INSTITUT FÜR INFORMATIK
Datenbanken und Informationssysteme

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Analyse von Online-Partizipationsverfahren: Themenextraktion, Visualisierung und Interaktion

Philipp Grawe

Masterarbeit

Beginn der Arbeit:	16. April 2018
Abgabe der Arbeit:	16. Oktober 2018
Gutachter:	Prof. Dr. Stefan Conrad
	Prof. Dr. Martin Mauve

Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 16. Oktober 2018

Philipp Grawe

Zusammenfassung

Online-Partizipation nutzt das Internet, um Menschen über Entscheidungen diskutieren zu lassen und ihnen die Möglichkeit der Teilhabe am Prozess zu gewähren. Am häufigsten wird dieses Format in der Politik genutzt, besonders in der Kommunalpolitik, um Bürgerinnen und Bürgern Partizipation zu ermöglichen. Dadurch soll die Akzeptanz der Entscheidungen gestärkt und die Prozesse durch mehr Blickwinkel bereichert werden.
In dieser Arbeit sollen verschiedene Methoden zur Analyse solcher Online-Partizipationsverfahren vorgestellt werden, die dabei helfen sollen Verfahren zu überblicken und mit ihrem Ergebnis zu arbeiten. Dazu werden Methoden zur Themenextraktion, Visualisierung und Interaktion besprochen.
Als Grundlage dieser Methoden dient das Natural Language Processing, auf dessen Grundlagen kurz eingegangen wird. Auch einige Probleme, die dabei im Zusammenhang mit Online-Partizipationsverfahren auftreten werden erörtert, wobei hier besonders die Named Entity Recognition zu nennen ist.
Bei der Themenextraktion werden neben den bekannten Methoden wie Latent Dirichlet Allocation, Non-negative matrix factorization oder Latent Semantic Indexing, auch Interaktive Methoden oder ein Verfahren zur Stabilitätsbeschreibung dieser Themen vorgestellt. Durch die stark subjektive Wahrnehmung, ob Themen eine Dokumentenmenge korrekt abbilden, ist ein Vergleich schwierig.
Zur Visualisierung werden verschiedene Verfahren vorgestellt, indes Metadaten zu visualisieren deutlich besser möglich ist, als extrahierte Themen. Um extrahierte Themen zu visualisieren, bieten sich besonders Wortlisten oder Visualisierungen, die Wortlisten integrieren an.
Interaktion kann viele Facetten haben und geht häufig von einem Informationsbedürfnis aus. Diesem kann am besten mit Methoden des Information Retrieval begegnet werden, wobei hier Eigenschaften extrahierter Themen verwendet werden können. Außerdem kann eine Suche auf extrahierten Themen oder Visualisierungen aufbauen.
Zur Verwendung mit einem laufenden oder abgeschlossenen Verfahren wurde eine Webanwendung entwickelt, die unabhängig verwendet, oder in vorhandene Websites eingebunden werden kann. Diese Webanwendung verwendet Docker und implementiert Methoden zur Themenextraktion, Visualisierung und Interaktion.

Inhaltsverzeichnis

1 Einleitung

In diesem Kapitel werden das Themenfeld und die Motivation dieser Arbeit erläutert. Abschließend werden die für Arbeit gesetzten Ziele vorgebracht.

1.1 Motivation

Um Menschen an Entscheidungen oder Prozessen teilhaben zu lassen, bietet sich die Möglichkeit online Plattformen anzubieten, auf denen Diskussionen stattfinden, Vorschläge eingereicht und über eben diese abgestimmt werden kann. Genutzt werden diese Möglichkeiten am prominentesten in der Kommunalpolitik, wo Bürgerinnen und Bürgern online an Themen wie z.B. Bürgerhaushalten teilhaben können. Aber auch Institutionen und Firmen setzen Online-Partizipation ein, um Menschen zu beteiligen. Die Teilhabe von Betroffenen oder Bürgern an Entscheidungen schafft, auch durch die Bereitstellung im Internet, Transparenz und Vertrauen. Vorschläge können neue, bisher nicht bedachte, Sichtweisen offenbaren und Abstimmungen dieser Vorschläge ein unverbindliches Stimmungsbild liefern. Durch die so entstandene Partizipation wird erhofft, mehr Blickwinkel der Betroffenen bei der Problemlösung einzubeziehen und eine höhere Akzeptanz der Entscheidungen zu erzielen.
Konkret heißt dies meist, dass Teilnehmer Vorschläge erstellen, kommentieren und zustimmend bzw. ablehnend bewerten können. Ein Beispiel eines Vorschlages findet sich in Abbildung 1.

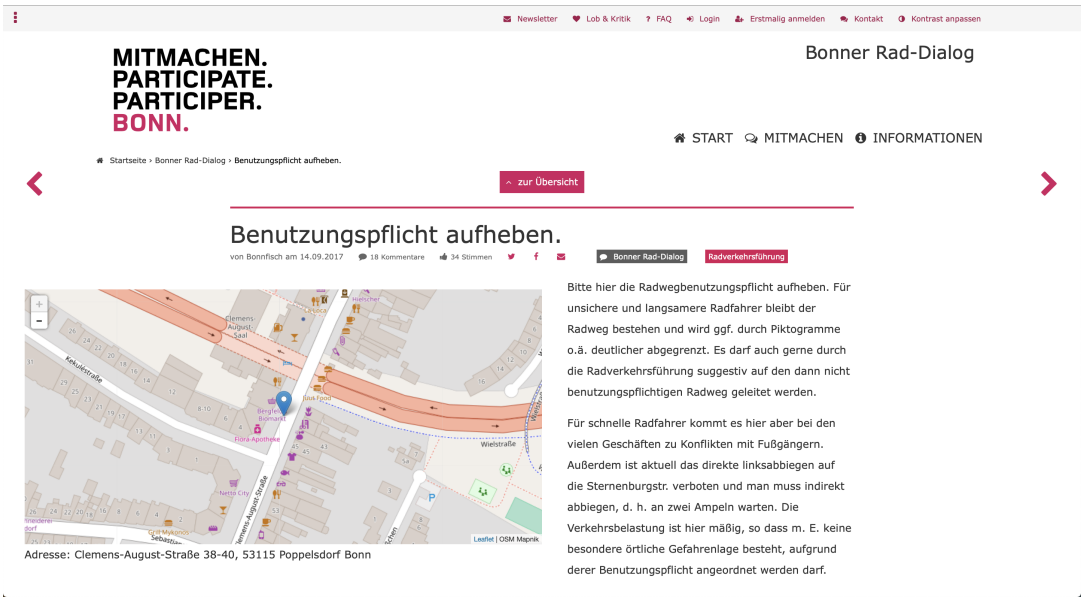


Abbildung 1: Screenshot eines Vorschlages des Bonner Rad-Dialoges.

Außerdem gibt es Verfahren bei denen Bürgerinnen und Bürger über Vorschläge der Verwaltung diskutieren und abstimmen können. Insgesamt gibt es sowohl auf einen Zeitraum begrenzte Verfahren wie z.B. Bürgerhaushalte, als auch dauerhafte

Plattformen,wie z.B. Mängelmelder für den kommunalen Raum. Meistens haben die Online-Partizipationsverfahren konkrete Themen als Bezug, beispielsweise der Verwendung des Tempelhoferfelds in Berlin.
Da die Verfahren oft einem großen Personenkreis offenstehen und nicht selten auf viel Engagement stoßen, kann so eine erhebliche Anzahl von Vorschlägen und Textbeiträgen entstehen. Eine manuelle Auswertung all dieser Beiträge ist mit einem nicht zu unterschätzenden Ressourcenaufwand verbunden. Deshalb bietet sich hier eine maschinell unterstütze Analyse an.
Weitere Hintergründe und eine gute Übersicht zum Einsatz von maschinellen Analyseverfahren bei Online-Partizipationsverfahren bieten Liebeck et al. (2017).
Diese Thematik wird viel diskutiert und erforscht, z.B. durch das *NRW-Fortschrittskolleg Online-Partizipation* welches auch Veranstaltungen abhält die den Austausch von Forschung und Praxis, also der Verwaltung, fördern soll. Dieses Jahr fand dazu das Praxissymposium „Online-Partizipation in Kommunen“ 2018 statt.

1.2 Zielsetzung

Ziel dieser Arbeit ist, verschiedene Aspekte von Online-Partizipationsverfahren maschinell zu analysieren. Dabei werden neben der Themenextraktion und Visualisierung extrahierter Themen und anderer Analysedaten auch eine auf Online-Partizipationsverfahren ausgerichtete Interaktion von Benutzern mit den durch die Analyse gewonnenen Daten betrachtet. All dies soll in einer Form zur Verfügung gestellt werden, die kein Wissen über die Funktionsweise der Verfahren benötigt, sondern sowohl für jeden Bürger als auch die Verwaltung eine verständliche und praktische Hilfe bei der Benutzung oder Betreuung von Online-Patrizipationsverfahren bietet.
Aus den gewonnen Erkenntnissen zu den verschieden Methoden wurde eine Webanwendung entwickelt, die dazu genutzt werden kann eine Analyse eines laufenden oder bereits beendeten Verfahrens vorzunehmen. Dabei lässt es sich sowohl in ein bestehendes System eingliedern als auch unabhängig betreiben. Diese Eingliederung kann über verschiedene Web-Techniken wie AJAX oder IFrames erfolgen. Denkbar ist auch eine Ergänzung von Funktionen, die über den beschriebenen Funktionsumfang hinausgehen. Somit kann diese Anwendung ein Startpunkt eines größeren Toolkits darstellen.
Die Verschiedenen Punkte haben dabei die folgende Zielsetzung:

1. Themenextraktion:
   Betrachtet werden Verfahren, die aus den Textbeiträgen Themen extrahieren. Dadurch werden nicht nur die Themen selbst berechnet, sondern auch eine Themenverteilung für jeden Vorschlag. Im Laufe dieser Arbeit werden verschiedene Verfahren vorgestellt und die Verwendung mit Online-Partizipationen hinterfragt.

2. Visualisierung:
   Um ein Online-Partizipationsverfahren besser zu verstehen, können verschiedene Aspekte der Analyse visualisiert werden. Dabei sollen sowohl die gefundenen Themen visualisiert werden als auch die erzeugten mathematischen Modelle zur Visualisierung von Zusammenhängen genutzt werden. Diese Zusammenhänge können zwischen Wörtern, Themen und Beiträgen bestehen. Generell kann eine Visualisierung

einen guten Überblick über Zusammenhänge bieten, aber auch zusammenfassen und Daten verknüpfen.

3. Interaktion:
   Auch die Interaktion kann von verschiedenen Seiten betrachtet werden. Die einfachste Form ist in diesem Zusammenhang eine Interaktion, in der vom Benutzer verschiedene Verfahren zur Themenextraktion ausgewählt und angewendet werden können. Bei den in dieser Arbeit benutzen Verfahren soll dazu ebenfalls eine Anzahl von Themen, die extrahiert werden sollen, angegeben werden. Der Benutzer kann also interaktiv bestimmen wie viele Themen extrahiert werden. Dies ist auch deshalb von Vorteil ist, da ein Mensch schnell über den Sinngehalt der gefundenen Themen entscheiden kann. So kann z.B. kann ein Mitglied der Verwaltung eine oder mehrere Themenanzahlen auswählen, die allen Benutzern angezeigt werden. Allerdings stellt eine auf den extrahierten Themen beruhende Suche ebenso eine Interaktion dar, die auch dazu genutzt werden kann ähnliche Beiträge (im Sinne der erkannten Themen) vorzuschlagen. Die Suche selbst sowie die Visualisierung können interaktiv gestaltet werden, was zur Zufriedenheit des Benutzers beiträgt.

1.3 Gliederung der Arbeit

Im nächsten Kapitel folgen die Grundlagen, die nötig sind um die Verfahren zu verwenden. Dabei wird auch auf die Eigenheiten von Online-Partizipationsverfahren eingegangen.
In Kapitel 4 werden Verfahren zur Themenextraktion und dessen Ergebnisse bei Online-Partitipationsverfahren besprochen. Da ein Teil der Interaktion das Information Retrieval darstellt, erklärt Kapitel 3 die theoretischen Grundlagen. Die Grundlagen der Visualisierung stellt Kapitel 5 vor. Welche dieser Techniken wie verwendet wurde und wie diese zusammen spielen, erklärt das Kapitel 6. Darin wird sowohl auf Details der Implementierung als auch darauf eingegangen, als auch Beispiele aus der Anwendung gezeigt. Auf die Interaktion wird verstärkt im Kapitel 6.6 eingegangen. Abschließend wird ein Fazit gezogen.

2 Grundlagen

Im folgenden Kapitel werden einige Grundlagen behandelt, die notwendig sind, um eine Analyse der Verfahren vorzunehmen. Dabei wird auf die Datengrundlage eingegangen sowie auf Natural Language Processing und Information Retrival.

2.1 Gewinnung der Daten von Online-Partizipationsverfahren

Um die verwendeten Analysemethoden zu testen und vergleichen, wurden Daten von beendeten und teilweise, zum Zeitpunkt der Erstellung dieser Arbeit, noch laufenden Verfahren verwendet. Somit war eine von den Betreibern des Verfahrens unabhängige Erstellung dieser Masterarbeit und Entwicklung der Anwendung möglich. Da die Daten öffentlich zugänglichen sind, wurden diese mit Hilfe von sogenannten *Webcrawlern* erlangt. Diese Programme laden automatisch alle gewünschten Informationen einer Website herunter und werden deshalb auch bei Suchmaschinen eingesetzt. Der Einsatz von *Webcrawlern* hat mehrere Vorteile. Zum einen kann der gesamte Prozess der Datenverarbeitung gesteuert werden, sodass die gewünschten Daten in dem benötigten Format erhalten werden. Folgenderweise sind die Daten für jedes Verfahren einheitlich, was die Vergleichbarkeit erhöht. Zum anderen werden so datenschutzrechtliche Bedenken hinsichtlich der Weitergabe von Inhalten, die Benutzer erstellt haben, ausgeräumt.
Implementiert wurden die *Webcrawler* von Max Schubert, Markus Brenneis, Matthias Liebeck und mir. Dies geschah im Rahmen der Projektarbeit des Masterstudiums. Zudem wurden die Daten sowohl für Masterarbeiten als auch Dissertationen wie z.B. Liebeck (2018) verwendet, die sich verschiedenen Fragestellungen rund um das Thema Online-Partizipationsverfahren widmen.
Das Projekt mit den Webcrawlern ist öffentlich auf *GitHub*[1] verfügbar. Die Veröffentlichung dieser Webcrawler verbessert so auch den Austausch und die Vergleichbarkeit von Forschungsergebnissen.
Tabelle 1 bietet eine Übersicht über Verfahren, die im Rahmen dieses Projektes bereits implementiert wurden. Der Schwerpunkt liegt dabei auf kommunalen Verfahren in Nordrhein-Westfalen. Von diesen Verfahren wurden alle gemachten Vorschläge gesammelt. Meistens ist es Bürgern möglich die Vorschläge zu kommentieren und zustimmend bzw. ablehnend zu bewerten. Neben diesen Kommentaren und Bewertungen wurden auch Metadaten wie der Benutzername und das Datum des Verfassens gespeichert. Bei einigen Verfahren gab es ebenfalls die Möglichkeit den Vorschlag mit einer Adresse zu versehen. Adressen machen vor allem bei Verfahren wie den Raddialogen oder einem Mängelmelder Sinn. Insgesamt wurde darauf geachtet, dass man die Verfahren aus den gewonnenen Daten wiederherstellen könnte, sie also auch in ihrer Struktur vollständig sind.
Am Ende des Prozesses erhält der Anwender eine *JSON*-Daten, was ein für diese Art der Anwendung übliches und leicht weiterzuverarbeitendes Datenformat darstellt.

[1] https://github.com/Liebeck/OnlineParticipationDatasets

Name des Verfahrens	Vorschläge	Kommentare
Bonn 2011 †	1015	8903
Bonn 2015/2016	335	2937
Bonn 2017/2018	55	109
Bonn 2019/2020		
Braunkohle †	7	1296
Bürgerbudget Wuppertal	261	
Köln 2012	594	1879
Köln 2013	592	3095
Köln 2015	631	1855
Köln 2016	827	1314
Leitbild Bad Godesberg	556	698
Mängelmelder Braunschweig	$\geq$ 3220	
Raddialog Bonn	2331	2425
Raddialog Koeln-Ehrenfeld	378	277
Raddialog Moers	463	300

Tabelle 1: Bei Erstellung der Arbeit zur Verfügung stehende Online-Partizipationsverfahren. †: Diese Verfahren sind mittlerweile nicht mehr online Verfügbar.

2.2 Natural Language Processing

Natural Language Processing (NLP) ist ein wichtiger Baustein für die Analyse von Texten, also auch Online-Partizipationsverfahren. Im Folgenden wird zunächst NLP allgemein erklärt und danach auf einige Besonderheiten im Zusammenhang mit Online-Partizipationsverfahren eingegangen. Damit eine maschinelle Verarbeitung natürlicher Sprache möglich ist, müssen verschiedene Modelle auf die Sprache angewandt und eine einheitliche Darstellung gewählt werden. Diese Modelle sollen die Eingabe so strukturieren, dass damit eine statistische Auswertung möglich ist (Manning und Schütze, 1999). Zu beachten ist dabei, dass diese Modelle nicht in der Lage sind natürliche Sprache in ihrer Komplexität, Grammatik und Syntax zu verstehen. Ein Regelwerk zu erstellen, welches ein Computer verarbeiten kann, ist so nicht möglich. Auch müssen die Kreativität und Veränderungen der Sprache bedacht werden. Statistische Modelle sind oft nur eine Annäherung.
Dabei wird die Verarbeitung in Teilprobleme zerlegt, die nacheinander abgearbeitet werden. In diesem Zusammenhang wird auch von einer Pipeline gesprochen. Dabei verwenden die einzelnen Verarbeitungsschritte jeweils Ergebnisse vorheriger Schritte. So kann nicht nur Konsistenz sichergestellt werden, sondern es wird auch die Möglichkeit erhalten Zwischenergebnisse zu betrachten und gegebenenfalls zu speichern. Im Folgenden orientiert sich diese Arbeit an der Pipeline von Conrad (2017), wobei im Laufe dieses Abschnitts jedoch lediglich die für diese Arbeit notwendigen Verarbeitungsschritte betrachtet werden. Abbildung 2 skizziert dabei eine solche Pipeline, enthält aber nur die für diese Arbeit relevanten NLP Schritte. Die zusätzlichen Verarbeitungen erfolgen an der Stelle, die durch drei Punkte dargestellt wurde.

Abbildung 2: Schaubild einer Natural Language Processing Pipeline.

Verwendung finden in dieser Arbeit die Schritte: Tokenisierung, Part-Of-Speech Tagging, Lemmatisierung, Stoppworteliminierung sowie Named Entity Recognition. Diese werden im Rest des Kapitels erläutert. Das Erkennen der Sprache erübrigt sich, da man die Sprache des Online-Partizipationsverfahren kennt. Sollten innerhalb dieses jedoch mehrere Sprachen zum Einsatz kommen, böte sich eine Spracherkennung an. Dependenzen zwischen Wörtern werden für die eingesetzten Methoden nicht benötigt. Die Erkennung der Tonalität könnte eine sinnvolle Erweiterung des Anwendungsumfangs sein, siehe dazu Kapitel 7.2.

2.2.1 Tokenisierung

Die Eingabe des Textes ist zu Beginn der Pipeline nicht mehr als eine zusammenhängende Zeichenkette. Zur weiteren Verarbeitung gilt es nun, diese in sinnvolle Segmente zu unterteilen. Diese Segmente können auf Wortebene oder Satzebene gefunden werden, aber z.B. auch Silben oder Absätze sein. Segmente auf Wortebene werden auch Token genannt. Dabei können diese Token auch aus mehreren Wörtern bestehen, was beispielsweise bei Namen der Fall ist. Auch wenn dieser Schritt trivial erscheinen mag, bilden eben diese Token die Grundlage für die weitere Verarbeitung. Ein möglichst genaues Segmentieren ist daher wünschenswert.
In der vorliegenden Arbeit ist die Segmentierung auf Wortebene besonders wichtig. Der naive Ansatz ist, den Text an den Leerzeichen und Interpunktion zu zerteilen und so zu segmentieren. Dieses Vorgehen ist im Deutschen oder Englischen ein einigermaßen guter Ansatz und kann dann durch regelbasierte Algorithmen verbessert werden, wie etwa im Patent von Bennett (2011). Der Algorithmus beinhaltet sowohl sprachspezifische Daten, die zuvor erstellt wurden, als auch einen hierarchischen Ansatz, der verfolgt wird bis die Zeichenkette segmentiert wurde. Verwendet werden bei der Tokenisierung auch Wörterbücher, dessen Einsatz sich bewährt hat (Grefenstette und Tapanainen (1994)). Damit können unter anderem Abkürzungen nachgeschlagen werden.

2.2.2 Part-Of-Speech Tagging

Part-of-Speech (POS) ist der englische Begriff für die Wortart, deren Erkennung mit einem sogenannten POS Tagger umgesetzt wird. Dieser ordnet jedem der im vorherigen Schritt segmentierten Token eine Wortart zu. Bei der weiteren Verarbeitung kann das Unterscheiden der Wortarten wichtig sein, um nur bestimmte Wortarten zu verwenden. So empfehlen Debortoli et al. (2016) die Verwendung von nicht allen Wortarten. Etwa Symbole oder Leerzeichen sollten von der weiteren Verarbeitung ausgenommen sein. Das Filtern der Wortarten findet jedoch nicht erst in der Pipeline statt, sondern stellt einen Schritt vor dem Anwenden der weitere Verarbeitungsschritte.
Die Wortarten werden als sogenannte POS Klassen definiert. Im Deutschen wird sehr oft des Stuttgart Tübingen TagSet (STTS) Schiller et al. (1995) mit 54 Klassen verwendet. Das

Universal Part-of-Speech Tagset Petrov et al. (2011) stellt eine Vereinfachung der 54 Klassen auf zwölf Wortarten dar und reicht für die in dieser Arbeit verwendeten Methoden aus.
Da auch unbekannte Token verarbeitet werden sollen, ist ein Ansatz unter Verwendung eines Wörterbuchs nicht ratsam. Deshalb werden Features bestimmt, welche gewisse Worteigenschaften beschreiben oder beziehen sich auf die Umgebung des Wortes. Es gibt mehrere Ansätze diese Features zu bestimmen, Poel et al. (2007) und Nakagawa et al. (2001) beschreiben dabei gute Herangehensweisen.
So können n Token vor und m Token nach dem zu klassifizierenden Token betrachtet werden. Oft gilt dabei $n = m$ und $n, m \in \{1, 2\}$. Es wird also die unmittelbare Umgebung betrachtet. Den Token selber auch als ein Feature zu benutzten ist hilfreich, falls der zu klassifizierende Token schon bekannt ist. Um von der Wortstruktur auf die Wortart zu schließen können Präfixe bzw. Suffixe zur Entscheidung heran gezogen werden. Nomen enden im Deutschen z.B. oft auf *ung* oder Verben beginnen häufiger mit bestimmten Präfixen. So können auch Präfixe bzw. Suffixe von verschiedener Länge als Features in Frage kommen, wobei alles bis zu dieser Länge jeweils ein Feature ist. Schließlich sind neben der Länge des Token auch noch Klassen zu Worteigenschaften zur Klassifikation benutzt worden. Das Vorhandensein und die Lage von Zahlen als auch Großbuchstaben können dabei interessant sein.
Mit diesen Features kann eine Support Vector Machine (SVM) trainiert werden, ähnlich wie Nakagawa et al. (2001). Andere Ansätze verwenden regelbasierte Methoden Brill (1992), Hidden Markov Models Kupiec (1992) oder auch Entscheidungsbäume Schmid (2013).

