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

Methoden der Netzwerkanalyse im Topic Modeling

eingereicht von Jana Thelen
geboren am 26.08.1991 in Adenau

Dezember 2019

Betreuung:
Dr. Martin Gronemann
Institut für Informatik
Universität zu Köln

Dr. Andreas Hamm
Think Tank
Deutsches Zentrum für Luft- und Raumfahrt

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während meiner Masterarbeit unterstützt und motiviert haben.
An allererster Stelle möchte ich mich bei Herrn Andreas Hamm für seine exzellente thematische sowie moralische Unterstützung bedanken. Vielen Dank für die gute Zusammenarbeit, für die guten Ideen und die Geduld.
Ein großer Dank gebührt auch dem Betreuer Herr Martin Gronemann. Vielen Dank für das Interesse an dem Thema und die tolle Zusammenarbeit.
Ich bedanke mich zudem bei Herrn Mark Azzam, der es erst möglich gemacht hat, im Deutschen Zentrum für Luft- und Raumfahrt in der Abteilung des Think Tanks meine Masterarbeit schreiben zu dürfen.
Auch möchte ich mich bei allen Team Mitliedern des Think Tanks bedanken, die mich in allen Phasen der Masterarbeit begleitet haben.
Ein großes Dank geht auch an alle fleißigen Korrekturleser.
Abschließend möchte ich mich bei meinen Eltern und meinen Brüdern bedanken, die mich sowohl finanziell unterstützt als auch jederzeit an mich geglaubt haben.

Inhaltsverzeichnis

1 Einleitung

Aufgrund der fortschreitenden Digitalisierung liegen immer mehr Sammlungen von Dokumenten in elektronischer Form vor. Es ist sehr aufwendig diese alle zu lesen, um die Hauptthemen und damit den Inhalt des Textes wiederzugeben. Topic Modeling, als eine wichtige Technik im Text Mining, kann die thematische Struktur einer großen Sammlung an Textdokumenten zusammenfassen.

1.1 Problematik und Methoden

Um Topics in einer Dokumentensammlung zu bestimmen, können unterschiedliche Topic-Modeling-Methoden angewendet werden: Es gibt probabilistisch begründete Methoden des Topic Modelings. Unter diesen ist LDA (Latent Dirichlet Allocation) eine sehr verbreitete und häufig verwendete. Die Idee hierbei ist folgende: Für jedes Topic gibt es eine charakteristische Verteilung von Wörtern und jedes Dokument enthält dementsprechend Wörter der in ihm behandelten Topics. Die Methode lernt unüberwacht aus den Wortzählungen in allen Dokumenten der Sammlung die Verteilung von Wörtern in Topics. Das Modell gibt letztlich nicht die Topic-Namen aus, sondern charakterisiert die Topics anhand ihrer typischen Wortverteilungen. Anschließend muss interpretiert werden, welche Themen mit diesen Wörtern verknüpft sind.
Alternativ dazu wird von unterschiedlichen Autoren in neuerer Zeit ein anderer Zugang zum Topic Modeling vorgeschlagen: Dieser fasst Sammlungen von Dokumenten als Netzwerke auf und versucht diese Themen mit Methoden der Community Detection zu identifizieren. Die prominenten Vertreter der Community Detection arbeiten mit einer Optimierung der Netzwerk-Modularität.
Das Hauptanliegen dieser Arbeit ist es, verschiedene Varianten dieses Ansatzes gegenüberzustellen und aussichtsreiche Modelle auf Probleme, die für den DLR Think Tank relevant sind, anzuwenden.

1.2 Relevanz und Anwendung im DLR

Die Abteilung Think Tank im Deutschen Zentrum für Luft- und Raumfahrt (DLR) arbeitet Handlungsempfehlungen für den Vorstand aus, indem die Abteilung kontinuierlich technische, gesellschaftliche, wirtschaftliche und politische Entwicklungen analysiert und zukünftige globale Herausforderungen in ihrer Bedeutung für das DLR bewertet. Ziel hierbei ist es, Verfahren zu entwickeln, welche Trends in den genannten Bereichen frühzeitig erkennen. Für eine solche Trenderkennung sind folgende Beispiele möglich:

- Themen in Patentbeschreibungen erkennen
- Themen in wissenschaftlichen Publikationen erkennen
- Themen in politischen Dokumenten erkennen

Zum letzteren läuft aktuell eine Promotionsarbeit in Zusammenarbeit mit dem Lehrstuhl für internationale Politik und Außenpolitik der Universität zu Köln.

2 Textaufbereitung

Das Ziel dieser Arbeit ist es, Topics in Textsammlungen (Korpora) zu erkennen. In diesem Kapitel werden Verfahren beschrieben, die sich zur Textaufbereitung auf die Korpora anwenden lassen. Es ist nämlich sehr wichtig, nur die Begriffe der jeweiligen Datenquelle zu selektieren, die für eine thematische Interpretation als sinnvoll und wichtig erscheinen. Zudem werden in diesem Kapitel zwei Sortierungsverfahren von Wörtern vorgestellt, um die Topic-Listen hinterher gut und einfach interpretieren zu können. Die beschriebenen unterschiedlichen Schritte zur Textaufbereitung werden dann hinterher im Kapitel 7 auf zwei Korpora, nämlich die Kleinen Anfragen an die Bundesregierung und auf die DLR-Publikationsdatenbank Elib angewendet.

2.1 Datenbereinigung

Die ersten drei Schritte der Korpus-Generierung wurden in der Open-Source Bibliothek spaCy in Python umgesetzt. Dies ist eine bekannte Bibliothek um natürliche Sprache zu verarbeiten (das heißt natural language processing, NLP). Hierbei können beispielsweise unstrukturierte Textdateien in einer Aneinanderreihung von Wörtern oder nur bestimmte Wortarten im Dokument ausgegeben werden.

1. Entfernung Stoppwörter: Häufig vorkommende Wörter, die für den Inhalt der Texte keine Bedeutung haben, werden Stoppwörter genannt und sind in einer vorgefertigten sogenannten Stoppwortliste zusammengefasst. Beispiele für Einträge der Liste sind Artikel, Fragewörter, Konjunktionen, Präpositionen. Alle Wörter der Stoppwortliste werden in diesem Schritt aus dem Korpus entfernt.

2. Beschränkung auf Nomen, Adjektiven und Eigennamen: Bei der Topic-Findung sind in unseren Anwendungsfällen vor allem Nomen, Eigennamen und Adjektive interessant.

3. Lemmatisierung: Hierbei wird jedes Wort auf seine Grundform zurückgeführt. Diese Grundform wird Lemma genannt. Beispielsweise wird Katzen zu Katze und intelligenter zu intelligent.

4. Entfernung bedeutungsloser Zeichenketten: Für die Topic-Identifikation sind Zahlen und nicht lesbare Zeichen nicht von Interesse. Daher werden diese, sofern sie erkannt werden, aus dem Korpus entfernt.

2.2 Keyword Extraction

Hier wird der letzte Schritt der Textaufbereitung beschrieben. Dieser erfolgt nach den vier Schritten der Datenbereinigung im vorherigen Kapitel 2.1.

5. Anwendung einer Keyword Extraction – Positional idfRank: Der Korpus wird weiter reduziert, indem eine Keyword Extraction, die sogenannte Positional idfRank, angewendet wird. Sie ist auf den vorliegenden Korpus abgestimmt.

Die Keyword-Extraction-Methode Positional idfRank wurde von Andreas Hamm entwickelt. Eine Übersicht sowie ein Anwendungsbeispiel ist in [29] zu finden. Die Idee der Methode ist es, zwei bekannte Verfahren der Keyword Extraction, nämlich das *tf-idf*-Maß und das PositionRank-Verfahren, miteinander zu kombinieren.
Das *tf-idf*-Maß bestraft Wörter im Korpus, die in vielen Dokumenten vorkommen, da diese weniger für die Beschreibung eines spezifischen Topics relevant sind. Außerdem fließt die Worthäufigkeit innerhalb des Dokuments in die Berechnung mit ein. Somit erhalten die Wörter einen hohen *tf-idf*-Wert, wenn sie häufig im Dokument, aber nur in verhältnismäßig wenigen Dokumenten vorkommen.
Dieses Maß setzt sich nach [14, S.68 ff.] aus zwei Bestandteilen zusammen: Der Term tf_{x_d} beschreibt die normierte Vorkommenshäufigkeit eines Wortes x in einem Dokument d. Aufgrund der Tatsache, dass verschiedene Dokumente unterschiedlich lang sein können, ist es hier sinnvoll, die absolute Häufigkeit zu normieren

$$tf_{x,d} = \frac{|x_d|}{\sum_{x' \in d} |x'_d|}. \tag{1}$$

Hierbei steht $|x_d|$ für die Anzahl der Wörter x im Dokument d. Neben der Vorkommenshäufigkeit der Wörter ist die inverse Dokumentenhäufigkeit, idf_x, relevant

$$idf_x = log\left(\frac{N}{\sum_{d,x \in d} 1}\right). \tag{2}$$

Hierbei ist N die Anzahl der Dokumente im Korpus und $\sum_{d,x \in d} 1$ die Anzahl der Dokumente, die das Wort x enthalten. Je kleiner die Anzahl der Dokumente ist, die das Wort x enthalten, desto größer wird der *idf*-Wert des Wortes. Das *tf-idf*-Maß setzt sich dann zusammen aus dem Produkt dieser beiden Werte 1 und 2

$$tf\text{-}idf_{x,d} = tf_{x,d} \cdot idf_x. \tag{3}$$

Der *tf-idf* begünstigt es nun, wenn Wörter selten im Korpus vorkommen, aber häufig in den einzelnen Dokumenten. Das Maß enthält allerdings bisher keinerlei Information über

die Position der Wörter innerhalb der Dokumente. Da der PositionRank die Position der Wörter in den Dokumenten bei der Berechnung eines Wertes mit einbezieht, wird dieser im nächsten Schritt eingeführt.
Der TextRank basiert auf der Idee des PageRank-Algorithmus. Ziel des PageRanks ist es, jede Seite im World Wide Web nach Wichtigkeit zu bewerten. Im TextRank hingegen ist es so, dass die Wörter im Korpus nach ihrer Wichtigkeit bewertet werden sollen. Um hinterher den TextRank zu verstehen, folgt hier zunächst eine kurze Erläuterung des PageRanks basierend auf [15]:
Die Seiten im World Wide Web können anhand ihrer Verlinkungsstruktur gewichtet und somit nach ihrer Wichtigkeit bewertet werden. Für jede Webseite wird ein Gewicht, nämlich der PageRank-Wert, berechnet. Das Prinzip lautet: Je mehr Links auf eine Seite verweisen, desto größer ist das Gewicht dieser Seite. Seien $T_1, ..., T_n$ Webseiten, die auf die Seite A zeigen, sei $\alpha \in (0, 1)$ ein Dämpfungsfaktor und sei $C(A)$ die Anzahl der Links, die von der Seite A auf andere Webseiten zeigen, gegeben. Dann lässt sich der PageRank-Wert der Webseite A folgendermaßen berechnen

$$PR(A) = \frac{\alpha}{n} + (1 - \alpha) \left(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)} \right). \tag{4}$$

Der PageRank soll das Verhalten eines Random Surfers im Internet wiedergeben. Dieser startet auf einer beliebigen Webseite und klickt sich von dort aus über Links weiter. Der Faktor α in der PageRank-Formel 4 stellt die Wahrscheinlichkeit dafür dar, dass der Surfer auf einer beliebigen, zufälligen Seite neu startet. Somit wird verhindert, dass der Surfer auf einer Webseite endet, von der keine weiteren Links mehr zu anderen Seiten gehen. Die berechneten PageRank-Werte geben die Wahrscheinlichkeitsverteilung über alle Webseiten an, das heißt, dass die Summe über alle PageRanks 1 ergibt.
Nach der Erläuterung des *tf-idf*-Maßes und des PageRank-Algorithmus, als Veranschaulichung des TextRank-Algorithmus, können nun die einzelnen Schritte der 5. Anwendung einer Keyword Extraction – Positional idfRank dargestellt werden:

- Erstellung des Netzwerks: Für jedes Wort im gemäß Schritt 1 bis 4 aufbereiteten Korpus wird ein Knoten im Netzwerk erstellt. Zwei Knoten werden durch eine Kante verbunden, wenn sie im selben zu definierenden Nachbarschafts-Fenster mit einer Größe von f Wörtern auftreten. Der Parameter f ist hierbei frei zu wählen. Das Kantengewicht zwischen den Wörtern x und y wird folgendermaßen gesetzt:

$$w_{x,y} = pf_{x,y} \cdot idf_x \cdot idf_y, \tag{5}$$

wobei $pf_{x,y}$ die Häufigkeit des gemeinsamen Auftretens von x und y im Fenster f ist. Der Parameter idf_x gibt die inverse Anzahl all der Dokumente an, die das Wort x enthalten (vgl. Formel 2).

- Bewertung der Position des Wortes im Dokument: Der Parameter pos_x wird bestimmt durch die Nummerierung der Wörter im jeweiligen Dokument. Enthält das Dokument beispielsweise 100 Wörter, so ist die Position *pos* des ersten Wortes gleich 1 und *pos* des letzten Wortes gleich 100. Es wird $p_x = (1 + pos_x)^\beta$ für ein frei zu wählendes $\beta \leq 0$ bestimmt. Je größer pos_x, desto kleiner wird p_x. Das heißt ein Wort zu Beginn eines Dokuments hat einen größeren p_x Wert als die abschließenden Wörter des Dokuments.

- Simulation eines Random Reader anstelle eines Random Surfers (vgl. PageRank): Das Wort x wurde im t-ten Schritt gelesen. Die Wahrscheinlichkeit für das nächste gelesene Wort y wird mithilfe folgender Markov-Kette bestimmt: Es gilt $x_{t+1} = Gx_t$, wobei

$$G_{x,y} = \alpha\widetilde{w_{x,y}} + (1 - \alpha)p_x\delta_{x,y}. \tag{6}$$

  Das α gibt hier analog wie in Formel 4 die Wahrscheinlichkeit für ein zufälliges Springen im Dokument an. $\widetilde{w_{x,y}}$ stellt hierbei das normierte Gewicht dar, das heißt

$$\widetilde{w_{x,y}} = \frac{|w_{x,y}|}{max_{x \in d} w_{x,y}}. \tag{7}$$

  Der Term $(1 - \alpha)p_x\delta_{x,y}$ gibt an, dass man mit Wahrscheinlichkeit $(1 - \alpha)p_x\delta_{x,y}$ auf ein Wort außerhalb des Fensters springt, wobei wegen der pos_x-Abhängigkeit eher Wörter am Anfang gewählt werden.

In [29] wurden die Parameter an einem Datensatz SemEval (vgl. [17]) optimiert und anschließend die folgenden Werte gewählt: Die Fenstergröße $f = 11$, die Wahrscheinlichkeit $\alpha = 0.9$ und der Parameter für die Bewertung der Position des Wortes $\beta = -0.9$. Da sowohl die Kleinen Anfragen an die Bundesregierung als auch die DLR-Publikationen ähnliche Eigenschaften aufweisen wie der Datensatz SemEval, wurden die optimierten Parameter auch in der Keyword Extraction in den Kleinen Anfragen und den DLR-Publikationen verwendet.

2.3 Wortreihenfolge in Topics

Die Community-Detection-Verfahren erzeugen Wortlisten, die als Topics interpretiert werden. Um diese Interpretation zu erleichtern, stellt es sich als hilfreich dar, die Wörter in

den Topics in einer Reihenfolge nach Bedeutsamkeit zu sortieren. So kann der Analyst durch Ablesen der oberen $15 - 30$ Wörter des jeweiligen Topics erkennen, um welches Thema es sich dabei handelt.
Eine einfache und naheliegende Möglichkeit ist es hier, die Wörter nach der Häufigkeit ihres Auftretens im gesamten Korpus zu sortieren. Somit werden allgemeiner verwendete Wörter weiter nach oben und selten verwendete Wörter weiter nach unten gerankt. Diese Sortierung werden wir in der Masterarbeit das Common Word Ranking nennen.
Eine andere Möglichkeit, die Wörter sinnvoll nach ihrer Wichtigkeit zu sortieren, ist es, die berechneten Positional-IdfRank-Werte für jedes Wort in jedem Dokument zu verwenden. Den 5 % der Wörter des jeweiligen Dokuments mit dem höchsten Positional-IdfRank-Werten werden 3 Sterne zugeteilt, den nächsten 5 % 2 Sterne und wiederum den nächsten 5 % 1 Stern. Allen übrigen 85 % der Wörter des Dokuments werden 0 Sterne zugeteilt. Für jedes Dokument liegt also nun eine Einteilung der Wörter in vier Gruppen vor. Aufgrund der prozentualen Sternchen Vergabe sind in längeren Dokumenten mehr Wörter mit Sternchen versehen als in kürzeren Dokumenten.
Die Anzahl der Positional-IdfRank-Werte für ein Wort ist gleich der Anzahl der Dokumente, die das jeweilige Wort enthalten. Wir möchten allerdings für jedes Wort nur genau einen Wert berücksichtigen. Eine Möglichkeit wäre es hierbei den durchschnittlichen Sternwert für jedes Wort zu berechnen. Angenommen aber es gibt ein Wort x, das nur in zwei unterschiedlichen Dokumenten auftritt und beide Male mit 3 Sternchen bewertet wurde. Hingegen gibt es ein Wort y, dass in 50 unterschiedlichen Dokumenten im Korpus auftritt, hier 45 Mal 3 Sterne und 5 mal 2 Sterne erhalten hat. Dann schließt das Wort x bei einer einfachen Mittelwertberechnung besser ab, obwohl es doch eigentlich viel seltener auftritt als das Wort y und damit weniger Bewertungen aufweist. Der Bayessche-Durchschnitt löst genau dieses Problem. Das heißt, je häufiger das Wort in unterschiedlichen Dokumenten auftritt, desto mehr Gewicht erhält dieses Wort. Der Bayessche-Durchschnitt des Wortes x berechnet sich mit den Positional-IdfRank-Werten x_i abhängig von den zu wählenden Parametern C und m wie folgt

$$\overline{x} = \frac{C \cdot m + \sum_{i=1}^{m} x_i}{C + \sum_{d, x \in d} 1}. \tag{8}$$

Der Parameter m steht für den Prior der durchschnittlichen Anzahl der Sterne und C steht für die Anzahl der Ratings die mit dem Prior-Wert m zu den tatsächlichen Ratings addiert werden. Angenommen wir haben immer den gleichen Datensatz, dann sollte der Parameter C groß gewählt werden, da wir die Lösung m kennen. Wenn der Datensatz allerdings immer eine zufällige Größe hat, dann sollte ein kleineres C gewählt werden, da wir kein

so großes Vertrauen in unseren Prior m haben. Da der Prior m für die durchschnittliche Anzahl der Sterne steht, berechnen wir hierfür den Erwartungswert aller vorkommenden Sterne

$$m = 0.85 \cdot 0 + 0.05 \cdot 1 + 0.05 \cdot 2 + 0.05 \cdot 3 = 0.3. \tag{9}$$

Als Parameter C möchten wir die faire Anzahl der Stimmen wählen, die jedes Wort bekommen sollte. Und dafür bestimmen wir abhängig vom Korpus die durchschnittliche Anzahl der Dokumente, in denen das Wort auftritt. Das heißt wir berechnen für jedes Wort, in wie vielen Dokumenten es auftritt und addieren dies über alle Wörter im Korpus auf und teilen es durch die Anzahl aller Wörter L_c im Korpus

$$C = \frac{\sum_{\forall x} \sum_{d, x \in d} 1}{L_c}. \tag{10}$$

Der berechnete Wert ist von den jeweiligen Zahlen im Korpus abhängig und wird in Kapitel 7.4 berechnet. Diese Sortierung der Wörter nennen wir in der Masterarbeit das Bayes Ranking.

3 Netzwerkbildung

Um eine Sammlung von Texten mit Netzwerk-Methoden analysieren zu können, muss vorher aus der gegebenen Textsammlung ein Netzwerk erstellt werden. Zunächst wird dafür in diesem Kapitel in die graphentheoretischen Grundlagen eingeführt. Anschließend werden dann zwei Methoden zur Netzwerkbildung aus den gegebenen Textsammlungen beschrieben. Auf diese Graphen können hinterher sogenannte Community-Detection-Verfahren zur Topic-Findung angewendet werden.

3.1 Graphentheoretische Grundlagen

In diesem Kapitel werden die Grundlagen der Graphentheorie nach Diestel [1] eingeführt. Ein Netzwerk oder ein Graph $G = (V, E)$ besteht aus einer endlichen Menge V von Knoten und einer endlichen Menge $E = \{(x, y)|x, y \in V\}$ von Kanten. In ungerichteten Graphen besteht E aus ungeordneten Paaren; in gerichteten Graphen aus geordneten Paaren. Somit beschreibt (x, y) und (y, x) im ungerichteten Fall die gleiche Kante, wohingegen die beiden Kanten im gerichteten Fall verschieden sind. In dieser Arbeit stehen vor allem ungerichtete Graphen im Vordergrund. Die Begrifflichkeiten werden im Folgenden für ungerichtete Graphen definiert, lassen sich aber problemlos auch auf den gerichteten Fall übertragen. Die Anzahl der Knoten $|V|$ wird häufig mit n bezeichnet und die Anzahl der Kanten $|E|$ häufig mit der Variable m.
Ein Knoten x heißt inzident zu einer Kante e, wenn er ein Endknoten dieser Kante ist, das heißt wenn $x \in e$ gilt. Zwei Knoten $x, y \in V$ heißen benachbart oder adjazent, wenn $(x, y) \in E$ gilt. Die Menge der Nachbarn eines Knotens $x \in V$ wird mit $N_G(x)$ oder kurz mit $N(x)$ bezeichnet. Der Knotengrad eines Knotens $x \in V$ ist definiert als die Anzahl seiner Nachbarn. Er wird mit $deg(x)$ bezeichnet. Ein Knoten mit Grad 0 wird isolierter Knoten genannt.
Ein Pfad ist ein nichtleerer Graph $P = (V, E)$ mit $V = \{x_0, x_1, ..., x_k\}$ und
$E = \{(x_0, x_1), ..., (x_{k-1}, x_k)\}$, wobei x_i für $0 \leq i \leq k$ paarweise verschieden sind. x_0 und x_k sind Endknoten von P, die durch den Pfad verbunden sind und $x_1, ..., x_{k-1}$ sind innere Knoten. Ein Pfad kann gelegentlich durch eine Knotenreihenfolge $x_0, x_1, ..., x_k$ identifiziert werden. Die Länge des Pfades ist die Anzahl der Kanten im Pfad. Für zwei Knotenmengen A, B bezeichnet $P = (V, E)$ einen $A - B$ Pfad, wenn $V \cap A = \{x_0\}$ und $V \cap B = \{x_k\}$ gilt. Einen $\{a\} - B$ Pfad bezeichnen wir kürzer als einen $a - B$ Pfad. Sei ein Pfad $P = (V, E) = (\{x_0, ..., x_{k-1}\}, \{(x_0, x_1), ..., (x_{k-2}, x_{k-1})\})$ mit $k \geq 3$ in G gegeben, dann ist der Graph $(V, E) \cup \{(x_{k-1}, x_0)\}$ ein Kreis. Wir bezeichnen diesen als Knotenfolge $C = x_0...x_{k-1}x_0$.
Ein nichtleerer Graph $G = (V, E)$ heißt zusammenhängend, wenn er für jeweils zwei seiner

Knoten $x, y \in V$ ein $x - y$ Pfad enthält.
Ein Baum ist ein Graph, der keine Kreise enthält und zusammenhängend ist. Knoten vom Grad 1 im Baum sind die Blätter des Baums. Alle anderen Knoten werden innere Knoten genannt. Ein gewurzelter Baum ist ein Baum, dessen Kanten eine ausgezeichnete Richtung besitzen. Das heißt, dass die Wurzel keine eingehende Kante und alle anderen Knoten im Baum genau eine eingehende Kante haben. Somit kann ein Knoten im gewurzelten Baum als Wurzel identifiziert werden. Zeigt in einem Baum eine Kante (x, y) von x nach y, so heißt x Vater/Vorgänger von y und y Kind/Nachfolger von x. Ein Binärbaum ist ein gewurzelter Baum in dem jeder Knoten nur höchstens zwei Kinder hat.
Ein Graph G kann dargestellt werden, indem die Knoten V als Punkte gezeichnet werden und zwei Punkte genau dann durch eine Linie verbunden werden, wenn die entsprechenden Knoten in G benachbart sind. Desweiteren kann ein Graph durch seine Adjazenzmatrix angegeben werden. Eine Adjazenzmatrix A enthält die Information, welche Knoten im Graphen durch eine Kante verbunden sind. Die Einträge der Adjazenzmatrix eines Graphen sind wie folgt definiert:

$$A_{x,y} = \begin{cases} 1, \text{ falls eine Kante zwischen Knoten } x \text{ und Knoten } y \text{ existiert.} \\ 0, \text{ sonst.} \end{cases} \tag{11}$$

Ein Graph G heißt gewichtet, wenn jeder Kante $e = (x, y) \in E$ ein Gewicht w zugeordnet wird. Das Kantengewicht wird durch eine Kantenfunktion $w : E \to \mathbb{R}$ gegeben. Das Kantengewicht der Kante $e = (x, y)$ wird mit $w_{x,y}$ bezeichnet. Die Einträge einer Adjazenzmatrix eines gewichteten Graphen werden mithilfe seiner Gewichte definiert:

$$A_{x,y} = \begin{cases} w_{x,y}, \text{ falls eine Kante zwischen Knoten } x \text{ und Knoten } y \text{ existiert.} \\ 0, \qquad \text{sonst.} \end{cases} \tag{12}$$

Ein Random Walk in einem Graph lässt sich folgendermaßen beschreiben: Sei ein Startknoten gegeben, dann wird ein zufälliger Nachbar des Startknotens mit Wahrscheinlichkeit proportional zum entsprechenden Kantengewicht besucht; dann wird von diesem Knoten aus ein zufälliger Nachbar besucht usw. Die zufällige Folge der so gewählten Knoten ist dann ein Random Walk im Graphen [9]. Ein ungerichteter Graph $G = (V, E)$ ohne Mehrfachkanten und ohne Schleifen heißt bipartiter Graph, wenn sich die Menge V in zwei disjunkte Knotenmengen aufteilen lässt, sodass die Knoten innerhalb dieser Mengen nicht durch eine Kante verbunden sind.

3.2 Kookkurrenznetzwerke

In diesem graphentheoretischen Ansatz wird ein ungerichtetes, gewichtetes Netzwerk $G = (V, E)$ erzeugt, mit der Knotenmenge V und der Kantenmenge E, wobei gilt [4]:

$$\begin{cases} V & \hat{=} \quad \text{verschiedene Wörter in der Dokumentensammlung.} \\ E & \hat{=} \quad \text{Co-occurence der Wortpaare (d.h. gemeinsames Auftreten im Dokument).} \end{cases}$$

Die Kanten der Menge E können hierbei nach der Häufigkeit und der Art des Zusammenauftretens gewichtet werden. Hierfür werden wir in den folgenden zwei Unterkapiteln die gewählten Gewichtsalternativen a und b vorstellen.

3.2.1 Gewichtswahl a

Wenn zwei Wörter x und y gemeinsam in einem Dokument auftreten, wird das Gewicht $w_{x,y}$ der Kante zwischen diesen zwei Knoten x und y um 1 erhöht. Für die Beschreibung der Gewichte zwischen den Knoten definieren wir die Adjazenzmatrix A von G durch ihre Einträge $A_{x,y}$ wie folgt

$$A_{x,y} = \begin{cases} w_{x,y}, \text{ falls eine Kante zwischen Knoten } x \text{ und Knoten } y \text{ existiert.} \\ 0, \quad \text{sonst.} \end{cases} \tag{13}$$

Als Veranschaulichung betrachten wir den Korpus aus drei Dokumenten in Tabelle 1:

Dokument	**Text**
Dok1	LDA ist ein bekanntes Topic Modeling Verfahren.
Dok2	Alternativ ist dies mit dem Community Detection Verfahren möglich.
Dok3	Topic Modeling mit Community Detection liefert gute Ergebnisse.

Tabelle 1: Dokumente mit Text

Auf diesen Korpus werden die in Kapitel 2.1 erläuterten Schritte angewendet. Somit bleiben nur die in Abbildung 1 dargestellten Wörter übrig. Die Wörter werden jeweils durch eine Kante im Netzwerk verbunden und mit Gewicht 1 initialisiert, wenn sie gemeinsam in einem Dokument vorkommen. Tauchen diese zwei Wörter erneut zusammen in einem anderen Dokument auf, so wird das Gewicht wiederum um 1 erhöht.

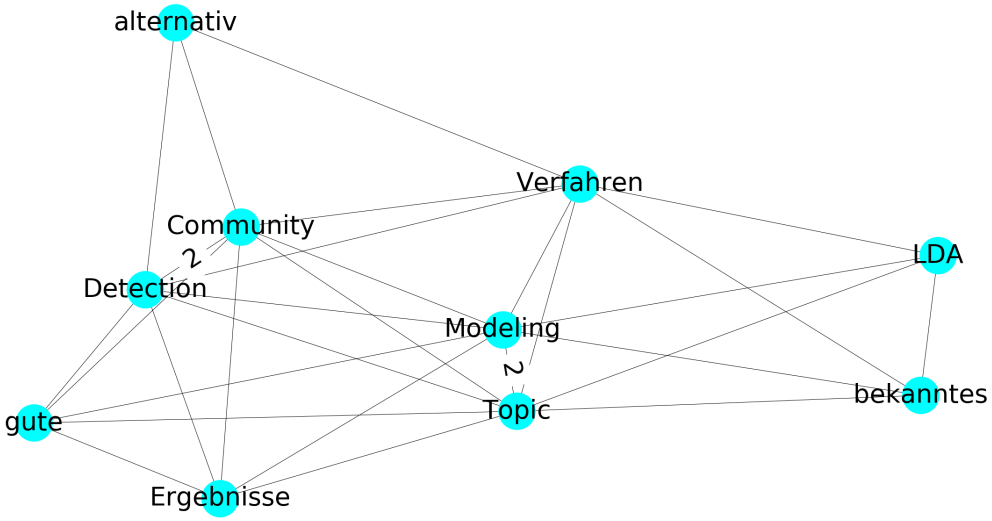

Abbildung 1: Gewichtetes Netzwerk mit Nomen und Adjektiven aus Text

3.2.2 Gewichtswahl b

Mit dem Ziel die Anzahl der Kanten im Netzwerk zu reduzieren und die Häufigkeiten der Wörter innerhalb der Dokumente mit einzubeziehen wird die Gewichtsalternative b eingeführt. Die Idee der Gewichtsberechnung ist es, von der gemeinsamen Auftrittshäufigkeit der Wörter eine zufällige Auftrittshäufigkeit abzuziehen. Zunächst werden hierfür einige neue Parameter definiert und aufbauend darauf in Anlehnung an [18] die Gewichtswahl b erläutert.

Sei $s_x = \sum_{x \in d} |x_d|$ die Häufigkeit des Wortes x in allen Dokumenten im Korpus, wobei $|x_d|$ die Vorkommenshäufigkeit des Wortes x im Dokument d ist (vgl. Kapitel 2). Weiter sei L_d die Anzahl der Wörter im Dokument d.

Wir berechnen im Folgenden $z_{x,y}$, die Häufigkeit des Zusammenauftretens der Wörter x und y. Dies wird berechnet, indem man die Skalarprodukte der Auftrittshäufigkeiten summiert:

$$z_{x,y} = \sum_{d} |x_d| \cdot |y_d|. \tag{14}$$

Wenn Wörter zusammen in einem Dokument auftreten, kann es Zufall sein. Bei der Gewichtswahl b berücksichtigen wir das Zusammenauftreten nur dann, wenn es so häufig vorkommt, dass es unwahrscheinlich durch Zufall erklärt werden kann. Sei s_x die Häufigkeit des Wortes x im Korpus und L_c die Gesamtzahl der Wörter im Korpus. Dann wird die Frage, wie wahrscheinlich es ist, dass im Dokument d mit Wortanzahl L_d das Wort x x_d-mal vorkommt, von der hypergeometrischen Verteilung beantwortet. Ihr Erwartungswert

ist

$$E(|x_d|) = \frac{L_d \cdot s_x}{L_c}. \tag{15}$$

Da eine sehr große Anzahl an Dokumenten vorliegt, dürfen wir annehmen, dass die Zufallszahlen $|x_d|$ und $|y_d|$ nicht korrelieren und stochastisch unabhängig sind. Das Produkt in der Erwartungswertberechnung von $z_{x,y}$ kann folgendermaßen geschrieben werden:

$$E(z_{x,y}) = E(\sum_d |x_d| \cdot |y_d|) = \sum_d E(|x_d|) \cdot E(|y_d|) = \frac{s_x s_y \sum_d L_d^2}{L_c^2}. \tag{16}$$

Da die Grundgesamtheit, das heißt die Anzahl der Wörter im Korpus, sehr groß ist, und die Wahrscheinlichkeit, dass Wörter gemeinsam auftreten gleichzeitig sehr klein ist, kann für letztere die Poisson-Verteilung $\text{Pois}_{z_{x,y}}(z)$ verwendet werden.
Die Gewichte werden nun berechnet, indem von der beobachteten Häufigkeit des gemeinsamen Auftretens der Wörter ein Nullmodell abgezogen wird, welches das zufällige Zusammenauftreten der Wörter mithilfe der Poisson-Verteilung berechnet:

$$w_{x,y} = z_{x,y} - Z_p(s_x, s_y). \tag{17}$$

Die Kante mit dem jeweiligen Gewicht wird nur dann erzeugt, wenn $w_{x,y} > 0$ gilt. Das Nullmodell wird folgendermaßen berechnet:

$$Z_p(s_x, s_y) = max_v\{v| \sum_{z=v}^{\infty} Pois_{z_{x,y}}(z) > p\}. \tag{18}$$

Für den p-Wert wählen wir 0.05. Das heißt, wenn die aufsummierte Wahrscheinlichkeit $\sum_{z=v}^{\infty} Pois_{z_{x,y}}(z)$ größer ist als das 0.05-Quantil, dann ist es wahrscheinlich, dass x und y nur zufällig gemeinsam auftreten.

3.3 Wort-Dokument-Netzwerke

Anstelle eine Kookkurrenznetzwerks (vgl. Kapitel 3.2) kann ein Korpus auch durch ein sogenanntes Wort-Dokument-Netzwerk beschrieben werden. Das Wort-Dokument-Netzwerk eines Korpus ist ein bipartites Netzwerk $G = (V, E)$, wobei V und E wie folgt definiert sind:

$$\begin{cases} V & \triangleq & \text{Dokumente } D \text{ und Wörter } W \text{ in der Dokumentensammlung.} \\ E & \triangleq & \text{Kante zwischen } d \in D \text{ und } w \in W, \text{ falls Wort } w \text{ in Dokument } d \text{ auftritt.} \end{cases}$$

Die Knotenmenge besteht also aus der Menge D der Dokumente sowie der Menge W der

Wörter der Dokumentensammlung. Ein Dokument d wird genau dann mit einem Wort w verbunden, wenn w in d vorkommt. Somit ist $\{D, W\}$ eine Bipartition von G. In Abbildung 2 ist beispielhaft ein Wort-Dokument-Netzwerk eines Korpus aus 3 Dokumenten und 5 Wörtern dargestellt.

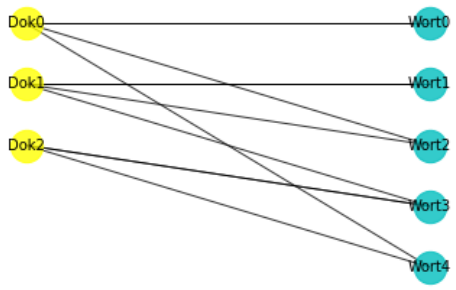

Abbildung 2: Bipartites Netzwerk aus Dokumenten und Wörtern

Auf das Wort-Dokument-Netzwerk wird im hierarchical Stochastic Block Model (hSBM) zurückgegriffen (siehe Kapitel 6).

