Datenexploration: Wie hat sich das deutschsprachige Twitter von 2021 bis 2023 verändert?

Luca Hammer, Martina Schories

2023-06-02

Inhaltsverzeichnis

Abbildungsverzeichnis

Vorwort

Twitter war Vorreiter beim Thema Datenzugang und Datentransparenz. Der kostenlose Zugang zu umfangreichen Daten über die API (Programmierschnittstelle) für alle, ermöglichte es Dritten, Twitter auf unterschiedliche Art und Weisen zu untersuchen. Twitter führte für Wissenschaftler*innen einen akademischen API-Zugang ein, der Zugriff auf das gesamte Twitter-Archiv gewährte. Wenn Twitter weitreichende Manipulationsversuche identifizierte, stellte das Unternehmen Datensätze über die gelöschten Tweets zur Verfügung. Trotz zahlreicher Kritikpunkte, wie etwa die Intransparenz darüber, wer akademischen Zugang bekam und wer nicht oder nach welche Kriterien Stichproben-APIs Daten ausgaben, war Twitter im Vergleich zu anderen großen Social-Media-Plattformen transparenter und einfacher zu untersuchen.

Auch diese Datenexploration war nur möglich, weil wir noch Zugang zur API verfügten. Nach der Übernahme durch Elon Musk wurde die Anmeldung zum akademischen Zugang entfernt und der kostenlose Standardzugang auf das Veröffentlichen von 1500 Tweets pro Monat limitiert. Das Abfragen von Tweets ist mit neuen Zugängen nicht mehr kostenlos möglich und die bezahlten Zugänge sind mit rechnerisch 0,5 bis 1 USD Cent pro Tweet für die meisten Auswertungen zu teuer. Obwohl wir Vorgehen und Programmiercode veröffentlichen, wird es für Dritte nicht möglich sein eine solche Auswertung für einen anderen Zeit- und/oder Sprachraum durchzuführen. Das reduziert die Transparenz und verhindert datenbasierte Einblicke in öffentliche Debatten und die Funktionsweise von Twitter.

> **Transparenzhinweis**
>
> Diese Untersuchung wurde vom ZDF Magazin Royale in Auftrag gegeben und wurde erstmals unter www.vogel.rip veröffentlicht.

1. Einleitung

Elon Musk hat im April 2022 angekündigt, dass er Twitter kaufen möchte. Manche haben dies gefeiert, Musk würde endlich Meinungsfreiheit auf Twitter bringen, bei anderen machte sich ein Unwohlsein breit, wie es mit der Plattform weitergehen würde, die bei zahlreichen sozialen Bewegungen eine Rolle gespielt hat. Weil Musk versuchte vom Kauf zurückzutreten, hat es ein halbes Jahr gedauert bis er rechtlich Besitzer von Twitter wurde. Seitdem ist ein weiteres halbes Jahr vergangen und er hat zahlreiche Änderungen vorgenommen. Von der Entlassung des Großteils der Mitarbeiter*innen über die Veränderung der Algorithmen zugunsten von zahlenden Nutzer*innen bis zur Umdeutung was eine Verifizierung bedeutet.

In dieser Datenexploration versuchen wir herauszufinden, wie sich die Übernahme von Twitter durch Elon Musk auf das Nutzungsverhalten ausgewirkt hat. Vor allem auf einer strukturellen Ebene. Dazu haben wir einen fast vollständigen Datensatz aller deutschsprachigen Tweets der letzten zwei Jahre ausgewertet. Wie hat sich das Tweetvolumen insgesamt verändert? Welche Communities twittern mehr und welche weniger? Hat sich die Vielfalt auf Twitter verändert? Nicht alle Fragen lassen sich eindeutig beantworten. Vor allem nicht mit Daten alleine. Wir haben in den Daten Trends gefunden, die darauf hinweisen, dass weniger deutschsprachige Tweets veröffentlicht werden und eine Community aus vorwiegend rechten Accounts die deutschsprachige Twittersphäre immer stärker dominiert.

2. Über den Datensatz

Der Untersuchung liegen drei Datensätze zu Grunde:

- 1.276.345.791 deutschsprachige Tweets (Dezember 2020 bis Mai 2023)
- Accountinfos von 4.203.007 Twitteraccounts
- Folgeverbindungen von 625.374 Twitteraccounts

2.1. Datenerhebung

Alle Daten wurden über die Twitter Standard API erhoben.

2.1.1. Tweets

Die Erhebung der Tweets fand täglich statt. Jeweils um 03:34 Uhr wurde automatisiert ein Skript gestartet, das von der Twitter-API alle öffentlichen deutschsprachigen Tweets mit einer höheren ID als der des ersten gesammelten Tweets des Vortages abgefragt hat. Von den Tweets wurden die IDs (ID des Tweets, ID des twitternden Accounts und falls vorhanden ID des Tweets, auf den geantwortet, der geretweetet oder der zitiert wurde) sowie das Erstelldatum (created_at) in einer jsonl-Datei abgespeichert. Die Abfrage dauert pro Tag etwa 12 Stunden. Manchmal kam es durch Fehler am Server oder bei Twitter zu Abbrüchen, wodurch wenige Tage nicht vollständig erhoben wurden. In Einzelfällen wurde eine Nacherhebung gestartet.

```
import pandas as pd
import seaborn
seaborn.set()
figsize=(10,5)

ax = pd.read_csv(
    'data/tweets_pro_tag.csv',
    index_col='created_at',
    parse_dates=['created_at']).plot(
        figsize=figsize,
        title='Tweets im Datensatz', legend=False
        )
```

