

Sentimentanalyse von Tweets zur Meinungsbild Analyse im Bezug auf die Covid-19 Pandemie

Studienarbeit

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Ravensburg Campus Friedrichshafen

von

Paul Groß und Philipp Kappus

Juli 2021

Bearbeitungszeitraum
Matrikelnummer, Kurs
Gutachter

24 Wochen
2630435, 4265963, TIT18
Prof. Dr. Andreas Judt

Erklärung

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema: *Sentimentanalyse von Tweets zur Meinungsbild Analyse im Bezug auf die Covid-19 Pandemie* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Friedrichshafen, Juli 2021

Paul Groß und Philipp Kappus

Abstract

In this paper we investigate the use of Open-Source sentimentanalysis tools in terms of clustering users on Twitter. Furthermore we present two clustering methods to determine users with similar opinions and common topics on the Covid-19 pandemic and the related public debate in Germany. We believe, they can help gaining an overview over similar-minded groups and could support the prevention of fake-news distribution. The first method uses a new approach to create a network based on retweet-relationships between users and the most retweeted characters. The second method extracts hashtags from users posts to create a user feature vector which is then clustered using a combination of the k-Means and DB-SCAN algorithm to identify groups using the same language. With both approaches it was possible to identify clusters that seem to fit groups of different public opinion in Germany. However, we also found that clusters from one approach can not be associated with clusters from the other due to the filtering steps in the two methods.

Inhaltsverzeichnis

Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
Listings	VIII
1 Einleitung	1
1.1 Fragestellung	2
1.2 Umfeld	2
2 Vorüberlegungen	3
2.1 Hypothesen	3
2.2 Vorgehensweise	4
3 Daten Akquisition	5
3.1 Tweets sammeln	5
3.2 Struktur eines Tweets	6
4 Sentimentanalyse	7
4.1 Technologieauswahl	7
4.2 Analyse	8
5 Clustering Retweets	10
5.1 Daten vorbereiten	10
5.2 Influencer identifizieren	11
5.3 Nutzergruppen aggregieren	12
5.4 Kanten filtern	14
5.5 Kantengewichte normalisieren	15
5.6 Clustering	16
5.7 Einblick in die Cluster	17
5.8 Rückschluss auf Nutzer*innen	19
5.9 Bestimmung der Themen innerhalb der Cluster	19
6 Gruppierung der Nutzer*innen anhand der verwendeten Wörter	21
6.1 Berechnung der Ähnlichkeit von Texten	21
6.2 Aufbereitung von Texten zur Ähnlichkeitsanalyse	23
6.2.1 Entfernen von Stopwörtern	23
6.2.2 Lemmatisierung der Schlüsselworte	23

6.3	Verifikation der Ähnlichkeitsanalyse	25
6.3.1	Vorgehen und Auswahl der Stichprobe	25
6.3.2	Auswertung der Ergebnisse	26
6.3.3	Auswahl einer Distanzfunktion	26
6.4	Erstellung der Merkmalsvektoren für Nutzer*innen	28
6.5	Clustering anhand der Merkmalsvektoren	29
6.5.1	Berechnung einer Ähnlichkeitsmatrix mithilfe des k-Means Algorithmus	29
6.5.2	Einteilung der Nutzer*innen mit dem DB-SCAN Algorithmus . . .	30
6.6	Darstellung der Ergebnisse	31
7	Zusammenführung und Vergleich	34
7.1	Filterschritte im Netzwerkansatz	34
7.2	Filterschritte im Sprachansatz	35
8	Auswertung	36
8.1	Ergebnisinterpretation des Retweetclustering	36
8.2	Ergebnisinterpretation des Keywordclustering	37
8.3	Anwendungsfälle	37
9	Rückblick	38
10	Ausblick	39
Literaturverzeichnis		41
Glossar		43
Anhang		46

Abkürzungsverzeichnis

AWS Amazon Web Services

API Application Programming Interface

JSON JavaScript Object Notation

NLP Natural Language Processing

MDS Multidimensional Skalierung

Abbildungsverzeichnis

3.1	Amazon Web Services (AWS) - Architektur zur Tweet Akquisition	6
4.1	Verteilung der Nutzer*innen anhand der durchschnittlichen Sentimentwerte ihrer Tweets	8
5.1	Verteilung der Anzahl an Retweets (blau) und das inverse Potenzgesetz (orange)	12
5.2	Graph mit drei Influencern (blau)	13
5.3	Graph mit aggregierten Superknoten	14
5.4	The distribution of retweets (blue) and the power-law from equation 4(orange)	15
5.5	Ergebnisgraph März 2021	16
5.6	Nachbarn eines Influencer über verbundene Superknoten	17
5.7	Graph mit Markierungen um die gefundenen Cluster	18
5.8	Themen der Cluster als Wortwolken	19
5.9	Bestimmung der Themen innerhalb der Cluster	20
6.1	Vergleich unterschiedlicher Bibliotheken zur Lemmatisierung	25
6.2	Bestimmung der Merkmalsvektoren der Nutzer*innen	29
6.3	Gutes (links) und schlechtes (rechts) Ergebnis des k-Means Algorithmus unter gleichem Input	30
6.4	Berechnung einer Ähnlichkeitsmatrix durch mehrere Iterationen von k-Means	31
6.5	Einteilung von Punkten in Cluster und Rauschen durch DB-SCAN	31
6.6	Überführung der Ähnlichkeitsmatrix zu einem Graphen	32
6.7	Visualisierung der Gruppierung von Nutzer*innen anhand der von ihnen verwendeten Wörter	33
7.1	Benutzerassoziationen zwischen Clustern der beiden Ansätze	34

Tabellenverzeichnis

6.1	Überführung von Wortmengen zu Vektoren für einzelne Nutzer*innen . . .	22
6.2	Ähnlichkeitsmaße der einzelnen Artikel unter Verwendung der Kosinus-Ähnlichkeit	27
6.3	Differenzierbarkeit von Werten berechnet mithilfe der Kosinus-Ähnlichkeit im Vergleich zur euklidischen Distanz	28

Listings

5.1 Metdaten des Nutzer: „tagesschau“	11
5.2 Abwandlung der DBSCAN Nachbarfunktion	17

1 Einleitung

Auf Twitter bringen täglich 192 Millionen aktive (monetarisierbaren) Nutzer*innen ihre Ansichten und Meinungen in kurzen Texten zum Ausdruck [1]. Dies stellt eine beliebte Basis für Datenanalysten dar, die das Interesse von Menschen an Themen oder Produkten analysieren, Persönlichkeitstudien durchführen, Menschen gruppieren und sogar Grippeausbrüche vorhersagen [2]. Insbesondere das Einteilen von Tweets oder Benutzer in Gruppen hat großes Interesse bei Wissenschaftler erweckt, darunter [3] die Kunden für gezielte Werbung clustert, [4] der wichtige Ereignisse durch eine hierarchische Einteilung unter Verwendung einer paarweisen Cosinus-Distanz erkennt oder [5] der Tweets nach Hashtags mit dem K-Means-Algorithmus, agglomerativem hierarchischem Clustering und einem Fuzzy-Nachbarschaftsmodell gruppiert. [6] hat einen Klassifikator entwickelt, der Benutzer in politisch-links und politisch-rechts gruppiert, indem ein Netzwerk von Referenzbenutzern basierend auf Retweets (und Erwähnungen) aufgebaut wurde und dann jeder Gruppe Schlüsselwörter zugewiesen werden, die aus den Tweets der Referenzbenutzern extrahiert wurden.

Diese Idee erweiternd ist das Ziel dieser Arbeit, differenziertere Cluster unter den Menschen zu finden, die über ein bestimmtes Thema sprechen. An [7] angelehnt war die anfängliche Idee, Cluster anhand einer Sentimentanalyse der Tweets zu finden. Diese liefert für einen Text eine Einschätzung ob dieser eine eher positive oder negative Haltung ausdrückt. Im Zuge der Datenanalyse stellte sich allerdings heraus, dass dieser Ansatz nicht zielführend ist. Parallel dazu wurden zwei weitere Clusteringmethoden entworfen. Die erste Methode baut ein Kommunikationsnetzwerk auf, das auf Beziehungen zwischen Benutzern basiert, bei denen einer den anderen retweetet. Durch intelligente Filterung kann dann der DB-SCAN-Algorithmus verwendet werden, um Cluster zu erkennen.

Basierend auf dem in [8] evaluierten Verfahren, clustert der zweite Ansatz Benutzer mit einer Kombination aus k-Means und DB-SCAN anhand ihrer am häufigsten verwendeten Schlüsselwörtern und Hashtags.

In den Jahren 2020 und 2021 polarisierte die Covid-19-Pandemie und die daraus resultierenden gesetzlichen Regelungen die Gesellschaft und führten zu einer großen Diskussion, welche die sozialen Netzwerke einschließlich Twitter dominierte. So ist der meist verwendeter Hashtag in Deutschland im Jahr 2020 "Corona" [9]. In Deutschland erlangten die sogenannten „Querdenker“ bundesweite Aufmerksamkeit, indem sie Proteste gegen

Covid-19-bezogene Regelungen organisierten, die Existenz des Virus teilweise leugneten und verschiedene andere Verschwörungstheorien auf Twitter verbreiteten.

Die Identifizierung solcher Gruppen kann von großem Interesse sein um die Verbreitung von Fake News zu verhindern und einen Überblick über verschiedene Meinungs-„Lager“ zu gewinnen. All das macht Covid-19 zum perfekten Thema für die Anwendung der Clustering-Ansätze. Ergebnisse beider Methoden können unter <https://andfaxle.github.io/twitteranalysis/> eingesehen werden.

1.1 Fragestellung

Ein Bericht der Online Civil Courage Initiative stellt dar, wie Online-Hass zur gesellschaftlichen Polarisierung und extremistischer Radikalisierung beitragen kann. Verantwortlich dazu sind unter Anderem sogenannte „Echokammern“ in denen Menschen vorrangig mit Informationen und Meinungen konfrontiert sind, die den eigenen Einstellungen und Sichtweisen entsprechen. [10] In dieser Arbeit soll die Frage beantwortet werden, inwieweit solche „Echokammer“ wissenschaftlich exakt aus dem chaotischen Datensatz der Covid-19 Pandemie von Twitter extrahiert werden können. Daraus ergeben sich folgende Fragen:

Lassen sich Gruppen von Nutzer*innen finden, die 1) die gleiche Meinung zur Covid-19 Pandemie haben, 2) diese Meinung mit dem selben Sentiment kundtun, 3) diese Meinung mit der gleichen Sprache kommunizieren und 4) Informationen nur aus ihrer assoziierten Gruppe erhalten.

1.2 Umfeld

Die Arbeit wurde als Studienarbeit an der Dualen Hochschule Baden-Württemberg (DHBW) Ravensburg am Campus Friedrichshafen geschrieben. Die DHBW bietet das duale Studium an, das Studierenden die Integrierung von Praxisphasen bei einem Partnerunternehmen ermöglicht. Von den insgesamt 3700 Studenten, befinden sich 1500 am Campus Friedrichshafen an der Fakultät Technik. Diese Arbeit wurde unabhängig und ohne Unterstützung des Partnerunternehmens geschrieben.

2 Vorüberlegungen

In diesem Kapitel sollen zunächst die Hypothesen dargestellt werden, anhand derer die in der Fragestellung formulierten Fragen potentiell beantworten kann. Von diesen wird dann eine Vorgehensweise abgeleitet.

2.1 Hypothesen

Will man Objekte in Gruppen einteilen, so muss man zunächst definieren, anhand welcher Merkmale man dies umsetzen will. Bei einer Menge an Punkten aus dem zweidimensionalen Raum bedient man sich einfach deren Ortsvektor. Bei einer Menge an Twitter Nutzer*innen ist dies schwieriger. Diese Arbeit versucht, diese anhand von drei Merkmalskategorien einzuteilen, die nun hypothetisch erläutert werden:

Zum einen wird davon ausgegangen, dass Nutzer*innen sich anhand ihrer positiven bzw. negativen Einstellung zu einem bestimmten Thema kategorisieren lassen können sollten. Durch eine Sentimentanalyse aller Tweets eines*r Nutzers*in kann auf die Stimmung der Person geschlossen werden. Vergleicht man die Stimmungsbilder der Nutzer*innen untereinander, so könnten sich Kategorien ergeben. Ist dies Realisierbar, könnte auch über den zeitlichen Verlauf die Stimmung in Deutschland zum Thema Corona dargestellt werden. In Kapitel 4 wird dieser Hypothese nachgegangen.

Ein weiteres markantes Merkmal könnte sein, wen ein bestimmte*r Nutzer*in wie oft retweetet, also den Text eines*r anderen*en (manchmal auch kommentiert) an seine/ihre eigene Followerschaft weiterleitet. Retweetet eine Person eine andere, so ist sie der Meinung, dass der Tweet der/des Anderen es wert ist, (kommentiert) weiter propagiert zu werden. Dies könnte einer Zustimmung gleichgesetzt werden, jedoch kann es gut sein, dass das eigene Kommentar die Aussage des original Tweets kritisiert. In diesem Fall ist das Gegenteil der Fall. In der Hoffnung, dass dieser Effekt statistisch gesehen keine großen Auswirkungen hat wird diese Theorie in 5 in die Praxis umgesetzt.

Die dritte Merkmalskategorie betrifft die Sprache, präziser die Worte die verwendet werden. Die Sprache einer Person, also welche Worte sie in welchem Zusammenhang wann wählt, ist eine wichtige Charaktereigenschaft. Zum einen dient sie häufig als Merkmal

der gesellschaftlichen Klassenzugehörigkeit, zum anderen definieren die Wörter auch die Themen mit denen wir uns auseinandersetzen. Man kann nicht über den Klimawandel diskutieren, ohne sich des Wortes *Klimawandel* zu bedienen. Somit sollten alle Personen, bei welche das Wort Klimawandel in ihren Tweets zu finden ist, sich auch mit dem Klimawandel beschäftigen. Man kann sie also anhand der Verwendung oder Nichtverwendung dieses Wortes klassifizieren. Eine Einschränkung ist jedoch, dass man dadurch zwar bestimmen kann, ob sich jemand mit einem bestimmten Thema beschäftigt, aber nicht, welche Meinung er/sie zu diesem hat. Dies könnte sich aber aus der Summen seiner/ihrer Themen ergeben. Diesem Ansatz soll in Kapitel 6 nachgegangen werden.

2.2 Vorgehensweise

Ein detailliertes Vorgehen ist aufgrund der hohen Komplexität des Sachverhalts schwer zu definieren und noch schwerer einzuhalten. Grundsätzlich muss zunächst der Datensatz agglomeriert werden. Anschließend soll nach einem agilen Ansatz jeder der drei Hypothesen nachgegangen werden. Dabei soll eine Arbeitsaufteilung nach Hypothese erfolgen. Des weiteren soll zunächst nach dem Prinzip des Most Valuable Products ein Walking Skeleton der Analysemethoden erschaffen werden, bevor man sich um eine detaillierte Optimierung der Algorithmen bemüht. Somit soll verhindert werden, dass in etwas viel Arbeit gesteckt wird, welches sich im ganzen nicht realisieren lässt. Alle wichtigen Erfolge und Fehlschläge sollen mit dem Projektteam kommuniziert werden.