3.4 Community Detection

Komplexe Netzwerke, wie beispielsweise Soziale Netzwerke, Kommunikationsnetzwerke, Transportnetzwerke und biologische Netzwerke wachsen in ihrer Größe, Vielfalt und Komplexität. Im Zeitalter der Digitalisierung tauchen auch neuere, vorher nicht da gewesene Typen von Netzwerken, wie Internet of Things oder Multiagente Systeme, auf.
Um all solche zu verstehen und zu interpretieren, wollen wir Gemeinschaftsstrukturen (Community Structures) darin erkennen. Dies ist das am meisten untersuchte strukturelle Merkmal in Netzwerken ([23], S.1). Community Detection ist die Methode, die solche Gemeinschaftsstrukturen in Form von Communities erkennt. Communities in einem Netzwerk sind Gruppierungen von Knoten, die häufiger untereinander verbunden sind als zu den restlichen Knoten im Netzwerk. Betrachtet man beispielsweise ein Soziales Netzwerk, so sind die gefundenen Communities als Freundschafts- bzw. Bekanntschaftscommunities zu identifizieren. In dem in dieser Masterarbeit betrachteten Anwendungsfall des Topic Modelings werden Wort-Communities als Topics identifiziert.
Sei eine Community $C \subseteq V$ als eine Teilmenge der Knoten des Netzwerks $G = (V, E)$ definiert. Dann lässt sich eine Partition $\mathcal{C} = \{C_1, C_2, ..., C_t\}$ von V in Communities definieren, sodass jedes $v \in V$ in genau einer Community $C_s \in \mathcal{C}$ liegt, wobei $1 \leq s \leq t \leq |V|$ ([3]). Ziel der Community Detection ist es nun eine optimale Partition $\mathcal{C}$ der Knoten in

Communities zu finden, die Aufschluss über die Gemeinschaftsstrukturen im Netzwerk gibt.
Es gibt viele Techniken eine optimale Community-Einteilung $\mathcal{C}$ in einer guten Laufzeit zu finden. Eine gute Übersicht findet man in der Grundlagenliteratur von Fortunato [24]. Drei der meist verwendeten Methodentypen werden hier vorgestellt und jeweils ein Repräsentant zu jedem Typ wird in den darauffolgenden Kapiteln erläutert:

- Nullmodell: Das sind Methoden, die Maße, welche die Konnektivität zwischen Gruppen von Knoten mit denen der zufälligen Verteilung von Knoten vergleichen. Wenn die Konnektivität der Knoten am meisten von dem Nullmodell abweicht, das heißt maximal ist, so nennen wir die gefundene Gruppeneinteilung der Knoten Community. Die bekannteste und am weit verbreitetste Methode ist hierbei die Optimierung der Modularitätsfunktion, welche in Kapitel 4.1 explizit erläutert wird.

- Random Walk: Der Random Walk im Netzwerk (vgl. Kapitel 3.1 Graphentheoretische Grundlagen) wird genutzt um Communities zu identifizieren. Wenn der Walk auf ein beliebiges Netzwerk zur Community Detection angewendet wird, so neigt er dazu eine längere Zeit innerhalb einer dichten Community zu bleiben, da dort eine hohe Dichte an Pfaden vorliegt [23]. In diesem Verfahrenstyp steht die Dynamik des Netzwerks im Vordergrund und nicht wie beispielsweise beim Nullmodell die Topologie. Ein bekanntes Fluss-basiertes Verfahren ist Map Equation. Dies werden wir in Kapitel 5 näher erläutern.

- Blockmodell: Ein weiterer Methodentyp zur Community Detection sind Block Models. Die Verfahren identifizieren mithilfe der Maximierung der Likelihood Funktion Blöcke von Knoten, die häufig miteinander verbunden sind. Knoten desselben Blocks sind also statistisch äquivalent in Bezug auf die Konnektivität der Knoten im Block und außerhalb des Blocks. Als Repräsentanten sehen wir uns hierfür das hierarchical Stochastic Block Model (hSBM) in Kapitel 6 genauer an.

4 Modularitätsoptimierung

Das erste hier betrachtete Verfahren zur Community Detection ist der Louvain-Algorithmus, welcher die sogenannte Modularitätsfunktion optimiert.
Zum Verständnis des Verfahrens wird zunächst die Modularität in Kapitel 4.1 definiert, um im darauffolgenden Kapitel 4.2 den Louvain-Algorithmus zu erläutern.

4.1 Modularität

Das Problem der Community Detection, (vgl. Kapitel 3.4) besteht darin, Knoten eines Netzwerks so in Communities einzuteilen, sodass Knoten innerhalb der Communities häufig und Knoten unterschiedlicher Communities selten verbunden sind [2].
Die Qualität einer solchen Knoten-Einteilung kann mithilfe der Modularitätsfunktion berechnet werden ([2], S.2). Diese misst die Qualität der vorliegenden Community-Einteilung, indem die Häufigkeit der Kanten innerhalb der Community mit der zu erwartenden Häufigkeit von zufälligen Verbindungen zwischen den Knoten verglichen wird.
Sei ein gewichteter, ungerichteter Graph $G = (V, E)$ und eine Adjazenzmatrix A gegeben, welche als Einträge die Gewichte der Kanten in G enthält. Sei eine Community C und eine Partition $\mathcal{C}$ von V in Communities wie in Kapitel 3.4 definiert. Die Modularität einer Community-Einteilung $\mathcal{C}$ im Graph lässt sich dann folgendermaßen ausdrücken:

$$Q(\mathcal{C}) = \frac{1}{2m} \sum_{x,y \in V} \left(A_{x,y} - \frac{k_x k_y}{2m} \right) \delta_{c(x),c(y)}. \tag{19}$$

Der Index $c(x)$ im Kronecker-Delta δ ist der Index der Community, in der der Knoten x liegt. Das Kronecker-Delta δ ist hierbei wie folgt definiert:

$$\delta_{u,v} = \begin{cases} 1, \text{ falls u = v.} \\ 0, \text{ sonst.} \end{cases} \tag{20}$$

In der Definition der Modularität in Formel 19 ist $A_{x,y}$ der x, y-te Eintrag der Adjazenzmatrix, das heißt $\sum_{x,y \in V} A_{x,y} \delta_{c(x),c(y)}$ steht für die doppelt aufsummierten Gewichte von Kanten, die innerhalb einer Community verlaufen. Der Parameter m ist gleich dem Ausdruck $\frac{1}{2} \sum_{x,y \in V} A_{x,y}$, das heißt m ist das Gesamtgewicht aller Kanten im Graph [2]. Der Parameter $k_x = \sum_y A_{x,y}$ steht für die aufsummierten Gewichte der Kanten, welche inzident zu x sind. Das heißt k_x steht für die gewichteten Knotengrade von x und $\frac{\sum_{x,y} k_x k_y}{2m}$ steht somit für die Summe der Gewichte einer zufälligen Kantenverteilung zwischen x und y.

Da wir wegen des Kronecker-Deltas nur die Knoten x und y betrachten, welche in derselben Community C liegen, lässt sich die Modularität weiter zusammenfassen. Hierbei laufen wir über alle Communities in $\mathcal{C}$:

$$\begin{aligned} Q(\mathcal{C}) &= \frac{1}{2m}\left(\sum_{C\in\mathcal{C}}\sum_{x,y\in C} A_{x,y} - \frac{\sum_{C\in\mathcal{C}}\sum_{x,y\in C} k_x k_y}{2m}\right) \\ &= \frac{\sum_{C\in\mathcal{C}}\sum_{x,y\in C} A_{x,y}}{2m} - \sum_{C\in\mathcal{C}}\left(\frac{\sum_{x\in C} k_x}{2m}\right)^2. \end{aligned} \tag{21}$$

Diese Formel 21 vergleicht für alle Communities in der Community-Einteilung $\mathcal{C}$ und für alle Knotenpaare innerhalb der aktuellen Community, das Gewicht $A_{x,y}$ mit dem Erwartungswert des Gewichts im Zufallsgraphen mit gleichen Knotengraden.
Der Modularitätswert einer Partition liegt zwischen -1 und 1. Liegt der Wert nahe bei 0, so ist die Häufigkeit der Kanten innerhalb der Community nicht besser als eine zufällige Einteilung. Ist der Wert negativ, so ist die Qualität der gefundenen Community Einteilung schlechter als eine zufällige Einteilung. Ist der Modularitätswert möglichst groß, so hat die vorliegende Community-Struktur eine sehr gute Qualität ([36], S.47).
Die Modularität ist auch für andere Netzwerktypen definiert. So kann sowohl für ungerichtete als auch für gerichtete, sowohl für ungewichtete als auch für gewichtete Graphen eine Modularität berechnet werden. Eine Übersicht zu allen unterschiedlichen Modularitäten ist im Paper ([23], S.4f.) zusammengefasst.
Die Modularität kann somit zur Bewertung einer vorliegenden Community-Partition herangezogen werden. Sie kann allerdings auch in einem Algorithmus zur Community-Findung verwendet werden. Genau dies passiert im Louvain-Algorithmus, in dem eine abgewandelte Modularitätsformel optimiert wird, die wir im nächsten Teilkapitel näher erläutern werden.

4.2 Louvain-Algorithmus

Das Problem der Modularitätsoptimierung ist ein NP-hartes Problem [36]. Um das Problem in einer absehbaren Laufzeit zu lösen, wird als Heuristik ein Greedy-Ansatz verwendet. Ein Greedy-Ansatz zeichnet sich dadurch aus, dass er sich zum aktuellen Zeitpunkt für die Lösung oder den Folgezustand entscheidet, der in dem Moment den größten Gewinn verspricht.
Ein überzeugender Greedy-Algorithmus hinsichtlich schneller Laufzeit ist der Louvain-Algorithmus [6]. Er ermittelt eine hierarchische Struktur von Partitionen, indem er die Modularität maximiert.

Der Louvain-Algorithmus besteht aus zwei Phasen, die wiederum in einer Schleife eingebettet sind [2]. In der ersten Phase werden Knoten im Netzwerk in einer beliebigen Reihenfolge durchlaufen und in die jeweilige Nachbarcommunity mit dem größten Modularitätsgewinn verschoben. In Phase 2 wird ein neues Netzwerk erstellt, indem die gefundenen Communities zu jeweils einem Knoten zusammengefasst werden. Die zwei Phasen bilden zusammen ein sogenanntes Level und werden in der Schleife iterativ wiederholt, bis die Modularität nicht mehr verbessert werden kann. Für jedes Level gibt der Louvain-Algorithmus eine Community-Partition aus. Im folgenden ist der Algorithmus in einem Pseudocode 1 zusammengefasst, der hinterher ausführlich erläutert wird.

Algorithm 1 Louvain-Algorithmus

Input: $G = (V, E)$

$G^i = G$; $i = 0$

1: **repeat**

2: **Phase 1:**

3: **Input:** Graph $G^i = (V, E)$, $V = \{x_1, ..., x_n\}$

4: **Output:** Partition $\mathcal{C}^i = \{C_1, ..., C_m\}$ von G^i

5: - Initialisierung Communities: $C_j = \{x_j\}$ $\forall j \in \{1, ..., n\}$

6: **repeat**

7: - wähle zufällige Reihenfolge der Knoten

8: **for all** $x \in V$ **do** (Abarbeitung entsprechend der zufälligen Knotenreihenfolge)

9: - berechne $\Delta Q(x, C_u)$ für alle $u \in N(x)$, wobei C_u die Community ist, die u enthält und $\Delta Q(x, C_u)$ die Modularitätsveränderung bei Verschieben von x aus C_x in Community C_u.

10: **if** $\max_{u \in N(x)} \Delta Q(x, C_u) > 0$ **then**

11: - verschiebe x in Nachbarcommunity mit größtem Modularitätsgewinn.

12: **end if**

13: **end for**

14: **until** $\max_{u \in N(x)} \Delta Q(x, C_u) \leq 0$ $\forall x \in V$

15: **return** $\mathcal{C}^i$

16: **Phase 2:**

17: **Input:** Graph $G^i = (V^i, E^i)$, Community Partition $\mathcal{C}^i$

18: **Output:** Graph $G^{i+1} = (V^{i+1}, E^{i+1})$

19: - $V^{i+1} = \mathcal{C}^i$, $E^{i+1} = \emptyset$

20: **for all** Kanten $(x, y) \in E^i$ **do**

21: - füge Kante (C_x, C_y) zu E^{i+1} hinzu mit Gewicht

$$w(C_x, C_y) = \begin{cases} w(x, y), & \text{falls } C_x \neq C_y \\ 2 \cdot w(x, y), & \text{falls } C_x = C_y \end{cases}$$

bzw. erhöhe Gewicht um den Betrag, falls Kante bereits vorhanden.

22: **end for**

23: **return** $G^{i+1} = (V^{i+1}, E^{i+1})$

24: $i = i + 1$

25: **until** keine Änderung der initialen Communities in Phase 1

Als Input geben wir den Graph $G = (V, E)$ ein, dessen Community-Struktur uns interessiert. Im 0-ten Durchlauf wird dieser gleich G^0 gesetzt, in späteren Durchläufen wird G^i durch einen in Phase 2 gebildeten Graphen des obigen Durchlaufs ersetzt. Output der ersten Phase ist eine Community-Partition $\mathcal{C}^i$ des jeweiligen i-ten Durchlaufs mit G^i.
Phase 1 beginnt in Zeile 5 mit der Initialisierung der Communities. In der initialen Community-Partition bildet jeder Knoten eine eigene Community. Das heißt es gibt genauso viele Communities wie es Knoten gibt.
Im darauffolgenden Schritt in Phase 1, in Zeile 8 und 9, werden dann für eine zufällig gewählte Reihenfolge der Knoten, für jeden Knoten $x \in V$ alle Nachbarn $u \in N(x)$ betrachtet und der Modularitätsgewinn $\Delta Q(x, C_u)$ berechnet, der eintritt, wenn wir Knoten x in die jeweilige Nachbarcommunity C_u verschieben. Der Knoten wird in die Community verschoben, die den maximalen Modularitätsgewinn verspricht, vorausgesetzt der Gewinn ist positiv (vgl. Zeile 10-12). Falls kein Modularitätsgewinn zu verzeichnen ist, so bleibt der Knoten in der ursprünglichen Community. Diese Phase wird so häufig wiederholt, bis keine weitere Modularitätsverbesserung mehr möglich ist (vgl. Zeile 14) [2].
Wichtig ist hier zu verstehen, dass das Ergebnis dieser Phase von der Reihenfolge abhängt, in der man die Knoten durchläuft. Hier werden die Knoten in einer zufälligen Reihenfolge durchlaufen (vgl. Zeile 7). Unterschiedliche Reihenfolgen der Knoten können zu unterschiedlichen Ergebnissen führen, das heißt es liegt ein nichtdetermisitisches Verfahren vor.
Die Modularitätsveränderung $\Delta Q(x, C_u)$ im Fall, dass der Knoten x in die Nachbarcommunity C_u verschoben wird, lässt sich berechnen, indem die Modularität vor dem Verschmelzen der zwei Communities von der Modularität nach dem Verschmelzen der zwei Communities abgezogen wird. Bei der Berechnung einer Näherung des Modularitätsgewinns wird angenommen, dass der Knoten x isoliert ist, das heißt wir machen die Annahme dass er in einer Einzelcommunity C_1 liegt. Zunächst wird $\Delta Q(x, C_u)$ wie folgt umgeformt:

$$\begin{aligned}\Delta Q(x, C_u) = &\left[\underbrace{\frac{\sum_{v,b \in C_u} A_{v,b}}{2m}}_{(1)} + \underbrace{\frac{2\sum_{v \in C_u} A_{x,v}}{2m}}_{(2)} - \underbrace{\left(\frac{\sum_{v \in C_u} k_v + k_x}{2m}\right)^2}_{(3)}\right] \\ &- \left[\underbrace{\frac{\sum_{v,b \in C_u} A_{v,b}}{2m} - \left(\frac{\sum_{v \in C_u} k_v}{2m}\right)^2}_{(4)} + \underbrace{\frac{\sum_{x,y \in C_1} A_{x,y}}{2m}}_{(5)} - \underbrace{\left(\frac{k_x}{2m}\right)^2}_{(6)}\right].\end{aligned} \tag{22}$$

Der linke Teil des Terms 22 besteht aus (1) der Kantenmenge innerhalb der Community C_u, (2) der Kantenmenge inzident zu dem hinzugefügten Knoten x und (3) der Summe der Gewichte inzident zu Knoten in der Nachbarcommunity addiert mit den Gewichten inzident zu x. Somit entspricht dieser linke Teil des Terms der Modularitsfunktion 19, nur dass wir hierbei noch den Knoten x hinzugefügt haben.
In der zweiten Zeile des Terms, entspricht (4) der Modularitätsfunktion 19. (5) enthält die Summe aller Kantengewichte der Community, in der x liegt. Dieser Wert ist gleich 0, da wir die Annahme machen, dass der Knoten in einer Einzelcommunity liegt. Die Formel (6) sind die Gewichte der Kanten, die inzident zu dem hinzugefügten Knoten x sind. Die Formel 22 lässt sich vereinfachen zu

$$\Delta Q(x, C_u) = \frac{1}{m}\left(\sum_{v \in C_u} A_{x,v} - \frac{(\sum_{v \in C_u} k_v)k_x}{2m}\right). \tag{23}$$

Veranschaulichen wir uns den Zusammenhang der Formeln 23 und der Modularitätsformel 21 an dem Beispielgraph in Abbildung 3:

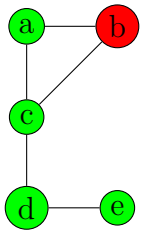

Abbildung 3: Zusammenhang Modularitätsformel 21 und 23 an einem Beispiel

Sei die Community Partition $\mathcal{C}^1 = \{C_1, C_2\}$ mit der Community $C_1 = \{b\}$, welche rot hinterlegt ist, und $C_2 = \{a, c, d, e\}$, welche grün hinterlegt ist, gegeben. Wir berechnen die Modularitätsveränderung, wenn wir den Knoten b zur Community C_2 hinzufügen und aus Community C_1 entfernen, zunächst mithilfe der Modularitätsformel 21.
Die Modularität wird zunächst für die gegebene Partition $\mathcal{C}^1 = \{C_1, C_2\}$ und dann für die neue Partition $\mathcal{C}^2 = \{C_2\}$, in der alle Knoten in C_2 liegen, berechnet und dann für die Ermittlung des Modularitätsgewinns voneinander abgezogen:

$$
\begin{aligned}
Q(\mathcal{C}^1) &= \left(\frac{6}{10} - \left(\frac{1+2+3+2}{10}\right)^2\right) + \left(0 - \left(\frac{2}{10}\right)^2\right) = -0.08 \\
Q(\mathcal{C}^2) &= \left(\frac{10}{10} - \left(\frac{10}{10}\right)^2\right) + (0-0) = 0 \\
\Delta Q(b, C_2) &= Q(\mathcal{C}^2) - Q(\mathcal{C}^1) = 0 - (-0.08) = 0.08
\end{aligned} \tag{24}
$$

Alternativ kann die Modularitätsveränderung auch mit der soeben eingeführten Formel 23 berechnet werden

$$
\Delta Q(b, C_2) = \frac{1}{5}\left(1 + 1 + 0 + 0 - \frac{(2+3+2+1)\cdot 2}{10}\right) = 0.08. \tag{25}
$$

Für diesen Fall wurde somit die Gleichheit der Formeln gezeigt

$$
\Delta Q(b, C_2) = Q(\mathcal{C}^2) - Q(\mathcal{C}^1). \tag{26}
$$

Im Allgemeinen gilt aber, dass die Formel 23 nur eine Näherung für die Berechnung darstellt. Es sollte noch angemerkt werden, dass der Louvain-Algorithmus nicht das einzige Modularitätsoptimierungsverfahren ist, sondern dass es sich gerade durch die Verwendung der Näherung der Formel 23 auszeichnet.
In der zweiten Phase des Algorithmus wird ein neues Netzwerk erstellt, indem jede Community des alten Netzwerks G^i zu einem Knoten im neuen Netzwerk G^{i+1} zusammengefasst wird (vgl. Zeile 19). Für die Berechnung der Kanten im neuen Netzwerk werden in Zeile 21 zwei Fälle unterschieden: Zunächst betrachten wir den Fall, dass die Knoten x, y in unterschiedlichen Communities liegen. Hier werden die neuen Gewichte gleich der Summe der alten Gewichte zwischen diesen Knoten gesetzt. Zweitens wird der Fall betrachtet, dass die Knoten x, y innerhalb einer Community im alten Netzwerk liegen. Dann werden die Gewichte für die Schleifen der neuen Knoten berechnet, indem die Summe der Gewichte der Kanten innerhalb dieser alten Community verdoppelt werden.
Wenn die zweite Phase durchlaufen ist, beginnt erneut die erste Phase. Phase 1 und Phase 2 werden solange wiederholt bis sich die initialen Communities in Phase 1 nicht ändern, das heißt dass die Modularität der initialen Community-Struktur in Phase 1 nicht weiter verbessert werden kann (vgl. Zeile 25).
Abschließend möchten wir uns den Louvain-Algorithmus für ein Level an einem Beispiel in Abbildung 4 veranschaulichen. Es ist ein Graph mit 12 Knoten gegeben.

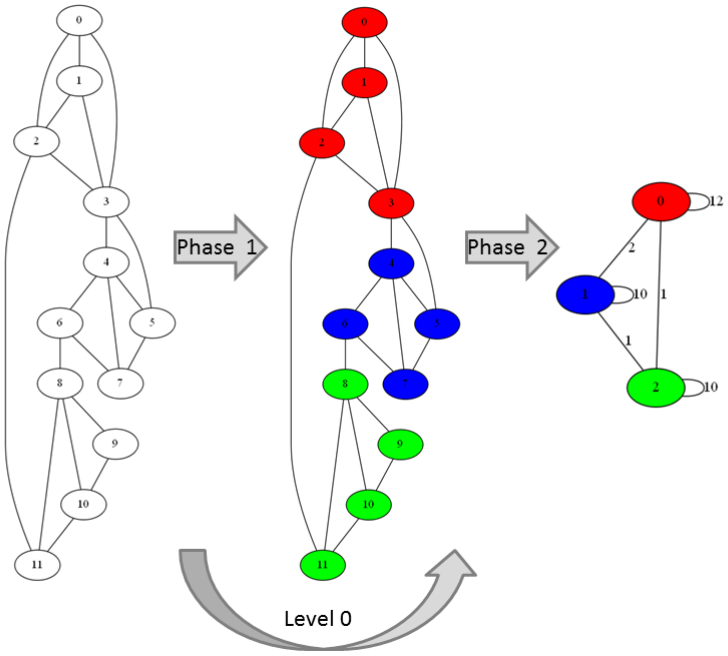

Abbildung 4: Vorgehensweise Louvain Algorithmus an einem Beispiel

In Phase 1 des Algorithmus werden drei Communities gefunden. Diese sind in der Abbildung in den Farben rot, blau und grün abgebildet. Phase 2 erzeugt dann ein neues Netzwerk, indem jede Community zu jeweils einem Knoten zusammengefasst wird. Die Knoten innerhalb der roten Community werden zum Knoten 0, die Knoten innerhalb der blauen Community werden zum Knoten 1 und die Knoten der grünen Community werden zum Knoten 2 verschmolzen. Die Kantengewichte zwischen den Knoten im neuen Netzwerk entspricht der Summe der alten Gewichte zwischen den jeweiligen Communities. Die Kantengewichte der Schleifen im neuen Netzwerk werden durch Verdopplung der Summe der Kantengewichte innerhalb der jeweiligen alten Community berechnet. Level 0 setzt sich aus Phase 1 und Phase 2 zusammen.

5 Map Equation

Die Optimierung der sogenannten Map Equation ist ein Verfahren zur Community Detection, welches auf der Minimierung der Codewortlänge eines Random Walk im Netzwerk basiert. Das heißt hier wird der Vorteil der Dualität zwischen dem Auffinden der Gemeinschaftsstrukturen in einem Netzwerk in Form von Communities und der Minimierung der Codewortlänge eines Random Walk ausgenutzt.
Zunächst wird, um das gesamte Vorgehen des Verfahrens gut zu verstehen, die Entropie als Informationsmaß definiert und die Huffman Codierung als eine Entropiecodierung in Kapitel 5.1 erläutert. Jeder Knoten eines Random Walk im Netzwerk wird dann in Kapitel 5.2.1 durch Anwendung der Huffman Codierung codiert. Hierbei ist es effizienter den Random Walk zunächst in Regionen von Knoten aufzuteilen. Bleibt der Random Walk eine längere Zeit in einer Region von Knoten, so erhält diese Region ein eigenes Codebuch. Für jede Region wird eine Codierung von Knoten gespeichert. Dieses Vorgehen wird in Kapitel 5.2.2 erklärt. Die Map-Equation-Funktion, welche die Codewortlänge des Random Walk beschreibt, wird dann minimiert, um eine gute Wahl dieser Regionen zu gewährleisten. Diese Funktion wird in Kapitel 5.3 hergeleitet und schließlich im Infomap-Algorithmus minimiert (vgl. Kapitel 5.4).

5.1 Entropie

Um die Entropiecodierung zu verstehen und darauf aufbauend die Huffman Codierung einzuführen, ist es erst einmal notwendig mit den Begriffen des Informationsgehalts und der Entropie, vertraut zu werden.

5.1.1 Informationsgehalt und Entropie

Der Begriff Informationsgehalt wurde erstmalig in der Informationstheorie von dem Mathematiker Claude E. Shannon verwendet. Die folgende Darstellung lehnt sich an [8] an. Sei $X = (S(n))$ eine Folge von Zeichen aus einer Menge $\{s_i | 1 \leq i \leq k\}$, bei der $p_i = p(s_i)$ die Auftrittswahrscheinlichkeit des Zeichens s_i ist. Der Informationsgehalt des Symbols s_i ist dann in Abhängigkeit von p_i definiert als:

$$I(s_i) = log_2\left(\frac{1}{p_i}\right) = -log_2(p_i)\ [bit].$$

Die Einheit des Informationsgehalt ist 1 Bit. Je kleiner die Auftrittswahrscheinlichkeit eines Symbols ist, desto höher ist sein Informationsgehalt. Wenn das Symbol hingegen oft auftritt, so ist der Informationsgehalt gering. Zur Veranschaulichung der Berechnung

des Informationsgehalts betrachten wir im folgenden Beispiel die Zeichenfolge „Hippie“. Für die jeweiligen Zeichen s_i mit den relativen Wahrscheinlichkeiten p_i lässt sich der Informationsgehalt $I(s_i)$ wie in Tabelle 2 dargestellt berechnen:

$\boldsymbol{s_i}$	h	i	p	e
Häufigkeit	1	2	2	1
$\boldsymbol{p_i}$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\boldsymbol{I(s_i)}$	2.58	1.58	1.58	2.58

Tabelle 2: Berechneter Informationsgehalt der Buchstaben im Wort „Hippie“

Die Entropie $H(X)$ der Zeichenfolge X berechnet den mittleren Informationsgehalts einer statistisch unabhängigen Folge von Symbolen:

$$H(X) = \sum_{i=1}^{K} p_i \; I(s_i) = -\sum_{i=1}^{K} p_i \; log_2(p_i) \; [bit/Symbol \,].$$

Der berechnete Entropiewert liegt nach Definition, im Fall K verschiedener Symbole, im Wertebereich

$$0 \leq H(X) \leq log_2(K). \tag{27}$$

Im „Hippie“ Beispiel berechnet sich die Entropie $H(X)$ der Zeichenfolge
$X = (\text{„H“}, \text{„i“}, \text{„p“}, \text{„p“}, \text{„i“}, \text{„e“})$ als Erwartungswert des Informationsgehalts folgendermaßen:

$$H(X) = \frac{1}{6} \cdot 2.58 + \frac{1}{3} \cdot 1.58 + \frac{1}{3} \cdot 1.58 + \frac{1}{6} \cdot 2.58 = 1.91.$$

Der Wert der Entropie ist am größten, wenn alle Wahrscheinlichkeiten der Zeichen gleich sind.

5.1.2 Huffman Codierung als Entropiecodierung

In der Entropiecodierung werden Symbole bijektiv auf Bitfolgen (Codewörter), das heißt einer Aneinanderreihung von Nullen und Einsen, abgebildet. Ziel ist es hierbei, den Speicheraufwand an die Entropie $H(X)$ anzunähern und dabei die Codierungsredundanz durch Ausnutzung der Symbolverteilung zu minimieren ([8], S.31). Den Zeichen sollen also nur so viele Bits zugeordnet werden, wie aufgrund des Informationsgehalts des Zeichens erforderlich ist ([8], S.32). Häufige Zeichen werden mit kurzen Codewörtern und seltene Zeichen werden ggf. mit längeren Codewörtern belegt, um den mittleren Aufwand klein zu halten.

Angenommen l_i sei die Länge eines Codeworts c_i, das dem Symbol s_i mit einer Auftrittswahrscheinlichkeit p_i zugeordnet wird. Dann lässt sich die mittlere Codewortlänge $\bar{l}$ als Symbolfolge von K Zeichen beschreiben als:

$$\bar{l} = \sum_{i=1}^{K} p_i \cdot l_i \, [bit/Symbol \,].$$

Die durchschnittliche Codewortlänge $\bar{l}$ gibt die Anzahl der bits pro Symbol an. Nun wird die kleinste Anzahl an Bits erreicht, wenn der Code den kleinsten Wert für $\bar{l}$ ausgibt. Also suchen wir im nächsten Schritt eine untere Grenze für die mittlere Codewortlänge $\bar{l}$. Auch hier liefert Shannon mit der folgenden Abschätzung eine Lösung für die Problemstellung:

$$\bar{l} \geq H(X). \tag{28}$$

Zur Veranschaulichung betrachten wir erneut das „Hippie“ Beispiel. Den vier unterschiedlichen Symbolen „h“,„i“,„p“, „e“ werden Codewörter der Länge $l_i = log_2(4) = 2$ (vgl. Abschätzung 27) zugeordnet. Die mittlere Codewortlänge berechnet sich also wie folgt:

$$\bar{l} = \frac{1}{6} \cdot 2 + \frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 2 + \frac{1}{6} \cdot 2 = 2 \, [bit/Symbol \,].$$

Offensichtlich gilt hier $2 = \bar{l} > H(X) = 1.91$.
Ein Beispielalgorithmus für die Entropiecodierung ist die Huffman Codierung, welche in unserem Verfahren Anwendung findet und somit in diesem Unterkapitel genau erläutern wird. Als ein Beispiel der Entropiecodierung möchten wir nun die Huffman Codierung mit einer variablen Codewortlänge einführen. Dies ist eine präfixfreie Codierung. Das heißt, kein Codewort kommt zu Beginn eines anderen Codeworts vor. Wie bereits erwähnt wurde, werden häufige Zeichen mit kurzen Codewörtern und seltene Zeichen ggf. mit längeren Codewörtern belegt ([7]). Aus den zu codierenden Zeichen wird eine spezielle Datenstruktur, nämlich eine Baumstruktur, aufgebaut. Der hier betrachtete Codebaum ist eine übliche Darstellungsform für Codes. Er ist ein Binärbaum, das heißt es gibt immer eine Wurzel und jeder Knoten hat 0 oder 2 Kinder. Knoten, die keine Kinder haben, heißen Blattknoten. Um eine Huffman Codierung zu finden, durchläuft man die folgenden Schritte ([8], S.36):

1. Berechne relative Wahrscheinlichkeit für jedes Zeichen und füge sie als Blätter in den Codebaum ein.

2. Wähle die zwei Zeichen mit den kleinsten Wahrscheinlichkeiten aus. Wenn hier mehrere mit gleicher Wahrscheinlichkeit vorliegen, so muss man eine Wahl vereinbaren.

Füge einen Vaterknoten hinzu, in dem die Summe der Wahrscheinlichkeiten als Wert eingetragen wird.

3. Beschrifte Pfad von linkem Kind zum Vater mit 0 und Pfad vom rechten Kind zum Vater mit 1.

4. Wenn Wurzel mit Wahrscheinlichkeit 1 erreicht ist, Stopp; sonst gehe zu 2. Schritt.

Die Beschriftung der Zweige von der Wurzel bis zum Blatt gibt das jeweilige Codewort des Blatts zurück. Betrachten wir den Ausdruck „Hippie“, so lässt sich der Codebaum in Abbildung 5 für die jeweiligen Zeichen erzeugen:

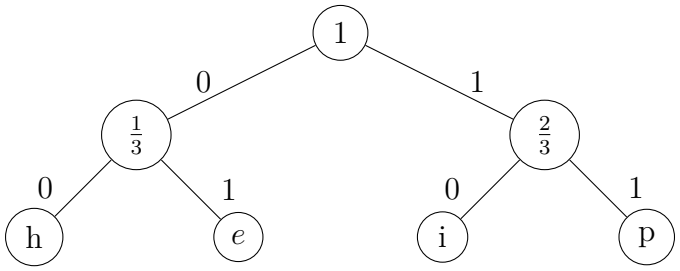

Abbildung 5: Codebaum für Beispielwort „Hippie“

Die im Codebaum abzulesende Codierung c_i ist für jedes Symbol s_i mit Index i in der folgenden Tabelle 3 zusammengefasst:

$\boldsymbol{i}$	1	2	3	4
$\boldsymbol{s_i}$	h	i	p	e
$\boldsymbol{p_i}$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\boldsymbol{c_i}$	00	10	11	01
$\boldsymbol{l_i}$	2	2	2	2

Tabelle 3: Codewörter c_i mittels Huffman Codierung für Beispielwort „Hippie“

5.2 Codierung eines Random Walk

Wie bereits in Kapitel 3.4 eingeführt, ist ein Random Walk im Graph eine zufällige Folge benachbarter Knoten: Es ist sowohl ein Graph als auch ein Startknoten gegeben. Dann wird vom Knoten ausgehend ein zufälliger Nachbar mit einer Wahrscheinlichkeit proportional zum entsprechenden Kantengewicht besucht; dann wird wieder von diesem Knoten ausgehend ein zufälliger Nachbar besucht usw.

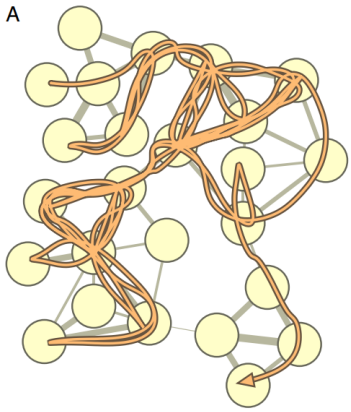

Abbildung 6: Random Walk im Beispielnetzwerk [10]

In Abbildung 6 ist ein Random Walk in 71 Schritten durch die orangefarbige Linie in einem Beispielnetzwerk mit 25 Knoten abgebildet. Dieser Random Walk kann beschrieben werden, indem man die Folge der Codewörter der besuchten Knoten angibt. Um den Random Walk zu minimieren schätzen wir den Random Walk nach oben ab. Dafür wird erneut die Ungleichung 27 herangezogen. Sie liefert eine obere Abschätzung mit der Annahme, dass jedes Codewort die gleiche Länge aufweist. Jedes Codewort benötigt $log_2(25) = 5$ Bits. Für die 71 Schritte, des Random Walks, sind dies also

$$71 \cdot 5 = 355 \; bit. \tag{29}$$

5.2.1 Einfache Codierung

In Abbildung 7 wurde jedem Knoten ein eindeutiger Code mittels Huffman Codierung zugeteilt. Unter der Abbildung lässt sich der Random Walk als die durch eine Bitfolge dargestellte Folge der durchlaufenen Knoten ablesen. Die Codierung weist 314 *bit* auf, und ist somit günstiger als die in Gleichung 29 berechnete Abschätzung.

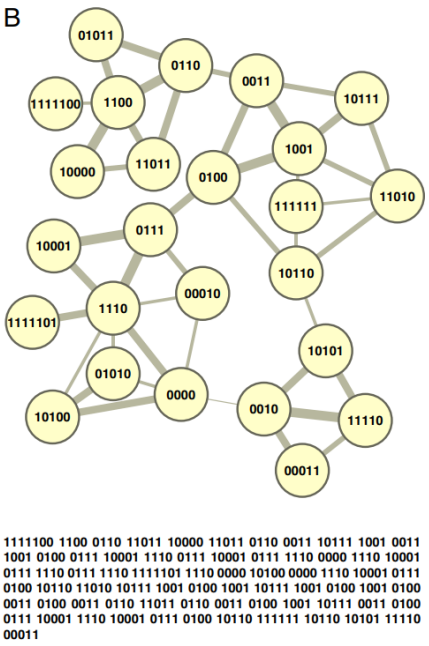

Abbildung 7: Random Walk im Beispielnetzwerk mit Huffman Codierung [10]

5.2.2 Codierung mithilfe Codebücher

Wenn der Random Walk dazu tendiert, längere Zeit in gewissen Regionen des Graphen zu verweilen und nur seltener zwischen solchen Regionen des Graphen zu wechseln, dann kann man durch die Einführung verschiedener Codebücher – wie weiter unten erläutert – eine effiziente Codierung finden. Die Regionen längeren Aufenthalts sind die gesuchten Communities.