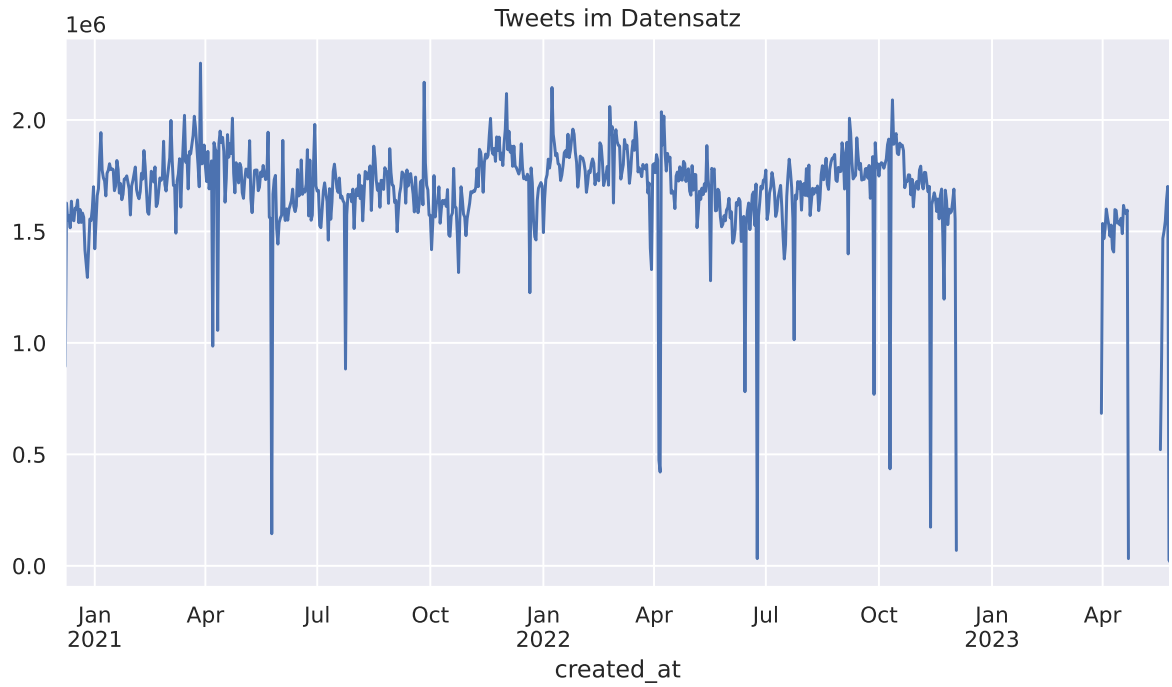


Abbildung 2.1.: Tweets im Datensatz

2.1.2. Accountinfos

Am 10.04.2023 wurde für alle Accounts, die mit mindestens 10 Tweets im Tweetdatensatz vertreten waren, die öffentlichen Accountinformationen abgefragt.

2.1.3. Follows

Für Accounts, die mit mehr als 100 Tweets im Tweetdatensatz vertreten waren, wurden zusätzlich die Follow-Informationen abgefragt, falls diese öffentlich waren.

2.2. Datenfehler und Einschränkungen

Wie im Abschnitt zur Erhebung der Tweets erwähnt, kam es bei der Abfrage von Tweets an einzelnen Tagen zu Abbrüchen. An 757 Tagen wurde die Abfrage gestartet, an 725 wurde die Sammlung abgeschlossen. An 32 Tagen ist die Sammlung unvollständig geblieben. An weiteren 142 Tagen konnte die Sammlung nicht gestartet werden. Der Großteil der kompletten Fehltage fällt in die ersten Monate von 2023, weil das Skript wegen eines Fehlers an einem Tag für mehrere Monate ausgefallen ist.

Neben den fehlenden Tagen gibt X Inc. (ehemals Twitter Inc.) keine Garantie für die Vollständigkeit der Daten, die über die Standard API abgefragt werden. Ein Vergleich mit der kostenpflichtigen Premium-API hat gezeigt (Hammer (2020)), dass die Standard-API knapp über 90% der über die von damals Twitter als vollständig vermarkteten Premium-API ausgibt.

Eine weitere Einschränkung sind Tweets, die zum Zeitpunkt der Sammlung bereits gelöscht wurden. Pfeffer u. a. (2023) haben 2022 gezeigt, dass der größte Rückgang (-6,1%) in den ersten 24 Stunden nach Veröffentlichung stattfindet. Innerhalb der ersten 10 Tagen verschwinden insgesamt 10,8% der Tweets.

Tweets von privaten Accounts wurden aus technischen und ethischen Gründen nicht erfasst.

Auch die Sprachklassifizierung durch Twitter führt zu Problemen mit False-Positives und False-Negatives. Das bedeutet, dass Tweets gesammelt wurden, die nicht deutschsprachig sind sowie Tweets nicht gesammelt wurden, die deutschsprachig sind. 2016 wurde die Sprachklassifizierung im Vergleich zu den Jahren zuvor verbessert (Alshaabi u. a. (2020)). Dennoch kommt es immer wieder zu Fehlern. Die False-Positives kommen vor allem durch das hohe Volumen an Tweets in anderen Sprachen zustande. Wenn in diesen deutschsprachige Begriffe (etwa Namen wie “Klein” oder “Bündchen”) vorkommen, kann es zu einer Falschklassifizierung kommen. Da im Datensatz keine Inhalte sind, kann keine eigene Klassifizierung zur Überprüfung stattfinden. Stattdessen filtern wir über die Anzahl der Tweets je Accounts im Datensatz sowie über Follow- und Interaktionsverbindungen, um die Menge der False-Positives zu minimieren. Über Stichproben können wir Aussagen über die Zuverlässigkeit der Filterung treffen. Mehr dazu in Kapitel 5.3.

Bei dem Followwetzwerk kommt es aufgrund der Grundgesamtheit zu Verzerrungen. Accounts, die in den Randbereichen des Untersuchungszeitraums aktiv waren, mussten hochfrequenter twittern, um über das Mindestvolumen von 100 Tweets in die Stichprobe aufgenommen zu werden, als Accounts, die über den gesamten Zeitraum twitterten. Da die Communities aus den Netzwerken zur Auswertung des Tweetvolumens herangezogen wurden, wird die Verzerrung dort weitergetragen. Über den Anteil der Tweets in identifizierten Communities, können wir feststellen, wie groß die Verzerrung ist. Für die Aprilstichprobe von 2021 konnten 73% der Tweets einer Community zugeordnet werden, 2022 waren es 76% und 2023 70%.

Beim Interaktionsnetzwerk tritt der Effekt lediglich für 2022 auf, weil die Interaktionen der drei Stichproben kombiniert wurden. Für das Interaktionsnetzwerk mussten Accounts mindestens zwei Interaktionen mit Tweets im Datensatz haben. Wenn Accounts nur für ein Jahr aktiv ist, kommt er 2022 eher auf die benötigten Interaktionen, weil sie sich entweder mit 2021 oder 2023 überlappen können, während sie 2021 und 2023 jeweils nur mit 2022 überlappen würden. Auch hier kann die Auswirkung über den Anteil an Tweets in identifizierten Communities verglichen werden. Für April 2021 konnten 49% der Tweets einer Interaktionskommunity zugeordnet werden. 2022 45% und 2023 42%. Da für das Interaktionsnetzwerk Interaktionen in Tweets ausgewertet werden, aber nicht alle Tweets eine Interaktion enthalten, sorgt der Rückgang an

Tweets insgesamt ebenfalls für einen Rückgang zuordenbaren Tweets, da weniger Accounts die Schwelle von zwei Interaktionen erreichen.

