann die Inhalte zweier Themen entsprechend schlecht identifizieren, wenn er die jeweils am höchsten gewichteten Wörter vergleicht und sich davon die meisten gleichen. Wei et al. normieren daher die Frequenz eines Terms in

einem Thema mit dessen Vorkommen in anderen Themen durch die folgende Formel:

$$weight(w_m) = \Phi_{j,m} * log \frac{\Phi_{j,m}}{(\prod_{k=1}^{K} \Phi_{k,m})^{\frac{1}{K}}} \tag{3.1}$$

Die stochastische Berechnung der LDA bedingt, dass die Term-Themen Gewichtungen zwar sehr klein, aber nicht 0 werden. Die Multiplikation der erwartungsgemäß kleinen Werte $\Phi_{k,m} < 1$ ergibt einen Wert $\prod_{k=1}^{K} \Phi_{k,m} \ll 1$. Dieser wird durch die Wurzelfunktion $\frac{1}{K}$ gedämpft, damit der anschließend berechnete Logarithmus keine zu großen Werte ergibt[14].

Sensemaking

Die Interaktion in Tiara erfolgt über den Stacked-Graph. Dort kann, neben dem Zoomen und Verschieben der Ansicht, direkt das unterliegende Datenmodel modifiziert werden. Wei et al. versuchen die Rechenlast dieser Modifikationen möglichst gering zu halten und nur die nötigsten Neuberechnungen vorzunehmen. Das Problem der falschen Wahl von latenten Themen führt bekanntermaßen zu verschmolzenen oder zu verteilten Themen. Um diese Fehler zu beheben können Themenstränge nachträglich zusammengefügt oder unterteilt werden. Für das Zusammenfügen von Themen muss lediglich der Themen-Index in den Dokumenten neu gesetzt werden muss. Soll beispielsweise t_5 und t_6 miteinander verschmolzen werden, so werden automatisch alle Dokumente ausgewählt, die entweder t_5 oder t_6 als höchst gewichtetes Thema besitzen und bekommen nun das Thema t_{K+1} zugewiesen. Für das Aufteilen eines Themas muss hingegen ein neues Topic Model berechnet werden. Dazu werden wiederum alle Dokumente des selektierten Themas ausgewählt und die LDA mit einer vom Nutzer definierten Zahl von Sub-Themen berechnet. Durch die kleine Dokument- und Themenzahl fällt dieser Vorgang weitaus kürzer aus, als die erste und alle Dokumente umfassende Berechnung. Damit werden in Tiara effiziente Interaktionsmöglichkeiten in der Themenanalysen und explorative Suchen geboten. Die Unterstützung der Externalisierung ist von den Autoren nicht vorgesehen.

3.2.9 Hiérarchie

Smith et al. haben *Hiérarchie* mit dem Ziel entwickelt Topic Model für Endnutzer zugänglich zu machen. Ihrer Meinung nach ist die Interpretation von Topic Models in vielen Anwendungen fast genauso Zeitaufwendig wie das Lesen der einzelnen Dokumente. Dem wollen sie durch eine Vereinfachung der Resultate entgegenwirken. Mit Hiérarchie stellen Smith et al. eine hierarchische Variation der LDA vor, dessen Resultate in einer *Sunburst-*

[14] Für ein besseres Verständnis können die Funktionen $log\frac{1}{x}$ und $x^{\frac{1}{y}}$ betrachtet werden.

Chart visualisiert werden. Sie sehen eine Beschränkung in bestehenden Visualisierungen von Topic Models, die für nicht mehr als 20 Themen geeignet seien und sich nicht auf umfangreiche Dokumentkorpora mit einer Vielzahl von Themen skalieren ließen. Durch die Hierarchisierung und die Visualisierung in der Sunburst-Chart, wollen sie das Topic Model in eine neue Struktur fügen, die sich auch für umfangreiche Textkorpora effizient explorieren lässt.

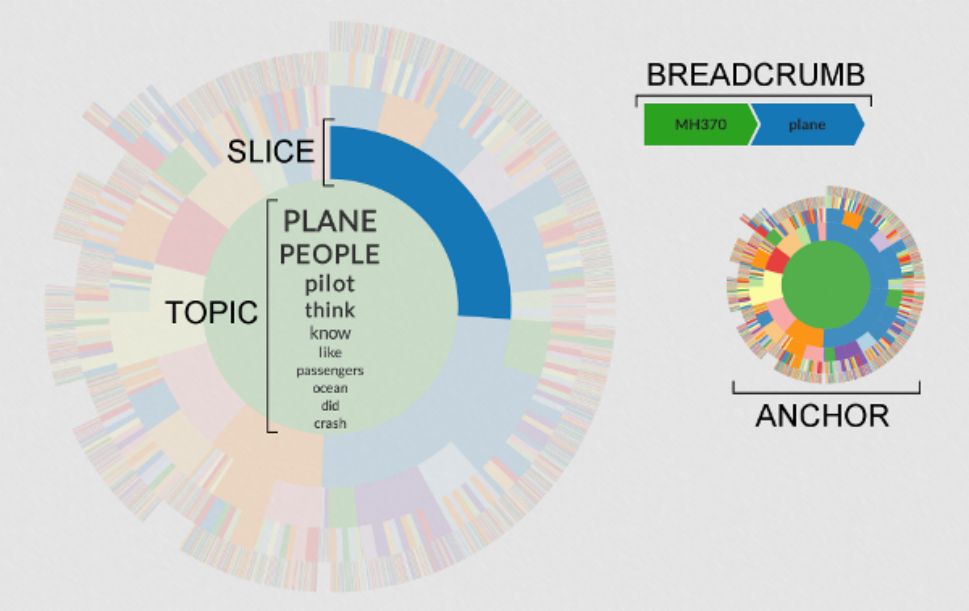


Abbildung 3.13: *hiérarchie*: Interface mit Beschreibung der einzelnen Elemente. *Quelle: https://github.com/mlvl/hierarchie , 29.11.2014*

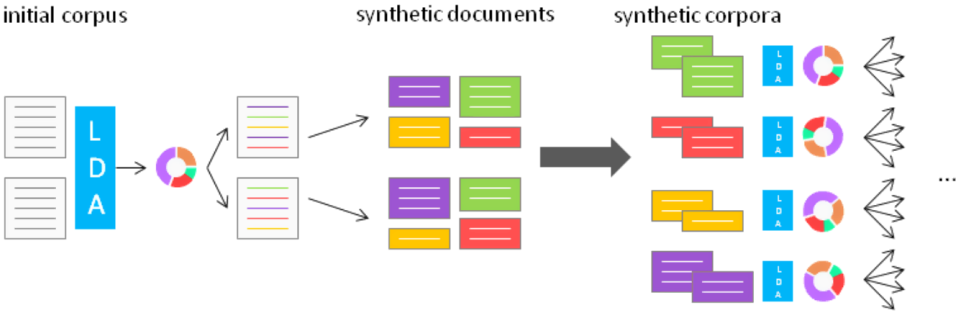


Abbildung 3.14: *hiérarchie*: Identifikation thematischer Texteinheiten in Dokumenten und erzeugung synthetischer Dokumentkorpora. *Quelle: Smith et al, 2014*

Bei Hiérarchie handelt es sich nicht um eine vollständige Explorationsumgebung, sondern um ein Konzept der hierarchischen Themenexploration [50]. Die Evaluation erfolgt auf der Arbeit [50] und dem frei zugänglichen Prototypen[15].

[15] `http://mlvl.github.io/hierarchie/` , 29.11.2014

Abbildung 3.14 zeigt die wiederholte Anwendung der LDA. Den Vorgang bezeichnen Smith et al. als hierarchische LDA, die aus einem Dokumentkorpus für jede hierarchieebene einen neuen, synthetischen Textkorpus aus den vorherigen Ergebnissen generiert. Der Algorithmus arbeitet sich somit rekursiv in die Tiefe und erzeugt schrittweise höhere Detailstufen. Jeder Schritt umfasst die folgenden Aktionen:

1. Anwenden der LDA auf den (synthetischen) Dokumentkorpus um K latente Themen zu identifizieren.
2. Jedem Dokument das jeweils höchste Thema zuweisen
3. Generieren von K neuen synthetischen Dokumentkorpora, die sich aus der vorherigen Zuweisung ergeben

Da Hiérarchie eine Webanwendung ist und die wiederholte Berechnung der LDA viel Zeit beansprucht, werden alle Kalkulationen vor der Präsentation durchgeführt. Die Ergebnisse werden im *JSON*-Format gespeichert und mit der JavaScript-Bibliothek *D3* visualisiert.

Interface

Abbildung 3.13 zeigt die vier Elemente des Interfaces von *hiérarchie*. Die Navigation erfolgt über den mehrschichtigen Sunburst-Chart, von dem jeder *Slice* einem Thema entspricht. Alle sich davon nach außen hin verzweigenden Slices sind Sub-Themen. Durch das Auswählen eines äußeren Themas wird der innere Kreis ausgeblendet und das äußere durch die Subthemen des selektierten Themas ersetzt, sodass diese die gesamten 360 Grad nutzen. Durch das hoovern des Mauszeigers über eines der Slices erscheinen die dazugehörigen Terme in der Mitte der Sunburst-Chart. Die Elemente *Breadcrumb* und *Anchor* dienen der Orientierung.

Sensemaking

Die *Breadcrumbs* helfen dabei seine Position zu bestimmen und werden anhand der am höchsten gewichteten Terme der übergeordneten Themen benannt. Der *Anchor* hebt die absolute Position in der Sunburst-Chart hervor. Mit der Kombination von *Breadcrumbs* und *Anchor* gibt es eine visuelle und textuelle Unterstützung der Orientierung. Problematisch ist, dass der für das selektiert Thema am höchsten gewichtete Term als Label verwendet wird. Es ist sehr wahrscheinlich, dass wichtige Terme des Oberthemas auch für die Unterthemen relevant sind und es somit zu Mehrfachbezeichnungen kommt. Etwa im Beispiel von Abbildung 3.14 sind in tieferen Navigationsstufen stellenweise alle Breadcrumbs mit *plane* bezeichnet, da als Quellen Blogeinträge über das Verschwinden des

Fluges *Mh-370* verwendet wurden. Außerdem sind die Themenbezeichnungen sehr allgemein, da etwa das Thema *Plane, Moment, crashing, wreckage, water, ...* einfach auf *plane* reduziert, einem Begriff der in der sich im gegebenen Beispiel über den ganzen Datensatz erstreckt und entsprechend wenig Information trägt.

Slices die durch das gleiche Wort beschrieben werden, haben die gleiche Farbe. Im Beispiel haben alle Themen mit der Bezeichnung *plane* eine hell-grüne Färbung. Problematisch ist, dass Farben mehrfach zugewiesen werden, ohne dass sie einen thematischen Zusammenhang aufweisen. Daraus ergibt sich ein Farbschema das schön anzusehen ist, im Sinne der Navigation jedoch eher verwirrt.

3.3 Zusammenführung

Der Stand und die Zielsetzung der in Abschnitt 3.2 betrachteten Arbeiten, unterscheiden sich teilweise deutlich voneinander. So sind Hiérarchie und TopicViz eher von konzeptioneller Natur, während Qiqqa, Jigsaw und Serendip bereits nahezu fertige Anwendungen sind. Die Zielsetzung von Jigsaw, Serendip und weiteren Anwendungen lag auf der Analyse von Dokumentkorpora, wogegen Carrot^2 und TopicViz auf das zügige Auffinden einzelner Dokumente ausgerichtet sind.

Auch wenn das Sensemaking, im Sinne der Definition Pirolli et al., überraschend wenig unterstützt wird, konnten interessante neue Funktionen für den Umgang mit Topic Models herausgestellt werden. In den folgenden Abschnitten werden die zuvor einzeln behandelten Arbeiten anhand der Algorithmen, des Interfaces und des Sensemaking zusammengeführt. Tabelle 3.1 gibt einen Überblick der wichtigsten Attribute, die in mindestens einer der Anwendungen umgesetzt wurden.

3.3.1 Algorithmen

Alle Tools lassen sich auf Textdokumente beliebigen Formates anwenden und lediglich durch mögliche Funktionseinbußen und sinkende Performanz werden die Auswahl der Eingabedaten beschränkt. Verwendet werden LDA, LSA, k-Means Clustering, sowie dynamische Variationen der LDA und durch die rekursive Anwendung der LDA auch hierarchische Topic Model. Lediglich Jigsaw wurde nicht für die direkte Anwendung eines Topic Models ausgelegt. Dort werden vorab definierte Entitäten gesucht, auf die wiederum die SVD angewandt wird. Es lässt sich allerdings auch direkt die LSA anwenden, wodurch jedoch die Typ-Informationen verloren gehen, auf denen viele Funktionen der Anwendung beruhen. Carrot^2 und Jigsaw wurden für die Anwendung auf kurze Dokumente entwickelt, was sich durch die verwendete *SVD* bedingt, die für eine hohe Dimensionalität sehr rechenintensiv ist. Beide Anwendungen verwenden neben statistischen Verfahren

auch sprachtypische Informationen, wodurch sie auf zuvor erstellte Filter beschränkt werden. Qiqqa, Serendip und Jigsaw verwenden Metadaten wie Autoren, Zeitangaben und Titel von Dokumenten, die für die weitere Filterung verwendet werden. Dazu ist eine Einschränkung der Anwendung auf ein bestimmtes Datenformat nötig. Qiqqa ist auf die Verwaltung von wissenschaftlichen Publikationen beschränkt, was die spezielle Funktion ermöglichte, Metadaten automatisch von Google-Scholar zu beziehen. D-Vita und Tiara verwenden die dynamische LDA, die den chronologischen Verlauf von Themen erfasst. Dazu werden keine Metadaten benötigt, jedoch müssen die Eingangsdaten vorab in Zeiteinheiten aufgeteilt werden. Die dynamische LDA wurde mit der Absicht der zeitlichen Analyse entwickelt, aber lässt sich ebenso auf beliebige andere lineare Abfolgen anwenden. Alexander et al., die Autoren von Serendip, haben in ihrer Arbeit [37] nicht erläutert wie der Themenverlaufs-Graph im Serendip-*TextViewer* berechnet wurde. Ein solcher inhaltlicher Verlauf in einem Dokument ließe sich ebenfalls mit einem dynamischen Topic Model bestimmen.