Das Codierungsvorgehen ist dann nach ([10], S.17) wie folgt: Die Knoten jeder Community werden mit je einem eigenen Community-Codebuch codiert. Zusätzlich gibt es ein Index-Codebuch, in dem alle Communities verzeichnet sind. Zunächst gibt ein Eintrag aus dem Index-Codebuch an, welches Community-Codebuch als nächstes verwendet wird. Und dann gibt das Community-Codebuch die Reihenfolge der Knoten innerhalb dieser Community an, bis ein Exit-Zeichen das Verlassen der Community markiert. Dann wird wieder das Index-Codebuch herangezogen, um zu bestimmen, welche Community als nächstes durchlaufen wird usw.

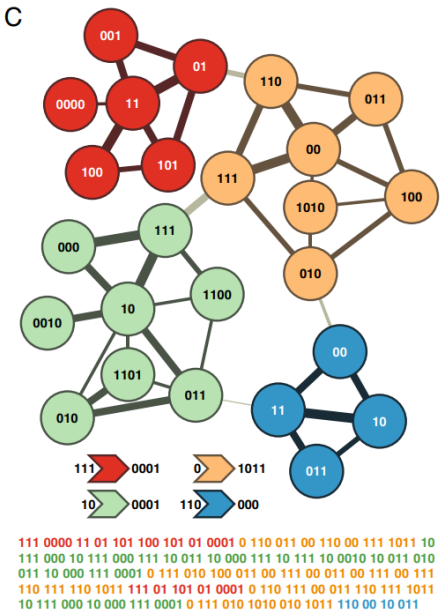

Abbildung 8: Communitiy-Struktur durch Codierung mithilfe Codebücher [10]

Wenn man für den Random Walk in Abbildung 6 unterschiedliche Codebücher verwendet, erhält man eine kürzere Codierung. In Abbildung 8 werden vier unterschiedliche Communitiy-Codebücher benutzt und es gibt ein Index-Codebuch, welches aus Einträgen besteht, die den Übergang in eine der anderen Communities beschreiben. Die Codewortlänge beträgt hierbei nur noch 243 *bit*. Dies ist weniger Speicherbedarf als für den Random Walk in Abbildung 7 bestimmt wurde. Wichtig ist hierbei, dass bei einem Übergang eines Pfads in eine neue Community, das Community-Codebuch auf die richtige Community gestellt wird. Ansonsten ist die Codierung sinnlos. Es ist zu beachten, dass die Knoten erst in Kombination mit der jeweiligen Community eindeutig werden. Die gleiche Codierung kann also in mehreren Communities auftreten, so wie beispielsweise der Code '111' in der orangenen und in der grünen Community auftritt.
Somit existiert eine Dualität zwischen der Minimierung der Codewortlänge des Random Walk, und dem Auffinden von Gemeinschaftsstrukturen im Netzwerk.

5.3 Herleitung Map Equation

Um eine optimale Partition des Netzwerks für die oben beschriebene Fragestellung zu finden, wird die Codewortlänge für gegebene Partitionen des Netzwerks berechnet und diejenige mit der kürzeste Länge ausgewählt. Das heißt wir leiten eine Funktion, welche die minimale durchschnittliche Codewortlänge bezüglich der Codebücher beschreibt, her, die so genannte Map Equation, und dann wird diese minimiert, um die Partition mit der kürzesten Codewortlänge zu finden.

Der Random Walk im Graphen ist ein Beispiel einer Markow-Kette. Das ist ein stochastischer Prozess, in dem Prognosen ohne Kenntnis der Vorgeschichte genauso gut möglich sind wie mit Kenntnis. Das heißt in unserem Fall, dass der nächste Knoten im Random Walk nicht von dem bisher durchlaufendem Random Walk abhängt, sondern nur von der Übergangswahrscheinlichkeit des aktuellen Knotens zu einem seiner Nachbarn. Die Übergangswahrscheinlichkeit $p_{v\to u}$ von einem Knoten v zu seinem Nachbarn u wird als das normierte Kantengewicht berechnet:

$$p_{v\to u} = \frac{w_{v\to u}}{\sum_t w_{v\to t}}. \tag{30}$$

Sei eine Partition $\mathcal{C} = \{C_1, ...C_m\}$ des Netzwerks von n Knoten in m Communities gegeben. Gesucht ist nun die minimale Codewortlänge über diese Partition. Die Abschätzung 28 von Shannon liefert uns, dass die durchschnittliche Anzahl von Bits, um einen einzelnen Schritt im Random Walk zu beschreiben, nach unten beschränkt ist durch die Entropie $H(P)$, wobei P hier die Verteilung der Besuchshäufigkeit der Knoten des Random Walk im Netzwerk darstellt [11]. Diese untere Grenze, die wir hier Map Equation Funktion $L(\mathcal{C})$ nennen werden, setzt sich aus Anteilen von einem Index-Codebuch und m Community-Codebüchern zusammen.

$$L(\mathcal{C}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p_{i\circlearrowright} H(\mathcal{P}_i). \tag{31}$$

Der rote Term beschreibt hier die Läufe des Random Walk zwischen den Communities, das heißt hier wird das Index-Codebuch für die Berechnung herangezogen. Der blaue Term beschreibt hingegen die Läufe des Random Walk innerhalb der Community, das heißt hier werden die m Community-Codebücher bei der Berechnung benutzt. Für die Erklärung von 31 werden im Folgenden einige Formeln eingeführt. Sei

$$q_{i\curvearrowright} = \sum_{v\in C_i, u\notin C_i} p_{v\to u} \tag{32}$$

die Übergangsrate des Random Walk, der aus einer Community i hinaus und in eine von i verschiedene Community hinein geht. Diese wird mithilfe der oben definierten Übergangswahrscheinlichkeit (vgl. Formel 30) berechnet. Über alle Communities summiert, ist dies die Wahrscheinlichkeit, dass der Random Walk die Community wechselt, dass also das Indexcodebuch benutzt wird:

$$q_{\curvearrowright} = \sum_{i=1}^{m} q_{i\curvearrowright}. \tag{33}$$

$H(\mathcal{Q})$ als Entropie der Benennung der Communities ist die untere Grenze der durch-

schnittlichen Länge eines Codeworts, um eine Community zu benennen. $\mathcal{Q}$ stellt hierbei die normalisierte Häufigkeitsverteilung der Indexcodebuch-Benutzung dar. $H(\mathcal{Q})$ kann wie folgt berechnet werden:

$$\begin{aligned} H(\mathcal{Q}) &= -\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{\sum_{j=1}^{m} q_{j\curvearrowright}} log\left(\frac{q_{i\curvearrowright}}{\sum_{j=1}^{m} q_{j\curvearrowright}}\right) \\ &= -\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{q_{\curvearrowright}} log\left(\frac{q_{i\curvearrowright}}{q_{\curvearrowright}}\right). \end{aligned} \tag{34}$$

Der erste rote Term der Map Equation $q_{\curvearrowright}H(\mathcal{Q})$ gibt die schrittweise durchschnittliche Länge der Läufe zwischen den Communities innerhalb eines Random Walks an.
Berechnen wir nun die Entropie der Läufe innerhalb der Communities: Sei p_v die Wahrscheinlichkeit, dass der Knoten v im Random Walk besucht wird. Die Wahrscheinlichkeit, dass Community-Codebuch i benutzt wird, ist

$$p_{i\circlearrowright} = \sum_{v \in C_i} p_v + q_{i\curvearrowright}; \tag{35}$$

sie setzt sich zusammen aus $\sum_{v \in C_i} p_v$, der Wahrscheinlichkeit, dass der Random Walk die Knoten aus der Community C_i besucht, und $q_{i\curvearrowright}$, der Wahrscheinlichkeit, dass der Random Walk aus der Community C_i heraus geht (Exit-Zeichen). Die untere Grenze für die Codewortlänge im Community-Codebuch von C_i ist

$$H(\mathcal{P}_i) = -\sum_{v \in C_i} \frac{p_v}{p_{i\circlearrowright}} log\left(\frac{p_v}{p_{i\circlearrowright}}\right) - \frac{q_{i\curvearrowright}}{p_{i\circlearrowright}} log\left(\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}}\right). \tag{36}$$

Der erste Term dieser Formel betrifft die Codierung des Verlaufs innerhalb der Community C_i und der zweite Teil die Codierung des Exits aus C_i.
Kombinieren wir die Formeln 35 und 36 und laufen über alle m Communities im Netzwerk, erhalten wir den zweiten, blau markierten Teil der Map Equation, $\sum_{i=1}^{m} p_{i\circlearrowright} H(\mathcal{P}_i)$, für die schrittweise durchschnittliche Codewortlänge eines Laufs des Random Walk innerhalb der Communities.
Der rote und blaue Term der Map Equation 31 lassen sich weiter zusammenfassen:

$$\begin{aligned} q_{\curvearrowright}H(\mathcal{Q}) &= q_{\curvearrowright}\left(-\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{q_{\curvearrowright}} log\left(\frac{q_{i\curvearrowright}}{q_{\curvearrowright}}\right)\right) \\ &= -\sum_{i=1}^{m} q_{i\curvearrowright}(log(q_{i\curvearrowright}) - log(q_{\curvearrowright})) \\ &= \sum_{i=1}^{m} q_{i\curvearrowright} log(q_{\curvearrowright}) - \sum_{i=1}^{m} q_{i\curvearrowright} log(q_{i\curvearrowright}) \end{aligned} \tag{37}$$

$$
\begin{aligned}
\sum_{i=1}^{m} p_{i\circlearrowright} H(\mathcal{P}_i) &= \sum_{i=1}^{m} p_{i\circlearrowright} \cdot \left[- \sum_{v \in C_i} \frac{p_v}{p_{i\circlearrowright}} (log(p_v) - log(p_{i\circlearrowright})) - \frac{q_{i\curvearrowright}}{p_{i\circlearrowright}} (log(q_{i\curvearrowright}) - log(p_{i\circlearrowright})) \right] \\
&= - \sum_{i=1}^{m} \sum_{v \in C_i} p_v log(p_v) + \sum_{i=1}^{m} \sum_{v \in C_i} p_v log(p_{i\circlearrowright}) - \sum_{i=1}^{m} q_{i\curvearrowright} log(q_{i\curvearrowright}) + \sum_{i=1}^{m} q_{i\curvearrowright} log(p_{i\circlearrowright}) \\
&= - \sum_{v=1}^{n} p_v log(p_v) - \sum_{i=1}^{m} q_{i\curvearrowright} log(q_{i\curvearrowright}) + \sum_{i=1}^{m} (\sum_{v \in C_i} p_v + q_{i\curvearrowright}) log(p_{i\circlearrowright}). \qquad (38)
\end{aligned}
$$

Die Summe aus Term 37 und 38 liefert uns die vereinfachte Map Equation:

$$
\begin{aligned}
L(\mathcal{C}) &= \sum_{i=1}^{m} q_{i\curvearrowright} log(\sum_{i=1}^{m} q_{i\curvearrowright}) - 2 \cdot \sum_{i=1}^{m} q_{i\curvearrowright} log(q_{i\curvearrowright}) \\
&\quad - \sum_{v=1}^{n} p_v log(p_v) + \sum_{i=1}^{m} (\sum_{v \in C_i} p_v + q_{i\curvearrowright}) log(\sum_{v \in C_i} p_v + q_{i\curvearrowright}). \qquad (39)
\end{aligned}
$$

5.4 Infomap-Algorithmus

Der Infomap-Algorithmus minimiert die Map Equation 39 über alle möglichen Netzwerk-Partitionen.

Der Infomap-Algorithmus ist ein Multi-Level-Algorithmus, das heißt er ist als ein hierarchisches Verfahren in der Lage, Muster auf verschiedenen Ebenen zu erkennen. Communities in gröberen Leveln sind also größer als Communities in feineren Leveln, da immer mehr Communities auf höheren Level zusammengefasst werden. Im hierarchischen Map Equation gibt es auf jeder Hierarchieebene ein Index-Codebuch und m Community-Codebücher auf jeder Ebene.

Die Idee des Infomap-Algorithmus basiert auf der gleichen Idee wie der Louvain-Algorithmus (vgl. Kapitel 4.2) und ist im Pseudocode 2 skizziert:

Algorithm 2 Infomap-Algorithmus

Input: $G = (V, E)$

$G^i = G$; $i = 0$

repeat

  Phase 1:

  Input: Graph $G^i = (V, E)$, $V = \{x_1, ..., x_n\}$

  Output: Partition $\mathcal{C}^i = \{C_1, ..., C_m\}$ von G^i

  - Initialisierung Communities: $C_j = \{x_j\}\ \forall j \in \{1, ..., n\}$

  repeat

    - wähle zufällige Reihenfolge der Knoten

    for all $x \in V$ **do** (Abarbeitung entsprechend der zufälligen Knotenreihenfolge)

      - berechne $\Delta L(x, C_u)$ für alle $u \in N(x)$ mittels 39 wobei C_u die Community ist, die u enthält und $\Delta L(x, C_u)$ die Änderung von $L(\mathcal{C}^i)$ bei Verschieben von x aus C_x in Community C_u.

      if $\min\limits_{u \in N(x)} \Delta L(x, C_u) < 0$ **then**

        - verschiebe x in Nachbarcommunity C_v, mit $v = \arg\min\limits_{u \in N(x)} \Delta L(x, C_u)$

      end if

    end for

  until $\min\limits_{u \in N(x)} \Delta L(x, C_u) \geq 0\ \forall x \in V$

  return $\mathcal{C}^i$

  Phase 2:

  Input: Graph $G^i = (V^i, E^i)$, Community Partition $\mathcal{C}^i$

  Output: Graph $G^{i+1} = (V^{i+1}, E^{i+1})$

  - $V^{i+1} = \mathcal{C}^i$, $E^{i+1} = \emptyset$

  for all Kanten $(x, y) \in E^i$ **do**

    - füge Kante (C_x, C_y) zu E^{i+1} hinzu mit Gewicht

$$w(C_x, C_y) = \begin{cases} w(x, y), & \text{falls } C_x \neq C_y \\ 2 \cdot w(x, y), & \text{falls } C_x = C_y \end{cases}$$

    bzw. erhöhe Gewicht um den Betrag, falls Kante bereits vorhanden.

  end for

  return $G^{i+1} = (V^{i+1}, E^{i+1})$

  $i = i + 1$

until keine Änderung der initialen Communities in Phase 1

Zunächst werden die Communities analog wie im Louvain-Algorithmus initialisiert (vgl. Zeile 5). In Phase 2 jeder Iteration wird eine zufällige Reihenfolge der Knoten gewählt und für diese dann gemäß der gewählten Reihenfolge die Abnahme der Map Equation für die Verschiebung des Knotens v in eine Nachbarcommunity C_u berechnet (vgl. Zeile 8, 9). Diese Veränderung der Map Equation notieren wir hier analog wie beim Louvain mit ΔL. Der Knoten wird dann in genau die Nachbarcommunity verschoben, die den größten Verlust der Map-Equation-Formel aufweist (vgl. Zeile 10-12). Im Algorithmus wird der Schritt dann solange wiederholt, bis keine Verschiebung eines Knotens in eine Nachbarcommunity die Zielfunktion weiter reduziert (vgl. Zeile 14). Es wird ein neuer Graph erstellt, indem jede Community zu einem Knoten verschmolzen wird, und die Kanten zwischen den Communities als eine neue Kante zusammengefasst werden (vgl. Phase 2).

6 Idee des hierarchical Stochastic Block Models (hSBM)

Neben den auf Modularitätsoptimierung und Random Walk basierten Methoden der Community Detection gibt es unter dem Namen Stochastic Block Model (SBM) einen wahrscheinlichkeitstheoretischen argumentierten Zugang zum Auffinden von Zusammengehörigkeitsstrukturen in Netzwerken. Dieser Zugang steht den generativen probabilistischen Topic-Modeling-Verfahren wie LDA sehr nahe. In [19] wird dies näher untersucht und dazu benutzt, eine hierarchische Verallgemeinerung von SBM auf das Topic Modeling anzuwenden. In dieser Masterarbeit möchten wir die Idee des sogenannten hierarchical Stochastic Block Model (hSBM) aufgreifen und für die genaue Herleitung der Formeln auf die Literatur [20] verweisen.

6.1 Generatives Modell

Zunächst möchten wir mit der Idee eines generativen Modells beginnen: Ein generatives Modell kann man sich hier so vorstellen, dass wir den Graphen, dessen Community-Struktur wir herausfinden möchten, als ein Ergebnis eines fiktiven Zufallsprozesses ansehen. Wir modellieren die Wahrscheinlichkeit $P(A|\theta)$ für eine Menge von Modellparametern θ und die Adjazenzmatrix A von G. $P(A|\theta)$ drückt die Wahrscheinlichkeit für das Entstehen eines Graphen G mit Adjazenzmatrix A, gegeben der Modellparameter θ, aus. Umgekehrt kann man für einen gegebenen Graphen bestimmen, für welche Modellparameter-Werte θ der Graph am wahrscheinlichsten ist. Dafür wird die Likelihood-Funktion $\theta \to P(A|\theta)$ maximiert. Der hier beschriebene Schluss von Graph auf Modellparameter ist der Schritt der sogenannten Inferenz.

6.2 Standard Stochastic Block Model

Ziel der Community Detection ist es, eine Einteilung der Knoten in Communities zu erhalten, sodass die Knoten innerhalb jeder Community häufig miteinander verbunden sind und es nur wenige Verbindungen zwischen Knoten unterschiedlicher Communities gibt. Das Problem wird in diesem Ansatz mit einem generativen Modell, dem Standard Stochastic Block Model (SBM), gelöst. Das SBM konstruiert einen Graphen mit solch einer Community-Struktur. Die Community-Struktur entspricht auf der Ebene der Adjazenzmatrix einer Blockstruktur.
Das Blockmodell besteht aus B Blöcken, wobei jeder Knoten genau einem Block zugeordnet wird. Es lässt sich also die Partition $\boldsymbol{b} = \{b_x\}$ von n Knoten in B Blöcke definieren, wobei $b_x \in [1, B]$ die Blockzugehörigkeit des Knotens x ist. Die Anzahl der Kanten

zwischen den Blöcken lässt sich aus der Adjazenzmatrix ableiten und in einer $B \times B$ Blockmatrix $\boldsymbol{e} = \{e_{r,s}\}$ zusammenfassen. Die Einträge lassen sich wie folgt definieren:

$$e_{r,s} = \begin{cases} \text{Anzahl der Kanten zwischen Block r und s, falls } r \neq s. \\ 2 \times \text{Anzahl der Kanten innerhalb des Blockes r, falls } r = s. \end{cases} \tag{40}$$

Die Kanten werden dann entsprechend dieser Matrix zufällig zwischen und innerhalb der Blöcke verteilt. Veranschaulichen wir uns die Blockmatrix $\boldsymbol{e}$ und die Idee des Blockmodells an einem Beispiel:
Seien die Knoten in einem Beispielgraph G in Abbildung 9 in zwei Blöcke eingeteilt. Die Knoten des ersten Blocks sind in grün dargestellt, und die Knoten des zweiten Blocks in rot.

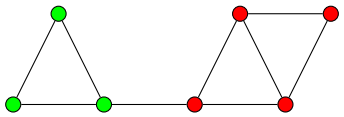

Abbildung 9: Graph G mit Community-Struktur

Die Adjazenzmatrix zu diesem Graphen ist wie folgt definiert:

$$A = \left(\begin{array}{ccc|cccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right).$$

Und mithilfe dieser lässt sich die Blockmatrix $\boldsymbol{e}$ bestimmen:

$$\boldsymbol{e} = \begin{pmatrix} 6 & 1 \\ 1 & 10 \end{pmatrix}. \tag{41}$$

Die soeben definierte Partition $\boldsymbol{b} = \{b_x\}$, wobei $\{b_x\}$ die Blockzugehörigkeit des Knotens x darstellt, ist für den Beispielgraphen G in der folgenden Abbildung 10 dargestellt:

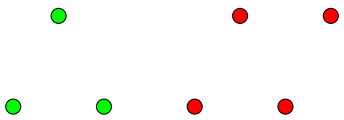

Abbildung 10: Partition $\boldsymbol{b}$ von G

Fügen wir dem Standard Stochastic Block Model einen dritten Parametersatz hinzu, die Gradverteilung $\boldsymbol{k} = \{k_x\}$, wobei k_x den Grad des Knotens x angibt, so erhalten wir das in [22] erläuterte Degree Corrected SBM. Das generierte Netzwerk G hängt hier nicht nur von den Parametern $\boldsymbol{b}$ und $\boldsymbol{e}$, sondern auch von der Gradverteilung $\boldsymbol{k}$, ab. Die Gradverteilung $\boldsymbol{k}$ der Knoten in G sind in der Abbildung 11 dargestellt.

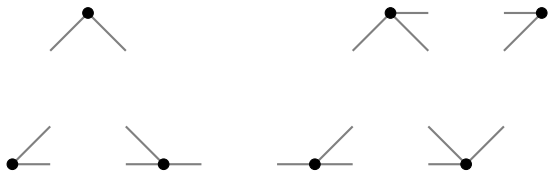

Abbildung 11: Gradverteilung $\boldsymbol{k}$ von G

Eine Möglichkeit um die Community-Struktur des Graphen aufzudecken wäre es nun $P(A|\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})$ zu bestimmen. Hierfür kann die Wahrscheinlichkeit für das Auftreten einer Adjazenzmatrix als proportional zu der Anzahl der verschiedenen Realisierungsmöglichkeiten bei den jeweiligen Modellparameter angesetzt werden. Danach kann diese Wahrscheinlichkeit $P(A|\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})$ mittels Likelihood-Funktion maximiert werden. In diesem parametrisierten Ansatz, wird allerdings verlangt, dass wir mindestens die Anzahl der Blöcke B, mit vorgeben. Es wäre hingegen wünschenswert, wenn, wie in einem nicht parametrisierten Ansatz, die Anzahl der Blöcke sowie alle anderen Parameter aus den Daten selber bestimmt werden können. Darunter versteht man ein Modell, indem es nach der Inferenz keine freien Modellparameter mehr gibt.

6.3 Hierarchical Stochastic Block Model

Auf der Grundlage von Peixoto in der Literatur [20] wird nun das Vorgehen beschrieben, um zu einem nicht parametrisierten Modell, dem hierarchical Stochastic Block Model

(hSBM) zu gelangen. Zunächst wird dafür die a-Posteriori-Wahrscheinlichkeit mit gegebener Adjazenzmatrix A nach Bayes bestimmt

$$P(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}|A) \propto P(A|\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})P(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}), \tag{42}$$

wobei die enthaltene a-Priori-Wahrscheinlichkeit geschachtelt geschrieben werden kann als $P(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}) = P(\boldsymbol{k}|\boldsymbol{e}, \boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b})P(\boldsymbol{b})$. Nun kann für die Wahrscheinlichkeit $P(\boldsymbol{e}|\boldsymbol{b})$ eine iterative Formulierung gewählt werden. Denn das hSBM geht iterativ vor und versucht die Blockstruktur zunächst auf feineren Leveln und dann auf gröberen zu identifizieren. Starten wir bei Level 1, so kann für eine Blockstruktur $\boldsymbol{e}_1$ und Knotenpartition $\boldsymbol{b}_1$ eine Adjazenzmatrix bestimmt werden. Die Adjazenzmatrix enthält eine Community-Struktur. Erneut kann dann für die Adjazenzmatrix mit Blockstruktur $\boldsymbol{e}_2$ und Partition $\boldsymbol{b}_2$ eine Adjazenzmatrix mit einer enthaltenden Community-Struktur bestimmt werden. Dies wird iterativ wiederholt, bis für die Blockanzahl einer Ebene L gilt $B_L = 1$, das heißt es ist nur noch ein Block auf dieser Ebene vorhanden. Anstelle der Berechnung von $P(\boldsymbol{e}|\boldsymbol{b})P(\boldsymbol{b})$ wird nun auf allen Ebenen die Wahrscheinlichkeit $\prod_{l=1}^{L} P(\boldsymbol{e}_l|\boldsymbol{e}_{l+1}, \boldsymbol{b}_l)P(\boldsymbol{b}_l)$ ermittelt.

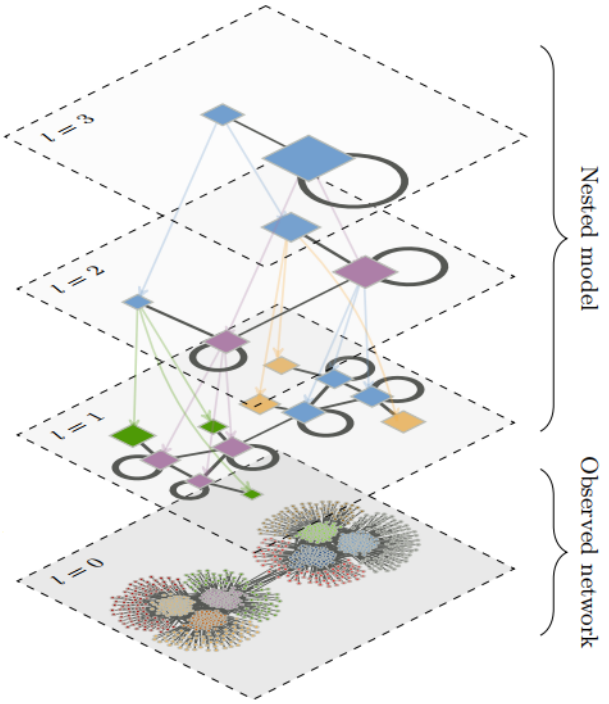

Abbildung 12: Beispiel Erzeugung unterschiedliche Level hSBM [21]

Veranschaulichen wir uns das beschriebene Vorgehen nochmal in einem Beispiel in Abbildung 12: Sei $\boldsymbol{b_l}$ die Partition von n_l Knoten in B_l Blöcke in Level l. Und sei $\boldsymbol{e_l}$ die Adjazenzmatrix in Level l und es gilt $B_L = 1$, wobei L die Gesamtanzahl der Level ist.

Die Anzahl der Kanten m bleibt auf allen Leveln gleich, wohingegen die Anzahl der Knoten n_l für größer werdende Level kleiner wird. Zunächst wird ein Netzwerk beobachtet mit n_0 Knoten und m Kanten. Die Anzahl der gefundenen Blöcke in diesem Netzwerk ist B_0. Das Netzwerk hierzu ist in Level 0 abgebildet (l=0). Auf dem nächsten Level $l = 1$ werden dann die Knoten gleich der letzten gefundenen Partition B_0 gesetzt und jede Kante innerhalb einer Community in Level 0 wird dann zur Schleifenkante des neuen Knotens in Level 1 (bzw. der Community in Level 0). Das heißt es gilt für die Anzahl der Knoten im jeweiligen Level $n_l = B_{l-1}$. Dies wird rekursiv solange wiederholt bis auf einem Level $B_l = 1$ gilt. Dies ist in der Abbildung in $B_4 = 1$ der Fall. Die letzte gefundene Ebene mit gröbster Community-Partition ist hier also $l = 3$.
Wichtig ist an dieser Stelle noch einmal zu betonen, dass gerade das Vorteilhafte in diesem Ansatz ist, keine Parameter wie Anzahl der Blöcke vorgeben zu müssen, sondern, dass die Blockhierarchie $\boldsymbol{b}_l$ im Modell durch Inferenz ermittelt werden kann. Für das weitere Zusammenfassen der Ausdrücke möchten wir an dieser Stelle auf die Gleichung (45) in Peixoto ([20], S.9) verweisen. Der hier soeben beschriebene hierarchische Ansatz kann auf den unterschiedlichen Leveln mit einem Greedy-Algorithmus, welcher das Monte Carlo Importance Sampling (MCIS) Verfahren nutzt, umgesetzt werden ([20], S.6 f.).
Das dargestellte allgemeine Verfahren von Peixoto wird von Gerlach et al. in [19] für ein bipartites Wort-Dokumenten-Netzwerk abgewandelt. Dies geschieht, indem die a-Priori-Wahrscheinlichkeit für die Blöcke disjunkte Partitionen von Wörtern und Dokumenten vorgibt. Somit darf jeder Block entweder nur Dokumente oder nur Wörter enthalten.

7 Experimentelle Ergebnisse und Vergleich der Verfahren

In diesem anwendungsorientierten Kapitel werden die drei Verfahren auf zwei Datensätze, nämlich die Kleinen Anfragen an die Bundesregierung und die Publikationen der DLR-Publikationsdatenbank Elib, angewendet.
Zunächst werden Kapitel 7.1 bis 7.3 dazu dienen, die Grundlagen dafür zu schaffen. Der erste Teil (vgl. Kapitel 7.1) widmet sich der Vorstellung der ausgewählten Datenquellen und den daraus aufbereiteten Korpora. In 7.2 wird eine Übersicht über die KA-Sachgebiete und die Elib-Forschungsgebiete gegeben. Es existiert eine Zuordnung unserer Daten auf diese Felder. Uns interessiert dabei, ob deren Klassifikation mit unseren gefunden Topics übereinstimmt. Im Fokus des Kapitels 7.3 steht die Erzeugung eines Graphen mit Gewichtsalternativen a und b aus den soeben eingeführten Korpora. Bestandteil des Kapitels wird es sein, die wichtigsten Eigenschaften der Graphen wiederzugeben. Zur Ausführung des Louvain- und Infomap-Verfahrens wird dann genau dieser Graph als Input benötigt. Die Ausgabe zweier gerankter Wortlisten für die ausgewählten Korpora ist Inhalt des Kapitels 7.4. Die Wortliste wird zum einen nach dem Bayes Ranking und zum anderen nach dem Common Word Ranking sortiert, welche in 2.3 erläutert wurden. Die Ausgabe von gerankten Wortlisten innerhalb der Topics ist von wichtiger Bedeutung, da es sowohl die Interpretation der Topics als auch alle weiteren Auswertungsschritte erleichtert.
Dann folgt der Hauptteil der Arbeit: In den Kapiteln 7.5 und 7.6 werden die Ergebnisse der drei Community-Detection-Verfahren präsentiert. Hierbei wurden verschiedene Python Bibliotheken und Funktionen verwendet, welche in den zugehörigen Unterkapiteln vorgestellt werden.
Abschließend erfolgt in Kapitel 7.7 ein Vergleich der angewendeten Verfahren untereinander und mit den Ergebnissen des LDAs. Hierbei werden Methoden wie die Modularitätsberechnung, das Word Embedding und die Einschätzung von Sachexperten berücksichtigt.

7.1 Aufbereitete Korpora

Als eine Datenquelle aus dem Politik-Bereich haben wir die Kleinen Anfragen im deutschen Bundestag ausgewählt. Mit den Kleinen Anfragen kann eine Gruppe von Abgeordneten im Parlament der jeweiligen Regierung Fragen stellen, die von dieser zeitnah beantwortet und veröffentlicht werden müssen [30]. Die Antworten auf die Kleinen Anfragen liefern einen aussagekräftigen Korpus mit interessanten Details über die aktuelle politische Diskussion. Aus deren Analyse erhoffen wir uns, aktuelle Themen aufzudecken, welche sowohl die Politik als auch die Gesellschaft betreffen.
Als einen zweiten Datensatz haben wir aus dem Wissenschaft-Bereich englische Publi-

kationen des Deutschen Zentrums für Luft- und Raumfahrt ausgewählt. Diese sind auf dem Open Access DLR-Publikationsserver Elib erfasst und archiviert. Aus den Topic-Ergebnissen erhoffen wir uns Aufschluss über Forschungstrends am DLR.
Im folgenden werden vier unterschiedliche Versionen der zwei Korpora Kleine Anfragen und Elib-Publikationsdaten vorgestellt:

- KA 2017: Der Korpus enthält alle Dokumente der Kleine Anfragen an die Bundesregierung im Jahr 2017. Die Anzahl der Dokumente beträgt 920. Die Textlänge pro Dokument wurde im letzten Schritt der Datenaufbereitung auf 12.5 % der eigentlichen Länge reduziert. Die Motivation dabei ist, das Netzwerk zu verkleinern, um die technischen Vorteile in den nächsten Schritten auszunutzen.

- Elib condensed: Die Version des Elib-Korpus enthält alle Dokumente aus Elib seit dem Jahr 2015. Die Anzahl der Dokumente beträgt 17651. In dem Korpus sind sowohl Überschriften als auch Abstracts vorhanden. Falls ein Dokument eine Überschrift, aber keinen Abstract aufweist, so wird die Überschrift dennoch mit aufgenommen. Hierbei wurde die Textlänge pro Dokument im letzten Schritt der Datenaufbereitung auf 80 % der eigentlichen Länge reduziert.

- Elib condensed a2: Die Version des Elib-Korpus enthält alle Dokumente aus Elib seit dem Jahr 2015, die zwingend einen Abstract enthalten. Das sind 12528 Dokumente. Das heißt Dokumente, die nur eine Überschrift aufweisen, fallen hier raus. Die Textlänge pro Dokument wurde hier auf 50 % reduziert. Die Motivation dafür, eine andere Variante des Elib-Korpus zu betrachten, lag in dem Verdacht, dass Dokumente, die nur aus einem Titel bestehen, zu kurz sind, um sich positiv auf die Topic-Bildung auszuwirken. Die weitere Reduzierung der Textlänge im Vergleich zu Elib condensed zielt auf eine technische und inhaltliche Optimierung der Auswertung ab.

- Elib condensed a3: Diese Version wird analog zu Elib condensed a2 erzeugt, jedoch wurden hierbei die Dokumente auf nur noch 20 % der eigentlichen Länge reduziert.

In den Kapiteln 7.2 und 7.3 werden wir der Übersichtlichkeit halber nur den Korpus KA 2017 und Elib condensed a2 betrachten. Für die anderen Korpora kann der Graph zusammen mit den Graph-Eigenschaften und deren gerankte Wortlisten im Jupyter Notebook Programm „Topic Modeling - Louvain und Infomap“ ausgegeben werden.

7.2 Elib-Forschungsgebiete und KA-Sachgebiete

Bestandteil der Kapitel Experimentelle Ergebnisse der Community-Detection-Verfahren (vgl. 7.5 und 7.6) wird es sein, unsere gefundenen Topics mit den vorgegebenen Klassifikationen der Datenquelle abzugleichen. Zu diesem Zweck werden wir eine Heat Map erstellen, welche das Zusammenauftreten der Wörter in den Topics und den Sachgebieten bei den Kleinen Anfragen (KA) oder den Forschungsgebieten bei Elib abbildet. Dafür möchten wir im Folgenden zunächst kurz die vorgegebenen Klassifikationslisten erläutern.