3. Netzwerk und Communities

Um Veränderungen in der deutschsprachigen Twittersphäre festzustellen, haben wir zuerst aufgrund der gesammelten Daten die Twittersphäre als Netzwerk simuliert und darin Communities identifiziert. Dabei haben wir zwei unterschiedliche Ansätze verfolgt (Folgen und Interaktion) und diese miteinander verglichen. In den folgenden Abschnitten wird das Vorgehen und die Ergebnisse beschrieben.

3.1. FollowNetzwerk

Das klassische **FollowNetzwerk** ist am einfachsten zu verstehen: Aus jedem Account wird ein Knoten und wenn ein Account einem anderen folgt, wird dies als eine Verbindung betrachtet. Das Netzwerk basiert auf der Annahme, dass ein Interesse an einem anderen Account besteht, wenn diesem gefolgt wird.

Um die Menge der False-Positives zu reduzieren und das Erfassen sowie berechnen des Netzwerks zu beschleunigen, wurden nur Accounts betrachtet, die mit mehr als 100 Tweets (inklusive Retweets) im Datensatz vertreten sind. Das entspricht durchschnittlich einem Tweet pro Woche, wenn der Account über den gesamten Zeitraum aktiv war. Entsprechend mehr, falls ein Account kürzer aktiv war.

Wir gehen davon aus, dass die meisten Nutzer*innen seltener Folgen und Entfolgen als twittern, wodurch das Folgenetzwerk im Gegensatz zum Interaktionsnetzwerk stabiler ist: Es verändert sich im Laufe der Zeit nur langsam.

Ein weiterer Vorteil des FollowNetzwerks ist die Anzahl der Verbindungen für Accounts, die selten mit anderen Accounts über Tweets interagieren. Dadurch können auch diese mit einer höheren Zuverlässigkeit im Netzwerk verortet werden.

Von 850.713 Accounts, die in den letzten zwei Jahren mehr als 100 deutschsprachige Tweets veröffentlicht haben, konnten für 625.374 Accounts die Folgeverbindungen abgefragt werden. Daraus ergeben sich 112.829.101 Folgeverbindungen zwischen den betrachteten Accounts.

Abbildung 3.1.: Followwetzwerk aus 625.374 Accounts

3.2. Interaktionsnetzwerk

Das zweite Netzwerk, das wir ausgewertet haben, ist das **Interaktionsnetzwerk**. In diesem Netz sind die Knoten ebenfalls die Accounts. Allerdings werden die Verbindungen aufgrund der Interaktion mit den Tweets eines anderen Accounts erstellt. Als Interaktion wurde eine Reply, also eine Antwort auf einen Tweet, ein Retweet, also das Weiterverbreiten, sowie ein Quote-Tweet, das Zitieren eines Tweets, gewertet.

Dieses Netzwerk ändert sich im Gegensatz zum Followwnetzwerk dynamischer und es kommt häufiger zu Verbindungen zwischen Gruppen. Etwa führen kontroverse Themen zu Interaktionen in Form von Replys und Quote-Tweets zwischen Accounts, die sich nicht folgen.

Ein Vorteil des Interaktionsnetzwerks ist die Unabhängigkeit vom Erhebungszeitpunkt. Es kann aus den Tweetdaten selbst generiert werden und somit ein Netzwerk über einen längeren Zeitraum abbilden.

Aufgrund der riesigen Datenmengen bei Interaktionsnetzwerken, haben wir als Grundlage den Monat April für die Jahre 2020, 2021 und 2023 stellvertretend ausgewählt und die Interaktionen miteinander kombiniert, um den gesamten Zeitraum abdecken zu können.

3.2.1. Communities

In der Netzwerkanalyse wird die `community detetection` verwendet, um Nähe- und Distanzverhältnisse unter verschiedenen Aspekten betrachten zu können. Im folgenden wurden die Communities mittels des Modularity Algorithmus (Blondel u. a. (2008)) gebildet. Dieser Algorithmus versucht Gruppen zu finden, die innerhalb besonders intensiv verknüpft sind und zu Accounts in anderen Gruppen nur wenige Verbindungen besitzen.

Diese Communities sind in der folgenden Grafik durch eine Farbe je Community sichtbar.

Durch eine stichprobenartige Betrachtung der größten Cluster wurden zwei Spamcluster identifiziert und von der Auswertung des Tweetvolumens je Community ausgeschlossen.

3.3. Vergleich Follow- und Interaktionsnetzwerke

Grundlage für das Followwnetzwerk sind Daten aus dem März 2023. Wenn nun die beiden unterschiedlichen Netzwerke miteinander verglichen werden, schauen wir durch eine aktuelle Schablone auf die Daten aus der Vergangenheit. Das bietet für die Analyse den Rahmen, um später Aussagen darüber treffen zu können, wie sich die Interaktion über die Zeit hinweg verändert, vielleicht sogar verschoben hat.