Um während der Anwendung eine störungsfreie Interaktion zu gewährleisten, berechnen D-Vita, Serendip und Hiérarchie das Topic Model vorab und beschränken sich während der Anwendung auf einfache Kalkulationen und Filterungen über den resultierenden *topic-document* und *term-topic* Matrizen. In allen drei Fällen handelt es sich um Webapplikationen, die ihre Präsentationsebene strikt von der Daten- und Applikationsebene trennen. Dass das nicht notwendig ist, zeigt die effiziente Umsetzung von Carrot^2. Ebenso wie in Tiara und iVisClustering werden dort die Berechnungen zur Laufzeit durchgeführt und ermöglichen eine direkte Variation der Eingabedaten und Parameter. Das Detailniveau kann vom Nutzer selbst bestimmt werden, wodurch der Freiheitsgrad aber auch die Komplexität steigt. In Qiqqa, Tiara und Jigsaw können die Topic Models beziehungsweise die Cluster, beliebig neu berechnet werden. Eine Berechnung von wenigen Dokumenten und Themen mit sinnvollen Ergebnissen, also einer ausreichend hohen Iterationszahl, benötigt auf einem leistungsfähigen Desktop-Computer immerhin mehrere Minuten. Eine praktische Anwendung ist daher nur möglich wenn entweder wenige Dokumente verwendet werden, oder die Berechnungen auf leistungsstarke Server ausgelagert werden. Der alternative Ansatz von Hiérarchie, möglichst viele Möglichkeiten vorab zu berechnen, kostet sehr viel Zeit und erzeugt eine Struktur, die in dieser Form vielleicht überhaupt nicht benötigt wird.

3.3.2 Interface

Die Leitfragen der Spezifikation werden nun verwendet um die Arbeiten in den einzelnen Aspekten zusammenzuführen. Chaney et al. beschrieben einige grundlegende Funktionen für Anwendungen der explorativen Suche über Topic Models, die in vielen Anwendungen

noch immer die Kernbestandteile bilden [4]:

- Die Auflistung von Termen entsprechend ihrer Relevanz zu einem Dokument.
- Die Auflistung von Themen entsprechend ihrer Relevanz und das Anzeigen des Themengewichtes.
- Das Anzeigen von Dokumenten die einem betrachteten Dokument ähnlich sind.
- Das direkte Betrachten eines Dokumentes und das Anzeigen der Themengewichtungen des Dokumentes

Diese Funktionen wurden mittlerweile erweitert und differenziert. Im Folgenden wird beschrieben, wie darüber hinaus mit den verschiedenen Aspekten der Visualisierung und Interaktion umgegangen wird.

Aufteilung der Ansichten

Es lassen sich verschiedene Herangehensweise für die Aufteilung der Ansichten erkennen. Es wird entweder nach Funktionen in der Interaktion, nach dem Abstraktionsniveau der Betrachtung oder den Datentypen unterschieden.

Die Aufteilung nach Funktionen führt zu Ansichten, die die gleichen Daten auf unterschiedliche Weise darstellen, um Interaktionen in verschiedenen Kontexten zu ermöglichen. In Qiqqa etwa werden Relationen einmal als Graph und einmal in Form von Gruppierungen dargestellt. Damit können zum einen Verbindungen genauer betrachtet werden und zum anderen Gemeinsamkeiten herausgestellt werden. Serendip hingegen separiert die Arbeitsbereiche nach Typen (Dokument, Thema, Term). Jede Ansicht hat einen eigenen Schwerpunkt, wobei versucht wird, die Verbindung zu anderen Typen aufrecht zu erhalten. So wird im Serendip-*Corpus Viewer* der Fokus auf die Themenverteilung gelegt, aber auch die Verknüpfung zu der jeweiligen Termgewichtung hergestellt.

Obwohl eine Trennung in mehrere Arbeitsbereiche stattfindet, werden viele Visualisierungs- und Interaktions-Artefakte mehrfach verwendet. Solch eine Redundanz wird in D-Vita sehr deutlich, in dem sich zwei Arbeitsbereiche lediglich durch die Verwendung eines Navigationsprotokolls unterscheiden und ansonsten funktional völlig gleichen. Dieses wiederkehrende Problem löst sich wohl erst, wenn neue Interfaces nicht mehr in Anlehnung an die Entwürfe von Channey et al. [4] konzipiert werden. Stattdessen sollte eine Modularisierte und Werkzeug-orientierte Struktur verwendet werden, wie sie sich für Editor-Umgebungen etabliert hat.

Ansichten sind entweder durch Aktionen oder durch Hervorhebungen miteinander verbunden. In *Jigsaw* und *Hiérarchie* werden temporäre Selektionen global hervorgehoben, während in *Serendip* Themen bis zur Aufhebung farblich markiert bleiben. *Qiqqa,*

iVisClustering und *D-Vita* verbinden die Ansichten durch Navigationsketten, die den Nutzer von einer abstrahierenden Übersicht zum Detail, hin zu einem schlussendlichen Dokument führen.

Umgang mit Abstraktionsebenen und Dimensionen

Informationen können entweder in ihrer berechneten Form exakt dargestellt, oder zu Gunsten der Informationslast abstrahiert werden. In *Serendip* werden alle Themengewichtungen direkt als Matrix und als Verlauf in der Dokumentansicht dargestellt. Lediglich Terme werden einzelnen Themen zugewiesen. *Qiqqa* ist im Brainstorm-Bereich ebenfalls sehr detailliert, indem für jedes Dokument die genauen Themenanteile gezeigt werden. Dagegen erfolgt in *iVisClustering* und *D-Vita* eine Vereinfachung, da ein Dokument nur durch das jeweils relevanteste Thema beschrieben wird.

In der Darstellung als node-link Graph erfolgt durch die Projektion der K-dimensionalen Dokumentvektoren[16] auf eine zweidimensionale Ansicht zwangsläufig ein Informationsverlust. *TopicViz* begegnet diesem Problem mit dem *dust-and-magnet* Verfahren, durch das die Charakteristiken der einzelnen Dimensionen spielerisch erfahren werden können. *iVisClustering* ergänzt den node-link Graph durch einen parallel-Plot, in dem Trends einzelner Themen erfasst werden können, die sich nur schwer durch die Reduktion auf zwei Dimensionen darstellen lassen.

Darstellung von Relationen

Eine häufige Form der Darstellung von Relationen ist die Verwendung von geordneten Listen. Eine Liste lässt sich nur nach *einem* Vergleichsfaktor ordnen, woraus sich das Problem ergibt, dass Dokumentrelationen nur durch eine generelle Ähnlichkeit beschrieben werden. Die genaue Charakteristik einer Relation lässt sich daran nicht erkennen.

Für Themen-Term Relationen wird in der Regel die lokale Gewichtung verwendet, die der Vorkommenshäufigkeit eines Wortes in einem Thema entspricht. Diese sollte aber nicht als Relevanz interpretiert werden, da ein lokal häufig vorkommendes Wort ebenso häufig in anderen Themen vorkommen kann. Für Dokument-Themen Relationen gilt das gleiche. Junktopics die sich über den gesamten Dokumentkorpus ausbreiten, haben in der Regel eine geringe Aussagekraft für ein spezielles Dokument. Für beide Relationsformen sollte daher ein Umdenken stattfinden, dass nicht schlichtweg auf die Resultate des Topic Models vertraut. Dokument-Dokument Relationen werden durch das Kosinus-Maß beschrieben, das sich als gutes Maß der semantischen Ähnlichkeit ergeben hat. Die direkte Verwendung lässt allerdings die Präferenzen des Nutzers unbeachtet. In Qiqqa und D-Vita

[16] K ist die Anzahl der Themen und definiert damit die Gewichtsvektoren der Dokumente.

ist es etwa nicht möglich die Form einer Relation zu bestimmen, sodass auf die Vorschläge des Programmes vertraut werden muss, oder die vorgeschlagenen Dokumente einzeln betrachtet werden müssen. Praktischer wäre es bestimmte Relationen zu Filtern oder hervorzuheben. In *Serendip* wird dieses Problem mit der Matrix-Ansicht gelöst, in der alle Details direkt ablesbar sind und die Ordnung der Themen- und der Dokument-Achse nach verschiedenen Attributen erfolgen kann.

Eine nichtlineare Darstellung von Elementrelationen bieten node-link Graphen. Sie erfahren durch die Projektion zwar ebenfalls einen Informationsverlust, aber behalten dabei die Nachbarschaftsbeziehung zwischen Elementen weitgehend bei und geben daher die Charakteristik eines Datensatzes besser wieder als es mit einer Liste möglich ist. Die visuellen Attribute der Knoten und Kanten können verschiedene Parameter encodieren. Node-link Graphen werden auch als Karten bezeichnet, da sie unter Umständen hohe Informationsdichten enthalten, die im zweidimensionalen Raum erkundet werden können. Carrot^2, iVisClustering und Qiqqa verwenden solche Graphen, wobei die Möglichkeiten der Visualisierung und der Interaktion noch nicht vollends genutzt werden. Beispielsweise wird in keiner der Anwendungen die Stärke der Verbindungen angezeigt, was sich sehr leicht umsetzen ließe.

Die Verwendung von interaktiven force-directed Graphen ermöglicht es dem Nutzer, Knoten frei zu positionieren und eine individuelle visuelle Struktur zu formen. Die Vergleichbarkeit mehrerer Graphen leidet unter dieser Beliebigkeit natürlich. In Qiqqa wird eine self-organizing Map verwendet die sich automatisch Veränderungen anpasst. Wenn ein neuer Knoten geöffnet wird streben alle Elemente auseinander, um Verdeckungen zu vermeiden. Dadurch schwebt der Graph in einem ständigen Optimierungszustand über die Ansicht. Das erschwert nicht nur die Vergleichbarkeit zwischen (sich geringfügig unterscheidenden) Visualisierungs-Stadien, sondern belastet darüber hinaus die Konzentration des Nutzers. Für eine Übersicht von Publikationen die sich mit der Visualisierung von Relationen in Form von Graphen beschäftigen, sei noch einmal auf die Arbeit von Dadzie et al. [36] verwiesen, deren Prinzipien in Abschnitt 2.3.2 behandelt wurden.

Bezeichnung von Themen

Die Themen eines Topic Model ergeben sich aus den Gewichtungen einzelner Wörter. Es stehen keine anderen Informationen zur Verfügung, um ein Thema sprachlich zu beschreiben. *Carrot^2* und *TopicViz* verwenden jeweils den wichtigsten Term eines Themas als Bezeichner. Ohne spezielle Selektionsregeln kann es zu Doppelungen kommen, wie es an den *Breadcrumbs* von *Hiérarchie* zu sehen war. Ein einzelner Term ist schwer zu interpretieren und kann nicht ein ganzes Thema beschreiben. Um eine breitere Bedeutung zu schaffen werden daher in *Qiqqa, Jigsaw, iVisClustering, Hiérarchie* und *Carrot^2* die

wichtigsten n Terme eines Themas verwendet. In Abschnitt 3.2.3 wurde beschrieben, dass es auch trotz der Verwendung mehrerer Terme zu der mehrfachen Verwendung von Bezeichnern kommen kann. Mit Formel 3.1 wurde für Tiara eine Gewichtung entwickelt, die das lokale Gewicht eines Terms in Relation zu dessen globalen Gewicht setzt und damit die charakteristischsten Wörter eines Themas findet. Die Autoren von *Serendip* umgehen dieses Problem komplett, indem sie eine einfache Nummerierung verwenden, die der Nutzer manuell anpassen muss. Im Hinblick auf die Beschränkung des Kurzzeitgedächtnisses auf 3 ± 1 Informationseinheiten [22] erschweren längere Bezeichner die Identifikation von Themen. Somit muss zwischen Aussagekraft und Informationslast abgewogen werden. In allen fünf der genannten Anwendungen wird jedes Thema zusätzlich durch eine Farbe gekennzeichnet.

Einstiegspunkte

Dadzie et al. [36] sehen, vor allem für ungeübte Nutzer, einen visuellen Einstiegspunkt als einfachste Möglichkeit sich der Datenanalyse zu nähern. In *Qiqqa, Carrot*^{2}*, TopicViz* und *Tiara* erfolgt der Einstieg wahlweise über eine Volltextsuche, oder die Suche nach Schlagworten. *Qiqqa* bietet des Weiteren die Filterung nach Metadaten und Themen. Die restlichen Anwendungen beginnen die Exploration mit der Selektion und Analyse von Themen. Etwa in *Serendip* wird der Nutzer direkt mit der detaillierten Relationsmatrix von Dokumenten und Themen konfrontiert. Die Menge an Informationen erschwert das Identifizieren von Charakteristiken deutlich. Durch die häufige Umsortierung der Matrix gehen Bezugspunkte schnell verloren, sodass es unter Umständen einfacher ist sich direkt ein interessantes Dokument herauszusuchen, und von dort aus weiter vorzugehen. Der Einstieg in *iVisClustering* kann entweder über den Relationsgraphen oder den Parallel-Plot erfolgen. Der Parallel-Plot ist Lee et al. zufolge ein gutes Maß für die Exaktheit des Topic Models [43]. Der erste Schritt wäre daher das optimieren des Topic Models und einer anschließenden Dokumentbetrachtung. *Carrot*^{2} ermöglicht dem Nutzer verschiedene Komplexitätsstufen der Suche zu wählen, abhängig von seinen eigenen Fähigkeiten und der zu erfüllenden Aufgabe. Damit wird das Prinzip der *Support for querying* erfüllt.

Informationslast

Eine hohe Zahl angezeigter Informationen erlaubt dem Nutzer eine vielseitige und freie Arbeitsweise, doch je weniger Informationen angezeigt werden, desto effektiver lässt sich damit arbeiten. Die goldene Mitte zu finden hängt sehr stark von den Zielen und Fähigkeiten des Nutzers ab und sollte von diesem selbst bestimmt werden. Einige für Topic Models typische Variablen, die die Informationslast eines Interfaces bestimmen:

- Die Anzahl der Terme, durch die ein Thema bezeichnet wird.
- Der Detailgrad mit dem die Relationen von Dokumenten beschrieben werden. Dazu zählen die Kanten in einem Graphen, die Menge von vorgeschlagenen Dokumente und Zahl der Themen durch die eine Relation beschrieben wird.
- Der Detailgrad mit dem das Gewicht eines Themas beschrieben wird. Das kann sich auf ein ganzes Dokument beschränken, oder bis auf Teile des Dokumentes ausgeführt werden.