KA-Sachgebiete

Bei den Kleinen Anfragen an die Bundesregierung gibt es eine Zuordnung der Dokumente zu 28 Sachgebieten. Es existieren 264 Dokumente mit einer eindeutigen Sachgebiet-Zuordnung. Aus diesem Grund können die von uns gefundenen KA-Topics für 264 Dokumente mit den zugeordneten Sachgebieten in einer Heat Map verglichen werden. Die Liste der KA-Sachgebiete wird in alphabetischer Reihenfolge sortiert:

KA-Sachgebiete alphabetisch sortiert
['Arbeit und Beschäftigung', 'Ausländerpolitik, Zuwanderung', 'Außenpolitik und internationale Beziehungen', 'Außenwirtschaft', 'Bildung und Erziehung', 'Bundestag', 'Energie', 'Entwicklungspolitik', 'Europapolitik und Europäische Union', 'Gesellschaftspolitik, soziale Gruppen', 'Gesundheit', 'Innere Sicherheit', 'Kultur', 'Landwirtschaft und Ernährung', 'Medien, Kommunikation und Informationstechnik', 'Neue Bundesländer/innerdeutsche Beziehungen', 'Politisches Leben, Parteien', 'Raumordnung, Bau- und Wohnungswesen', 'Recht', 'Soziale Sicherung', 'Sport, Freizeit und Tourismus', 'Staat und Verwaltung', 'Umwelt', 'Verkehr', 'Verteidigung', 'Wirtschaft', 'Wissenschaft, Forschung und Technologie', 'Öffentliche Finanzen, Steuern und Abgaben']

Elib-Forschungsgebiete

In der Elib-Publikationsdatenbank wurde jedes Dokument einem Forschungsgebiet zugeordnet. Die Liste der existierenden Forschungsgebiete ist auch hier in alphabetischer Reihenfolge sortiert, unter Nutzung der Abkürzungen „E“ für Energie, „L“ für Luftfahrt, „R“ für Raumfahrt, „V“ für Verkehr und „W“ für Weltraum:

Elib-Forschungsgebiete alphabetisch sortiert
['E - keine Zuordnung', 'E EV - Energieverfahrenstechnik', 'E MS - Management und Systemanalyse', 'E SF - Solarforschung', 'E SP - Energiespeicher', 'E SW - Solar- und Windenergie', 'E SY - Energiesystemanalyse', 'E VG - Verbrennungs- und Gasturbinentechnik', 'E VS - Verbrennungssysteme', 'L - keine Zuordnung', 'L AO - Air Traffic Management and Operation', 'L AR - Aircraft Research', 'L AR - Starrflüglerforschung', 'L ER - Engine Research', 'L RR - Rotorcraft Research', 'L VU - Luftverkehr und Umwelt', 'R - keine Zuordnung', 'R EO - Erdbeobachtung', 'R EW - Erforschung des Weltraums', 'R FR - Forschung unter Weltraumbedingungen', 'R KN - Kommunikation und Navigation', 'R RP - Raumtransport', 'R SY - Technik für Raumfahrtsysteme', 'V - keine Zuordnung', 'V BF - Bodengebundene Fahrzeuge', 'V SC Schienenverkehr', 'V ST Straßenverkehr', 'V VM - Verkehrsmanagement', 'V VS - Verkehrssystem', 'W EO - Erdbeobachtung', 'W EW - Erforschung des Weltraums', 'W KN - Kommunikation/Navigation', 'W RP - Raumtransport', 'W SY - Technik für Raumfahrtsysteme', 'keine Zuordnung']

„W“ ist ein früher verwendetes Forschungsgebiet für Raumfahrt, nämlich Weltraum. Das Kürzel „W“ wird aufgrund der wenig zugeordneten Dokumente und aufgrund von Wiederholungen zu dem Forschungsgebiet Raumfahrt in der Auswertung nicht berücksichtigt.

7.3 Erzeugung der Graphen

Als Grundlage für das weitere Vorgehen haben wir für alle ausgewählten Korpora jeweils zwei Graphen erstellt. Diese wurden mithilfe des Gewichts a und des Gewichts b, deren Motivation in Kapitel 3.2 dargestellt wurde, erzeugt.
An dieser Stelle möchten wir kurz einige Graph-Eigenschaften wie Knotenanzahl, Kantenanzahl und Gewichtsverteilung der unterschiedlichen Graphen gegenüberstellen.

7.3.1 KA 2017

In Tabelle 4 sind sowohl Knoten- als auch Kantenanzahl des jeweiligen Netzwerks (Gewicht a und Gewicht b) für die KA 2017 zusammengefasst:

	|V|	|E|
Gewicht a	77286	32617334
Gewicht b	77284	19563633

Tabelle 4: Knotenanzahl und Kantenanzahl von G - KA 2017

Hier lässt sich erkennen, dass das Netzwerk mit Gewicht a dichter ist als das Netzwerk mit Gewicht b. Betrachten wir zunächst den Graphen, welchen wir mit dem Korpus der KA 2017 und Gewicht a erstellt haben. Das heißt die Gewichte wurden um 1 erhöht, wenn die jeweiligen adjazenten Knoten (bzw. Wörter) erneut gemeinsam in einem Dokument auftreten. Die Gewichtsverteilung ist in der Tabelle 5 und in der Graphik 13 zusammengefasst.

Gewicht	1	2	3	4	5	...	62	63
Häufigkeit	30604695	1487828	490966	222856	83101	...	1	1

Tabelle 5: Gewichtsverteilung - KA 2017, Gewicht a

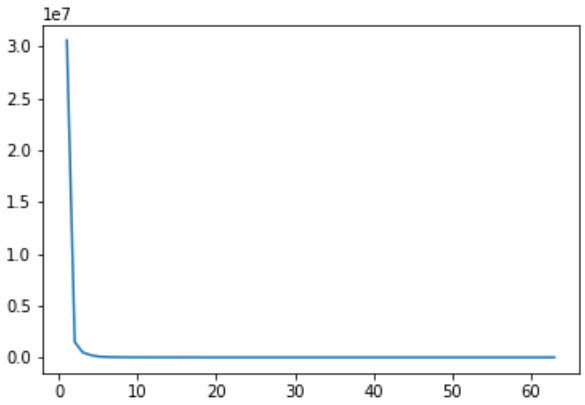

Abbildung 13: Gewichtsverteilung - KA 2017, Gewicht a

Betrachten wir nun die Gewichtsverteilung des Graphen, der aus dem Korpus KA 2017 mit Gewicht b generiert wurde.

Gewicht	1	2	3	4	5	...	18559590	22706706
Häufigkeit	4119482	2928918	1704152	956153	1031063	...	1	1

Tabelle 6: Gewichtsverteilung - KA 2017, Gewicht b

In Tabelle 6 wird leicht ersichtlich, dass die Gewichte b um einiges größer sein können als die Gewichte a. Dafür gibt es weniger Kanten mit sehr niedrigen Gewichten, wie beispielsweise für die Gewichte 1, 2 und 3.

7.3.2 Elib condensed a2

In Tabelle 7 werden Knoten- und Kantenanzahl des jeweiligen Netzwerks für den Elib condensed a2 Korpus gegenübergestellt.

	|V|	|E|
Gewichte a)	40087	7439951
Gewichte b)	40087	3419182

Tabelle 7: Knoten- und Kantenanzahl von G - Elib condensed a2

Der Korpus wurde aus mehr Dokumenten als die KA 2017 generiert (vgl. Kapitel 7.1). Die eingelesenen Dokumente sind allerdings kürzer als die Dokumente der KA 2017. Aufgrund der Kürze der Dokumente liegen hier weniger Wörter vor als in den KA 2017. Somit enthält das Netzwerk weniger Knoten und auch weniger Kanten. Die Gewichtsverteilung des Graphen, erzeugt durch den Elib condensed a2 Datensatz mit Gewicht a, sind in Tabelle 8 dargestellt.

Gewicht	1	2	3	4	5	...	437	461
Häufigkeit	5095504	1257542	534466	297589	189897	...	1	1

Tabelle 8: Gewichtsverteilung - Elib condensed a2, Gewicht a

Analog wie auch bei den KA 2017 bewirkt die Einführung des Gewichts b, dass wir in Summe weniger Kanten haben und dass die Gewichtsverteilung breiter gestreut ist (vgl. Tabelle 9). Das größte vorkommende Gewicht ist hier 2476. Zum Vergleich ist das größte vorkommende Gewicht bei Gewicht a 461.

Gewicht	1	2	3	4	5	...	2409	2476
Häufigkeit	1700738	650995	321318	194036	130906	...	1	1

Tabelle 9: Gewichtsverteilung - Elib condensed a2, Gewicht b

7.4 Gerankte Wortlisten

Um eine Interpretation der Topics und um weitere Analyseschritte zu den drei Community-Detection-Verfahren zu erleichtern, sollen die Wörter innerhalb der Topic-Ausgabe in eine sinnvolle Reihenfolge nach absteigender Bedeutsamkeit angeordnet werden (vgl. Kapitel 2.3). Dafür wurde sowohl das Common Word Ranking als auch das Bayes Ranking, welches auf den Werten des Positional Idf Ranks basiert, eingeführt.
Als ersten Schritt möchten wir nicht für die resultierenden Topic Wortlisten, sondern für alle Wörter im jeweiligen Korpus eine Wortliste bestimmen, die nach Bedeutsamkeit absteigend sortiert ist. Dabei sollten viel aussagenden Wörter für den jeweiligen Korpus zu Beginn der Liste und weniger gehaltvolle weiter hinten auftreten.

7.4.1 KA 2017

Die Wörter des KA 2017-Korpus wurden zunächst mit dem Common Word Ranking sortiert. Die nachstehende Liste bildet die 15 höchst gerankten Wörter zusammen mit ihrer Häufigkeit im gesamten Korpus ab:

Common Word Ranking - KA 2017
[(StGB, 9474), (Straße, 7620), (Maßnahme, 7108), (Herstellung, 6235), (Erbringung, 4166), (Land, 3956), (sonstig, 3793), (Mercedes, 3743), (Landkreis, 3712), (polizeilich, 3693), (Euro, 3690), (Projekt, 3601), (Wasserlauf, 3600), (Fahrverkehr, 3489),(Dienstleistungen, 3446)]

Die Wahl des Parameters C für das Bayes Ranking ist korpusabhängig. In Kapitel 2.3 lässt sich die Formel zur Berechnung von C nachschlagen und das C für den KA 2017-Korpus ermitteln:

$$C = \frac{\sum_{\forall x} \sum_{d, x \in d} 1}{L_c} = \frac{493983}{134211} = 3.68 \approx 4$$

Wenden wir nun das Bayes Ranking auf den gesamten Korpus an, so erhalten wir die folgende Liste mit den 15 am höchsten gerankten Wörtern und deren berechneten Bayes-Werten:

Bayes Ranking - KA 2017
[(Hasskriminalität, 2.46), (Marode, 2,46), (Breitbandanschluss, 2.36), (Durchschnittsalter, 2.3), (Ausländerfeindliche, 2.28), (Schneller, 2.2), (Eisenbahnbrücken, 2.14), (Breitbandversorgung, 2.11), (Tihange, 2.1), (libyschen, 2.1), (Überstellungen ,2.09), (Doel, 2.02), (Cannabis, 2.02), (UN-BRK, 2.02), (BÄRGIDA, 2.02)]

Vergleichen wir die zwei erzeugten Listen nach Aussagekraft der einzelnen Wörter, so lässt sich zusammenfassen, dass in der Common-Word-Liste mehr allgemeine und weniger aussagekräftige Wörter vorkommen. Allgemeine Wörter sind beispielsweise „Herstellung“ oder „Projekt“ und nichts aussagende beispielsweise „sonstige“.
Die durch Bayes Ranking erzeugte Liste beinhaltet zwar auch ein allgemeines Wort wie „schneller“. Wenn dies aber beispielsweise mit dem Wort „Breitbandanschluss“ in einem Topic vorkommen würde, so wäre dies doch ein zusätzlicher Erkenntnisgewinn zur Beschreibung des Topics.
Zur Ausgabe der Wörter in den Topics werden wir somit in den meisten Fällen das Bayes Ranking verwenden, da dieses Verfahren nach unserer Untersuchung gehaltvollere und aussagekräftigere Wörter und weniger allgemeingültige Wörter weiter nach oben rankt.
Die Ausgabe der Topics beim hSBM-Verfahren erfolgt automatisiert in einer Common-

Word-Sortierung. Die eigentliche Motivation für das Common Word Ranking war es einen Vergleich der Ergebnisse der anderen Verfahren (Louvain und Infomap) mit denen des hSBM zu ermöglichen. Somit haben wir die Common-Word-Sortierung innerhalb der Topics auch für die anderen Verfahren mit ausgegeben.

7.4.2 Elib condensed a2

In der folgenden Liste sind die 15 wichtigsten Wörter nach Common Word Ranking des Elib condensed a2-Korpus zusammengefasst.

Common Word Ranking - Elib condensed a2
[(system, 7644), (model, 6418), (datum, 6178), (high, 4957), (method, 3934), (measurement, 3357), (design, 3262), (time, 3244), (image, 3004), (simulation, 2960), (flow, 2937), (surface, 2928), (approach, 2872), (process, 2848), (mission, 2740)]

Wie zuvor wird vorab ein passendes C für das Bayes Ranking berechnet:

$$C = \frac{802626}{40332} = 19.9 \approx 20 \tag{43}$$

Bayes Ranking - Elib condensed a2
[(contrail, 1.65), (Philae, 1.59), (aerogel, 1.59), (sleep, 1.56), (CMAS, 1.48), (DESIS, 1.47), (spore, 1.46), (Venus, 1.46), (LDACS, 1.4), (Aeolus, 1.39), (sail, 1.38), (InSight, 1,37), (SpaceLiner, 1.35), (MASCOT, 1.35), (VIRTIS, 1.33)]

Auch hier kommen wir zu dem Entschluss, dass die Bayes-Wortliste aussagekräftigere Wörter zur Beschreibung von Topics für Luft-, Raumfahrt, Energie und Verkehr enthält als die Common-Word-Wortliste. Hier treten sehr spezifische Wörter auf, wie beispielsweise „Philae“, „aerogel“ und „DESIS“. Dabei steht das Wort „Philae“, für die erste Raumsonde, die weich auf einem Kometen landete, und „aerogel“ ist ein Festkörper, der zu fast 100 % aus Poren besteht. Hierzu wird viel im Bereich Werkstoff-Forschung im DLR geforscht. Die Abkürzung „DESIS“ steht für ein Instrument zur Erdbeobachtung. In der Common-Word-Wortliste treten häufiger allgemein gültige Wörter wie „system“, „high“ und „method“ auf. Hier stellt es sich als schwieriger heraus, anhand dieser Wörter erste Themen der Luft- und Raumfahrt zu erkennen.

7.5 KA 2017 - Experimentelle Ergebnisse der Community-Detection-Verfahren

In diesem Kapitel sowie in dem analogen Elib-Kapitel 7.6 werden die Ergebnisse der Verfahren zusammengefasst. Die folgenden Punkte liefern eine Übersicht über die unterschiedlichen Analyseschritte.

Vergleich verschiedener Durchläufe: Da die Verfahren nichtdeterministisch sind, können in jedem Durchlauf unterschiedliche Wortlisten der Topics generiert werden. Um einen beliebigen Durchlauf als Ergebnis wählen zu dürfen, schauen wir uns sowohl die Stabilität als auch die Qualität der Ergebnisse an. Wir analysieren dafür im Hinblick auf die Bewertung der Stabilität der ausgegebenen Topics, wie häufig die Wörter für unterschiedliche Durchläufe im selben Topic auftreten. Somit wollen wir sicherstellen, dass die Wörter meist in den selben Topics vorkommen. Um die Ergebnisse nach ihrer Qualität zu bewerten, vergleichen wir zusätzlich sowohl die ausgegebenen Modularitäten als auch die Codelänge für das Infomap-Verfahren.

Berechnung der Modularität/Codelänge: Der Modularitätswert wird für die gefundene Community-Partition auf jeder ausgegebenen Ebene berechnet. Diese Werte werden hinterher in Kapitel 7.7.1 zum Vergleich der Verfahren verwendet. Im Map Equation-Verfahren möchten wir hier zusätzlich die minimale Codelänge ausgeben.

Ausgabe der Wortlisten der Topics: Das Verfahren wird auf den Graphen mit den ausgewählten Gewichten angewendet. Dabei werden für unterschiedliche Ebenen Topics in Form von Wortlisten generiert. Die Wortlisten werden nach dem ausgewählten Ranking sortiert. In der Ausgabe werden die 200 wichtigsten Wörter nach dem gewählten Ranking berücksichtigt.

Topic-Verteilung innerhalb der Dokumente: Hier werden die Dokumente zusammen mit den gefundenen Topics innerhalb dieser Dokumente ausgegeben. Entscheidend ist hierbei, dass keine gleichmäßige Verteilung der Topics innerhalb eines Dokuments vorliegt. Ein Dokument soll im besten Fall durch ein oder wenige Topics charakterisiert sein. Um zu gewährleisten, dass wir je Dokument nur die Topics auswählen, die wichtig sind, haben wir eine Schranke eingeführt. Wir betrachten nur all die Topics, deren Wörter mehr als 33 % des jeweiligen Dokuments ausmachen. Also müssen mehr als 33 % der Wörter des jeweiligen Dokuments dem Topic zugeordnet sein.

Topics vs. Klassifikationen: Wir vergleichen für jedes Dokument die gefundenen Topics mit den vorgegebenen Elib-Forschungsgebieten oder KA-Sachgebieten. Die Ergebnisse werden in einer Heat Map veranschaulicht.

Für die Ergebnisse des Verfahrens Louvain mit Gewicht a und sortiert nach Bayes Ranking werden alle diese Analyseschritte durchlaufen. Für die darauffolgenden Verfahren werden meist nur ausgewählte und auf die Ergebnisse abgestimmte Analysen durchgeführt und dargestellt.

7.5.1 Modularitätsoptimierung

In Python gibt es eine mit der Graphenbibliothek networkx [38] kompatible Implementierung des Louvain-Algorithmus. In der Python Paketverwaltung pip heißt sie python-louvain; nach Installation kann sie als Python-Paket community aufgerufen werden. In GitHub [32] sind alle Codes von Thomas Aynaud et al. hinterlegt und in [31] sind sämtliche Python Funktionen beschrieben, die zur Ausführung des Codes benötigt werden. Im Code „Topic Modeling - Louvain und Infomap“ können die Ergebnisse für gegebene Korpora erneut generiert werden.
Nach Aufbereitung der Datensätze (vgl. Kapitel 2) und Erstellung der Graphen für das jeweilige Gewicht a und Gewicht b haben wir den in Python hinterlegten Algorithmus auf den erstellten Graphen angewendet.
Der Algorithmus gibt eine hierarchische Struktur von Knoten-Partitionen aus. Auf jeder Ebene wird also eine Anzahl von Topics in Form von Wortlisten generiert und ausgegeben und die Modularität zu dieser Knoten-Partition berechnet.
In diesem Kapitel möchten wir drei unterschiedliche Topic-Ergebnisse beleuchten. Im ersten Teil werden die Topic-Ergebnisse zu Louvain Gewicht a und Bayes Sortierung dargestellt. Zum Vergleich der Sortierungsverfahren für die Topic-Ausgabe wird das Common Word Ranking für die gleiche Topic-Ausgabe generiert. Schließlich möchten wir uns die generierten Topic-Listen für Gewicht b mit Bayes Ranking genauer ansehen.

Vergleich verschiedener Durchläufe - Gewicht a

Hier vergleichen wir die Topic-Ausgabe für zwei verschiedene Durchläufe anhand einer Stichprobe. Betrachten wir die Wörter aus Topic t im ersten Durchlauf auf Level l. Dabei stellen wir uns die Frage, wie viele dieser Wörter auch im zweiten Durchlauf auf dem selben Level l in einem Topic landen. Für jedes Topic des ersten Durchlaufs wird die Topic-Verteilung des zweiten Durchlaufs in einem Histogramm dargestellt (vgl. Abbildung 14). Wichtig ist hierbei zu berücksichtigen, dass im besten Fall beide Durchläufe die gleiche

Anzahl an Topics ausgeben, sodass dies einen Vergleich gut zulässt. Zudem wählen wir bei dieser Betrachtung nur die best gerankten Wörter des Korpus nach Bayes aus. Wir beziehen ausschließlich die Wörter mit Bayes Wert > 0.84 in die Auswertung mit ein. Dies sind im KA 2017-Korpus 2490 Wörter.
Veranschaulichen wir uns das Vorgehen in Abbildung 14 genauer: Wir vergleichen die Ausgabe von 17 Topics $(0 - 16)$ in zwei Durchläufen auf der gröbsten Ebene (Level 2) miteinander. Im ersten Histogramm (1. Zeile, 1. Spalte) lässt sich ablesen, dass mehr als 60 % der Wörter aus Topic 0 des ersten Durchlaufs im zweiten Durchlauf in einem Topic liegen. Im zweiten Histogramm (1. Zeile, 2. Spalte) lässt sich Ähnliches für Topic 1 beobachten. Das dritte Histogramm (2. Zeile, 1. Spalte) zeigt sogar, dass mehr als 75 % der Wörter aus Topic 2 im ersten Durchlauf auch hier in einem Topic, und zwar Topic 2 liegen. Ähnliches gilt für die Histogramme $4, 5$ und 6 (bzw. Topic $3, 4$ und 5). Bei Topic 6, im Histogramm, abzulesen in der 4. Zeile und 1. Spalte, werden die Wörter des Topics auf zwei Topics verteilt, und zwar Topic 1 und Topic 8. Dennoch kommen die meisten Wörter, ca. 35 %, in Topic 3 vor. Das gleiche gilt für das Histogramm in der 1. Spalte und 6. Zeile. Auch hier verteilen sich die Wörter des Topics des ersten Durchlaufs größtenteils auf zwei unterschiedliche Topics. Es lässt sich zusammenfassen, dass in den meisten Fällen die Wörter der Topics des ersten Durchlaufs größtenteils auf ein jeweiliges Topic im zweiten Durchlauf zugeteilt werden.

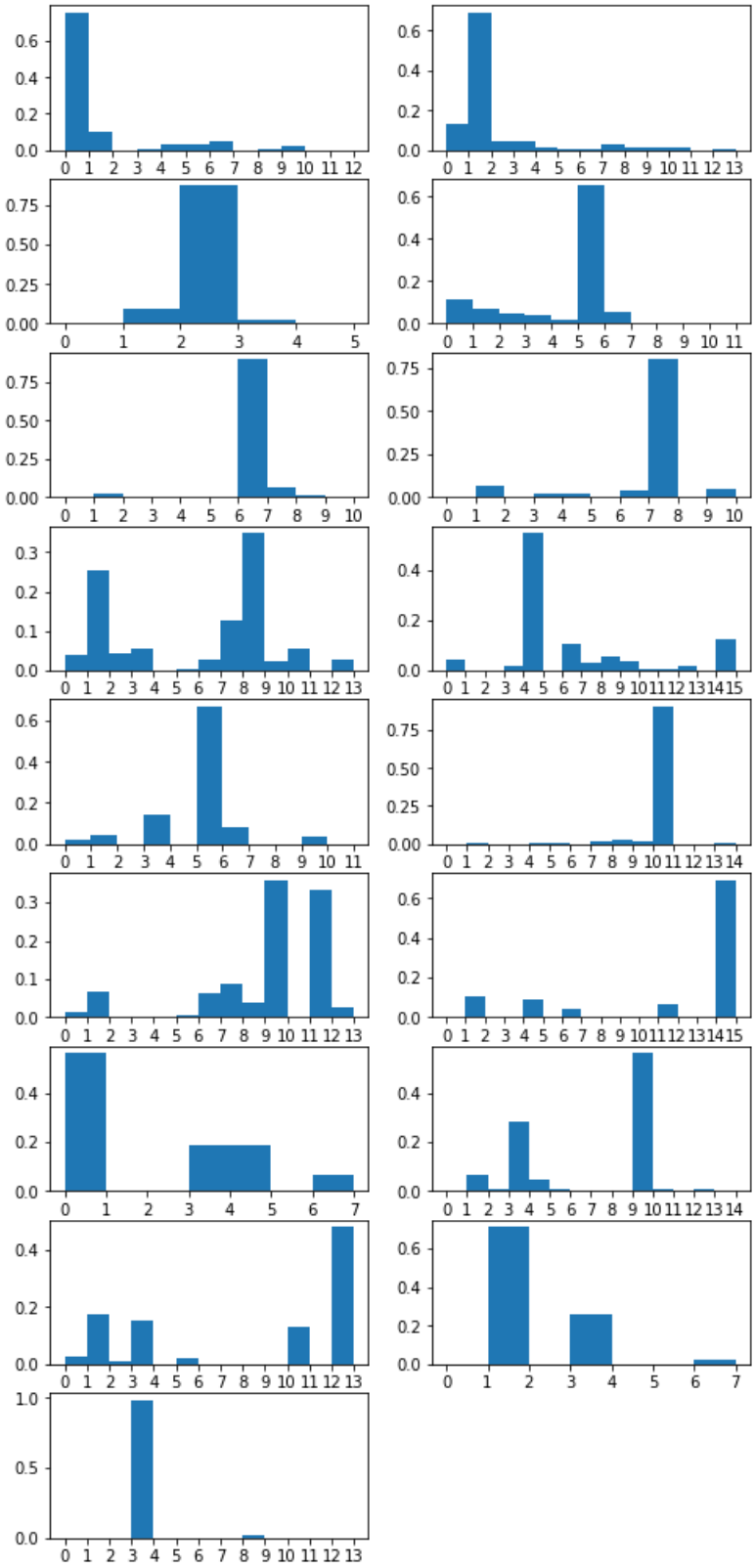

Abbildung 14: Vergleich Topic-Ausgabe für zwei Durchläufe auf Ebene 2, Louvain

Dies ist eine notwendige, aber keine hinreichende Bedingung einen beliebigen Durchlauf

zu wählen. Um zusätzlich die Qualität der Ergebnisse zu überprüfen können wir an dieser Stelle für unterschiedliche Durchläufe i die Modularität berechnen. In der folgenden Tabelle ist diese für alle ausgegebenen Ebenen zusammengefasst:

Ebene	0	1	2
Modularität i=0	0.459	0.506	0.506
Modularität i=1	0.468	0.505	0.506
Modularität i=2	0.468	0.507	0.508
Modularität i=3	0.473	0.509	0.51
Modularität i=4	0.467	0.500	0.501

Tabelle 10: Modularität für fünf Durchläufe - Louvain, KA 2017, Gewicht a

Vergleichen wir die ausgegebenen Modularitätswerte auf der gröbsten Ebene 2 in Tabelle 10, so stellen wir fest, dass die Werte immer im Bereich $[0.501, 0.51]$ liegen.
Mit den zwei beschriebenen Argumenten, stellen wir uns frei eine beliebigen Durchlauf aus den vielen als Ergebnis auszuwerten. Natürlich kann es an dieser Stelle auch nützlich und sinnvoll sein, die zufälligen Topic-Generierungen genauer zu berücksichtigen. Dies bedarf allerdings weiteren Überlegungen, die wir in dieser Arbeit nicht anstellen werden.

Berechnung der Modularität - Gewicht a

Der berechnete Modularitätswert ist für die gefundene Community-Einteilung auf den unterschiedlichen Ebenen in Tabelle 11 festgehalten:

Ebene	0	1	2
Modularität	0.47	0.501	0.502

Tabelle 11: Modularität - Louvain, KA 2017, Gewicht a

Ausgabe der Wortlisten der Topics - Gewicht a, Bayes

In der folgenden Auflistung ist eine beliebige Topic-Ausgabe für Gewichte und Bayes Ranking auf der gröbsten Ebene (Ebene 2) festgehalten.

Topic 0
libyschen EUNAVFOR türkisch Arbeitsbesuch ISIS Vorschub Konfrontation palästinensisch schwerbewaffnet EU-Außenpolitik Europol libysche zweitgrößte HOME Fortbildungsveranstaltung
Topic 1
Freibetrag Tarifverträgen Arbeitsvolumen Versicherungspflicht Rentenversicherung Versicherungszeiten Arbeitsverträgen Rente Verpackungsmüll Statist Sozialversicherungsabkommen Prekäre MiLoG Künstlersozialkasse JOBBÖRSE
Topic 2
DITIB Zurückziehung Stattgefundene Angefragte Franco Wehrmacht G20-Gipfel Diyanet Vorratsdatenspeicherung UETD Sanitätsdienstliche Nutzungsüberlassung TKÜV Strafvollstreckung Snowden
Topic 3
DPMA Infrastrukturabgabe Mozilla License Lesser Laufend LGPL GPLv2 Freeware Firefox Ergänzende Eclipse AGPL Apache BZSt
Topic 4
Hasskriminalität Ausländerfeindliche Überstellungen BÄRGIDA PMK- ausländerfeindlich Schwerpunktfragen Schultyp Oberschule Mittelschule Jugendoffiziere Heeresmusikkorps Gemeinschaftsschule Berufskolleg Kriminalität-rechts
Topic 5
Cannabis UN-BRK betäubungsmittelrechtlicher Arzneimittel vernachlässigen Telefonwerbung IfSG Patientin Patient Wirkstoff BfArM WpHG Restschuldversicherungen Inkassodienstleistungen Cannabispatientinnen
Topic 6
CETA Daimler Audi Dieselgipfel Volkswagen Aktueller unseriöse BaFin NABU Dividendenstichtag Siemens Opel Abgasskandals Übertragungsnetzbetreiber Hans-Georg
Topic 7
Schnellladestationen Verbundprojekt Rast Ladepunkten Ladepunkte Stickstoffdioxid Stationsname Stationscode Harnstoff Grundwasserkörper Stickoxid Ladeinfrastruktur Ladesäulen Porsche blog
Topic 8
Handelsabkommen schrumpfend Spätestens Freihandelsabkommen BBSR Tier Erwärmung Klimakrise Produktgruppen Polizeigewalt Geflügel EU-Japan halbstädtischen Pflanzenschutz Methan

Topic 9
Kriegswaffen Gewehr Waffenexporteure EG-Dual-Use-Verordnung Kleinwaffen Rüstungsgüter Sammelausfuhrgenehmigungen Sammelausfuhrgenehmigung Güterbewegungen Rüstungsexporte Luftweg Dual-Use-Güter Sportgewehre Sipri Kumulierter
Topic 10
Marode Durchschnittsalter Eisenbahnbrücken Zustandskategorie KFZ-Verkehr Streckenkilometer Schieneninfrastruktur Schienenwege Qualitätskennzahl Netzsegmentes Ortsumgehung Verkehrsstationen Fahrverkehr leistungsfähige Bahnsteige
Topic 11
Tihange Doel Atomkraftwerk Atomkraftwerken FANC Wasserstoffflocken Fukushima Bodenseegürtelbahn URENCO Uran russische radioaktiv Übungsflüge seismische leise
Topic 12
BeschV Maghreb-Staaten Teilvorhaben Unternehmertum Anti-IS-Kämpferinnen Entwicklungsländern Internationalisierung Schwangerschaft Vielseitigkeit Hochschulzugang Mittelalter Hochschulbereich Chancengerechtigkeit Internets Körper
Topic 13
belgisch Konzert StAG Menschheit Kriegsgefangene Gedenkstättenkonzeption Familiennachzug Eheschließung Ablehnungsquote Horchposten Staatsangehörigkeitsgesetzes Rückkehrbereitschaft Goethe-Institute Annullierung Nachzug
Topic 14
Breitbandanschluss Schneller Breitbandversorgung Internetzugängen Mbit Kreisfreie Breitbandanschlüssen Haltepunkt Fördergegenstand Wirtschaftlichkeitslückenmodell halbstädtischem Standortfaktor städtisch Landkreisen Downstream
Topic 15
Oberflächengewässer Fließgewässern BauGB WRRL Vornutzung Verbilligung VerbR Namensbezeichnung Leerstand LMBV Gewässerkörper Hauptstraße Sozialwohnung Wohneinheiten Allee
Topic 16
Hooligans Athlet Hooligan-Szene Doping Sportler Spitzensportler Olympiastützpunkte Fußballvereine Einheitspartei Dopingopfer-Hilfegesetz DOHG HoGeSa sozialistisch Sportstätten Sportlerinnen

Es bleibt die Aufgabe des Analysten aus den gegebenen Wortlisten jeweils ein Topic zu interpretieren. Schauen wir uns beispielsweise Topic 1 an, so können wir nach Ablesen der Wörter darauf schließen, dass es hier vermutlich um Arbeit geht. In Topic 14 kann man erahnen, dass Informationstechnik eine gute Beschreibung für das Topic ist. Topic 9 lässt

sich mit den Begriffen Verteidigung und Außenpolitik beschreiben.

Topic-Verteilung innerhalb der Dokumente - Gewicht a, Bayes

Wir möchten uns für alle Kleinen Anfragen Dokumente die Topic-Verteilung genauer anschauen. Somit lassen wir uns für alle Dokumente die gefundenen Topics innerhalb dieser Dokumente ausgeben. In Tabelle 12 sind für eine kleine Stichprobe, nämlich für vier Dokumente auf der gröbsten Ebene (Level 2) die Topics mit deren relativen Auftrittshäufigkeiten innerhalb des jeweiligen Dokuments abgebildet.

	1811833	1900334	1812335	1812046
0	0.21	0.216		
1	0,027			
6			0.902	
9	0.763	0.784	0.098	
10				1
$\sum$	1	1	1	1

Tabelle 12: Vier Dokumente + Topic-Verteilung - Louvain, KA 2017, Gewicht a

Ein gutes Ergebnis ist es, festzustellen, dass keine gleichmäßige Verteilung der Topics innerhalb der Dokumente vorliegt. Eine gleichmäßige Verteilung der Topics innerhalb der Dokumente ist nicht wünschenswert, da sich somit keine bedeutsamen Topics in den Dokumenten zu erkennen gibt.
Betrachten wir die Kleine Anfrage 1811833: 76.3 % der Wörter des Dokuments wurden dem Topic 9, eben mit den Begriffen Verteidigung und Außenpolitik beschrieben, zugeordnet. Der Titel der Kleinen Anfrage ist „Rüstungsexporte 2013 bis 2017“. Für die Kleine Anfrage mit der Nummer 1812046, können wir hier den Titel „Marode Eisenbahnbrücken in Nordrhein-Westfalen“ herausfinden. In der Topic-Liste beinhaltet Topic 10 viele Wörter zum Thema Infrastruktur, insbesondere Schieneninfrastruktur.
Zudem ist es wesentlich zu erkennen, dass es pro Dokument nur wenige, wirklich wichtige Topics gibt. Uns interessiert es weniger, wenn 1 % der Wörter innerhalb eines Dokument einem Topic zugeordnet wird. Stattdessen sind die wirklich starken Topics von großem Interesse. Somit haben wir eine Schranke eingeführt, die festlegt ab wann ein Topic für ein Dokument als charakteristisch erscheint. Diese haben wir auf 33 % festgelegt. Für diese Schranke erhalten wir für die gleichen ausgewählten Dokumente wie in Tabelle 12 die folgende in Tabelle 13 zusammengefasste Topic-Verteilung.

	1811833	1900334	1812335	1812046
6			0.902	
9	0.763	0.784		
10				1

Tabelle 13: Vier Dokumente + Topic-Verteilung 33 % - Louvain, KA 2017, Gewicht a

Topics vs. Klassifikationen - Gewicht a, Bayes

Die Topic-Verteilung der Dokumente mit der 33 % Schranke wird für den nächsten Analyseschritt vorausgesetzt. Hier möchten wir unsere gefundenen Topics mit den Sachgebieten der Kleinen Anfragen abgleichen. Diese wurden in Kapitel 7.2 aufgelistet.
Das Vorgehen ist wie folgt: Für jedes der 216 Dokumente mit Sachgebiet Zuordnung wurde genau das Topic bestimmt, das dem jeweiligen Dokument in unserem Verfahren zugeordnet wurde. Die Zuordnung wird dann in einer Matrix zusammengefasst. Wenn beispielsweise 10 Dokumente, welche mit dem Sachgebiet Außenpolitik klassifiziert sind, von uns das Topic 0 zugewiesen bekommen haben, so wird der Matrixeintrag, Zeile für Topic 0 und Spalte für Sachgebiet „Außenpolitik“, mit einer 10 befüllt. In der abgebildeten Matrix ist dieser Eintrag mit gelb hinterlegt. Wenn die aufsummierten Einträge einer Zeile, das heißt aufsummierten Einträge eines Topics < 4 betragen, dann haben wir dieses Topic in der Darstellung nicht weiter berücksichtigt.

$$\underbrace{\left(\begin{array}{ccccc} 10 & & & & \\ 25 & 1 & & & \\ & & 30 & & \\ & & & 64 & \\ & & & & \\ & & & & 30 \\ & & 35 & & \end{array}\right)}_{\text{Sachgebiete}} \Big\} \text{Topics}$$

Aus der soeben definierten Matrix werden dann zwei Heat Maps, Heat Map col und Heat Map row, für jede Ebene erstellt:

- Heat Map col: Hier wurde für jede Spalte, das heißt für jedes Sachgebiet, die Topic-Häufigkeiten aufsummiert. Anschließend wurde jeder Eintrag der Spalte durch diese Summe der Topic Häufigkeit dividiert. Dies ermöglicht ein Vergleich der Topics für jedes Sachgebiet. Wollen wir beispielsweise wissen welche Topics im Sachgebiet „Arbeit und Beschäftigung“ wichtig sind, so können wir dies in der Heat Map col

ablesen.

- Heat Map row: Hier wurde für jede Zeile der Matrix, das heißt für jedes Topic, die Summe der Sachgebiet-Häufigkeiten summiert. Anschließend wurde jeder Eintrag der Zeile durch die Summe der berechneten Sachgebiet-Häufigkeiten dividiert. Dies ermöglicht eine Vergleich der Sachgebiete für jedes Topic. Interessiert uns beispielsweise mit welchem Sachgebiet das Topic 0 beschrieben werden kann, so können wir dies in dieser Heat Map row ablesen.