2.2.3 Lemmatisierung

Ein Lemma bezeichnet die Grundform eines Wortes. In der maschinellen Verarbeitung von natürlicher Sprache kann es wichtig sein, Wörter auf ihre Grundform zu reduzieren, um konsistente Ergebnisse zu erhalten. Beim Information Retrival ist es vorteilhaft Lemmatisierung einzusetzen, wie zum Beispiel Balakrishnan und Lloyd-Yemoh (2014) zeigen. Beim Topic Modeling findet Lemmatisierung ebenfalls Anwendung, etwa bei Hu et al. (2014).
Zur Lemmatisierung wird im Deutschen üblicherweise ein Ansatz unter Verwendung von Wörterbüchern verfolgt. Das Erstellen dieser kann jedoch auf unterschiedliche Weise erfolgen. Klassisch werden dort manuell Worte und ihre Lemmas eingegeben, wie von Lezius et al. (1998). Dieser Ansatz erfordert viele Ressourcen, was sich am Umfang des Wörterbuchs bemerkbar macht . Das Problem fehlender Wörter versuchen Liebeck und Conrad (2015) zu lösen, in dem die Daten von einem öffentlich zugänglichem und einer großen Community gepflegtem Wörterbuch erlangt werden.

2.2.4 Stoppworteliminierung

Die Eliminierung von Stoppwörtern meint zunächst einmal die Erkennung von Stoppwörtern. Stoppwörter sind Wörter, die zwar eine grammatikalische Funktion erfüllen, jedoch nichts zur Semantik beitragen. Das Entfernen kann zur Genauigkeit von Metho-

den beitragen, mit denen man Strukturen erkennen möchte. Generell gibt es verschiedene Methoden um Stoppwörter zu erkennen. Neben Wörterbüchern und statistischen Algorithmen können auch domänenspezifische Filterlisten zum Einsatz kommen. Alle Methoden haben individuelle Vorteile. Statistische Verfahren eliminieren vor allem häufige Wörter oder Wörter die nicht zu dazu beitragen verschiedene Dokumente auseinanderzuhalten, wie in dem Verfahren von Wilbur und Sirotkin (1992). Allgemeine Wörterbücher werden von Menschen erstellt und enthalten Wörter z.B. Präpositionen oder Artikel. Dieses Vorgehen erzielt einen ähnlichen Effekt, wie das Filtern von Wortarten. Domänenspezifische Filterlisten enthalten Wörter, die in einer Domäne kein weiteres Verständnis bringen.
Generell ist es empfehlenswert mehrere dieser Methoden zusammen einzusetzen, vor allem nicht nur eine statistische. Das liegt vor allem daran, dass ein Mensch im Gegensatz zu einem Computer über die Semantik eines Wortes entscheiden kann.

2.2.5 Named Entity Recognition

Mit der Tokenisierung verknüpft, ist die Erkennung von Eigennamen, Named Entity Recogition genannt. Hier sollen Namen von z.B. Menschen, Firmen, Organisationen oder auch Orten erkannt werden. Oft geht damit eine Klassifikation dieses Namens einher. Der gesamte Text wird nach Named Entinties durchsucht, wobei diese aus einzelnen oder mehreren Token bestehen können.
Zunächst werden, analog zum POS Tagging, Features der Token bestimmt, die jedoch auch die Wortart beinhalten können. Weitere Features können z.B. orthographische Informationen oder ob Anführungszeichen in der Umgebung sind. Diese Features werden dann wieder in statischen oder lernenden Modellen verwendet. Eine sehr gute Übersicht über verschiedene Features und Modelle bieten Tjong Kim Sang und De Meulder (2003). Generell ist anzumerken, dass die Erkennung von der Domäne der Texte abhängt, was beim Training oder der Auswahl solcher Modelle berücksichtigt werden sollte.
Beim Topic Modeling können Named Entities als ein Begriff verwendet werden, anstatt die einzelnen Token des Named Entity.

2.2.6 Besonderheiten bei Online-Partizipationsverfahren

Online-Partizipationsverfahren haben einige Besonderheiten gegenüber anderen Textdokumenten. Zwar können auch Vorschläge in der Verwaltung oder von einer Organisation erarbeitet werden. Meistens werden die Beiträge von allerdings Bürgern verfasst. Dabei werden häufig Umgangssprache und untypische Abkürzungen verwendet sowie nicht immer die Rechtschreibung eingehalten. Dies ist vor allem bei Kommentaren zu Vorschlägen der Fall. Ein Kommentar hat eine andere Tonalität als ein formell erarbeiteter Vorschlag. Auch ist zu bedenken, dass Kommentare, anders als Vorschläge, Sarkasmus und Ironie enthalten können, was Ergebnisse einer Sprachverarbeitung verzerren kann. In der Natur von Kommentaren liegt auch deren große Abhängigkeit vom Kontext, den sowohl der Vorschlag als auch andere Kommentare darstellen. Nicht selten beinhalten Kommentare Antworten auf andere Kommentare, so dass die Kommentare nicht getrennt von einander betrachtet werden sollten. Dies führt zu dem Schluss, dass beim

Erheben der Daten die Struktur der Kommentare beibehalten werden soll. Viel wichtiger ist jedoch, dass der Vorschlag und die Kommentare als ein Dokument betrachtet werden sollten.
Aufgrund der Konkretisierung von Online-Partizipationsverfahren können domänenspezifische Stoppwörter auftreten. Nicht immer ist dabei unstrittig, ob ein Wort ein Stoppwort ist. So können etwa der Name einer Kommune bei einem kommunalen Verfahren, das Wort *„Rad“* bei einem Rad-Dialog oder das Wort *„Vorschlag“* wenig zur Semantik beitragen. Das Erstellen einer solchen Filterliste benötigt ein gewisses Expertenwissen. Auch durch die Moderation der Vorschläge und Kommentare können Stoppworte entstehen, zum Beispiel kann dies der Name des Betreuenden oder Worte die im Zusammenhang mit der Moderation stehen, wie *„verschieben“* oder *„Kategorie“*.
Ein besonderes Augenmerk sollte auch auf die Erkennung von Straßennamen, also Named Entities, gelegt werden. Online-Partizipationsverfahren in Kommunen enthalten sehr oft Namen von Straßen oder Stadtteilen, wenn sich der Vorschlag auf einen bestimmten Ort bezieht. Besonders stark war dies bei den Raddialogen (siehe Tabelle 1) der Fall, weshalb zu den Vorschlägen eine Adresse angegeben werden bzw. auf einer Karte markiert werden konnte wie in Abbildung 1 zu sehen ist. Diese Adressen sollten ebenfalls zum Dokument gezählt werden, da sie stark zur Semantik der Vorschläge und Kommentare beitragen.
Im Laufe meiner Projektarbeit habe ich mich schon mit der Verarbeitung von Straßennamen beschäftigt. Der Gedanke ist, dass nachdem alle Named Entities erkannt wurden, Ortsangaben zu einem Token zusammengefügt und in eine einheitliche Schreibweise überführt werden. Die einheitliche Schreibweise dient dem gleichen Zweck wie die Lemmatisierung. Wird in zwei Beiträgen über die gleiche Straße oder den gleichen Platz gesprochen, soll dies als die gleiche Straße erkannt werden, auch wenn dies anders aufgeschrieben ist.
Die aus zwei Schritten bestehende Vorgehensweise beginnt damit Named Entities zusammenzufügen, die ein für einen Ort übliches Schlüsselwort enthalten, wobei alle Begriffe auf Kleinschreibung gebracht werden. Leerzeichen bleiben in diesem Schritt erhalten. Als Schlüsselwörter wurden gewählt:

strasse, straße, str, str., platz, gasse, allee, ufer, weg

Im nächsten Schritt werden die zusammengefügten Begriffe auf eine einheitliche Schreibweise gebracht. Dazu wird eine Heuristik benutzt: Alle Abkürzungen und die Schreibweise mit doppeltem *s* von Straße auf *„Straße“* bringt. Bei Begriffen mit einem Leerzeichen wird dieses entfernt, bei mehr als einem Leerzeichen werden diese durch Bindestriche ersetzt. Auch die Groß- und Kleinschreibung wird verändert. Tabelle 2 zeigt Beispiele aus einem Raddialog dazu.

Token des Named Entity	Zusammengefügter Token	Vereinheitlicht
„endenicher“ , „str.“	*„endenicher str.“*	*„Endenicherstraße‘‘*
„endenicher“ , „strasse“	*„endenicher strasse“*	*„Endenicherstraße‘‘*
„clemens“ , „august“ , „strasse“	*„clemens august straße“*	*„Clemens-August-Straße‘‘*
„clemens-august-str“	*„clemens-august-str“*	*„Clemens-August-Straße‘‘*

Tabelle 2: Beispiele zur Verarbeitung von Straßennamen. Zusammenhängende Token stehen in Anführungszeichen.

Eine kurze Evaluation hat gezeigt, dass die *Precision* dieses Ansatzes bei mehreren Online-Partizipationsverfahren bei über 80 % lag. Beim größten der drei Raddialoge, dem aus Bonn, wurden in 2331 Dokumenten 1318 Straßennamen ersetzt. Dies zeigt deutlich wie wichtig domänenspezifische Überlegungen sind.
Jedoch stellt dieser Ansatz lediglich eine Heuristik da und ist relativ naiv. Die Performance hängt wesentlich an der Performance der Named Entity Recognition des eingesetzten NLP-Toolkits, verbessert diese jedoch. Diese Heuristik erkennt weder Rechtschreibfehler in Straßennamen, noch alle Abkürzungen. Auch beinhalten nicht alle Straßennamen einen der oben aufgeführten Begriffe. Für eine bessere Erkennung bietet es sich also an, Named Entity Recognition gezielter auf Straßennamen zu trainieren oder ein Verzeichnis von Straßennamen ähnlich einem Wörterbuch zu benutzten.

3 Information Retrieval

Information Retrieval (IR) bezeichnet das Erlangen von Informationen aus einer Menge von Daten aufgrund einer Interaktion mit einem Benutzter. Die entwickelte Anwendung enthält Verfahren des IR, die neben der Interaktion auch zur Visualisierung genutzt werden. Diese Daten liegen dabei oft unstrukturiert, bzw. nicht nach der gesuchten Information strukturiert vor. Wird nun nach Informationen gesucht, müssen diese Daten gefiltert werden, idealerweise auch in einer Reihenfolge. Eingesetzt wird IR vor allem von Suchmaschinen bzw. Suchfunktionen auf Websites. Dabei kann eine große Menge von Daten eine der Herausforderungen sein, denen IR begegnen muss. Gesucht werden kann dabei nach Texten, Metadaten aber auch dem Inhalt von Bildern oder Audiodateien. Williams (2003) setzt Information Retrieval zur Suche nach Genomen ein was zeigt, dass es vielfältige Anwendungsmöglichkeiten von IR gibt.
Im Folgenden werden der Suchvorgang an sich und einige Modelle erläutert, die zum Retrieval von Text geeignet sind. Dabei wird vor allem auf das Vektorraummodell eingegangen sowie auf die Möglichkeit mit Feedback des Benutzers das Ergebnis zu verbessern. Eine Erläuterung über die Evaluation von IR-Systemen wird in dieser Arbeit nicht gegeben.
In Abbildung 3 ist der Suchvorgang skizziert. Neben den Repräsentationen der Dokumente können auch Metadaten zur Filterung, beispielsweise das Erstellungsdatum, mitberücksichtigt werden. Die interne Darstellung der Dokumente kann gespeichert werden, also neben den Dokumenten bereit stehen. Dies wird auch Indexierung genannt.

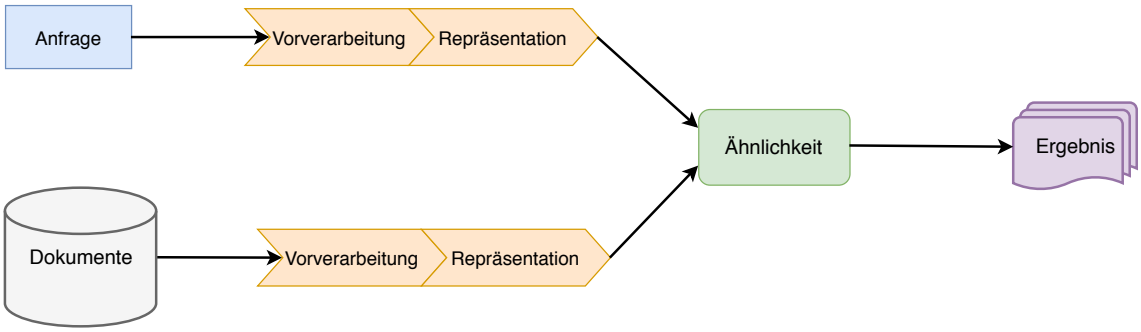


Abbildung 3: Schaubild des Suchvorgangs.

3.1 Vorverarbeitung

Bevor die erste Suchanfrage an das System gestellt wird, werden alle Dokumente in eine Form überführt, die es erleichtert Anfragen zu stellen, die ebenfalls in diese Form überführt werden. Diese Form entsteht bei einer Vorverarbeitung, die bei Text das Natural Language Processing darstellt. Welche Schritte der Pipeline dabei zum Einsatz kommen, hängt vom verwendeten Modell ab. Bei der Volltextsuche wird am wenigsten verarbeitet, da ein Suchmuster exakt gefunden werden soll. Dem gegenüber steht die wortbasierte Suche, die die genaue Reihenfolge der im Text befindlichen Wörter nicht beachtet, sondern Texte als Mengen von Wörtern betrachtet. Damit dies möglich ist, wird notwendigerweise die Tokenisierung der NLP-Pipeline verwendet. Auch der Einsatz von

Lemmatisierung kann sinnvoll sein, was Suchanfragen auf alle grammatikalischen Formen eines Wortes ausdehnt. Verschiedene grammatikalische Formen oder Wörter mit gleichem Wortstamm in die Suche mit einzubeziehen, liefert oft bessere Ergebnisse wie Balakrishnan und Lloyd-Yemoh (2014) zeigen.
Ein weiterer wichtiger Aspekt ist das Entfernen von Stoppwörtern im Zusammenhang mit Information Retrievel. Aus Semantischer Sicht, kann es sinnvoll sein Stoppwörter zu entfernen, da dies die Genauigkeit des Systems steigert. Dokumente die lediglich die Stoppwörter der Anfrage enthalten, sollten nicht als der Anfrage ähnlich gelten. Da außerdem besonders häufige Wörter als Stoppwörter gelten, sollten diese auch entfernt werden um das Ergebnis nicht zu verfälschen.
Insgesamt zeigen Manning et al. (2008, S. 87) die Wichtigkeit der Vorverarbeitung. Dabei verdeutlicht Tabelle 5.1, dass die Vorverarbeitung die Anzahl der Terme verringert, was es wahrscheinlicher macht passende Dokumente zu finden. Jedoch merken sie an, dass bei Suchmaschinen der Trend zu weniger umfangreichen, bis keinen Stoppwortlisten geht. Google hingegen soll eine Stoppwort benutzen[2].

Es gibt verschiedene Modelle um die Dokumente und Anfrage zu repräsentieren, die ein schnelleres Vergleichen einer Anfrage mit einem Dokument ermöglichen. Diese betrachten die Dokumente als Menge von Wörtern. Dabei gibt es verschiedene Ansätze. Für diese Arbeit ist größtenteils nur das Vektorraummodell interessant. Jedoch können auch Ergebnisse der Verfahren aus Kapitel 4 genutzt werden.

3.2 Boolesches Modell

Das Boolesches Modell verwendet boolesche Operatoren und entscheidet nur zwischen zwei Zuständen. Zunächst wird über die Terme aller Dokumente nach der Vorverarbeitung ein Index T gebildet, wobei T die Menge aller Terme $t_i \in T$ beschreibt. Dabei ist $0 \leq i \leq m$ und m ist die Anzahl aller Terme. Sei D die Menge der Dokumente n und $d_j \in D$ ein Dokument mit $0 \leq j \leq n$ (Manning et al., 2008). Für d_j spielen weder die Reihenfolge in welcher die Terme im Text vorkommen noch die Häufigkeit des Auftretens der Terme eine Rolle.
Die Anfrage Q ist ein boolescher Ausdruck, der Terme aus T mit booleschen Operatoren verknüpft. Oft ist dieser Ausdruck in konjunktiver oder disjunktiver Normalform. Für die drei Terme t_a, t_b und t_c kann eine Anfrage Q, die nach allen Dokumenten sucht, die t_a oder t_b und außerdem t_c enthalten, zum Beispiel so aussehen:

$$Q = (t_a \vee t_b) \wedge t_c$$

Es gibt verschiedene Ansätze die Dokumente zu indizieren und somit Anfragen zu bearbeiten.
Ein Model lässt sich analog zu booleschen Variablen beschreiben, bei denen das Vorkommen eines Terms t_i in d_j mit einer 1 und das Gegenteil mit 0 beschrieben wird. Mit dieser Vorgehensweise kann man eine Term-Dokument-Matrix der Größe $m \times n$ bilden, wie es Manning et al. (2008, S. 4) tun. Dabei wird jedes Dokument als Vektor über jeden Term dargestellt, wobei eine 1 bzw. 0 das Vorhandensein ausdrückt. Abbildung 4 zeigt ein Beispiel für eine solche Matrix.

[2] https://meta.wikimedia.org/wiki/Stop_word_list/google_stop_word_list#German

$$
\begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array}
\begin{array}{c}
\begin{array}{cccc} d_1 & d_2 & d_3 & d_4 \end{array} \\
\left(\begin{array}{cccc}
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0
\end{array} \right)
\end{array}
$$

Abbildung 4: Term-Dokument-Matrix der Größe 6×4.

Um eine Anfrage zu beantworten können nun die Vektoren der in Anfrage befindlichen Terme eingesetzt werden. Diese Vektoren habe die Dimension n. Das Ergebnis der Auswertung des booleschen Ausdruck ergibt einen Vektor, der im Ergebnis der Anfrage enthaltene Dokumente mit einer 1 bezeichnet. Das nachfolgende Beispiel zeigt die Anwendung der Anfrage $Q = (t_a \vee t_b) \wedge t_c$ auf das Beispiel der Abbildung 4, wobei $t_a = t_1, t_b = t_2$ und $t_c = t_3$ sein sollen. Die Vektoren wurden hier als Bitfolgen aufgeschrieben.

$$
\begin{aligned}
Q &= (t_a \vee t_b) \wedge t_c = (t_1 \vee t_2) \wedge t_3 \\
Q &= (1001 \vee 1100) \wedge 0011 \\
Q &= 0001 \\
Q &: \{d_4\}
\end{aligned}
$$

Ein Problem dieses Ansatzes mit der Term-Dokument-Matrix ist die Größe die diese Matrix annehmen kann. Es muss so für jeden Term ein Vektor der Dimension n bereitgehalten werden, was bereits bei der Erstellung der Matrix zum Problem werden kann. Aus diesem Grund bietet sich eine effizientere Indexstruktur an, zum Beispiel invertierte Listen. Diese speichern für jeden Term eine Liste von Dokumenten, die diese Terme enthalten. Somit reduziert sich der benötigte Speicher, da nicht jedes Dokument alle Terme enthält, sondern nur einen Bruchteil aller Terme aus T. Für das Beispiel aus Abbildung 4 ist eine invertierte Liste in Abbildung 5 angegeben.

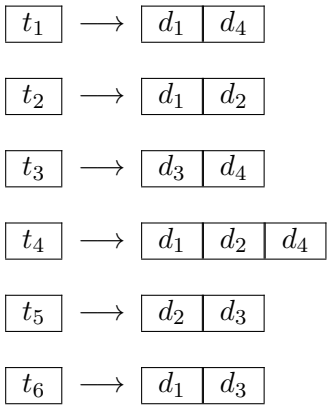


Abbildung 5: Invertierte Listen zum Beispiel aus Abbildung 4.

Die Rückgabe kann dann nicht mehr mittels Bitfolgen angeben werden, sondern es muss mit Mengen gerechnet werden. So werden die booleschen Operatoren mit den korrespondierenden Mengenoperatoren ersetzt, z.B.$\wedge$ mit $\cap$, $\vee$ mit $\cup$ und $\oplus$ mit $\triangle$. Dafür sei s_i die Menge der Dokumente die t_i enthalten, also die Menge die als Liste für t_i gespeichert ist. Die Anfrage aus dem laufenden Beispiel ist nachfolgend gezeigt.

$$
\begin{aligned}
&Q = (t_a \vee t_b) \wedge t_c = (t_1 \vee t_2) \wedge t_3 \\
&Q : (s_1 \cup s_2) \cap s_3 \\
&Q : (\{d_1, d_4\} \cup \{d_1, d_2\}) \cap \{d_3, d_4\} \\
&Q : \{d_4\}
\end{aligned}
$$

Das boolesche Modell hat in dieser Form jedoch auch einige Nachteile. Neben der Tatsache, dass die Anfragen für Benutzer wenig intuitiv sind, kann jedoch nur exakt gesucht und keine Ähnlichkeiten betrachtet werden. Dies kann dazu führen, dass kein Ergebnis gefunden wird oder nur unzufriedenstellende Ergebnisse erzielt werden. Auch ist in dieser Form so keine Reihenfolge der Ergebnisse gegeben. Für manche Anwendungen ist das boolesche Modell bestimmt besser geeignet, als für Texte.
Etwas ausgeglichen werden können diese Nachteile durch die Erweiterung der invertierten Listen und der Vereinfachung Anfragen. Neben den Referenzen zu den Dokumenten können die Listen an den Token ebenfalls die Termhäufigkeit (siehe Kapitel 3.3.1)speichern Lin und Dyer (2010). Dies bietet den Vorteil, dass die erhaltenen Dokumente in eine Reihenfolge gebracht werden können, indem die Termhäufigkeiten addiert werden. Dies bevorzugt jedoch längere Dokumente gegenüber kürzeren. Wird die Anfrage derart vereinfacht, dass alle vom Nutzer eingegeben Begriffe mit *oder* verknüpft werden, lässt sich so eine einfache Schlagwortsuche implementieren.
Außerdem lassen sich invertierte Listen mit Techniken aus 3.3 verwenden.

3.3 Vektorraummodelle

Ein anderer Ansatz Dokumente und Anfragen zu repräsentieren, sind Vektorraummodelle. Der Unterschied zum Booleschen Modell, welches man auch mit Vektoren darstellen kann, ist, dass Terme in diesen Vektoren gewichtet werden und auch Anfragen als Vektoren dargestellt werden. Aus diesem Grund können in diesem Vektorraum Ähnlichkeitsmaße angewendet werden. Dies bietet neben einer Ähnlichkeitssuche den Vorteil, Dokumente in der Reihenfolge der Ähnlichkeit zu erhalten.
Jedoch hat ein Vektorraummodel auch einige Nachteile. Zum einen folgt aus einer hohen Anzahl an Termen eine große Dimensionalität, was üblicherweise der Fall ist. Da sich Vektoren bei zunehmender Dimensionalität immer ähnlicher werden (Donoho et al., 2000), verschlechtert sich folglich die Güte von ähnlichkeitsbasierten Verfahren wie IR. Ein weiter Nachteil ist, dass für die Berechnung der Vektoren (wie in Kapitel 3.3.1) keine semantischen Beziehungen berücksichtigt werden. Aus diesem Grund ist die Ähnlichkeit nicht semantisch, sondern beruht auf der Verwendung gleicher Wörter. Wie in Kapitel 3.4 gezeigt wird, können dimensionsreduzierende Verfahren diesen beiden Problemen begegnen.