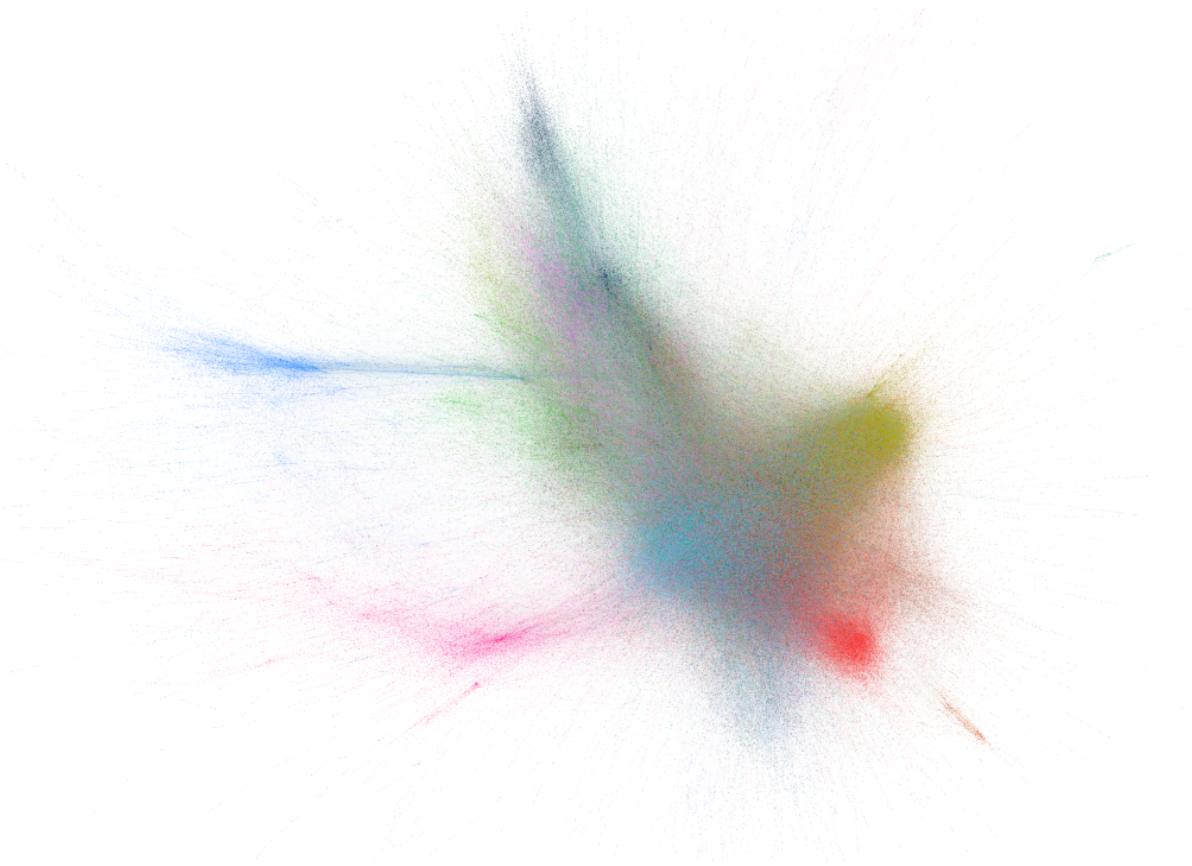
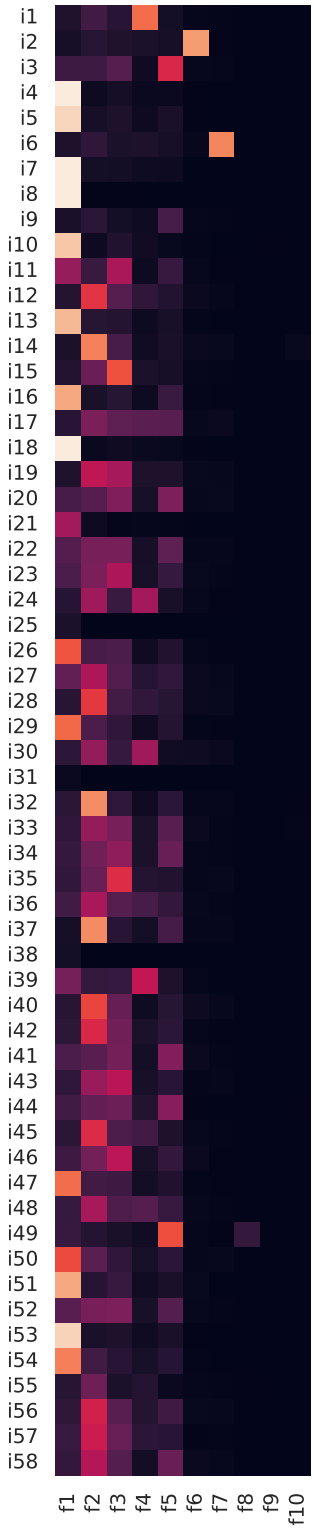


Abbildung 3.2.: Interaktionsnetzwerk aus 533.859 Accounts

Die folgende Grafik zeigt auf der vertikalen Achse die größten Gruppen aus dem Followwnetzwerk. Auf der horizontalen Achse sind die Gruppen angetragen, die sich aus dem Interaktionsnetzwerk (aus den Aprilmonaten 21/22/23) ergeben haben. Je heller die Farbe in einem Feld, desto mehr Personen sind sowohl in dem Cluster des Interaktionsnetzwerks als auch in dem des Followernetzwerk.

```
import pandas as pd
import seaborn

df = pd.read_csv('data/abdeckung-i_mod0_5-f_mod1.csv',index_col=0)

seaborn.set(rc={'figure.figsize':(df.shape[1]/4,df.shape[0]/4)})
seaborn.heatmap(df, annot=False, cbar=False, vmax=80)
```

```
<Axes: >
```


i1
i2
i3
i4
i5
i6
i7
i8
i9
i10
i11
i12
i13
i14
i15
i16
i17
i18
i19
i20
i21
i22
i23
i24
i25
i26
i27
i28
i29
i30
i31
i32
i33
i34
i35
i36
i37
i38
i39
i40
i42
i41
i43
i44
i45
i46
i47
i48
i49
i50
i51
i52
i53
i54
i55
i56
i57
i58
f1
f2
f3
f4
f5
f6
f7
f8
f9
f10

Beispiel: Die Followcommunity f4 befindet sich in Spalte 4: Vergleicht man Spalte 4 mit den Spalten links daneben fällt auf, dass es die erste Spalte ist, die nur wenig helle Flecken in der Vertikalen aufweist. Im Vergleich zu den Follow-Communities links daneben (mit Accounts rund um Unterhaltung, Nachrichten in f1, Politik in f2 und Nachrichten und Unterhaltung in f3), agiert ein großer Teil dieser Accounts, hauptsächlich mit den eigenen Peers, weshalb sich von den 18.903 Accounts in Interaktionscommunity i1 in der Followcommunity f4 wiederfinden.

Die Gruppen, in den Spalten links davon zeichnen sich durch entschieden mehr Interaktionscluster aus, in die sich die enthaltenen Accounts aufteilen.

Eine ähnliche hohe Überscheindung von Follow- und Interaktionsclustern lässt sich bei kleineren Communities feststellen.Etwa bei f6 mit vorwiegend Accounts rund um Österreich, welche sich zu einem sehr großen Teil mit Interaktionscommunity i2 überschneiden oder f7, deren Accounts vor allem mit der Schweiz beschäftigen und die größte Überschneidung mit i6 hat.

3.4. Welche Accounts sind in welchen Communities

Die Identifizierung von Communities in Netzwerkdaten bringt immer Unschärfe mit sich. Die Accounts werden aufgrund ihrer Verbindungen einer Community zugeordnet. Im Falle von Followverbindungen beeinflussen sie den Teil der Follower nur indirekt durch ihre Inhalte. Wem sie folgen entscheiden sie selbst. Je nach Accountgröße, macht dies aber nur einen geringen Anteil aus. Bei Interaktionen ist es ähnlich. Je populärer ein Account ist, desto geringer ist die Rolle der eigenen Interaktionen, weil die Menge der Interaktionen von anderen mit dem Account viel größer ist. Entsprechend vorsichtig muss man bei der Interpretation der Daten sein. Die Zusammensetzung der Communities erlaubt Aussagen über die Community als ganzes zu treffen. Von der Beschreibung der Community, der ein Account zugeordnet wurde, auf Eigenschaften des Accounts zu treffen, funktioniert nicht ohne weiterer Überprüfung. Accounts werden jedoch nicht zufällig einer Community zugeordnet, sondern aufgrund der Verbindungen. Die Zuordnung kann daher ein Hinweis sein, dass aus einer Community ein großes Interesse an einem Account und/oder dessen Inhalten besteht.