Durch die stochastische Bestimmung eines Topic Models besteht zwischen fast allen Dokumenten und Themen eine gewisse Relation. Viele davon sind von geringer Bedeutung und sollten als *Grundrauschen* in Darstellungen gefiltert werden. In der Matrix und der Dokumentansicht von *Serendip* und den Termen von *Tiara* wird diese Methode verwendet. Es bleiben dennoch viele Elemente übrig, die einen ungeübten Nutzer verwirren können. Dem wird durch visuelle Gruppierung, etwa durch Färbung, versucht zu begegnen. Ein hoher Detailgrad fordert eine längere Auseinandersetzung mit der Anwendung und ist daher für die kontinuierliche Nutzung oder für Spezialisten geeignet. In *iVisClustering* kann der Nutzer den minimal notwendigen Wert zur Anzeige einer Relation frei bestimmen. Das Prinzip wird unabhängig in mehreren Ansichten angewandt, sodass die Visualisierungen aller Daten separat behandelt werden können. In *Jigsaw* ist das temporäre Ausblenden nach Typen von Entitäten möglich. In der Anzeige von Dokumentrelationen in *Qiqqa* und *D-Vita* werden Detailinformationen abgeschnitten und damit das Prinzip von *detail-on-demand* nicht erfüllt [36]. Details sollten nicht vollständig eliminiert, sondern ausschließlich temporär ausgeblendet werden.

3.3.3 Sensemaking

Im Folgenden wird zusammengefasst wie in den Anwendungen der Prozess des Sensemaking unterstützt wurde. Dazu zählt neben der Externalisierung von Wissen die Orientierung im Informationsraum und das Nachvollziehen des eigenen Explorationshergangs, wodurch es einem Nutzer ermöglicht wird sein eigenes Suchverhalten zu reflektieren. Im Sinne der Evaluation von Hypothesen wird auch der Umgang mit der Beurteilung und Vergleichbarkeit von Daten hervorgehoben.

Orientierung

Die *Datenübersicht* kann über abstrakte und zusammenfassende Perspektiven unterstützt werden und ist besonders für Detailanalysen wichtig. *Hiérarchie* verwendet daher neben dem Interaktionsbereich, die statische *Anker*-Ansicht, auf der die Position des Nutzers

in der hierarchischen Schichtung der Daten angezeigt wird. Selektionen in *iVisClustering* sowie in *Jigsaw* werden in allen Ansichten der Programme übernommen. Die Selektion einer Entität oder eines Dokumentes im Detailbereich führt zum Highlighten im Graphen und in der Cluster-Ansicht. Dadurch lassen sich Details leicht in das gesamte Konzept einordnen. In *Jigsaw* ist dieses Prinzip besonders wichtig, da mit sehr vielen Ansichten parallel gearbeitet wird.

Die Dokumentation und Umkehrbarkeit von Aktionen wird von den betrachteten Anwendungen nur äußerst spärlich unterstützt. Ohne eine *un- & redo*-Funktion muss jeder Schritt gut überlegt sein, da er unter Umständen zu einer spontanen Informationsflut führt. Die Brainstorm-Ansicht von *Qiqqa* weist genau dieses Problem auf. *Jigsaw* ist auf eine umfassende Protokollierung von Aktionen ausgerichtet und ermöglicht überdies das Festhalten expliziter Zustände im *Tablet*. Es ist in der Protokollierung zwischen Aktionen der Darstellung, etwa dem Ausklappen von Knoten oder definieren eines Filterparameters, und denen der Datenmodifikation, also dem Löschen von Dokumenten oder der erneuten Berechnung des Topic Models, zu unterscheiden. Beide werden in *Jigsaw* separat unterstützt. In *D-Vita* wird die Betrachtungsreihenfolge von Dokumenten automatisch in einer Baumstruktur festgehalten. Die Protokollierung der Exploration verliert damit ihre Linearität und erlaubt die Erkundung paralleler Suchpfade. Simple Formen der Orientierung bieten *Hiérarchie* und *Carrot*^{2}. Die Übersicht von *Hiérarchie* in Form der *Breadcrumbs* ist ausschließlich auf hierarchische Strukturen anwendbar, während *Carrot*^{2} den Suchverlauf in einzelnen Tabs festhält.

Datenvergleich

Um entscheiden zu können welche Informationen relevant sind und um Urteile fällen zu können, müssen sich Daten vergleichen lassen. In *Qiqqa* und *Jigsaw* lassen sich Ansichten nebeneinander stellen, sodass parallel gearbeitet und visuell verglichen werden kann. Graphen der Themen- und Dokumentrelationen entstehen immer als Projektion und unter Dimensionsreduktion, folglich sind viele verschiedene Anordnungen des Graphen möglich und entsprechen damit nicht dem Prinzip der *Wiederverwendbarkeit*. Eine manuelle und rein visuelle Gegenüberstellung ist daher aufwändig oder gar ergebnislos. In *Jigsaw* ist ein visueller Vergleich möglich, wenn eine Ansicht dupliziert wird, sodass in einem kleinen Radius parallel in verschiedene Richtungen vorgegangen werden kann. Wenn sich keine Ansichten nebeneinanderstellen lassen, ist die einzige Vergleichsmöglichkeit das abwechselnde Selektieren verschiedener Elemente. Im Parallel-Plot von *iVisClustering* könnten damit etwa die Themenvektoren zweier Dokumente abwechselnd hervorgehoben werden. In *Serendip* hingegen kann die Matrix beliebig umgeordnet werden, sodass alle Themen und Dokumente direkt nebeneinanderstellt werden können. Ein direkter Vergleich meh-

rerer Elemente ist jedoch nicht möglich. Über das Kosinus-Maß hinausgehend werden in keiner Anwendung Vergleiche unterstützt.

Durch die häufige Verwendung von node-link Graphen, bieten sich entsprechende visuelle Vergleichsmethoden an, wurden in den genannten Arbeiten jedoch noch nicht in Betracht gezogen. Abbildung 3.15 ist der Survey von Landesberger et al. entnommen [35] und zeigt die Vereinigung zweier Graphen in der *Difference Map*, wodurch sich Gemeinsamkeiten und Unterschiede visuell schnell erfassen lassen. In Abschnitt 2.3.2 wurde dieses als *Graph matching* bezeichnete Verfahren bereits erläutert. In der parallelen Erkunden verschiedener Bereiche eines Datensatzes würde ließe sich damit die Orientierung besser behalten.

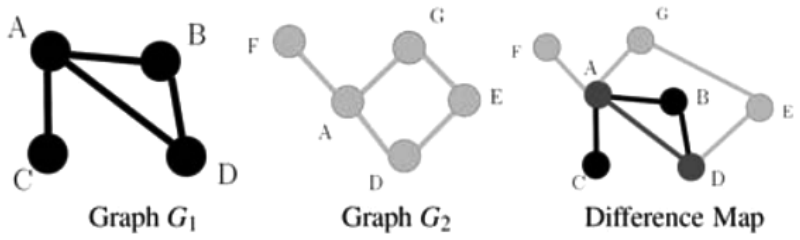


Abbildung 3.15: Schematische Darstellung der Vereinigung und Differenzierung zweier Graphen. *Quelle: Landesberger et al. (2011), S. 19*

Externalisierung

Entsprechend der Auffassung der Wissensspeicherung in Form von semantischen Einheiten, werden neue Informationen besser aufgenommen und langfristig gelernt wenn der Rezipient die Möglichkeit hat sie seinem bestehenden Wissen anzugliedern. Die aktive Umstrukturiert von Daten nach eigenem Verständnis ist dabei ein wichtiger Aspekt [25]. Ein aktives Ansammeln von Informationen ist ausschließlich in *Jigsaw* und in geringem Umfang in *Qiqqa* möglich. Besonders innovativ sind die Funktionen des *Jigsaw-Tablets*, auf dem die Analysezustände jeder Ansicht festgehalten werden können, wodurch eine effektive Verbindung zwischen Externalisierung und weiterführender Interaktion erfolgt. In neueren Versionen von Jigsaw wurde das *Tablet* dahingehend erweitert, dass individuelle Strukturen aus den gesammelten Daten erstellt werden können. Es vereint damit die Funktionen der Shoebox und der Schemata des Sensemaking Prozesses. Die anderen Arbeiten beschränken sich auf das reine Konsumieren visueller Informationen, der Filterung von Daten und der Parametrisierung von Visualisierungen. Eine Datenmodifikation durch wiederholte Ausführung eines Topic Model wie es in *iVisClustering* und *Tiara* möglich ist, ist ein erster Schritt in die individuelle Strukturierung von Topic Models. *Qiqqa* ermöglicht lediglich das Einfügen eigener Verbindungen und Texte in node-link Graphen.

Objekte können nicht gruppiert werden, wodurch das Umordnen von Elementen sehr beschränkt bleibt und auch in keiner Form Auswirkungen auf die Daten zeigt. Vom Nutzer geschaffene Strukturen lassen sich daher nicht festhalten und nicht weiter verwenden. Auch wenn alle Arbeiten das Ziel der Sinnstiftung verfolgen, ist auffällig wie wenig auf die tatsächlichen kognitiven Prozesse der Wissensaneignung eingegangen wird. hauptsächlich umfassen die Funktionen ausschließlich interaktive Visualisierungen, ohne den Prozess der aktiven Wissensaneignung zu unterstützen. Damit bleibt der Aktionsradius auf den Konsum von Informationen beschränkt.

	Serendip	**Qiqqa**	**Jigsaw**	**TopicViz**	**Tiara**	**iVisClustering**	**D-Vita**	**hiérarchie**	**Carrot²**
Ziel	ES mittels mit Zoom über mehrere Abstraktionsniveau	Organisation von, und ES in wissenschaftlichen Arbeiten	Analyse von Dokumenten unbestimmten Formats (z.B. Polizeiberichten)	Entgegenwirken des Informationsverlusts von Dimensionsreduktionen mittels interaktivem Graph	Zeitlicher Verlauf von Themen in kurzen Dokumenten	Optimierung der Cluster eines Topic Models	Zeitlicher Verlauf von Themen	ES über hierarchische Topic Model	Clustern von Suchresultaten und kleinen Dokumentkorpora
Algorithmus	LDA	LSI	k-Means	-	k-Menas, LDA, ...	LDA	dyn.-LDA	(h)LDA	Lingo(3G)
Orientierung	Globales Highlighten	-	Globales Highlighten	-	-	Globale Färbung & visuelle Filter	Verlaufsprotokoll als Baumstruktur	*Anker*-Ansicht	-
Externalisierung	-	Mindmap mit eigenen Kanten & Texte	*Tablet*: Festhalten von Zuständen & Erzeugung eigener Strukturen	-	-	Ordnen von Themen in einer Baumstruktur	-	-	-
Parametrisierung	-	-	Wahl des Clusteralgorithmus & der Clusterzahl	-	Mergen & splitten von Themen	Mergen & splitten von Themen, Umordnen von Dokumenten	-	-	u.A. Wahl des Clusteralgorithmus & der Clusterzahl
Recommendation anhand von ...	Dokumenten & Themen	Dokumenten, Themen & eigener Auswahl (nur in der Brainstorm)	Entitäten & Clustern	Räumlicher Nähe im Graphen (visuell)	Themen	Räumlicher Nähe von Dokumenten im Graphen (visuell)	Dokumenten & Themen	-	Clustern
Themenvergleich anhand von ...	Gewichtung im Textverlauf & Dokumentumfang	Graphen (visuell)	Clustern (visuell)	Räumlicher Nähe im Graphen (visuell)	Stapelgraph (visuell)	Parallelplot	Stapelgraph (visuell)	-	Volumen (visuell)
Gewichtung über den gesamten Korpus	Terme	Terme (ohne genauen Anteil)	Terme	Terme	Terme (Anteil entspricht Schriftgröße)	Terme, Themen (Parallelplot)	Terme & Themen im zeitlichen Verlauf	Terme (ohne genauen Anteil), Themen	Themen
Gewichtung innerhalb eines Dokuments	Themen als Verlauf & Terme durch farbliche Markierung & Sättigung	Terme, Themen	Terme	-	-	Terme	Terme	-	-

Tabelle 3.1: Zusammenfassung der Evaluationsergebnisse.

4 Ableitung

In diesem Kapitel werden die Erkenntnisse der Evaluation verwendet um identifizierte Beschränkungen zu konkretisieren und eine Auswahl exemplarischer Lösungsansätze zu beschreiben, die einen Ausblick auf mögliche Fortführungen der Forschung geben sollen.

Zwar wurde in den betrachteten Arbeiten das Sensemaking nur in geringem Umfang unterstützt, dafür wurden interessante neue Funktionen für den Umgang mit Topic Models beschrieben. Dazu gehört vor Allem die direkte Manipulation des Models, wie etwa durch *mergen* und *splitten* von Themen in Tiara und das Verändern der Ausgangsparameter in iVisClustering. Für eine effiziente Integration der Funktionen in den Workflow, wurden Veränderungen vorwiegend durch Inference-Mechanismen auf die Daten angewandt. Damit wird Funktionalität und Effizienz verbunden und zeigt, dass auch aus der algorithmischen Perspektive noch Möglichkeiten offen stehen, um die Interaktion mit Topic Models zu bestärken.

Das Sensemaking wird von den meisten Autoren als das umfassende Ziel ihrer Arbeit beschrieben, auf das sie mit einer neuen und speziellen Funktion hinarbeiten. Dabei vernachlässigen sie überwiegend die Einordnung neuer Erkenntnisse in ein globales Konzept, wodurch die Vergleichbarkeit von Arbeiten erschwert wird. Neben dieser generellen Erkenntnis fielen in der Evaluation folgende Beschränkungen auf:

- Die Strukturierung von Elementen in Hierarchien oder in einfachen Gruppen wird unzureichend unterstützt.
- Überwiegend werden die Resultate der Topic Models direkt verwendet, ohne sie in einen lokalen und elementspezifischen Kontext zu setzen. In der interaktiven Analyse sind solche Charakteristiken aber oftmals wichtiger als generelle Ausprägungen.
- Bei der Verwendung mehrerer Arbeitsbereiche kommt es zu Redundanzen, die durch die wiederholte Verwendung gleicher Funktionen entstehen.
- Node-link Graphen werden fast ausschließlich als statische Visualisierungen verwendet, ohne weitere Informationen wie etwa die Relationsgewichtung zu integrieren.
- Der Einstieg in die Analyse und die Orientierung in der Analyse werden nicht aktiv unterstützt.

- Es gibt beinahe keine Externalisierungsmöglichkeiten für gesammelte Informationen.[1]
- Der sensemaking loop wird nicht aktiv unterstützt. Dazu müssten Externalisierungen weiterverwendbar sein.
- Datenvergleiche von Gruppen und Elementen verschiedener Typen werden nicht unterstützt.