Die zwei soeben erklärten Heat Maps werden für die Topic-Ausgabe Louvain auf der gröbsten Ebene ausgegeben.

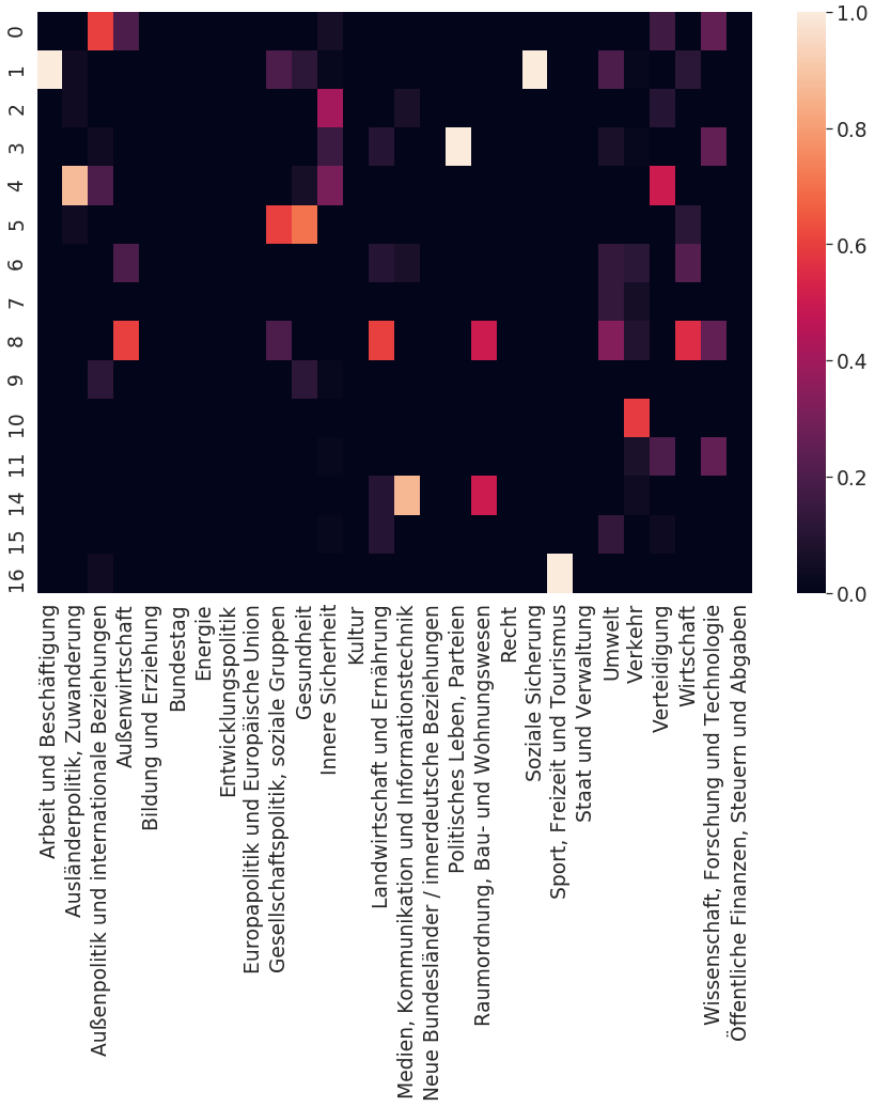

Abbildung 15: Heat Map col - Louvain, Gewicht a, Bayes

Stellen wir uns die Frage in welchem Topic vor allem das Sachgebiet „Arbeit und Beschäftigung“ vertreten ist, so liefert uns die Heat Map col 15 das Topic 1. Analog können wir uns die Frage für das Sachgebiet „Medien, Kommunikation und Informationstechnik“ stellen. Hier sehen wir, dass vor allem Topic 14 in diesem Bereich vertreten ist.

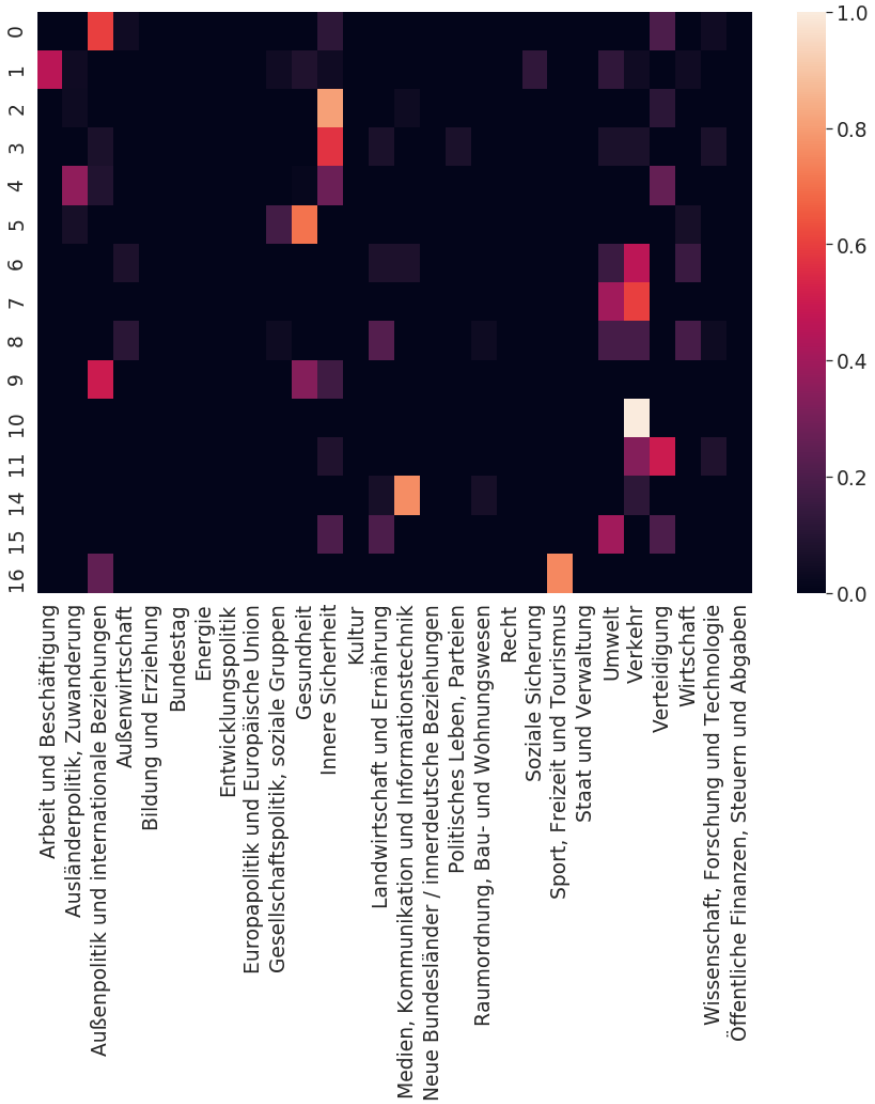

Abbildung 16: Heat Map row- Louvain, Gewicht a, Bayes

Die Heat Map row 16 gibt Auskunft darüber, welches Sachgebiet häufig im jeweiligen Topic vorkommt. Schauen wir uns beispielsweise Topic 4 an: Die Heat Map row gibt an, dass vor allem Themen der „Ausländerpolitik, Zuwanderung “ und etwas weniger „Innere Sicherheit “ in dem Topic 4 vorkommen. Vergleichen wir diese Sachgebiete mit jenen, die oben zusammengefassten 15 Wörtern von Topic 4, so stellen wir fest, dass die durch die Heat Map gefundenen Themen das Topic gut beschreiben.

Ausgabe der Wortlisten der Topics - Gewicht a, Common Word

Als Alternative zur Bayes-Sortierung der Wörter innerhalb der Topics kann die Common-Word-Sortierung der Wortlisten in den Topics ausprobiert werden. In Kapitel 7.4 haben wir bereits erkannt, dass die Bayes-Sortierung aller Wörter im Korpus bedeutsamere Wörter weiter nach oben rankt als die Common-Word-Sortierung. Die Wortlisten der Topics $1, 4$, und 14 sind zum einen nach dem Common Word Ranking und zum anderen nach dem Bayes Ranking sortiert. Für jedes Topic wird die jeweilige gefundene Heat-Map-Klassifikation wiedergegeben.

Klassisikation Heat Map	Bayes	CommonWord
Topic 1: Arbeit, Beschäftigung	Freibetrag Tarifverträgen Arbeitsvolumen Versicherungspflicht Rentenversicherung Versicherungszeiten Arbeitsverträgen Rente Verpackungsmüll Statist Sozialversicherungsabkommen Prekäre MiLoG Künstlersozialkasse JOBBÖRSE	Erbringung Anteil Prozent Entwicklung handeln Kraftfahrzeug Tätigkeit Frau Mann Region Reparatur Verwaltung Erde Arbeit Information
Topic 4: Ausländerpolitik, Innere Sicherheit	Hasskriminalität Ausländerfeindliche Überstellungen BÄRGIDA PMK- ausländerfeindlich Schwerpunktfragen Schultyp Oberschule Mittelschule Jugendoffiziere Heeresmusikkorps Gemeinschaftsschule Berufskolleg Kriminalität-rechts	StGB Herstellung Land sonstig Person Berlin Hamburg Volksverhetzung Quartal Algerien Bundeswehr Absatz Anlage Datum Straftat
Topic 14: Medien, Kommunikation, Informationstechnik	Breitbandanschluss Schneller Breitbandversorgung Internetzugängen Mbit Kreisfreie Breitbandanschlüssen Haltepunkt Fördergegenstand Wirtschaftlichkeitslückenmodell halbstädtischem Standortfaktor städtisch Landkreisen Downstream	Landkreis Dienstleistungen Stadt Förderrichtlinie ländlich Beratungsleistungen externe Haushalt Gemeinde Unternehmen Stein Kommune Mbit wirtschaftlich Raum

Betrachten wir die ausgegebenen Wörter für das Topic 1, welches nach der Heat-Map-Klassifikation mit den Begriffen „Arbeit und Beschäftigung“ kategorisiert wurde. In der Bayes-Sortierung sind viele Wörter dabei, die auf das Thema Arbeit hinweisen. Die Begriffe „Tarifverträge“, „Rentenversicherung“, „Arbeitsverträge“, „Rente“, „Jobbörse“ sind Arbeitsthemen. In der Common-Word-Liste tauchen allerdings nur die Wörter „Tarifverträge“ und „Arbeit“ auf, welche klar diesem Thema zuzuordnen sind. Ähnliche Muster lassen sich in Topic 14 erkennen. Das Topic handelt von „Medien, Kommunikation und Informationstechnik“. In der Bayes-Liste wird das IT-Topic mit den Worten „Breitband-

anschluss“, „Internetzugängen“, „Mbit“, „Wirtschaftlichkeitslückenmodell“ sehr gut umrissen. Wohingegen die Common-Word-Liste hier nur das Wort „Mbit“ beinhaltet, das sich klar dem IT-Topic zuordnen lässt. Dieses Ergebnis veranlasste uns dazu, die Common-Word-Sortierung in den weiteren Analysen erst einmal zurückzustellen.

Berechnung der Modularität - Gewicht b

Der Modularitätswert verbessert sich. Dieser beträgt nun auf der gröbsten Ebene 0.615.

Ausgabe der Wortlisten der Topics - Gewicht b, Bayes

Es kann ein alternatives Netzwerk mit Gewicht b, anstelle von Gewicht a, erzeugt werden. Ein so konstruierter Graph besitzt nach Kapitel 7.3.1 weniger Kanten als der Graph, der mit Gewicht a erzeugt wurde. Dies liefert uns technische Vorteile. Wir erhoffen uns aber auch durch einen solchen Graphen inhaltliche Vorteile in der Topic-Bildung. Bei der Konstruktion eines Graphen mit Gewicht b wird im Gegensatz zu Gewicht a die Häufigkeit jedes Wortes im Dokument mit berücksichtigt.
Wenden wir den Louvain-Algorithmus auf ein solches mit Gewicht b konstruiertes Netzwerk an, so erhalten wir nur wenige Topics auf der gröbsten Ebene. Die Ausgabe beinhaltet in diesem Durchlauf lediglich 8 Topics.

Topics vs. Klassifikationen - Gewicht b, Bayes

Betrachten wir in Abbildung 17 die Heat Map row auf der gröbsten Ebene mit 8 Topics, so können wir erkennen, dass in jedem Topic verschiedene Sachgebiete als wichtig erscheinen. So ist beispielsweise in Topic 0 „Ausländerpolitik“, „Innere Sicherheit“ und „Verkehr“ wichtig. In Topic 1 „Verkehr“ und „Verteidigung“. In Topic 2 sind die Sachgebiete „Medien, Kommunikation und Informationstechnik“, „Umwelt“ und ein bisschen „Verkehr“ sowie „Gesellschaftspolitik“ und „Gesundheit“ relevant.

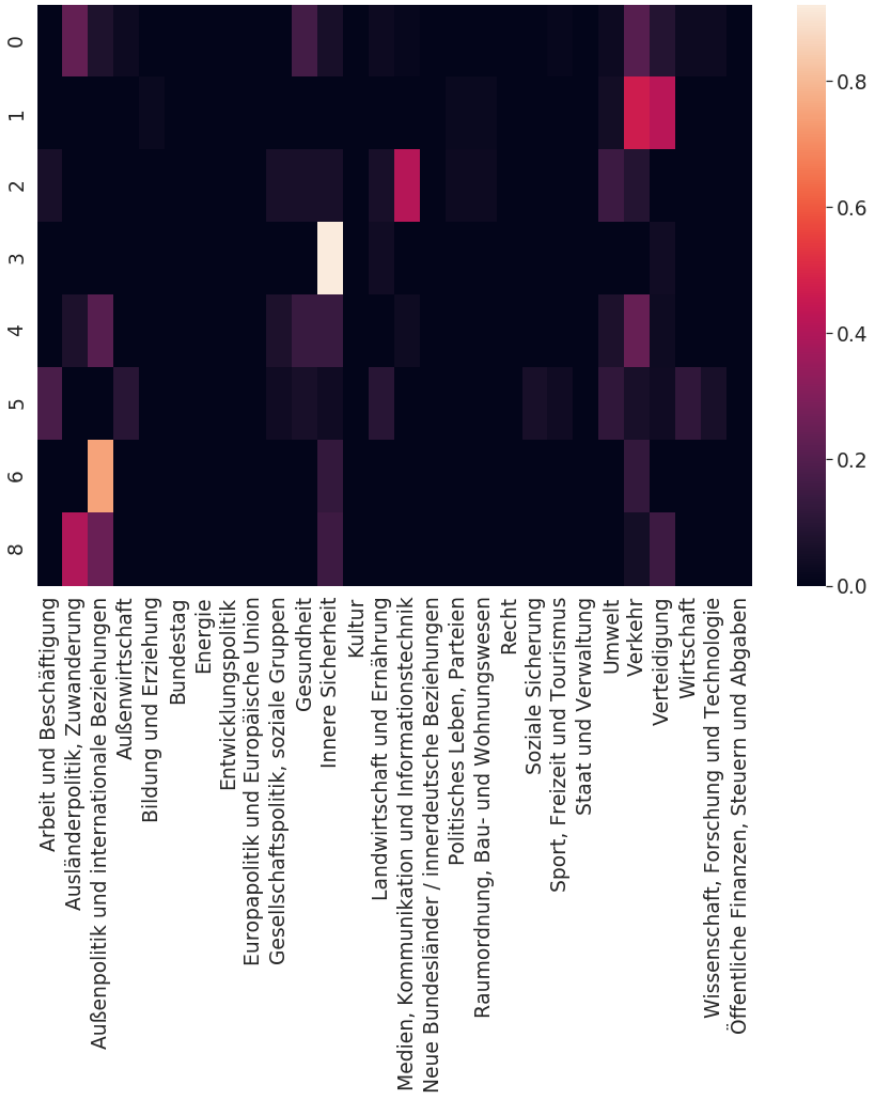

Abbildung 17: Heat Map row - Louvain, Gewicht b, Bayes

Zu diesem Durchlauf schauen wir uns im Folgenden die Wortliste der ersten drei Topics an:

Topic 0
Tihange libyschen Überstellungen Doel Cannabis UN-BRK Atomkraftwerk EUNAVFOR CETA BeschV türkisch Schwerpunktfragen Qualitätskennzahl DITIB Atomkraftwerken
Topic 1
Marode Durchschnittsalter Eisenbahnbrücken Zustandskategorie KFZ-Verkehr Streckenkilometer Schieneninfrastruktur Schienenwege Schultyp Oberschule Mittelschule Jugendoffiziere Heeresmusikkorps Gemeinschaftsschule Berufskolleg
Topic 2
Breitbandanschluss Schneller Breitbandversorgung Internetzugängen Mbit Kreisfreie Oberflächengewässer Breitbandanschlüssen schrumpfend Fördergegenstand Wirtschaftlichkeitslückenmodell Fließgewässern Spätestens halbstädtischem BBSR

Auf den ersten Blick wird bereits ersichtlich, dass Topic 0 jetzt unterschiedliche Themen abdeckt: So sind zum einen Begriffe der „Außenpolitik“ wichtig, wie die Worte „libyschen“ und „türkisch“. Auf der anderen Seite sind aber auch Begriffe der „Atompolitik“ wie „Tihange“ und „Atomkraftwerk“ in Topic 0 enthalten. Und zusätzlich Wörter die im Bereich „Gesundheit“ einzuordnen sind wie „Cannabis“. Es macht also den Anschein, dass für ein Topic mehrere Sachgebiete relevant sind. Aufgrund der kleineren Topicanzahl als in Gewicht a ist es eine logische Schlussfolgerung, dass sich die Sachgebiete auf die wenigen Topics verteilen müssen. In einem Topic spielen also mehr Themen zusammen. Dies verdeutlicht wie wichtig es ist, sich vor der Auswahl der Gewichte und der Ebene bewusst zu werden, welchen Erkenntnisgewinn man erwartet. Möchte man viele spezifische Topics oder möchte man wenige gröbere Topics generieren.

7.5.2 Map Equation

Die Idee, Beschreibung und der Code des verwendeten Infomap-Algorithmus ist von Martin Rosvall und Daniel Edler auf der Homepage [33] und [34] hinterlegt. Hier wird nicht die Basisform des in Kapitel 5.4 beschriebenen Algorithmus programmiert, sondern es wurden noch zusätzliche Tricks angewendet. Es wurde ein modifiziertes Verfahren implementiert, das nicht nur einzelne Knoten zwischen Communities verschiebt, sondern auch Verschiebungen von Teilcommunities zulässt (vgl. [13, S.9]). Als Input-Format haben wir uns dafür entschieden ein Pajek-Format einzulesen. Und die Knoten in den jeweiligen Communities werden nach dem Infomap-Durchlauf als tree-Format ausgegeben. Diese Schritte, von der Erstellung Pajek-Format der Knoten und Kanten, bis hin zum Auslesen der Tupel aus dem tree-Format als geeignetes Dictionary sind im Code „Topic Modeling -

Louvain und Infomap“ zu finden. Die Lösungen können hier auch bei Interesse neu generiert werden. Neben der Community-Einteilung der Knoten auf unterschiedlichen Ebenen wird die kleinste gefundene Codewortlänge des Random Walks, die Minimum Description Length (MDL), der gefundenen Knoten-Partition mit ausgegeben. Diese Zielfunktion wird in dem Algorithmus minimiert.
Zunächst wird die Topic-Ausgabe für das Infomap-Verfahren auf den Graphen mit Gewicht a angewendet und die Topic-Listen nach Bayes sortiert. Die Anzahl der ausgegeben Topics ist hier immens groß. Dies veranlasste uns anfangs dazu über eine alternative Gewichtswahl zu Gewicht a nachzudenken. Somit haben wir einen Graphen mit Gewicht b erstellt, das Pajek-Format hierzu generiert und darauf den Infomap-Algorithmus erneut angewendet. Im Folgenden sind die Ergebnisse dargestellt.

Vergleich verschiedener Durchläufe - Gewicht b

Da als Ergebnis für das Infomap-Verfahren Gewicht a nur eine Ebene mit sehr vielen Topics generiert wird, haben wir uns in diesem Schritt dazu entschieden die Topic-Ausgabe mit Gewicht b genauer anzusehen. Auf Level 1 wurden hier 445 Topics ausgegeben, das heißt weniger als bei Gewicht a. Die gröbste Ebene ist in den Ergebnissen zu Gewicht b leider nicht interessant, da fast alle Wörter nur in zwei Topics eingeteilt werden. Auch hier kann man für zwei verschiedene Durchläufe die Topic-Einteilung analysieren und dabei feststellen, dass meist mehr als 75 % der Wörter eines Topics im ersten Durchlauf auch in einem Topic im zweiten Durchlauf liegen. Die Anzahl der Topics beträgt auf der ersten Ebene 445 Topics. Dies kann an dieser Stelle nicht in Form von Histogrammen wie bei der Auswertung in Modularitätsoptimierung veranschaulicht werden. Auch eine Tabelle wäre hier für alle diese Topics zu groß. Somit möchten wir auch an dieser Stelle auf das Jupyter Notebook File „Topic Modeling - Louvain und Infomap“ verweisen.
Des Weiteren wird eine Übersicht mit der minimalen Codelänge für fünf verschiedene Infomap-Durchläufe wiedergegeben:

Ebene	2
Codelänge i=0	7.828
Codelänge i=1	7.828
Codelänge i=2	7.828
Codelänge i=3	7.828
Codelänge i=4	7.828

Tabelle 14: Minimale Codelänge für fünf Durchläufe - Infomap, KA 2017, Gewicht b

Auf die dritte Nachkommastelle gerundet sind die minimalen Codelängen für die fünf unterschiedlichen Durchläufe identisch. Dies und die Tatsache, dass die Wörter eines Topics des ersten Durchlaufs sehr häufig wieder zusammen in einem Topic des zweiten Durchlaufs auftauchen, ermöglicht es uns einen beliebigen Durchlauf als Ergebnis zu wählen.
Dennoch möchten wir die Reihenfolge in den weiteren Analyseschritten beibehalten und uns zunächst die Ergebnisse für Gewicht a und danach Gewicht b anschauen.

Berechnung der Modularität/Codelänge - Gewicht a

Die Modularität für diese eine ausgegebene Ebene mit 467 Topics beträgt 0.48. Auch hier stellt sich wieder die Frage welchen Erkenntnisgewinn man sich erhofft. Möchte man viele spezifische Topics generieren oder nur wenige allgemeine. Die ausgegebene minimale Codewortlänge L dieser Topic-Einteilung beträgt 13.655.

Ausgabe der Wortlisten der Topics - Gewicht a, Bayes

Der Durchlauf des Infomap-Algorithmus auf den Graphen mit Gewicht a hat 467 Topics auf nur einer Ebene ausgegeben. In der folgenden Übersicht sind die 15 am besten gerankten Wörter nach Bayes für vereinzelte Topics abgebildet:

0
Breitbandanschluss Schneller Breitbandversorgung Internetzugängen Mbit Kreisfreie Breitbandanschlüssen Haltepunkt Fördergegenstand Wirtschaftlichkeitslückenmodell halbstädtischem Standortfaktor Landkreisen Downstream halbstädtisch
2
Wildtierhandels Amphibie Wildtiere Haustier exotische nicht-domestizierte Zoonosen Wildfänge Waschbär Tierheimen Salmonellen Privathaltung Managementmaßnahmen Internethandel IAS-VO
8
libyschen EUNAVFOR libysche Einheitsregierung Seenot on-scene Zawiya TRITON Schlauchboot Rettungseinsatz G5-Sahel-Staaten Einsatztruppe Bootsflüchtlingen Abermals Küstenwache
9
SACHS NB-Grenze Schöna -Pirna Vogtl)-BF Rathen Kottewitz Dresden-Klotzsche Erzgeb -Wilthen -GLASHÜTTE -Chemnitz-Siegmar -Bischofswerda Pockau-Lengefe-BF Oberlichtenau

Topics vs. Klassifikationen - Gewicht a, Bayes

Betrachten wir die Kookkurrenzen der soeben ausgegebenen Wortliste mit den hinterlegten KA-Sachgebieten in der Heat Map row in Abbildung 18. Es wurden alle Topics entfernt, die eine Zeilensumme kleiner als 4 haben. Die verbliebenen Topics sind in der Heat Map row dargestellt.

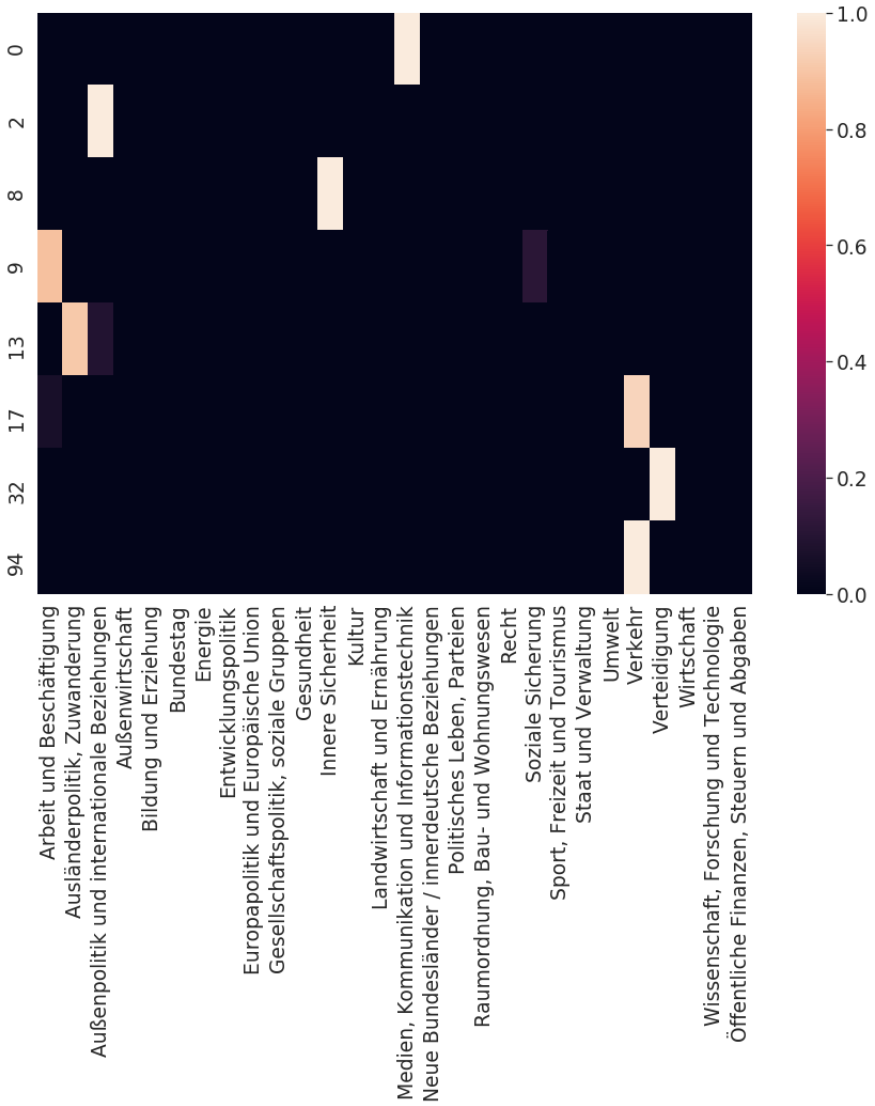

Abbildung 18: Heat Map row - Infomap, Gewicht a, Bayes

Nach Interpretation der Heat Map besteht das Topic 0 zum größten Teil aus „Medien, Kommunikation und Informationstechnik“. Topic 2 besteht aus „Außenpolitik und internationale Beziehungen“. Topic 8 aus Innere Sicherheitsthemen und in Topic 9 sind die Themen „Arbeit und Beschäftigung“ und ein wenig „Soziale Sicherung“ relevant. Ein Vergleich mit den obigen Wortlisten ergibt, dass diese Topics zum Teil etwas spezifischer sind

als die bisher dargestellten. So handelt Topic 2 von Tierhandel beziehungsweise Tierfang. Und wurde zum KA-Sachgebiet „Außenpolitik“ zugeordnet.
Wie eben bereits erwähnt, ist es unser Ziel an dieser Stelle weniger Topics als die 467 erzeugten Topics ausgegeben zu bekommen. Somit wurde ein Graph mit Gewicht b erzeugt mit der Erwartung weniger Topics zu erhalten.

Berechnung der Modularität/Codelänge - Gewicht b

Ebene	0	1	2
Modularität	0.605	0.606	0.022

Tabelle 15: Modularität - Infomap, KA 2017, Gewicht b

Die Vermutung, dass die Topic-Ausgabe auf gröbster Ebene nicht gut ist wird mittels Modularitätsberechnung bestätigt. Diese beträgt auf diesem Level nur 0.022. Für die anderen Ebenen 0 und 1 lassen sich aber hohe Modularitätswerte ablesen. Zudem liegt bei diesem Verfahren eine bessere Codewortlänge vor als bei Gewicht a. Diese beträgt 7.828. Verglichen mit der minimalen Codewortlänge von Gewicht a ist die hier Berechnete um einiges kleiner und somit besser.

Ausgabe der Wortlisten der Topics - Gewicht b, Bayes

Der Durchlauf des Infomap-Algorithmus auf den Graphen mit Gewicht b hat auf der gröbsten Ebene nur noch fünf Topics ausgegeben. Aufgrund der hierarchischen Struktur des Verfahrens wurden viele Topics zusammengefasst. Topic 0 sowie Topic 1 sind sehr groß und Topics 2, 3 und 4 beinhalten lediglich ein Wort. Da es wenig Sinn ergibt, alle Wörter des Korpus auf nur 2 Topics einzuteilen, schauen wir uns hier einge Topic-Listen auf der mittleren Ebene, nämlich Ebene 1 an. In der folgenden Tabelle ist die Ausgabe der 15 bedeutsamsten Wörter nach Bayes-Sortierung ausgegeben.

0
Marode Durchschnittsalter Eisenbahnbrücken Zustandskategorie KFZ-Verkehr Streckenkilometer Schieneninfrastruktur Schienenwege Netzsegmentes Fahrverkehr leistungsfähige Streckenbezeichnung Flutöffn Rad- Streckennummer
11
Städtepartnerschaften Überregional Städtepartnerschaftskonferenz Ostukraine Deutsch-Ukrainische BVVG Halbinsel ukrainisch Haager Fehlanzeige Luftverschmutzung Donbass psychiatrisch Tschernobyl Krim
22
Reaktorkatastrophen Uranabbau Feuchtgebiet Uranabbaus Tansanias naturwelt naturverbrechen Mkuju Energieprobleme Bahi Uranium Dodoma Bergbauindustrie tansanische nuclear-energy-contributes-development-energyconference-africa-looks-options
36
stability koenders-concludes-migrant-return- agreement-with-mali-for-eu afrique-europe-interact Rücküberweisungen Programmtitel Malier KoFinanzierungsbeitrag Identifizierungsmissionen Finanzierungsquelle(n 2016 aei-zeitung web EUTF Migrationsmanagement malischer irregular

Topics vs. Klassifikationen - Gewicht b, Bayes

Geben wir die Heat Map auf gröbster Ebene aus, so stellen wir auch hier fest, dass die Ausgabe wenig sinnvoll erscheint. Die Ebene beinhaltet nur zwei richtige Topics, welche in der Heat Map dargestellt werden. Darin ist zu erkennen, dass Topic 1 Innere Sicherheitsthemen enthält und Topic 0 alle anderen Themen abdeckt.

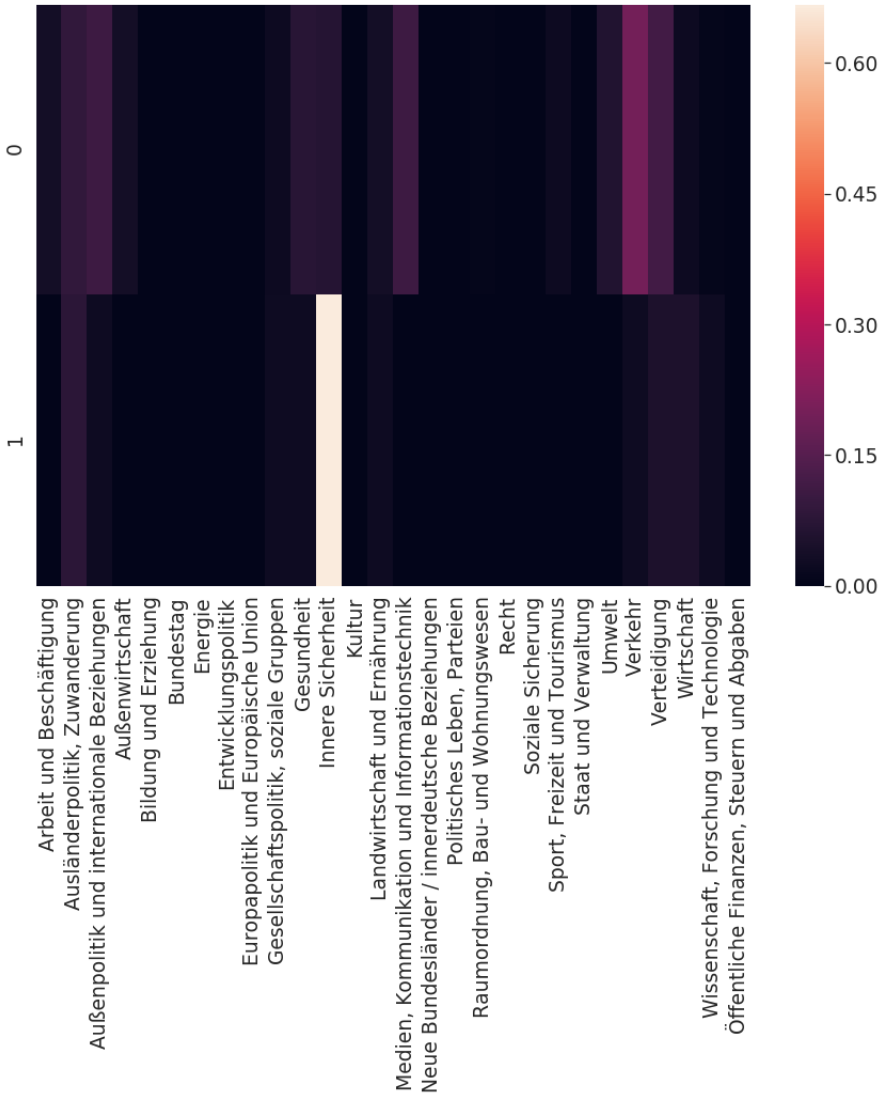

Abbildung 19: Heat Map row - Infomap, Gewicht b, Bayes

In der Heat Map auf Ebene 1 (vgl. Abbildung 20) ist eine größere Anzahl von Topics zu erkennen als in der soeben dargestellten Heat Map zur Ebene 2 (vgl.Abbildung 19).