Sowohl für die Vektoren der Dokumente und Anfragen als auch für die Ähnlichkeit dieser gibt es verschiedene Überlegungen, von denen einige in den folgenden Abschnitten vorgestellt werden.

3.3.1 Dokumentenrepräsentation

Die Grundlage für die Berechnung der Dokumentenvektoren stellt die Häufigkeit der Terme dar. Dabei wird ein Vektorraum aufgespannt, in dem jeder Term eine Dimension darstellt, sodass bei m Termen der Vektorraum die Dimension m hat. Jedes der n Dokumente hat somit einen Vektor $\overrightarrow{(v)}$. Jedoch beschreibt der i-te Eintrag in $\vec{v}(d)$, das Gewicht des Terms t_i. Der naive Ansatz wählt für die Gewichte die Anzahl der einzelnen Terme. So beschreibt $tf_{t,d}$ die absolute Häufigkeit eines Terms $t \in T$ im Dokument $d \in D$. Da dieser Ansatz jedoch nicht die Gesamtanzahl der Terme in einem Dokument berücksichtigt, werden so bei Anfragen lange Dokumente, in denen die gesuchten Terme relativ selten vorkommen, stärker berücksichtigt als kurze Dokumente, beiden die gesuchten Terme zwar absolut weniger häufig sind als im langen Dokument, aber in diesem Dokument relativ häufig. Es gibt mehrere Wege sich diesem Problem zu nähern. Eine Herangehensweise logarithmiert tf mit

$$l\text{-}tf_{t,d} = log(tf_{t,d} + 1)$$

, sodass je häufiger ein Wort ist, desto weniger fließt ein weiteres Auftreten in das Gewicht ein.
Werden nun zwei Dokumente verglichen, fließen häufige Begriffe besonders stark in die Ähnlichkeitsberechnung ein. Dies wird dann zum Problem, wenn die häufigen Terme eines Dokumentes in allen Dokumenten häufig sind. Die Repräsentation sollte die Besonderheiten des Dokuments hervorheben, also all jene Terme stärker gewichten, die in der Gesamtheit der Dokumente weniger häufig sind. Dadurch werden bei der Betrachtung wie ähnlich zwei Dokumente sind, die Unterschiede adäquat berücksichtigt. Um dies zu erreichen wird für jeden Term t die Dokumentenhäufigkeit df_t errechnet, also in wie vielen Dokumenten der Term vorhanden ist. Da jedoch seltene Terme ein höheres Gewicht haben sollten, wird die inverse Dokumentenfrequenz idf_t berechnet:

$$idf_t = \log \frac{n}{df_t}$$

So lässt dich aus $l\text{-}tf$ und idf die $tf\text{-}idf$ Gewichtung bilden:

$$tf\text{-}idf = l\text{-}tf_{t,d} \cdot idf_t$$

Da es sich um Vektoren handelt, können sie mit ihrer Länge normalisiert werden. Dazu kann die Euklidische Norm verwendet werden, was dazu führt dass alle Einträge im Intervall $[0, 1]$ liegen. Sei $\vec{v}(d) \in D$ ein Dokumentenvektor mit m Dimensionen, wobei m die Anzahl der Terme ist. Dann lässt sich dessen Länge bei Verwendung von $tf_{t,d}$ wie folgt berechnen:

$$tf\text{-}idf_{t,d} = \frac{l\text{-}tf_{t,d} \cdot idf_t}{\sqrt{\sum_{i=1}^{m} (l\text{-}tf_{t_i,d} \cdot idf_{t_i})^2}}$$

3.3.2 Ähnlichkeit mit einer Anfrage

Es gibt insgesamt sehr viele Möglichkeiten die Ähnlichkeit einer Anfrage zu den Dokumenten zu berechnen, wie Zobel und Moffat (1998) zeigen. Jedoch können sie keine Möglichkeit als besonders gut herausstellen.
Die Ähnlichkeit zweier Vektoren, in diesem Fall einer Anfrage und einem Dokument, wird üblicherweise als Wert zwischen 0 und 1 ausgedrückt. Je größer der Wert ist, desto ähnlicher sind die beiden verglichenen Vektoren. Dies ist mit der Kosinus-Ähnlichkeit möglich, die den Winkel zwischen zwei Vektoren angibt. Der Kosinus von $0°$, also zwei gleichen Vektoren, ist 1 und somit sind die Vektoren gleich. Bei zwei orthogonal zueinander stehenden Vektoren hat der Kosinus einen Wert von 0. Da Dokumentenvektoren keine negativen Einträge, also keine negativen Termhäufigkeiten, haben können, begrenzt das den Wertebereich auf $[0, 1]$. Berechnet wird die Kosinus-Ähnlichkeit von nicht normalisierten Vektoren einer Anfrage q und eines Dokumentes d mit:

$$\text{sim}(q,d) = \frac{\vec{v}(\mathbf{q}) \cdot \vec{v}(\mathbf{d})}{\|\vec{v}(\mathbf{q})\|\|\vec{v}(\mathbf{d})\|} = \frac{\sum_{i=1}^{m} \vec{v}(q)_i \vec{v}(d)_i}{\sqrt{\sum_{i=1}^{m} \vec{v}(q)_i^2} \sqrt{\sum_{i=1}^{m} \vec{v}(d)_i^2}}$$

Bei der Verwendung von $tf\text{-}idf$ sind die Vektoren allerdings schon normalisiert, weshalb sich die Kosinus-Ähnlichkeit mit

$$\text{sim}(q,d) = \vec{v}(\mathbf{q}) \cdot \vec{v}(\mathbf{d}) = \sum_{i=1}^{m} \vec{v}(q)_i \vec{v}(d)_i$$

berechnen lässt.
Es gibt aber auch Überlegungen den Anfragevektor anders zu berechnen als einen Dokumentenvektor. Salton und Buckley (1988) empfehlen Anfragen nicht zu normalisieren und mit jeden Term der Anfrage mit mindestens 0.5 zu gewichten:

$$\vec{v}(q)_t = \left(0.5 + \frac{0.5 \cdot tf_{t,q}}{\max\limits_{\tilde{t} \in T} tf_{\tilde{t},q}}\right) \cdot \log \frac{n}{df_t}$$

So sind Terme in kurzen Anfragen wichtiger, wobei in längeren Anfragen häufige Terme weniger Einfluss erhalten. Mit dieser Berechnung ist es weiterhin möglich die Kosinus-Ähnlichkeit einzusetzen.
Die *state of the art* Methode von Robertson, Zaragoza et al., 2009 berechnet die Ähnlichkeit jedoch nicht mit der Kosinus-Ähnlichkeit sondern schlägt ein eigenes Maß vor und ist unter dem Namen *Okapi BM25* bekannt. Diese ändert die Skalierung der Termfrequenz nicht mit dem Logarithmus, sondern mit dem Verhältnis der Dokumentenlänge $|d$ zur durchschnittlichen Dokumentenlänge $avgdl$. Kommt ein Wort in einem statistisch kurzem Dokument häufig vor wird es so mehr gewichtet, als wenn es mit gleicher Häufigkeit in einem überdurchschnittlich langem Dokument vorkommt. Zusätzlich gibt es Gewichte in der Funktion und sie wird zu den probabilistischen Verfahren gerechnet.

$$BM25(q,d) = \sum_{t \in q} idf_t \cdot \frac{tf_{t,d} \cdot (k_1 + 1)}{tf_{t,d} + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})}$$

Manning und Schütze (1999) empfehlen einen Wert für k_1, also die Skalierung von tf, zwischen 1.2 und 2.0. Für den Faktor b, der die Längennormalisierung skaliert, wird ein Wert von 0.75 empfohlen.

3.3.3 Weitere Ansätze

Daneben gibt es auch Ansätze, die anstatt eines Anfragevektors die Eigenschaften der Anfrage wie die Termhäufigkeit in das Ähnlichkeitsmaß mit einbeziehen. Jones et al. (2000) beschreiben ein probabilistisches Verfahren, welches unter anderem die durchschnittliche Anzahl an Termen in Dokumenten mit einbezieht. Sie bieten ebenfalls extra Berechnungen für lange Anfragen.
Das Vektorraummodell wird von Wong et al. (1985) erweitert, wobei ebenfalls die Zusammenhänge zwischen den Termen mit einbezogen werden.
Ein weiterer Ansatz kann sein, die Ähnlichkeitsfunktion zu verändern. Ausgehend von *Okapi BM25* können für jedes Dokument verschiedene Features dazu addiert werden. Diese Features sind abhängig von den Dokumenten, die durchsucht werden. Bei Online-Partizipationsverfahren könnten dies z.B. die Anzahl an Zustimmungen oder Kommentaren sein, wobei aktuelle Zustimmungen stärker gewichtet werden können als ältere.
Machine Learning findet ebenfalls eine Anwendung beim Information Retrieval (Fuhr, 1992). So kann mit einer Trainingsmenge ein Modell trainiert werden, welches Dokumente in eine gewünschte Reihenfolge bringt. Diese Trainingsmenge enthält Anfragen, Dokumente sowie deren Relevanz zu den Anfragen.

3.4 Einsatz von feature-extrahierenden Verfahren

Die Verfahren, die in Kapitel 4 vorgestellt werden, können auch zum Information Retrieval benutzt werden und Problemen des Vektorraummodells begegnen. So verringern sie die Dimensionen, in dem Terme zu einer neuen Dimension zusammengefasst werden. In diesem verringerten Vektorraum wirkt der „Fluch der Dimensionalitäten“ weniger. Außerdem ermöglicht dieser eine Ähnlichkeitssuche im neuen Vektorraum, da die Dimensionen nach semantischen Zusammenhängen zusammengefasst werden. Die Güte der Information Retrieval Systeme, die diese Techniken einsetzen, ist deshalb auch von der Güte der dimensionsreduzierenden Verfahren abhängig.
Bei Verfahren wie NMF oder LSA, die jedem Dokument einen Vektor zuweisen, kann, analog zu Kapitel 3.3.2, die Kosinus-Ähnlichkeit verwendet werden. Dazu wird vorher die Dimensionsreduktion auf den Anfragevektor durchgeführt und anschließend die Kosinus-Ähnlichkeit berechnet.
Da es sich bei LDA um Wahrscheinlichkeitsverteilungen über alle Themen handelt, haben sich etwa eine symetrisch angewendete Kullback Leibler-Divergenz oder Jensen–Shannon-Divergenz bewährt (Steyvers und Griffiths, 2007).
Sollen alle Beiträge nach der Ähnlichkeit zu einem dieser Themen sortiert werden, reicht es die vorhandenen Daten nach der Dimension des gesuchten Themas zu sortieren. Würde dieser Anfragevektor binär mit einer 1 in der Dimension des gewünschten Themas formuliert werden, hätte die Kosinus-Ähnlichkeit als Ergebnis den Eintrag des Dokuments in eben dieser Dimension. Bei Verwendung von LDA funktioniert das Sortieren

ebenfalls.
Es gibt auch neuere Forschungsansätze, die mit neuronalen Netzen Repräsentationen lernen, die zum Information Retireval eingesetzt werden können (Shen et al., 2014; Liu et al., 2015). Dabei werden mit *CNNs* eine feste Anzahl an Repräsentationen gelernt, analog zu den dimensionsreduzierenden Verfahren. Jedoch arbeiten diese auf n-gram Ebene und nicht mit $tfidf$. Zum Trainieren werden Wahrscheinlichkeiten verwendet, die beschreiben wie wahrscheinlich ein Nutzer auf ein Dokument bei einer gegeben Anfrage klickt. Die Arbeiten von Shen et al. (2014) und Liu et al. (2015) unterscheiden sich in der Verwendung eines *convolutional layer*. Diese Methoden wurden in dieser Arbeit jedoch nicht aufgegriffen.

3.5 Relevance Feedback

Oft liefern Suchanfragen nicht genau die Ergebnisse, die ein Benutzer möchte. Neben fehlender Güte des Systems oder Ungenauigkeiten der Anfrage liegt dies oft im Wesen einer Suche begründet. Die Vorstellung was gesucht wird, konkretisiert sich oft beim Sichten erster Anfrageergebnisse.
Der wahrscheinlich meistbenutzte Lösungsansatz besteht in einer Veränderung der Anfrage. Entweder vom Nutzer explizit oder systemseitig implizit zu verändern, wobei beim impliziten Relevance Feedback Statistiken über Anfrage in die Berechnung von Anfragen verwendet wird. Geht diese Änderung vom Feedback des Nutzers aus, werden Beiträge als relevant markiert. Mit Hilfe dieses Feedbacks wird die Anfrage neu berechnet. Eine Übersicht dieses Vorgangs bietet Abbildung 6.

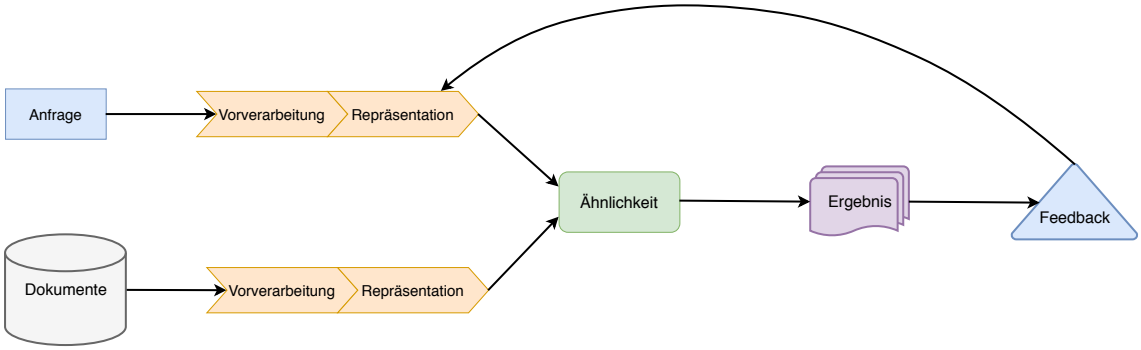


Abbildung 6: Schaubild des Suchvorgangs mit Relevance Feedback.

Die Idee ist, den Unterschied zwischen relevanten und nicht-relevanten Dokumenten als neue Anfrage zu benutzten. Dafür wird sowohl für die relevanten als auch für die nicht-relevanten Dokumente ein Durchschnittsvektor gebildet und vom Durchschnittsvektor der relevanten Dokumente wird der der nicht-relevanten abgezogen (Manning et al., 2008). Dabei beschreiben D_r die relevanten und D_n die nicht relevanten Dokumente. Das würde bei Verwendung der Kosinus-Ähnlichkeit die folgende neue Anfrage $\hat{q}$ erzeugen:

$$\vec{v}(\hat{q}) = \frac{1}{|D_r|} \cdot \sum_{d_r \in D_r} \vec{v}(d_r) - \frac{1}{|D_n|} \cdot \sum_{d_n \in D_n} \vec{v}(d_n)$$

Der Algorithmus von Rocchio (1971) bezieht in die Berechnung des neuen Anfragevektors $\vec{v}(\hat{q}$ jedoch auch die alte Anfrage mit ein und bietet zusätzlich die Möglichkeit der Gewichtung.

$$\vec{v}(\hat{q}) = \alpha \cdot \vec{v}q + \beta \cdot \frac{1}{|D_r|} \cdot \sum_{d_r \in D_r} \vec{v}(d_r) - \gamma \cdot \frac{1}{|D_n|} \cdot \sum_{d_n \in D_n} \vec{v}(d_n)$$

Manning et al. (2008) empfehlen als Gewichtung $\alpha = 1$, $\beta = 0.75$ und $\gamma = 0.15$, sodass die relevanten Dokumente stärker gewichtet werden, als jene Dokumente, die nicht als relevant ausgewählt wurden.
Einsetzbar ist es sowohl mit Vektorraummodellen als auch feature-extrahierenden Verfahren.

4 Themenextraktion

Die in diesem Kapitel vorgestellten Verfahren dienen mathematisch gesehen nicht primär einer Extraktion von Themen. Viel mehr können diese verwendet werden, um die Dimensionen von Matrizen zu verringern. Dafür werden ähnliche Dimensionen zusammengefasst. Die Verfahren unterscheiden sich darin wie sie die Dimensionen auswählen, die reduziert, also zusammengefasst, werden.
Diese Datenmatrix besteht aus m Features und n Objekten, analog zu der Matrix aus Abbildung 4. Werden diese Verfahren nun für Texte verwendet, steht jede dieser m Dimensionen für ein Wort im Vokabular, das über alle n Dokumente gebildet wird. Wie die Werte der Terme in den einzelnen Dimensionen berechnet werden können, wird in Kapitel 3.3 ausgeführt. Die Datenmatrix kann also auch als Matrix über alle Terme und Dokumente verstanden werden.
Reduziert wird die Anzahl der m Dimensionen, die jeweils für einen Term stehen, indem mehrere Dimensionen zusammengefasst werden. Die so erhaltenen, reduzierten Dimensionen repräsentieren mehrere Terme und können somit als Themen aufgefasst werden. Erkannt werden also semantische Strukturen. Deshalb wird der Einsatz dieser Verfahren auf Texte auch *Topic Modeling* genannt. Gemeinsam haben alle Verfahren, neben den erzeugten Themen, dass jedes Thema nun neue Features mit reduzierter Dimension hat. Die Anzahl der Dimensionen, auf die reduziert werden soll, wird ebenfalls vorher festgelegt.
Die Erklärungen der Verfahren werden immer mit Blick auf die Textverarbeitung erfolgen. Zu Beginn werden drei verschiedene Verfahren vorgestellt, die anschließend evaluiert werden. Anschließend wird ein Verfahren vorgestellt, dass eine Themenanzahl mit Hilfe der Stabilität der gefundenen Themen empfiehlt. Danach wird Interactive Topic Modeling vorgestellt und das Kapitel mit einer Schlussfolgerung mit Bezug auf Online-Partizipationsverfahren abgeschlossen.

4.1 Non-negative matrix factorization

Non-negative matrix factorization (NMF) ist ein Verfahren, welches meistens zum Clustering oder zur Dimensionsreduktion verwendet wird. Bei der Textverarbeitung kann es beim Topic Modeling zum Clustering und beim Information Retrieval zur Dimensionsreduktion zum Einsatz kommen. NMF zerlegt eine Matrix durch Annäherung in zwei Matrizen, wobei alle Matrizen keine negativen Einträge haben. Sei V die Datenmatrix der Größe $m \times n$, wobei m die Anzahl der Terme und n die Anzahl der Dokumente seien. Diese Matrix wird auch Term-Dokument-Matrix genannt und nun in zwei Matrizen W und H zerlegt, die durch Multiplikation eine zu möglichst angenäherte Matrix erzeugen sollen.

$$V \approx WH$$

Folglich hat W die Form $m \times k$ und H die Form $k \times n$, wobei k die Anzahl der Themen ist, die gefunden werden sollen.
Zur Berechnung wird eine Kostenfunktion benötigt, dessen Minimum gefunden werden soll. Eine mögliche Grundlage dieser Kostenfunktion ist die Frobeniusnorm $\|.\|_F$, also die euklidische Norm für Matrizen. Minimiert werden soll der Unterschied, also der Fehler,

zwischen V und dem Produkt von W und H. So lässt die das folgende Optimierungsproblem formulieren:

$$\min_{W,H} \|V - WH\|_F \text{ u.d.N. } W, H \geq 0 \quad (1)$$

Lee und Seung (2001) beschreiben einen viel implementierten Algorithmus zum finden dieser Matrizen W und H, der garantiert ein lokales Minimum findet. Die Initialisierung der Matrizen W und H erfolgt mit positiven Werten.
Mit den folgenden Regeln werden zuerst alle Einträge von Hund dann alle Einträge von W aktualisiert. Dabei wird zum Aktualisieren der Einträge in W schon die aktualisierte Matrix H verwendet. Die Indizes seien dabei $0 \leq i \leq m$, $0 \leq j \leq n$ und $0 \leq a \leq k \leq m$

$$H_{aj} \leftarrow H_{aj} \frac{(W^T V)_{aj}}{(W^T W H)_{aj}} \quad W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}} \quad (2)$$

Jeder Eintrag wird solange aktualisiert, bis der mit Gleichung 1 berechnete Fehler unter einen Schwellwert sinkt. Dies geschieht multiplikativ immer um den Faktor, den die Multiplikation von W und H falsch liegen.
Wird anstatt der Frobeniusnorm in Gleichung 1 die Kullback-Leibler-Divergenz verwendet, können die Regeln aus 2 angepasst werden, sodass es zu einem probabilistischen Verfahren wird. Dieses Verfahren ist dann äquivalent zu Probabilistic Latent Semantic Analysis (Gaussier und Goutte, 2005).
Die als Ergebnis erhaltene Matrix W enthält nun k Vektoren, die als Themen oder Konzepte aufgefasst werden können und m Dimensionen haben. Jedem Term ist nun ein nicht-negativer Zugehörigkeitswert zu jedem der k zugewiesen. So lässt sich also sowohl bestimmen aus welchen Wörtern die Themen zusammengesetzt sind als auch zu welchen Themen die Terme gezählt werden. In H hingegen besitzt jeder der n Terme einen Vektor mit Zugehörigkeitswerten zu den k Themen.

4.2 Latent Semantic Indexing

Das Verfahren Latent Semantic Indexing (Deerwester et al., 1990) ist auch unter dem Namen Latent Semantic Analysis bekannt. Es zerlegt ähnlich zu NMF die Datenmatrix V, um Zusammenhänge zwischen Termen zu finden. Jedoch wird dazu die Dimensionsreduktion auf eine Singulärwertzerlegung angewendet. Der Vorteil der Singulärwertzerlegung gegenüber einer Eigenwertzerlegung ist, dass V nicht symmetrisch sein muss, was bei einer Term-Dokument-Matrix der Fall ist.
Eine Singulärwertzerlegung der Form $V = W\Sigma H^T$ wird dazu als Ausgangspunkt verwendet. Um diese zu bekommen, wird oft der Lanczos-Algorithmus (Lanczos, 1950) verwendet. Die Matrizen W und H sind dabei analog zu NMF die Themenmatrix bzw. Dokumentenmatrix. Σ ist eine $r \times r$ große Diagonalmatrix auf dessen Diagonale die Wurzeln der Eigenwerte von V stehen, wobei r der Rang von V ist. In dieser Zerlegung haben W die Größe $m \times r$ und H die Größe $r \times n$. Mithilfe von Σ wird nun der semantische Raum verkleinert, in dem Wörter zusammengefasst werden. Dazu werden schrittweise die Dimensionen mit den geringsten Werten aus Σ entfernt, bis nur noch k Dimensionen vorhanden sind. So hat Σ die Größe $k \times k$, W die Größe $m \times k$ und H die Größe $k \times n$. Man erhält nun die folgende Zerlegung:

$$V = V_k \approx W_k \Sigma_k H_k^T$$

Die aus Σ entfernten Dimensionen fasst so Begriffe zusammen, die nur einen geringen Einfluss auf die Semantik haben. Die Vektoren in W und H haben auch negative Einträge, was es weniger intuitiv mach diese Themen auszuwerten. Landauer et al. (1998) haben gezeigt, dass mit LSI Themen extrahiert werden können, die Themen auf Grundlage von Menschen getroffener Extraktion nahe kommen. Anzumerken ist hier eine jedoch die als optimal angesehene Anzahl von 300 Themen bei einem Datensatz von bis zu 70000 Textpassagen. Möchte man die extrahierten Themen zur Übersicht über Dokumente verwenden, ist diese Anzahl jedoch wenig intuitiv, da sie zu hoch ist.