Chuang et al. stellen in mehreren Arbeiten vier grundlegende Probleme von Topic Models heraus: *nonsens Themen (Junk-Topics), verbundene Themen, fehlende Themen* und *wiederholte Themen* [42, 38, 51, 52]. Chuang et al. beschreiben interaktive Methoden zur Behebung dieser Probleme, die jedoch allesamt auf die Modifizierung der LDA-Parameter und erneuten Filterung der Eingabedaten basieren. Im Rahmen der folgenden Konzepte wird versucht eine Auswahl der von Chuang et al. als auch der zuvor herausgearbeiteten Probleme mit den Resultaten des Topic Models zu beheben. Dazu wird die explorative Suche mit dem Sensemaking Prozess nach Pirolli et al. verbunden [1]. Im ersten Schritt wird die automatische und manuelle Ordnung der Datenstruktur behandelt, um die Übersicht und den Grad der Individualisierung zu Steigern. Anschließend wird die Externalisierung von Daten in der Shoebox und als Schemata bearbeitet und analysiert wie diese weiter Verwendet werden können, um den foreaging und sensemaking loop zu schließen.

4.1 Strukturieren von Themen und Relationsmatrizen

Topic Models stellen Relationen zwischen Dokumenten, Themen und Termen her. Bislang stellt sich die Frage wie damit im ersten Schritt umgegangen werden soll. In der Visualisierung könnten besonders starke Relationen ausgemacht werden, alternativ könnte ein Dokument herausgesucht werden und von dort aus nach ähnlichen Dokumenten gesucht werden. Eine Strukturierung von Themen und Dokumenten hilft in der Übersicht und stellt wichtige Inhalte direkt heraus. Damit ließe sich ein einfacher Einstieg finden und Sinneinheiten schnell identifizieren. Im Folgenden soll eine Methode beschrieben werden, die das einfache und effektive Agieren mit Themen erlaubt und dem Nutzer zusätzlich als Orientierungshilfe dient.

Wie in Abschnitt 2.1.2 erläutert wurde, muss die Zahl der von der LDA zu findenden Themen K vorab vom Nutzer bestimmt werden. Themen werden optimal erfasst, wenn K den tatsächlich in einem Dokumentkorpus behandelten Themen $\dot{K}$ entspricht. $\dot{K}$ ist aber üblicherweise nicht bekannt und wie Chang et al. befunden haben unterscheidet sich die menschliche Interpretation eines Topic Models von dessen numerischer Exaktheit [19].

[1] Nur in Jigsaw war das möglich, dort wurde allerdings kein Topic Model verwendet.

Bei einer Vielzahl von Themen kommt es schnell zu einer Überforderung des Nutzers. Nach Millers *Magical Number* [23] kann davon ausgegangen werden, dass über mit fünf verschiedene Themen am besten gearbeitet werden kann. Wie lässt sich nun die Zahl der angezeigten Themen mit K und $\dot{K}$ vereinen? Die einfachste Möglichkeit wäre, dass unabhängig von K ausschließlich die fünf am höchsten gewichteten Themen $\max_5 P(k_n)$ angezeigt werden. Nach Alsumait et al. [53] teilen sich tatsächliche Themen in mehrere gefundene Themen auf, wenn $K \gg \dot{K}$ gewählt wird. Für die häufig getroffene Wahl $K = 50$, wie von Wei et al. empfohlen [17], käme es in vielen Textkorpora zu einer solchen Aufteilung. Das Auswählen von $\max_5 P(k_n)$ würde dann nur einen Bruchteil eines jeden tatsächlichen Themas anzeigen. Die Alternative scheint die Wahl eines kleinen K zu sein. Für diesen Fall weisen Alsumait et al. darauf hin, dass es für $\dot{K} \gg K$ zu Überschneidungen tatsächlicher Themen in gefundenen Themen kommt. Ein weiteres Problem wäre, dass für eine Differenzierung erneute Berechnungen der LDA durchgeführt werden müssen, wie es etwa in *Tiara* und *iVisClustering* erfolgt. Wie in Abschnitt 3.2.9 beschrieben, begegnet *hiérarchie* diesem Problem mit einer hierarchischen Anwendung der LDA. Der Nutzer ist dadurch hinsichtlich der Aufteilung, als auch in der Tiefe der hierarchiestufen, auf das gegebene Datenmodell reduziert. Des Weiteren ist eine wiederholte Anwendung der LDA rechenintensiv.

Um den genannten Problemen entgegen zu kommen wird ein beliebig großes K gewählt und die gefundenen Themen nachträglich nach Ähnlichkeiten gruppiert. Die zugrundeliegende Annahme ist, dass mehrere gefundene Themen die ein tatsächliches Thema vereinen in ähnlichen Textabschnitten auftreten. Unter dieser Voraussetzung können die Relationen der Themen durch das Kosinusmaß, Formel 2.4 aus Abschnitt 2.1.3, berechnet werden. Das Prinzip gleicht der Bestimmung von Dokumentrelationen in σ_D, nur dass stattdessen aus der Dokument-Themen-Relationsmatrix θ eine Relationsmatrix der Themen $\sigma_T \in \mathbb{R}^{T \times T}$ gebildet wird. Entsprechend der Formel 2.3 in Abschnitt 2.1.2, wird nun das Kosinusmaß der Spaltenvektoren von θ berechnet. Die Dimensionalität ergibt sich aus der Menge von Dokumenten D, ein Vektor ist daher beschrieben durch $\theta_j \in \mathbb{R}^D$.

Der nun beschriebene Algorithmus reduziert die Dimensionalität σ_T von $T \times T$ auf ein beliebiges $\dot{T} < T$ und erzeugt dabei eine hierarchische Struktur von Themen. Iterativ werden die folgenden Schritte durchgeführt, bis das gewünschte $\dot{T}$ erreicht ist:

1. Gegeben ist $\sigma \in \mathbb{R}^{T \times T}$
2. Bestimme $\underset{i,j}{\mathbf{argmax}}(\sigma_{i,j})$, also die Zelle mit der höchsten Relation
3. Erzeuge $\dot{\sigma} \in \mathbb{R}^{(T-1)\times(T-1)}$ und übernehme alle Einträge aus σ mit Ausnahme von Zeile j und Spalte j

4. Berechne den Durchschnitt aus den Spaltenvektoren von i und j durch $\dot{\sigma}_i = \frac{\sigma_i + \sigma_j}{2}$
5. Übertrage den Spaltenvektor $\dot{\sigma}_i$ auf den Zeilenvektor, sodass $\dot{\sigma}$ symmetrisch wird.
6. Speichere die Zuweisung von $i \mapsto j$
7. Setze $\sigma = \dot{\sigma}$

Der Programmcode ist in Abschnitt 5.3 in Liste 5.1 zu finden. Diese Implementierung arbeitet rekursiv[2] und speichert die Hierarchie der Themen im Objekt *TopicCluster* dessen UML-Klassendiagramm in Abbildung 5.1 zu sehen ist.

Durch die Eliminierung der höchsten Relationen entstehen Gruppen die sich inhaltlich klar voneinander abgrenzen. Es handelt sich bei dem Algorithmus also um ein einfaches Clusterverfahren für Relationsmatrizen. Liste 5.2 zeigt eine Ausgabe des Algorithmus, angewendet auf drei verschiedene Textquellen[3]. Der Algorithmus wurde für die Bestimmung von drei Oberthemen angewandt und hat tatsächlich alle drei Quellen klar unterschieden. Wenn man sich in der Liste die Themen 2 und 4 anschaut, werden die Probleme der LDA deutlich. Diese beiden Themen beinhalten keine (sichtbar) relevante Charakteristik und sind in der Hierarchie trotzdem recht hoch angesiedelt. Dieses Problem könnte durch die niedrige Gewichtung von *JunkTopics* gelöst werden. Momentan wird die Hierarchie nach den größten Relationen gebildet, sodass es in höheren Ebenen zu stärkeren Spezialisierungen kommt. Die Vergleichsfunktion des Algorithmus könnte auf andere Faktoren wie den Umfang eines Themas erweitert werden.

Der Vorteil des Algorithmus ist, dass vorab ein hohes K gewählt werden kann, ohne dass Themen unnötig weit aufgeilt werden. Unterteilte Themen können nämlich einfach in der Filterung und in der Visualisierung temporär in einem Meta-Knoten[4] zusammengefügt werden. Der Vorteil einer hierarchischen Ordnung in der Anwendung ist, dass ein Nutzer selbst bestimmen kann wie viele Themen angezeigt werden und entsprechend wie hoch die Informationslast ist. Des Weiteren ist davon auszugehen, dass hierarchische Strukturen die Orientierung unterstützen, da sie Dokumente in Sinneinheiten zusammenführen. Man vergleiche dazu die Matrix-Ansicht von Serendip mit der Sunflowchart von Hiérarchie. Der beschriebene Algorithmus kombiniert Vorteile beider Arbeiten indem eine einfache Übersicht ermöglicht wird, deren unterliegende Struktur jedoch nicht bindend ist und vom Nutzer frei verändert werden kann. Die Themen-Struktur könnte entweder direkt exploriert werden oder als Dokumentfilter verwendet werden. Der Programmcode in Liste 5.2

[2] Da es sich um eine konzeptuelle Umsetzung handelt, wird der Ressourcenverbrauch der rekursiven Implementationsform hier nicht beachtet.

[3] Es handelt sich dabei um wissenschaftliche Publikationen über Text- und Wikipedia-Mining, das Buch *Der Mann ohne Eigenschaften* und das Buch *Antichrist*.

[4] Damit ist ein Knoten bezeichnet, der in den Daten nicht existiert sondern nachträglich und temporär mit dem Zweck der Strukturierung eingefügt wurde.

macht deutlich, dass das Verfahren auch auf Dokumentrelationen angewandt werden kann. Auch wenn eine thematische Ordnung sinnvoller erscheint, kann eine Hierarchisierung von Dokumenten den Einstieg in die Dokumentexploration erleichtern.

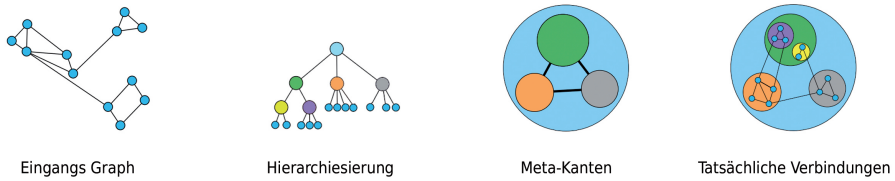


Abbildung 4.1: Stilisierte Darstellung der Hierarchisierung und aggregierten Darstellung eines Graphen. (a) erstellen und (b) löschen eines Knoten. *Quelle: Archambault et al. (2008)*

Eine direkte Exploration könnte mit der Methode *GrouseFlocks* von Archambault et al. erfolgen, einer Methode für die Exploration von hierarchischen Graphen [54]. Abbildung 4.1 zeigt wie ein eingehender Graph in eine hierarchische Struktur umgewandelt wird. Der eingehende Graph ist in diesem Fall durch die Relationsmatrix der Themen gegeben, die als Adjazenzmatrix dient. Die Hierarchie wird durch den vorab beschriebenen Algorithmus gebildet. Es ist zu beachten, dass weitere Verbindungen zwischen Knoten durch die Hierarchisierung nicht verloren gehen, sie sind bloß nicht eingezeichnet. Die dritte Abbildung zeigt die Übertragung des Graphen auf eine zweidimensionale Fläche. In der vierten Abbildung werden die Verbindungen der Blatt-Knoten deutlich, die in darüber liegenden Ebenen zu Meta-Kanten zusammengefügt wurden. Neben dem Durchforsten des hierarchischen Graphen sehen Archambault et al. die Möglichkeit vor, Meta-Knoten als individuelle Verbindungselemente zu verwenden, durch die eine weitere Freiheit in der Strukturierung von Daten entsteht.

4.2 Steigerung der Themencharakteristik

Die meisten Anwendungen verwenden die Ausgabe eines Topic Models direkt ohne diese weiter zu bearbeiten, obwohl die Charakteristik eines Elements oftmals viel interessanter ist als die allgemeinen Werte. Wenn ein Nutzer beispielsweise die Beziehung zweier Dokumente betrachten möchte, dann würde durch die direkte Verwendung des Topic Models das am höchsten gewichtete Thema ausgegeben werden. Kommt dieses Thema in sehr vielen Dokumenten vor, so sinkt seine Entropie und folglich sagt es nicht mehr viel über die spezielle Verbindung der gewählten Dokumente aus. Die Vorteile der Verarbeitung von Ergebnissen eines Topic Models sollen nun anhand der Normierung von Themen angedeutet werden. Dabei werden lokale Gewichtungen von Themen in Relation zu ihrem globalen Vorkommen im Dokumentkorpus gesetzt. In der Informationstheorie wird

von einem logarithmischen Abfall der Entropie in Abhängigkeit zur Aufkommensfrequenz ausgegangen, wie es etwa in Formel 5.1 des td-idf-Maß umgesetzt wird. Aber auch die einfache Normalisierung macht den Nutzen einer Relativierung von Themengewichtungen deutlich.