3 Daten Akquisition

Jede*r Nutzer*in, der die Dienste von Twitter benutzen will, kann dies nur nach Bestätigung der allgemeinen Geschäftsbedingung tun. Damit erteilt ein*e Nutzer*in Twitter die Erlaubnis „Ihre Inhalte weltweit verfügbar zu machen und dies auch Dritten zu ermöglichen“[11, Absatz 3] Diese Bedingung bietet Twitter die Möglichkeit veröffentliche Tweets für Dritte (Unternehmen, Organisationen oder Einzelpersonen) über eine [Application Programming Interface \(API\)](#) bereitzustellen und zu verkaufen. Im Folgenden wird beschrieben wie diese API genutzt wurde um eine Datenbasis an Tweets zu erhalten.

3.1 Tweets sammeln

Tweets die in der Vergangenheit veröffentlicht wurden, können über die sogenannte „Search API“ angefordert werden. Um die Suche einzuschränken kann die Sprache der Tweets und im Text vorkommende Wörter sowie ein Zeitraum definiert werden. Konstenlos stellt Twitter die Daten der letzten 7 Tage zur Verfügung; durch eine Hochrechnung wurde dies genutzt um eine Größenabschätzung der Anzahl an Tweets zu erhalten die monatlich zur Thematik der Covid-19 Pandemie veröffentlicht werden. So konnte eine Gesamtheit von *ca. 1.500.000* Tweets geschätzt werden. Um diese Methode in die Vergangenheit zu erweitern bedarf es den sogenannten „Full-Archive“ - Zugriff, der 1899\$ für 1.250.000 Tweets pro Monat kosten. Mit dem verfügbaren Budget der Arbeit war dies nicht zu realisieren.

Neben dem Zugriff auf in der Vergangenheit veröffentlichte Tweets, bietet Twitter außerdem die Möglichkeit in Echtzeit (sobald er veröffentlicht wurde) jeden Tweet zu erhalten, der vom Entwickler definierte Kriterien erfüllt. Dieser Dienst steht konstenlos zur Verfügung. Es wurde sich auf den sogenannte „Filtered Stream“ mit den Kriterien: 1. Ein Tweet soll das Schlagwort „Corona“ oder „Covid“ (und damit auch „Covid-19“) und 2. Der Tweet muss in deutscher Sprache verfasst sein, registriert um kontinuierlich jeden Tweet zu erhalten, der mit diesen Kriterien veröffentlicht wird.

Die gesammelten Tweets sollen nun in einer Datenbank gesichert werden. Für die Implementierung wurde [AWS](#) genutzt. Eine [EC2 - Instanz](#) ist für die Registerierung an der Twitter API und das Weiterleiten der Tweets verantwortlich. Als Speichermedium wurden [S3](#) -

Buckets verwendet. Um bei einem erhöhten Datenaufkommen keine Daten zu verlieren wurde zwischen die EC2 - Instanz und den S3 - Buckets ein Amazon Kinesis Data Firehose geschaltet, der sich durch eine hohe Datenaufnahme kennzeichnet. Dieser sammelt die einzelnen Tweets, fügt ca. 20 - 30 davon in einer Datei zusammen und speichert sie in den S3 - Buckets ab. Die Ordnerstruktur baut sich nach dem Zeitpunkt des Speicherns auf: *Jahr/Monat/Tag/Stunde/*. Eine Übersicht über die Architektur ist in Abbildung 3.1 gegeben.

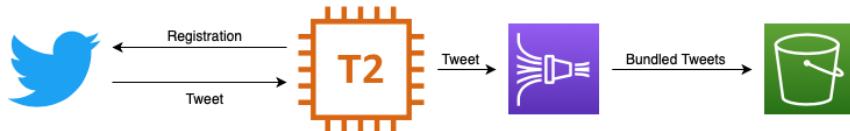


Abbildung 3.1: AWS - Architektur zur Tweet Akquisition

Der Prozess lief 5 Monate vom 01.12.2020 bis zum 30.04.2021. Dadurch konnten insgesamt ca. 10.000.000 Tweets gesammelt werden, also ca. 2.000.000 Tweets pro Monat.

3.2 Struktur eines Tweets

Ein Tweet ist ein JavaScript Object Notation (JSON) - Objekt, dass folgende, für die Arbeit relevanten, Informationen enthält:

1. Der Zeitpunkt, an dem dieser Tweet veröffentlicht wurde
2. Eine eindeutige Nummer, die diesen Tweet identifiziert
3. Eine Liste an verwendeten Hashtags
4. Informationen über den/die Nutzer*in der/die diesen Tweet veröffentlicht hat:
 - Eine eindeutige Nummer die diese*n Nutzer*in identifiziert
 - Ein einzigartiger gewählter Nutzernname
5. Ist ein Tweet ein Retweet, so enthält dieser das komplette Tweet-Objekt des Original-Tweet.

4 Sentimentanalyse

Die Sentimentanalyse ist ein Teilgebiet des [Text Mining](#) und beinhaltet die Klassifizierung von Aussagen und Meinungen in Texten in Kategorien wie z.B. „positiv“ und „negativ“[\[12\]](#), S. 113]. Wie in Abschnitt [2.1](#) beschrieben, soll der Versuch unternommen werden, die gesammelten Tweets auf ihre Meinung mittels Sentimentanalyse zu untersuchen und daraufhin in Gruppen mit gleichen Werten zu klassifizieren.

4.1 Technologieauswahl

Sentimentanalyse ist ein komplexer Vorgang, der Technologien aus dem Gebiet des [Natural Language Processing \(NLP\)](#), der Computerlinguistik und der Biometrik beinhaltet, weshalb hier auf schon existente Systeme zurückgegriffen wurde. Bestehende Tool unterscheiden sich in Qualität der Analyse, Differenzierung der Einteilung (können einzelne Emotion extrahiert werden?) und Preis. Mit einer Datenbasis von ca. 10.000.000 Texten fallen alle kostenpflichtigen System aufgrund des begrenzten Budget aus, weshalb die [OpenSource](#) Systeme „TextBlob“ und „NLTK - Vader“ verwendet wurden. Damit kann jedem Tweet folgende drei Werte zugeordnet werden.

1. *NLTK – Polarität* $\in [-1; 1]$

Ein Maß für die Stimmung des analysierten Textes
wobei $-1 \hat{=} \text{max. negativ}$, $1 \hat{=} \text{max. positiv}$

2. *TextBlob – Polarität* $\in [-1; 1]$

3. *TextBlob – Subjektivität* $\in [0; 1]$

Ein Maß für die Objektivität des analysierten Textes
wobei $0 \hat{=} \text{max. objektiv}$, $1 \hat{=} \text{max. subjektiv}$

Beide Systeme sind optimiert für englische Texte; bedeutet, alle gesammelten Tweets müssen vor der Analyse übersetzt werden. Im Bereich der maschinellen Übersetzung steigern sich die Kosten der Prozessierung der Datenbasis über das Budget weshalb eine Übersetzung aller Tweets nicht möglich war. Allerdings konnten Tweets von 4 Tagen (01 - 04 März 2021) übersetzt werden um anschließend eine Sentimentanalyse durchzuführen.

4.2 Analyse

Die Datenauswahl wurde mit den in Abschnitt 4.1 genannten Tools analysiert. Dabei zeigten sich die Probleme der Sentimentanalyse:

Probleme der Erkennung von doppelten Verneinung oder Ironie treten auf oder von Menschen intuitiv als negativ betrachtete Tweets werden falsch eingruppiert. So z.B. an diesem Beispiel:

„Wir brauchen keine „Maßnahmen“ wegen einem Virus von vielen! Wir haben ein Immunsystem! Wir wollen raus aus dieser Matrix des Corona-Hoax! STOP THE GREAT RESET, er dient nicht unserer Gesundheit! Ganz bestimmt nicht!“[13]

Dieser Tweet beinhaltet eine von Menschen eher als negativ aufgefasste Aussage. Trotzdem bewertet TextBlob die übersetzte Version mit einer Polarität von 0.659 und NLTK sogar mit 0.8.

Des Weiteren belastet die Covid-19 Pandemie die meisten Nutzer*innen, sodass ein allgemein eher negativer Tonus in den Texten herrscht; das macht eine Unterscheidung schwierig.

Eine Einteilung in positiv und negativ ist oft nicht ausreichend, eine Unterscheidung z.B. zwischen politischen Gruppierungen ist nicht möglich. Dafür bedarf es höher entwickelte Systeme die Aussagen bezüglich des Inhalts differenziert kategorisieren können.

Nichtsdestotrotz, wurde nach Gruppen von Nutzer*innen gesucht, die sich in ihren Sentimentwerten ähneln. Dafür wurde für jede*n Nutzer*in der Durchschnittswert seiner/ihrer Sentimentwerte berechnet und diese Nutzer*innen anschließend als Punkte in einem dreidimensionalen Graph dargestellt.

In Abbildung 4.1 liegt eine unstrukturierte Verteilung vor, es sind keine offensichtlichen

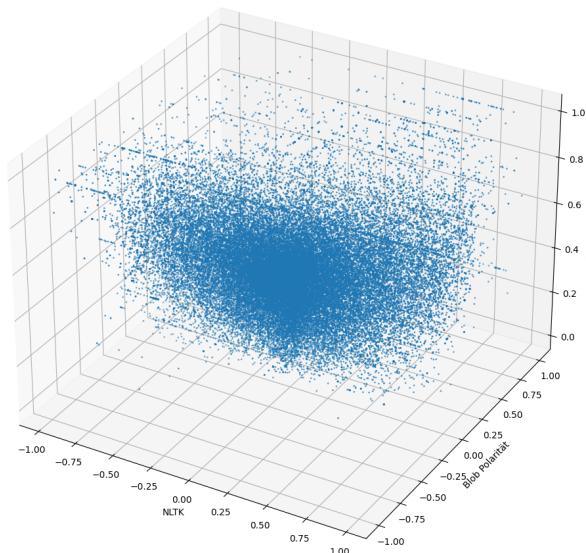


Abbildung 4.1: Verteilung der Nutzer*innen anhand der durchschnittlichen Sentimentwerte ihrer Tweets

Gruppierung erkennbar. Aufgrunddessen und weil der Datensatz wegen der Übersetzungsschranke nur schwer erweiterbar ist, sowie vor dem Hintergrund der oben genannten Ungenauigkeiten wurde der Versuch der Eingruppierung anhand von Sentimentdaten beendet.

5 Clustering Retweets

Wie in Abschnitt 3.2 beschrieben enthält ein **Retweet** die Informationen des/der Nutzer*in der/die ihn veröffentlicht hat, sowie die Informationen des/der Nutzer*in der/die den Original-Tweet veröffentlicht hat. Eine Beziehung kann also zwischen diesen beiden Nutzer*innen hergestellt werden. Außerdem kann durch die Anzahl an **Retweets** der Beziehung eine Gewichtung zugeschrieben werden. Durch Analyse aller **Retweets** kann so ein Netz von Beziehungen zwischen Nutzern*innen aufgebaut werden. Werden in diesem Netz Nutzer*innen identifiziert, die sich häufig untereinander **retweeten** so können diese einer Gruppe zugeordnet werden. Im Folgenden soll der Prozess solche Gruppen zu finden, dargestellt und die Ergebnisse präsentiert werden.

5.1 Daten vorbereiten

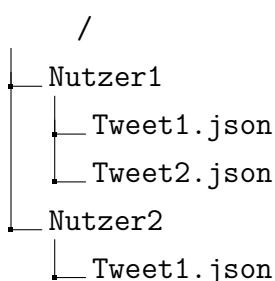
Zur Verarbeitung und Analyse der Daten wurde keine Datenbank verwendet sondern ausschließlich mit Dateioperationen und Ordnerstrukturen gearbeitet. Somit können die Daten in jeder beliebigen Art und Strukturierung abgelegt, Metainformationen hinzugefügt und jeder Schritt des Prozesses zwischengespeichert werden. Um die Tweets auf Beziehung zwischen Nutzer*innen zu schließen wurden folgende vorbereitende Schritte ausgeführt:

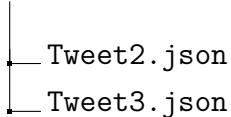
1. Daten aus **S3 - Bucket** herunterladen
2. Unrelevante Informationen filtern

So wird ein übersichtliches Format erreicht und Speicherplatz gespart. Das resulierende Tweet-Objekt entspricht der Beschreibung aus Abschnitt 3.2

3. Tweets in einer Ordnerstruktur nach Nutzer sortieren

Als eindeutige Identifikation wurde der in Abschnitt 3.2 genannte Nutzernname gewählt. Es entsteht folgende Ordnerstruktur:





4. Metadaten erstellen

Nun soll für jede*n Nutzer*in Metadaten erstellt werden, von wem und wie oft er von anderen Nutzer*innen [geretweetet](#) wurde. Dafür werden alle Nutzer*innen und ihre Tweets analysiert; ist einer davon ein [Retweet](#) eines andere*n Nutzer*in so wird das, zusammen mit der Anzahl, in den Metdaten desjenigen Nutzers, der [geretweetet](#) wurde, vermerkt. Für jeden Nutzer entstehen damit Metadaten der Form:

```

1   {
2     "i_was_retweeted_by": {
3       "RSplettsto": 2,
4       "Roxana50189220": 2,
5       "RobbyTipps": 14,
6       "RafaelOsswald": 2,
7       "rbbinfooradio": 16
8     },
9     "i_am_retweeting": {
10      "tagesthemen": 6,
11      "ARD_BaB": 3,
12      "aktuelle_stunde": 1,
13      "gudrun_engel": 1
14    }
15  }
  
```

Listing 5.1: Metdaten des Nutzer: „tagesschau“

Dieser Prozess besitzt eine Komplexität von $\mathcal{O}(n)$ mit $n = \text{Gesamtanzahl an Tweets}$.

5.2 Influencer identifizieren

Bei der Untersuchung der Metadaten von März 2021, ist aufgefallen, dass 31,54% aller [Retweets](#) auf die 100 meist geretweeteten Nutzer entfällt (0,04%). Genauer folgt die Verteilung einem inversen Potenzgesetz:

$$\begin{aligned}
 N_{\text{retweets}} &= b * x^{-m} \\
 m &= 0.65 \\
 b &= 39,215
 \end{aligned} \tag{5.1}$$

wobei x der Rang, m die Steigung und b der Skalierungsfaktor oder die Anzahl der Retweets des Influencers an Rang 0 ist.