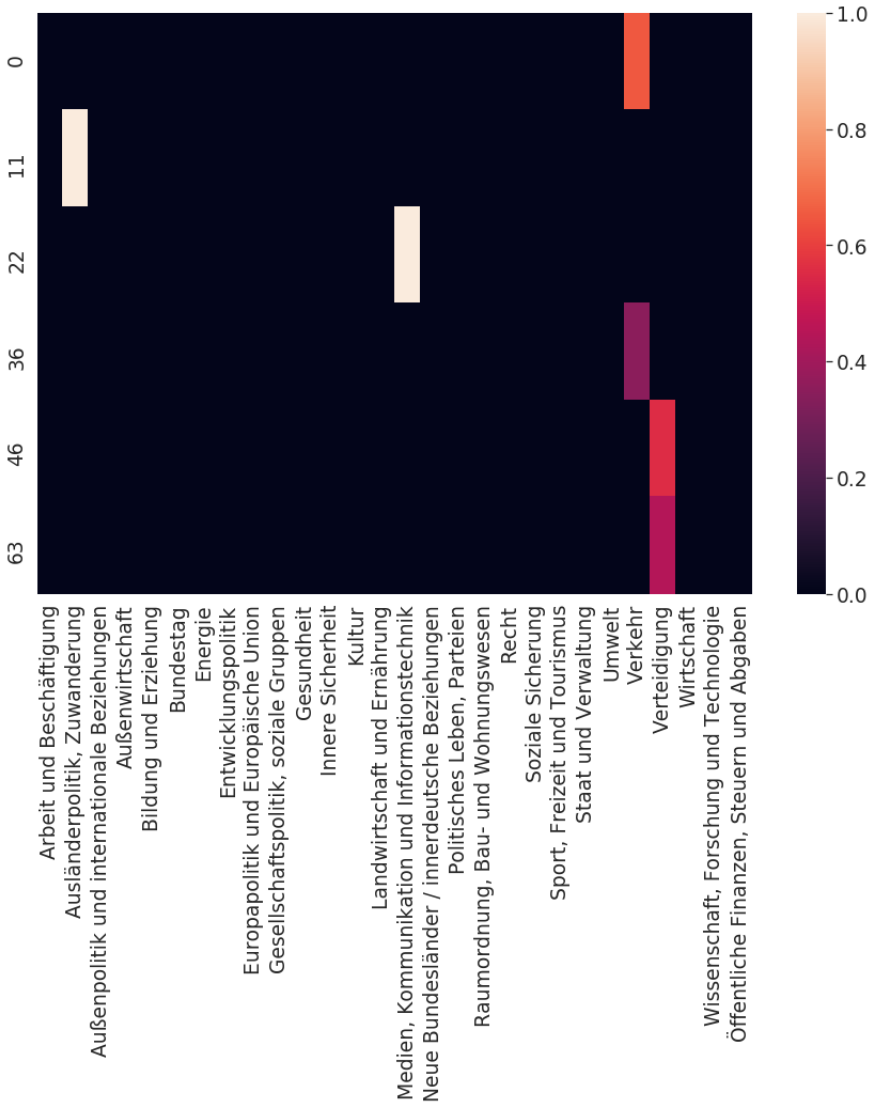

Abbildung 20: Heat Map col - Infomap, Gewicht b, Bayes

Wichtige Topics im Bereich „Verkehr“ sind hier Topic 0 und das Topic 36. In Topic 0 hingegen lassen sich zunächst keine klaren Verkehrsthemen erkennen. Im Bereich „Ausländerpolitik und Zuwanderung“ ist hier vor allem Topic 11 relevant. Dies passt auch zu einigen Wörtern der obigen Topic-Liste. So beinhaltet diese zum Beispiel „Städtepartnerschaftskonferenz“, „Ostukraine“, „Deutsch-Ukrainische“ usw.

7.5.3 hSBM

Der Code zum hSBM-Verfahren ist von Martin Gerlach in Git Hub [35] hinterlegt. Zusätzlich benötigt man noch das Peixoto graph tool [39]. Von Gerlach wurde ein Jupyter-Notebook-Beispielcode hinterlegt, in welchem alle wichtigen Schritte in einer übersichtlichen Weise zusammengefasst sind. Eine Besonderheit in diesem Verfahren ist, dass wir

kein von uns erstelltes Netzwerk benötigen. Wir können den Dokumenten-Korpus einfach importieren. In diesem Verfahren wird ein bipartites Netzwerk aus Dokumenten und Wörtern und kein Kookkurrenznetzwerk erzeugt. Die Erstellung des Netzwerks ist ebenfalls Teil der Implementierung von Gerlach. Somit benötigen wir in unserer Betrachtung auch kein Gewicht a oder Gewicht b. Es ist ein Wert anzugeben, wie viele Wörter pro Topic ausgegeben werden sollen. Analog wie bei den anderen Verfahren haben wir diesen auf 200 gesetzt. Zudem werden hier ganz automatisch die generierten Wortlisten in den Topics nach Common Word Ranking sortiert. Das heißt, dass die häufigsten Wörter oben stehen. Dies war die eigentliche Motivation das Common Word Ranking mit zu berücksichtigen. Somit haben wir die Topics auch hier neben der Common-Word-Sortierung in einer Bayes-Sortierung ausgegeben. Dies ermöglicht einen Vergleich mit dem Louvain- und Infomap-Verfahren.

Vergleich verschiedener Durchläufe

Auch hier ist es wieder von Interesse, analog zur Louvain-Auswertung, die Topic-Ausgabe für zwei verschiedene Durchläufe miteinander zu vergleichen. Dieser Vergleich ist in den Histogrammen 21 dargestellt. Hier lässt sich erkennen, dass sich die Wörter des Topics des ersten Durchlaufs meistens nicht nur auf ein Topic im zweiten Durchlauf verteilen.

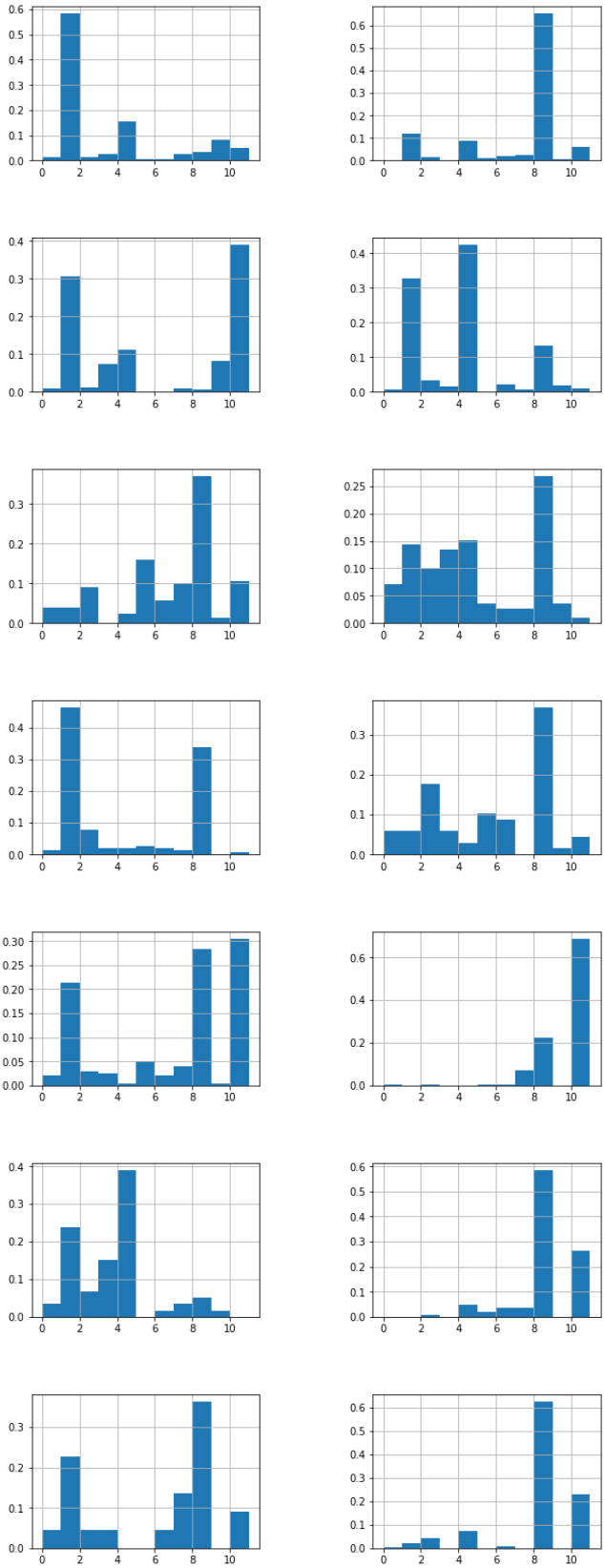

Abbildung 21: Vergleich Topic-Ausgabe für zwei Durchläufe auf Ebene 4, hSBM

Im Folgenden untersuchen wir in Tabelle 16 die Modularität für fünf verschiedene Durchläufe.

Ebene	0	1	2	3	4
Modularität i=0	0.223	0.243	0.257	0.275	0.278
Modularität i=1	0.214	0.236	0.255	0.284	0.281
Modularität i=2	0.22	0.239	0.258	0.287	0.274
Modularität i=3	0.218	0.247	0.262	0.278	0.277
Modularität i=4	0.223	0.243	0.257	0.275	0.278

Tabelle 16: Modularität für fünf Duchläufe - hSBM

Die Qualität der Ergebnisse im Sinne der Community-Identifikation sind vergleichbar. Im Folgenden nehmen wir einen beliebigen Durchlauf für die Auswertung der Ergebnisse, wobei allerdings aus den Ergebnissen des Histogramms anzumerken ist, dass die konkreten Topics nicht so stabil sind.

Berechnung der Modularität

In Tabelle 17 sind die Modularitätswerte für alle Ebenen für den gewählten Durchlauf zusammengefasst.

Ebene	0	1	2	3	4
Modularität	0.223	0.243	0.257	0.275	0.278

Tabelle 17: Modularität - hSBM, KA 2017

Ausgabe der Wortlisten der Topics - Bayes

In diesem Durchlauf werden fünf Ebenen mit jeweils einer Topic-Einteilung ausgegeben. Das sind mehr als bei der Louvain- und Infomap-Ausgabe. Die Tabelle 18 stellt eine Übersicht der Topic-Anzahl pro Ebene zusammen:

Ebene	0	1	2	3	4
Anzahl Topics	4568	2189	778	174	13

Tabelle 18: Topic-Anzahl pro Ebene - hSBM, KA 2017

Die Ausgabe enthält auf den Ebenen $0 - 3$ viele Topics, die nur ein Wort enthalten. Interessant könnte hier beispielsweise der Versuch sein alle Topics mit wenigen Wörtern auszusortieren.

Schauen wir uns zunächst die 13 Topics auf gröbster Ebene genauer an. Diese sind in der folgenden Tabelle zusammengestellt.

Topic 0
Neonazi BAMF Bundeswehr antisemitisch Asylsuchende Asylsuchenden Eritrea Abs. motivieren BMVg Abschiebung AsylG Iran Politisch Asylverfahren
Topic 1
Siemens Klimakrise Shell Elektromobilität KGaA warm Artenvielfalt Bank Impfstoff Adam Wintershall Bauer Investor Teilprojekt Körper
Topic 2
Hasskriminalität Afghanistan Oberthema Berufsschule Asylunterkünfte Gymnasium Tatdatum Syrien Ostukraine Tatzeit Irak Kosovo StGB Roma Familiennachzug
Topic 3
libyschen türkisch DITIB Europol Kriegswaffen libysche Gewehr Drohne G20-Gipfel Rüstungsgütern Rüstungsgüter kurdisch Einheitsregierung Islam russische
Topic 4
Bundesnetzagentur Flug Liegenschaften Zustand LuFV Marine Bahnhof Digitalisierung Tonne gesellschaftliche Vergütung Teilhabe rassistisch Innovation Erkrankung
Topic 5
palästinensisch israelisch israelische Regime Police Typ Übung OSZE Demokratie BPOL Kriegsverbrechen Minderheit Korruption Israel religiös
Topic 6
Tier Konzert Bundesverwaltung ThyssenKrupp Globale Art Weltgesundheitsorganisation Defence hungern italienisch Bundespolizei Geber Fragestellerinnen Sigmar Vertreterinnen
Topic 7
belgisch Verbundprojekt belgische Entwicklungsländern Compact rechtswidrige BMEL BMBF psychosozial BMUB Programmierung rechtsextrem Wiederverwendung Wolf Einfuhr
Topic 8
Daimler Audi Volkswagen Verbraucherinnen Albanien AufenthG BImA BVWP Jobcentern Automobilindustrie Ostsee Quartal Verbraucher Finanzierungsvereinbarung bahnen

Topic 9
Marode Breitbandanschluss Durchschnittsalter Schneller Eisenbahnbrücken Breitbandversorgung Zustandskategorie Mbit Streckenkilometer Schieneninfrastruktur Schienenwege Kreisfreie Netzsegmentes Breitbandanschlüssen Fahrverkehr
Topic 10
EUNAVFOR Aufrüstung EUCAP EUBAM Nordirak Provinz Riad Libya Jesiden Inherent Libyen Haftar Haiti Lehrgang afrikanisch
Topic 11
Cannabis CETA Verkehrsstationen Arzneimittel unseriöse Bahnsteige BaFin Patientin Patient BfArM Versicherungspflicht Rentenversicherung UN-Behindertenrechtskonvention BZgA Rente
Topic 12
Snowden Ladepunkten Ladepunkte Unionsbürger Stickstoffdioxid Zwischenlager Stickoxid Edward Gorleben BImSchV Beschäftigungsbedingungen Freistaates Betreuungsgericht Ablehnungsquoten Grenzwert
Topic 13
Tihange Doel Atomkraftwerk Atomkraftwerken FANC Wasserstoffflocken Fukushima BauGB URENCO Uran radioaktiv Übungsflüge Einheitspartei Dopingopfer-Hilfegesetz DOHG

Auf diese Topic-Liste werden wir im Analyseschritt Topics vs. Klassifikationen mit der Darstellung der Heat Map eingehen. Zunächst erfolgt die Berechnung der Modularität und die Analyse der Topic-Verteilung innerhalb der Dokumente.

Topic-Verteilung innerhalb der Dokumente - Bayes

Es erfolgt in Tabelle 19 eine Analyse der Topic-Verteilung für eine Stichprobe der Kleinen Anfragen Dokumente auf der gröbsten Ebene:

	1811833	1900334	1812335	1812046
0	0.073	0.211	0.077	0.038
1	0.032		0.038	0.002
2	0,037	0.049		0.019
3	0.436	0.439	0.086	0.002
4	0.042		0.129	0.233
5	0.134	0.13	0.019	0.023
6	0.109	0.041	0.392	0.004
7	0.035	0.033	0.148	0.013
8	0.028	0.098	0.024	0.115
9	0.027		0.02	0.523
10	0.03		0.048	
11	0.013		0.01	
12	0.003		0.01	0.029
$\sum$	1	1	1	1

Tabelle 19: Ausgabe vier Dokumente + Topic-Verteilung - hSBM, KA 2017

Wie bereits im selben Schritt bei der Modularitätsoptimierung erwähnt, ist es gut, wenn pro Dokument nur wenige, wichtige Topics vorkommen. Uns interessiert es nicht so sehr wenn lediglich 1 % der Wörter eines beliebigen Topics darin vorkommen. Dies ist in der Tabelle 19 allerdings häufig der Fall. So lässt sich in der Tabelle beispielsweise erkennen, dass das Dokument 1811833 viele Topics mit weniger als 10 % beinhaltet. Da wir uns aber für die starken Topics interessieren, führen wir wieder die 33% Schranke ein.

	1811833	1900334	1812335	1812046
3	0.436	0.439		
6			0.392	
9				0.523

Tabelle 20: Ausgabe vier Dokumenten + Topic-Verteilung 33% - hSBM, KA 2017

Wenn wir einen Vergleich der hier erhaltenen Ergebnisse (vgl. Tabelle 20) und der Ergebnisse der Dokumentenverteilung mit der Louvain-Methode (vgl. Tabelle 13) zulassen, so können wir bei Louvain ablesen, dass die maximalen Auftrittswahrscheinlichkeiten der Topics immer über 75 % liegen. In den hSBM-Ergebnissen liegen die maximalen Auftrittswahrscheinlichkeiten der Topics in den Dokumenten um die 40 % (vgl. Tabelle 20).
Wenn wir ein Ergebnis auf einer feineren Ebene z.B. Level 1 betrachten, so stellen wir fest, dass die Verteilung der Topics in den Dokumenten noch breiter gestreut ist. Außerdem

können vielen Dokumenten mit Berücksichtigung der 33 % Schranke keine Topics mehr zugeordnet werden.

Topics vs. Klassifikationen - Bayes

Da vielen Dokumenten auf feineren Ebenen, $0-2$, mit Berücksichtigung der 33% Schranke gar kein Topic zugeordnet wurde, konnten auch keine Heat Maps auf diesen Ebenen erzeugt werden. In der folgenden Abbildung ist eine Heat Map row auf der gröbsten gefundenen Ebene dargestellt.

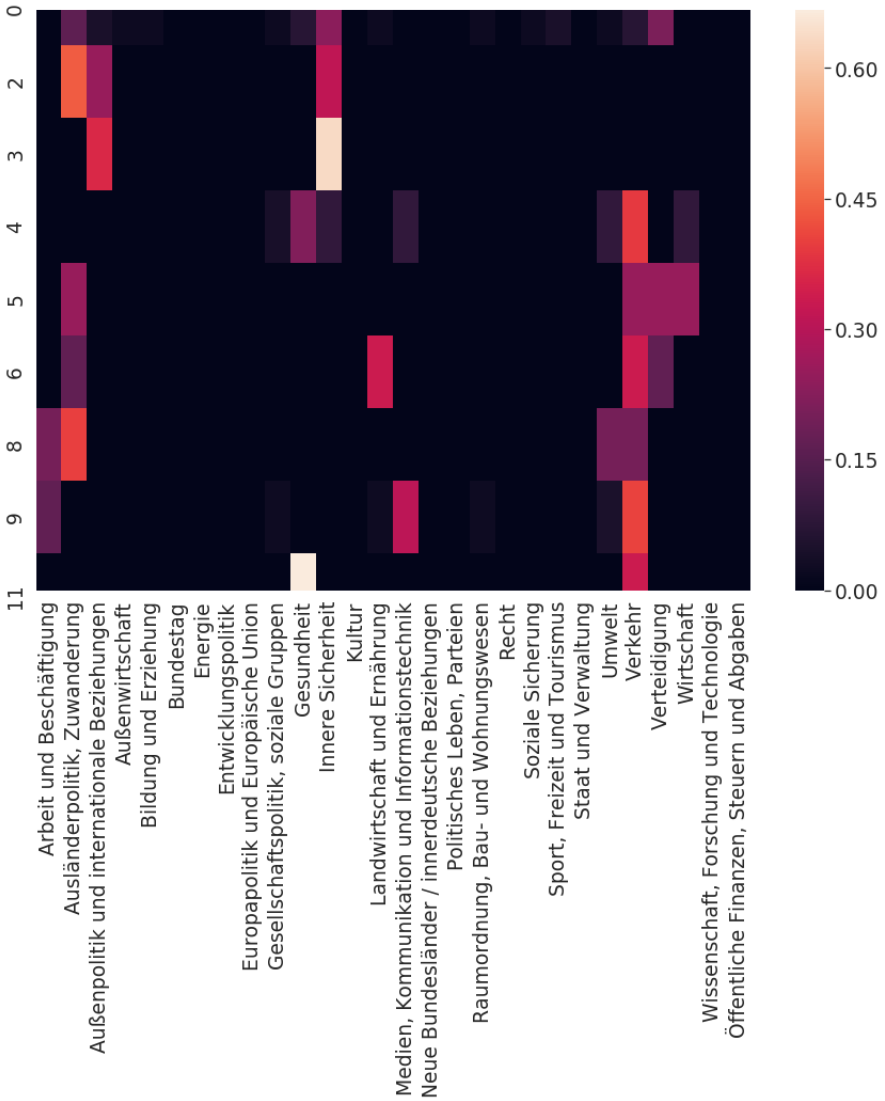

Abbildung 22: Heat Map row - hSBM, Bayes

Vergleichen wir die Ausgabe mit der eben generierten Topicliste: In jedem Topic sind verhältnismäßig viele Sachgebiete relevant. Einigen Topics wurden gleiche Sachgebiete

zugeordnet. Manche spezifizieren sich aber dann indem ein zusätzliches Sachgebiet hinzugenommen wird. So lässt sich sagen, dass Topic 3 um „Innere Sicherheit“ und „Außenpolitik“ geht, Topic 2 ebenfalls um beide Themen aber auch zusätzlich um „Ausländerpolitik“. Topic 0 lässt sich hingegen durch die Sachgebiete „Ausländerpolitik“, „Innere Sicherheit“ und zusätzlich „Verteidigung“ beschreiben. Alles sind sehr ähnliche Themen, aber keinem wurden die gleichen Gebiete zugeordnet. In den Topics 4,5,6,8,9 und 11 ist „Verkehr“ relevant. Jedoch variieren von Topic zu Topic die zusätzlichen Themen.

Ausgabe der Wortlisten der Topics - Common Word

An dieser Stelle möchten wir uns noch einmal das Common Word Ranking für die hSBM-Wortlisten genauer ansehen. In der Bibliothek von Gerlach wurde dies als vorgeschlagene Wortsortierung mit implementiert. Stellen wir die erzeugten Wortlisten wie im Louvain-Kapitel gegenüber:

Klassisikation Heat Map	Bayes	CommonWord
Topic 0: Ausländerpolitik, Innere Sicherheit, Verteidigung	Neonazi BAMF Bundeswehr antisemitisch Asylsuchende Asylsuchenden Eritrea Abs. motivieren BMVg Abschiebung AsylG Iran Politisch Asylverfahren	Euro Person Europäische europäisch deutsch Projekt Absatz Union Land Million Bundeswehr politisch Bundestagsdrucksache Anlage Kommission
Topic 2: Ausländerpolitik, Außenpolitik, Innere Sicherheit	Hasskriminalität Afghanistan Oberthema Berufsschule Asylunterkünfte Gymnasium Tatdatum Syrien Ostukraine Tatzeit Irak Kosovo StGB Roma Familiennachzug	Flüchtling Straftat Türkei Syrien Nation Irak Vereinte Republik russisch polizeilich humanitär StGB Ägypten Afghanistan Phänomenbereich
Topic 3: Außenpolitik, Innere Sicherheit	libyschen türkisch DITIB Europol Kriegswaffen libysche Gewehr Drohne G20-Gipfel Rüstungsgütern Rüstungsgüter kurdisch Einheitsregierung Islam russische	militärisch Vorbemerkung ausländisch Juli türkisch Waffe Mitglied Ermittlung Inwieweit Nachrichtendienste zivil Sicherheitsbehörden Europol Seite Geheimdienst

Vergleichen wir die Wortlisten miteinander: In Topic 0 in der Common-Word-Liste stehen viele europäische Begriffe. Als Verteidigungsthema ist der einzige auftauchende Begriff „Bundeswehr“. Möglicherweise liegt hier der Augenmerk darauf, dass Sicheheits- und Ver-

teidigungsthemen auf europäischer Ebene diskutiert werden. Topic 2 beinhaltet in beiden Sortierungen ähnliche Begriffe. Und in Topic 3 sind in Common-Word-Sortierung allgemeinere Begriffe mit aufgenommen. So z.B. „Vorbemerkung“, „Mitglied“ und „inwieweit“. Diese sind in der Bayes-Liste nicht vorhanden.

7.6 Elib - Experimentelle Ergebnisse der Community-Detection-Verfahren

Hier erfolgt analog zum Kapitel 7.5 eine Analyse der Ergebnisse des Elib-Korpus, indem die folgenden Schritte durchlaufen werden:

- Vergleich verschiedener Durchläufe
- Berechnung der Modularität/Codelänge
- Ausgabe der Wortlisten der Topics
- Topic-Verteilung innerhalb der Dokumente
- Topics vs. Klassifikationen

In Kapitel 7.1 wurden die drei aufbereiteten Versionen des Elib-Korpus zusammengestellt. Zunächst erfolgt eine Überlegung welche Version wir für die Auswertungen betrachten möchten. Welche Version liefert uns die besten Topics? Dafür haben wir zunächst für die drei Versionen in unterschiedlichen Durchläufen die Modularität auf der gröbsten gefundenen Ebene in der Tabelle 21 zusammengetragen.

	Elib condensed	Elib condensed a2	Elib condensed a3
i = 0	0.117	0.163	0.283
i = 1	0.118	0.161	0.279
i = 2	0.118	0.163	0.286
i = 3	0.117	0.163	0.286
i = 4	0.117	0.164	0.286

Tabelle 21: Modularität für verschiedene Elib-Korpora auf gröbster Ebene

Vergleichen wir die verschiedenen Durchläufe miteinander, so kommen wir zu dem Entschluss, dass die Modularität für die Elib condensed-Version den schlechtesten Modularitätswert liefert. Die Elib condensed a2-Version liegt mit dem Ergebnis im Mittelfeld und die Elib condensed a3-Version weist mit Abstand die beste Modularität auf.

Nun stellt sich die Frage, wie sich Topic-Ausgaben der Versionen a2 und a3 zueinander verhalten. Verschmelzen die Topics des Elib condensed a3-Korpus in Topics der Elib condensed a2-Version?
In Abbildung 23 ist ein Histogramm zum Vergleich der Topics aus a2 und a3 abgebildet.

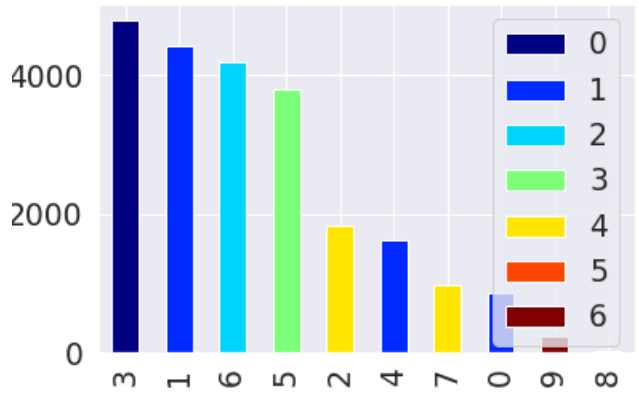

Abbildung 23: Vergleich Topics a2 und a3 - Louvain

Die Topics aus a2 wurden farblich hinterlegt. Das heißt dunkelblau steht für für Topic 0, kaiserblau für Topic 1, türkis für Topic 2 usw. Die Topics aus a3 sind auf der x-Achse abgebildet. Ein Topic wurde hier nur abgebildet, wenn die Wörter des Topics häufiger als 33 % im jeweiligen Dokument vorkommen. Da die Topics $7 - 10$ aus a2 und Topics $10 - 31$ aus a3 zu wenige Wörter enthalten, werden sie nicht im Histogramm abgebildet. Alle Topics aus a3 verschmelzen zu Topics in a2. So wird beispielsweise das Topic 0 in dunkelblau aus a2 zu Topic 3 in a3. Topic 1 aus a2 wird zu den Topics $1, 4$ und 0 in a3. Das heißt also die Topics aus a2 spalten sich in der a3-Version in mehrere Topics auf. Genauer werden wir dies an einer späteren Stelle mit einem Heat-Map-Vergleich beleuchten.
Zusammengefasst bedeutet das, dass die a3-Version mehr Topics als die a2-Version auf der gröbsten Ebene generiert und die Topics aus a3 in den Topics von a2 verschmelzen. Sind wir also an mehr Topics interessiert, ziehen wir a3 zur Auswertung heran. Sind wir aber an weniger zusammengefassten Topics interessiert, schauen wir uns a2 genauer an.
Das gerade beschriebene Ergebnis und der zuvor gegebene Überblick der Modularitäten hat zur Folge, dass wir uns im Teil der Modularitätsoptimierung auf die Ergebnisse der Elib condensed a2- und a3-Versionen beschränken möchten. Im darauffolgenden Teil der Infomap- und hSBM-Auswertung werden wir ausschließlich die a3-Version betrachten.

7.6.1 Modularitätsoptimierung

Es findet die gleiche Bibliothek wie in 7.5 Anwendung. In diesem Kapitel werden die Elib Versionen a2 und a3 betrachtet.

Vergleich verschiedener Durchläufe

Zunächst ist es wieder interessant, das Zusammenauftreten der Wörter in den Topics auf gröbster Ebene zu analysieren. Wir haben uns dazu entschieden den Elib condensed a2-Version auszuwerten, da diese Version weniger Topics enthält und somit an dieser Stelle besser zur Veranschaulichung geeignet ist. Es wurden alle Wörter der Topics in die Auswertung mit einbezogen. Für zwei Durchläufe ist in den Histogrammen 24 das Ergebnis abgebildet. In allen Fällen kommen mehr als 80% der Wörter eines Topics des ersten Durchlaufs auch im selben Topic des zweiten Durchlaufs vor. Die unteren fünf Histogramme zeigen, dass sogar alle Wörter des Topics im ersten Durchlauf auch in selben Topic des zweiten Durchlaufs liegen.

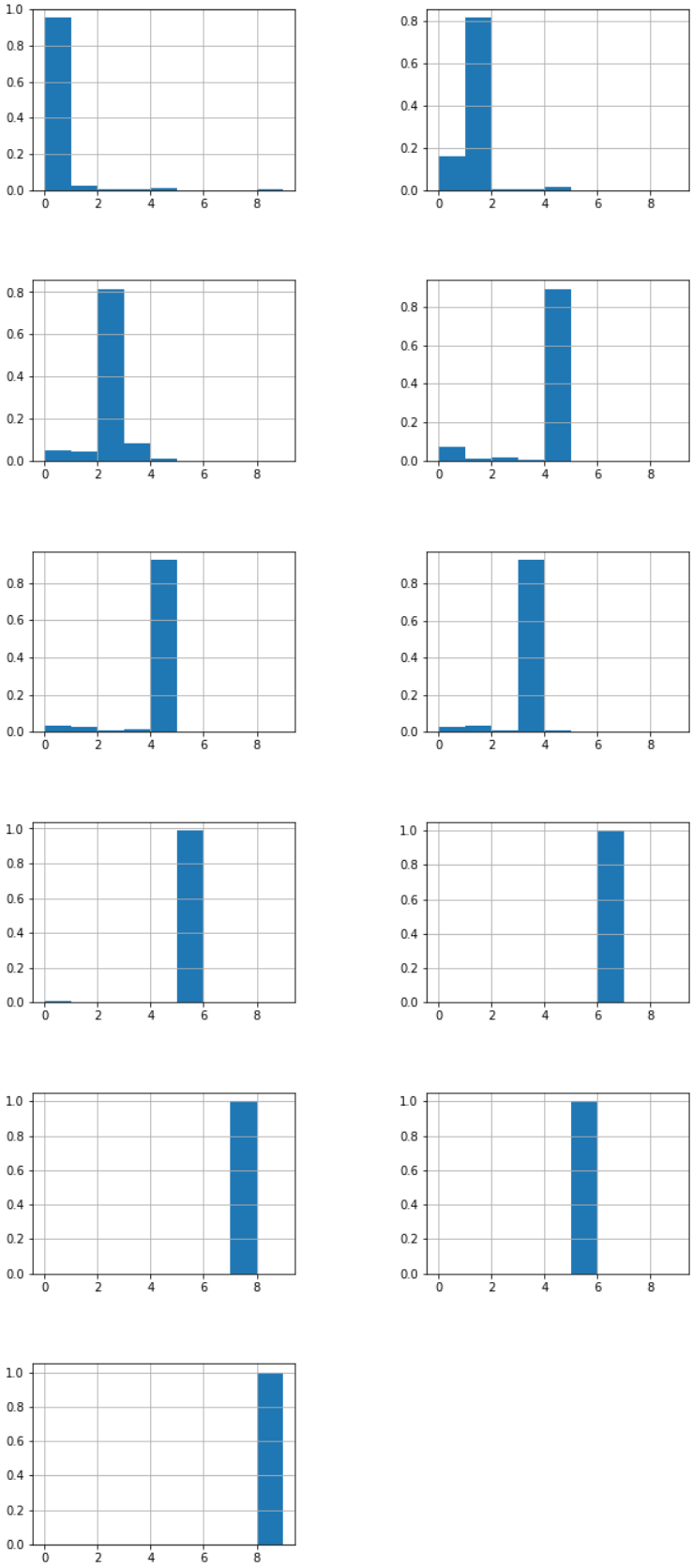

Abbildung 24: Topic Vergleich für zwei Durchläufe - Louvain

Der Modularitätsvergleich (vgl. Tabelle 21) sowie die Topic Ausgabe für zwei verschiedene Durchläufe in Form der Histogramme ermöglicht es an dieser Stelle einen beliebigen Durchlauf zu wählen.

Berechnung der Modularität

In Tabelle 22 sind die Modularitätswerte für diesen Durchlauf gegenübergestellt:

Ebene	0	1	2
Modularität a2	0.157	0.161	
Modularität a3	0.271	0.286	0.286

Tabelle 22: Modularität für a2, a3 - Elib, Louvain, Gewicht a

Ausgabe der Wortlisten der Topics - Bayes, Gewicht a

Zunächst betrachten wir in diesem Schritt die a2-Version: Es werden zwei Level mit Topic-Listen generiert. Auf Ebene 0 gibt es 123 Topics und auf der Ebene 1 werden 11 Topics ausgegeben. Schauen wir uns hier zunächst ausgewählte Topics auf der feineren Ebene an:

Topic 3
aerogel CMAS thermoelectric Cost Sulfur sulfur hydride Techno pipe ventilation Magnesium SOFC electrolysis biofuel diamond
Topic 6
Philae DESIS Venus sail InSight MASCOT VIRTIS Mercury comet quadrangle Titan MERTIS outburst Ceres euro
Topic 7
sleep bone muscle hypoxia LBNP athlete blind baroreflex Sleep urine pain deprivation tendon immobilization sympathetic

In Topic 3 tauchen Begriffe zur Werkstoffforschung, Batterien und Treibstoffen auf. Topic 6 hingegen beinhaltet spezifische Begriffe, wie Raum-/Landesonden, Planeten, Experimente der Weltraumsonden oder der ISS. Somit beschreibt Topic 0 ein Topic zur Raumfahrt. Und Topic 7 beinhaltet Begriffe des menschlichen Körpers und der Medizin. Dies lässt sich als Raumfahrtmedizin-Topic klassifizieren.
Im Folgenden werden die 15 besten gerankten Wörter nach Bayes der Topics $0 - 5$ auf der gröbsten Ebene dargestellt:

Topic 0
LDACS SpaceLiner EDEN Innovative Servicing provenance APNT Wireless ESSEX ReFEx AVANTI SWIM drone SpaceWire MBSE
Topic 1
contrail Aeolus slum TanDEM EnMAP GBAS ozone cirrus CTIPe MERLIN GNSS aquaculture Snow Coastal Radiometric
Topic 2
microparticle STRUCTURAL Crash Uncertainty Effect peridynamic liner Lining Aeroelastic Unsteady rudder combustor TESTING Combustion Combustor
Topic 3
aerogel CMAS thermoelectric Cost Sulfur sulfur hydride Techno pipe ventilation Magnesium SOFC electrolysis biofuel diamond
Topic 4
sleep spore bone MASE subtili biofilm DOSIS muscle hypoxia Columbus lichen Raman LBNP Cold athlete
Topic 5
Philae DESIS Venus sail InSight MASCOT VIRTIS Mercury comet quadrangle Titan MERTIS outburst Ceres euro

Für einen anschließenden Vergleich mit Elib condensed a3 möchten wir nun die Ergebnisse der Topics $0 - 9$ von a3 auf der gröbsten Ebene wiedergeben:

Topic 0
LDACS GBAS CTIPe GNSS APNT Wireless SiGe Ionospheric ionospheric SLAM Inland ALOHA EGNOS Jamming Robust
Topic 1
Philae DESIS Venus VIRTIS slum TanDEM Mercury EnMAP comet quadrangle Titan MERTIS aquaculture Snow Coastal
Topic 2
spore sail InSight MASCOT EDEN MASE subtili Servicing biofilm DOSIS AVANTI euro boom Relative Columbus
Topic 3
Innovative provenance SWIM drone SpaceWire MBSE boarding Speech SMILE bird Situation Bluetooth Haptic PHAROS Thematic
Topic 4
contrail Aeolus ozone cirrus MERLIN SCIAMACHY DEEPWAVE Dioxide MIPAS lidar Methane TROPOMI Turbo cm-1 soot
Topic 5
aerogel CMAS thermoelectric Cost Sulfur sulfur hydride Techno pipe ventilation Magnesium SOFC electrolysis biofuel wick
Topic 6 SpaceLiner microparticle STRUCTURAL Crash Uncertainty Effect peridynamic ReFEx liner Lining Aeroelastic Unsteady rudder combustor EAGLE
Topic 7 sleep bone muscle hypoxia LBNP athlete baroreflex Sleep urine pain deprivation Density immobilization sympathetic cybersickness
Topic 8 freeform SPDT Nussbaumer JExTRA Optische Walter2 Risse1 Peschel1 Ingo Funke Damm1 Christoph Sebastian2 Matthias Krutz2
Topic 9 Entwicklung Fertigung eine einer Untersuchung Jülich Implementierung Projekt werden Bewertung Towers reduktion offenen Verwendung einem

Topics vs. Klassifikationen - Bayes, Gewicht a

In diesem Teil der Auswertung möchten wir die Ergebnisse der Versionen a2 und a3 auf der gröbsten Ebene mithilfe von Heat Maps vergleichen. Und mithilfe der Maps und anhand des Histogramms 23 verdeutlichen, welche Topics aus a3 zu welche Topics aus

a2 verschmelzen. Zunächst wird in Abbildung 25 die Heat Map col der Elib condensed a2-Version dargestellt:

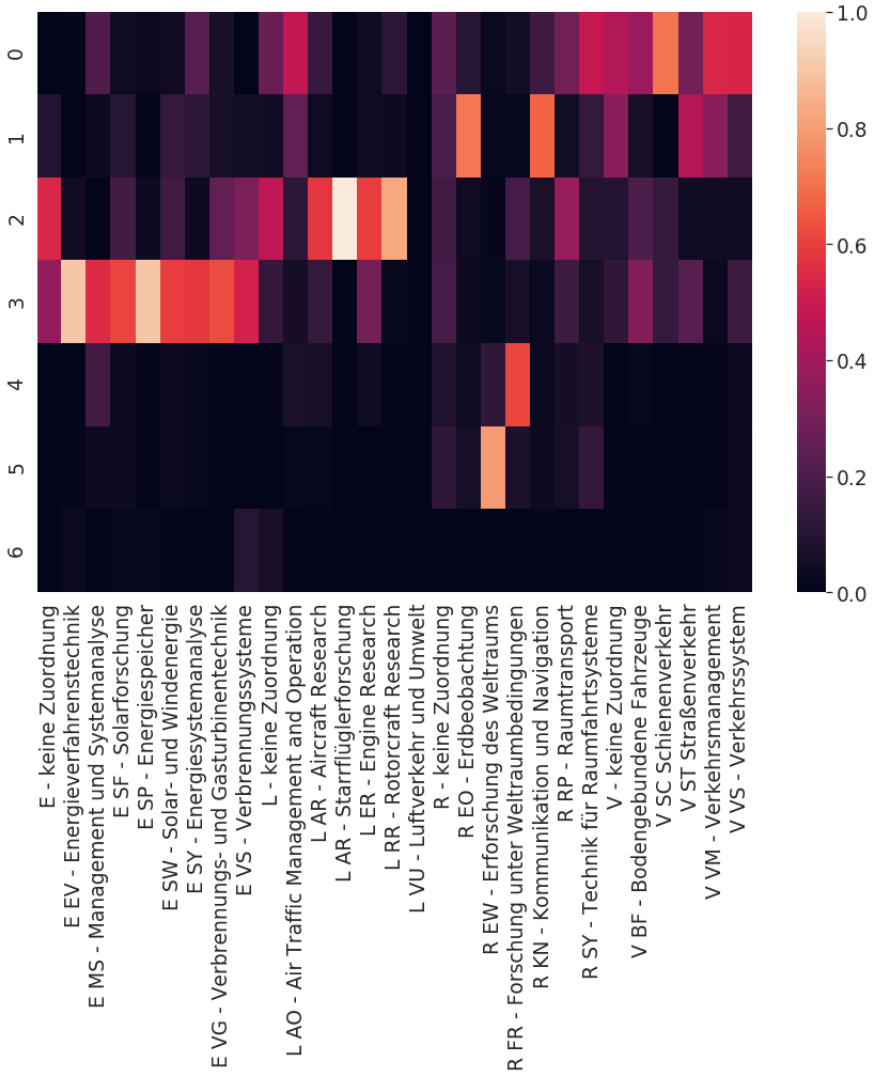

Abbildung 25: Heat Map col - a2, Louvain, Bayes, Gewicht a, Level 1

Wenn wir herausfinden möchten, welches Topic im DLR-Forschungsgebiet „Energie“ sehr stark ist, dann suchen wir in der Heat Map das Topic mit den meisten hellen Balken im Bereich „Engergie“. Dies ist in Abbildung 25 Topic 3. Im Forschungsgebiet „Luftfahrt“ ist Topic 2 stark vertreten. Und im Bereich „Verkehr“ das Topic 0. Das Gebiet „Raumfahrt“ teilt sich hier in mehrere Topics auf. So lässt sich erkennen, dass Topic 5 vor allem von der „Erforschung des Weltraums“ handelt und Topic 4 von der „Forschung unter Weltraumbedingungen“. Und auch Topic 1 ist mit den Bereichen „Erdbeobachtung“ sowie „Kommunikation und Navigation“ ein Thema der „Raumfahrt“.
Für einen Versionsvergleich wird im Folgenden die Heat Map in Abbildung 26 für a3 abgebildet.