Zur Orientierung folgen Tabellen zu den einzelnen Communities mit einer kurzen Beschreibung unserer Beobachtung sowie den jeweiligen Top-Accounts.

Die Communities sind nach der Anzahl der Tweets über den gesamten Untersuchungszeitraum sortiert. In der Tabelle werden die 10 Accounts mit den meisten Followern im Followwnetzwerk angezeigt. Da es nicht die Anzahl an Followern innerhalb der jweiligen Community ist, bedeutet es nicht, dass sie in der jeweiligen Community am einflussreichsten sind, sondern dass sie die einflussreichsten Accounts im Gesamtnetzwerk sind, die vom Algorithmus der jeweiligen Community zugeordnet wurden.

3.4.1. Followcommunity f4

- 2021: 4140911 Tweets von 32819 Accounts
- 2022: 6521449 Tweets von 50204 Accounts
- 2023: 6814446 Tweets von 48594 Accounts

In dieser Community befinden sich überwiegend rechte Accounts.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
SWagenknecht	682572	61590
reitschuster	189274	43221
argonerd	119256	41762
jreichelt	216944	40662
BILD	1.98427e+06	37592
Alice_Weidel	237231	36719
janfleischhauer	213864	34286
RolandTichy	321930	33730
HGMaassen	134121	32864
Tim_Roehn	95160	32094

3.4.2. Followcommunity f1

- 2021: 5477397 Tweets von 132332 Accounts
- 2022: 5259195 Tweets von 148956 Accounts
- 2023: 4127268 Tweets von 122224 Accounts

Vor allem Accounts rund um Unterhaltung, Gaming, Youtube und Fußball.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
rezomusik	573151	60500
NetflixDE	2.47126e+06	46790
damitdasklaas	1.84413e+06	42627
Gronkh	1.34834e+06	37319
MontanaBlack	1.39386e+06	36602
unge	2.40999e+06	33132
impf_progress	123189	31067
DFB_Team	3.28852e+06	30719
HandOfBlood	726750	26492
TANZVERBOTcf	233134	24928

3.4.3. Followcommunity f3

- 2021: 4531454 Tweets von 50352 Accounts
- 2022: 4163189 Tweets von 57214 Accounts
- 2023: 3807125 Tweets von 46701 Accounts

Mischung aus Nachrichten/Politik und Unterhaltung.

Nutzername	Follower gesamt	Follower im Follownetzwerk
c_drosten	971464	122932
janboehm	2.71917e+06	120941
Karl_Lauterbach	1.08773e+06	120869
Der_Postillon	1.3814e+06	96711
heuteshow	1.61643e+06	91411
maithi_nk	455148	85449
Volksverpetzer	351294	72295
Luisamneubauer	450532	71999
ABaerbockArchiv	393470	69743
extra3	1.08022e+06	68650

3.4.4. Followcommunity f5

- 2021: 4480355 Tweets von 36787 Accounts
- 2022: 3898346 Tweets von 41298 Accounts
- 2023: 3223065 Tweets von 33987 Accounts

Accounts rund um die Pandemie und was im englischsprachigen manchmal als TPOT (That Part of Twitter) bezeichnet wird. Im deutschsprachigen manchmal als Schmunzeltwitter bezeichnet.

Nutzername	Follower gesamt	Follower im Follownetzwerk
CiesekSandra	165441	41805
DieMaus	130707	33621
kriegundfreitag	108837	32244
Nell781	68946	29574
narkosedoc	71796	29164
drluebbers	71237	28063
Hoellenaufsicht	72287	28029
Lam3th	66860	27794
SchwesterFD	66553	25976
BahnAnsagen	257264	25436

3.4.5. Followcommunity f2

- 2021: 3050046 Tweets von 70415 Accounts
- 2022: 3498247 Tweets von 79261 Accounts
- 2023: 2648734 Tweets von 65005 Accounts

Nachrichten und Politik ohne Unterhaltung.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
tagesschau	4.02344e+06	109031
derspiegel	3.13181e+06	79682
ABaerbock	600810	77311
zeitonline	2.58239e+06	76921
Bundeskanzler	717572	70836
SZ	1.87894e+06	67588
rki_de	595670	59191
OlafScholz	639579	58100
ZDFheute	1.04455e+06	54888
c_lindner	687018	54211

3.4.6. Followcommunity f6

- 2021: 1116159 Tweets von 14849 Accounts
- 2022: 1001144 Tweets von 16432 Accounts
- 2023: 996846 Tweets von 14287 Accounts

Überwiegend Accounts aus ÖSterreich.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
ArminWolf	588768	31501
florianaigner	63397	23933
florianklenk	338663	23905
sebastiankurz	471132	19226
Martin_Moder	57284	19056
vanderbellen	298639	17395
ShouraHashemi	43220	15204
MartinThuer	143960	14856
corinnamilborn	160906	14006
brodnig	64743	12860

3.4.7. Followcommunity 57220.0

- 2021: 425538 Tweets von 22846 Accounts
- 2022: 589075 Tweets von 28846 Accounts
- 2023: 397708 Tweets von 22792 Accounts

Vor allem englischsprachige Accounts. Einige Spamaccounts, die automatisiert twittern. Die Community wurde von der Auswertung ausgeschlossen.

Nutzername	Follower gesamt	Follower im Followntezwerk
threadreaderapp	646637	10206
RaminNasibov	740483	9216
Independent	3.62886e+06	5643
thehill	4.47475e+06	5499
Havenlust	292143	5422
fredykoglin	104342	5278
gmail	6.99251e+06	4840
ParisAMDParis	289231	4547
nypost	2.89307e+06	4474
lrinaldetti	17154	4392

3.4.8. Followcommunity f7

- 2021: 484930 Tweets von 9793 Accounts
- 2022: 420583 Tweets von 10451 Accounts
- 2023: 395095 Tweets von 8854 Accounts

Vor allem Accounts aus der Schweiz.