4.3 Latent Dirichlet Allocation

Blei et al. (2003) beschreiben das statistische Verfahren Latent Dirichlet Allocation. Dieses verwendet Dirichletverteilungen, eine Art multivarianter Wahrscheinlichkeitsverteilungen, um latente Themen in Dokumenten zu finden. Gearbeitet wird dabei wieder mit dem Bag-of-Words-Modell, beidem jedes Dokument als Multimenge seiner Wörter repräsentiert wird. Die verwendeten Modelle und Berechnungen sind komplexer als bei den anderen Verfahren, weshalb die Betrachtung vergleichsweise oberflächlich stattfindet.
Jedes der n Dokumente d, wird mit einer Dirichlet-Verteilung über k Themen als θ_d beschrieben, also wie wahrscheinlich das Dokument zu einem Thema gehört. Dabei können in jedem Dokument mehrere Themen vertreten oder auch gleich wahrscheinlich sein. Jedes der k Themen a wird mit einer Dirichlet-Verteilung über die m Terme als φ_a beschrieben, also wie wahrscheinlich ein Thema von diesem Term beeinflusst wird. Außerdem erhält jeder Term innerhalb eines Dokumentes ein konkretes der k Themen.
LDA verfolgt einen generativen Prozess, Verteilungen zu finden die wahrscheinlich sind, ein Dokument so zu erzeugen, wie es vorliegt. Am häufigsten wird dazu Gibbs-Sampling (S. Geman und D. Geman, 1984) verwendet. Angefangen wird damit, jedem Term in jedem Dokument zufällig ein Thema zuzuweisen (Chen, 2011). Dadurch entstehen zufällige Verteilungen für alle θ_d und φ_a. Nun wird iterativ jedes Dokument durchlaufen und dessen Terme nacheinander betrachtet. Für jeden Term wird dann das wahrscheinlichste Thema gespeichert, wodurch sich die Verteilungen θ_d und φ_a verändern. Diese Iteration über alle Dokumente erfolgt, bis das Ergebnis hinreichend stabil oder eine maximale Anzahl an Iteration erreicht ist.

4.4 Evaluation

In diesem Kapitel werden die Ergebnisse vom Topic Modeling einiger Online-Partizipationsverfahren verglichen. Als Verfahren wurden dazu NMF, PLSI, LSI, und LDA verwendet, da für diese Verfahren mehrere bewährte Implementierungen existieren.
Da eine Auswertung eines jeden Online-Partizipationsverfahrens zu umfangreich wäre, wurden zwei ausgewählt: Der Raddialog-Bonn und der Kölner Bürgerhaushalt 2015, wobei diese sich sowohl in der Anzahl der Vorschläge als auch im Thema welches sie behandeln unterscheiden. Werden Vorschläge und Kommentare zu einem Dokument kombiniert, resultiert das in 2331 Dokumenten für den Raddialog-Bonn und 631 Doku-

menten des Kölner Bürgerhaushalts 2015. Hinzu kommt die Möglichkeit der Angabe einer Adresse bei den Raddialogen, was ebenfalls zum Dokument gezählt wird.
Wesentlich ist die Wahl der Anzahl an Themen. Der Nutzer einer Themenextraktion, die eine Online-Partizipation unterstützen soll, wird die Anzahl an Themen subjektiv auswählen, was nicht zwangsläufig der mathematisch optimalen Anzahl an Themen entspricht. Jedoch sollte die Anzahl der Themen auch über die Granularität der gefundenen Themen entscheiden.
Die hier ausgewählten Anzahlen an Themen dienen lediglich dem Vergleich der Verfahren und sind willkürlich. Diese sollen vielmehr die Unterschiede der Verfahren bei unterschiedlicher Anzahl Themen verdeutlichen. Soll ein Verfahren auch zur semantischen Suche verwendet werden (vgl. Kapitel 3.4), orientiert sich die Anzahl der Themen an anderen Gesichtspunkten, als dass der Zweck die Übersicht über Themen in einer Menge von Dokumenten ist.
Tabellen 3 bis 6 zeigen dabei jeweils 15 Themen des Bonner Raddialogs und Tabellen 7 bis 10, zehn Themen des Kölner Bürgerhaushalts 2015.
Hierbei fällt auf, dass beim Raddialog Themen gefunden werden die intuitiv verständlich sind und relativ konkret, obwohl nur 5 Begriffe pro Thema gezeigt werden. Bei den Verfahren lässt sich nicht klar erkennen, welches am besten ist. Beim Kölner Bürgerdialog mit weniger Dokumenten sind die Unterschiede zwischen den Verfahren deutlicher. LSI findet teilweise sehr ähnliche Themen und subjektiv scheint NMF die besten Themen zu extrahieren.

Thema	Top-Words
1	kreisel, siegburgerstraße, westlich, beuel-ost, tannenbusch
2	radweg, straße, bonn, fahren, richtung
3	anliegen, beitrag, dank, erfordern, herzlich
4	bad, godesberg, rüngsdorf, überdacht, wartebereich
5	unterführung, bonn, poller, bonn-zentrum, hauptbahnhof
6	graurheindorferstraße, konvexspiegel, verkehrsspiegel, kostengünstig, bonn-castell
7	radfahrer, auto, fahren, bonn, radweg
8	beitrag, kategorie, vorschlag, dank, herzlich
9	radweg, bonn, radfahrer, weg, fahren
10	duisdorf, name, firma, verstehen, bleiben
11	bonn, fahrrad, rad, b56, richtung
12	kölnstraße, kreuzung, rosental, bonn-castell, godesberger
13	weg, bordstein, absenken, bonn, friesdorf
14	fahrbahndecke, vorrang, ellerstraße, radfahrweg, haltemöglichkeiten
15	bonner, talweg, reihe, tourist, farblich

Tabelle 3: LDA

Thema	Top-Words
1	kategorie, beitrag, zuordnen, verständnis, vorschlag
2	radfahrer, fahren, autofahrer, straße, einbahnstraße
3	radweg, schmal, richtung, bonn, straße
4	auto, parkend, parken, straße, radstreifen
5	fahrrad, bonn-zentrum, rad, hauptbahnhof, bonn
6	name, beitrag, entfernt, firma, werbung
7	duisdorf, maarweg, vorfahrt, schlagloch, rochusstraße
8	ampel, grün, rot, ampelschaltung, warten
9	weg, fußgänger, beleuchtung, rheinaue, fahrradweg
10	schutzstreifen, parkend, türbereich, streife, pkw
11	anliegen, beitrag, dank, langfristig, planerischen
12	kreuzung, kölnstraße, richtung, abbiegen, kommend
13	unterführung, poppelsdorferallee, poller, brücke, königstraße
14	fahrradstraße, kaiserstraße, fahrradstraßen, südstadt, straße
15	godesberg, bad, alt-godesberg, fehlen, zusatzzeichen

Tabelle 4: NMF

Thema	Top-Words
1	beitrag, kategorie, dank, zuordnen, herzlich
2	radweg, radfahrer, fahren, auto, richtung
3	ampel, grün, radfahrer, kreuzung, autofahrer
4	auto, schutzstreifen, parkend, straße, fahrradstraße
5	name, unterführung, entfernt, beitrag, fahrradständer
6	ampel, name, radweg, entfernt, beitrag
7	duisdorf, fahrradstraße, maarweg, vorfahrt, radweg
8	radfahrer, kölnstraße, richtung, kreuzung, fahren
9	duisdorf, weg, schutzstreifen, radfahrer, fußgänger
10	schutzstreifen, kölnstraße, duisdorf, radweg, bonn-zentrum
11	anliegen, autofahrer, kreisverkehr, handeln, straße
12	radfahrer, kaiserstraße, fußgänger, schmal, bus
13	poller, radweg, unterführung, radfahrer, auto
14	duisdorf, kölnstraße, auto, schlagloch, gehweg
15	kreisverkehr, gehweg, nordstadt, straße, ampel

Tabelle 5: LSI

Thema	Top-Words
1	moderation, herzlich, aufheben, richtig, beitrag
2	bonn, radfahrer, fahren, richtung, straße
3	radweg, bonn, schmal, nutzen, weg
4	auto, bonn, parkend, parken, straße
5	bonn, fahrrad, rad, bonn-zentrum, stadt
6	stehen, ampel, sicht, grün, bleiben
7	fahrradstraße, duisdorf, vorfahrt, fahrradstraßen, vilich
8	bonn, ampel, grün, auto, gronau
9	weg, radfahrer, brücke, fahren, endenich
10	bonn, bonn-zentrum, richtung, radfahrer, radweg
11	handeln, moderation, dank, beitrag, ort
12	bonn, straße, fahren, fehlen, schlagloch
13	radfahrer, radweg, autofahrer, poller, gefährlich
14	gefährlich, fahren, richtung, fahrradweg, fahrradfahrer
15	bad, godesberg, radfahrer, richtung, alt-godesberg

Tabelle 6: pLSI

Thema	Top-Words
1	platz, parkplatz, gehweg, vorschlag, bolzplatz
2	radfahrer, straße, radweg, fußgänger, fahren
3	kreisverkehr, sitzbank, chlodwigplatz, venloer, sperren
4	ebertplatz, legal, weg, eigelstein, kamera
5	zündorf, colonius, laubbläser, vorschlag, umgehungsstraße
6	zebrastreifen, ampel, kreuzung, müll, grün
7	wegweiser, vorgehen, djs, außengastronomie, sprache
8	kind, schule, schüler, bank, wichtig
9	frau, flüchtling, beratung, unterstützen, mann
10	köln, stadt, vorschlag, jahr, finden

Tabelle 7: LDA

Thema	Top-Words
1	stadt, köln, vorschlag, kölner, bürger
2	radfahrer, radweg, fußgänger, richtung, straße
3	zebrastreifen, ampel, kind, ampelanlage, sicherheit
4	kreisverkehr, kreuzung, ampel, ersetzen, doppelt
5	linie, bus, haltestelle, fahren, kvb
6	müll, mülleimer, wild, mülheim, awb
7	spielplatz, kind, bolzplatz, jugendliche, elter
8	bank, platz, baum, sitzgelegenheiten, schön
9	porz, zündorf, porzer, attraktiv, umgehungsstraße
10	parkplatz, auto, straße, anwohner, fahren

Tabelle 8: NMF

Thema	Top-Words
1	vorschlag, köln, stadt, straße, radfahrer
2	radfahrer, radweg, fußgänger, kreisverkehr, ampel
3	kind, zebrastreifen, spielplatz, kreisverkehr, ampel
4	kreisverkehr, linie, ampel, bus, kreuzung
5	linie, kind, spielplatz, bus, fahren
6	müll, mülleimer, wild, park, zebrastreifen
7	zebrastreifen, sicherheit, stadt, ampelanlage, köln
8	bank, platz, parkplatz, baum, bolzplatz
9	parkplatz, auto, kreisverkehr, parken, fahren
10	toilette, spielplatz, öffentlich, radfahrer, ampel

Tabelle 9: LSI

Thema	Top-Words
1	stadt, köln, vorschlag, jahr, sehen
2	radfahrer, fußgänger, straße, sicher, richtung
3	kind, zebrastreifen, sicher, spielplatz, sehen
4	kreuzung, vorschlag, kvb, jahr, kosten
5	linie, fahren, bus, auto, haltestelle
6	müll, finden, aufstellen, liegen, eck
7	vorschlag, gut, können, kreisverkehr, gruß
8	platz, schön, nutzen, innenstadt, köln
9	parkplatz, auto, köln, stadt, gut
10	straße, vorschlag, anwohner, bereich, verkehr

Tabelle 10: pLSI

4.5 Finden der besten Themenanzahl

Ein Problem bei all diesen Methoden, ist die richtige Anzahl an Themen herauszufinden. Was in diesem Zusammenhang richtig bedeutet, ist schwierig zu beurteilen. Einerseits können die Verfahren als Clusteringverfahren angesehen werden und ähnlich zu diesen die Stabilität der gefundenen Cluster vergleichen werden. Andererseits muss dieses nicht immer dem menschlichen Empfinden entsprechen.
Im Rahmen der Projektarbeit habe ich mich bereits dieser Thematik gewidmet und mich dabei mit der Methode von Greene et al. (2014) beschäftigt. Diese vergleicht die verschiedenen Anzahlen an Themen mittels einer berechneten Stabilität.

4.5.1 Vergleich zweier Modelle

Ein wesentliches Ziel ihrer Arbeit war es, eine Vergleichsmöglichkeit zu schaffen, die unabhängig vom Verfahren ist, welches zwei Modelle mit gleicher Anzahl an Themen vergleicht. Erreicht wird dies, indem eine feste Anzahl an Top-Wörtern, also die Wörtern die ein Thema am stärksten ausmachen, vergleichen. Dafür wurde der Jaccard-Index (Levandowsky und Winter, 1971) verwendet, mit dem es möglich ist die Ähnlichkeit zweier Mengen zu vergleichen. Der Jaccard-Index γ_d beschreibt das Verhältnis der Kardinalität der Schnittmenge zweier Mengen A und B zur Vereinigung beider, also $\gamma(A, B) = \frac{|A \cap B|}{|A \cup B|}$
.
Für extrahierte Themen mit t Top-Wörtern spielt jedoch auch die Reihenfolge der Wörter eine Rolle, weshalb Greene et al. (2014) den sogenannten *Average Jaccard* einführen. Dieser vergleicht Teilmengen der sortieren Mengen von Topwörtern, angefangen mit dem am stärksten gewichteten Top-Wort bis alle t Top-Wörter enthalten sind. T_i beschreibt dabei das i-te Thema und $T_{i,d}$ die sortierte Menge der d Top-Wörter, wobei $0 \leq d \leq k$ ist:

$$AJ(T_i, T_l) = \frac{1}{t} \sum_{d=1}^{t} \gamma(T_{i,d}, T_{j,d})$$

Dieses Maß ermöglicht einen Vergleich zweier Top-Wort-Mengen unter Berücksichtigung der Reihenfolge. Stimmen beispielsweise die ersten 6 Top-Wörter zweier Themen zwar überein, jedoch in umgekehrter Reihenfolge, wäre der Jaccard-Index 1, der *Average Jaccard* hingegen $0,3056$.
Werden nun paarweise alle k Themen zweier Modelle S_x und S_y mittels *Average Jaccard* verglichen, entstehen so k^2 Werte. Da die Themen nicht in gleicher Reihenfolge stehen, entsteht das Problem eine Permutation zu finden, sodass der Durchschnitt über alle Werte maximal ist. Dies kann in $O(k^3)$ mit dem Kuhn-Munkres-Algorithmus gelöst werden. Das Ergebnis dieser Permutation wird mit π bezeichnet und findet Verwendung im folgendem Maß:

$$agree(S_x, S_y) = \frac{1}{k} \sum_{i=1}^{k} AJ(T_i(S_x), \pi(T_i(S_y)))$$

Das $agree$-Maß ist also der Durchschnitt der AJ Werte der besten Permutation von Themen und je größer der Wert, desto ähnlicher sind sich zwei Modelle.

4.5.2 Algorithmus

Die als beste empfohlene Themenanzahl, ist das k mit der größten Stabilität, deren Berechnung im Folgenden erklärt wird. Die Stabilität eines Modells mit k Themen beschreibt, wie gut diese k Themen in jeden der Teilmengen wiedergefunden werden können. Dafür werden von der Menge der Dokumente τ voneinander verschiedene, gleichgroße Teilmengen gebildet, die $\beta \cdot n$ Dokumente enthalten mit $0 \leq \beta \leq 1$. Diese Teilmengen müssen dabei nicht disjunkt sein. Dafür wird für jedes k in einem vorgegebenen Intervall sowohl ein Modell auf dem gesamten Datensatz als auch auf den Teilmengen berechnet. Über die Werte des $agree$-Maßes zwischen dem auf dem gesamten Datensatz berechneten Model zu jedem der auf einer Teilmenge berechneten Modelle wird der Durchschnitt berechnet. Dieser wird als Stabilität bezeichnet.
Das k mit der größten Stabilität ist dann die empfohlene Anzahl an Themen für diesen Datensatz. Greene et al. (2014) empfehlen Werte von $\beta = 0.8$ und $\tau = 100$ und, dass die Berechnung immer auf $t = 20$ Termen beruhen soll.

4.5.3 Evaluation

In der Evaluation von Greene et al. (2014) wurde das Verfahren nur mit NMF verwendet. Zudem ist das Intervall der Themen auf eine Anzahl 2 bis 12 begrenzt, wobei die Begrenzung an die Ground Truth angepasst wurde. Die Ground Truth der verwendenden Datensätze, welche aus Zeitungs- und Onlineartikeln bestehen, geben die Autoren selbst vor. Auch entsprechen die vom Algorithmus als stabilste gefundenen Themenanzahlen oft weder der Ground Truth noch einem subjektivem Empfinden. Die empfohlenen Themenanzahlen sind dabei auffallend gering, wobei zwei Themen fast immer am stabilsten sind. Eine automatische Auswahl ohne überlegter Einschränkung des Intervalls scheint somit schwierig.
Zu erwähnen ist auch die lange Laufzeit. Für jedes k müssen $\tau+1$ Modelle berechnet und vergleichen werden. Für hinreichend große Datensätze ist dies bei $\beta = 0.8$ und $\tau = 100$ sehr rechenintensiv.
Die Abbildungen 7 und 8 zeigen dabei die überprüften Stabilitätswerte für Themen zwischen 2 und 25. Zu beachten ist, dass aufgrund der langen Laufzeit, das Programm mit LDA beim Datensatz des Raddialogs Bonn nicht komplett gelaufen ist. Auffallend sind bei allen Datensätzen vor allem zwei Dinge. Die Problematik, dass zwei Themen am stabilsten sind, kann weitestgehend reproduziert werden, weshalb eine Anpassung des zu untersuchenden Intervalls sinnvoll erscheint. Außerdem sind die Modelle bei probabilistischen Verfahren wie PLSI und LDA weniger stabil, was in der Tatsache begründet ist, dass diese probabilistisch sind und somit mit Zufall arbeiten. Die geringere Stabilität deckt sich mit der Beobachtung, dass sich Ergebnisse dieser Verfahren bei verschiedenen Durchläufen weniger ähneln, als bei den nicht probabilistischen Verfahren.
Ein System, welches eine optimale Anzahl an Themen selbständig bestimmt, bietet für Online-Partizipationsverfahren Vorteile. Allerdings ist die Wahrnehmung, welche Themenanzahl korrekt ist, sehr subjektiv. Vergleicht man die hohe Laufzeit mit dem Nutzen, ist dieses Vorgehen wenig empfehlenswert.
Das $agree$-Maß kann jedoch auch dazu verwendet werden zwei Modelle unterschiedlicher Verfahren zur Themenextraktion oder unterschiedlicher Datensätze zu vergleichen.

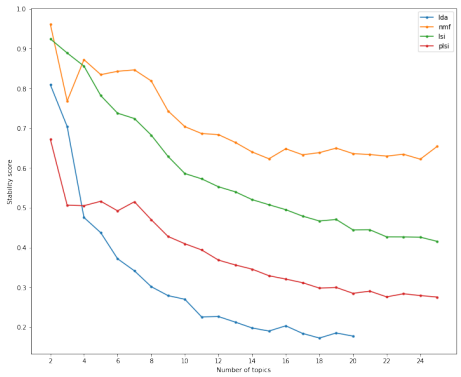


Abbildung 7: Stabilität der einzelnen Methoden angewendet auf den Raddialog Bonn.

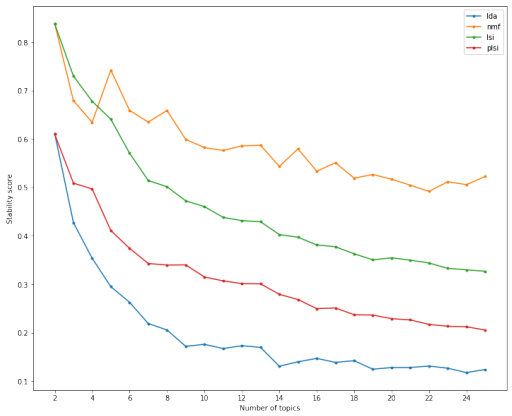


Abbildung 8: Stabilität der einzelnen Methoden angewendet auf den Kölner Bürgerhaushalt 2015.

Schlussfolgerung: Steht viel Rechenleistung zur Verfügung und ist eine maschinelle Festlegung auf ein Thema notwendig, ist dieser Ansatz intuitiv, bedarf jedoch einer gewissen Überlegung wie das Intervall gewählt werden soll. Eine manuelle Auswahl der Themenanzahl ist aber, auch wegen des subjektiven Charakters der Beurteilung der Güte, zu bevorzugen.

4.6 Interactive Topic Modeling

Hu et al. (2014) liefern einen Ansatz, LDA um *Constraints* zu erweitern, die auf Wortebene angewendet werden, womit baumbasierte multinominale Wahrscheinlichkeitsverteilungen erreicht werden. Wörter, welche zusammen berücksichtigt werden sollen, werden durch eine Constraints verbunden. Stellt man das Bag-of-Words Modell als Baum erster Ordnung da, dessen Blätter die Terme darstellen, können diese Blätter nun durch Constraints ersetzt werden, falls für die betreffenden Terme ein Constraint vorliegt. Diese Constraints sind selber Bäume erster Ordnung, dessen Blätter die Terme repräsentieren, die Contraints unterliegen. Anzumerken ist, dass die Constraints hier transitiv sind. Wenn t_a und t_b sowie t_b und t_c von einem Constraint betroffen sind, sind dies auch t_a und t_c. Folglich würden alle drei Terme dem gleichen Constraint angehören. Jeder Constraint tritt dann in der Iteration an die Stelle der Terme die er vereint. Dabei hat jeder Constraint eine Wahrscheinlichkeitsverteilung seiner Terme über die Themen.
Mit Hilfe dieser Constraints ist es möglich, dass der Benutzter interaktiv Themen verändert, indem er Wörter hinzufügt oder entfernt. Da LDA generativ ist, können weitere Iterationsschritte einfach mit den neuen Constrains durchgeführt werden, um so ein neues Ergebnis zu erhalten. Auch ist es möglich Constraints aus anderen Quellen zu lernen und zu verwenden.
NMF kann auch interaktiv verwendet werden, wie Kuang et al. (2015) zeigen. Sie verwenden eine veränderte Optimierungsfunktion und aktualisieren auch die Zugehörigkeitswerte mit anderen Regeln. Die Grundidee von NMF bleibt jedoch erhalten. Mit den veränderten Berechnungen sind fünf verschiedene Interaktionsmöglichkeiten gegeben,

nach deren Anwendung weitere Iterationsschritte erfolgen.
Sollen die Terme die ein Thema definieren verändert werden, kann die Matrix W verändert werden. Dazu werden die Gewichte der vom benutzter ausgewählten Terme entweder auf null gesetzt, wenn diese nicht im Thema enthalten sein sollen, oder angepasst. Diese Anpassung erfolgt, wenn der Benutzer die Reihenfolge eines Terms verändert und so den Zugehörigkeitsgrad des Terms zu dem betreffenden Thema in W erhöht bzw. verringert.
Zwei Themen können zusammengefügt werden, indem die Zugehörigkeiten der Dokumente zu den Themen in H gehärtet betrachtet und die Dokumente identifiziert, die somit zu einem der beiden Themen gehören, die vereint werden sollen. Danach werden die Zugehörigkeitswerte für diese Dokumente addiert. Ob sich die Themenanzahl dadurch verändert oder nicht, lassen die Autoren offen.
Weiterhin ist es auch möglich ein Thema in zwei aufzuteilen. Dazu wird der Vektor des Themas in W und H dupliziert. Der Benutzer verändert dann durch Anpassung der Terme die beiden neuen Themen.
Themen können auch auf der Grundlage einer Menge von Dokumenten oder Termen erstellt werden. Dazu werden entsprechende Vektoren in die Matrizen eingefügt, in denen die ausgewählten Dokumente bzw. Terme jeweils den Zugehörigkeitsgrad eins zum neu erstellten Thema haben. Danach erfolgt wieder ein weiterer Iterationsschritt.
Schlussfolgerung: Topic Modeling kann auch interaktiv eingesetzt werden. Jedoch sind die dazu verwendeten Verfahren komplexer als die zugrundeliegenden. Aufgrund der ständigen Neuberechnung, ist der Einsatz für Online-Partizipationsverfahren gut zu überlegenen. So sind die Anzahl der Dokumente, vorhandene Rechenleistung und Anzahl der Benutzer Faktoren die bei der Überlegung eine Rolle spielen.
Generell aber bietet dieser interaktive Ansatz auch die Möglichkeit, dass jeder Nutzer ein Basismodell mit wenig Iterationsschritten individualisieren kann.