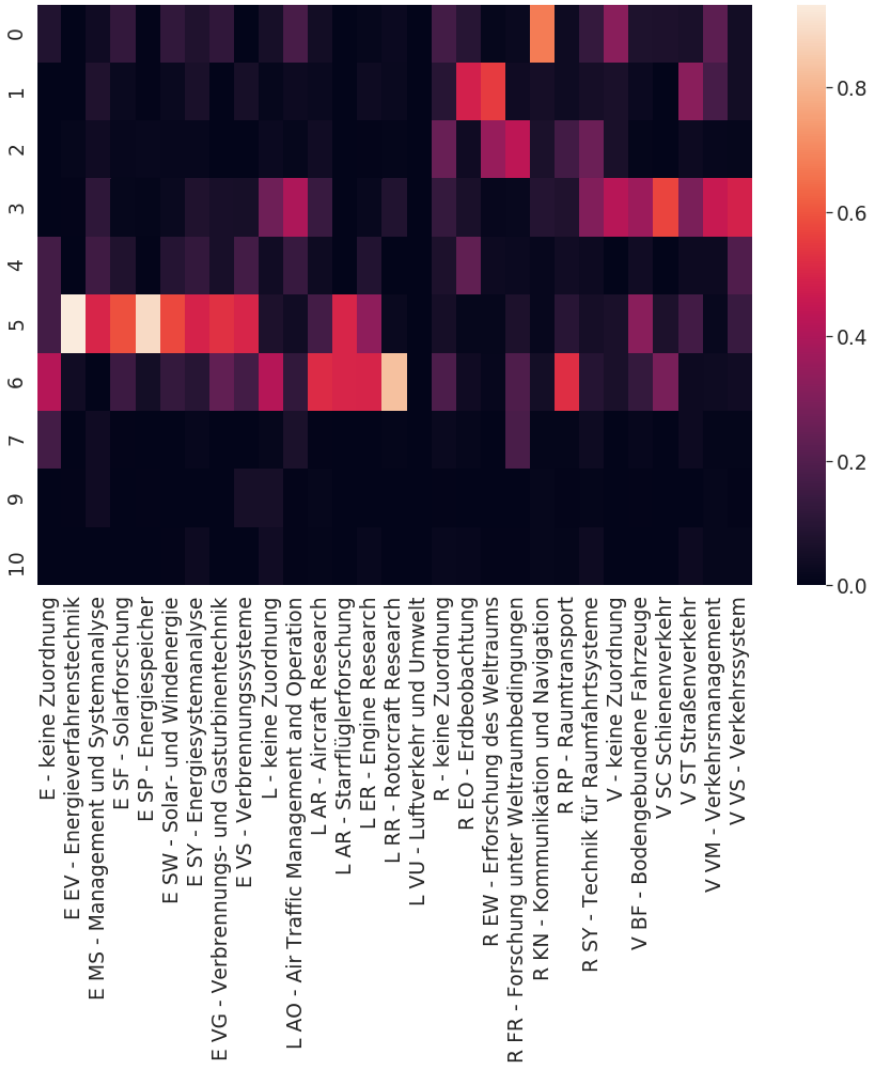

Abbildung 26: Heat Map col - a3, Louvain, Bayes, Gewicht a, Level 2

Schauen wir uns nun den Übergang der Topics im zuvor wiedergegebenen Histogramm 23 an:

Topic 0 aus a2, nach Interpretation der Heat Map in Abbildung 25 ein Topic aus dem Bereich „Verkehr“, geht über zu Topic 3 in a3 (vgl. Abbildung 23). Ein Abgleich mit der Heat Map in Abbildung 26 liefert uns, dass „Verkehr“ hier tatsächlich am stärksten in a3 vertreten ist. Schauen wir uns analog Topic 1 aus a2 an. Die Bereiche „Erdbeobachtung“ sowie „Kommunikation und Navigation“ sind in dem Topic relevant. Das Histogramm in Abbildung 23 gibt uns an, dass Topic 1 sich in a3 zu den Topics $1, 4$ und 0 aufspaltet. Schauen wir uns dafür die Heat Map von a3 (vgl. Abbildung 26) genauer an: Das Forschungsgebiet „Erdbeobachtung“ teilt sich in die Topics 1 und 4 auf. Das Forschungsgebiet „Kommunikation und Navigation“ der „Raumfahrt“ ist in Topic 0 stark vertreten. Zusammengefasst lässt sich also sagen, dass sich die Topics aus a2 in Topics aus a3 weiter

aufspalten und somit einen zusätzlichen Erkenntnisgewinn der Topics geben können.
Im Folgenden werden wir der Übersichtlichkeit halber nur noch die a3-Version berücksichtigen und auf die Darstellung der Ergebnisse von a2 verzichten:
Des Weiteren werden hier die Ergebnisse für Louvain mit Gewicht b dargestellt. Und zwar werden im Folgenden die berechnete Modularität, generierte Wortlisten und die ausgegeben Heat Maps dargestellt.

Berechnung der Modularität - Bayes, Gewicht b

Die Modularitätswerte sind für die verschiedenen Ebene in der Tabelle 23 dargestellt:

Ebene	0	1	2
Modularität	0.322	0.362	0.363

Tabelle 23: Modularität a3 - Elib, Louvain, Gewicht b

Ausgabe der Wortlisten der Topics - Bayes, Gewicht b

Wenden wir den Louvain-Algorithmus auf den mit Gewicht b konstruierten Graphen an, so erhalten wir drei Ebenen mit 315 Topics auf Ebene 0, 30 auf Ebene 1 und 28 auf Ebene 2. Für die Ergebnisse betrachten wir hier die Ausgabe der Topics auf der gröbsten Ebene:

Topic 2
spore EDEN MASE subtili biofilm DOSIS euro Columbus lichen cares Raman Heritage libs Cultural sterilization
Topic 5
DESIS slum TanDEM EnMAP provenance aquaculture Snow Coastal Radiometric eruption TOPS Building Acquisitions dsm rice
Topic 6
Philae InSight MASCOT VIRTIS Mercury comet quadrangle Titan MERTIS outburst Ceres SMILE Rosetta COSAC Situ
Topic 7
LDACS GBAS Servicing GNSS APNT Wireless ReFEx AVANTI SiGe SpaceWire SLAM Relative Inland ALOHA CROPIS

Topics vs. Klassifikationen - Bayes, Gewicht b

Betrachten wir nun die Heat Map col in Abbildung 28 und vergleichen diese mit der soeben generierte Topic-Ausgabe, so stellen wir fest, dass es sich bei allen um Raumfahrt-Topics handelt, die sich zu unterschiedlichen Topics aufgespalten haben.

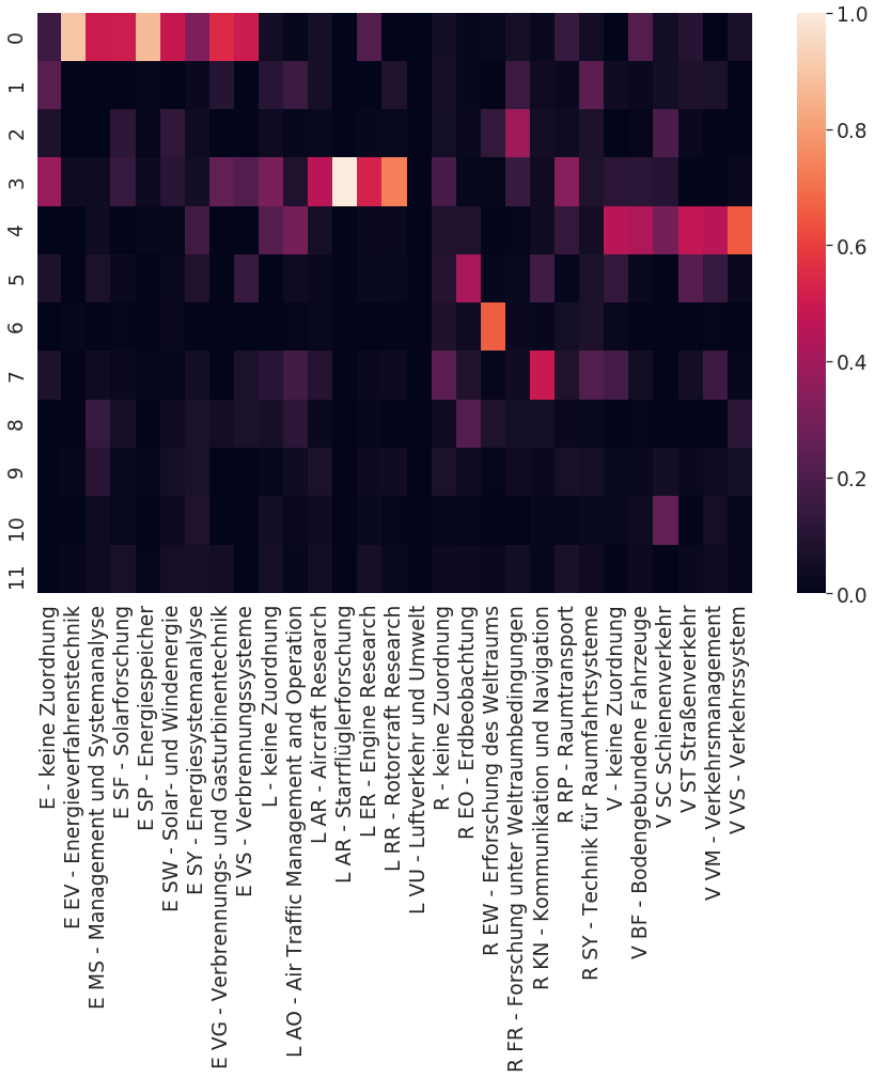

Abbildung 27: Heat Map col - a3, Louvain, Bayes, Gewicht b, Level 2

Topic 2, 5, 6 und 7 sind klare „Raumfahrt“-Topics. In Topic 2 ist nach Interpretation der Heat Map vor allem „Forschung unter Weltraumbedingungen“ relevant. Lesen wir die gefundenen Wörter in der obigen Worttabelle zu Topic 2 ab, stellen wir fest, dass hier vor allem Projekte wie „EDEN“ oder „MASE“ und „DOSIS“ auftreten. Dies sind alles DLR-Projekte, die „Forschung unter Weltraumbedingungen“ betreiben. Beispielsweise werden im Projekt „EDEN“ auf der ISS Pflanzen gezüchtet. Oder im Projekt „DOSIS“ möchte man herausfinden, wie viel Strahlung ein Astronaut verträgt. Topic 5 ist über „Erdbeobachtung“. In der obigen Wortliste steht das Wort „DESIS“ für eine Mission des Earth Observation Center des DLRs. „TanDEM“ steht für ein Radarsatellit und „EnMAP“ wird die erste deutsche optische Erdbeobachtungsmission sein. Gleiches kann man für das Topic 6 über „Erforschung des Weltraums“ und Topic 7 über „Kommunikation und Navigation“ in der „Raumfahrt“ feststellen. Zusammenfassend lässt sich hier also

sagen, dass die Topics der „Raumfahrt“ in diesem Fall thematisch aufgespalten werden können, wohingegen in der Heat Map „Energie“, „Verkehr“ und „Luftfahrt“ separat in Topics auftritt.
Zu diesem Korpus hat die Anwendung des Louvain-Algorithmus gute Ergebnisse geliefert. Die Topics der „Raumfahrt“ wurden zu gut interpretierbaren Topics aufgespalten. Beim Korpus KA 2017 waren wir nicht ganz so angetan von den Ergebnissen, da die Wörter in den Topics dort weiter zusammengefasst wurden und Wörter aus verschiedenen Themen in einem Topic relevant waren. Daher sind wir hier zu dem Schluss gekommen, dass es vermutlich Korpusabhängig ist, ob das Gewicht b gute Ergebnisse liefert.

7.6.2 Map Equation

Hier wird analog zu dem KA 2017-Kapitel der Infomap-Algorithmus auf den Elib-Korpus, nämlich die aufbereitete a3-Version, angewendet.

Berechnung der Modularität/Codelänge

In der Tabelle möchten wir die berechnete Modularität sowie Codelänge der des der Ergebnisse des Louvain-Algorithmus mit Gewicht a und Gewicht b gegenüberstellen.

Ebene	0
Modularität Gewicht a	0.039
Modularität Gewicht b	0.317
Codelänge Gewicht a	12.766
Codelänge Gewicht b	12.255

Tabelle 24: Modularität a3 - Elib, Louvain

Die Topic-Ergebnisse mit Gewicht b weisen sowohl eine höhere Modularität als auch eine niedrigere Codelänge auf.
Aufgrund der hierarchischen Struktur des Verfahrens werden auch hier wieder eine große Anzahl an Topics ausgegeben. Auf die Darstellung dieser möchten wir hier verzichten und auf den Code „Topic Modeling - Louvain und Infomap“ verweisen.

7.6.3 hSBM

Berechnung der Modularität/Codelänge

Ebene	0	1	2	3
Modularität	0.058	0.077	0.088	0.118

Tabelle 25: Modularität a3 - Elib, hSBM

Ausgabe der Wortlisten der Topics - Bayes, Gewicht b

Es wurden Topics auf vier Ebenen generiert. Die Anzahl der Topics auf der feinsten Ebene beträgt 1161 und auf der gröbsten Ebene 58. Im Folgenden möchten wir uns einige ausgewählte Topics mittels Common-Word-Sortierung auf Ebene 2 anschauen:

Topic 30
automation tower assistance attention Tower implication video physiological responsible Working Automation augmented workplace Factors haul
Topic 34
evaluation year version systematic precipitation relationship significant Coupled routine Ocean Physics timely Southern Intercomparison rainfall
Topic 70
travel survey choice freight logistic trip household intermodal residential personal Travel socio preference accessibility private

Topics vs. Klassifikationen - Bayes, Gewicht b

In Abbildung 28 ist die Heat Map col für die Ebene 2 dargestellt.

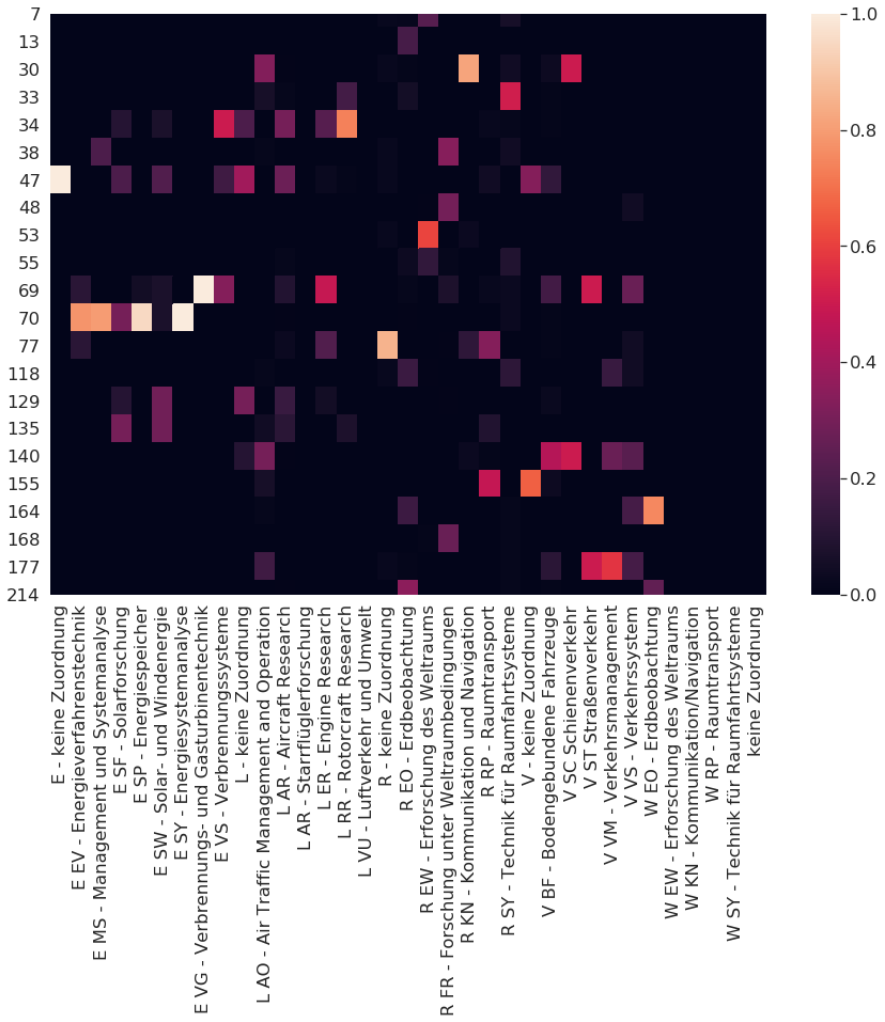

Abbildung 28: Heat Map col - a3, hSBM, Common Word, Level 2

Topic 30 wurde den Fachgebieten „Kommunikation und Navigation“ der „Raumfahrt“, „Air Traffic Management“ der „Luftfahrt“ und „Schienenverkehr“ aus dem Bereich „Verkehr“ zugeordnet. In der obigen Wortliste tauchen Begriffe wie „Tower“, „assistance“, „video“ usw. auf. „Tower“ könnte hier für Flugverkehrskontrollturm stehen und könnte dementsprechend zu den Themen „Kommunikation und Navigation“ als auch „Air Traffic Management“ passen. Allerdings können die anderen Begriffe nicht auf Anhieb mit den Forschungsgebieten in Zusammenhang gebracht werden. Und weder bei den Begriffen des Topics 34 ist ersichtlich, dass es hier um „Luftfahrt“ gehen könnte, noch bei Begriffen des Topics 70, dass hier „Energie“ eine große Rolle spielt.
Zusammengefasst lässt sich beobachten, dass in dieser Heat-Map-Ausgabe keine so schöne Verkehr-, Energie-, Luftfahrt- und Raumfahrt-Einteilung vorliegt wie in den bisher dargestellten Heat Maps des Elib-Korpus.

7.7 Vergleich der Verfahren

In diesem Kapitel erfolgt ein Vergleich der Ergebnisse der verschiedenen Verfahren. Es gibt zwei unterschiedliche Kriterien für einen Vergleich:

1. Graphentheoretisches Kriterium: Wie gut ist das Ergebnis im graphentheoretischen Sinne? Wie gut ist die gefundene Community-Struktur des Graphen? Darauf werden wir im ersten Schritt in Kapitel 7.7.1 eingehen.

2. Semantisches Kriterium: Wie gut eignen sich die gefundenen Wörter für eine Topic-Erkennung? Wie gut passen die Wörter zusammen? Und lässt sich ein Topic für die gefundene Wortliste finden? Hier liefern die Kapitel 7.7.2 und 7.7.3 Antworten.

7.7.1 Modularitätsberechnung

Für eine graphentheoretische Bewertung der Verfahren wird der Modularitätswert berechnet. Dieser gibt an, wie deutlich die gefundene Community-Struktur von einem Zufallsgraphen abweicht. Eine Modularität von 0 sagt aus, dass die Community-Struktur des Graphen nicht besser als eine zufällige Einteilung der Communities ist. Ist der Wert negativ, spricht dies für eine schlechte, beziehungsweise nicht vorhandene Community-Struktur, während die Qualität der Community-Einteilung umso besser ist, je größer die Modularität. Die Verknüpfungen der Wörter im erzeugten Graphen werden in der Berechnung berücksichtigt. In der folgenden Tabelle 26 sind die Modularitätswerte für alle analysierten Verfahren für die KA 2017 mit unterschiedlichen Gewichten zusammengetragen.

KA 2017

Ebene	0	1	2	3	4
Louvain Gewicht a	0.47	0.501	0.502		
Louvain Gewicht b	0.609	0.615	0.615		
Infomap Gewicht a	0.48				
Infomap Gewicht b	0.606	0.606	0.022		
hSBM	0.223	0.243	0.257	0.275	0.278

Tabelle 26: Modularitätsvergleich - KA 2017

Es erscheint sinnvoll, die maximalen Werte pro Verfahren miteinander zu vergleichen. Folglich wird der Infomap Gewicht b-Wert in Ebene 2 nicht mit berücksichtigt. Sowohl Louvain als auch Infomap weisen sehr gute Ergebnisse für Gewicht a auf. Der Wert für

Gewicht b ist sogar noch höher. An dieser Stelle muss allerdings beachtet werden, dass der berechnete Modularitätswert vom konstruierten Graphen abhängt. Da der Graph mit Gewicht b weniger Kanten aufweist, ergibt sich hier ein besserer Modularitätswert. Die Modularität für die Ergebnisse des hSBM-Verfahren beträgt auf gröbster Ebene 0.278 und ist damit niedriger als die Werte der Konkurrenzverfahren.

Elib condensed a3

Für die Elib condensed a3 Version wurden die höchsten Modularitätswerte berechnet. Demnach werden diese in der folgenden Tabelle 27 abgebildet.

Ebene	0	1	2	3
Louvain Gewicht a	0.271	0.286	0.286	
Louvain Gewicht b	0.322	0.362	0.363	
Infomap Gewicht a	0.039			
Infomap Gewicht b	0.317			
hSBM	0.058	0.077	0.088	0.117

Tabelle 27: Modularitätsvergleich - Elib condensed a3

Vergleichen wir die Modularitäten des Elib condensed a3-Korpus, so liefern sowohl Louvain mit Gewicht a und b und Infomap mit Gewicht b einen guten Wert. Die Modularität für Infomap mit Gewicht a ist nur minimal besser als die Modularität eines Zufallsgraphen. Auch der Modularitätswert für hSBM ist nicht konkurrenzfähig mit den Ergebnissen des Louvain- und Infomap-Verfahrens mit Gewicht b.

7.7.2 Einschätzung von Sachexperten

Für den Zweck des Topic Modelings ist es jedoch wichtiger, die Topic-Qualität nicht nach graphentheoretischen Eigenschaften sondern nach ihrem Sachinhalt zu bewerten. Deshalb haben wir zunächst Sachexperten(SE)-Einschätzungen für eine inhaltliche, das heißt semantische, Bewertung einbezogen. Es wurden zum einen Sachexperten(SE)-Bewertungen für die KA 2017 und zum anderen für den Elib-Korpus berücksichtigt.

KA 2017

Hier haben wir Topic-Ergebnisse, generiert durch Louvain Gewicht a, Louvain Gewicht b und hSBM zu den KA 2017, von Sachexperten der Politikwissenschaft bewerten lassen. Die Ergebnisse des Infomap-Algorithmus haben wir an dieser Stelle nicht berücksichtigt, da hier die generierte Anzahl von Topics so groß war, dass eine Bewertung aller Topics

zu aufwändig gewesen wäre. Zunächst möchten wir als ein beliebiges Beispiel zur Veranschaulichung die Bewertung der Wortlisten, generiert durch Anwendung des Louvain-Algorithmus, wiedergeben. Für die Ausgabe der Wortliste wurde Gewicht a und Bayes Ranking gewählt und die mittlere Ebene (Level 1) mit 30 Topics bewertet. Die folgende Tabelle beinhaltet in einer Zelle jeweils zunächst das jeweilige Topic, die Topic-Bezeichnung des ersten Sachexperten, die Bewertung des Topics mit Werten zwischen 1 (sehr schlechtes Topic) und 5 (sehr gutes Topic) des ersten Sachexperten, die Topic-Bezeichnung des zweiten Sachexperten und schließlich in der letzten Angabe die Bewertung des Topics des zweiten Sachexperten. Diese Angaben werden in der Darstellung durch ein Semikolon getrennt. In der jeweils nächsten Zelle wird sodann die zu bewertende Wortliste des jeweiligen Topics wiedergegeben.

Topic 0: Nahostpolitik; 4; Außenpolitk/EU-Außenpolitik; 3
libyschen EUNAVFOR türkisch Arbeitsbesuch ISIS Vorschub Konfrontation palästinensisch schwerbewaffnet EU-Außenpolitik Europol libysche HOME Fortbildungsveranstaltung Autonomiebehörde
Topic 1: Gesundheitspolitik und insbesondere die Diskussion um die Legalisierung von Cannabis für den medizinischen Gebrauch; 5; Gesundheitspolitik ; 4
Cannabis UN-BRK betäubungsmittelrechtlicher Arzneimittel vernachlässigen IfSG Patientin Patient Wirkstoff BfArM Cannabispatientinnen UN-Behindertenrechtskonvention Cannabisblüten BZgA Erreger
Topic 2: Altersvorsorge; 4; Sozialpolitik; 3
Freibetrag Tarifverträgen Arbeitsvolumen Versicherungspflicht Rentenversicherung Versicherungszeiten Arbeitsverträgen Rente Note Warmwasser Verpackungsmüll Statist Sozialversicherungsabkommen Prekäre Patientenberatung
Topic 3: Software-Sicherheit; 4; Cyber; 2
INTCEN EU-Polizeiagentur DPMA Infrastrukturabgabe Antwortteil WLAN-Catchern WLAN-Catcher Stiller Mozilla License Lesser Laufend LGPL Halbjährlich GPLv2
Topic 4: Rechtsextremismus; 5; Rechtsextreme Politisch Motivierte Kriminalität (PMK) & Ausländerpolitik; 3
Hasskriminalität Ausländerfeindliche Überstellungen BÄRGIDA PMK- ausländerfeindlich BeschV Schwerpunktfragen Kriminalität-rechts PEGIDA nichtmuslimischen Islamfeindlichkeit zweitgrößte islamfeindlicher Voigt
Topic 5: Rüstungsexportpolitik; 5; Waffenexporte; 5
Kriegswaffen Gewehr Waffenexporteure EG-Dual-Use-Verordnung Kleinwaffen Rüstungsgüter Sammelausfuhrgenehmigungen Luftangriffen Sammelausfuhrgenehmigung Güterbewegungen Rüstungsexporte Dual-Use-Güter US-Army Sipri Reassurance
Topic 6: Schulpolitik; 4; Werbung der Bundeswehr in Schulen; 2
Schultyp Oberschule Mittelschule Jugendoffiziere Heeresmusikkorps Gemeinschaftsschule Berufskolleg Regelschule Schulzentrum Karriereberatern Durchführungszeitraum unpopulär rechtspopulistischen großangelegte Wohltätigkeitskonzert
Topic 7: Wohnungspolitik; 4; Themenkomplex "Raumordnung"; 4
schrumpfend BBSR BauGB halbstädtischen interkommunale LMBV Kinderbetreuung Estate Dividende Übernachtung Flächenverbrauch Anleger Ballungszentren Ostsee Verbrennungsmotor
Topic 8: Drohnen; 4; Training des Militärs; 2
Zurückziehung Stattgefundene Angefragte Sanitätsdienstliche Nutzungsüberlassung Stellflächen Ausbildungsdurchführung G8-Gipfel bewaffnungsfähigen Amtshilfemaßnahmen Sanitätsdienst Heron US-Marine US-Basen RAVEN

Topic 9: Finanzpolitik im weitesten Sinne; 3; Sensible Aspekte bzw. Skandale rund um finanzielle Themen; 4
CETA Aktueller unseriöse BaFin Telefonwerbung Snowden Dividendenstichtag WpHG Restschuldversicherungen Inkassodienstleistungen Bundesfernstraßen Verbraucherinnen Rüstungsindustrie ÖPP-Projekt Mietpreisbremse
Topic 10: Aktivitäten türkischer Religionsverbände in Deutschland; 5; Rolle islamischer Vereinigungen/des Islam in Deutschland; 3
DITIB Diyanet UETD Maghreb-Staaten Teilvorhaben Religionsgemeinschaft Graue Türk Hakan antisemitisch Unternehmertum Islam Agententätigkeit İslam Desinformation
Topic 11: Verkehrsinfrastrukturen; 5; Infrastrukturpolitik; 5
Marode Durchschnittsalter Eisenbahnbrücken Zustandskategorie KFZ-Verkehr Streckenkilometer Schieneninfrastruktur Schienenwege Qualitätskennzahl Netzsegmentes Ortsumgehung Verkehrsstationen Fahrverkehr leistungsfähige Bahnsteige
Topic 12: Dieselskandal; 5; Dieselskandal; 4
Daimler Audi Dieselgipfel Volkswagen NABU Siemens Opel Abgasskandals Hans-Georg Greenpeace Verbundprojekt Shell Außenwirtschaftsförderung Airbus Energieunion
Topic 13: Gefahren durch AKW bzw. Atomwaffen; 5; Kernkraft; 5
Tihange Doel Atomkraftwerk Atomkraftwerken FANC Wasserstoffflocken Fukushima URENCO Uran russische radioaktiv seismische Thorium SAMOFAR Grenznahes
Topic 14: Musik und Extremismus; 5; Rechtsextremismus; 4
Rechtsrock Hooligans Einstiegsdroge Demonstrationspolitik Aktionsrepertoire Hooligan-Szene Tonträger Besucherzahlen Regener Musikstile Liederabende Musikveranstaltungen Sportler Berührung Mahnwache
Topic 15: Lebensmittelsicherheit; 5; EU-Freihandelsabkommen (CETA, JEFTA etc.); 5
Handelsabkommen Spätestens Drogenbeauftragte Freihandelsabkommen Produktgruppen Geflügel EU-Japan correctiv Pflanzenschutz Kompetenzzentrums Glyphosat trächtig Zuchtbecken Wirtschaftspartnerschaftsabkommen Wiedergenehmigung
Topic 16: Biotope; 3; Klimawandel; 5
Oberflächengewässer Fließgewässern Übertragungsnetzbetreiber Tier Erwärmung Klimakrise WRRL Methan Agrarlandschaft wolfssicheren wassersparend Wolfes Wetterdienstes Temperaturaufzeichnungen Starkregen

Topic 17: Schadstoffe; 5; Schadstoffe/Schadstoffbelastung in der Luft; 5
Stickstoffdioxid Stationsname Stationscode verkehrsnahen Grundwasserkörper Feinstaub BImSchV belastung Stickstoffdioxidgrenzwert Stickoxidabgasen Stickoxid-Werte Phosphat PCB-138 Lockergestein Jüngst
Topic 18: Netzversorgung; 5; Breitbandausbau; 5
Breitbandanschluss Schneller Breitbandversorgung Internetzugängen Mbit Kreisfreie Breitbandanschlüssen Haltepunkt Fördergegenstand Wirtschaftlichkeitslückenmodell halbstädtischem Standortfaktor städtisch Landkreisen Downstream
Topic 19: Gründungen/Startups; 5; Unternehmertum; 4
Sozialunternehmen EXIST Programmlinie EXIST-Gründerstipendium Skalierung Gründungskultur Gründerinnen profitorientierten nicht-technischen gesellschaftsdienlichen Zukunftsaufgaben Wagniskapital Vernetzungspotenziale Umsetzen Summer
Topic 20: Cybersecurity; 5; Cybersicherheit; 2
Gedenkstättenkonzeption Cyberraum Meldestelle Cyber-Angriffe Abhaltung Uranabbau BAköV Schadsoftware Verschlüsselung Datensicherheit Bundespolizeiakademie Privacy Tagesbefehl Kreml Hackerangriff
Topic 21: Militärgeschichte; 4; Traditionsverständnis der Bundeswehr; 1
Franco Wehrmacht Oberleutnant belgisch Konzert Bundesverteidigungsministerin ZMSBw Enthüllung StAG Schreiber Menschheit Lents Traditionsverständnis Lent-Kaserne Wehrmachtsandenken
Topic 22: Fliegen/Sicherheit/Gesundheit; 4; Flugverkehr; 4
Übungsflüge Passagierflüge Frachtflüge Lärmbelastungen Mitgesellschafterin Fluglärm Planfeststellungsbeschluss Flugbetrieb Flugbewegungen Verlagerungspotenzial Inlandsflüge Übungszone Übungsraum Übungsflugbetriebs fallweise
Topic 23: Digitale Überwachung; 5; Vorratsdatenspeicherung; 1
Vorratsdatenspeicherung TKÜV VerkDSpG Erbringer Speicherpflicht Telekommunikationsanbieter Verkehrsdaten anlasslosen Berufsgeheimnisträgern Kommunikationsdienste ungesicherte TK-Transparenzverordnung Streamingdienstleistungen Sicherheits-Updates SMS-Inhalten
Topic 24: Waffen; 3; Politische Kriminalität; 1
Schießübungen Tathergang Fensterscheibe Maaßen zurückgelassen Parteibüros Parteibüro Flugnummer CORRECTIV Schriftzüge Oldschool Schusswaffe Scheibe Vermieter Zeuge
Topic 25: Infrastruktur für E-Mobilität; 5; Ladeinfrastruktur für die Elektromobilität; 5
Schnellladestationen Rast Ladepunkten Ladepunkte Ladeinfrastruktur Ladesäulen Schnellladepunkten Normalladestationen Normalladepunkte Lade Marktdurchdringung tanken Elektrofahrzeuge Bundesautobahn Autobahnraststätten

Topic 26: Doping; 5; Sport; 4
Athlet Doping Spitzensportler Olympiastützpunkte Einheitspartei Dopingopfer-Hilfegesetz DOHG sozialistisch Sportlerinnen DOSB Nachwuchssportlerinnen Nachwuchssportler Spitzensportförderung Leistungssports Athletin
Topic 27: Fahrräder und Unfälle; 5; Fahrradverkehr; 4
Verkehrssicherheit Schutzstreifen Grundlagenuntersuchung Baumreihen Aufprall Alleenschutzes Radwegen Fahrradtourismus Radwege Radverkehr Fahrrad Radinfrastruktur Pedelecs Landstraßen urn-newsml-dpa-com-20090101
Topic 28: Fischereipolitik; 5; Fischerei(-politik); 5
Inspektor Fischereifahrzeuge ICES Hols Fangfahrten CCTV Anlandegebotes Seekontrollen Schonzeit Fangzusammensetzung Dorschen Fischereiüberwachung Fangquoten Fang Anlandeverpflichtung

Auf die ausführliche Darstellung der Ergebnisse der weiteren Sachexperten-Bewertungen wird verzichtet. Tabelle 28 zeigt jedoch zusammengefasst die Bewertungen der zwei Sachexperten für die drei Verfahren als normierte Mittelwerte.