Abbildung 4.2 zeigt eine Visualisierung der Dokumentrelationsmatrix vor und nach der Normalisierung[5]. Der Dokumentkorpus setzt sich aus mehreren wissenschaftlichen Publikationen und vier Büchern zusammen, deren Seiten als jeweils einzelne Dokumente eingelesen wurden. Die Matrix beschreibt die Relationen zwischen allen Dokumenten und ist daher symmetrische. Jeder Pixel entspricht einem Eintrag und damit der Relation zweier Dokumente. Die Sättigung und Helligkeit eines jeden Punktes ergibt sich aus dem Kosinusmaß der beiden Vektoren. Jede Farbe entspricht einem Thema, wobei die Farbe eines Punktes das Thema beschreibt, das die beiden Dokumente am stärksten verbindet. Das am stärksten verbindende Thema zweier Vektoren $\vec{a}$ und $\vec{b}$ wird durch das Produkt der Vektorelemente bestimmt:

$$\underset{i}{\mathbf{argmax}}(a_i * b_i) \quad mit \quad 1 \leq i \leq K \tag{4.1}$$

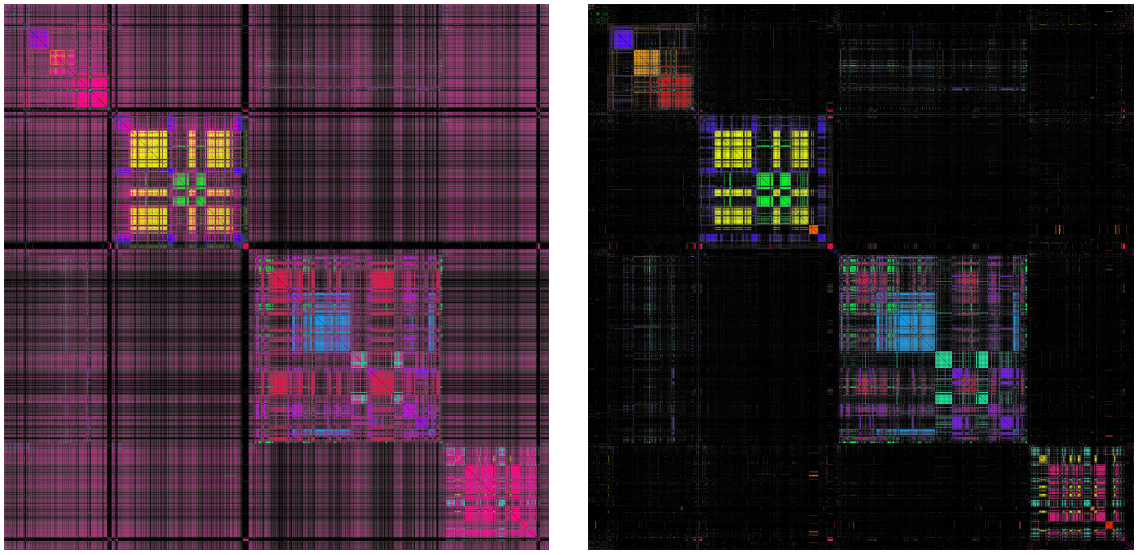
Im Bild vor der Normalisierung ist zu sehen, dass sich das pinke Thema (Nummer 45) über den ganzen Korpus erstreckt. In der Ausgabe der Terme und der Häufigkeit der Terme für das jeweilige Thema fällt auf, dass es sich sicherlich um ein *Junk-Topic* handelt, das viele Terme vereint die üblicherweise als *Stopwords* herausgefiltert werden. Zum Vergleich ist das gelbe Thema (Nummer 9) angegeben, das trotz einiger *Stopwords* wesentlich geringere Termfrequenzen aufweist. Es ist in beiden Bildern der Abbildung 4.2 deutlich erkennbar und beschränkt sich auf eine einzige Quelle[6].

```
45: the (78301) of (33022) a (27682) to (27530) is (19304) and (16691)
that (13011) for (9876) this (9736) as (8653)
9: the (1508) aspect (793) join (750) pointcut (634) method (576)
advice (570) account (569) we (523) public (451) class (451)
```

Die Normalisierung führt dazu, dass weit gestreute Themen an Gewicht verlieren und selten vorkommende Themen verstärkt werden. Dadurch wird der Themen-Kontrast zwischen Dokumenten verstärkt und die *Eigenheiten* betont. In Abbildung 4.2 ist zu sehen wie Thema 45 durch die Normalisierung beinahe vollständig eliminiert wurde und verdeckte Themen freigibt.

[5]Die Darstellung ist ausschließlich dafür gedacht das Verhalten des Datenmodelles zu erklären und wurde nicht im Hinblick auf eine Endanwendung konzipiert.

[6]Das Buch *Aspect J in Action* von Ramnivas Laddad

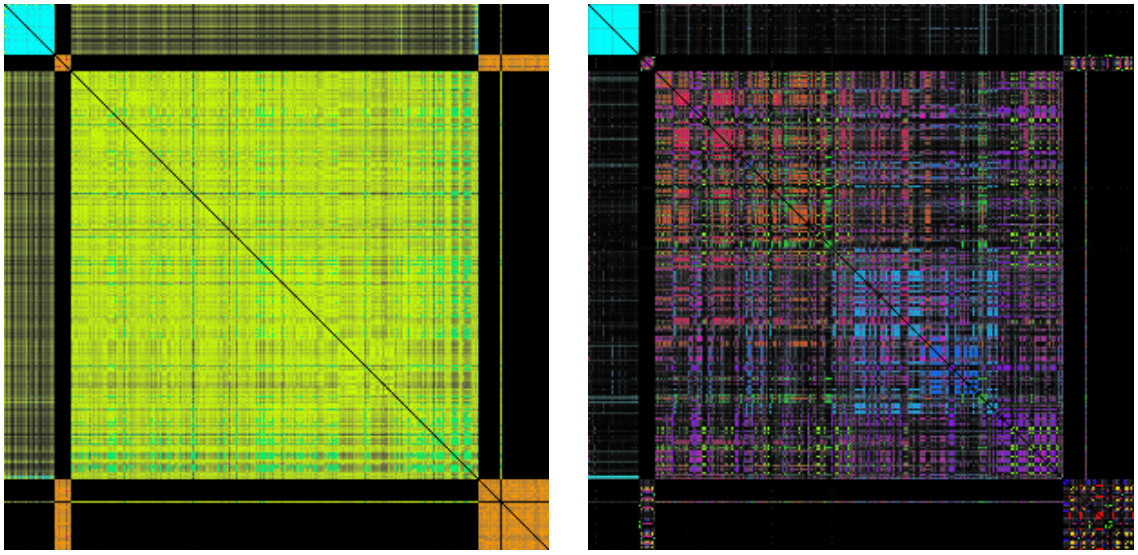


(a) Ohne Normalisierung (b) Mit Normalisierung

Abbildung 4.2: Der Plot einer Dokumentrelationsmatrix, in dem die Sättigung der Relationsstärke und die Farbe dem stärksten, zwei Dokumente verbindenden Thema entspricht. Das rötliche Thema löst sich auf und lässt die anderen Themen deutlicher hervortreten.

Wie das folgende Beispiel zeigt hängt der Nutzen stark vom vorliegenden Dokumentkorpus ab. Abbildung 4.3 zeigt die Anwendung des gleichen Plots auf einen Dokumentkorpus der aus drei inhaltlich sehr verschiedenen Quellen besteht, die auch in Abschnitt 5.3 verwendet werden[7]. Die klare Differenzierung der Gruppen in Bild (a) wird durch die Normalisierung aufgehoben. Bild (b) zeigt eine Differenzierung der Quellen, durch die unter Umständen besser die Verbindungen innerhalb einer Quelle analysiert werden können. Andererseits wird die *örtliche* Bindung der Quellen aufgehoben. Wie bereits erwähnt, unterliegt die LDA der Beschränkung, dass die Reihenfolge von Dokumenten nicht beachtet wird, eine Bedingung die in dem vorliegenden Fall von Bedeutung ist. Ohne die Normalisierung sind die Verbindungen eindeutiger und mit Normalisierung lässt sich unter Umständen besser verstehen, was genau Typisch für diese Verbindung ist. Auf Grund der Abhängigkeit vom Dokumentkorpus und der individuellen Interpretierbarkeit sollten einem Nutzer beide Ansichten angeboten werden. Somit ließen sich möglicherweise die Verbindungen zwischen Elementen besser verstehen. Die Normalisierung kann ebenso auf andere Relationsmatrizen angewandt werden und da die Berechnungen auf dem bereits trainierten Model der LDA basieren, können sie effizient auch auf große Datenmengen angewandt werden.

[7]Die drei Quellen sind wissenschaftliche Publikationen verschiedener Themen auf englischer Sprache, *Der Mann ohne Eigenschaften* von Robert Musil und *Der Antichrist* von Friedrich Nietzsche, jeweils auf Deutsch.



(a) Ohne Normalisierung (b) Mit Normalisierung

Abbildung 4.3: Der Plot einer Dokumentrelationsmatrix, in dem die Sättigung der Relationsstärke und die Farbe dem höchsten zwei Dokumente verbindenden Thema entspricht. Ein Negativbeispiel für die Zergliederung von Themen-Einheiten.

4.3 Vergleichbarkeit von Elementgruppen

Der Vergleich von Elementen und Elementgruppen wird in allen betrachteten Arbeiten sehr gering bis überhaupt nicht unterstützt. In den meisten Fällen beschränkt sich ein *Vergleich* auf die Angabe des Kosinusmaß zweier Dokumente. Auch die Vereinigung verschiedener Elementtypen und die Gruppierung von Daten werden nur sehr geringfügig behandelt. Für eine aktive Nutzung externalisierten Wissens ist die Rückführung festgehaltener Informationen und erstellter Strukturen notwendig. Wird im Falle der Shoebox eine Art Notizblock angeboten, so stellt sich die Frage wie mit diesen Notizen weiter umgegangen wird. Auf Abbildung 2.5 ist zu sehen, dass die Auslagerung von Elementen und von Wissensstrukturen eine einseitige Unterstützung des Sensemakings bedeutet. Für eine tatsächliche Wechselwirkung zwischen Mensch und Maschine dürfen diese Auslagerungen nicht nur gespeichert werden, sondern müssen genutzt werden um Informationen zu suchen und zu evaluieren. Damit ließe sich das Arbeitsgedächtnis entlasten und zugleich komplexe Suchen formuliert werden, die über den einfachen Vergleich zweier Dokumente hinausgehen.

Nun wird eine Methode vorgestellt durch die Dokumente, Themen und Terme in einer Gruppe vereint und mit anderen Elementen beliebigen Typs verglichen werden können. Es entsteht damit eine Generalisierung des VSM durch die sich die Ähnlichkeit von Dokument und Thema, Dokument und Term oder Thema und Term einfacher bestimmen lässt. Der

Nutzen dieser Verallgemeinerung ist die Verbindung von Datentypen, der sich auch in der Interaktion niederschlägt. Somit könnte ein *Werkzeug* geschaffen werden, welches die Fenster für einzelne Typvergleich obsolet macht.

Das hier verfolgte Ziel ist die Generierung von Gruppen, die aus Elementen verschiedenen Typs (Dokument, Themen, Termen) bestehen. Die alles verbindende Grundlage eines Topic Models sind Themengewichtungen, die durch das Kosinus-Maß verglichen werden. Daher sollen die Gruppen durch einen Vektor ausgedrückt werden, der sich im VSM verwendet lässt. Für die nun verwendeten Bezeichnungen sei auf Formeln 2.2 und 2.3 in Abschnitt 2.1.2 hingewiesen. Alle Elemente müssen sich für die Anwendung im VSM auf einen Vektor $\vec{v} \in \mathbb{R}^K$ reduzieren lassen, wobei K der Zahl der Themen entspricht. Die Bestimmung von $\vec{v}$ erfolgt für jeden Typus individuell. Der einfachste Fall ist das Dokument, für den die Vektoren schon in θ gegeben sind, sodass $\vec{v_i} = \theta_i$ gilt. Die Bestimmung des Vektors $\vec{v_i}$ für das Thema mit dem Index i erfolgt durch das Bilden eines K-dimensionalen $\vec{0}$ dessen Index i auf 1 gesetzt wird:

$$v_{i,k} = \begin{cases} 1 & \text{wenn } k = i \\ 0 & \text{sonst} \end{cases} \tag{4.2}$$

Der Vektor eines Terms entspricht einem Zeilenvektor der Matrix ϕ. Die Bestimmung dieser Werte gestaltet sich etwas komplizierter und wird genauer in Anhang 5.3 beschrieben. Wenn die Vektoren aller Elemente einer Gruppe bestimmt wurden lassen sie sich einfach Summieren und ergeben den Gewichtungsvektor der Gruppe $\vec{g}$. Die Summierung ist möglich da eine Gruppe eine ungeordnete Menge ist, sodass keine Beziehungen unter den Elementen beachtet werden müssen. Der Faktor der *Relation* unterscheidet das Konzept der Shoebox wesentlich von Schemata, die aus Vernetzungen von Elementen bestehen. Innerhalb einer Gruppe können den Elementen individuelle Gewichtungen α_i zugeteilt werden, sodass der Nutzer einen größeren Aktionsradius in der Auslagerung und Filterung erhält. Die Summe wird durch die Anzahl der Elemente $|V|$ dividiert, damit sich die Werte $\vec{g_i}$ im Intervall $[0, 1]$ befinden. Da das Kosinus-Maß die Länge eines Vektors nicht beachtet, ist dieser Schritt nicht zwingend notwendig. Durch die Beschränkung auf das Intervall $[0, 1]$ ist jedoch eine weitere Verwendung von $\vec{g}$ unproblematisch, sodass sich auch Gruppen wiederum Gruppieren ließen. Die gesamte Berechnung lässt sich folgendermaßen beschreiben:

$$\vec{g} = \frac{1}{|V|} * \sum_{i=1}^{|V|} \alpha_i * \vec{v_i} \tag{4.3}$$

4.4 Schemabildung

In Abschnitt 2 wurde die Funktion eines Schemas im Sinne der Kognitionspsychologie als Vorstellung eines Konzeptes beschrieben, dass auf die Informationen des Kurzzeitgedächtnisses gelegt wird, um aus diesen eine Sinneinheit zu bilden. Im Sensemaking-Prozess nach Pirolli et al. [1] nehmen Schemata eine wichtige Position für die Generierung von Hypothese ein. Stasko et al. haben in Jigsaw das *Tablet* entwickelt, in dem ein Nutzer eigene Strukturen bilden kann, die als externalisierte Schemata verstanden werden können. Für eine aktive Integration in den Prozess des Sensemaking muss eine weitere Verwendung dieser Strukturen erfolgen. Diese Form der Suche unterstützt das bedeutungserzeugende und das ineretielle Lernen. Damit wird das Ordnen von Wissen in direkter und externer, also nicht rein gedanklicher Form ermöglicht[8]. Durch die direkte Integration einer Externalisierung als Filter wird der Suchprozess, wie er in Abschnitt 2.2.3 und 2.2.2 beschrieben wird, aktiv unterstützt und die Iterationen des sensemaking loops wie sie Abbildung 2.5 zeigt, werden durch die direkte Auswertung der Eingabe beschleunigt. Die Verwendung von Schemata soll nun an einem Konzept angedeutet werden, mit dem einige Grenzen der Anwendung von Topic Models überwunden werden sollen. Dabei soll die Externalisierung mit der Visualisierung und Filterung verbunden werden, um dem Nutzer das schnelle Wechseln zwischen Abstraktionen und konkreten Inhalten zu ermöglichen.

Die Verwendung von (gewichteten) Graphen als Präsentationsform ist aus den folgenden Gründen eine gute Wahl:

- Die LDA generiert gewichtete Relationen zwischen Elementen auf der Basis von Themen.
- Node-Link Graphen werden bereits häufig verwendet und erfreuen sich einer gewissen Akzeptanz bei den Nutzer. Des Weiteren bestehen bereits Umsetzungen und Werkzeuge für die Arbeit mit Node-Link Graphen.
- Durch Graphen lassen sich gut die Verbindungen einer Vielzahl von Elementen darstellen.
- Nach Seel und Prinz wird das Wissensgedächtnis (semantisches Gedächtnis) als Netz von Objekten (Schemen) verstanden [24, 25][9].

In Abschnitt 4.3 wurde bereits erläutert wie einzelne Terme, Themen und Dokumente durch einen Gewichtungsvektor beschrieben werden können. Externe Quellen und vom Nutzer frei formulierte Texte können eingebunden werden indem das trainierte Topic Model auf die neuen Inhalte angewandt wird.