4.7 Problematik bei Online-Partizipationsverfahren

Generell ist es sinnvoll vor dem Einsatz solcher Verfahren die Datengrundlage zu begutachten. Online-Partizipationsverfahren können, wie beispielsweise die Raddialoge, sehr zielgerichtet sein. Dies macht es notwendig zu hinterfragen, ob die Themen in den Dokumenten gut genug trennbar sind. Es ist schwierig semantische Strukturen in Dokumenten zu finden, wenn diese wenig verschiedene semantische Strukturen aufweisen.
Ein weiter zu bedenkender Aspekt liegt in der Form der Online-Partizipationsverfahren. Mitunter sind Vorschläge, und damit die Dokumente, sehr kurz und beinhalten nur wenige Wörter. Damit diesem Dokument die richtigen Themen zugewiesen werden können, muss das eingesetzte Verfahren sehr präzise arbeiten. Auf der anderen Seite stehen lange Vorschläge mit vielen Kommentaren. Diese können gleich mehrere Themen behandeln. Oder aber der Diskurs in den Kommentaren wird nur über einen Aspekt geführt, was ebenfalls die Themen dieses Dokuments verfälscht.
Die üblichen in der Forschung eingesetzten Datensätze zum Topic Modeling entstammen Zeitungen oder Wikipedia. Meistens sind Dokumente aus diesen beiden Spektren nicht nur länger, sondern viel themenbezogener als bei Online-Partizipationsverfahren. Neben der Tatsache, dass diese Dokumente formaler geschrieben sind, unterliegen sie in der Regel einer Systematik der Kategorisierung. So steht schon das Erstellen im Zusam-

menhang mit beispielsweise einer Kategorie bei Wikipedia oder Rubrik einer Zeitung. Zusätzlich unterliegen diese Texte einer strengen Qualitätskontrolle, anders als Vorschläge oder Kommentare bei Online-Partizipationsverfahren, die diese Kategorisierung einigermaßen sicherstellt.
Im Gegensatz zur Forschung kann der Datensatz bei Online-Partizipationsverfahren nicht nach zuvor definierten Kriterien ausgewählt oder angepasst werden, da die extrahierten Themen die gesamten Dokumente repräsentieren sollen. Dazu kommt die Problematik, dass Online-Partizipationsverfahren oft nicht über eine hinreichend große Anzahl an Dokumenten verfügen, um stabile Themen extrahieren zu können.

5 Visualisierung

Daten wurden schon visualisiert, als noch keine Computer gab. Eine Visualisierung kann zum Einen eine Übersicht über große Datenmengen schaffen, zum anderen können gefundene Strukturen dargestellt werden. Auch kann eine Visualisierung Menschen ermöglichen, Strukturen oder Besonderheiten der Daten zu erkennen. Die Arten der Visualisierung und deren Möglichkeiten sind vielfältig und richten sich nach Art und Struktur der Daten, Anforderungen der Benutzer und nicht selten Ästhetik. Dabei sollte die Verwendung durch den Menschen im Vordergrund stehen und zu dessen Verständnis beitragen. Visualisierungen könne interdisziplinär unter verschiedensten Aspekten betrachtet werden, wobei diese Arbeit nur einen kleinen Ausschnitt über die Möglichkeiten bieten kann. Der Hauptfokus der zu visualisierenden Daten liegt auf Texten, aber auch statistische Daten sind verwendbar. Kucher und Kerren (2015) bieten eine interessante Übersicht über Techniken, die zur Visualisierung von Text verwendet werden können und stellen ein Online-Tool dafür bereit[3].
Die Daten die bei der Online-Partizipation anfallen, sind hauptsächlich Textbeiträge, aber etwa auch Abstimmungsdaten. Dazu gibt es auch erhobene Zeitpunkte, etwa wann ein Beitrag erstellt wurde. Je nachdem wie viel Einsicht in ein System vorliegt, können auch Zeitpunkte von Abstimmungen erhoben oder das Suchverhalten protokolliert werden. Analog zu Cao und Cui (2016) können so folgende Ebenen der Visualisierung bei der Online-Partizipation identifiziert werden: Gesamtheit aller Dokumente, Dokumentebene, Wortebene und Themenebene. Außerdem können zeitliche Verläufe und Suchanfragen visualisiert werden.
Im folgenden werden verschiedene Techniken vorgestellt und diskutiert, auf welchen Ebenen diese im Bezug auf Online-Partizipationsverfahren verwendet werden können.

5.1 Histogramme

Histogramme bilden eine recht einfache, aber übersichtliche Visualisierung. Dabei werden Datenpunkte eines kontinuierlichen Wertebereiches in Intervalle unterteilt, dessen Wahrscheinlichkeitsverteilungen über eine Variable dargestellt werden. (Velleman und Hoaglin, 1981). Diese Intervalle, Klassen genannt, werden über den gesamten Wertebereich der Datenpunkte gebildet. Die Addition der Häufigkeiten ergibt dann die Gesamtanzahl aller Datenpunkte. Dabei können entweder die absoluten oder die relativen Häufigkeiten betrachtet werden.
Jedes Intervall wird nun als Balken dargestellt, dessen Fläche die Häufigkeit der im Intervall enthaltenen Datenpunkte angibt. Die Breite ist abhängig von der Größe des Intervalls. Häufig werden die Intervalle gleich groß gewählt, was in einer einheitlichen Breite und somit Übersichtlichkeit resultiert.
Zwei Histogramme können auch verglichen werden, wozu es verschiedenste Distanzmaße gibt (Cha und Srihari, 2002). Diese praktische Eigenschaft ist aber zur Visualisierung nicht notwendig, weshalb an dieser Stelle nicht weiter darauf eingegangen wird.
Wesentlich für Histogramme sind die Klassen, dessen Anzahl nach verschiedenen Methoden bestimmt werden kann. Daraus lässt sich dann die breite der Intervalle, also

[3] http://textvis.lnu.se/

Klassen bestimmen. Diese Methoden nehmen zur Einteilung jedoch immer eine gewisse Verteilung der Daten an, weshalb sie nicht immer die optimale Anzahl an Klassen finden. Weitverbreitet sind die Methoden von Sturges (1926) oder Scott (1979) die jedoch die Breite der Klassen berechnen.
Im Zusammenhang mit Online-Partizipation können Histogramme eine Übersicht über numerische Werte bieten. Etwa Abstimmungsdaten von Vorschlägen, Anzahl an Kommentaren der Vorschläge, Anzahl der Wörter in einem Dokument oder auch Nutzungsdaten von Benutzern, falls diese erhoben werden.

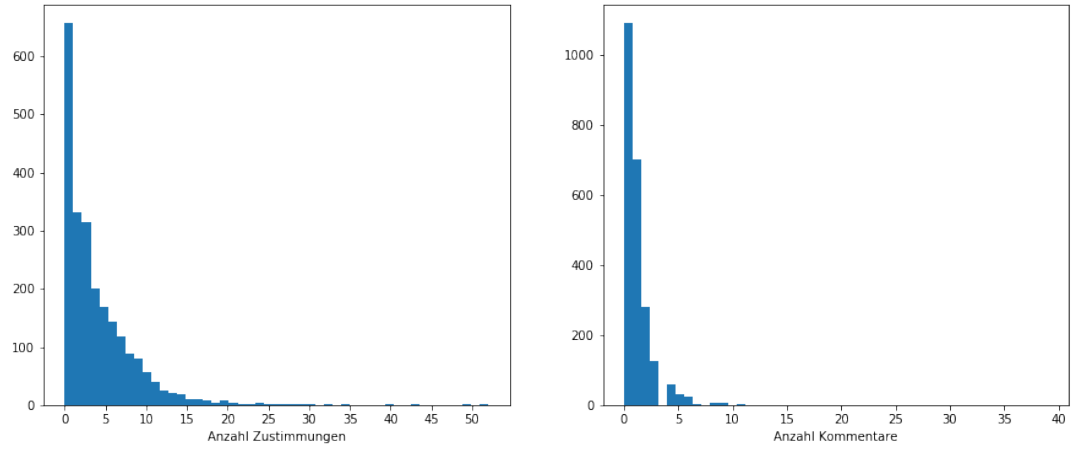


Abbildung 9: Histogramme über die 2331 Vorschläge des Bonner Rad-Dialoges.

5.2 Word Clouds

Eine relativ neue Visualisierung sind Word Clouds (Feinberg, 2010), die Wörter in einer Art Wolke darstellen. Die Größe der Worte wird durch ein Gewicht bestimmt. Dies kann die Häufigkeit eines Wortes sein und analog zu Kapitel 3.3.1 bestimmt werden. Es bietet sich etwa $tfidf$ an, oder tf unter Verwendung des Logarithmus, damit häufige Wörter nicht zu stark dominieren. Außerdem können die Worte auch verschiedene Farben haben, welche hauptsächlich der einfacheren Unterscheidung dienen.
Wesentlich ist die Anordnung der Worte innerhalb der Wolke, da dies die Wahrnehmung der Wolke beeinflusst. Lohmann et al. (2009) haben verschiedene Ansätze evaluiert und kommen zu dem Ergebnis, dass die Anordnung in einer Word Cloud von ihrem Verwendungszweck abhängt. Soll diese dazu dienen die Suche nach einem Wort zu vereinfachen, empfehlen sie eine sequentielle, alphabetisch geordnete Anordnung. Wird eine Word Cloud hingegen verwendet, um die häufigsten Begriffe zu visualisieren, sei eine zirkuläre Ausrichtung mit den häufigsten Begriffen im Zentrum zu bevorzugen. Schließlich zeigen Lohmann et al. (2009) auch die Möglichkeit auf, Word Clouds nach Themen zu strukturieren.
Für Online-Partizipationsverfahren können Word Clouds die Häufigkeit von Wörtern auf verschiedenen Ebenen visualisieren. Word Clouds können das gesamte Verfahren oder Beiträge, aber auch gefundene Themen darstellen. Denkbar ist auch die Darstellung von Suchergebnissen, die aber wenig sinnvoll erscheint da der Suchende an konkreten

Ergebnissen interessiert ist. Bei dem Einsatz von Word Clouds sollte die Zweckmäßigkeit im Auge behalten werden.

Abbildung 10: Word Cloud des Bonner Rad-Dialoges mit logarithmischer Gewichtung.

5.3 Visualisierungen mittels nicht-linearer Dimensionsreduzierung

Der nächste Teilbereich von Visualisierungen ist vergleichsweise komplex und viel beschrieben. Da Daten nur in zwei Dimensionen verständlich dargestellt werden können, verwenden viele nicht-lineare Ansätze eine Dimensionsreduzierung, um mehrdimensionale Daten in einem zweidimensionalen Raum darzustellen. Dafür werden die Daten zuerst auf zwei Dimensionen reduziert und dann mit einem Streudiagramm dargestellt. Es gibt einige Verfahren, wie das von Sammon (1969), Silva und Tenenbaum (2003) oder Roweis und Saul (2000). Als *state of the art* wird t-SNE von Maaten und Hinton (2008) angesehen. Ein relativ neues Verfahren, UMAP von McInnes und Healy (2018), verspricht eine bessere Performance als t-SNE bei geringerer Laufzeit. All diese Verfahren nehmen an, dass Datenpunkte eine Mannigfaltigkeit teilen und verwenden mathematische Verfahren, um diese abzuschätzen. UMAP beispielsweise verwendet die Riemannsche Geometrie zusammen mit Fuzzy-Sets (Klir und Yuan, 1995).
Für Online-Partizipationsverfahren sind theoretisch mehrere Möglichkeiten des Einsatzes möglich, wobei alle mit latenten Themen innerhalb der Beiträge arbeiten. So könnten nicht-lineare Dimensionsreduzierung verwendet werden, um Themen zu finden (analog zum linearen Topic-Modeling) und gegebenenfalls darzustellen. Außerdem könnten bereits reduzierte Vektoren und somit gefundene Themen (vgl. Kapitel 4) im zweidimensionalen Raum dargestellt werden. Die zweite Verwendung erscheint dabei deutlich sinnvoller, vor allem wegen der hohen Laufzeit der Reduzierung. Da die zweidimensionale Reduzierung die Dokumente als kleine Punkte visualisiert, kann dies die intuitive Verwendung durch den Benutzer beeinträchtigen.
Wie in Abbildung 11 zu sehen ist, können dabei unübersichtliche Graphen entstehen, die

wenig Erkenntnisgewinn bringen. Der Graph ist interaktiv, sodass Daten über jedes Dokument angezeigt werden, wenn die Maus über den entsprechenden Punkt bewegt wird. Jedoch ist diese Graphik zu irritierend, um bei Online-Partizipationsverfahren sinnvoll verwendet werden zu können.

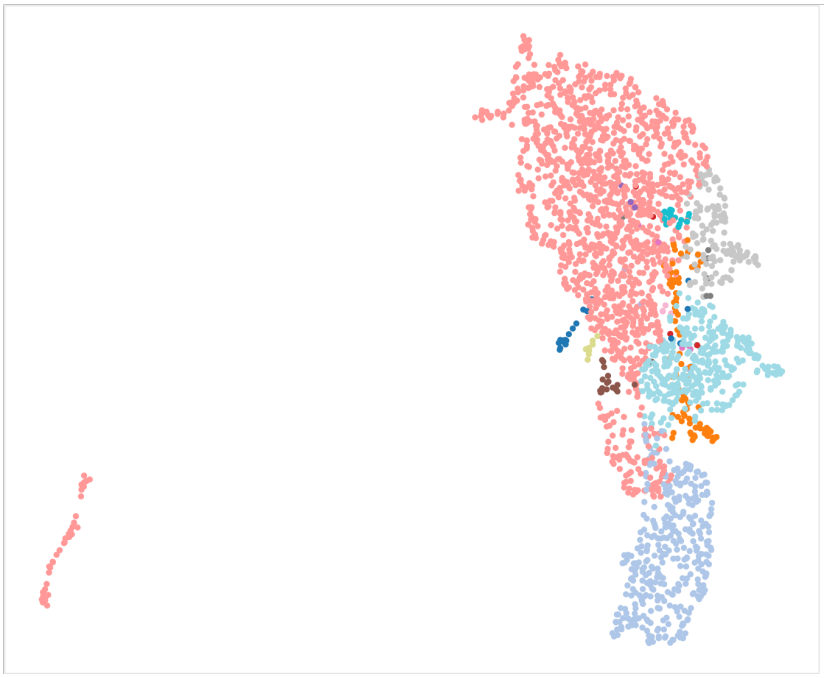


Abbildung 11: Screenshot eines UMAP-Plots des Raddialog Bonns. Jede Farbe repräsentiert ein Thema und jeder Punkt ein Dokument. Verwendet wurde LDA mit einer Zerlegung in 20 Themen.

5.4 Streudiagramm

Neben der Darstellung von dimensionsreduzierten Daten kann ein Streudiagramm eine Vielzahl von zweidimensionalen Daten Visualisieren. Dabei wird jedes Dokument als Punkt in einem Graphen dargestellt. So können zwei Features gegeneinander aufgetragen und so Korrelationen erkannt werden. Abgesehen von der Position der einzelnen Datenpunkte, können die Datenpunkte für verschiedene Eigenschaften verschiedene Farben erhalten.
Neben den bereits angesprochenen Daten bspw. zur Zustimmung, können damit auch zwei Textkorpora verglichen werden, wie Kessler (2017) aufzeigt. Die entwickelte Software trägt den Namen *Scatteretext*[4] Die Hauptidee ist, dass jedem Korpus eine Achse zugewiesen wird, die die relative Häufigkeit eines Terms beschreibt. So bestimmt sich die Position eines Terms durch die relative Häufigkeit in beiden Korpora, wodurch sich ebenfalls das Verhältnis der beiden Häufigkeiten erkennen lässt. In den beiden Ecken oben links bzw. unten rechts, sind die Terme verzeichnet, die die Korpora vom jeweils anderen unterscheiden. Die Farbe ändert sich in Abhängigkeit zur Distanz zu einer der

[4] https://github.com/JasonKessler/scattertext

Ecken. Außerdem wird an jedem Punkt der Term notiert, was erheblich zur Verständlichkeit beiträgt.
Neben der Visualisierung von nicht-linearer Dimensionsreduktion, werden Streudiagramme eingesetzt, um Korrelationen zwischen zwei numerischen Features zu visualisieren. *Scattertext* macht vor allem Sinn, wenn zwei Online-Partizipationsverfahren, etwa aus verschiedenen Jahren oder Städten, verglichen werden sollen. Denkbar ist auch ‚dass zwei Korpora vergleichen werden die jeweils mehrere Online-Partizipationsverfahren zusammenfassen.
Theorietisch können auch zwei Vorschläge oder Themen vergleichen werden, jedoch bieten sich hierfür bessere Visualisierungsformen an. Dafür würden die Begriffe eines Themas anstatt mit der Häufigkeit mit der Zugehörigkeit zu diesem Thema aufgetragen. Sowohl das Vergleichen zweier Vorschläge als auch zweier Themen ist mit *Scattertext* unverhältnismäßig aufwendig, da immer nur zwei Vorschläge bzw. Themen gleichzeitig verglichen werden können.
Wie in Abbildung 12 zu sehen ist, lassen sich relativ einfach die Unterschiede der Raddialoge erkennen. Fährt ein Benutzter mit der Maus über einen Datenpunkt, wird der entsprechende Term sowie das Häufigkeitsverhältnis dargestellt. Auf dem Praxissymposium „Online-Partizipation in Kommunen“ 2018 fanden die meisten Teilnehmenden diese Visualisierung jedoch wenig zusagend.

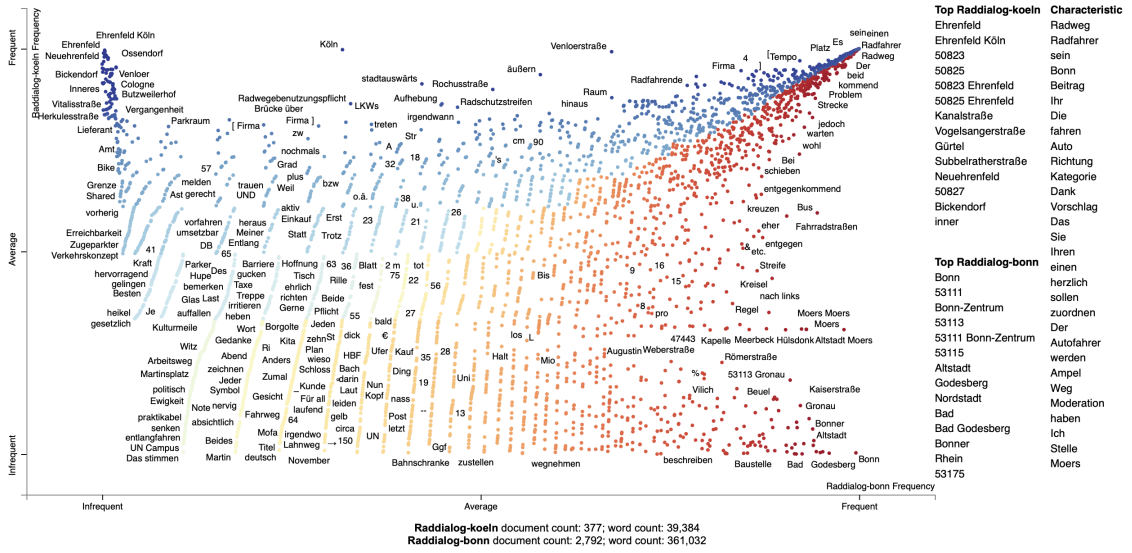


Abbildung 12: Screenshot von *Scattertext*: Vergleich der Online-Partizipationsverfahren der Raddialoge in Bonn und Köln-Ehrenfeld.

5.5 Netzdiagramm

Netzdiagramme stellen die Ausprägungen verschiedener Dimensionen in einer runden Form da. Dabei wird jede Dimension auf einer eigenen Achse um den Mittelpunkt eines Kreises herum aufgetragen (Tague et al., 2005). Zwar werden die Punkte auf den einzelnen Achsen verbunden, jedoch stehen die benachbarten Achsen nicht beabsichtigt

nebeneinander. Sollen in einem Diagramm mehrere Objekte dargestellt werden, sollten diese visuell unterscheidbar sein, etwa durch verschiedene Farben (Lydiard et al., 2010). Allerdings sollten damit nicht zu viele Objekte vergleichen werden, weshalb häufig auch Diagramme für verschiedene Objekte nebeneinander gezeigt werden, wobei die Position der Achsen in jedem Diagramm gleich sein sollte.
Daten von Online-Partizipationensverfahren können damit in vielfältiger Hinsicht verglichen werden. So können Netzdiagramme dazu genutzt werden einzelne Aspekte des Verfahrens zu visualisieren, beispielsweise die Verteilung von Vorschlägen oder Kommentaren auf die einzelnen Wochentage (siehe Abbildungen 13). Diese zu vergleichenden Aspekte, also Dimensionen, werden anders als bei Histogrammen jedoch vorher ausgewählt. Verwendet werden Netzdiagramme auch, um die Themenverteilung eines Dokuments oder einer Suchanfrage zu visualisieren (Sasaki et al., 2014). Dabei wird an jede Achse ein für das Thema aussagekräftiger Begriff vermerkt. Diese aussagekräftigen Begriffe können auch mit *Topic Labeling* (Mei et al., 2007) gefunden werden.
Schließlich können mit Netzdiagrammen auch verschiedene Dokumente, Dokumentenmengen oder gesamte Online-Partizipationen visuell vergleichen werden.