Daraus wurde der Schluss gezogen, dass Beziehungen zwischen „kleineren“ Nutzer*innen

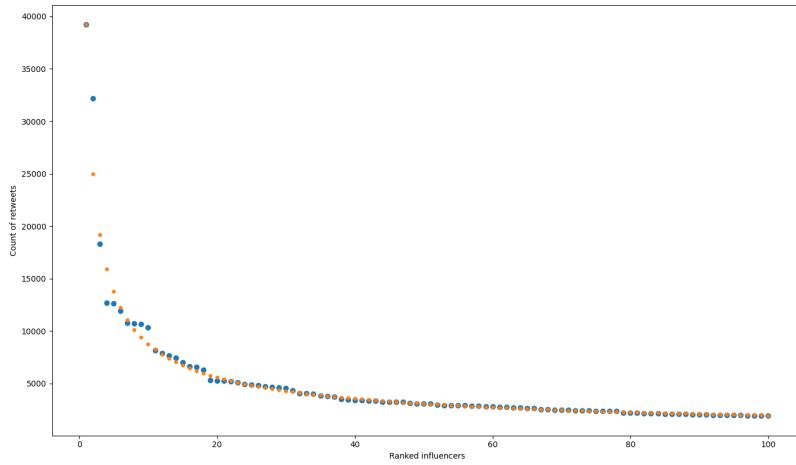


Abbildung 5.1: Verteilung der Anzahl an Retweets (blau) und das inverse Potenzgesetz (orange)

vernachlässigbar sind und nur Verbindungen zu den 100 meist geretweeteten Nutzer*innen von Bedeutung sind. Diese werden im Folgenden „Influencer“ genannt. Auch kann davon ausgegangen werden, dass Beziehung zwischen „kleineren“ Nutzern*innen repräsentiert sind, da zwei Nutzer*innen die sich gegenseitig retweeten, wahrscheinlich auch eine Beziehung zum gleichen Influencer haben. Diese Methodik vermindert die Anzahl an Verbindungen und führt zu einer effizienteren Verarbeitung der Daten sowie später zu einem übersichtlicheren Graphen (siehe Abschnitt 5.3) Die Anzahl an Influencern ist wählbar und führt zu verschiedenen Ergebnisse, allerdings hat sich 100 als ein passender Parameter herausgestellt.

5.3 Nutzergruppen aggregieren

Influencer I und Nutzer*innen N können als Knoten $K = I \cup N$, Beziehungen als gewichtete Kanten E eines Graphen $G = (K, E)$ betrachtet werden. Dieser kann als ungerichtet definiert werden, da Kanten immer zu einem Influencer zeigen. E ist hierbei eine Menge an Paaren der Form $(i \in I, n \in N, g)$ wobei g die Anzahl der Retweet darstellt. Dieser Graph kann visualisiert werden. Genutzt wurde dafür Python3, Matplotlib und eine NEATO Implementierung als Layoutalgorithmus (siehe [14]).

Abbildung 5.2 zeigt einen solchen Graph mit Daten aus 4 Tagen, den drei Influencern in blau (Karl_Lauterbach, reitschuster, Volksverpetzer) und allen Nutzer*innen die diese geretweetet haben in schwarz. Die Berechnung dieses Layout braucht auf einen handelsüblichen PC mit Linux 35,6 Sekunden. Mit Daten von mehr als 4 Tagen oder mit mehr als 3

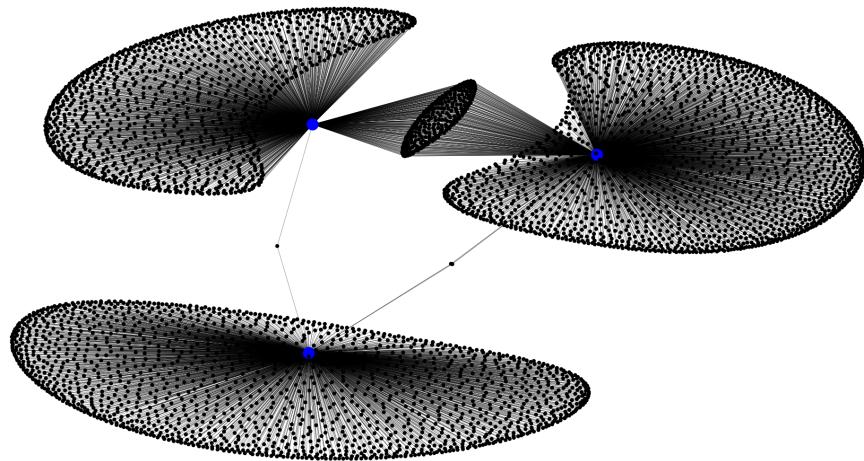


Abbildung 5.2: Graph mit drei Influencern (blau)

Influencern explodiert die Berechnungszeit.

Offensichtlich erkennbar ist jedoch, dass es für jede Influencer eine Menge an Nutzer*innen N_{i^x} gibt, die nur diese Influencer [geretweetet](#) haben. Analog dazu gibt es eine Menge an Nutzer*innen $N_{i^1 i^2}$ die sowohl Influencer i^1 und Influencer i^2 retweeten. Diese Mengen können als „Superknoten“ betrachtet werden, der alle Nutzer*innen repräsentiert, die z.B. sowohl i^1 als auch i^2 [geretweetet](#) haben. Es gilt also für einen Superknoten s :

$$s_{i^n \dots i^m} = \{n \in N \mid \forall e(n, i^n \dots i^m) \in E\} \quad (5.2)$$

Die Gewichte g der Kanten werden aufaddiert. Damit lässt sich der Graph auf maximal

$$|K| = |S| + |I| = \sum_{n=1}^{|I|} \binom{|I|}{n} + |I| = 2^{|I|} - 1 + |I| \quad (5.3)$$

Knoten beschränken (mit $|S|$ der Menge aller möglichen Superknoten). Abbildung 5.3 zeigt den selben Graph wie 5.2 mit zu Superknoten aggregierten Nutzern. Die Berechnung des Layout konnte dadurch auf 0.16 Sekunden reduziert werden. Auch dieser Prozess besitzt eine Komplexität von $\mathcal{O}(n)$ mit $n = \text{Gesamtzahl an Nutzern}$.

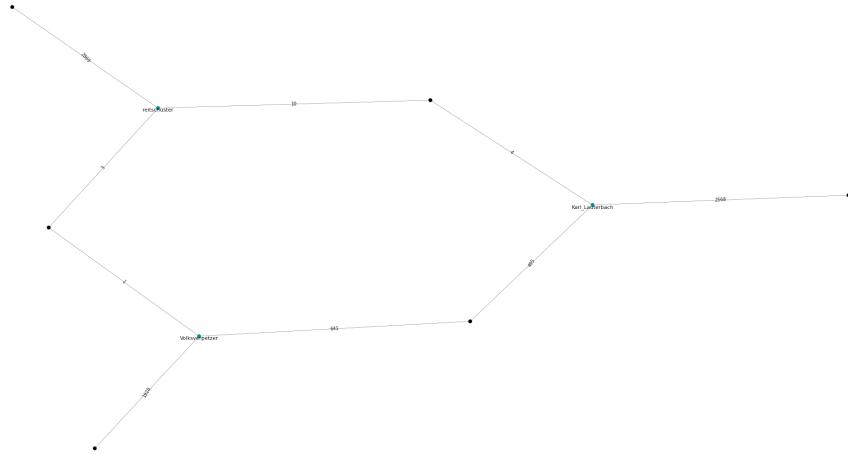


Abbildung 5.3: Graph mit aggregierten Superknoten

5.4 Kanten filtern

Wie in Formel 5.2 zu sehen, steigt die Anzahl an möglichen Superknoten exponentiell mit steigender Zahl an Influencern. Bei $|I| = 100$ könnten bis zu $1.26 * 10^{30}$ Superknoten existieren. Das ist für jegliche Berechnung oder Darstellung unpraktikabel. Deshalb wird auch hier eine Komplexitätsverringerung durch Filtern unrelevanter Information vorgenommen, indem alle Kanten von einem Superknoten zu einem Influencer gelöscht werden, die unter einem Schwellwert liegen. Es wird davon ausgegangen, dass bei einem niedrigen Kantengewicht die Nutzer*innen nur eine schwache Beziehung zu dem entsprechenden Influencer haben und diese damit für eine spätere Gruppierung irrelevant ist. Es gilt also mit dem Schwellwert T :

$$E_{gefiltert} = \{(s, i, g) \in E | g > T\} \quad (5.4)$$

Wurde eine Kante (z.B. (s, i^3)) von einem Superknoten $s_{i^1 i^2 i^3}$ gelöscht, so ändert sich dieser zu $s_{i^1 i^2}$. Nun ist es allerdings möglich, dass der Knoten $s_{i^1 i^2}$ bereits existiert, was zu einer Verdopplung der eigentlich gleichen Kante führt. Dies kann vermieden werden indem nach dem Löschen einer Kante überprüft wird ob der so entstandene Superknoten bereits existiert. Falls ja, werden diese zusammengeführt indem die jeweiligen Kantengewichte addiert werden. Des Weiteren werden diejenigen Superknoten gelöscht, die Nutzer repräsentierten, die nur einen Influencer **geretweetet** haben, da sie für eine Eingruppierung anhand von Beziehung keine Relevanz haben.

Dieser Prozess besitzt eine Komplexität von $\mathcal{O}(n^2)$ mit $n =$ Gesamtanzahl der Superknoten, da im schlechtesten Fall für jeden veränderten Knoten, dieser mit alle bereits bestehenden Knoten auf Gleichheit überprüft werden muss.

5.5 Kantengewichte normalisieren

Wie in Abschnitt 5.2 beschrieben folgt die Verteilung der Retweets auf die Influencer einem inversen Potenzgesetz. Der Natur eines inversen Potenzgesetzes folgend, haben die niedrigen Ränge eine deutlich höhere Anzahl von Retweets, was es schwierig macht, einen Schwellenwert zu finden, der einerseits die Komplexität in Bezug auf Verbindungen mit dem niedrigrangigen Influencer minimiert, aber trotzdem Gruppen erhält, die aus höherrangigen Influencern bestehen. Um diesem Problem entgegenzuwirken, werden die Kantengewichte mit dem Zehnerlogarithmus des Rangs des jeweiligen Influencers multipliziert, bevor der Schwellenwert angewendet wird. Dies normalisiert die Retweets nicht vollständig, da ansonsten [Retweets](#) von höherrangigen Influencern überbewertet würden. Trotzdem verringert es die Dominanz von niederrangigen Influencern.

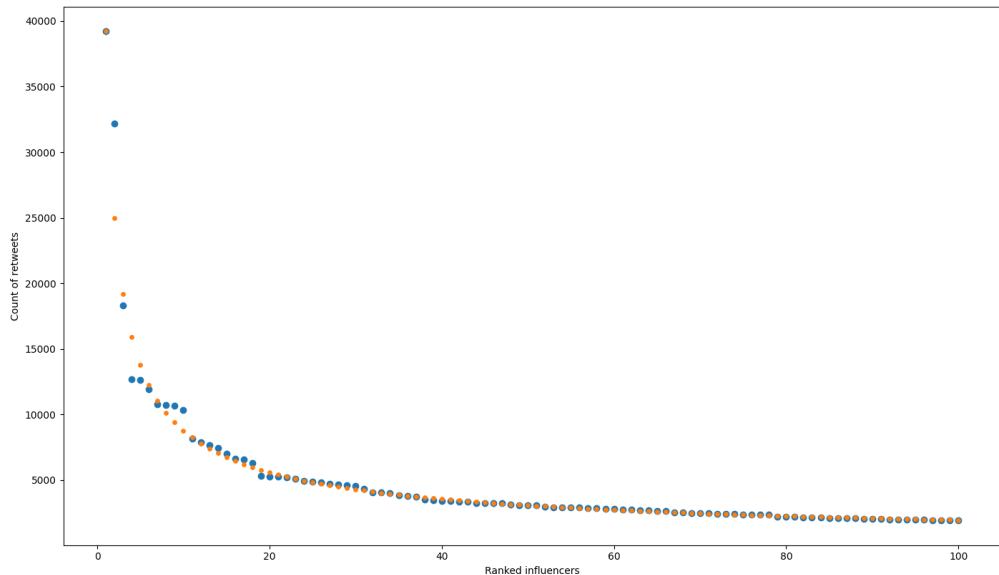


Abbildung 5.4: The distribution of retweets (blue) and the power-law from equation 4(orange)

5.6 Clustering

Das Ergebnis aller bisherigen Prozessschritte mit den Daten von März 2021 (2,955,282 Tweets, 260,954 Nutzer), 100 Influencern und einem Schwellenwert von 61 ist in Abbildung 5.5 zu sehen und wurde innerhalb von 2 Stunden und 58 Minuten berechnet.

Als Mensch kann man ein großes zentrales Cluster, eins das mit diesem verbunden ist

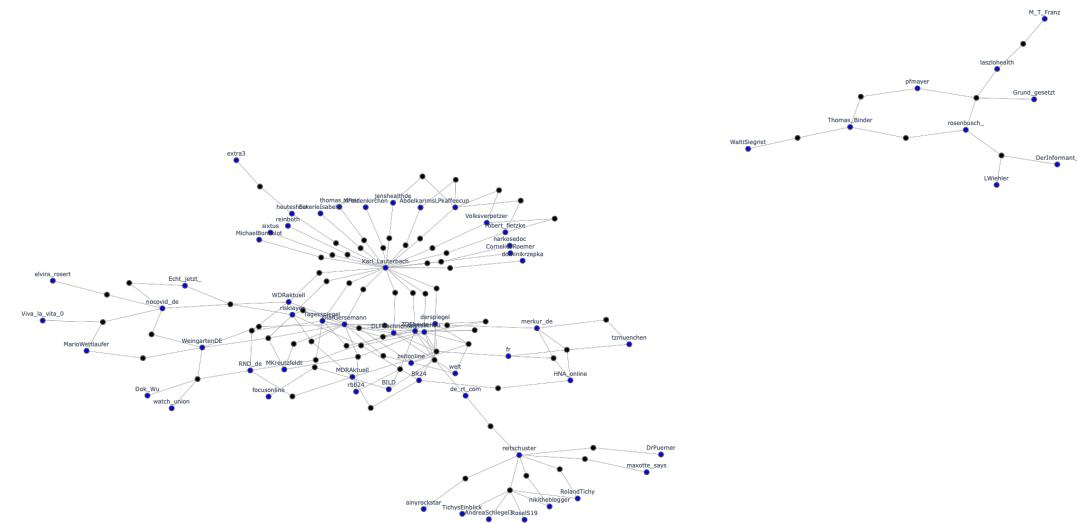


Abbildung 5.5: Ergebnisgraph März 2021

(unten) und ein separates Cluster (oben links) erkennen. Im Folgenden wird dargelegt, wie diese Cluster automatisch erkannt werden.

Zu einem Cluster können in diesem Kontext alle Knoten gezählt werden, die dicht miteinander verbunden sind; das heißt sie stehen über einige weitere Knoten in Beziehung. Da eine Anzahl an Cluster vorher nicht festgelegt werden kann und es kein geometrischen Raum gibt (nur die Verbindungen zählen, die Positionen der Punkte irrelevant) wurde sich hier für eine abgewandelte Form des DBSCAN - Algorithmus entschieden (siehe [15]). Als Knoten werden nur Influencern betrachtet da die Superknoten nur die Verbindung zu weiteren Influencern darstellen. Die Nachbar eines Influencer sind alle Influencern die mit dem Ausgangsinfluencer über ein Superknoten verbunden sind (siehe Abbildung 5.6). Eine Mindestdistanz ϵ wird nicht benötigt, da alle Verbindungen die dadurch nicht gezählt würden, bereits durch den in Abschnitt 5.4 definierten Schwellwert aussortiert wurden. Da mit diesem Ansatz jede Verbindung zu einem weiteren Influencer als Nachbar in den DBSCAN eingeht, werden Cluster die nur leicht verbunden sind (wie in Abbildung 5.5 das Zentrale und das unten Positionierte) als ein Cluster definiert. Um hier eine Trennung zu schaffen, werden nur Knoten als Nachbarn gezählt die noch nicht dem aktuellen Cluster zugeordnet wurden wie in Listing 5.2.