[8] Dazu sei auf Abschnitt 2.2.2 verwiesen.
[9] Eine Beschreibung findet sich in Abschnitt 2.2.2.

Die *Inference* (dt. Schlussfolgerung) bezeichnet das Anwenden eines antrainierten statistischen Modells auf neue Daten. Das Trainieren von Modellen dauert relativ lange und sollte möglichst selten durchgeführt werden. Wird ein Topic Model mit einem großen Textkorpus trainiert wird ein entsprechend großes Alphabet erstellt, das alle im Korpus verwendeten Wörter umfasst. Wenn eine verhältnismäßig geringe Zahl neuer Dokumente in das bestehende Schema eingeordnet werden soll, können die bestehenden Wahrscheinlichkeiten sehr schnell auf diese angewandt werden. Die Bedingung für eine sinnvolle Folgerung über neue Daten ist, dass das bestehende Alphabet weit genug gefasst ist und die meisten Wörter der neuen Daten bereits darin enthalten sind. Ein auf medizinische Fachzeitschriften trainiertes Topic Model lässt sich entsprechend schlecht auf ein Buch über ID anwenden. Das Verwenden externer Textdateien, Webinhalten oder eigenen Texten ist damit problemlos möglich. Ein node-link Diagramm das aus verschiedenen Elementtypen mit gewichteten Relationen besteht, könnte wie in Abbildung 4.4 aussehen.

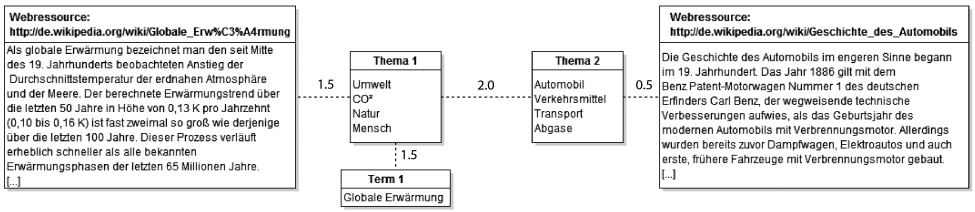


Abbildung 4.4: Beispiel eines vom Nutzer entworfenen gewichteten Graphen verschiedener Elementtypen.

Das Erzeugen eines Schemas hat den Vorteil, Informationen auslagern und individuell anordnen zu können. Die tatsächliche Integration erfolgt jedoch nur, wenn sich diese Strukturen wiederum auf die restlichen Daten anwenden lassen. Dazu ist ein Vergleich notwendig, für den bislang das VSM verwendet wurde. Die Vereinigung eines gewichteten Graphen mit dem VSM ist jedoch schwierig. Die Übliche Anwendung wie sie in Abbildung 2.3 zu sehen ist, beschreibt alle Elemente und Suchanfragen durch einen einzigen Vektor. Die Ähnlichkeit zweier Vektoren lässt sich dann durch die Differenz der Ausrichtung bestimmen, die mit dem Kosinusmaß berechnet wird. Das VSM wurde bereits in Abschnitt 2.1.3 beschrieben, es soll hier noch einmal kurz auf wesentliche Eigenschaften eingegangen werden. Stock [8] weist auf die limitierende Annahme des VSM hin, dass alle Dimensionen voneinander unabhängig sind. historisch gesehen erfolgte die erste Verwendung des VSM durch das Aufspannen von Dimensionen durch einzelne Terme des gesamten verwendeten Alphabetes. Durch die semantischen Relationen in denen Wörter jeder Sprache stehen, bleibt diese Annahme unerfüllt. Durch die LSI und auch die LDA wurde das Problem reduziert, da nun jede Dimension eine semantische Einheit repräsentiert. Die Unabhängigkeit der Dimensionen wird aber weiterhin angenommen. Stellen wir uns zwei Elemente in einem Graph vor die miteinander verbunden sind. Beide Elemente

lassen sich durch einen Vektor der Themengewichtung beschreiben. Nach der Behandlung von Abschnitt 4.3 ist es nicht relevant, um was für einen Typ von Element es sich handelt. Diese beiden Elemente werden von einem Nutzer in eine bestimmte Relation gesetzt, die so nicht in den Daten vorkommen muss, und der Vorstellung des Nutzers entspricht. Beide Vektoren können in das VSM eingetragen werden, sind aber voneinander Unabhängig, die Relation findet daher noch keine Beachtung. Es wird deutlich, dass sich ein Graph nicht durch einen Vektor ausdrücken lässt, der als Suchanfrage verwendet werden könnte.

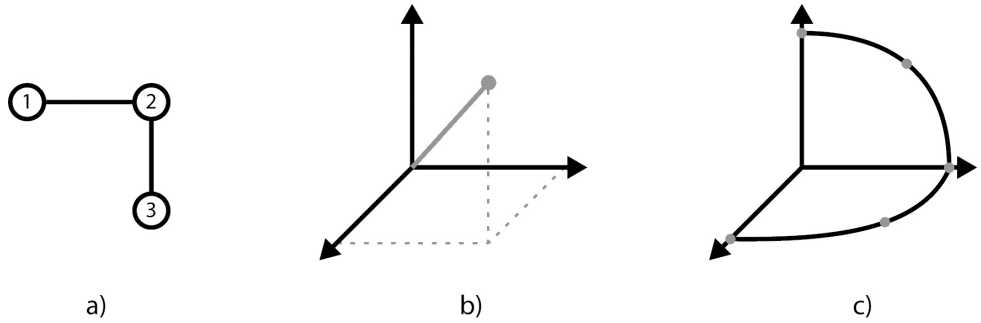


Abbildung 4.5: Schematische Darstellung eines Graphen bestehend aus drei Themen als a) node-link Diagramm, b) Reduktion auf einen Vektor im VSM und c) der Übertragung von Kanten und Fixpunkte.

Auch wenn ein Schema aus beliebigen Elementen erzeugt werden kann, wird sich nun erst einmal ausschließlich auf Themen beschränkt. Mit Abbildung 4.5 als Ausgangssituation ist ein Graph gegeben der aus drei Knoten (Themen) und zwei Kanten besteht, demnach sind zwei Knoten nicht miteinander verbunden. Um im VSM zu arbeiten müsste der Graph nun auf einen einzigen Vektor reduziert werden. Die mittlere Abbildung b) zeigt eine einfache Durchschnittsberechnung aller Themen. Das ist natürlich eine falsche Übertragung, da Elemente die sich stärker zwischen t_1 und t_2 oder zwischen t_2 und t_3 befinden relevanter sind, als solche die sich zwischen t_1 und t_3 befinden. Die rechte Abbildung c) zeigt eine schematische Übertragung des Graphen in das VSM. Dabei wurden die Kanten als Teile des Einheitskreises zwischen den verbundenen Themen eingezeichnet. Aus den eingezeichneten Fixpunkten lässt sich eine Suchmaske erzeugen, die auf den Vektorraum gelegt wird. Die Maske ist beschrieben durch eine K-dimensionale Funktion, die alle Ausrichtungen im Vektorraum bedient, sodass die Relevanz zu einem Schema nicht mehr durch das Kosinusmaß, sondern durch eben diese Funktion bestimmt wird. Eine den gesamten Vektorraum abdeckende Funktion entsteht durch die Interpolation über die gegebenen Punkte, die sich aus drei Knoten und den beiden Kanten des Graphen ergeben. Die Verwendeten Punkte müssen sich nicht auf Themen und damit auf die Achsen des Vektorraumes beschränken. Eine Interpolation kann genauso gut über Punkte anderer Elementvektoren durchgeführt werden. Dieses Konzept kann weiter ausgeführt werden, indem die Kanten eines Graphen gewichtet werden, wie es in Abbildung 4.4 dargestellt

ist. Darin werden zwei gefundene Themen in eine starke Verbindung gesetzt und um externe Quellen ergänzt. Diese können per Inference schnell in das VSM integriert werden. Durch die in Abschnitt 4.3 beschriebene Formel der Verallgemeinerung, kann auch problemlos ein einzelner Term verwendet werden. Die Abbildung zeigt ein exemplarisches Schema über die globale Erderwärmung durch Automobile. Irrelevante Verbindungen wie die *Geschichte des Automobils* können damit gezielt ausgeblendet werden.

Durch diese Umsetzung werden die Beschränkungen des VSM überwunden, sodass Schemata direkt für das Filtern von Daten verwendet werden können. In der Anwendung würden damit die Externalisierung von Wissen und die Filterung ineinander übergehen. Die Idee weiter gesponnen, lässt sich Vorstellen dass ein Nutzer mehrere Schemata entwirft die automatisch mit den momentan betrachteten Daten abgeglichen werden. Der Suchprozess gleicht sich damit der Funktionsweise des menschlichen Gedächtnisses an. Der Rechenaufwand einer hochdimensionalen Interpolation könnte Desktopsysteme überlasten und der Aufwand den ein Nutzer in die Suche investieren muss, steigt enorm an. Diese Probleme sind offensichtlich, aber nicht unüberwindbar.

5 Fazit und Diskussion

Die stetig wachsenden Datenmengen mit denen Unternehmen als auch Privatpersonen täglich konfrontiert werden verlangen nach neuen Methoden der Organisation, Analyse und Suche. Verfahren des Machine Learnings werden mittlerweile ganz selbstverständlich für die Analyse und Organisation von Daten angewandt. Die verwendeten Algorithmen werden immer leistungsstärker und exakter, verbunden mit steigenden Rechenleistungen lassen sie sich nun auch lokal und interaktiv verwenden. Bislang haben solche Systeme in der direkten Anwendung noch keinen Einzug in den Alltag erhalten, doch verspricht die engere Verknüpfung von menschlicher und maschineller Analyse eine effizientere Arbeitsweise in der Suche und der Datenverarbeitung. Das Bewusstsein für die Notwendigkeit von nutzerzentrierten Konzepten entwickelte sich im HCI schon sehr früh. Im speziellen Bereich der Exploration und besonders unter der Verwendung von Topic Models, muss dieses Bewusstsein erneut gebildet oder wenigstens vorangetrieben werden. In dieser Arbeit wurde die Hypothese behandelt, dass Modelle des Sensemakings den Explorationsprozess über Strukturen von Topic Models verbessern und einen Kontext für entsprechende Forschungsarbeiten bieten können.

Zur Überprüfung dieser Hypothese wurden Grundlagen der Vorverarbeitung von Texten und die Funktionsweise von Topic Models exemplarisch an der Latent Dirichlet Allocation (LDA) beschrieben. Die LDA wurde verwendet, da sie sich im Vergleich zu ähnlichen Verfahren von Menschen gut interpretieren lässt. Mit dem Vector Space Model zur Bestimmung von Dokumentähnlichkeiten wurde die gängige Nutzung der Resultate eines Topic Models beschrieben. Anschließend wurde die menschliche Informationsaufnahme und -verarbeitung behandelt und über Problemlösungsprozesse mit der explorativen Analyse und Suche unter der Bezeichnung des Sensemaking vereint. Dabei wurden die foreaging und sensemaking loops als zwei wesentliche Prozesse des Sensemakings herausgestellt. Im Rahmen des Interactive Information Retrieval wurden die Aufteilung von Datenansichten und der Umgang mit relationalen Daten behandelt. Relationen spielen in der Anwendung von Topic Models eine wichtige Rolle, da sich die von ihnen generierten Wahrscheinlichkeiten als ebensolche deuten lassen. Daraufhin wurden Erläuterungen der Grundlagen verwendet, um eine Evaluation von neun Arbeiten durchzuführen, die die Exploration von Textkorpora mittels Topic Models behandeln. Diese formative Evaluation verfolgte das

Ziel, Gemeinsamkeiten und Beschränkungen in der Unterstützung des Sensemaking zu identifizieren. Neben der algorithmischen Anwendung des Topic Models wurden spezielle Aspekte der Präsentation, der Interaktion und des Sensemaking betrachtet. Anschließend wurden die Arbeiten in den einzelnen Punkten zusammengeführt. In der Ableitung konnten dabei beobachtete Beschränkungen konkretisiert werden, von denen eine Auswahl exemplarisch behandelt wurde. Mit den vorgestellten Konzepten konnte gezeigt werden, dass sich Topic Models verstärkt mit den Modellen des Sensemaking vereinen lassen. Somit konnte eine Perspektive für zukünftige Forschungsarbeiten geboten werden.

5.1 Der Beitrag dieser Arbeit

Im ersten Schritt wurde eine Übersicht der Einflüsse verschiedener Disziplinen des behandelten Problembereichs gegeben. Der wesentliche Beitrag dieser Arbeit ist die Zusammenführung der Exploration mittels Topic Models mit Modellen der Kognitionspsychologie. Genauer zählt dazu die Identifizierung der Beschränkung momentaner Forschungsarbeiten, das spezifizieren wichtiger Aspekte der Exploration mit Topic Models und das Aufzeigen neuer Funktionen und Konzepte zur Unterstützung des Sensemaking.

Die Untersuchung wurde aufgegliedert in die Bereiche des Algorithmus, Interfaces und Sensemaking. Zu den Algorithmen konnte festgestellt werden, dass viele Arbeiten das Topic-Model vorab berechnen und anschließend einfache Filter und Visualisierungen auf die Ergebnisse anwenden. Jedoch wurden auch effiziente Verfahren für die Modifikation und Spezifizierung des trainierten Models gefunden. Dazu zählen etwa die Verwendung der Inference, das Zusammenfügen von Themen und die Detailanalyse von Subselektionen. Diese Verfahren sind notwendig für diesbezügliche interaktive Prozesse und können mit wenig Rechenaufwand umgesetzt werden. Im Bezug zum Interface wurden Fragen der Präsentation der Daten behandelt. Dazu zählte die Aufteilung und Verbundenheit verschiedener Perspektiven zum Zweck der Orientierung im Datenraum und die Minimierung der Informationslast. Beispielsweise konnte in Tiara eine dem tf-idf-Maß ähnliche Selektion für die Bezeichnung von Themen gefunden werden. Die Unterstützung der prozeduralen und der räumlichen Orientierung wurde in den Abschnitten des Sensemaking weiter ausgeführt. Des Weiteren wurde nach Methoden zum Vergleich von Daten und zur Externalisierung von Informationen gesucht. Die Leistung dieses Teils der Arbeit ist das Zusammentragen der verschiedenen Ansätze der Exploration von Textkorpora mit Topic Models. Eine weitere Ausführung könnte in ein klassifizierendes Framework führen, dass den Vergleich und das Verbinden von Forschungsarbeiten erleichtert.