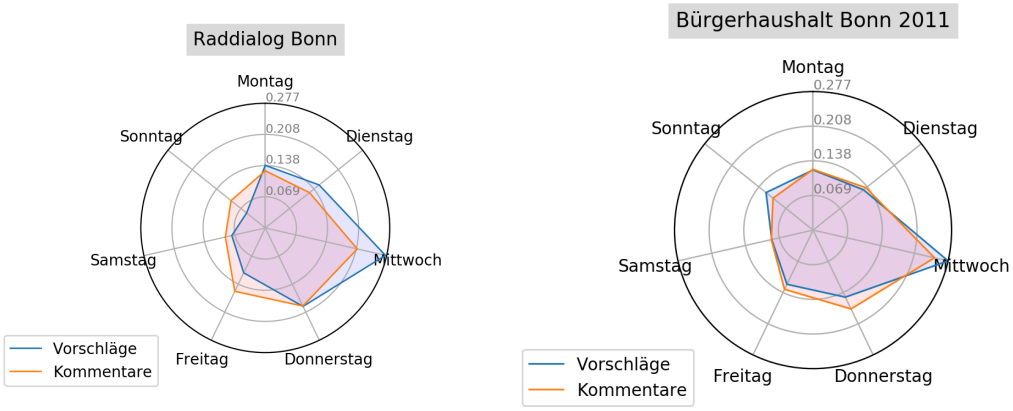


Abbildung 13: Relative Häufigkeiten der Vorschläge und Kommentare des Raddialog Bonn und des Bonner Bürgerhaushalt 2011 an den Wochentagen.

5.6 Visualisierung von extrahierten Themen

Extrahierte Themen zu visualisieren hat vor dem Hintergrund dieser Arbeit einen besonderen Stellenwert. Es gibt auch Arbeiten, die sich hauptsächlich damit beschäftigen Themen in unterschiedlicher Weise visualisieren, visualisierend vergleichen oder damit Interaktion visualisierend zu Unterstützen. Davon werden hier einige vorgestellt.
Murdock und Allen (2015) stellen eine Visualisierung von Ähnlichkeitssuchen unter Verwendung eines Topic Models vor. Dies ermöglicht dem Benutzer ähnliche Dokumente zu einem von ihm ausgewählten zu finden. Visualisiert werden die ähnlichsten Dokumente mit einem nach Ähnlichkeit sortierten Balkendiagramm. Die Breite der waagerechten Balken für jedes Dokument richtet sich nach der Ähnlichkeit zu dem angegebenen Dokument. Innerhalb der Balken wird die Themenverteilung des Dokumentes mit farblichen Segmenten angegeben, die jeweils ein Thema repräsentieren. Außerdem ist in jedem Balken der Titel des korrespondierenden Dokuments angegeben. Die Visualisierung ist zu-

sätzlich interaktiv in dem Sinne, dass auf die farblichen Blöcke geklickt werden kann oder die Top-Wörter der jeweiligen Themen angezeigt werden können. Der Charakter dieser Visualisierung ist mehr von unterstützender Natur und eher für Korpora geeignet, wo Dokumente leicht anhand ihres Titels identifiziert werden können. Somit erscheint eine Verwendung der Visualisierung sinnvoll, die dem Benutzer ähnliche Dokumente zu dem aktuell betrachtetem aufzeigt und dabei gefundene Themen mit darstellt. Denkbar ist auch die Ähnlichkeit einer Anfrage zu Dokumenten in dieser Weise zu visualisieren.
LDAvis, welches von Sievert und Shirley (2014) entwickelt wurde, kombiniert im wesentlichen zwei Darstellungformen. Eine davon ist, die Themen anhand ihrer Ähnlichkeit im zweidimensionalen Raum darzustellen. Dazu wird eine Distanzmatrix der Dokumente mit einem dimensionsreduzierenden Verfahren, z.B. PCA, auf zwei Dimensionen reduziert (Chuang et al., 2012). Dargestellt wird jedes Thema in diesen zwei Dimensionen als Kreis, dessen Fläche sich nach dem Einfluss des Themas auf den Korpus bezieht. Kreise können sich überlappen und auch komplett in anderen Kreisen liegen, wobei kein Zusammenhang zwischen Einfluss des Themas und der Lage erkennbar ist. Themen können so andere überlagern, auch wenn sie sich wenig ähneln, nur falls eines einen großen Einfluss hat. Dieses Überlagern kann Hierarchie vortäuschen, die so nicht aus der Berechnung abgeleitet werden kann.
Die Autoren schlagen ebenfalls Clustering innerhalb dieses zweidimensionalen Raums vor.
Die andere Darstellungsform zeigt ein Balkendiagramm der Top-Words eines ausgewählten Themas an. Jedes Wort hat zwei Balken, einen der den Anteil am Thema und einen der die Häufigkeit im Korpus wiedergibt. Bewegt man die Maus über einen Begriff, wird im Graph der Einfluss auf die Themen durch Veränderung der Fläche der Kreise angezeigt.
Grundsätzlich ist *LDAvis* auch für Online-Partizipationsverfahren geeignet, jedoch sollte die Art der Verwendung auch unerfahrene Benutzer nicht überfordern.
Ein Verfahren Begriffe innerhalb eines Themas zu visualisieren, haben Smith et al. (2014) vorgestellt. Dazu werden kräftebasierte Zeichenverfahren (Fruchterman und Reingold, 1991) unter Einbeziehung von statistischen Daten und Daten der Themen verwendet. Jedes Thema wird als eigener Graph dargestellt, wo eine ausgewählte Anzahl von Begriffen die Knoten darstellen. Kanten verbinden zwei Knoten nur, wenn der Kookorenzwert der dazugehörigen Begriffe hoch genug ist, sie also häufig genug zusammen auftreten. Zusätzlich beschreibt die Fläche der Knoten den Zugehörigkeitswert eines Begriffs zu einem Thema. Die Graphen der Themen werden dann nach Kovarianz zwischen den Themen angeordnet.
Diese Art der Visualisierung ist nur begrenzt sinnvoll, da sie schon bei einer begrenzten Anzahl an Themen und kräftebasierten Graphen mit vielen Kanten unübersichtlich werden können.
Schlussfolgerung: Visualisierungen für extrahierte Themen gibt es viele, die jedoch oft komplex sind und selten zusätzlichen Aufschluss über die Themen selbst liefern. Vielmehr werden die Beziehungen zwischen den Themen visualisiert, nicht selten in einer Form die einen erfahrenen Benutzer benötigt. Ob diese Visualisierungen einer rein textuellen Beschreibung von Themen, in Form von Wortlisten überlegen ist, hängt sehr von den Umständen ab.

5.7 Simple Formen der Visualisierung

Sinnvoll sein kann der Einsatz von simplen Formen der Visualisierung. Diese sind nicht selten intuitiver oder werden auch ohne viel Vorwissen verstanden.
Balkendiagramme sind nicht die schlichteste Visualisierungsform, jedoch lassen sie sich analog zu Netzdiagrammen verwenden, so dass sich ähnliche Überlegungen ergeben. Anders als Histogramme, visualisieren Balkendiagramme feste Variablen, dessen Ausprägungen dargestellt werden. Auch müssen Balkendiagramme nicht die Häufigkeit darstellen. Wie bereits erörtert, können Balkendiagramme auch horizontal angewendet werden und die Balken aus mehreren Segmenten bestehen.
Eine viel verwendete und leicht zugängliche Methode extrahierte Themen zu visualisieren, sind die jeweiligen Top-Wörter als Liste darzustellen, absteigend nach Zugehörigkeit zum Thema. Solche Wortlisten müssen nicht weiter erklärt werden.

5.8 Schlussfolgerung

Betrachtet man die vorhandenen Methoden zur Visualisierung fällt ins Auge, dass vor allem Metadaten visualisiert werden können. Numerische Daten wie Abstimmungszahlen, Anzahl an Kommentaren oder zeitliche Daten können mit gängigen Methoden visualisiert werden. Dazu gehören Graphen, Diagramme und Histogramme.
Texte zu visualisieren ist dagegen eine Herausforderung. Einerseits können simple Wortlisten oder Wortwolken verwendet werden, andererseits können Methoden angewendet werden, um extrahierte Themen zu visualisieren. Diese Visualisierungen sind oft komplex oder bieten wenig Vorteile gegenüber einer textbasierten Darstellung.
Schließlich kann eine gute Visualisierung auch bedeuten, Suchergebnisse oder Beiträge verständlich und übersichtlich darzustellen.

6 Das entwickelte System

Einige der vorgestellten Methoden wurden in einer Anwendung implementiert, welche eine Analyse von Online-Partizipationsverfahren unterstützen soll. Dies wurde als Webanwendung realisiert, die auf Dockercontainern aufbaut.
Zuerst werden die Anforderungen erläutert die an eine solche Anwendung bestehen und sodann die Architektur. Zwar gibt es einige Überlappungen in der Anwendung zwischen Themenextraktion, Visualisierung und Interaktion. Dennoch wird in den Kapiteln versucht, die Anwendung aus dem jeweiligen Blickwinkel zu betrachten. Dabei wird auch darauf eingegangen, welche Methoden verwendet wurden und welche Probleme in diesen Bereichen zu bewältigen sind.

6.1 Anforderungen

Wie der Name schon verrät, finden Online-Partizipationsverfahren im Internet, seltener in einem Intranet, statt. Dadurch ist eine Webanwendung das bevorzugte Mittel der Wahl, um eine Analyse zu implementieren. Diese Webanwendung wurde mit dem Ziel entwickelt, sowohl eigenständig als auch unterstützend eingesetzt werden zu können. Die Eigenständigkeit ist auch deshalb von Vorteil, da so bereits Verfahren betrachtet werden können, die nicht mehr online sind und nur noch die Rohdaten oder eine Datenbank existieren. Zwar könnte eine eigenständige Website mit der Anwendung auch begleitend verwendet werden, jedoch wurden Möglichkeiten eingerichtet die Webanwendung in bestehende Websites einzubinden. Dies kann über *Inlineframes* (Raggett et al., 1999) und teilweise auch *AJAX* (Paulson, 2005) geschehen.
Um eine Integration in bestehende Systeme zu erleichtern, wurde ein *Microframework* verwendet und die Architektur wurde mit *Docker* realisiert. Dadurch ist eine Anpassung oder Verschlankung leicht möglich.
Daten aus Online-Partizipationensverfahren zu bekommen ist aus verschiedenen, meist rechtlichen, Gründen schwierig. Deshalb ist ein Webcrawler, wie in Kapitel 2.1 beschrieben, Bestandteil der Architektur.
Generell wurde die Anwendung so entwickelt, dass diese auch um andere Verfahren zur Extraktion, Visualisierung oder Interaktion erweitert werden kann. Der Bedarf richtet sich dabei auch immer stark nach der Art und den Umständen der Online-Partizipation.

6.2 Architektur

Der Begriff Architektur bezieht sich in diesem Kapitel auf die Architektur des Containerverbunds. Zu diesem Zweck wurde Docker verwendet, welches eine Containervisualisierung bereitstellt. Container laufen zwar auf der gleichen Maschine, jedoch hat jeder Container seine eigenen Ressourcen, insbesondere Speicher. Für jede Anwendung sollte ein eigener Container vorgesehen werden, so dass die Container nur das enthalten, was sie zum Ausführen der Anwendung benötigen. Containervisualisierung einzusetzen hat somit den Vorteil, dass alle Abhängigkeiten und Bibliotheken im Container enthalten sind. So lassen sich konsistente Umgebungen schaffen, die einen einfacheren und von der Host-Maschine unabhängigen Betrieb gewährleisten. Diese Container lassen sich

dann fertig und einsatzbereit ausliefern.
In einem Containerverbund können sich Container mit verschiedenen virtuellen Netzwerken verbinden, so dass nur diejenigen im gleichen Netzwerk sind, die miteinander kommunizieren müssen. Die Kommunikation zwischen den Containern findet als Netzwerkkommunikation statt. Weiterhin kann festgelegt werden, welche Container von außerhalb des Verbunds erreichbar sind.
Für diese Anwendung waren ebenfalls die Modularität und Skalierbarkeit ein Vorteil. Es ist so leicht möglich eine Datenbank zu ersetzen oder auf den Crawler zu verzichten. Die Skalierbarkeit kann erreicht werden, indem Container dupliziert werden, die dann Anfragen mit Round-Robin abarbeiten. Bei der Anwendungsentwicklung muss diese Möglichkeit beachtet werden, falls Datein geschrieben werden.
Abbildung 14 zeigt einen Überblick über den Containerverbund, wobei geteilte Datensysteme und Netzwerke nicht eingezeichnet sind.
Im Folgenden wird auf wichtige Aspekte und Funktionen der Container eingegangen.

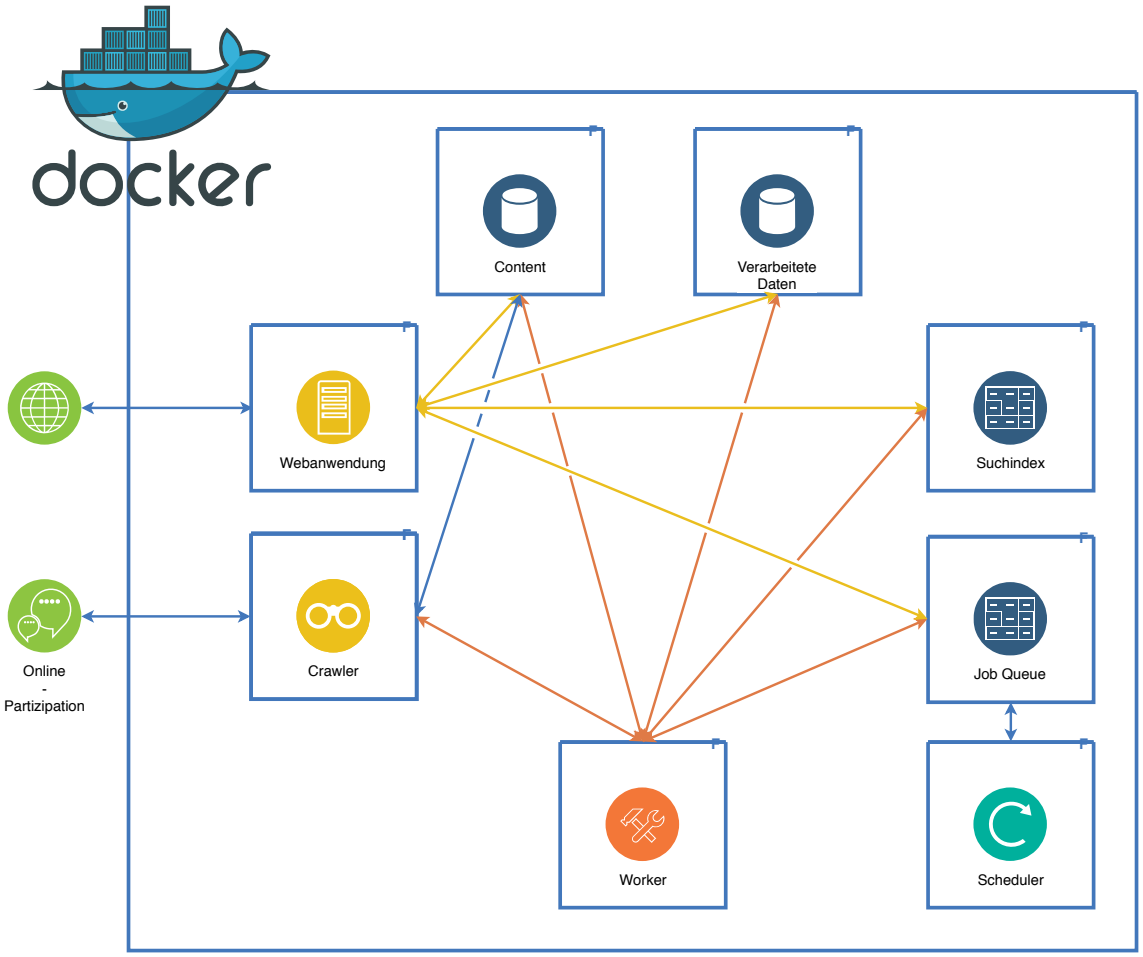


Abbildung 14: Schaubild der Container-Architektur, wobei jeder Kasten einen Container darstellt. Auf die darunterliegenden Dateisysteme und Netzwerke wurde verzichtet. Die Pfeile zeigen die Kommunikation zwischen den Containern.

6.2.1 Datenbankcontainer

In der Abbildung sind zwei Datenbanken zu sehen die als „Content“ und „Verarbeitete Daten“ bezeichnet sind, wobei erstere die Rohdaten enthält und letztere alle verarbeiteten Daten. Die Aufteilung erlaubt mehr Flexibilität, da somit die Datenbank mit den Rohdaten ausgetauscht werden kann, also direkt die Datenbank verwendet werden könnte, die zum Online-Partizipationsverfahren gehört. Dafür könnte jedoch eine Anpassung an der Webanwendung notwendig werden, um etwa Schema zu verändern.
Zu verarbeiteten Daten zählen Daten die durch NLP oder Topic Modeling erlangt wurden. So wird etwa für jedes Dokument die Annotierung gespeichert, sodass nur einmal die NLP-Pipeline durchlaufen werden muss. Hier ist anzumerken, dass die Daten einmalig bei beendeten und periodisch bei laufenden Verfahren verarbeitet werden. Die verarbeiteten Daten sind also nur immer eine Momentaufnahme. Gespeichert werden außerdem Vektorrepräsentationen der Dokumente, entweder von Bag-of-Words Modellen oder Topic Models. Ob dieses Vorgehen auch bei sehr großen Datenmengen noch die gewünschte Performance liefert, ist fraglich.
Wie in Abbildung 14 zu sehen, schreibt der Crawler die Rohdaten in Content, auf die sowohl die Webanwendung,als auch der Worker zugreifen. Auf die verarbeiten Daten greifen ebenfalls nur beide zu.

6.2.2 Crawler

Der Crawler verarbeitet Daten einer vorgeben Website (vgl. Kapitel 2.1) und speichert diese Daten strukturiert in der Datenbank. Es kann dabei angegeben werden, ob dieser Prozess einmalig oder in bestimmten Abständen stattfinden soll. Falls der Datensatz als Dokument vorliegt, etwa weil das Verfahren bereits beendet oder offline ist, kann statt dem Crawler auch ein Container gestartet werden, der die Rohdaten aus dieser Datei in die Datenbank einträgt.

6.2.3 Hintergrundprozesse

Lange laufende Prozesse wie die Berechnung von Modellen zur Themenextraktion finden nicht in der Webanwendung selbst, sondern in einem gesonderten Container statt. Dieser Container wird in Abbildung 14 als „Worker“ bezeichnet. Dieser Container verfügt in der Regel über den gleichen Quellcode wie die Anwendung, von der aus der Prozess initiiert wird.
Die gesamte Kommunikation zwischen Webanwendung und Worker findet über einen sogenannten *Message Broker* statt. Dieser wurde hier mit *Redis*[5] realisiert. *Redis* ist eine nicht relationale Datenbank und speichert zu gegebenen Schlüsseln Werte. Diese Werte können jedoch Strings, Listen, aber auch Hashtabellen sein. Dafür schreibt die Webanwendung den Job und seine Parameter auf die *Job Queue* im *Message Broker*. In sehr kurzen Abständen liest der Worker ob es neue Jobs gibt, die er bearbeiten kann. Der Rückgabewert wird dann für eine voreingestellte Zeit auf dem Message Broker gespeichert.

[5] https://redis.io

Außerdem hat der Worker die Möglichkeit in die Datenbanken zu schreiben. Der gesamte Prozess kann von der Webanwendung über den Message Broker überwacht werden. Ein Vorteil der Containervisualisierung ist, dass die Anzahl der Worker erhöht werden kann. Somit können mehrere Hintergrundprozesse gleichzeitig ausgeführt werden.
Zusätzlich dazu steht ein Container bereit, in dem ein Scheduler läuft, mit dem ebenfalls über den *Message Broker* kommuniziert werden kann. Dieser Scheduler schreibt Jobs zu einem definierten Zeitpunkt oder periodisch auf die Job Queue. Damit können wiederkehrend Anfragen an den Crawler gestellt und die erlangten Rohdaten verarbeitet werden, um Momentaufnahmen zu erhalten.

6.2.4 Suchindex

Für die Beschleunigung der Suche wurde ein weiterer *Redis* Container verwendet, um dort eine invertierte Liste mit Hashtabellen zu implementieren. Auch werden dort Informationen etwa über die Dokumentenlänge gespeichert. Da *Redis* eine Schlüssel-Werte-Datenbank ist, bietet es sich gut dafür an.

6.2.5 Webanwendung

Die Webanwendung läuft in einem Container und kann mit jedem der anderen Container kommunizieren. Dieser beantwortet Anfragen, lädt Content sowie verarbeitete Daten und initiiert Hintergrundjobs. Sein Port ist als einziger von Außen erreichbar.
Anzumerken ist, dass ein reverse Proxy z.B. *NGINX*[6] vor der Anwendung eingesetzt werden sollte, falls diese aus dem Internet erreichbar ist und nicht von einer anderen Webanwendung eingebunden wird.

6.2.6 Verbesserungspotenzial

Die Architektur könnte in dem Sinne verbessert werden, dass die Anwendung nach den Daten die sie verarbeiten, in *Microservices* (Newman, 2015) gesplittet werden. Programmteile können so leichter geändert werden und andere Anwendungen müssen nur die Details der Schnittstelle kennen. Microservices können also leicht ausgetauscht werden, wenn die Schnittstelle nicht verändert wird. So können schlanke Container gebaut werden, die nur ihre benötigten Abhängigkeiten enthalten und nicht an Bedingungen wie der Programmiersprache anderer Container gebunden sind. Ein weiterer Vorteil von Microservices ist die Skalierbarkeit ihrer Container.
Hier würde beispielsweise eine eigene kleine Anwendung die Suche realisieren und in einem eigenen Container laufen, mit dem kommuniziert werden kann. Newman (2015) empfiehlt erst eine monolithische Anwendung zu entwickeln und diese dann in Microservices aufzuteilen. Hier wurde dieser Ansatz nur in soweit verfolgt, dass der Crawler von der Webanwendung getrennt ist.

[6] http://NGINX.org

6.3 Themenextraktion

Die Themenextraktion ist ein wesentlicher Bestandteil der Analyse von Online-Partizipationsverfahren. In diesem Kapitel werden die Umsetzung der Verfahren aus Kapitel 4 sowie einige Hintergründe zur Implementierung erläutert. Zur Implementierung der Themenextraktion wurde auf das *Python*-Paket *scikit-learn* [7] zurückgegriffen, welches alle vorgestellten Verfahren beinhaltet. Somit stehen NMF, LDA, pLSI und LSI zur Verfügung.
Ausgewählt werden die zu speichernden Modelle von einem Benutzer mit Zugang, etwa einem Mitglied der Verwaltung. Dieser soll eine Themenanzahl, ein Verfahren und kann auch eine abweichende Filterliste angeben. Die Webanwendung erstellt daraufhin einen Hintergrundprozess, der von einem Worker abgearbeitet wird. Sollte sich der Benutzer entscheiden das Modell zu speichern, braucht das Modell nicht komplett neu berechnet zu werden, da die Job Queue das Ergebnis für einige Minuten gespeichert lässt. Die Speicherung beinhaltet sowohl das Modell, mit dem Suchanfragen in die Themendarstellung überführt werden können als auch die Dokumentenvektoren. In der Datenbank werden dann die Dokumentenvektoren mit den Zugehörigkeitswerten zu den Themen gespeichert. Dies kann zur Suche oder Visualisierung verwendet werden. Hierbei ist anzumerken, dass eine Speicherung der Vektoren in einer Datenbank nicht die optimale Lösung ist.
Damit nicht jedes mal das Modell geladen werden muss, sind auch 25 Top-Wörter gespeichert. Diese Anzahl sollte üblicherweise ausreichen um sie Benutzern anzuzeigen. Die Modelle werden immer auf Momentaufnahmen der Daten und Versionen einer Filterliste berechnet, weshalb Informationen darüber ebenfalls Teil des Schemas sind.
Es gibt auch Probleme die es bei einer Themenextraktion in diesem Zusammenhang zu beachten gibt. Die Momentaufnahmen, und damit die Modelle, sollen möglichst aktuell sein. Da die Berechnung von Natur aus eine nicht zu geringe Laufzeit hat, kommt hinzu dass bei einer neuen Momentaufnahme oder Änderung der Filterliste alle Modelle neu berechnet werden. Natürlich ist dies von der Anzahl der Daten und Modelle abhängig. Es böte sich zwar auch eine nachträgliche Filterung der Begriffe an, jedoch können so wenig sinnvolle Themen entstehen.
Im wesentlichen wurden die Überlegungen aus Kapitel 4 umgesetzt. Auf das automatische Finden der Themenanzahl und Interaktive Topic Modeling wurde verzichtet Die hohe Laufzeit des Verfahrens, um eine Themenanzahl zu empfehlen, rechtfertigt den geringen Nutzen nicht. Dafür benötigt diese Anwendung Benutzer, die Modelle speichern. Beide Verfahren zum Interactive Topic Modeling hätten von Grund auf implementiert werden müssen, was der Zeitrahmen dieser Arbeit nicht erlaubt.
Die Visualisierung und Interaktion der extrahierten Themen wird in den entsprechenden Kapiteln 6.4 und 6.6 erläutert.