Nutzername	Follower gesamt	Follower im Followntezwerk
NZZ	487508	27354
srfnews	381973	8916
RepublikMagazin	37281	8009
FabianEberhard	33848	7049
tagesanzeiger	206564	6717
BAG_OFSP_UFSP	185009	6686
20min	469956	6182
viktorgiacobbo	212410	5964
SRF	160446	5904
watson_news	144331	5819

3.4.9. Followcommunity 61835.0

- 2021: 477267 Tweets von 33973 Accounts
- 2022: 382353 Tweets von 38027 Accounts
- 2023: 226161 Tweets von 26339 Accounts

Accounts rund um kpop. Bei der Auswertung nicht beachtet.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
convomf	1.10883e+06	5507
starfess	698734	4110
convomfs	1.51205e+06	3877
ShopeeID	907152	2922
leadernim_yang	105169	2887
tanyakanrl	910269	2825
nctzenbase	823218	2572
OD7SSEUS	34735	2471
tuanxcoco	175151	2431
markbeomnyoung	107166	2405

3.4.10. Followcommunity 60818.0

- 2021: 20639 Tweets von 1381 Accounts
- 2022: 203212 Tweets von 3098 Accounts
- 2023: 9397 Tweets von 940 Accounts

Vor allem nicht deutschsprachige Donbelle Fanaccounts. Bei der Auswertung nicht beachtet.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
DonBelleOFC	264520	2205
fearlessDB___	27821	1649
bacsilowg	19040	1388
ABSCBNNews	8.93812e+06	1359
DonBelleTagSen_	57259	1298
hesintoher	261574	1278
onlyfordonbelle	18302	1271
mikaydb	17279	1267
msbabybabe	10726	1141
dunbilsus	9868	1086

3.4.11. Followcommunity 40025.0

- 2021: 51413 Tweets von 5590 Accounts
- 2022: 69394 Tweets von 6772 Accounts
- 2023: 49947 Tweets von 5721 Accounts

Vor allem nigerische Accounts. Bei der Auswertung nicht beachtet.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
Naija_PR	3.84879e+06	1630
gyaigyimii	1.33588e+06	1483
ellyserwaaa	180734	1031
SneakerNyame_	400710	1026
akreana_	838567	1014
AsieduMends	368260	1010
Opresii	421368	969
Kayjnr10	327343	961
iamyourspec	20045	955
thatEsselguy	284674	931

3.4.12. Followcommunity 33000.0

- 2021: 34668 Tweets von 3570 Accounts
- 2022: 36590 Tweets von 5113 Accounts
- 2023: 20006 Tweets von 3528 Accounts

Überwiegend indische Accounts. Bei der Auswertung nicht beachtet.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
BCCI	2.08715e+07	1910
IamRealHasan	18540	1864
TeamPratikOFC	9475	1605
Judas_1994	13473	1559
PratikOffcialFC	20372	1523
beingGavy_	18772	1313
Priyankaa112	7996	1288
lokeshsiddh1408	4680	1127
ritam_de_scribe	9054	1098
pratikismine	5933	1058

3.4.13. Followcommunity f8

- 2021: 40364 Tweets von 1423 Accounts
- 2022: 28376 Tweets von 1207 Accounts
- 2023: 22390 Tweets von 755 Accounts

Vor allem Accounts rund um kink.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
herrin_***	48534	1248
To***	65431	1096
Zu***	22195	1057
HerrinS***	27784	1039
herrin_so***	23636	1012
HerrinLe***	11135	893
tski***	152896	839
Lady_Ch***	7459	815
HerrinLa***	16651	813
goddess_***	6709	808

3.4.14. Followcommunity 70656.0

- 2021: 6347 Tweets von 363 Accounts
- 2022: 83827 Tweets von 1124 Accounts
- 2023: 700 Tweets von 182 Accounts

Überwiegend Accounts mit dem Fokus auf die ethnischen Säuberungen in der Region Tigray in Ethopien. Bei der Auswertung nicht beachtet.

Nutzername	Follower gesamt	Follower im Followwnetzwerk
peterbelayneh	28108	1220
Nohelema	15563	1137
tekian_	17016	1133
Great_tgray	19388	1127
gebre_krstos	17012	1119
meronawit_Axum	16079	1074
KibatTigraweyti	12502	1004
Rahel122224	13967	998
FerwiniB	15483	980
BashaDesta	54670	968

3.4.15. Followcommunity f9

- 2021: 14250 Tweets von 825 Accounts
- 2022: 12006 Tweets von 859 Accounts
- 2023: 3904 Tweets von 574 Accounts

Überwiegend türkischsprachige Accounts, die zwischendurch auf auf deutsch schreiben und deshalb als Teil der deutschsprachigen Twittersphäre betrachtet wurden.

Nutzername	Follower gesamt	Follower im Follownetzwerk
UensalArik	189774	1022
barbarosansalfn	719010	932
TurkishMinuteTM	232633	752
KronosHaber	103548	631
MehmetEfe_Caman	94302	614
ReisenderMann	5251	598
hy_06_	22175	582
PatrickJane02	21252	569
SALIHBAYRAK7102	24633	556
bkenes	198088	550

3.4.16. Followcommunity 72341.0

- 2021: 2461 Tweets von 97 Accounts
- 2022: 11787 Tweets von 367 Accounts
- 2023: 11754 Tweets von 244 Accounts

Vor allem Fans des kasachischen Sänger Dimash Qudaibergen. Bei der Auswertung nicht beachtet.

Nutzername	Follower gesamt	Follower im Follownetzwerk
SabineLuhs	6219	561
nikitaLila	8886	555
Dimaterry1	4603	542
Dear_3_Lara	3703	527
DearsRamo	3227	526
RamoSos	2177	519
FrencyDear	2209	514
monica_valdi	2153	509
MarinaDears	3218	503
Vera32746352	2206	502

3.4.17. Followcommunity f10

- 2021: 7478 Tweets von 318 Accounts
- 2022: 11974 Tweets von 483 Accounts
- 2023: 6086 Tweets von 381 Accounts

Überwiegend Accounts aus Luxemburg.

Nutzername	Follower gesamt	Follower im Followntezwerk
Wort_LU	30246	870
lessentielde	8691	591
RTLlu	52826	590
svnee	6927	579
tageblatt_lu	13406	575
diego_bxl	9404	571
gouv_lu	34560	502
PoliceLux	20711	464
100komma7	9589	447
reporter_lu	5084	433

4. Twittervielfalt

Vielfalt ist die Menge an unterschiedlichen Stimmen zu unterschiedlichen Themen. Vielfalt der Accounts und Meinungen lässt sich nicht über die Struktur feststellen. Wir haben als Annäherung die Größe und Aktivität der Interaktionscommunities verwendet.