Viele Ansätze versuchen dem Sensemaking durch eine Kombination aus Visualisierungen, Filtern und Dokumentempfehlung zu begegnen. Komplexere Suchvorgänge werden

bislang nicht unterstützt. Dazu zählt etwa die parallele Suche in verschiedenen Themensträngen, der Vergleich von Daten oder das Externalisieren neu gewonnenen Wissens. Die dazu notwendigen Algorithmen könnten in vielen Fällen aus benachbarten Disziplinen abgeleitet werden. Es ist daher nicht von einer technischen Beschränkung auszugehen, sondern von einem fehlenden Bewusstsein für die Anwendungsmöglichkeiten von Topic Models. Ein Beispiel bietet die Darstellung der Elementrelationen als Graph, die in den betrachteten Anwendungen vorwiegend statisch erfolgt. Verfahren der Graph-Analyse wie das Graph Matching, die Detailanalyse mit Magic Lenses und die Wegfindung mit dem $A*$-Algorithmus sind bekannt, aber werden bislang nicht genutzt. Die Mensch-Maschine Interaktion beschränkt sich momentan zu großen Teilen auf die Berechnung eines Modelles, das anschließend visualisiert und vom Menschen analysiert wird. Dieser formuliert daraufhin Filter, nach denen die Daten selektiert oder sortiert werden. Dem gegenüber könnte die Schemabildung eine höhere Ebene der Analyse eröffnen, in der ein Nutzer eigene Vorstellungen ausdrückt, die direkt in Form von Recommendations auf die Daten angewandt werden. Dadurch wird das Wechselspiel von Filtern und Betrachten direkter und dynamischer. Für die Unterstützung des Sensemaking fehlen tatsächlich noch Algorithmen, die die Möglichkeiten von Topic Models ausschöpfen. Ein Beispiel der genannten Beschränkung ist die Verwendung des Kosinus-Maß. Dies ist ein beliebtes Maß der Information Retrieval und liefert relativ verlässliche Empfehlungen in einfacher Form. Für eine interaktive Exploration müssen Angaben jedoch nicht so beschränkt sein, da auf weitere Details individuell eingegangen werden kann. Die Informationen die von einem Topic Model zur Verfügung gestellt werden, werden bislang noch nicht umfassend genutzt. Relevanter als die Bestimmung konkreter Funktionen ist auch hier die Schaffung eines Bewusstseins für die Anwendung von Topic Models für die direkte Exploration, die sich von den Methoden des automatischen Gebrauchs entfernen muss. Eine weitere und allgemein auf das Gebiet der Exploration zutreffende Aufgabe ist die nutzerzentrierte Evaluation solcher Funktionen. Da bislang keine verlässlichen quantitativen Verfahren existieren, bleiben die meisten Forschungsarbeiten ohne abschließende Bewertung. Folglich lassen sich ähnliche Funktionen schwer in einen Bezug setzen, wodurch es zu der Wiederholung gleicher Ansätze kommt. Die wiederholte Anwendung ähnlicher Funktionen und das immer wieder neue Erstellen von Explorationsumgebungen zeigen, dass ein Framework benötigt wird in das die einzelnen Tools eingeordnet werden können. Auch wenn ein Framework hier nicht gegeben werden konnte, wurde die Möglichkeit der Orientierung an Modellen des Sensemaking gegeben und eine Zusammenführung verschiedener Arbeiten eingeleitet. Erstaunlicherweise finden Modelle wie das von Pirolli et al. bislang kaum Beachtung, jedoch kann seit kurzem die Ausrichtung des Forschungsbereiches auf nutzerzentrierte Konzepte beobachtet werden. Dazu zählen beispielsweise das Sensema-

king Framework von Qu et al. [27], die neuartige Position eines Nutzers im Datenraum als Information Flaneur von Dork et al. [11] oder das Knowledge Generation Model von Sacha et al. [6].

Neben der Identifizierung der Entwicklungsrichtung des Forschungsbereiches wurden Vorschläge gemacht, wie einige der bislang ungelösten Beschränkungen überwunden werden könnten. Die dabei erzeugten Konzepte streben allesamt nach verallgemeinerten Funktionen für die Strukturierung von Daten. Dazu müssen die verschiedenen Elementtypen eines Topic Models (Thema, Dokument und Term) stärker vereint werden. Im Gegensatz zu automatischen Verfahren in denen endgültige Ergebnisse produziert werden sollen, können in der Exploration die verschiedenen Details interaktiv analysiert werden, um eine Vielzahl von Perspektiven auf die Daten zu eröffnen. Von der Vereinigung der Elementtypen werden neue Funktionen und ein tieferes Verständnis für die Zusammenhänge eines Dokumentkorpus erhofft. Da die Interpretation von Informationen auf individuellen Vorstellungen beruht, wurde das Umordnen des Datenraumes als ein wichtiger Faktor der sinnstiftenden Exploration herausgestellt. Es wurden mehrere Ebenen der Strukturierung und des Bildens eigener Strukturen beschrieben. Dazu gehört das Konzept zur automatischen Organisation von Relationsmatrizen aus Abschnitt 4.1, das die Übersicht verbessern soll und um manuelle Modifikationen ergänzt werden kann. Einen Schritt weiter geht die in Abschnitt 4.3 vorgestellte Formel, mit der sich Typen und ganze Elementgruppen vergleichen lassen. Eine Anwendung von Gruppen und eine Generalisierung der Recommendation konnte bislang für Topic Models nicht gefunden werden. Das höchste Ziel der Vereinigung von Exploration und Sensemaking ist das Bilden von Schemata und das Anwenden dieser Schemata zum Auffinden weiterer Daten. Dieses Ziel wurde mit dem in Abschnitt 4.4 vorgestellte Konzept behandelt. Durch die Verwendung zwei neuartiger Anwendungsformen für Topic Models und das Vector Space Model mag es sich daher etwas weit vor wagen. Es darf daher lediglich als experimenteller Ausblick gesehen werden, der sich als logische Ableitung aus gemachten Beobachtungen ergibt und nicht auf spezifische Fragen wie etwa der mathematischen Komplexität einer K-dimensionalen Interpolation eingeht. Die vorgestellten Konzepte wurden aus der Beobachtung heraus entwickelt, dass Externalisierungsprozesse noch weitgehend unerforscht sind und externalisierte Strukturen nicht auf die Daten zurückgeführt werden. An Abbildung 2.5 erklärt, wird bislang und auch nur in begrenztem Umfang, der Weg von links unten nach rechts oben gegangen, ohne die Wechselseitigkeit des Prozesses zu beachten. Daraus formte sich die Erkenntnis der Notwendigkeit, den foreaging loop als auch den sensemaking loop zu *schließen*.

5.2 Offene Fragen

Ein großer Teil dieser Arbeit besteht aus der Übersicht bestehender Ansätze, sodass neue Fragen gestellt wurden, die die Richtung der weiteren Forschung weisen sollen.

In der Zusammenführung in Abschnitt 3.3 wurden die Arbeiten nach einzelnen funktionalen Aspekten gegenüber gestellt. Die Strukturierung von Funktionen kann weiter ausgeführt werden und in einem Framework münden, das Forschungsarbeiten die sich auf detaillierte Probleme konzentrieren in einen Kontext setzt, der die Vereinigung mehrere Ansätze vereinfacht. Des Weiteren ließen sich Arbeiten von benachbarten Disziplinen einfacher übertragen, wenn der genaue Anwendungsbereich bekannt ist.

Neben dieser Meta-Behandlung des Forschungsbereiches werden auch anwendungsbezogene Probleme in Aussicht gestellt. Die Exploration von Textkorpora wird häufig als Zoom bezeichnet, der verschiedene Perspektiven und Abstraktionsebenen, von der Übersicht bis in Details umfasst. Die Konzeption einer entsprechenden Arbeitsumgebung ist weiterhin offen und müsste sich damit auseinandersetzen, verschiedene Funktionen zu vereinen um die momentan bestehende Redundanz der Interfaces zu beheben. Das Ziel könnte ein Editor sein, der wie Excel oder Photoshop von Laien als auch von Experten gleichermaßen verwendet werden kann. Dazu sind ein großer Funktionsumfang und hohe Freiheitsgrade notwendig, die durch die Vereinigung von Werkzeugen, Datentypen und Abstraktionsebenen entstehen und einen sehr differenzierten Anwendungsbereich ergeben. Dies sind Aufgabenstellungen der HCI und des ID, die aufzeigen in wie viele verschiedene Richtungen sich die aktuelle Forschung weiter entwickeln könnte.

Zum Sensemaking bleiben weiterhin viele Fragen offen, da dieser Bereich nicht nur in der Anwendung von Topic Models noch relativ neu ist. Bezüglich der Orientierung muss gelöst werden, wie verschiedene Themenstränge parallel behandelt werden können und wie sich verzweigte Suchverläufe verwenden lassen. In Betrachtung des Sensemakingprozesses stellt sich die Frage, wie dieser in einzelne Komponenten aufgeteilt werden könnte. Eine Unterteilung nach Visualisierung, Shoebox und Schema führt zu drei Interaktionsbereichen und steigert damit die Komplexität beträchtlich. Darüber hinaus ist auch noch nicht beantwortet, wie genau sich eine Explorationsumgebung an den menschlichen Informationsverarbeitungsprozessen orientieren sollte. In Abschnitt 2.2.2 wurde darauf hingewiesen, dass eine übermäßige Dokumentierung des Suchprozesses diesen behindert, anstatt ihm zu nutzen. Wie ließen sich demnach die verschiedenen Schritte verbinden, sodass der foreaging und der sensemaking loop in einem einzigen Prozess erfolgen? Da die Externalisierung in der Exploration noch relativ unerforscht ist, müssen dafür neue Wege der Interaktion gefunden werden. In dieser Behandlung werden sich dann wiederum Fragen des Grades der Automatisierung stellen, da die Hypothesen grundsätzlich vom Menschen formuliert werden sollen. Abgesehen von potentiellen Entwicklungen ist der Nutzen von Schemata

und Externalisierungen noch nicht bestätigt. Dieser ist bislang nur eine Hypothese, die sich vorwiegend aus kognitionspsychologischen Modellen ableitet.

Die in Abschnitt 4 vorgestellten Konzepte wurden aus den vorher gemachten Beobachtungen logisch hergeleitet, aber nicht auf Effizienz und Nutzen geprüft. Besonders die Schemabildung in Abschnitt 4.4 besteht aus mathematischen Operationen im Vector Space Model (VSM), die auf ihre Komplexität hin analysiert werden müssen. Die Gruppier- und Vergleichbarkeit beliebiger Elementtypen in Abschnitt 4.3 ist ein Schritt in die Vereinfachung des Workflows. Da die Methode auf der Relationsbestimmung durch das Kosinus-Maß basiert, ist von einer erfolgreichen Anwendung auszugehen. Es muss dennoch überprüft werden wie Sinnvoll die Vereinigung mehrerer Elemente in einen Vektor tatsächlich ist, also wie exakt die Resultate sind und wie sie von Nutzern interpretiert werden.

5.3 Aussicht

Menschen werden sich immer häufiger mit großen Dokumentmengen konfrontiert sehen, deren tieferen Inhalt sie erschließen müssen ohne zu wissen wo und wie sie mit der Suche beginnen sollen. Mit der Entwicklung effizienter Algorithmen und der stetig steigenden Rechenleistung lassen sich diese Probleme durch die Verbindung von interaktiver Visualisierung und maschineller Analyse lösen. Die Anwendung von Topic Models in der Exploration ist noch ein neuer Forschungsbereich von dem zu erwarten ist, dass er Funktionen hervorbringt, die mit einem neuen Verständnis der Exploration und des Sensemaking konzipiert werden. Die Zahl der Arbeiten die diese beiden Bereiche verbindet steigt seit kurzer Zeit, sodass ein Wandel der Forschungsrichtung abzusehen ist, der den bisherigen Schwerpunkt von den Algorithmen, hin zu nutzerzentrierten Betrachtungen verschiebt. Die bisherigen Explorationsstrukturen orientieren sich vorwiegend an der professionellen Verwendung durch geschulte Nutzer. Wegen der steigenden Zahl von Daten mit denen ein Mensch mittlerweile auch im privaten Bereich alltäglich konfrontiert wird, müssen in Zukunft entsprechende Suchverfahren auch für ungeübte Anwender entwickelt werden.

Neben der lokalen Datenanalyse wird mittlerweile am kollaborativen Sensemaking geforscht, dass die Hypothesenbildung in geschlossenen Arbeitsgemeinschaften, oder öffentlich im Web behandelt. In dieser Hinsicht ist eine Verbindung mit Bereichen des *social webs* und der *collaborative knowledge* möglich, wodurch die Stärken der Automatismen von Topic Models mit Metadaten aus *Folksonomies* vereint werden könnten.

Durch den Zuwachs des Forschungsbereiches wird sich in kurzer Zeit zwangsläufig ein Framework entwickeln, in dem sich die bisherigen Erkenntnisse bündeln. Dazu muss der Explorationsprozess weiter konzeptionell und in der Praxis analysiert und verstanden

werden. Es ist absehbar, dass die Zahl der nutzerzentrierten Evaluationen steigen und die neue Anwendungsbereiche aufdeckt werden. Es ist jetzt schon ersichtlich, dass sich Explorationsumgebungen nicht nur auf die Visualisierung und Filterung von Informationen verlassen dürfen, sondern verstärkt die hypothesen- und Schemabildung unterstützen. Die weitere Erforschung der Möglichkeiten von Topic Models kann neue Funktionen hervorbringen, durch die sich Mensch und Maschine effizienter ergänzen.

Literaturverzeichnis

[1] P. Pirolli and D. M. Russell, "Introduction to this special issue on sensemaking," *Human–Computer Interaction*, vol. 26, no. 1-2, pp. 1–8, 2011.

[2] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," in *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 159–168, ACM, 1998.

[3] D. M. Blei, "Introduction to probablistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[4] A. J.-B. Chaney and D. M. Blei, "Visualizing topic models.," in *ICWSM*, 2012.

[5] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith, "Interactive topic modeling," *Machine Learning*, vol. 95, no. 3, pp. 423–469, 2014.

[6] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. Keim, "Knowledge generation model for visual analytics," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, pp. 1604–1613, Dec 2014.

[7] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card, "The cost structure of sensemaking," in *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pp. 269–276, ACM, 1993.

[8] W. Stock, *Information retrieval: Informationen suchen und finden.* Einführung in die Informationswissenschaft, Oldenbourg, 2007.

[9] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[10] P. E. R. Ingwersen, *Information Retrieval Interaction.* Taylor Graham, 1992.

[11] M. Dörk, S. Carpendale, and C. Williamson, "The information flaneur: A fresh look at information seeking," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1215–1224, ACM, 2011.

[12] R. Rogers, *Digital methods*. MIT Press, 2013.