6.4 Visualisierung

In diesem Kapitel wird erklärt welche Visualisierungsformen aus Kapitel 5 gewählt wurden. Diese werden mit Screenshots abgebildet. Gewählt wurden mehrheitlich einfache Visualisierungsformen, da diese weniger die Gefahr bieten, den Benutzer zu überfordern.

[7] http://scikit-learn.org/stable/

Die Visualisierungen werden mehrheitlich im Browser gezeichnet, sodass keine Bilder geladen werden, die zuvor vom System erstellt wurden, wodurch sie leichter durch Interaktionen erweitert werden können.
Verständlichkeit ist neben Relevanz das wichtigste Kriterium, das eine Visualisierung erfüllen muss. Beides ist im Vorfeld schwierig zu beantworten und wird stark vom Verfahren und dem Benutzerkreis beeinflusst. Nicht für jede Gruppe von Benutzern sind die gleichen Aspekte eines Online-Partizipationsverfahrens interessant. Je nach Verfahren, können die in der Anwendung verwendeten Tools auch benutzt werden, um diese zu erweitern. Alle Visualisierungen beruhen auf AJAX-Anfragen, sodass sie flexibel in anderen Anwendungen und Websiten verwendet werden können.
Strukturiert wird dieses Kapitel nach Teilen der Website mit denen ein Benutzer in Berührung kommt, wenn man sie als unabhängige Website betrachtet.
Die Interaktion mit Visualisierungen erfolgt dann im folgenden Kapitel 6.6.

6.4.1 Startseite

Meist ist die Startseite das erste, was eine Benutzer von einer Website aufruft. Diese soll einen gewissen Überblick liefern, jedoch sind in dieser Anwendung dazu keine Texte notwendig die das Online-Partizipationsverfahren beschreiben, welches die Anwendung ergänzt.
Die Startseite zeigt Statistiken über das Verfahren, also wie viele Beiträge und Kommentare verfasst, außerdem wie viele Stimmen insgesamt abgegeben wurden. Dies soll dem Benutzer eine Übersicht über die Größe des Verfahrens verschaffen. Darunter befindet sich eine Word Cloud mit den häufigsten Begriffen aller Dokumente. Die 200 häufigsten Begriffe werden mit ihren Häufigkeiten einmal für jede Momentaufnahme gespeichert, was die Performance verbessert. Mit einer AJAX-Anfrage bekommt man die 200 häufigsten Wörter mit ihrer relativen Häufigkeit. Die Skalierung und andere Parameter, wie die Word Cloud ausgestaltet werden soll, wird im Frontend berechnet. Ein Screenshot der Startseite ist in Abbildung 15 zu sehen.

6.4.2 Statistiken

Die Seite über Statistiken zeigt im wesentlichen Graphen der Statistiken. Dabei sind drei Arten von Visualisierungen zu sehen: Säulendiagramme, Histogramme und Streudiagramme. Die generelle Idee hierfür ist zwar, dass schon interessante Statistiken ausgewählt wurden, allerdings hauptsächlich die Infrastruktur geschaffen wurde, um den Code leicht zu erweitern und anderen Statistiken zu visualisieren.. Welche Dinge interessant sind, liegt hierbei im Auge des Betrachters und ist abhängig von der Datenlage.
Das oberste Diagramm zeigt die Verteilung der Vorschläge und Kommentare nach Wochentagen. Dabei werden jeweils die relativen Anteile der Vorschläge und Kommentare dargestellt, die am jeweiligen Wochentag erstellt wurden, wobei nur die Verwendung von relativen Häufigkeiten zu einer Vergleichbarkeit führt.
Interessant ist diese Statistik, weil sie mehrere Dinge verdeutlicht. So können die stärksten Wochentage identifiziert werden, was für die Verwaltung von Interesse sein könnte.

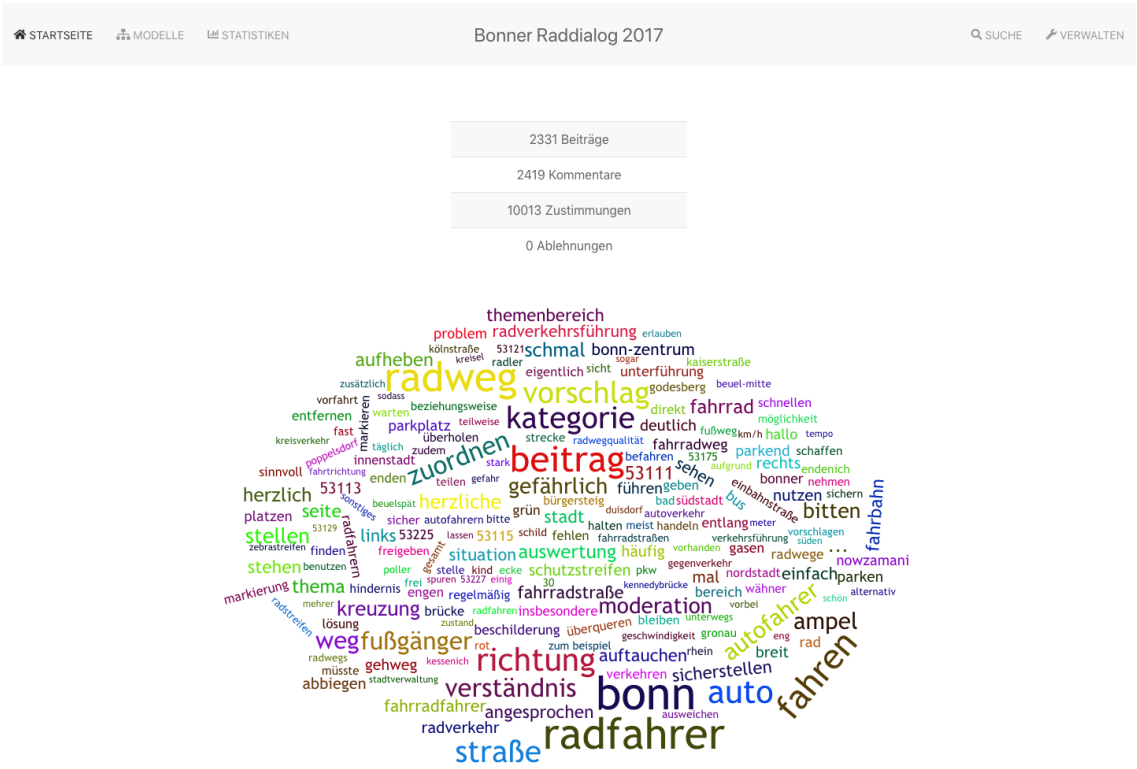


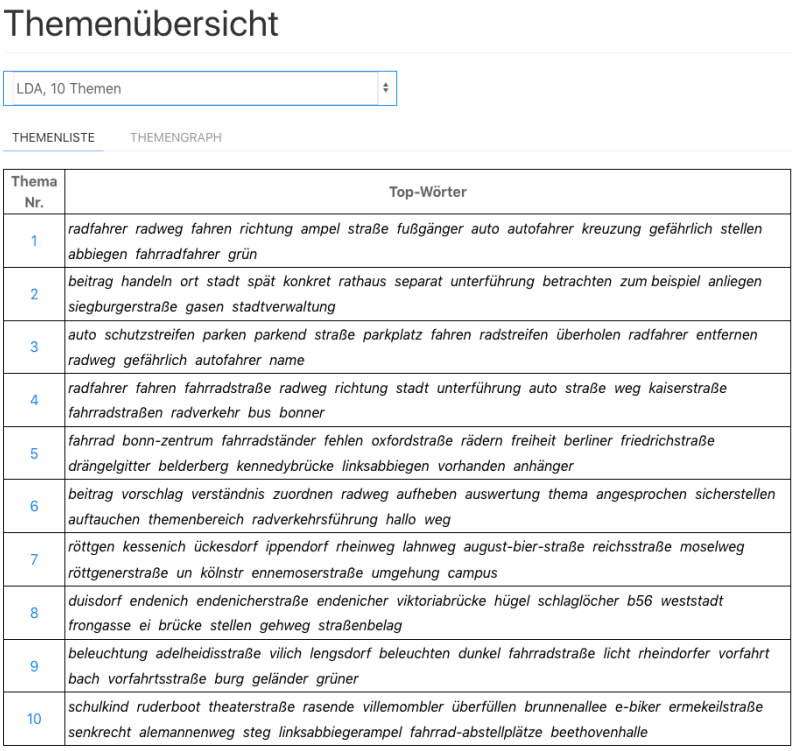
Abbildung 15: Screenshot der Startseite für den Raddialog Bonn, mit Statistiken und einer Word Cloud.

Auch bekommt man so ein Gefühl, wann vor allem Vorschläge und wann Kommentare gemacht werden und ob beide den gleichen Trends unterliegen.
Die beiden Histogramme zeigen die Verteilung von Anzahl der Kommentare bzw. Anzahl der Zustimmungen. Daraus kann hervorgehen wie die gemachten Vorschlägen beim Rest der Benutzer ankommen.
Um die Beziehung zwischen Anzahl der Kommentare bzw. Anzahl der Wörter eines Dokumentes zu verdeutlichen, können die beiden Streudiagramme verwendet werden. Dies kann Fragen beantworten, ob zum Beispiel besonders viel diskutierte Vorschläge besonders viel Zustimmung bekommen haben. Viel diskutiert heißt hier einmal nach Anzahl der Kommentare und einmal nach Gesamtanzahl der Wörter in einem Vorschlag sowie dessen Kommentaren. Eingezeichnet sind auch die jeweiligen Durchschnittswerte.
Der zeitliche Verlauf, wann Vorschläge gemacht und Kommentare erstellt worden sind, ist ebenfalls dargestellt. Ab diesem kann man ablesen, wann Verfahren besonders viel Aktivität stattfand. Die Darstellung ist interaktiv, der Benutzer kann den Bereich als vergrößern oder verschieben.
Bei Histogrammen und Säulendiagrammen kann der Nutzer mit der Maus die genauen Werte erfahren. Das Streudiagramm kann von Benutzer so vergrößert werden, dass ein beliebiger Bereich angezeigt wird. Ein Problem was bei Erstellungen jeder Statistik im Zusammenhang mit Online-Partizipationsverfahren entsteht, ist dass nicht alle Verfah-

ren die gleichen Informationen teilen. Dieses Problem liegt auch darin begründet, dass nur öffentlich verfügbare Daten vom Crawler erfasst werden können. Nicht für jedes Verfahren können also die gleichen Statistiken gemacht werden, da zum Beispiel nicht jedes Verfahren eine Ablehnung von Vorschlägen zulässt. Auch ist es bei der Verwendung eines Crawler nicht möglich Daten zu erheben, die mehr Einblick über Nutzerverhalten geben. Etwa fehlen Daten wann Zustimmungen zu einem Vorschlag gegeben wurden.

6.4.3 Themen

Themen darzustellen ist eine schwierige Aufgabe, da die Visualisierungen entweder wenig neues über die Themen hervorbringen oder zu komplex und umfangreich sind. Bei dieser Anwendung wurde sich für zwei Visualisierungen entschieden. Einmal werden Modelle als Wortlisten der Themen und einmal analog zu *LDAbis* (vgl. Kapitel 5.6) dargestellt.
Die Themenliste beinhaltet 15 Begriffe, welche absteigend nach deren Einfluss auf das Thema sortiert sind. Ein Screenshot davon ist in Abbildung 16 zu sehen.
In dem anderen Tab wird für dieses Modell eine Darstellung geladen, die Themen in



Thema Nr.	Top-Wörter
1	radfahrer radweg fahren richtung ampel straße fußgänger auto autofahrer kreuzung gefährlich stellen abbiegen fahrradfahrer grün
2	beitrag handeln ort stadt spät konkret rathaus separat unterführung betrachten zum beispiel anliegen siegburgerstraße gasen stadtverwaltung
3	auto schutzstreifen parken parkend straße parkplatz fahren radstreifen überholen radfahrer entfernen radweg gefährlich autofahrer name
4	radfahrer fahren fahrradstraße radweg richtung stadt unterführung auto straße weg kaiserstraße fahrradstraßen radverkehr bus bonner
5	fahrrad bonn-zentrum fahrradständer fehlen oxfordstraße rädern freiheit berliner friedrichstraße drängelgitter belderberg kennedybrücke linksabbiegen vorhanden anhänger
6	beitrag vorschlag verständnis zuordnen radweg aufheben auswertung thema angesprochen sicherstellen auftauchen themenbereich radverkehrsführung hallo weg
7	röttgen kessenich ückesdorf ippendorf rheinweg lahnweg august-bier-straße reichsstraße moselweg röttgenerstraße un kölnstr ennemoserstraße umgehung campus
8	duisdorf endenich endenicherstraße endenicher viktoriabrücke hügel schlaglöcher b56 weststadt frongasse ei brücke stellen gehweg straßenbelag
9	beleuchtung adelheidisstraße vilich lengsdorf beleuchten dunkel fahrradstraße licht rheindorfer vorfahrt bach vorfahrtsstraße burg geländer grüner
10	schulkind ruderboot theaterstraße rasende villemombler überfüllen brunnenallee e-biker ermekeilstraße senkrecht alemannenweg steg linksabbiegerampel fahrrad-abstellplätze beethovenhalle

Abbildung 16: Screenshot einer Liste der Top-Wörter von extrahierten Themen.

zwei Dimensionen darstellt. Diese Visualisierung beruht auf der Arbeit von Sievert und Shirley (2014). Jedoch haben die Themen alle den gleichen Radius, da nicht nachvollziehbar ist, warum ähnliche Themen in anderen enthalten sind, nur weil ein häufig auftretendes Thema einen großen Radius hat. Dies suggeriert Hierarchie, die sich so aus dem

Modell nicht ergibt. Fährt der Benutzer mit der Maus über ein Thema, werden die Nummer des Themas und fünf Top-Wörter angezeigt. Nach subjektiver Wahrnehmung funktioniert diese Darstellung am besten mit LDA Modellen. Zu sehen ist dies in Abbildung 17.

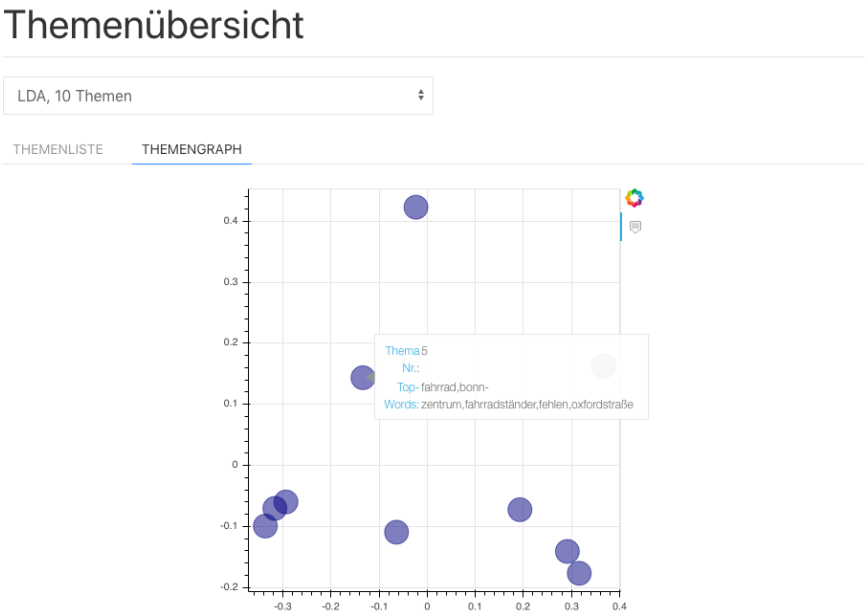


Abbildung 17: Screenshot einer 2D-Darstellung des Modells aus Abbildung 16.

6.5 Themenverteilung der Beiträge

Bei der Betrachtung eines Beitrags, kann sich der Benutzer für jede vorhandene Modell die Themenverteilung als horizontales Balkendiagramm darstellen lassen, wie in Abbildung 18 zu sehen. Jedes Thema ist mit einer anderen Farbe dargestellt und entsprechend seines Einflusses skaliert. Bewegt der Benutzer die Maus über einen farbigen Block werden die Nummer des Themas, dessen Zugehörigkeit sowie fünf Top-Wörter dargestellt.

6.6 Interaktion

Bei einer Webanwendung bietet es sich gut an, den Benutzer Analysedaten interaktiv entdecken zu lassen. Interaktion ergänzt Visualisierungen um eine weitere Ebene und lässt den Benutzer die Daten individuell erleben. Dadurch erhält man den Vorteil, nicht jede Darstellung bis ins kleinste Detail zu planen. Menschliche Interaktion kann an vielen Stellen besseres leisten, als maschinell Ausgewähltes.
Die Interaktion in dieser Anwendung basiert auf zwei Überlegungen. Ein normaler Benutzer möchte mit dem dargestellten interagieren und mehr darüber erfahren. Es besteht also ein *Information Need* (Taylor, 1962), weshalb dies zum Bereich des Information Retrieval zählt. Die andere Interaktion besteht zwischen einem privilegierten Benutzer und dem System, um Modelle zu verwalten.

Abbildung 18: Screenshot von einer Themenverteilung eines Beitrags.

6.6.1 Interaktion mittels Information Retrieval

Betrachtet ein Benutzer die Word Cloud oder die Themenliste eines Modells, kann er mehr über einen Begriff erfahren, indem er auf ihn klickt. Daraufhin wird eine Anfrage mit dem Begriff gestartet und das Ergebnis unter der Cloud bzw. Tabelle angezeigt. Der Benutzer bleibt so auf der gleichen Seite und wird nicht im Gedanken unterbrochen. Um dies zu unterstützen, wird der Scrollen zu den Ergebnissen animiert. Alternativ lassen sich die Links in neuen Tabs öffnen. Sowohl die Begriffe in der Word Cloud, als auch in den Wort Listen werden optisch hervorgehen, wenn der Benutzer mit der Maus über sie fährt. Außerdem steht eine extra Seite zum Suchen von beliebigen Suchbegriffen zur Verfügung.
Möchte ein Benutzer mehr über ein Thema des Modells erfahren, reicht ein Klick auf die Nummer des Themas, um die Beiträge angezeigt zu bekommen, die am stärksten mit diesem Thema assoziiert sind. So werden die Themen, die als Wortlisten dargestellt wurden, konkret veranschaulicht. Abbildung 19 zeigt Beispiele für solche Ergebnisse.
Damit dies möglich gemacht werden kann, musste eine Suchfunktion implementiert werden. Da schon durch NLP verarbeitete Daten vorliegen, wobei auch Straßennamen anders behandelt wurden, wurde das meiste dieser Suchfunktion selbst implementiert. Sollten sehr große Datenmengen verarbeitet werden, wäre der Einsatz einer hochperformanten Software mit z.B. *Lucene*[8] empfehlenswert.
Die Suchfunktion soll nicht die Suchfunktion des Online-Partizipationsverfahrens ergänzen, sondern die Anwendung von dieser unabhängig machen. Deshalb sind die hier implementierten Suchfunktionen textbasiert. Dabei werden drei verschiedene Ansätze zur Suche verwendet. Die Standartsuche, die auch zum Einsatz kommt wenn auf einen Beitrag geklickt wird, verwendet Okapi BM25. Um diese Suche effizienter zu gestalten, wurde eine invertierte Liste in *Redis* angelegt. Dort wird für jeden Term eine Liste von Dokumenten mit der Termhäufigkeit in diesem Dokument gespeichert. Außerdem werden in der Datenbank die Dokumentenlängen, durchschnittliche Dokumentenlänge und Anzahl an Dokumenten gespeichert, da diese für Okapi 25 benötigt werden. Die Laufzeit verringert sich so, da nur noch die Dokumente sortiert werden müssen, die

[8] https://lucene.apache.org

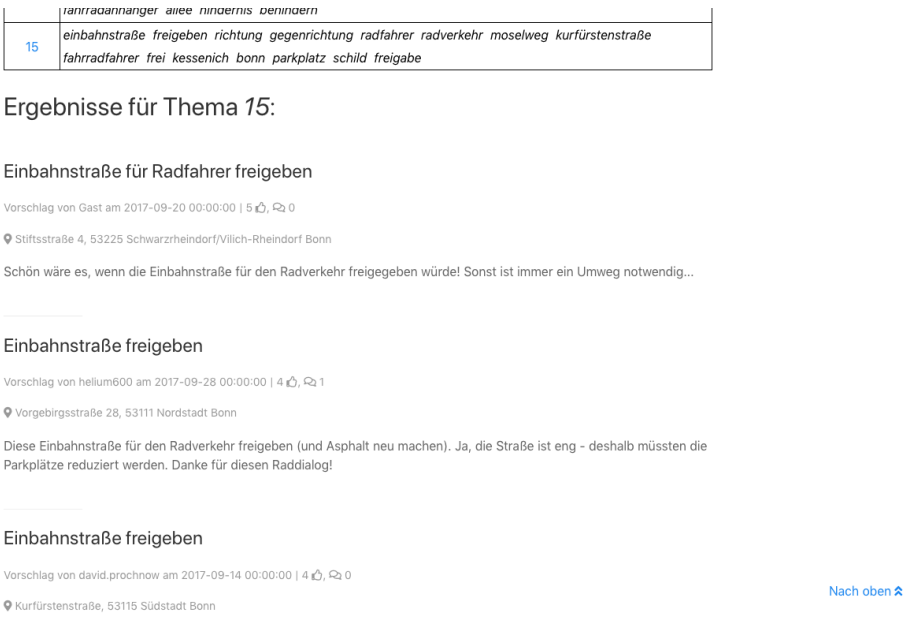


Abbildung 19: Screenshot von Beiträgen, die einem Thema besonders zugeordnet werden.

einen der Suchbegriffe enthalten. (vgl. 3.2)
Eine weitere Möglichkeit zu suchen ist, eines der berechneten Modelle zu verwenden. Dies hat den Vorteil, dass nicht nur mit den Wörtern gesucht wird, die in der Anfrage enthalten sind, sondern mittels semantischen Features. Dafür wird für jedes Dokument der Featurevektor gespeichert, um diesen nicht für jede Suche berechnen zu müssen, und dann die Ähnlichkeit mit dem Abfragevektor berechnet. Das Ähnlichkeitsmaß hängt dabei von der Art des Modells ab. Ähnliche Beiträge können ebenfalls auf Grundlage eines Modells gesucht werden, wozu bei jedem Beitrag ein Link hinterlegt ist (siehe Abbildung 18). Diese Ergebnisse öffnen sich dann in einem neuen Tab.
Eine sehr Interaktive Möglichkeit der Suche, kann mit Relevance Feedback geschaffen werden. Implementiert wurde dies mit klassischen $tf\text{-}idf$ Features. Der Benutzer kann dann für die Anfrage relevante Dokumente auswählen und die Anfrage neu berechnen lassen. Zur Berechnung wird der alte Anfragevektor benötigt, der einige Minuten in der Datenbank verbleibt.
Angezeigt werden die zehn relevantesten Beiträge zu der Anfrage, jwobei sich 100 Beiträge nachladen lassen, ohne dass eine Neuberechnung der Suche stattfinden muss. Klickt ein Benutzer auf ein Suchergebnis, kann er den ganzen Vorschlag inklusive Kommentare ansehen.