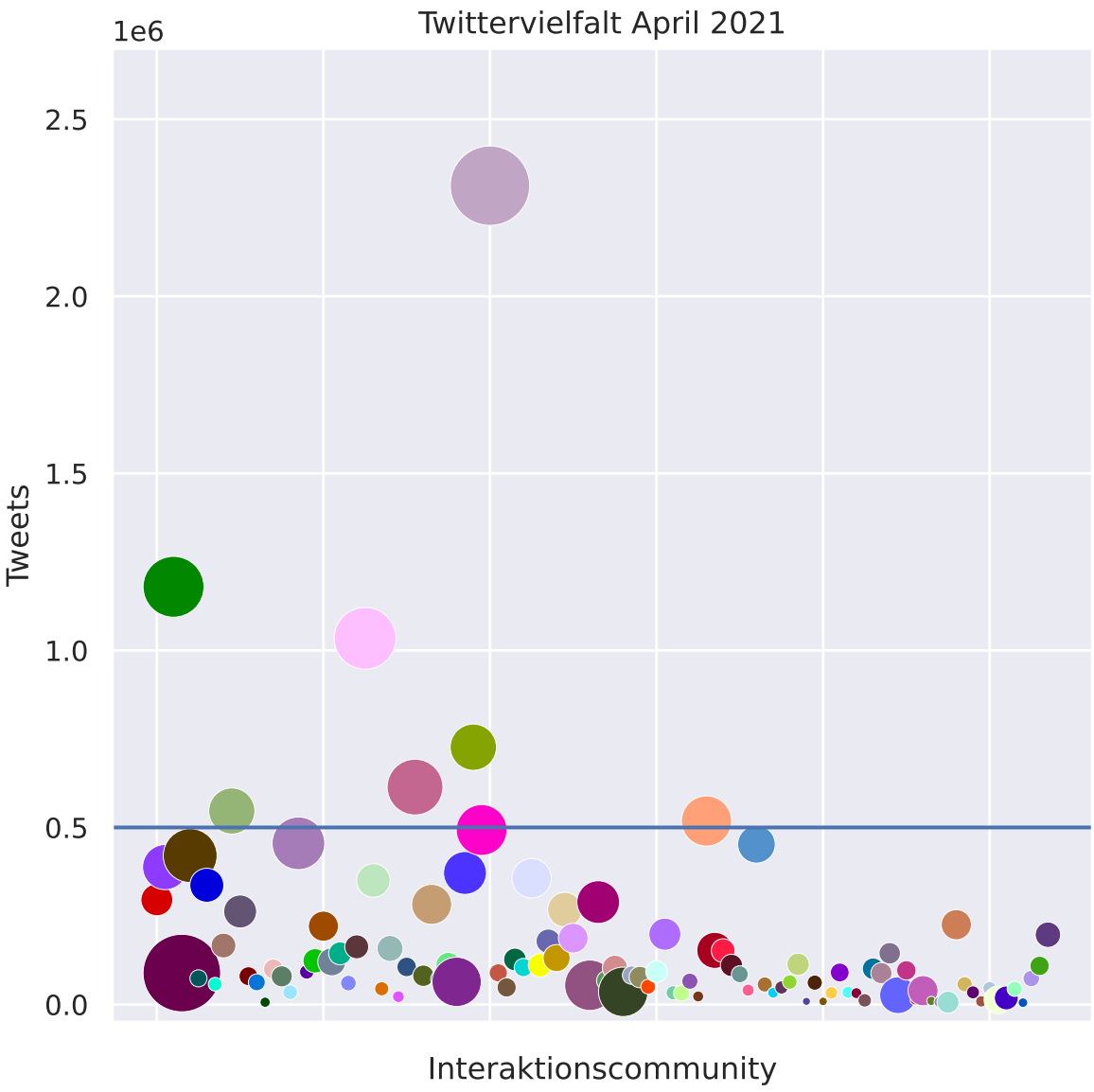
Interaktionscommunities entstehen, wenn Accounts häufiger miteinander interagieren. In den meisten Fällen passiert dies aufgrund von gemeinsamen Interessen. Aktivität der kleinen und mittleren Interaktionscommunities sind ein Hinweis, dass sich Leute zu unterschiedlichen Themen austauschen. Wenige große Interaktionscommunities deuten hingegen auf weniger Vielfalt hin.

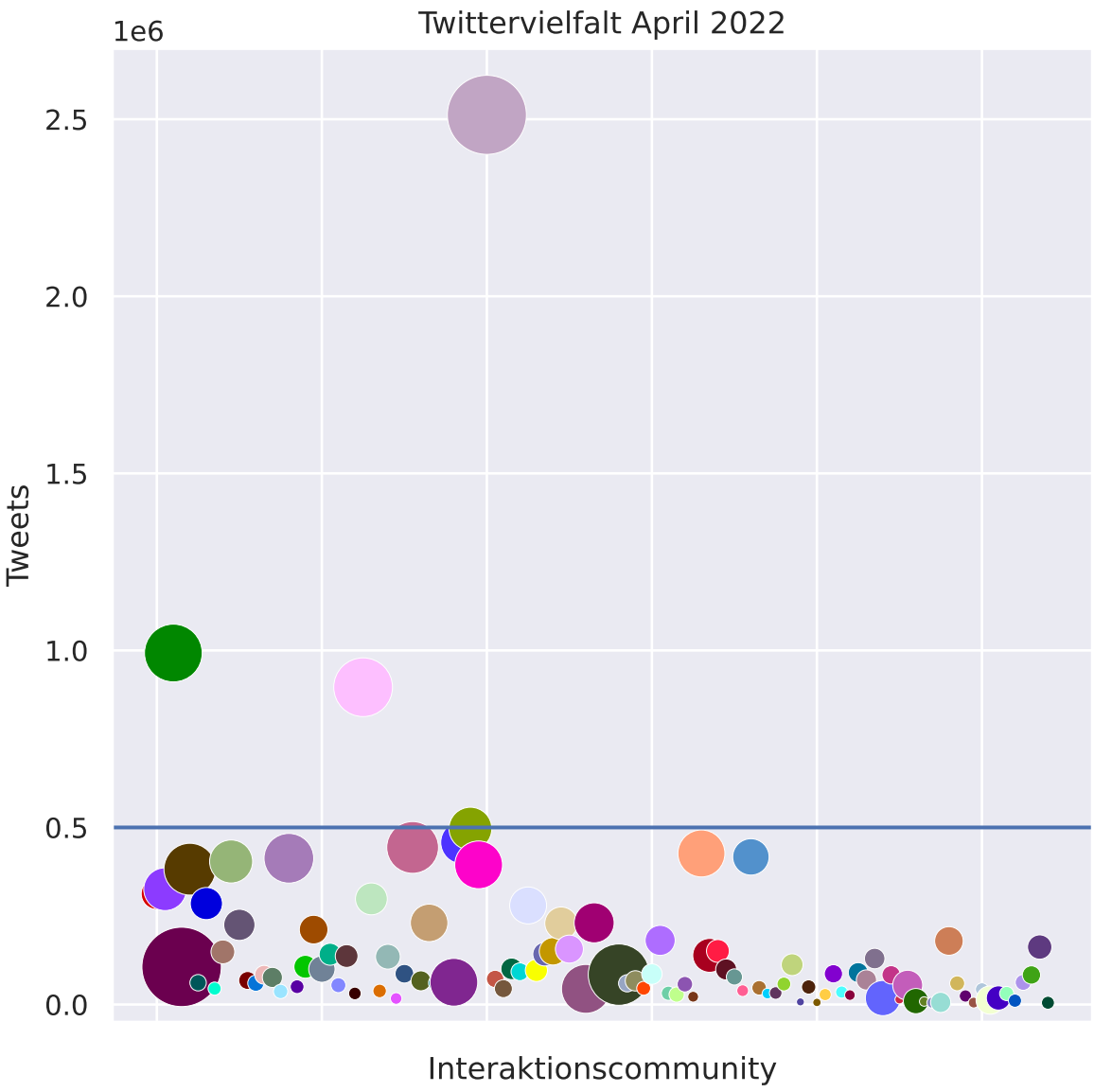
In den drei Grafiken sind jeweils die Interaktionscommunities für drei Wochen aus dem April je Jahr dargestellt. Die Größe des Kreises ist die Anzahl der Accounts in der jeweiligen Community. Je weiter oben ein Kreis, desto mehr Tweets hat die jeweilige Community im jeweiligen Zeitraum veröffentlicht. Die Linie bei 500.000 Tweets dient der Orientierung. Man kann sehen, wie die Aktivität der mittleren und kleinen Communities immer weiter abnimmt, wodurch die Dominanz der lautesten Community wächst.