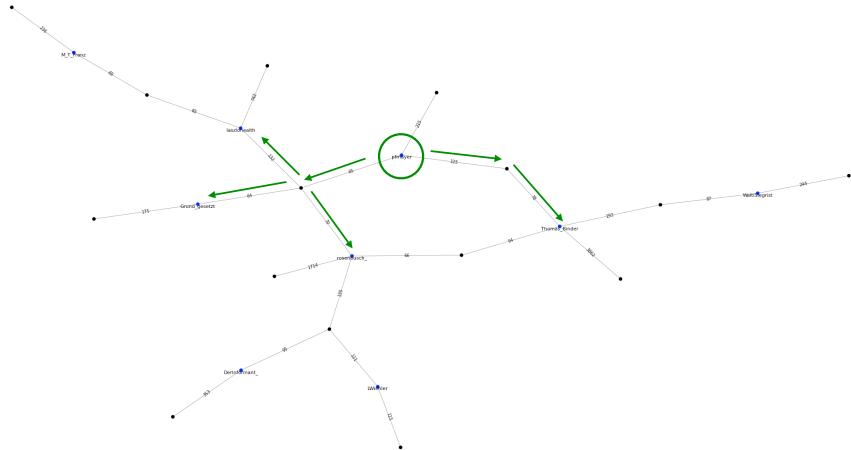


Abbildung 5.6: Nachbarn eines Influencer über verbundene Superknoten

```

1 def getNeighbors(node):
2     all_neighbors = []
3     supernodes = G.neighbors(node_name)
4     for supernode in supernodes:
5         real_neighbors = G.neighbors(supernode)
6         for real_neigbor in real_neighbors:
7             #Dieser Knoten wurde schon erreicht oder ist schon im
8             #aktuellen Cluster
9             if(real_neigbor in cluster_nodes or neighbor_name in
10                all_neighbors):
11                 continue
12             all_neighbors.append(real_neigbor)
13     return all_neighbors

```

Listing 5.2: Abwandlung der DBSCAN Nachbarfunktion

Die Mindestanzahl an Nachbar damit ein Knoten zum Kernknoten wird, ist wählbar, allerdings hat sich ein Wert von 2 bewährt. Diese Wahl wird in Abschnitt diskutiert. Auf den oben vorgestellten Datensatz angewendet, braucht das Berechnen der Cluster 2,3 Minuten und liefert das in Abbildung 5.7 dargestellte Ergebnis (in größerer Form im Anhang).

5.7 Einblick in die Cluster

Um die anfängliche Frage zu beantworten, ob Gruppierungen von Menschen mit ähnlicher Sichtweise auf die Covid-19 Pandemie und die dazugehörigen Regelungen identifiziert werden können, soll nun ein kurzer Überblick über die Charaktere der einzelnen Cluster

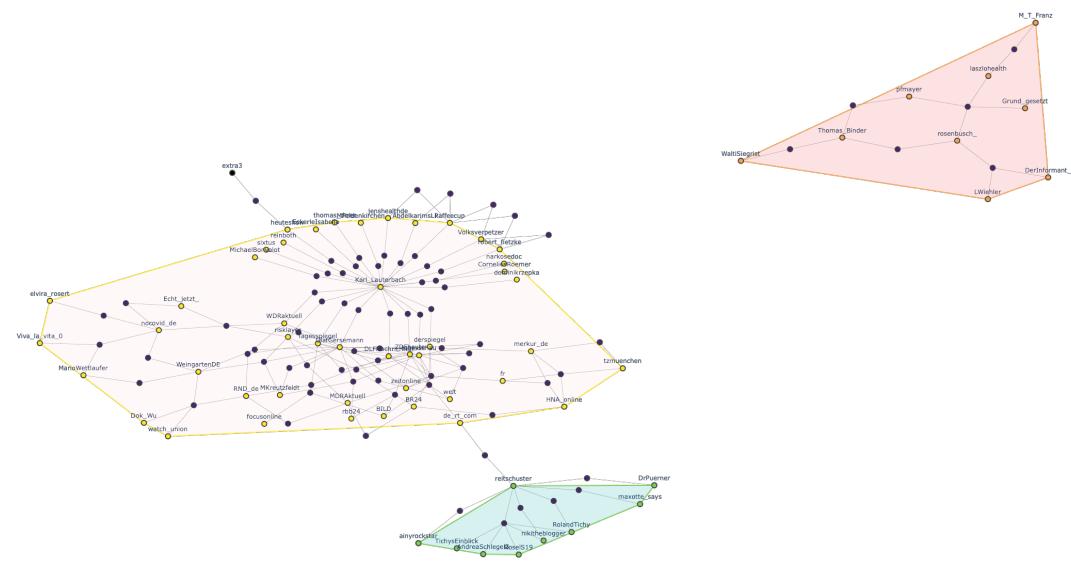


Abbildung 5.7: Graph mit Markierungen um die gefundenen Cluster

gegeben werden. Das gelb markierte Hauptcluster besteht aus öffentlich-rechtlichen und privaten Nachrichtenagenturen wie „ZDF“, „tagesschau“, „derspiegel“ oder „BILD“ sowie dem am häufigsten [geretweetet](#) Nutzer im Datensatz: „Karl_Lauterbach“, der, aufgrund seines Hintergrunds in der Sozialdemokratischen Partei (SPD) und als Mediziner, in der öffentlichen Debatte über die Pandemie eine präsente Figur ist.

Einige Influencer aus dem grünen Cluster sind: „maxotte_says“ (Max Otte) ehemaliger Vorsitzender der „Werteunion“, eine Fraktion innerhalb der Christlich Demokratischen Union (CDU), bekannt dafür, konservativere Positionen zu vertreten, „ainyrockstar“ eine rechter Journalistin [16] und „RolandTichy“ ehemaliger Herausgeber von „Impuls“ und „Euro“.

Ein Blick in das obere rechte, orangefarbene Cluster: „rosenbusch_...“ (Henning Rosenbusch) ist ein unabhängiger Journalist, der sich für den „Schweden-Weg“ einsetzt, „laszlohealth“ (unbekannt) mit einem gepinnten Tweet: „*Corona ist eine abartige Mischung aus Glaubens- & Polit-Krieg mit allen Mitteln geworden. Die Maske das Symbol der Zugehörigkeit. PCR-Massen-Testung die Waffe. Sachlichkeit, Meinungsfreiheit, normalen Umgang gibt es nicht mehr.*“[17] und „Thomas_Binder“, dessen Account gesperrt wurde, ist ein schweizer Arzt, der eine Anti-Regulierungs-Position vertritt und verhaftet und in die Psychiatrie eingeliefert wurde wegen mutmaßlicher Drohungen gegen Politiker.[18].

5.8 Rückschluss auf Nutzer*innen

Um nun nicht nur die Influencer eines Cluster sondern auch die Nutzer*innen, die diese geretweetet haben zu identifizieren, wurde der Prozessschritt des Aggregierens (Abschnitt 5.3) erweitern, sodass für jeden Supernutzer gespeichert wird, welche Nutzer ihn ausmachen. Wurden nun die Cluster erstellt, können diesem Cluster auch alle Supernutzer zugeordnet werden, die mindestens eine Verbindungen zu einem Influencer aus diesem Cluster besitzen. Aus den zugehörigen Supernutzer können nun auch alle „normalen“ Nutzer*innen des Cluster extrahiert werden.

5.9 Bestimmung der Themen innerhalb der Cluster

Während in Kapitel 5.7 die Cluster anhand einzelner Individuen analysiert und damit in einen thematischen Zusammenhang gebracht wurden, soll im folgenden Anhand einer Wortanalyse das Thema innerhalb der Cluster bestimmt werden.

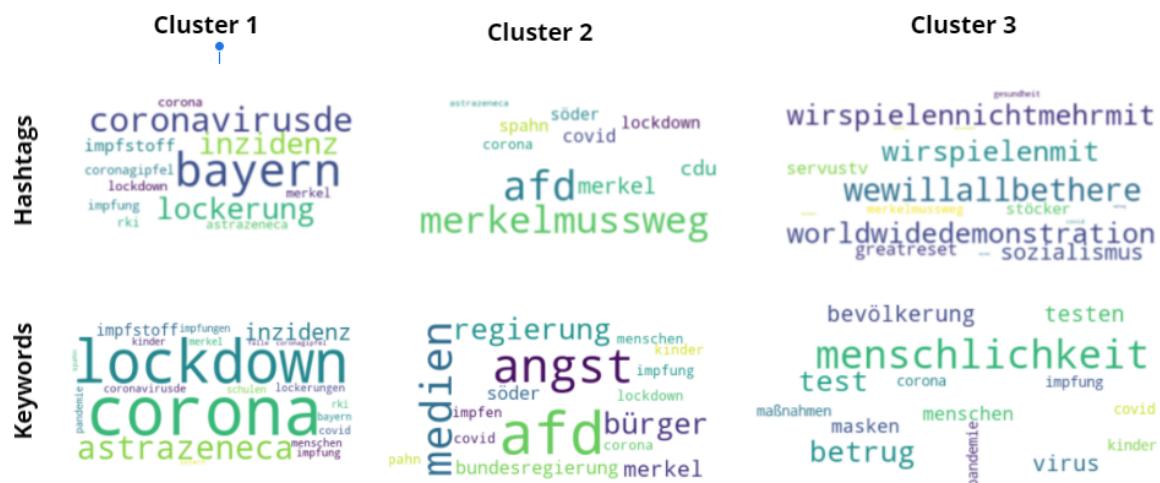


Abbildung 5.8: Themen der Cluster als Wortwolken

Um zu bestimmen, welche Worte für ein Cluster *besonders* sind, wird die relative Häufigkeit des Wortes innerhalb des Clusters mit der relativen Häufigkeit des Wortes im gesamten Datensatz verglichen. Das sich daraus vergebene Verhältnis beschreibt, wie viel mal häufiger ein Wort innerhalb des Clusters als außerhalb dessen verwendet wird. Nun kann man eine Wortwolke erstellen, in welcher die Wörter entsprechend dieses Wertes skaliert werden. Der Prozess ist in Abbildung 5.9 visualisiert. Abbildung 5.8 zeigt das Ergebnis dieser Analyse, jeweils für die Schlüsselwörter und die Hashtags. Die Nummerierung der Cluster entspricht der Reihenfolge der Aufzählung in Kapitel 5.7: Cluster 1 entspricht dem gelben, Cluster 2 dem grünen und Cluster 3 dem orange eingefärbten

Clustern. Die Auswahl der zu betrachtenden Schlüsselwörter wird in Kapitel 6.2 besprochen.

Auffallend ist, dass die so gefunden Wörter sich teilweise sogar sehr gut den (politischen) Leitlinien der aufgeführten Personen entsprechen.

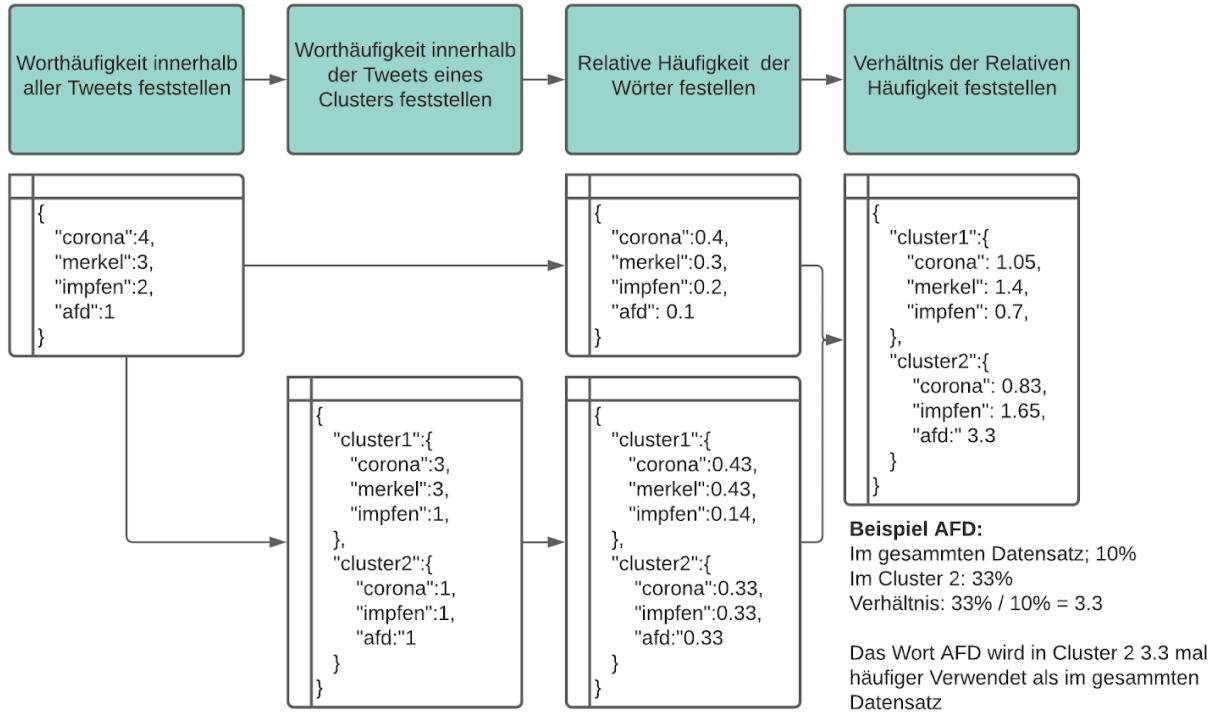


Abbildung 5.9: Bestimmung der Themen innerhalb der Cluster

6 Gruppierung der Nutzer*innen anhand der verwendeten Wörter

Es scheint offensichtlich, dass wenn wir über ein bestimmtes Thema diskutieren auch bestimmt Wörter benutztten, um uns überhaupt mit dem Thema auseinander setzen zu können. Eine Diskussion über den Klimawandel scheint schwer praktikabel, ohne auch nur einmal das Wort "Klimawandel" zu benutztten. Diese, schon fast als Axiom betrachtbare Tatsache soll nun als Fundament für folgende These gelten: Bei Menschen, welche über ein bestimmtes Themen sprechen, deckt sich die Auswahl der verwendeten Wörter eher als bei Menschen, welche präferiert über andere Themen diskutieren. Diese Hypothese legt also nahe, dass Texte, welcher sich mit dem Klimawandel auseinandersetzt sich untereinander in der Wortwahl weniger unterscheiden als wenn man sie mit einem Text über den Nahostkonflikt vergleicht.

6.1 Berechnung der Ähnlichkeit von Texten

Um die Ähnlichkeit der Worte nun vergleichen zu können, muss zunächst ein Kriterium oder eine Maßzahl gefunden und anschließend berechnet werden, anhand der die Texte verglichen werden können. Distanzfunktionen kennt man vor allem aus der Geometrie. Hier ist ein häufiger Vertreter die Euklidische Distanz.