6.6.2 Interaktion mit einem privilegierten Benutzer

Die Interaktion mit Benutzern, die Online-Partizipationsverfahren betreuen, stellt abermals eine andere Herausforderung da. So haben sie neben der Interaktion, die jeder Benutzer verwenden kann, ebenfalls die Möglichkeit Modelle zu berechnen, zu speichern

und diese durch Veränderung der Blacklist zu beeinflussen. Zur Berechnung können das Topic Model und die Anzahl an Themen ausgewählt werden.
Dies wurde mit Hintergrundprozessen gelöst, was mehrere Vorteile bringt. Sollten mehrere Benutzer das System gleichzeitig verwenden, läuft dennoch alles nur über eine Job Queue, die deshalb besser überwacht werden kann.
Wenn ein Modell berechnet wird, unterscheidet sich die Sichtweise nicht sehr von der eines normalen Benutzers. Hingegen können Begriffe direkt aus den Wortlisten zu einer Filterliste hinzugefügt werden. Modelle können mit einer Filterliste berechnet werden, die sich von der auf dem Server unterscheidet, jedoch nicht gespeichert werden ohne dass auch die Filterliste gespeichert wird. Viele verschiedene Filterlisten würden schnell unübersichtlich werden, sind aber generell denkbar.
Abbildung 20 zeigt die Sicht auf die Modellverwaltung.

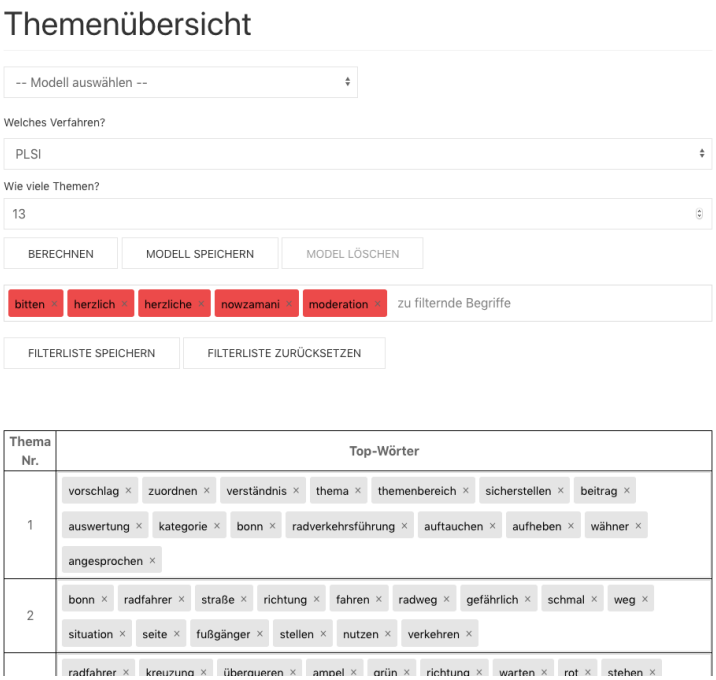


Abbildung 20: Screenshot eines berechneten Modells, sowie der Filterliste.

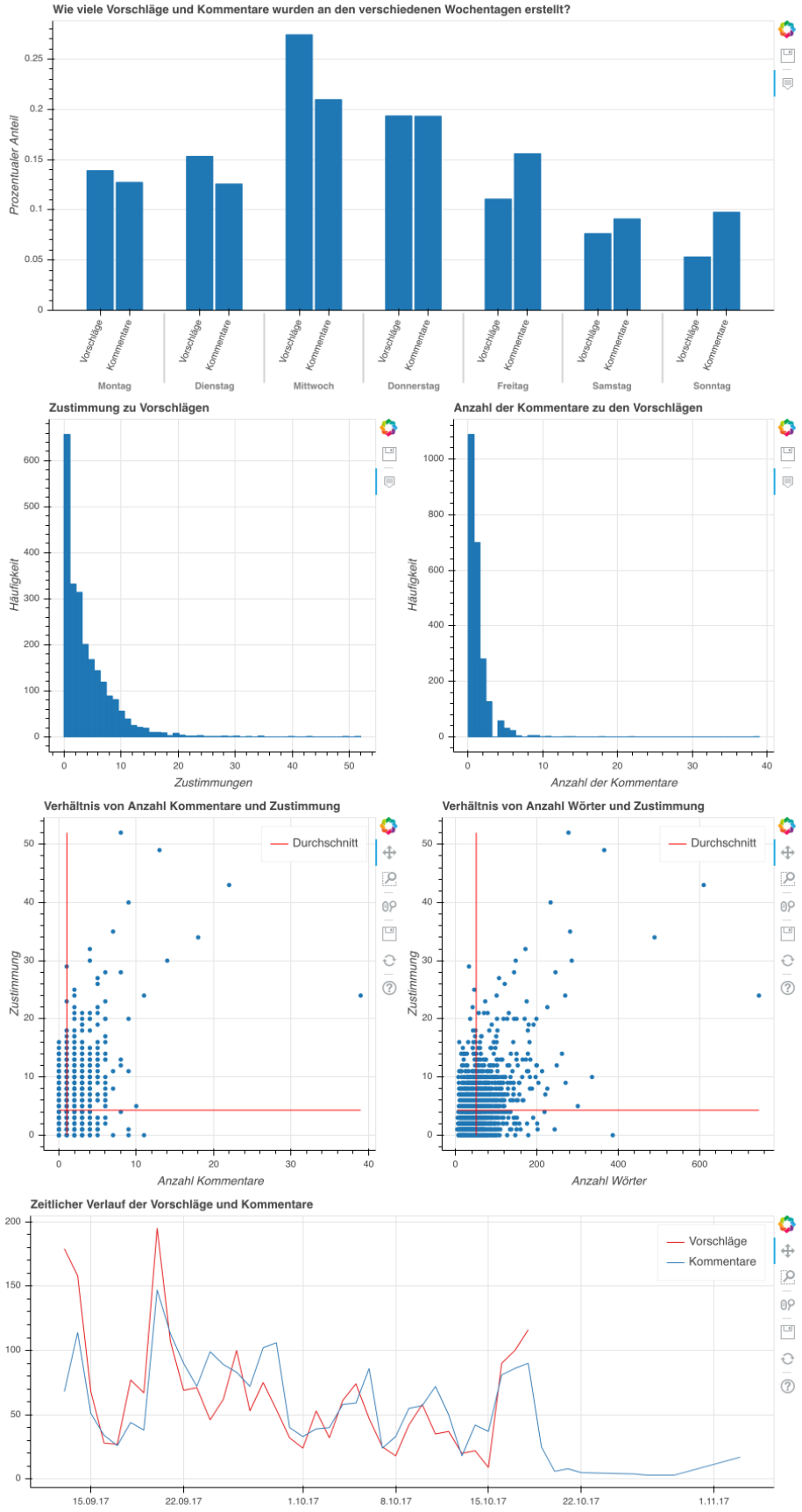


Abbildung 21: Screenshot der von Statistiken über den Raddialog Bonn.

7 Abschließendes

In den beiden folgenden Kapiteln wird ein Fazit gezogen und ein Ausblick für die Thematik gegeben.

7.1 Fazit

Online-Partizipationsverfahren maschinell zu analysieren, kann Teilnehmenden und Betreuenden einen Einblick in und eine Übersicht über das Verfahren geben. Beides kann helfen Verfahren so zu gestalten, dass sich mehr Menschen beteiligen und diese Beitilgung besser ausgewertet wird. Eine zentraler Wunsch bei Online-Partizipation ist, die Akzeptanz von Entscheidungen bei den Betroffenen zu steigern.
In dieser Arbeit wurden verschiedene Methoden zur Themenextraktion, Visualisierung und Interaktion vorgestellt. Unter Verwendung dieser wurde daraufhin eine Webanwendung entwickelt, die so gestaltet wurde, dass sie sowohl mit laufenden als auch abgeschlossenen Verfahren verwendet werden kann.
Als Grundlage aller weiteren Methoden wurde Natural Language Processing eingesetzt und die Besonderheiten bei Online-Partizipation besprochen.
Zur Themenextraktion bieten sich neben der state-of-the-art Methode LDA ebenfalls NMF und LSI an. Diese extrahieren Themen werden mit einer sortierten Menge von Wörtern assoziiert. Damit kann eine Übersicht über die Beiträge gegeben werden. Die Ergebnisse, also gefundenen Themen sind jedoch stark subjektiv (Chang et al., 2009). Darüber hinaus wurde eine Methode zur Berechnung einer Themenanzahl vorgestellt, die jedoch nicht in der Anwendung verwendet wurde. Auch wurden Methoden diskutiert, die einen interaktiven Ansatz verfolgen. So werden etwa von einem Benutzer stammende Informationen über Begriffe oder Themen in die Berechnung mit einbezogen oder Modelle daraufhin umgerechnet.
Da Wortlisten extrahierter Themen nicht alle Aspekte eines Online-Partizipationsverfahrens abdecken können, wurden Visualisierungen vorgestellt. Neben Darstellungen für Metadaten, gibt es auch Visualisierungen für extrahierte Themen oder Modelle. Diese sind leider oft kompliziert oder zeigen selten neue Aspekte auf. Weiterhin ist die Beurteilung einer Visualisierung stark subjektiv. Visualisierungen sollten intuitiv, informativ und leicht zugänglich sein.
Neben der Interaktion mit der Webanwendung, wird hauptsächlich Information Retrieval zur Interaktion verwendet. Einige der hier vorgestellten Methoden verwenden dabei Eigenschaften extrahierter Themen oder bieten Interaktion über die extrahierten Themen hinaus.

7.2 Ausblick

7.2.1 Evaluation

Eine Anwendung wie die hier vorgestellte, sollte mit Anwendern evaluiert und daraufhin angepasst werden. Erfolgen sollte eine solche Evaluation mit Blick auf die unterschiedlichen Benutzer und ihre verschiedenen Interessen. Gestaltet werden könnte diese

Evaluation etwa nach Borlund und Ingwersen (1997).

7.2.2 Erweiterung des Funktionsumfangs

Eine Erweiterung des Funktionsumfangs ist in mehrere Richtungen denkbar. So kann die Suchfunktion ausgebaut werden, sodass etwa eine Facettensuche möglich ist (Tunkelang, 2009). So kann der Benutzer noch mehr Informationen in die Anfrage eingeben, als nur Text.
Vor allem für die Verwaltung, könnte eine Funktion interessiert sein, die *Collaborative Information Seeking* (Hansen et al., 2015) unterstützt. Damit wäre es verschiedenen Benutzer möglich zusammen zu Suchen und etwa Suchprofile oder Ergebnisse zu speichern.
Vielversprechend könnte auch die Verknüpfung mit klassifizierten Dokumenten oder Informationen, z.B. aus der Verwaltung oder von Wikipedia sein. Dies würde die Vorschläge in den richtigen Kontext bringen und könnte Nutzer schon beim Erstellen ihrer Vorschläge unterstützen. Der Vorschlagende könnte so auf bereits beschlossene Maßnahmen hinweisen, die noch nicht umgesetzt sind. Auch könnten so Missverständnisse ausgeräumt werden, die aufgrund fehlender Kompetenz von zum Beispiel Stadträten in Landessachen entstehen.

References

Vimala Balakrishnan und Ethel Lloyd-Yemoh (2014). "Stemming and Lemmatization: A Comparison of Retrieval Performances". In: *Lecture Notes on Software Engineering* 2.3, S. 262–267.

Ian Bennett (Jan. 2011). *Natural language speech lattice containing semantic variants*. US Patent 7,873,519.

David M Blei, Andrew Y Ng und Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, S. 993–1022.

Pia Borlund und Peter Ingwersen (1997). "The development of a method for the evaluation of interactive information retrieval systems". In: *Journal of documentation* 53.3, S. 225–250.

Eric Brill (1992). "A simple rule-based part of speech tagger". In: *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, S. 112–116.

Nan Cao und Weiwei Cui (2016). "Introduction to Text Visualization". In: *Atlantis Briefs in Artificial Intelligence*. Kap. 10.

Sung-Hyuk Cha und Sargur N Srihari (2002). "On measuring the distance between histograms". In: *Pattern Recognition* 35.6, S. 1355–1370.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang und David M. Blei (2009). "Reading Tea Leaves: How Humans Interpret Topic Models". In: *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*. S. 288–296.

Edwin Chen (2011). *Introduction to Latent Dirichlet Allocation*. URL: http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/.

Jason Chuang, Daniel Ramage, Christopher D. Manning und Jeffrey Heer (2012). "Interpretation and trust: designing model-driven visualizations for text analysis". In: *CHI*.

Stefan Conrad (2017). *Vorlesung: Natural Language Processing und Information Retrival, WS 2017/2018*. Lehrstuhl für Datenbanken und Informationssysteme an der Heinrich-Heine-Universität Düsseldorf.

Stefan Debortoli, Oliver Müller, Iris A Junglas und Jan vom Brocke (2016). "Text mining for information systems researchers: an annotated topic modeling tutorial." In: *CAIS* 39, S. 7.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer und Richard Harshman (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, S. 391–407.

David L Donoho et al. (2000). "High-dimensional data analysis: The curses and blessings of dimensionality". In: *AMS math challenges lecture* 1.2000, S. 32.

Jonathan Feinberg (2010). "Wordle". In: *Beautiful visualization: looking at data through the eyes of experts*. Hrsg. von Julie Steele und Noah Iliinsky. " O'Reilly Media, Inc.". Kap. 3.

Thomas MJ Fruchterman und Edward M Reingold (1991). "Graph drawing by force-directed placement". In: *Software: Practice and experience* 21.11, S. 1129–1164.

Norbert Fuhr (1992). "Probabilistic models in information retrieval". In: *The computer journal* 35.3, S. 243–255.

Eric Gaussier und Cyril Goutte (2005). "Relation between PLSA and NMF and implications". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, S. 601–602.

Stuart Geman und Donald Geman (1984). "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6, S. 721–741.

Derek Greene, Derek O'Callaghan und Pádraig Cunningham (2014). "How many topics? stability analysis for topic models". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, S. 498–513.

Gregory Grefenstette und Pasi Tapanainen (1994). "What is a word, what is a sentence?: problems of Tokenisation". In:

Preben Hansen, Chirag Shah und Claus-Peter Klas (2015). *Collaborative information seeking: Best practices, new domains and new thoughts*. Springer.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff und Alison Smith (2014). "Interactive topic modeling". In: *Machine learning* 95.3, S. 423–469.

K Sparck Jones, Steve Walker und Stephen E. Robertson (2000). "A probabilistic model of information retrieval: development and comparative experiments: Part 2". In: *Information processing & management* 36.6, S. 809–840.

Jason S. Kessler (2017). "Scattertext: a Browser-Based Tool for Visualizing how Corpora Differ". In: *CoRR* abs/1703.00565.

George Klir und Bo Yuan (1995). *Fuzzy sets and fuzzy logic*. Bd. 4. Prentice hall New Jersey.

Da Kuang, Jaegul Choo und Haesun Park (2015). "Nonnegative matrix factorization for interactive topic modeling and document clustering". In: *Partitional Clustering Algorithms*. Springer, S. 215–243.

K. Kucher und A. Kerren (2015). "Text visualization techniques: Taxonomy, visual survey, and community insights". In: *2015 IEEE Pacific Visualization Symposium (PacificVis)*, S. 117–121.

Julian Kupiec (1992). "Robust part-of-speech tagging using a hidden Markov model". In: *Computer Speech & Language* 6.3, S. 225–242.

Cornelius Lanczos (1950). *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA.

Thomas K Landauer, Darrell Laham und Peter W Foltz (1998). "Learning human-like knowledge by singular value decomposition: A progress report". In: *Advances in neural information processing systems*, S. 45–51.

Daniel D Lee und H Sebastian Seung (2001). "Algorithms for non-negative matrix factorization". In: *Advances in neural information processing systems*, S. 556–562.

Michael Levandowsky und David Winter (1971). "Distance between sets". In: *Nature* 234.5323, S. 34.

Wolfgang Lezius, Reinhard Rapp und Manfred Wettler (1998). "A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for German". In: *Proceedings of the 17th international conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, S. 743–748.

Matthias Liebeck (2018). "Automated Discussion Analysis in Online Participation Projects". Diss. Heinrich-Heine-Universität Düsseldorf.

Matthias Liebeck und Stefan Conrad (2015). "IWNLP: Inverse Wiktionary for Natural Language Processing". In: *Proceedings of the 53rd Annual Meeting of the Association for*

Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers, S. 414–418.

Matthias Liebeck, Katharina Esau und Stefan Conrad (2017). "Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung". In: *HMD Praxis der Wirtschaftsinformatik* 54.4, S. 544–562.

Jimmy Lin und Chris Dyer (2010). *Data-Intensive Text Processing with MapReduce*. Morgan und Claypool Publishers.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh und Ye-Yi Wang (2015). "Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval". In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, S. 912–921.

Steffen Lohmann, Jürgen Ziegler und Lena Tetzlaff (2009). "Comparison of tag cloud layouts: Task-related performance and visual exploration". In: *IFIP Conference on Human-Computer Interaction*. Springer, S. 392–404.

R Bruce Lydiard, Karl Rickels, Barry Herman und Douglas E Feltner (2010). "Comparative efficacy of pregabalin and benzodiazepines in treating the psychic and somatic symptoms of generalized anxiety disorder". In: *International Journal of Neuropsychopharmacology* 13.2, S. 229–241.

Laurens van der Maaten und Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov, S. 2579–2605.

Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Christopher D. Manning und Hinrich Schütze (1999). *Foundations of statistical natural language processing*. MIT press.

L. McInnes und J. Healy (Feb. 2018). "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *ArXiv e-prints*. arXiv: 1802.03426 [stat.ML].

Qiaozhu Mei, Xuehua Shen und ChengXiang Zhai (2007). "Automatic labeling of multinomial topic models". In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, S. 490–499.

Jaimie Murdock und Colin Allen (2015). "Visualization Techniques for Topic Model Checking." In: *AAAI*, S. 4284–4285.

Tetsuji Nakagawa, Taku Kudo und Yuji Matsumoto (2001). "Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines." In: *NLPRS*, S. 325–331.

Sam Newman (2015). *Building microservices: designing fine-grained systems*. " O'Reilly Media, Inc.".

Linda Dailey Paulson (2005). "Building Rich Web Applications with Ajax". In: *IEEE Computer* 38.10, S. 14–17.

Slav Petrov, Dipanjan Das und Ryan McDonald (2011). "A universal part-of-speech tagset". In: *arXiv preprint arXiv:1104.2086*.

Mannes Poel, Luite Stegeman und Rieks op Den Akker (2007). "A support vector machine approach to dutch part-of-speech tagging". In: *Lecture Notes in Computer Science* 4723, S. 274.

Praxissymposium „Online-Partizipation in Kommunen" (2018). Köln, Deutschland.

Dave Raggett, Arnaud Le Hors, Ian Jacobs et al. (1999). "HTML 4.01 Specification". In: *W3C recommendation* 24.

Stephen Robertson, Hugo Zaragoza et al. (2009). "The probabilistic relevance framework: BM25 and beyond". In: *Foundations and Trends® in Information Retrieval* 3.4, S. 333–389.

Joseph John Rocchio (1971). "Relevance feedback in information retrieval". In: *The SMART retrieval system: experiments in automatic document processing*, S. 313–323.

Sam T Roweis und Lawrence K Saul (2000). "Nonlinear dimensionality reduction by locally linear embedding". In: *science* 290.5500, S. 2323–2326.

Gerard Salton und Christopher Buckley (1988). "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5, S. 513–523.

John W. Sammon (1969). "A Nonlinear Mapping for Data Structure Analysis". In: *IEEE Transactions on Computers* C-18, S. 401–409.

Shoto Sasaki, Kazuyoshi Yoshii, Tomoyasu Nakano, Masataka Goto und Shigeo Morishima (2014). "LyricsRadar: A Lyrics Retrieval System Based on Latent Topics of Lyrics." In: *Ismir*, S. 585–590.

Anne Schiller, Simone Teufel und Christine Thielen (1995). "Guidelines f"ur das Tagging deutscher Textcorpora mit STTS". In: *Universit"aten Stuttgart und T"ubingen*.

Helmut Schmid (2013). "Probabilistic part-ofispeech tagging using decision trees". In: *New methods in language processing*, S. 154.

David w. Scott (1979). "On optimal and data-based histograms". In: *Biometrika* 66.3, S. 605–610.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng und Grégoire Mesnil (2014). "Learning semantic representations using convolutional neural networks for web search". In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM, S. 373–374.

Carson Sievert und Kenneth Shirley (2014). "LDAvis: A method for visualizing and interpreting topics". In: *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, S. 63–70.

Vin D Silva und Joshua B Tenenbaum (2003). "Global versus local methods in nonlinear dimensionality reduction". In: *Advances in neural information processing systems*, S. 721–728.

Alison Smith, Jason Chuang, Yuening Hu, Jordan L. Boyd-Graber und Leah Findlater (2014). "Concurrent Visualization of Relationships between Words and Topics in Topic Models". In:

Mark Steyvers und Tom Griffiths (2007). "Probabilistic topic models". In: *Handbook of latent semantic analysis* 427.7, S. 424–440.

Herbert A. Sturges (1926). "The Choice of a Class Interval". In: *Journal of the American Statistical Association* 21.153, S. 65–66.

Nancy R Tague et al. (2005). *The quality toolbox*. Bd. 600. ASQ Quality Press Milwaukee, WI.

Robert S Taylor (1962). "The process of asking questions". In: *American documentation* 13.4, S. 391–396.

Erik F Tjong Kim Sang und Fien De Meulder (2003). "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, S. 142–147.

Daniel Tunkelang (2009). "Faceted search". In: *Synthesis lectures on information concepts, retrieval, and services* 1.1, S. 1–80.

Paul F Velleman und David C Hoaglin (1981). *Applications, basics, and computing of exploratory data analysis*. Duxbury Press.

W John Wilbur und Karl Sirotkin (1992). "The automatic identification of stop words". In: *Journal of information science* 18.1, S. 45–55.

Hugh E. Williams (2003). "Genomic Information Retrieval". In: *Proceedings of the 14th Australasian Database Conference - Volume 17*. ADC '03. Australian Computer Society, Inc., S. 27–35.

S. K. M. Wong, Wojciech Ziarko und Patrick C. N. Wong (1985). "Generalized Vector Spaces Model in Information Retrieval". In: *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '85. ACM, S. 18–25.

Justin Zobel und Alistair Moffat (Apr. 1998). "Exploring the Similarity Space". In: *SIGIR Forum* 32.1, S. 18–34.

Abbildungsverzeichnis

Tabellenverzeichnis