	MW SE1	MW SE2	MW SE1 & SE2
Louvain Gewicht a	1	1	1
Louvain Gewicht b	0.914	0.713	0.813
hSBM	0.553	0.454	0.504

Tabelle 28: Zusammenfassung - Normierte Sachexperten(SE)-Bewertung

Es lässt sich zusammenfassen, dass beide Sachexperten die Topic Models jeweils ähnlich eingeschätzt haben. Louvain Gewicht a ist den anderen Verfahren gemäß dieser inhaltlichen Bewertung überlegen. An zweiter Stelle reiht sich Louvain Gewicht b ein. Die schlechteste Bewertung unter diesen Dreien hat das hSBM erhalten.

Elib

Ein DLR-Sachexperte hat die Louvain-Ergebnisse der drei Versionen des Elib-Korpus – Elib condensed mit Gewicht a, Elib condensed a2 mit Gewicht a und Elib condensed a3 mit Gewicht b – bewertet. Nach Anwendung des Louvain-Algorithmus auf Elib condensed wurden fünf Topics generiert. Für diese haben wir jeweils die am besten gerankten 200 Wörter nach Bayes ausgegeben und dem Sachexperten für eine Bewertung vorgelegt. Seine Aufgabe war es, für jedes generierte Topic drei Begriffe zu finden, die das Topic treffend beschreiben. Als Resultat hat er zwar für jedes Topic unterschiedliche Begriffe geliefert, aber hat er nach der Auswertung ausdrücklich betont, dass die Wörter innerhalb der Topics

zu breite Themenfelder abdecken. In den Ergebnissen der Version elib condensed a2 mit Gewicht a wurden ebenfalls fünf Topics mit Wortlisten ausgegeben. Hier hat er ebenfalls als Einschätzung vermittelt, dass die Themen sehr breit sind und in einigen sogar nicht zusammenhängende Wörter vorkommen. Nach Anwendung des Louvain-Algorithmus auf die letzte Version wurden 11 Topics ausgegeben. Hier fiel es dem Sachexperten leicht treffende Klassifikationen in Form von drei Begriffen für alle Topics zu finden. Die resultierten Topics des Louvain mit Gewicht b haben uns bereits in der Analyse der Ergebnisse im Auswertungskapitel 7.6.1 überzeugt.

7.7.3 Word Embedding

Im letzten Kapitel zur Sachexperten-Einschätzung standen bereits die inhaltlichen Zusammenhänge der Wortlisten im Vordergrund. Die Bewertung der Wortlisten durch Sachexperten ist mit viel Aufwand verbunden, sodass wir die Ergebnisse gerne automatisieren wollen. Somit haben wir nach einem automatisierten Verfahren gesucht, das uns eine ähnliche Einschätzung liefert, wie unsere Sachexperten-Ergebnisse. Bei der Bewertung der Topic-Qualität im Sinne einer inhaltlich gut zusammenpassenden Wortzusammenstellung werden in der Literatur - insbesondere beim LDA-Verfahren - unterschiedliche Kohärenzmaße benutzt. Beispiele dazu sind die Kohärenzmaße C_{UCI} oder C_{UMass} [25]. Bei unseren Topic-Listen können wir allerdings keine Übereinstimmung zwischen diesen Kohärenzmaßen und der Expertenmeinung erkennen.
Eine neue Möglichkeit, die Topic-Kohärenz zu messen, liefern die sogenannten Word Embeddings. Darunter versteht man mit Methoden des maschinellen Lernens an großen Korpora trainierte Abbildungen aller Wörter einer Sprache in einen euklidischen Raum mit wenigen hundert Dimensionen, welche semantisch ähnliche Wörter auf ähnliche Vektoren abbilden (im Sinne der Kosinus-Ähnlichkeit, also mit kleinem Winkel zwischen den Wortvektoren). Word Embeddings sind sprachabhängig. Für die für uns relevanten Sprachen Deutsch und Englisch benutzen wir die jeweiligen Sprachversionen von FastText [FastText], s. auch `https://fasttext.cc/`. Diese wurden an der deutschen, beziehungsweise englischen Wikipedia und mit zusätzlichen Common-Crawl-Inhalten, trainiert. Für das gesuchte Embedding-Kohärenzmaß liegt es nun nahe, für alle Paare verschiedener Wörter (a, b) eines Topics die Kosinus-Ähnlichkeit zu berechnen und dann zu mitteln. Es stellt sich aber als zweckdienlich heraus, hier nicht alle Wortpaare gleich zu gewichten, sondern Paare ausgefallener Wörter stärker zu gewichten als Paare gewöhnlicher Wörter. Hierfür ziehen wir die inverse Dokumenthäufigkeit Idf heran. Deshalb bilden wir

$$Coh_{Emb-Idf} = \frac{\sum_{a,b} sim(a,b) * Idf(a) * Idf(b)}{\sum_{a,b} Idf(a) * Idf(b)}. \tag{44}$$

Der Term $sim(a, b)$ für die Kosinus-Ähnlichkeit zwischen den Wortvektoren für a und b (im betrachteten Embedding) Dabei beziehen wir Idf aus den Korpora deWaC bzw. ukWaC [WaCky] (s. auch `https://wacky.sslmit.unibo.it/`), welche beide durch Webcrawling entstanden sind und jeweils fast 2 Milliarden Wörter enthalten.
Im nächsten Schritt haben wir die Expertenmeinung, bestehend aus den in 7.7.2 beschriebenen sowie auch die noch später beschriebenen, Bewertungen zum LDA, mit den Embedding-Kohärenzwerten verglichen. Die Embedding-Kohärenz wurde für die obersten 50 Wörter der Liste berechnet, da die Sachexperten auch die ersten 50 Wörter der Liste bewertet haben.
Die Korrelation der Sachexperten-Bewertungen und der Embedding-Kohärenz ist in Abbildung 29 abgebildet.

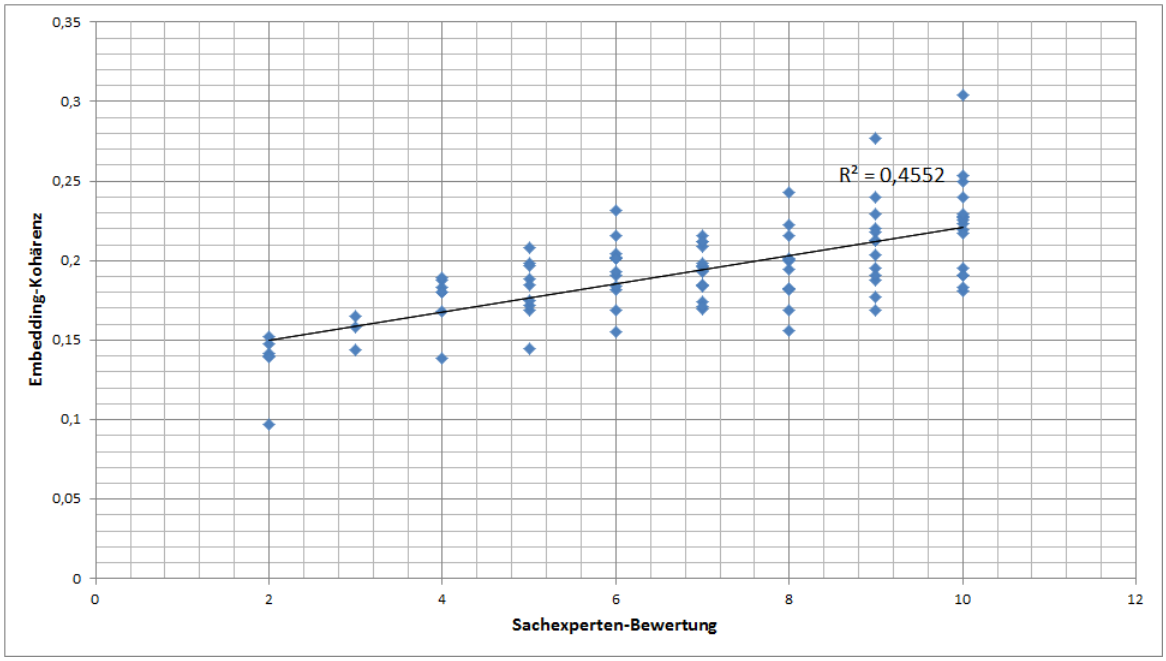

Abbildung 29: Korrelation - SE-Bewertung & Embedding-Idf-Kohärenz

Lineare Regression für den Zusammenhang zwischen Sachexperten-Bewertung und Embedding-Kohärenz ergibt einen Korrelationskoeffizienten von $R = 0.67$. Es ist also ein positiver Zusammenhang zwischen den Variablen erkennbar, so dass die Embedding-Kohärenz ersatzweise zur Beurteilung der Topic-Qualität herangezogen werden kann, um den Aufwand der Sachexperten-Bewertung einzusparen.
In der folgenden Tabelle 29 sind die normierten Sachexperten(SE)-Mittelwerte und die berechnete Embedding-Idf-Kohärenz für 50 Wörter zusammengefasst:

Verfahren	SE1	SE2	MW SE1 & SE2	MW Embedding-Idf-Kohärenz
Louvain, Gewicht a	1	1	1	0.208
Louvain, Gewicht b	0.914	0.713	0.813	0.197
hSBM	0.553	0.454	0.504	0.163

Tabelle 29: Normierte SE-Bewertung & Embedding-Idf-Kohärenz, 50 Wörter, Bayes

Umso kleiner die Einschätzung der Sachexperten, umso geringer der berechnete Embedding-Idf-Kohärenzwert.
Im nächsten Schritt werden die Ausgaben für jedes Verfahren der KA 2017 eigenständig verglichen. Im Anschluss erfolgt der Vergleich der einzelnen Verfahren miteinander. Die Embedding-Kohärenz wurde für 200, 100, 50 und 20 Wörter angewendet. Es ist offensichtlich, dass bei Berücksichtigung einer kleineren Wortanzahl die Embedding-Kohärenz bessere Werte annimmt. Zunächst erfolgt eine ausführliche Auswertung der Louvain-Ergebnisse.

KA 2017 - Louvain

In Tabelle 30 wurde die Embedding-Kohärenz des Bayes und des Common Word Rankings einander nach Anwendung des Louvain-Verfahrens mit Gewicht a auf Level 0 gegenübergestellt.

Level 0	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Bayes	0.228	0.228	0.232	0.243
Common Word	0.226	0.227	0.227	0.234

Tabelle 30: Louvain, Gewicht a, Level 0 - Embedding-Idf-Kohärenz

In der darauffolgenden Tabelle 31 werden die analogen Werte auf Level 1 ausgegeben.

Level 1	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Bayes	0.193	0.195	0.204	0.219
Common Word	0.185	0.185	0.193	0.208

Tabelle 31: Louvain, Gewicht a, Level 1 - Embedding-Idf-Kohärenz

Und schließlich in 32 auf Level 2.

Level 2	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Bayes	0.179	0.179	0.186	0.21
Common Word	0.173	0.175	0.179	0.193

Tabelle 32: Louvain, Gewicht a, Level 2 - Embedding-Idf-Kohärenz

Aus den Tabellen 30, 31 und 32 kann man ablesen, dass die feinsten Ebenen die besten Ergebnisse liefern. Ein zusätzliches Word-Embedding-Ergebnis ist, dass die Bayes-Sortierung besser abschneidet als die Common-Word-Sortierung. Hier bestätigt sich somit wiederum die Annahme, dass die Bayes-Sortierung besser für die Darstellung unserer Ergebnisse geeignet ist als die Common-Word-Sortierung. Offensichtlich ist auch, wie eben schon angedeutet, dass bei einer kleineren Wortanzahl in der Auswertung die Embedding-Kohärenz größer beziehungsweise besser ist.
Wenn wir nun die Ausgaben für Gewicht a und Gewicht b in Tabelle 33 vergleichen möchten, wählen wir hierfür die Ergebnisse auf der feinsten Ebene und für die Sortierung der Wörter das Bayes Ranking.

Bayes, Level 0	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Gewicht a	0.228	0.228	0.232	0.243
Gewicht b	0.226	0.227	0.232	0.245

Tabelle 33: Louvain, Bayes, Level 0 - Embedding-Idf-Kohärenz

Die Ergebnisse für Gewicht a und b sind sehr ähnlich. Auf der feinsten Ebene sollte es demnach keine Rolle spielen, ob wir uns hier für Gewicht a oder Gewicht b entscheiden. Auch auf den gröberen Levels erkennt man nur minimale Unterschiede. In Tabellen 34 und 35 sind die Ergebnisse für 20 Wörter zusammengefasst.

Bayes, Level 1	20 Wörter
Gewicht a	0.219
Gewicht b	0.207

Tabelle 34: Louvain, Bayes, Level 1 - Embedding-Idf-Kohärenz

Bayes, Level 2	20 Wörter
Gewicht a	0.21
Gewicht b	0.208

Tabelle 35: Louvain, Bayes, Level 2 - Embedding-Idf-Kohärenz

KA 2017 - Infomap

Vergleichen wir auch hier die berechnete Embedding-Kohärenz der Bayes-Sortierung mit der Common-Word-Sortierung, so erhalten wir die Werte in Tabelle 36. Also ist auch hier die Bayes-Sortierung für die Topic-Ausgabe besser geeignet.

Gewicht a, Level 0	20 Wörter
Bayes	0.241
Common Word	0.231

Tabelle 36: Infomap, Gewicht a, Level 0 - Embedding-Idf-Kohärenz

In der Anwendung des Infomap-Algorithmus ist es gerade interessant, das Gewicht b mit einzubeziehen, da wir durch ein solches Netzwerk kleinere Topic-Anzahlen generieren können.

Bayes, Level 0	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Gewicht a	0.226	0.227	0.23	0.241
Gewicht b	0.223	0.225	0.23	0.244

Tabelle 37: Infomap, Bayes, Level 0 - Embedding-Idf-Kohärenz

In Tabelle 37 haben wir die Embedding-Kohärenz auf der feinsten Ebene (Ebene 0), verglichen. Es sind nur minimale Unterschiede zu erkennen. Dies zeigt, dass wenn wir uns für die Gewichtsalternative b entscheiden, die gefundenen Topics qualitativ mindestens genauso gut sind.

KA 2017 - hSBM

Da auf feineren Ebenen sehr viele Topics vorliegen und diese häufig nur aus einem Wort oder sehr wenigen Wörtern bestehen, erfolgt der Vergleich hier auf der gröbsten Ebene. Denn wenn ein Topic nur ein Wort enthält, lässt sich zu diesem natürlich keine Embedding-Kohärenz berechnen, da sich das eine Wort mit keinem anderen vergleichen lässt. Auch das Auftreten von nur wenigen Wörter verfälscht das Ergebnis, da diese häufig besser zusammen passen, als 20 Wörter oder mehr.
Bei der Netzwerkbildung für das hSBM-Verfahren sind keine Gewichte notwendig. Deshalb vergleichen wir lediglich die Sortierung. Dies ist an dieser Stelle interessant, da die Common-Word-Sortierung die Standardsortierung im Programmcode von Gerlach ist.

Level 4	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Bayes	0.168	0.172	0.175	0.188
CommonWord	0.168	0.179	0.196	0.207

Tabelle 38: hSBM, Level 4 - Embedding-Idf-Kohärenz

Level 3	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Bayes	0.176	0.178	0.181	0.19
CommonWord	0.176	0.178	0.182	0.194

Tabelle 39: hSBM, Level 3 - Embedding-Idf-Kohärenz

Das Common Word Ranking schneidet bei den hSBM-Ergebnissen sowohl auf Level 4, als auch auf Level 3, besser ab als das Bayes Ranking.

KA 2017 - Vergleich: Louvain, Infomap, hSBM

Nun gilt es die Ergebnisse der unterschiedlichen Verfahren mithilfe der berechneten Embedding-Kohärenz zu vergleichen.
Wenn wir die Topic-Ausgabe der drei Verfahren einander gegenüberstellen möchten, so müssen wir die unterschiedliche Topic-Anzahl auf den verschiedenen Ebenen berücksichtigen. Beispielsweise ist es nicht sinnvoll, das gröbste Level von Infomap mit dem gröbsten Level von Louvain zu vergleichen. Denn Infomap enthält über 400 Topics und Louvain nur ca. 13. Es folgt ein Vergleich der Louvain- und Infomap-Ergebnisse auf der feinsten Ebene, und ein Vergleich der hSBM- und Louvain-Ergebnisse auf der gröbsten Ebene.
Zunächst zum Vergleich Louvain und Infomap auf der feinster Ebene.

Gewicht a, Bayes, Level 0	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Louvain	0.228	0.228	0.232	0.243
Infomap	0.226	0.227	0.23	0.241

Tabelle 40: Vergleich - Louvain, Infomap, Gewicht a, Bayes, Level 0

Wir sehen, dass die Topics des Louvain-Algorithmus ein wenig besser abschneiden als die Topics des Infomap-Algorithmus. Im letzten Vergleich erfolgt eine Gegenüberstellung der Embedding-Kohärenzen für Louvain Gewicht a auf Level 2 und hSBM auf Level 4. Die gewählte Sortierung ist das Bayes Ranking.

Bayes	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Louvain, Gewicht a, Level 2	0.179	0.179	0.186	0.21
hSBM, Level 4	0.168	0.172	0.175	0.188

Tabelle 41: Vergleich - Louvain, hSBM, Bayes

Die Topics des Louvain-Verfahrens liefern bessere Ergebnisse als die des hSBM-Verfahrens.

Elib

An dieser Stelle möchten wir für die Elib-Version Elib condensed a2 Gewicht a und b auf gröbstem Level einander gegenüberstellen.

Louvain, Bayes, Level 2	200 Wörter	100 Wörter	50 Wörter	20 Wörter
Gewicht a	0,144	0,15	0,156	0,151
Gewicht b	0,142	0,158	0,176	0,189

Tabelle 42: Elib condensed a2: Vergleich - Louvain, Bayes, Level 2

Hier ist klar zu erkennen, dass für diese Version das Gewicht b besser abschneidet als das Gewicht a.

7.7.4 Vergleich der Verfahren mit LDA

Wir haben den Sachexperten nicht nur mit Louvain und hSBM erzeugte Wortlisten vorgelegt, sondern auch durch LDA generierte Wortlisten. Hier wurde als Sampling Methode gibbs sampling gewählt. Die Programmierung wurde mit der mallet Bibliothek [36] in der gensim Variante umgesetzt [37]. Ein Nachteil des LDA-Verfahrens ist es, dass man die Topic-Anzahl mit angeben muss. Mithilfe anderer Kohärenzmaße wurde eine geeignete Topic-Anzahl gefunden. Es hat sich als gut herausgestellt, die Anzahl 16 und 27 zu wählen. Somit haben wir die so erzeugten Topic-Listen den Sachexperten zur Bewertung vorgelegt.

Verfahren	MW SE1	MW SE2	MW SE1 & SE2	MW Word Embedding
Louvain Gewicht a	1	1	1	0.208
Louvain Gewicht b	0.914	0.713	0.813	0.197
LDA TA 16	0.858	0.836	0.847	0.196
LDA TA 27	0.951	0.877	0.914	0,197
hSBM	0.553	0.454	0.504	0.163

Tabelle 43: Zusammenfassung - Normierte Sachexperten(SE)-Bewertung mit LDA

Die Sachexperten haben neben den Ergebnissen für Louvain Gewicht a, Louvain Gewicht b und hSBM zwei verschiedene durch LDA generierte Wortlisten bewertet. Die Bewertung der Sachexperten und die gemittelte Embedding-Kohärenz sind in Tabelle 43 wiedergegeben. Für die KA 2017 ist das Louvain-Verfahren mit den Gewichten a demnach eindeutig überlegen. Die Verfahren Louvain mit Gewicht b, LDA mit Topic-Anzahl 16 und 27 haben eine vergleichbare Qualität. hSBM ist nach Sachexperten-Einschätzung wie auch nach Bewertung durch Word Embedding das schlechteste Verfahren.

8 Zusammenfassung

Die Arbeit hat die folgenden zwei Hauptanliegen:

1. Vorstellung dreier Netzwerkverfahren für das Topic Modeling als Alternative zu den derzeit populären Verfahren (wie LDA).

2. Bewertung und Vergleich der Qualität der daraus erhaltenen Ergebnisse/Topics.

Dafür haben wir die folgenden Punkte untersucht:

a) Wie kann eine Textsammlung für die Verfahren vorbereitet werden?

b) Wie können Netzwerke gebildet und darauf Community-Detection-Verfahren angewendet werden?

c) Wie können Topics so ausgegeben werden, dass sie interpretierbar sind?

d) Und schließlich wie kann die Tauglichkeit der Topics und damit der Verfahren bewertet werden?

All das haben wir in zwei Datensätzen durchgespielt: Zum einen mittels Kleiner Anfragen an die Bundesregierung, zum anderen mittels Dokumenten der DLR-Publikationsdatenbank Elib.
Im Punkt a) wird zunächst ein Korpus aus den genannten Datenquellen generiert. Nach einer mehrstufigen Datenbereinigung wird ein Keyword-Extraction-Verfahren – Positional idfRank – auf den Korpus angewandt, um die Wortzahl im Korpus zu reduzieren. Hierbei werden die Wörter, die in den einzelnen Dokumenten zentral (im Sinne eines modifizierten PageRank-Verfahrens) sind, bevorzugt und jene, die häufig im gesamten Korpus vorkommen, bestraft. Auf diese Weise werden ausschließlich die Begriffe berücksichtigt, die für eine Interpretation der Topics als bedeutend erscheinen. Im Anschluss werden verschiedene Versionen der Korpora erstellt, indem unterschiedlich starke Reduzierungen der Textlängen vorgenommen werden. Wir sehen bei unseren Untersuchungen an Elib-Dokumenten, dass eine Reduktion der berücksichtigten Wortzahl zu spezifischeren Topics führt: Topics, die bei großer Wortzahl gefunden werden, spalten sich bei kleinerer Wortzahl weiter auf.
Der nächste Punkt b) spaltet sich in b1) Netzwerkbildung und b2) Anwendung der Community-Detection-Verfahren auf. Nach der Datenbereinigung und der Keyword Extraction wird in b1) ein Netzwerk aus den verbliebenen Wörtern des Korpus konstruiert. Wir haben verschiedene Varianten betrachtet. Die einfachste Variante ist ein Kookkurrenznetzwerk mit den Wörtern als Knoten, die durch eine Kante verbunden werden, wenn zwei Wörter gemeinsam in einem Dokument vorkommen. Die Kanten werden gewichtet:

Das Gewicht gibt an, in wie vielen Dokumenten ein vorhandenes Wortpaar gemeinsam vorkommt. Mit dem Ziel, dass sich das Netzwerk reduziert und wir uns dadurch neben technischen auch inhaltliche Vorteile verschaffen können, haben wir auch eine andere Gewichtsalternative betrachtet. Diese bezieht die Auftrittshäufigkeit der Wörter innerhalb der Dokumente mit ein.
Nach dieser Netzwerkbildung werden in b2) zwei bekannte Community-Detection-Verfahren angewendet, Louvain und Infomap. Ein drittes Verfahren, hierarchical Stochastic Block Model (hSBM), wird auf ein ähnlich gebildetes Wort-Dokumenten-Netzwerk angewandt. Alle drei Verfahren erzeugen Wortlisten, die als Topics zu interpretieren sind. Bei allen drei Verfahren wird eine Hierarchie von Topics erzeugt, beginnend mit feineren, die in gröberen zusammenfließen.
Um die Interpretation der Wortlisten in Punkte c) zu vereinfachen, ist es hilfreich die Wörter in eine geschickte Reihenfolge zu bringen. Eine Möglichkeit ist es hier, die Wörter absteigend nach ihrer Vorkommenshäufigkeit zu sortieren. Die Sortierung nennen wir Common Word Ranking. Eine alternatives Sortierverfahren, das es ermöglicht die Wörter nach Bedeutsamkeit zu sortieren, ist das sogenannte Bayes Ranking. Dies basiert darauf, die Häufigkeit im Gesamtkorpus und in einzelnen Dokumenten sowie die Position der Wörter in den einzelnen Dokumenten zu berücksichtigen. Wir stellen bei der Auswertung der Verfahren fest, dass das Bayes Ranking die Interpretierbarkeit und Aussageschärfe der Topics wesentlich verbessert.
Bei der Bewertung der Verfahren in Punkt d) benutzen wir unterschiedliche Kriterien: Wir vergleichen Modularitäten als graphentheoretisches Merkmal, wir analysieren, wie die gefundenen Topics mit vorliegenden Klassifizierungen der Textsammlungen zusammen passen - in Form von Heat Maps -, wir befragen Sachexperten und wir berechnen ein aus Word Embeddings abgeleitetes Kohärenzmaß, von dem wir nachweisen können, dass es mit der Einschätzung von Sachexperten korreliert.
Insgesamt ist das Louvain-Verfahren am erfolgreichsten. Es liefert sogar bessere Ergebnisse als das State-Of-The-Art-Verfahren LDA. Im Detail ergeben sich folgende interessante Beobachtungen: Die Modularität, welche als Kennzahl die Qualität der Community-Struktur wiedergibt, liefert sehr gute Ergebnisse. Für den KA Korpus werden Topics generiert, welche mit den in der Heat Map abgelesenen Sachgebieten übereinstimmen. Für den Elib-Korpus können wir zudem in den von uns erstellten Heat Maps alle DLR-Bereiche ablesen: So sind Energie, Verkehr, Luft- und Raumfahrt als jeweiliges Topic erkennbar. Sowohl die Sachexperten als auch das Embedding-Kohärenzmaß weisen darauf hin, dass Louvain die besten hier vorgestellten Topics generiert.
Das Infomap-Verfahren gibt auf den KA 2017 Korpus mit der einfachsten Gewichtsvari-

ante eine große Topic-Anzahl aus. Ein Vergleich mit den KA-Sachgebieten und die Modularitätsberechnung weisen darauf hin, dass dies qualitativ gute, wenn auch spezifische Topics sind. Es ist aber eine Topic-Anzahl wünschenswert, welche sich gut und einfach analysieren und bewerten lässt. Der Versuch das Verfahren auf ein mit der alternativen Gewichtsvariante konstruierten Graphen anzuwenden, hat nach unseren Analysen auch keine vielversprechenderen Ergebnisse ergeben. Für den Fall, dass sehr spezifische Topics gesucht sind, so liefert das Verfahren sehr gute Ergebnisse. Wir sind für unsere weiteren Untersuchungen zu dem Entschluss gekommen, dass uns die hierarchische Topic-Ausgabe nicht zufrieden stellt. Somit wurde das Verfahren in der Expertenbewertung und dem Word-Embedding-Verfahren nicht mitberücksichtigt.
Die hSBM-Ergebnisse weisen einen schlechteren Modularitätswert auf als seine Konkurrenzverfahren. Vorsicht ist hier bei der Verwertung der Topics auf feineren Ebenen geboten. Hier liegen extrem viele Topics vor, die häufig nur ein Wort enthalten. Somit sind die Topiclisten nur auf den gröberen Ebenen in dieser ungekürzten Form verwertbar. Bei der Heat Map Erstellung des Elib Korpus machen wir zudem die Feststellung, dass Topics sehr zerstreut vorliegen und die einzelnen Bereiche des DLR nicht klar erkennbar sind. Auch die Sachexperten Bewertung sowie das Word Embedding wertet das Verfahren als klaren Verlierer. Es reiht sich in der Bewertung als schlechtestes Verfahren ein, und ist demnach sogar schlechter als das Konkurrrenzverfahren LDA.
Diese Arbeit beschäftigt sich mit dem Auffinden von Topics in Texten mittels graphentheoretischen Methoden. Der Erfolg dieser Methoden in der Anwendung lässt sich nicht allein durch mathematische Kriterien beurteilen, da es um Sprache und Bedeutung von Texten geht. Dennoch zeigt diese Arbeit, dass es gut gelingt, mit den vorgeschlagenen Verfahren den Sachexperten eine hilfreiche Automatisierung in der Auswertung großer Textbestände an die Hand zu geben. Weitergehende Untersuchungen zur Verbesserung der Verfahren erscheinen sinnvoll, da diese Arbeit auch erkennen lässt, dass es viele Variationsmöglichkeiten gibt, deren Einfluss man noch näher analysieren kann.

Literatur

[1] R. Diestel, Graphentheorie, 3rd ed. Berlin: Springer, 2006.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech., vol. 2008, no. 10, P10008, 2008.

[3] U. Brandes et al., "Maximizing modularity is hard," arXiv preprint physics/0608255, 2006.

[4] M. D. Salerno, C. A. Tataru, and M. R. Mallory, "Word Community Allocation: Discovering Latent Topics via Word Co-Occurrence Network Structure," 2015.

[5] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community Detection via Maximization of Modularity and Its Variants," IEEE Trans. Comput. Soc. Syst., vol. 1, no. 1, pp. 46–65, 2014.

[6] J. Zeng and H. Yu, "A Scalable Distributed Louvain Algorithm for Large-Scale Graph Community Detection," in 2018 IEEE International Conference on Cluster Computing: 10–13 September 2018, Belfast, United Kingdom : proceedings, Belfast, 2018, pp. 268–278.

[7] M. Werner and O. Mildenberger, Information und Codierung: Grundlagen und Anwendungen. Wiesbaden: Vieweg+Teubner Verlag, 2002.

[8] T. Strutz, Bilddatenkompression. Wiesbaden: Springer Fachmedien, 2009.

[9] L. Lovász, "Random walks on graphs: A survey," Combinatorics, Paul erdos is eighty, vol. 2, no. 1, pp. 1–46, 1993.

[10] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," Eur. Phys. J. Spec. Top., vol. 178, no. 1, pp. 13–23, http://arxiv.org/pdf/0906.1405v2, 2009.

[11] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," Proceedings of the National Academy of Sciences, vol. 105, no. 4, pp. 1118–1123, 2008.

[12] M. Rosvall and C. T. Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems," (eng), PloS one, vol. 6, no. 4, e18209, 2011.

[13] L. Bohlin, A. Lancichinetti, D. Edler, and and M. Rosvall, "Community detection and visualization of networks with the map equation framework," 2014.

[14] R. Baeza-Yates and B. Ribeiro-Neto, Modern information retrieval: ACM press New York, 1999.

[15] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Computer networks and ISDN systems, vol. 30, no. 1-7, pp. 107–117, 1998.

[16] A. Lancichinetti et al., "High-Reproducibility and High-Accuracy Method for Automated Topic Classification," Phys. Rev. X, vol. 5, no. 1, p. 993, 2015.

[17] S. N. Kim, O. Medelyan, M. Kan, and T. Baldwin, "SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles," 2010.

[18] A. Lancichinetti, M. Irmak Sirer, J. X. Wang, D. Acuna, K. Körding, L. A. Nunes Amaral, "High-Reproducibility and High-Accuracy Method for Automated Topic Classification - Supplemental Material," Phys. Rev. X, vol. 5, no. 1, p. 993, 2015.

[19] M. Gerlach, T. P. Peixoto, and E. G. Altmann, "A network approach to topic models," Science advances, vol. 4, no. 7, eaaq1360, 2018.

[20] T. P. Peixoto, "Nonparametric Bayesian inference of the microcanonical stochastic block model," (eng), Physical review. E, vol. 95, no. 1-1, p. 12317, 2017.

[21] T. P. Peixoto, "Hierarchical Block Structures and High-resolution Model Selection in Large Networks," Phys. Rev. X, vol. 4, no. 1, p. 9851042, http://arxiv.org/pdf/1310.4377v6, 2014.

[22] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," (eng), Physical review. E, Statistical, nonlinear, and soft matter physics, vol. 83, no. 1 Pt 2, p. 16107, 2011.

[23] B. S. Khan and M. A. Niazi, "Network community detection: A review and visual survey," arXiv preprint arXiv:1708.00977, 2017.

[24] S. Fortunato, "Community detection in graphs," Physics Reports, vol. 486, no. 3-5, pp. 75–174, http://arxiv.org/pdf/0906.0612v2, 2010.

[25] M. Röder, A. Both, and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," in Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15, Shanghai, China, 2015, pp. 399–408.

[26] M. T. Schaub, J.-C. Delvenne, M. Rosvall, and R. Lambiotte, "The many facets of community detection in complex networks," Applied network science, vol. 2, no. 1, p. 4, 2017.

[27] A. Lancichinetti and S. Fortunato, “Community detection algorithms: a comparative analysis,” Phys. Rev. E, vol. 80, no. 5, p. 161, http://arxiv.org/pdf/0908.1062v2, 2009.

[28] Z. Yang, R. Algesheimer, and C. J. Tessone, “A Comparative Analysis of Community Detection Algorithms on Artificial Networks,” (eng), Scientific reports, vol. 6, p. 30750, 2016.

[29] A. Hamm, Complex word networks - comparing and combining information extraction methods. [Online] Available:https://www.researchgate.net/publication/333488824_Complex_word_networks_-_comparing_and_combining_information_extraction_methods. Accessed on: Nov. 17 2019.

[30] Open Knowledge Foundation Deutschland e.V., Kleine Anfragen. [Online] Available: https://kleineanfragen.de/. Accessed on: Nov. 17 2019.

[31] Thomas Aynaud, Community detection for NetworkX’s documentation. [Online] Available: https://python-louvain.readthedocs.io/en/latest/. Accessed on: Nov. 17 2019.

[32] Thomas Aynaud, GitHub - Community detection for NetworkX’s documentation. [Online] Available: https://github.com/taynaud/python-louvain/blob/master/docs/index.rst. Accessed on: Nov. 17 2019.

[33] D. Edler, A. Eriksson and M. Rosvall, The MapEquation software package, available online at http://www.mapequation.org.

[34] D. Edler and M. Rosvall, GitHub - Infomap Software Package. [Online] Available: https://mapequation.github.io/infomap/. Accessed on: Nov. 17 2019.

[35] Martin Gerlach, GitHub - hSBM Topicmodel. [Online] Available: https://github.com/martingerlach/hSBM_Topicmodel. Accessed on: Nov. 17 2019.

[36] McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit."http://mallet.cs.umass.edu. 2002.

[37] Software Framework for Topic Modelling with Large Corpora, Radim Řehůřek and Petr Sojka, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks,45–50, 2010, May, 22, ELRA, Valletta, Malta,http://is.muni.cz/publication/884893/en, English.

[38] networkx https://networkx.github.io/

[39] https://git.skewed.de/count0/graph-tool

[FastText] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," arXiv preprint arXiv:1802.06893, 2018.

[WaCky] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta, "The WaCky wide web: a collection of very large linguistically processed web-crawled corpora," Language resources and evaluation, vol. 43, no. 3, pp. 209–226, 2009.

Eigenständigkeitserklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Bonn, den 23. November 2019

Jana Thelen