Eine hierfür geeignete Methodik hierzu ist die Berechnung der Kosinus-Ähnlichkeit [19, 2]. Sind die Merkmale zweier zu vergleichenden Merkmalsträger als n-dimensionaler Vektor vorhanden, so bestimmt die Kosinus-Ähnlichkeit den Winkel zwischen diesen beiden Vektoren. Die Euklidische-Distanz jedoch bestimmt die Abstand zwischen den beiden Enden der Vektoren. Außerdem ist die Kosinus-Ähnlichkeit schneller zu berechnen, da keine nummerische Berechnung einer Wurzel erfolgen muss.

$$\begin{aligned} \text{Berechnung der Kosinus-Ähnlichkeit} \quad & \cos(\alpha) = \frac{x \cdot y}{|x| \cdot |y|} \\ \text{Berechnung der Euklidischen Distanz} \quad & d_{xy} = \sqrt{\sum_{i=0}^n |x_i - y_i|^2} \end{aligned} \tag{6.1}$$

w aus W_{acc}	Marc	isst	gerne	Bananen	kauft	Lea
$w \in W_A$	ja (1)	ja (1)	ja (1)	ja (1)	nein (0)	nein (0)
$w \in W_B$	nein (0)	nein (0)	ja (1)	ja (1)	ja (1)	ja (1)

Tabelle 6.1: Überführung von Wortmengen zu Vektoren für einzelne Nutzer*innen

Die Formeln (6.1) zeigen die Berechnung der Kosinus-Ähnlichkeit und der Euklidischen-Distanz zweier Vektoren im n-dimensionalen Raum. Um nun die Texte miteinander vergleichen zu können müssen für diese zunächst in Merkmalsvektoren überführt werden.

Da die Merkmale in diesem Kontext die verwendeten Wörter sind, braucht es einen Vektor, welcher beschreibt, ob ein bestimmtes Wort in einem Text enthalten ist oder nicht. Dazu müssen zuerst alle Wörter bestimmt werden, welche in der Summe W_{acc} aller verwendeten Texte verwendet werden. Diese bilden die Summe aller möglichen Merkmale. Anschließend wird überprüft, welcher der Worte, welche über alle Texte hinweg existieren in den einzelnen Texten wieder gefunden werden. Dies soll nun anhand des Satzes A „Marc isst gerne Bananen“ und des Satzes B „Bananen kauft Lea gerne“ dargestellt werden. Die Menge W_x stellt die Menge der im Satz enthaltenen Wörter dar.

$$\begin{aligned} \text{Satz A } W_A &= \{\text{Marc, isst, gerne, Bananen}\} \\ \text{Satz B } W_B &= \{\text{Bananen, kauft, Lea, gerne}\} \\ \text{Vereinigung } W_{acc} &= W_A \cup W_B = \{\text{Marc, isst, gerne, Bananen, kauft, Lea}\} \end{aligned} \tag{6.2}$$

Aus der Formel (6.2) ergeben sich die Wortmengen der einzelnen Sätze sowie die Vereinigung aller Wortmengen. Um diese Wortmengen zu überprüfen wird nun für jedes Wort w in W_{acc} geprüft, ob w in der Wortmenge des Satzes W_x enthalten ist.

[h]

Somit ergibt sich aus den Zeilen der Tabelle für die Beiden Sätze A und B folgende Merkmalsvektoren

$$\begin{aligned} \text{Merkmalsvektor A } v_A &= \{1, 1, 1, 1, 0, 0\} \\ \text{Merkmalsvektor B } v_B &= \{0, 0, 1, 1, 1, 1\} \end{aligned} \tag{6.3}$$

Auf diese lässt sich nun die Kosinus-Ähnlichkeit sowie die euklidische Distanz für die beiden Beispielvektoren anwenden. Zu beachten ist, dass bei der Kosinus-Ähnlichkeit der

Wert höher ist, desto ähnlicher die Vektoren sind, während bei der Kosinus-Distanz der Wert niedriger wird, je ähnlicher zwei Vektoren sind.

$$d_{cos} = \frac{v_A \cdot v_B}{|v_A| \cdot |v_B|} = 0.5$$
$$d_{euk} = \sqrt{\sum_{i=0}^n |v_{Ai} - v_{Bi}|^2} = 2 \quad (6.4)$$

6.2 Aufbereitung von Texten zur Ähnlichkeitsanalyse

Ein großes Problem bei der semantischen Analyse von Texten ist der gering Informationsgehalt pro Wort. Die meisten Worte innerhalb eines Satzes dienen dem Menschen zwar zum besseren Verständnis und Einordnung des Textes, dienen aber nicht wirklich dem Inhalt des Textes. So sind zur Einordnung des Themas des Satzes „Nun hat die Bundesregierung einen Aktionsplan vorgelegt“ nur die Wörter „Bundesregierung“ und „Aktionsplan“ von entschiedener Bedeutung. Wichtig für die Qualität der Berechnung einer Distanz zwischen zwei Texten ist also die Auswahl der Wörter, welche zum Merkmausvektor beitragen. Eine Methodik zur Auswahl dieser soll nun vorgestellt werden.

6.2.1 Entfernen von Stoppwörtern

Um das Rauschen in einem Satz nun zu verringern, muss man zunächst alle Wörter herausfiltern, welche nicht maßgeblich zur Semantik des Satzes beitragen. Diese Art der Wörter, unter welchen vor allem Personal- und Possessivpronomen fallen, werden allgemein Stoppwörter genannt. Im Kontext dieser Arbeit wurde eine sehr erweiterte Liste an Stoppwörtern verwendet, welche sich unter folgenden Link finden lässt: https://github.com/solariz/german_stopwords/.

Zudem sind durch die Regeln der Grammatik eigentlich gleiche Wörter innerhalb unterschiedlich Sätze in unterschiedlichen Kasus wiederzufinden. Auch unterschiedliche Tempora führen zu einer Abänderung der Wortstämme. So sollten die Wörter „demonstrieren“ und „demonstrierte“ zusammengeführt werden.

6.2.2 Lemmatisierung der Schlüsselworte

Um in einen Text also weiter das Rauschen zu verringern, müssen die einzelnen Wörter auf ihren Wortstamm zurückgeführt werden, also die original flektierten und abgeleiteten Wörter zu Ihrer Grundform zurückgeführt werden. Hierfür gibt es zwei unterschiedliche

Methodiken: Beim Stemming wird über Heuristik der Suffix der Wörter entfernt. Dies funktioniert zwar beim Wort *Pflanzen*, welches durch das entfernen des n auf den korrekten Singular *Pflanze* zurückgeführt werden kann, bei *Bäumen* stoßt man jedoch schon auf Probleme. Eine bessere Methodik ist daher die Lemmatisierung. Hier wird die Rückführung anhand einer Datenbank realisiert. In dieser würde so für das Wort *Häuser* der korrekte Singular *Haus* zu finden sein. Da präzisere Ergebnisse der Methodik der Lemmatisierung immanent sind, soll diese verwendet werden. Für Deutsche Texte gibt es hierfür unterschiedliche Bibliotheken für Python, welche im Nachfolgenden verglichen werden sollen:

Eine sehr gute Vorarbeit findet sich in [20]. Hier werden mehrere deutsche Bibliotheken zur Lemmatisierung aufgeführt und verglichen. Der Autor schließt hier durch einen Versuchsaufbau auf drei als sehr gut geeignete Bibliotheken zur Lemmatisierung mit den Namen HanTa, IWNLP und SpaCy. Von diesen soll nun SpaCy und HanTa verglichen werden, IWNLP ließ sich leider schwer installieren.

Ein guter Lemmatisierungsalgorithmus zeichnet sich dadurch aus, dass möglichst viele unterschiedliche Formen eines Wortes auf die selbe Grundform gebracht werden. Das führt dazu, dass die Anzahl unterschiedlicher Worte in einem Text zurück geht. Je besser die Lemmatisierung, desto weniger unterschiedliche Worte. Somit soll verglichen werden, wie viele unterscheidbare Wörter in einem Text überbleiben, wenn dessen Wörter durch den HanTa oder SpaCy lemmatisiert wurden. Da jedoch die Anzahl an unterschiedlicher Wörter eines Textes auch mit der Länge des Textes zusammenhängt, soll zur besseren Darstellung verglichen werden, um wie viel Prozent die ursprünglichen unterscheidbaren Wörter des Text reduziert werden. Hat Text A also vor Lemmatisierung 10 unterscheidbare Wörter und danach 9, so wurden diese um 10% reduziert. Abbildung 6.1a zeigt diese Werte für unterschiedliche Textlängen. Als Textgrundlage wurde der Roman Steppenwolf von Hermann Hesse verwendet. Aus der Abbildung ist zu erkennen, dass die Bibliothek HanTa bessere Ergebnisse liefert als SpaCy.

Ein weiteres wichtiges Auswahlkriterium ist aufgrund der großen Menge der zu Verarbeitenden Daten die Performanz der Bibliothek. Diese wird in 6.1b abgebildet. Aus der Abbildung lässt sich schließen, dass die Bibliothek HanTa performanter ist als die zu Vergleichende. Da diese somit performanter ist und zudem bessere Ergebnisse liefert, soll diese für das Projekt verwendet werden.

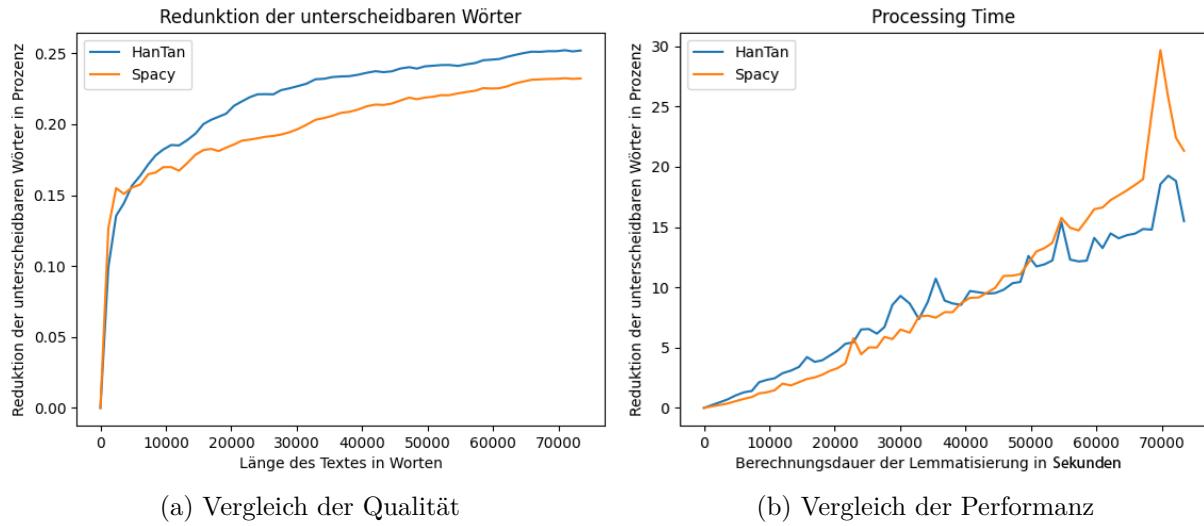


Abbildung 6.1: Vergleich unterschiedlicher Bibliotheken zur Lemmatisierung

6.3 Verifikation der Ähnlichkeitsanalyse

Die am Anfang des Kapitels getroffene Überlegungen legen nahe, dass Texte, welche sich mit dem gleichen Thema beschäftigen, eine ähnliche Auswahl von Wörtern verwenden. Genau dieses Phänomen soll hier in einer Stichprobe von vier Texten untersucht werden. Dazu soll der vorausgehend dargestellte Prozess verwendet werden.

Bei diesen handelt es sich um Artikel der Tagesschau, drei zum Thema Klimawandel und einer zum Thema Nahostkonflikt. Konkret sind es die Artikel unter den Titeln „Deutschland soll „Klimafest“ werden“ [21], „Klimawandel bleibt größte Gefahr“ [22] und „Wo der Klimawandel längst Realität ist“ [23] zum Thema Klimawandel und der Artikel „Die Gewalt nimmt nicht ab“ [24], welcher sich mit dem Nahostkonflikt beschäftigt.

6.3.1 Vorgehen und Auswahl der Stichprobe

Mithilfe der im ?? mathematischen Grundlage soll nun unter Verwendung der Kosinus-Ähnlichkeit die Ähnlichkeit der Texte zueinander in einer Matrix dargestellt werden. Um diese These zu beweisen, muss nun eine Methode entwickelt werden, um die Ähnlichkeit der Texte darzustellen. Dabei müssen sich die drei ersten Texte, welche alle das Thema Klimawandel behandeln, untereinander Ähnlich sein. Zudem müssen sie zum vierten Text mit dem Thema Nahostkonflikt unterscheiden.

Im nachfolgenden soll nun der Prozess dargestellt werden, mit welchem diese Unterscheidung realisiert wurde.

6.3.2 Auswertung der Ergebnisse

Die Tabellen in Abbildung 6.2 stellen das Ergebnis der in Abschnitt 6.1 und 6.2 Methodik zur Unterscheidung von Texten dar. Jede Zelle der Tabelle stellt dar, wie Ähnlich sich Text A, welcher aus der Spalte entnommen werden kann und Text B, welcher durch die Zeile angegeben ist, sind. Mit jeder Tabelle ein Prozessschritt eingeführt. So werden in Tabelle (b) die Stopwörter herausgefiltert, in Tabelle (c) wird zudem noch die Wortstandrückführung durchgeführt. Zur besseren Visualisierung wurden die Zellen abhängig vom Ähnlichkeitswert eingefärbt: Der höchste Wert wird grün gefärbt, werden der niedrigste Wert rot markiert wird. Die Farbwerte der restlichen Zellen werden über eine lineare Interpolation und additive Farbmischung bestimmt.

Die vier Tabellen zeigen die Sinnhaftigkeit jedes Schrittes eindeutig auf: Während in der Tabelle ganz oben das Ergebnis noch alles andere als das gewünschte ist, werden die Texte in Tabelle (d) eindeutig nach Thema unterscheidbar. So sieht man, dass in Tabelle (a) der Text „Wo der Klimawandel längst Realität ist“ dem Text „Nahostkonflikt: Die Gewalt nimmt nicht ab“ mit einem Wert von 0.023 ähnlicher ist als dem Text „Klimawandel bleibt größte Gefahr“ mit dem Wert 0.219. In Tabelle (d) werden die beiden Texte zum Thema Klimawandel deutlich von dem dritten Text unterschieden.

6.3.3 Auswahl einer Distanzfunktion

Die in Abbildung 6.2 gezeigten Werte wurden alle mit der Kosinus-Ähnlichkeit berechnet. Durch den nun beschriebenen Schritte wurde beschrieben, wie für einzelnen Texte Merkmalsvektoren so berechnet werden können, dass sie von anderen Texten unterscheidbar sind. Für die vier Texte wurden jeweils Merkmalsvektoren berechnet werden. Nun soll unter der Verwendung dieser eine Distanzmatrix erstellt werden, einmal mithilfe der Kosinus-Ähnlichkeit und einmal mithilfe der euklidischen Distanz. Das in Abbildung 6.3 abgebildete Ergebnis zeigt eindeutig, dass die Kosinus-Ähnlichkeit besser differenzierbare Werte liefert als die euklidische Distanz. Somit soll im weiteren die Kosinus-Ähnlichkeit verwendet werden. Nun muss der Prozess, der bei Tagesschauartikeln funktioniert auf die einzelnen Nutzer*innen angewendet werden. Ziel ist die Erstellung eines Merkmalsvektoren analog zu dem aus Formel 6.3.