[13] K. Carstensen, C. Ebert, C. Ebert, S. Jekat, H. Langer, and R. Klabunde, *Computerlinguistik und Sprachtechnologie*. Spektrum Lehrbuch, Spektrum Akademischer Verlag GmbH, 2010.

[14] R. Ferber, *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. dpunkt-Verlag, 2003.

[15] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[17] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 178–185, ACM, 2006.

[18] G. Heyer, U. Quasthoff, and T. Wittig, *Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse*. IT lernen, W3L-Verlag, 2006.

[19] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in neural information processing systems*, pp. 288–296, 2009.

[20] R. C. Atkinson and R. M. Shiffrin, "Human memory: A proposed system and its control processes," *Psychology of learning and motivation*, vol. 2, pp. 89–195, 1968.

[21] A. Baddeley, "Working memory," *Science*, vol. 255, no. 5044, pp. 556–559, 1992.

[22] K. Sokolowski, *Allgemeine Psychologie für Studium und Beruf*. Pearson Studium, 1. aufl. ed., 2013.

[23] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information.," *Psychological review*, vol. 63, no. 2, p. 81, 1956.

[24] W. Prinz, *Wahrnehmung und Tätigkeitssteuerung*. Springer, 1983.

[25] N. M. Seel and D. Ifenthaler, *Psychologie des Lernens*. UTB, Stuttgart, 2. aufl. ed., 2003.

[26] R. C. Anderson, "Role of the reader's schema in comprehension, learning, and memory," *Learning to read in American schools: Basal readers and content texts*, vol. 29, pp. 243–257, 1984.

[27] Y. Qu and G. W. Furnas, "Model-driven formative evaluation of exploratory search: A study under a sensemaking framework," *Information Processing & Management*, vol. 44, no. 2, pp. 534–555, 2008.

[28] B. Dervin and P. Dewdney, "Neutral questioning: A new approach to the reference interview," *Rq*, pp. 506–513, 1986.

[29] G. Klein, B. M. Moon, and R. R. Hoffman, "Making sense of sensemaking 1: Alternative perspectives.," *IEEE intelligent systems*, vol. 21, no. 4, pp. 70–73, 2006.

[30] G. Klein, B. Moon, and R. R. Hoffman, "Making sense of sensemaking 2: A macrocognitive model," *Intelligent Systems, IEEE*, vol. 21, no. 5, pp. 88–92, 2006.

[31] Y. B. Shrinivasan and J. J. van Wijk, "Supporting the analytical reasoning process in information visualization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1237–1246, ACM, 2008.

[32] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for using multiple views in information visualization," in *Proceedings of the working conference on Advanced visual interfaces*, pp. 110–119, ACM, 2000.

[33] M. Dörk, S. Carpendale, and C. Williamson, "Edgemaps: Visualizing explicit and implicit relations," in *IS&T/SPIE Electronic Imaging*, pp. 78680G–78680G, International Society for Optics and Photonics, 2011.

[34] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining.," *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, vol. 20, no. 1, pp. 19–62, 2005.

[35] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner, "Visual analysis of large graphs: State-of-the-art and future research challenges," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1719–1749, 2011.

[36] A.-S. Dadzie and M. Rowe, "Approaches to visualising linked data: A survey," *Semantic Web*, vol. 2, no. 2, pp. 89–124, 2011.

[37] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher, "Serendip: Topic model-driven visual exploration of text corpora," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, 2014.

[38] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization techniques for assessing textual topic models," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 74–77, ACM, 2012.

[39] J. Stasko, C. Görg, and Z. Liu, "Sensemaking across text documents: human-centered, visual exploration with jigsaw," in *Sensemaking Workshop@ CHI 2008*, 2008.

[40] C. Gorg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko, "Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 10, pp. 1646–1663, 2013.

[41] C. Görg, Z. Liu, and J. Stasko, "Reflections on the evolution of the jigsaw visual analytics system," *Information Visualization*, p. 1473871613495674, 2013.

[42] J. Chuang, D. Ramage, C. Manning, and J. Heer, "Interpretation and trust: Designing model-driven visualizations for text analysis," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 443–452, ACM, 2012.

[43] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park, "ivisclustering: An interactive visual document clustering via topic modeling," in *Computer Graphics Forum*, vol. 31, pp. 1155–1164, Wiley Online Library, 2012.

[44] N. Günnemann, M. Derntl, R. Klamma, and M. Jarke, "An interactive system for visual analytics of dynamic topic models," *Datenbank-Spektrum*, vol. 13, no. 3, pp. 213–223, 2013.

[45] S. Osiński and D. Weiss, "Carrot2: Design of a flexible and efficient web information retrieval framework," in *Advances in Web Intelligence*, pp. 439–444, Springer, 2005.

[46] J. Eisenstein, D. H. Chau, A. Kittur, E. P. Xing, *et al.*, "Topicviz: Semantic navigation of document collections," *arXiv preprint arXiv:1110.6200*, 2011.

[47] J. Eisenstein, D. H. Chau, A. Kittur, and E. Xing, "Topicviz: Interactive topic exploration in document collections," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pp. 2177–2182, ACM, 2012.

[48] J. S. Yi, R. Melton, J. Stasko, and J. A. Jacko, "Dust & magnet: multivariate information visualization using a magnet metaphor," *Information Visualization*, vol. 4, no. 4, pp. 239–256, 2005.

[49] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang, "Tiara: a visual exploratory text analytic system," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 153–162, ACM, 2010.

[50] A. Smith, T. Hawes, and M. Myers, "Hiérarchie: Interactive visualization for hierarchical topic models," in *Proceedings of the Workshop on Interactive Language Learning, Visualization and Interfaces*, pp. 71–78, Association for Computational Linguistics, DECISIVE ANALYTICS Corporation, 2014.

[51] J. Chuang, Y. Hu, A. Jin, J. D. Wilkerson, D. A. McFarland, C. D. Manning, and J. Heer, "Document exploration with topic modeling: Designing interactive visualizations to support effective analysis workflows," in *NIPS Workshop on Topic Models: Computation, Application, and Evaluation*, 2013.

[52] J. Chuang, S. Gupta, C. Manning, and J. Heer, "Topic model diagnostics: Assessing domain relevance via topical alignment," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 612–620, 2013.

[53] L. AlSumait, D. Barbará, J. Gentle, and C. Domeniconi, "Topic significance ranking of lda generative models," in *Machine Learning and Knowledge Discovery in Databases*, pp. 67–82, Springer, 2009.

[54] D. Archambault, T. Munzner, and D. Auber, "Grouseflocks: Steerable exploration of graph hierarchy space," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 4, pp. 900–913, 2008.

[55] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[56] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2011.

Anhang

Strukturieren von Relationsmatrizen

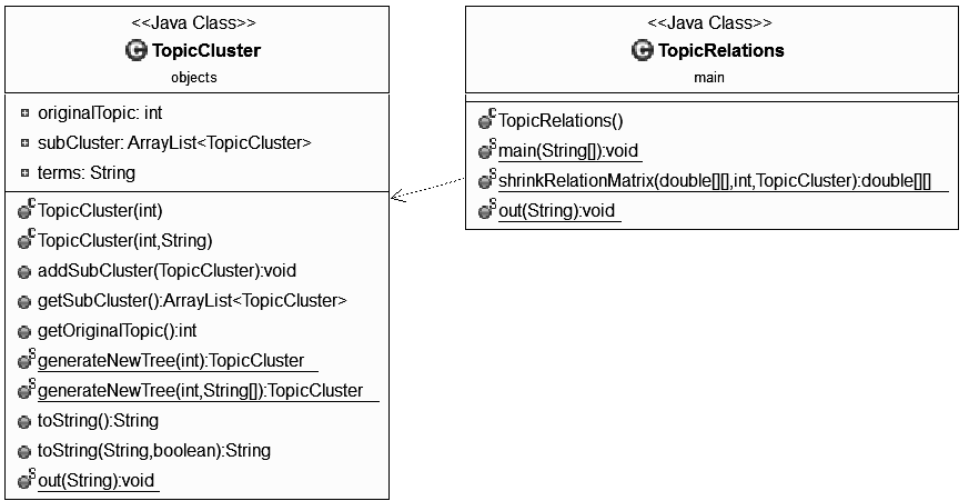


Abbildung 5.1: UML-Klassendiagramm der in Liste 5.1 verwendeten Klassen. Generiert mit *Eclipse - ObjectAid UML Explorer*.

Listing 5.1: Reduktion einer symmetrischen Matrix *sigma* mit hierarchischer verschachtelung.

```
1  public static double[][] shrinkRelationMatrix(double[][] sigma, int k, TopicCluster
       topicClusterTree) {
2    if (sigma.length <= k)
3      return sigma;
4
5    // Find highest relation
6    double max = 0;
7    int p1 = 0, p2 = 0;
8    for (int x = 0; x < sigma.length; x++)
9      for (int y = x + 1; y < sigma[x].length; y++)
10       if (sigma[x][y] > max) {
11         max = sigma[x][y];
12         p1 = x;
13         p2 = y;
14       }
15
```

```
16    // Reorganize topicClusterTree
17    ArrayList<TopicCluster> currentCluster = topicClusterTree.getSubCluster();
18    TopicCluster tcSub = currentCluster.get(p2);
19    currentCluster.get(p1).addSubCluster(tcSub);
20    currentCluster.remove(p2);
21
22    // Create new Array and copy old values
23    double[][] mO = new double[sigma.length - 1][sigma[0].length - 1];
24    int jX = 0;
25    for (int x = 0; x < sigma.length; x++) {
26      int jY = 0;
27      if (x != p2){
28        for (int y = 0; y < sigma[x].length; y++)
29          if (y != p2) {
30            mO[jX][jY] = mO[jY][jX] = sigma[x][y];
31            jY++;
32        jX++;
33      }
34    }
35
36    // Calculate conjunction
37    int j = 0;
38    for (int i = 0; i < sigma.length; i++)
39      if (i != p2) {
40        if (i != p1)
41          mO[j][p1] = mO[p1][j] = (sigma[i][p1] + sigma[i][p2]) / 2.0;
42        j++;
43      }
44    return shrinkRelationMatrix(mO, k, topicClusterTree);
45  }
```

Listing 5.2: Hierarchische Gruppierung von Themen über einen Dokumentkorpus von drei inhaltlich verschiedenen Quellen.

```
1  \-root
2   |-0: wikipedia articles 0 categories article
3   | |-9: yago facts entities i y
4   | |-2: the of and to a
5   | | |-3: topic document documents topics text
6   | | |-18: topic model we topics word
7   | | |-13: visualization user view views design
8   | | \-14: information search visual visualization and
9   | \-6: user dbpedia context type dbpo
10  |-1: genie vetter geist arnheims geistes
11  | |-7: ordnung stadt kakanien kultur schmeisser
12  | | \-19: leinsdorf tuzzi rachel graf erlaucht
13  | |-4: man so fuer nur noch
14  | | |-8: war ihn ihm noch um
15  | | |-17: waren ulrich zwei lieben insel
16  | | |-15: ciarisse moosbrugger meingast friedenthal walter
17  | | |-16: mir habe hat mich dir
18  | | |-11: lindner schwester hagauer bruder agathe
19  | | \-12: gefuehl liebe zustand gefuehls gefuehle
20  | \-5: gerda fischel hans leo geld
21  \-10: gott gegen hat man macht
```

Erzeugen von Gewichtungsvektoren

Die Generierung von Gewichtungsvektoren über Themen wird in Abschnitt 4.3 behandelt. Für Erzeugung von Gewichtungsvektoren für Terme ergibt sich ein komplizierterer Fall, da die LDA ein Thema über eine zufällige Menge von Begriffen bildet. Es gibt daher nicht direkt eine Themen-Term-Relationsmatrix ϕ, sondern lediglich die Term Frequenzen für die einzelnen Themen. In Programmbibliotheken wie *MALLET* lässt sich ϕ daher auch nicht ausgeben, kann allerdings aus den Frequenzen abgeleitet werden. Ebendies wird nun beschrieben.

Ähnlich wie Gorg et al. in *Jigsaw* Terme ausfiltern, die in weniger als drei Dokumenten vorkommen [40], werden hier alle Terme ausgefiltert, die nicht mehr als zwei Mal in mindestens einem Thema vorkommen. Die Reduktionsrate verdeutlicht sich am folgenden Beispiel. Ein Alphabet der Größe 70000 und die Verwendung von 50 Themen ergibt eine Matrixgröße ϕ von 3500000. Durch die beschriebenen Reduktionsmaßnahmen und die Verwendung einer *HashMap* kann die Menge der Einträge auf ~ 40000, also um den Faktor 87.5 reduziert werden. Da die Werte bis dahin lediglich als Termfrequenzen $tf_{i,j}$ des Terms i in Thema j vorliegen, müssen sie in relative Wahrscheinlichkeiten überführt werden. Dazu werden sie normiert und in das Intervall $[0, 1]$ umgerechnet. Ein in der IR häufig verwendetes Maß der Relevanz eines Terms für ein Dokument ist das *Tf-idf-Maß*, das sich auch auf die Relevanz eines Terms für ein Thema übertragen lässt. Das *Tf-idf-Maß* wurde von Sparck Jones das erste Mal beschrieben [55].[1] Das Gewicht $W_{i,j}$ des Terms i in Thema j ergibt sich wie folgt:

$$W_{i,j} = tf_{i,j} \cdot idf_i = tf_{i,j} \cdot log\frac{K}{k_i} \tag{5.1}$$

Die $tf_{i,j}$-Werte werden bereits von der LDA gegeben, daher muss nur noch k_i bestimmt werden. k_i gibt an in wie vielen Themen der Term i vorkommt. Da für $k_i \ll K$ folgt, dass $idf_i > 1$, müssen alle $W_{i,j}$ normalisiert werden.

Mit $\phi_{i,j} = W_{i,j}$ ergibt sich aus W die Themen-Term-Relationsmatrix ϕ. Dieser kann mit ϕ_i einfach der Gewichtungsvektor eines Terms für alle Themen entnommen werden.

[1] Es findet sich eine ausführliche Beschreibung von Rajaraman et al. die hier verwendet wurde [56].

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass die vorliegende Arbeit von mir selbstständig, ohne fremde Hilfe angefertigt worden ist. Inhalte und Passagen, die aus fremden Quellen stammen und direkt oder indirekt übernommen worden sind, wurden als solche kenntlich gemacht. Ferner versichere ich, dass ich keine andere, außer der im Literaturverzeichnis angegebenen Literatur verwendet habe. Diese Versicherung bezieht sich sowohl auf Textinhalte sowie alle enthaltenden Abbildungen, Skizzen und Tabellen. Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Köln, den 20. April 2015

Lennart Renker