	Deutschland soll Klimafest werden	Klimawandel bleibt größte Gefahr	Wo der Klimawandel längst Realität ist	Nahostkonflikt: Die Gewalt nimmt nicht ab
Deutschland soll Klimafest werden	1	0,20595599	0,25594026	0,18101746
Klimawandel bleibt größte Gefahr	0,20595599	1	0,21879749	0,16568225
Wo der Klimawandel längst Realität ist	0,25594026	0,21879749	1	0,23029958
Nahostkonflikt: Die Gewalt nimmt nicht ab	0,18101746	0,16568225	0,23029958	1

(a) Ohne Datenaufbereitung

	Deutschland soll Klimafest werden	Klimawandel bleibt größte Gefahr	Wo der Klimawandel längst Realität ist	Nahostkonflikt: Die Gewalt nimmt nicht ab
Deutschland soll Klimafest werden	1	0,08775165	0,09319647	0,0443767
Klimawandel bleibt größte Gefahr	0,08775165	1	0,06182071	0,02971977
Wo der Klimawandel längst Realität ist	0,09319647	0,06182071	1	0,0649313
Nahostkonflikt: Die Gewalt nimmt nicht ab	0,0443767	0,02971977	0,0649313	1

(b) Mit Herausfiltern der Stoppwörter

	Deutschland soll Klimafest werden	Klimawandel bleibt größte Gefahr	Wo der Klimawandel längst Realität ist	Nahostkonflikt: Die Gewalt nimmt nicht ab
Deutschland soll Klimafest werden	1	0,10795838	0,10816986	0,05563511
Klimawandel bleibt größte Gefahr	0,10795838	1	0,06784096	0,05725983
Wo der Klimawandel längst Realität ist	0,10816986	0,06784096	1	0,07769125
Nahostkonflikt: Die Gewalt nimmt nicht ab	0,05563511	0,05725983	0,07769125	1

(c) Mit Herausfiltern der Stoppwörter und Wortstammrückführung

	Deutschland soll Klimafest werden	Klimawandel bleibt größte Gefahr	Wo der Klimawandel längst Realität ist	Nahostkonflikt: Die Gewalt nimmt nicht ab
Deutschland soll Klimafest werden	1	0,14342743	0,1096817	0,02366905
Klimawandel bleibt größte Gefahr	0,14342743	1	0,09176629	0,01980295
Wo der Klimawandel längst Realität ist	0,1096817	0,09176629	1	0,02271554
Nahostkonflikt: Die Gewalt nimmt nicht ab	0,02366905	0,01980295	0,02271554	1

(d) Mit Herausfiltern der Stoppwörter, Wortstammrückführung und Minimalanzahl für Wort

Tabelle 6.2: Ähnlichkeitsmaße der einzelnen Artikel unter Verwendung der Kosinus-Ähnlichkeit

	Deutschland soll Klimafest werden	Klimawandel bleibt größte Gefahr	Wo der Klimawandel längst Realität ist	Nahostkonflikt: Die Gewalt nimmt nicht ab
Deutschland soll Klimafest werden	1	0,14342743	0,1096817	0,02366905
Klimawandel bleibt größte Gefahr	0,14342743	1	0,09176629	0,01980295
Wo der Klimawandel längst Realität ist	0,1096817	0,09176629	1	0,02271554
Nahostkonflikt: Die Gewalt nimmt nicht ab	0,02366905	0,01980295	0,02271554	1

(a) Distanzmatrix berechnet mit der Kosinus-Ähnlichkeit

	Deutschland soll Klimafest werden	Klimawandel bleibt größte Gefahr	Wo der Klimawandel längst Realität ist	Nahostkonflikt: Die Gewalt nimmt nicht ab
Deutschland soll Klimafest werden	0	0,32871967	0,36656921	0,2799177
Klimawandel bleibt größte Gefahr	0,32871967	0	0,29727163	0,21826404
Wo der Klimawandel längst Realität ist	0,36656921	0,29727163	0	0,26717189
Nahostkonflikt: Die Gewalt nimmt nicht ab	0,2799177	0,21826404	0,26717189	0

(b) Distanzmatrix berechnet mit der (normalisierten) euklidischen Distanz

Tabelle 6.3: Differenzierbarkeit von Werten berechnet mithilfe der Kosinus-Ähnlichkeit im Vergleich zur euklidischen Distanz

6.4 Erstellung der Merkmalsvektoren für Nutzer*innen

Will man die Merkmalsvektoren der Nutzer*innen bestimmen, muss man zunächst die Schlüsselwörter aller Ihrer Texte zusammenführen, also alle Tweets, die von ihnen verfasst wurde. Die Schlüsselwörter werden anhand dem in den vorherigen Kapiteln beschriebenen Prozess aus dem Text extrahiert und um einen weiteren Filter ergänzt: Es werden nur Wörter in Betracht gezogen, welche als Hashtag markiert wurden.

Pro Nutzer*in wird also für jeden der von ihm verfassten Tweets die Häufigkeitsverteilung der Schlüsselwörter berechnet und dann in eine große Häufigkeitsverteilung pro Nutzer*in zusammengefasst. Die beinhaltet nun, welche Schlüsselwörter der Nutzer*innen wie Häufig über den gesamten Datensatz hinweg verwendet hat. In einem weiteren Schritt wird nun berechnet, wie oft ein Schlüsselwort im gesamten Datensatz vorkommt. Um den Datensatz zu reduzieren und so auch Rauschpunkte zu entfernen, sollen nun Schlüsselwörter, welche nicht sehr häufig verwendet werden, herausgefiltert werden. Da die Tweets nach einen bestimmten Thema ausgesucht wurden, werden Schlüsselwörter die dieses beinhalten unverhältnismäßig häufig auftauchen, dadurch wird auch ein Schwellwert für eine obere Grenze benötigt. Als geignet erwies sich, nur Wörter zu verwenden, deren Häufigkeit zwischen dem 97% und dem 99.98% Quantil der Häufigkeiten aller Wörter lagen. Somit werden 98% aller möglichen Schlüsselwörter nicht betrachtet. Für einen Datensatz von

einem Monat entspricht dies 404,905 Wörtern.

Weiterhin sollen für den einzelnen Nutzer*innen auch nur die Wörter verwendet werden, welche er häufig verwendet. Hierfür wird zunächst eine Liste gebildet, welche für jeden Nutzer*innen und jedes Wort beinhaltet, wie häufig der Nutzer*innen dieses Wort verwendet hat. Als Schwellwert dient das 99% Quantil dieser Liste verwendet, welches bei den Tweets eines Monats bei sechs liegt. Nutzer*innen, welche mit keinem ihrer Schlüsselwörter diesen Schwellwert erreichen, werden ausgefiltert. Durch beide Filterschritte zusammen fallen 99.5% der Nutzer*innen aus dem Datensatz, was für einen Monat eine Zahl von 238.717 Nutzer*innen entspricht. Der gesamte Prozess ist in Abbildung 6.2 zusammengefasst.

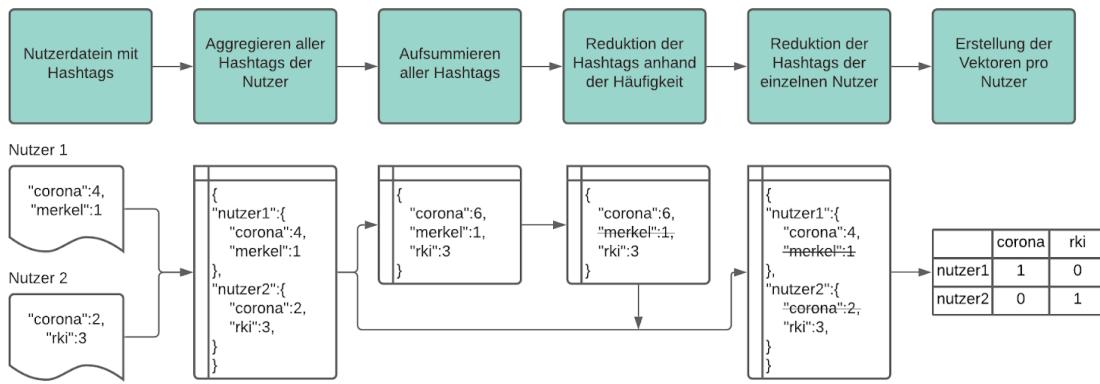


Abbildung 6.2: Bestimmung der Merkmalsvektoren der Nutzer*innen

6.5 Clustering anhand der Merkmalsvektoren

Da nun für jede*n Nutzer*in ein Merkmalsvektor bekannt ist, soll nun versucht werden, die Nutzer*innen anhand dieser in Kategorien einzuteilen. Der Prozess soll analog zum Clustering aus [19] erfolgen: Es soll zunächst durch mehrere Iterationen des k -Means Algorithmus eine Ähnlichkeitsmatrix gebildet werden, anschließend soll das Ergebnis mithilfe dieser und dem DB-SCAN Algorithmus zusammengefasst werden.

6.5.1 Berechnung einer Ähnlichkeitsmatrix mithilfe des k-Means Algorithmus

Der k-Means Algorithmus ist ein sehr bekannter Algorithmus, mit dem Ziel, n Merkmalsträger in k Cluster zu partitionieren, in denen jede Beobachtung zu dem Merkmalsträger

mit dem nächstgelegenen Mittelwert (Clusterzentren oder Clusterschwerpunkt) gehört, der als Prototyp des Clusters dient. Allerdings birgt er auch Probleme: Dadurch, dass die Clusterzentren am Anfang zufällig gewählt werden kann der Algorithmus bei gleicher Eingabe einen ungleiches Ergebnis. Abbildung 6.3 visualisiert dieses Verhalten. Zudem muss bei diesem Algorithmus die Anzahl an zu findenden Clustern vorgegeben werden. Dies kann insofern zu nicht optimalen Ergebnissen führen, da sich der Datensatz potentiell sinnvoller in eine Anzahl von Clustern unterteilen lässt, welche ungleich k ist. Für jeden

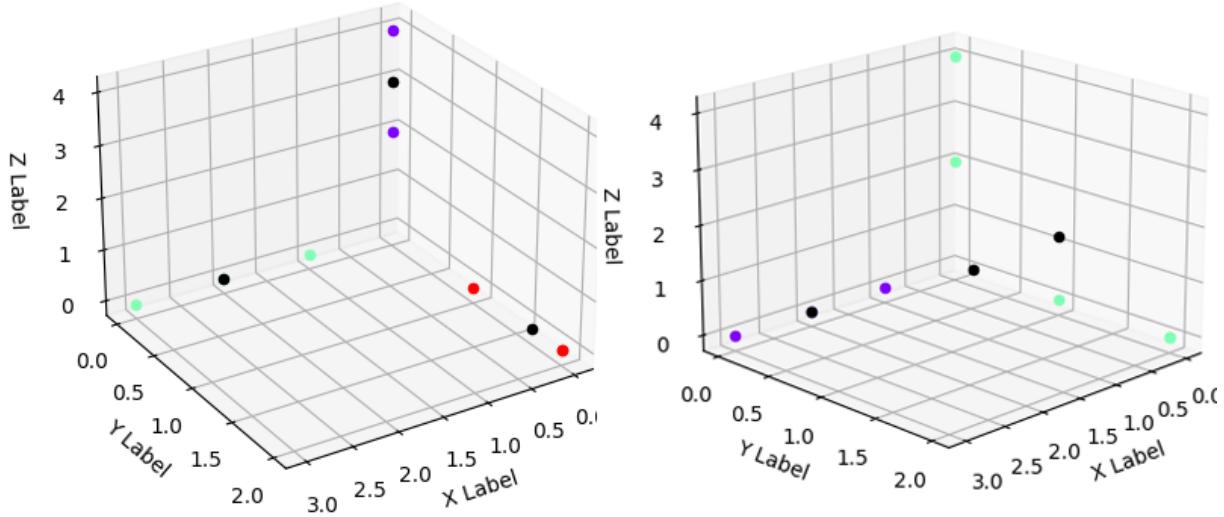


Abbildung 6.3: Gutes (links) und schlechtes (rechts) Ergebnis des k-Means Algorithmus unter gleichem Input

Lauf von k-means wird der Wert innerhalb der Matrix bei (Benutzer*in A, Benutzer*in B) und (Benutzer*in B, Benutzer*in A) um einen eins erhöht, wenn diese*r Benutzer*in im selben Cluster wie der/die andere*r landen. Daraus ergibt sich eine Matrix, anhand derer man für jeden Benutzer*in sagen kann, wie oft er mit einem anderen Benutzer*in in ein Cluster gefasst wurde und somit, wie ähnlich die Eigenschaften der die Benutzer*in sind. Dieser Prozess ist in Abbildung 6.4[19] noch einmal visualisiert.

6.5.2 Einteilung der Nutzer*innen mit dem DB-SCAN Algorithmus

Durch die Verwendung der Ähnlichkeitsmatrix und die Anwendung des DB-SCAN Algorithmus können nun die Benutzer*innen nun in verschiedene Cluster eingeteilt werden. Darüber hinaus bietet dieser Algorithmus die Möglichkeit Benutzer*innen, die sich nicht in einem ausreichend großen Agglomerationszentrum befinden, als Rauschpunkte zu markieren, anstatt sie einem Cluster zuzuordnen, vgl. Abbildung 6.5. In diesem Fall hängt Epsilon von der Anzahl der Iterationen des k-Means Algorithmus ab, da diese definiert, wie oft ein*e Benutzer*in maximal zusammen mit einem*r andere*n Nutzer*in bei einem

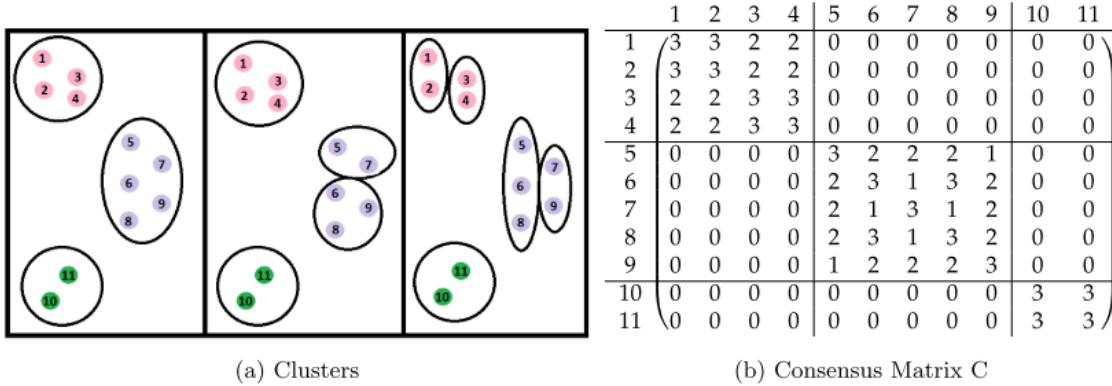


Abbildung 6.4: Berechnung einer Ähnlichkeitsmatrix durch mehrere Iterationen von k-Means

Durchlauf von k-Means dem selben Cluster zugeordnet werden können. Gute Ergebnisse lieferten ein Epsilon von 80 % dieses Wertes. Die Anzahl der minimalen Benutzer*innen zur Bildung eines Cluster ist definiert durch die Anzahl der Benutzer*innen im Datensatz mal 0,02. Diese Werte fallen für den Datensatz von einem Monat und dem Ausfiltern von Nutzer*innen nach 6.4 auf etwa 20, das Epsilon liegt für 15 Iterationen des k-Means Algorithmus entsprechend bei 12.

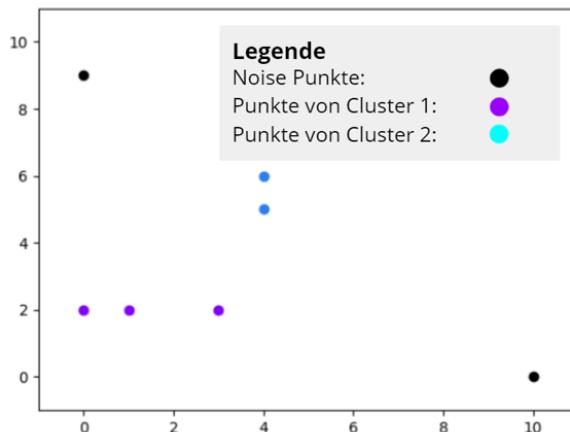


Abbildung 6.5: Einteilung von Punkten in Cluster und Rauschen durch DB-SCAN

6.6 Darstellung der Ergebnisse

Die Darstellung von hoch dimensionalen Daten ist insofern schwer zu realisieren, da eine Projizierung in einen mindestens dreidimensionalen Raum, besser in einen zweidimensionalen Raum stattfinden muss, welcher unweigerlich mit Informationsverlust verbunden ist. Hierzu wurden zwei Ansätze ausgewertet.

Ersterer ist unter dem Begriff **Multidimensional Skalierung (MDS)** bekannt: Dieser bezeichnet eine Familie von Verfahren, die hochdimensionale Daten anhand der Proximität zweier Datenpunkten auf einen zweidimensionalen Raum überführen soll, in dem für alle Punkte die Abstände im hochdimensionale Raum deren im zweidimensionalen Raum im Verhältnis entsprechen [25]. Als Distanzfunktion kann hier jede beliebige Funktion genutzt werden, welche etwas über die Proximität zweier Datenpunkte aussagt, häufig verwendet wird die euklidische Distanzfunktion nach Formel 6.1. Dieses Methode hat sich allerdings nach einigen Experimenten mit reduzierten Datensätzen als nicht zielführend erwiesen, da die Verteilung im Vergleich zur nächsten Methode sehr zufällig gewirkt hat und Schwellwerte zur Reduzierung des Datensatzes nicht implementierbar waren.

In einem nächsten Versuch wurde die Ähnlichkeitsmatrix aus den Iterationen des k-Means Algorithmus in einen Graph überführt. Hierbei entspricht jeder Knoten einem* Nutzer*in. Aus den Werten der Matrix ergab sich die Gewichtung der einzelnen Kanten zwischen den Nutzern: Wurde Nutzer*in A und Nutzer*in B laut Matrix insgesamt 12 mal in einem Cluster zusammengefasst, so wurde zwischen den beiden eine Kante mit dem Gewicht 12 eingetragen. Um die Kanten zu reduzieren wurde ein Schwellwert definiert. Wenn das Gewicht einer Kante unter diesem lag wurde sie entfernt. Der Graph wurde dann mithilfe der Software Gephi und dem Force Atlas Algorithmus gelayoutet. Pro Cluster des DB-SCAN Algorithmus wurde eine Farbe definiert, mit dem alle Knoten deren Nutzer*innen im entsprechenden Cluster zu finden waren eingefärbt wurden. Der Prozess der Überführung von Ähnlichkeitsmatrix zu Graph ist in Abbildung 6.6 visualisiert.

Durch die große Datenmenge und die große Anzahl an Kanten musste zum besseren Layou-

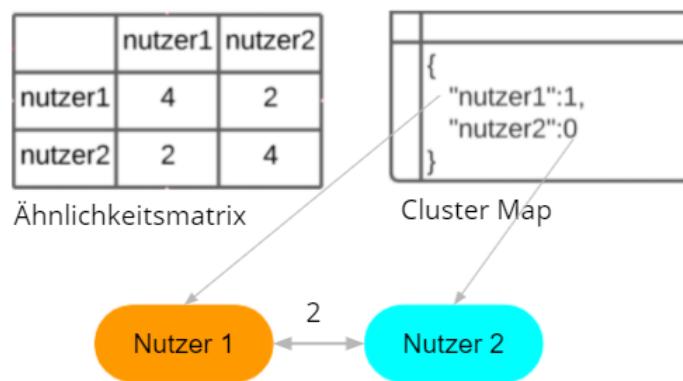


Abbildung 6.6: Überführung der Ähnlichkeitsmatrix zu einem Graphen

ten des Graphen weitere Optimierungen vorgenommen werden. Bei genaueren Betrachtung der Ähnlichkeitsmatrix fiel auf, dass einige Zeilen der Matrix identisch waren, also einige Nutzer*in identisch häufig mit jedem anderen Nutzer*in in einem Cluster zusammengefasst

wurden. Dieser Effekt ist auch in Abbildung 6.4 für die Datenpunkte 1 und 2 zu sehen, bei welchen sich die Zeilen entsprechen. Da diese in der Überführung zu einem Graphen zu identischen Knoten führt, können sie auch in einem Knoten zusammengefasst werden. Um graphisch darzustellen, dass ein Knoten mehr als eine Person repräsentiert, soll in der Darstellung der Flächeninhalt eines Knotens proportional zur Anzahl der durch in repräsentierten Personen sein. Der Radius für einen Knoten lässt sich somit aus Formel 6.5 ableiten. Das Ergebnis des nun beschriebenen Prozesses ist als Abbildung 6.7. Hier wurden die charakteristischen Schlüsselwörter einiger Cluster schon nach Kapitel 5.9 als Wortwolken ergänzt.

$$r = \sqrt{n \cdot \pi}, n = \text{Anzahl der Personen} \quad (6.5)$$

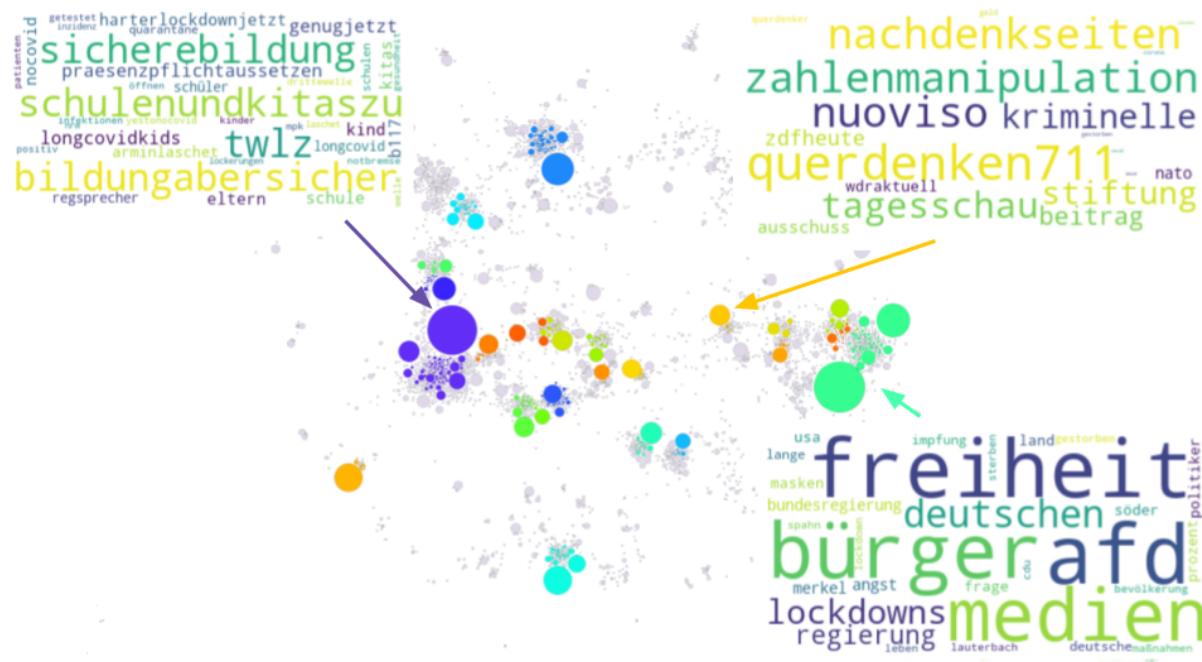


Abbildung 6.7: Visualisierung der Gruppierung von Nutzer*innen anhand der von ihnen verwendeten Wörter

7 Zusammenführung und Vergleich

Um die mit beiden Ansätzen gefundenen Cluster weiter zu untermauern, wurden die zu jedem Cluster gehörenden Benutzer*innen verglichen, um zu untersuchen, ob es möglich ist, Cluster aus dem Retweet-Netzwerk-Ansatz Cluster aus dem Sprach-Clustering-Ansatz zuzuordnen. Um die Beziehung zu veranschaulichen, wurde ein Sankey-Diagramm erstellt, das links die Netzwerkcluster, rechts die Sprachcluster und Benutzer, die Teil von zwei Clustern sind, als grauen Fluss darstellt. Benutzer aus den Netzwerkclustern, die keinem Sprachcluster angehören, fließen nach undefined"(siehe Abbildung 7.1). Mehr als 50% der

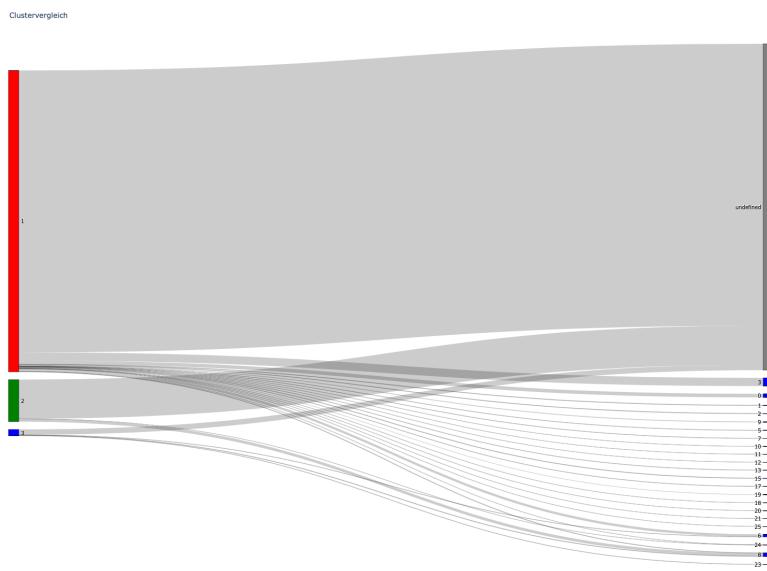


Abbildung 7.1: Benutzerassoziationen zwischen Clustern der beiden Ansätze

Benutzer aus den Netzwerkclustern wurden nicht im Sprachansatz geclustert (auch vice versa). Dies kann durch die Schritte in beiden Ansätzen erklärt werden die das Filtern von Benutzern beinhalten: Der Rohdatensatz enthält 260.954 Benutzer.

7.1 Filterschritte im Netzwerkansatz

- Nur Benutzer*innen werden berücksichtigt, die retweeten: 37% (966.322) der Benutzer*innen werden herausgefiltert

- Nur Benutzern*innen werden berücksichtigt, die Influencer retweeten: 57% der Benutzer, die retweeten (94.730), werden herausgefiltert
- Anwenden eines Schwellenwerts auf Superuser-Verbindungen:
Da einige Superuser gelöscht werden, wenn alle ihre Verbindungen unter dem Schwellenwert liegen, werden ihre zugehörigen Benutzer*innen aus dem Cluster gefiltert. 96% der Nutzer, die Influencer retweeten (67.119), werden herausgefiltert

7.2 Filterschritte im Sprachansatz

- Es werden nur die Hashtags verwendet, deren Anzahl zwischen dem 97% und 99,98% Quantil der Häufigkeitsverteilung liegt: 98% (404.905) der Hashtags werden herausgefiltert
- Nur die Benutzer, die diese Hashtags mehr als 6 Mal innerhalb des Monats verwendet haben, werden beibehalten: 99,5% (238.717) aller Benutzer werden herausgefiltert

Da beide Ansätze den Datensatz nach unterschiedlichen Merkmalen filtern, ist die durchschnittliche Anzahl der Nutzer, die in den Ergebnissen für beide Ansätze vorhanden sind, sehr gering Ein identisches Filterverfahren für beide Ansätze würde dieses Problem lösen.

8 Auswertung

Rückblickend auf die vier Fragestellungen aus [1.1](#) soll nun diskutiert werden, inwieweit die Ergebnisse der Arbeit die Fragen beantworten. Lassen sich Gruppen von Nutzer*innen finden, die 1) die gleiche Meinung zur Covid-19 Pandemie haben, 2) diese Meinung mit dem selben Sentiment kundtun, 3) diese Meinung mit der gleichen Sprache kommunizieren und 4) Informationen nur aus ihrer assoziierten Gruppe erhalten ?

Frage 2) konnte aufgrund von Budget- und Qualitätsbeschränkungen in dieser Arbeit nicht beantwortet werden.

Für die Untersuchung soll im Folgenden eine Interpretation der Ergebnisse beider durchgeführten Clusteringmethode vorgestellt werden. Aus dem in Abschnitt [7](#) vorgestellte Vergleich kann auf Grund der unterschiedlichen Teilmengen der Nutzer*innen keine Interpretationen gezogen werden. Vor Allem im Retweetclustering sollte hier eine Möglichkeit gefunden werden, mehr Nutzer*innen in die Cluster miteinzubeziehen.

8.1 Ergebnisinterpretation des Retweetclustering

Mithilfe dieser Methode konnten 3 Gruppierung gefunden werden, die Informationen (Retweets) nur innerhalb dieser Gruppe teilen und liefert damit eine Teilantwort auf Frage 4). Da es sich allerdings nur um die Retweets eines/r Nutzer*in handelt und keine Aussage gemacht wird, ob diese*r Nutzer*in eventuell anderen Influencern folgt (und damit auch dessen Informationen erhält) ist der eindeutige Schluss auf eine „Echokammer“ nicht gegeben. Außerdem, ist in weiterführenden Arbeiten zu untersuchen, inwieweit ein Retweet tatsächlich eine Meinungszuordnung zur Folge haben kann. Beispielsweise könnte ein*e Nutzer*in einen Influencer mit zu ihm gegensätzlicher Meinung retweeten, nur um diesen Gegensatz aufzuzeigen. Eine Einteilung in das gleiche Meinungscluster wäre hier falsch. Ein solcher Fall wird bei einem/r Nutzer*in allerdings nur vereinzelt auftreten; man könnte diesen mit einem Threshold auf Nutzerebene aussortiert. Da ein Threshold allerdings nur auf Supernetzerebene angewandt wird und hier die Retweets aller Nutzer*innen schon aufsummiert wurden, sind solche Nutzer*innen in den Cluster trotzdem enthalten. Allerdings kann vor Allem die Gruppierung zweier Influencern in ein gleicher Cluster als Meinungsgleichheit angesehen werden. Frage 1) konnte aus dieser Perspektive also mithilfe

des Retweetansatzes positiv beantwortet werden. Des Weiteren ist zu beachten, dass wie in Abschnitt 7 dargestellt, nur ein Bruchteil aller Nutzer*innen der Datenbasis in den Clustern vorhanden sind.

8.2 Ergebnisinterpretation des Keywordclustering

Diese Methode lieferte 23 Gruppierung von Nutzer*innen die in der Gesamtheit ihrer Tweets dieselbe Schlagwörter verwenden und damit die selbe „Sprache“ sprechen. Das heißt, Frage 2) konnte positiv beantwortet werden. Auch gilt es ein ähnliches Problem zu diskutieren wie im Retweetclustering. Für eine/n Nutzer*in der/die z.B. ein Schlagwort einer im gegensätzlichen Meinung verwendet um diesen Kontrast aufzuzeigen wäre eine entsprechende Meinungszuordnung falsch. Im Gegensatz zu 8.1 werden diese vereinzelten Fälle allerdings schon auf Nutzerebene aussortiert, weshalb hier Frage 1) unter der Prämisse, dass von den verwendeten Schlagwörtern auf eine Meinung geschlossen werden kann, positiv beantwortet werden kann. Natürlich gilt es auch hier diese Prämisse wissenschaftlich zu untermauern.

8.3 Anwendungsfälle

Während dem Präsidentschaftswahlkampf in den USA 2016 waren zu diesem Thema 25% aller auf Twitter verbreiteten Informationen Fakenews [26]. Als Problem identifiziert, versuchen soziale Platformen inklusive Twitter solche zu erkennen und zu unterbinden. Die in dieser Arbeit vorgestellten Clusteringmethoden könnten bei dieser Unterbindungen unterstützen. Ist ein*e Nutzer*in dafür bekannt, Fakenews zu verbreiten, so können alle Nutzer*innen die mit ihm/ihr im selben Cluster sind ebenso gezielt auf das Verbreiten von Fakenews untersucht werden.

9 Rückblick

Twitter ist eine Plattform, auf der Millionen von Menschen jeden Tag öffentlich ihre Gefühle und Meinungen teilen. Die Analyse selbst großer Mengen dieser Informationen ist durch die Zunahme der Computerleistung möglich geworden, aber auch durch die Hilfe vieler engagierter Forscher, die immer bessere Wege finden, diese ansonsten chaotischen Daten zu strukturieren.

In diesem Beitrag wurden zwei Ansätze für das Problem des Auffindens von Clustern in diesem unstrukturierten Datensatz vorgestellt. Wir konnten zeigen, dass es Cluster auf Twitter gibt, die nur sich selbst retweeten und dieselbe Sprache bezüglich der Debatte über die Covid-19-Pandemie verwenden.

Mit Hilfe der Sentimentanalyse ist es uns aus den in Kapiteln 4 genannten Gründen schwer gefallen aussagekräftige Ergebnisse zu errechnen. Eine sinnvolle Einteilung in Gruppen gelang nicht. Es ist jedoch noch viel Raum für Verbesserung, auf den wir in folgenden Kapitel noch weiter eingehen wollen.

10 Ausblick

Trotz der sehr aussagekräftigen Ergebnisse gibt es viel Raum für Optimierung. Um diese Arbeit auf noch festere Füße stellen zu können, wäre eine statistische Verifikation der Hypothesen, welche in Kapitel 2.1 vorgestellt wurden. Anhand eines händisch erstellten Datensatzes könnte so herausgefunden werden, in wie viel Prozent ein Retweet tatsächlich einer Zustimmung entspricht und in wie viel Prozent das Gegenteil der Fall ist. Diese Erkenntnisse könnten so unsre Berechnungen mehr an die Realität annähern.

Ein wichtiger Punkt ist die Verbesserung der Sentimentanalyse. Die Stimmung des Verfassers aus einem Text abzuleiten ist um ein einiges komplexer als das Vorkommen einer Wörter zu überprüfen, weshalb deutlich komplexere und adaptiver Algorithmik benötigt wird um Sinnvolle Ergebnisse zu erzielen. Die jüngsten Vorschritte im Bereich der Verarbeitung natürlicher Sprache zeigen aber, dass dies mit aktuellem Stand der Technik durchaus realisierbar sein sollte. Durch die Verwendung von modernen Modellen der Sprachverarbeitung wie zum Beispiel der Watson KI [27], welche auch über eine Stimmungsanalysefunktion verfügt könnten Stimmungen deutlich differenzierter und realitätsnäher herausfinden. Allerdings ist die Verwendung dieser kostenpflichtig, eine Analyse des hier verwendeten Datensatzes wäre im Kontext dieser Studienarbeit zu teuer. Ein weiteres Problem ist, dass viele der kostenlosen Lösungen zwar im Englischen sehr gute Ergebnisse liefern, für die Deutsch aber nur schlecht anwendbar sind. Eine Möglichkeit für die Verbesserungen der Methodik zur Sentimentanalyse allgemein wäre, die Berechnung der Stimmung des Nutzers durch den die durchschnittliche Stimmung seiner Tweets um eine detailliertere Methode zu ersetzen: Sinnvoller könnte es sein, die Stimmung eines Tweets in Verbindung mit in ihm verwendeten Schlüsselwörter zu verbinden um so herauszufinden, in welchem Verhältnis der Nutzer zu einem bestimmten Wort steht, statt nur auf eine allgemeines Verhältnis zum Thema. Letztendlich finden wir alle Corona nervig, der eigentliche Erkenntnisgewinn könnte sich eine Ebene darunter verbergen. Diesem könnte in weiteren Untersuchungen nachgegangen werden.

Auch für bei den anderen Vorgehensweisen gibt es Raum für Verbesserungen: Das Auffinden von Clustern im Retweet-Netzwerk hängt stark von der Anzahl der Influencer und dem gewählten Schwellenwert ab. In zukünftigen Arbeiten eine Methode zur Wahl dieser Parameter, um die Komplexität nur so weit wie nötig zu reduzieren und dabei möglichst wie nötig zu reduzieren und dabei möglichst viele Nutzer im Datensatz zu behalten, wird einen Schritt nach vorne bedeuten. Die Ausweitung der Sprachcluster-Methode auf

Schlüsselwörter und Feinabstimmung der Parameter für k-means und DB-SCAN kann mehr eindeutige Cluster ergeben. Ein wichtiger Faktor ist die Anzahl der Iterationen des k-Means-Algorithmus. Die fünfzehn Iterationen in diesem Fall dauerte die Berechnung etwa 23 Stunden, aber es gibt Möglichkeiten für Optimierungen. Mehr Iterationen mit Werten für $k \geq 20$ oder mehr würden die Ergebnisse noch detaillierter machen. Eine Vergrößerung des gesamten Datensatzes auf mehr als einen Monat würde bessere und deutlichere Ergebnisse liefern. Außerdem, da beide Methoden einfache Ordner- und Dateioperationen verwenden, könnte eine Architektur aufgebaut werden, um die Rechenzeit zu verringern. Viele Aufgaben sind auch theoretisch parallelisierbar, in die Praxis umzusetzen, was die Rechenzeit ebenfalls verringern würde.

Ein wichtiger Schritt wäre die Generalisierung des nun dargestellten Prozesses: Satt einem Datensatz aus Tweets zum Thema Corona zu verwenden könnten diese auch einem ganz anderen Themengebiet entstammen. So könnte man zum Beispiel einen Datensatz zum Thema Bundestagswahl dazu verwenden, Wähler einzelner Parteien zu identifizieren. Auch eine Marketingabteilung könnte die Wahrnehmung der eigenen Marke in der Twittercommunity besser nachvollziehen. Theoretisch sollte der Prozess auf jedes Thema anwendbar sein, jedoch wurde dies noch nicht in der Praxis überprüft.

Wir hoffen, in dieser Arbeit einen Beitrag zur Analyse der Struktur des Twitter-Ökosystems erarbeitet zu haben und hoffen, dass weitere Forschung aufbauend auf unserer Arbeit durchgeführt werden kann. Wie nun dargestellt, ist weder das Thema, noch der Datensatz und auch unsere Ideen noch lange nicht ausgeschöpft. Wir hoffen, uns auch in der Zukunft noch weiter mit dem Thema beschäftigen zu können und bedanken uns herzlichst bei der Dualen Hochschule Baden-Württemberg und unserem Betreuer für die Unterstützung.

Literaturverzeichnis

- [1] T. Inc. (2021) Q1 2021 letter to shareholders.
- [2] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu, *Predicting Flu Trends using Twitter data*, 2011.
- [3] V. Friedemann, “Clustering a customer base using twitter data,” 2015.
- [4] G. Ifrim, B. Shi, and I. Brigadir, “Event detection in twitter using aggressive filtering and hierarchical tweet clustering,” *CEUR Workshop Proceedings*, vol. 1150, 01 2014.
- [5] S. Miyamoto, S. Suzuki, and S. Takumi, “Clustering in tweets using a fuzzy neighborhood model,” 06 2012.
- [6] C. et al., “Predicting the political alignment of twitter users,” 10 2011.
- [7] V. Kharde and S. Sonawane, “Sentiment analysis of twitter data: A survey of techniques,” *International Journal of Computer Applications*, vol. 139, pp. 5–15, 04 2016.
- [8] G. et.al., “A case study in text mining: Interpreting twitter data from world cup tweets,” 2014. [Online]. Available: <http://arxiv.org/pdf/1408.5427v1>
- [9] H. Eberhardt. (2020) Twitter 2020: ein jahresrückblick aus trends und hashtags. [Online]. Available: <https://www.absatzwirtschaft.de/twitter-2020-ein-jahresueckblick-mit-trends-und-hashtags-176871/>
- [10] J. E. u. J. G. Johannes Baldauf. Hassrede und radikalisierung im netz. [Online]. Available: <http://www.isdglobal.org/wp-content/uploads/2018/09/ISD-NetzDG-Report-German-FINAL-26.9.18.pdf>
- [11] T. Inc. Twitter allgemeine geschäftsbedingungen. [Online]. Available: <https://twitter.com/de/tos>
- [12] M. K. Siddhartha Chatterjee, *Python Social Media Analysis*. Packt Publishing Ltd., vol. 1.
- [13] B. Stifter. Tweet. [Online]. Available: <https://twitter.com/Babsy1963/status/1364112098964471817>
- [14] S. C. North, “Drawing graphs with neato.” [Online]. Available: <http://www.graphviz.org/pdf/neatoguide.pdf>

- [15] H.-P. K. Martin Ester, “A density-based algorithm for discovering clusters.”
- [16] A. Graen. Anabel schunke ist eine der wichtigsten figuren der neu-rechten szene: Wir waren mit ihr feiern. [Online]. Available: https://www.focus.de/panorama/welt/panorama-anabel-schunke-ist-eine-der-wichtigsten-figuren-der-neurechten-szene-wir-waren-mit-ihr-feiern_id_10281656.html
- [17] laszlohealth. (2021) Tweet. [Online]. Available: <https://twitter.com/laszlohealth/status/1319338449149874181>
- [18] Medinside. (2021) Verhafteter aargauer arzt in der psychiatrie. [Online]. Available: <https://www.medinside.ch/de/post/verhafteter-aargauer-arzt-in-der-psychiatrie>
- [19] D. Godfrey, C. Johns, C. Meyer, S. Race, and C. Sadek, “A case study in text mining: Interpreting twitter data from world cup tweets.” [Online]. Available: <http://arxiv.org/pdf/1408.5427v1>
- [20] N. Reinert. (2020, 12) Einführung in stemming und lemmatisierung deutscher texte mit python. [Online]. Available: <https://nickyreinert.de/blog/2020/12/09/einfuehrung-in-stemming-und-lemmatisierung-deutscher-texte-mit-python/>
- [21] M. Heberlein, “Deutschland soll "klimafest" werden.” [Online]. Available: <https://www.tagesschau.de/inland/klimawandel-massnahmen-bundesregierung-101.html>
- [22] D. K. Mäurer, “Klimawandel bleibt größte gefahr.” [Online]. Available: <https://www.tagesschau.de/wirtschaft/weltwirtschaft/weltrisikobericht-weltwirtschaftsforum-klimawandel-klimaschutz-101.html>
- [23] K. Bensch, “Wo der klimawandel längst realität ist.” [Online]. Available: <https://www.tagesschau.de/ausland/ostafrika-klimawandel-101.html>
- [24] I. Spinner, “Die gewalt nimmt nicht ab.” [Online]. Available: <https://www.tagesschau.de/ausland/asien/israel-angriffe-113.html>
- [25] I. Borg, “Multidimensionale skalierung,” in *Handbuch der sozialwissenschaftlichen Datenanalyse*, C. Wolf and H. Best, Eds. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, pp. 391–418.
- [26] A. Bovet and H. A. Makse, “Influence of fake news in twitter during the 2016 us presidential election,” *Nature Communications*, 2019.
- [27] IBM. Ibm tone analyser. [Online]. Available: <https://www.ibm.com/de-de/cloud/watson-tone-analyzer>

Glossar

Amazon Kinesis Data Firehose

Ein Service von AWS, der Streaming Daten aufnehmen, transformieren und in Datenspeicher ablegen kann.

Amazon Web Services

Cloud-Infrastruktur die Server, Speicher uvm. einfach und kostengünstig bereitstellt.

Application Programming Interface

Von Softwaresystemen zur Verfügung gestellte Schnittstelle für andere Programme um zu interagieren.

Cluster

Ein Ganzes zu betrachtende Menge von Einzelementen.

EC2 - Instanz

Ein virtueller Server, der bei AWS gemietet werden kann.

Fakenews

in den Medien und im Internet, besonders in sozialen Netzwerken, in manipulativer Absicht verbreitete Falschmeldungen.

geretweetet

Ein Tweet dieses/dieser Nutzer*in wurde von einem /einer anderen Nutzer*in veröffentlicht.

Hashtag

Ein Schlagwort angeführt von einem Doppelkreuz(#), das dazu dient Texte mit einem bestimmten Thema zu versehen.

JavaScript Object Notation

Ein Datenformat aufgebaut in hierarischer Form das Zeichenketten, Zahlen, Listen und weitere Objekte erlaubt.

Layoutalgorithmus

Ein Algorithmus, der zu einem gegebenen Graph G absolute Positionen in einem Koordinatensystem berechnet, sodass der Graph möglichst übersichtlich dargestellt wird.

Matplotlib

Eine Programmbibliothek für Python mit der mathematische Darstellungen erstellt werden können.

Natural Language Processing

Die Methodiken mit denen, Computer Texte analysieren.

OpenSource

Software deren Quellcode öffentlich zugänglich ist, meistens auch kostenlos.

Python3

Eine universelle, interpretierte, höhere Programmiersprache.

Retweet

Ein veröffentlichter Tweet kann von anderen Nutzer auf deren Seite veröffentlicht werden. Dieser neue Tweet, der dem Original gleich ist, wird Retweet genannt.

retweeted

Die Aktion einen Retweet eines/einer andere*n Nutzer*in zu veröffentlichen.

S3 - Bucket

Ein Filehosting Service auf dem über HTTPS Daten in Ordnerstrukturen abgelegt und heruntergeladen werden können.

Text Mining

Ein Feld der Informatik das sich mit der Aufgabe beschäftigt, mit Computern Informationen und Wissen aus Texten zu erhalten.

Anhang