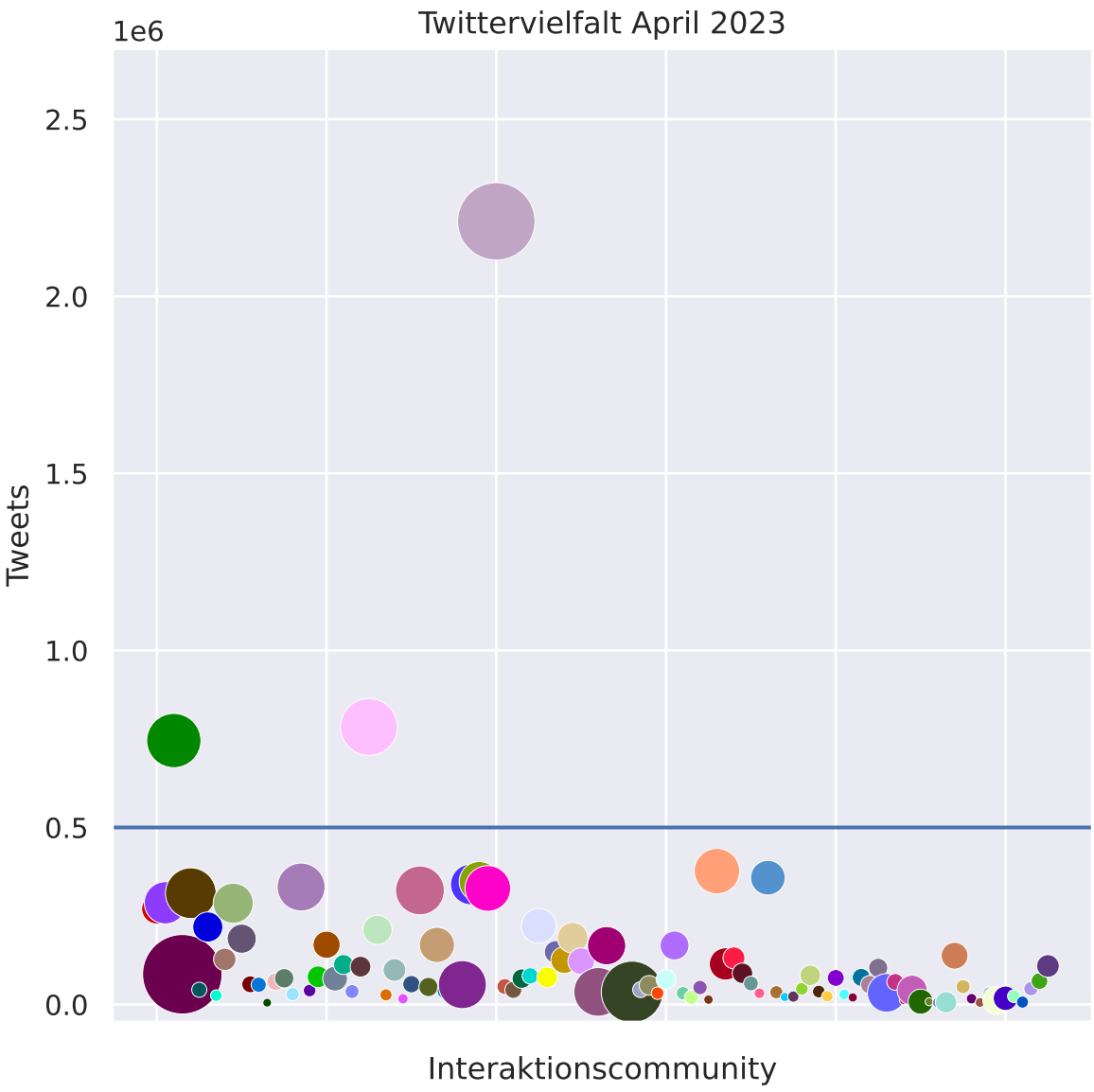
Die kleinen und mittleren Communities mit mehr als 5000 Tweets und weniger als 10000 Accounts sind im Durchschnitt von 2021 auf 2022 um 5% und von 2022 auf 2023 um 11% geschrumpft während die größte Community erst um 15% gewachsen und dann um nur 7% geschrumpft ist.

```
import json
import pandas as pd
import matplotlib.pyplot as plt
import seaborn
import colorcet as cc
seaborn.set()
seaborn.set(rc={'figure.figsize':(7,7)})

with open('data/interaktion_color_map.json', 'r') as f:
    color_map = json.load(f)
    color_map = {int(k.replace('.0','')):v for k,v in color_map.items()}

for year in [2021,2022,2023]:
    df = pd.read_csv(
```

```
    f'data/twittervielfalt_{year}.csv')

    seaborn.scatterplot(
        df,
        x=df.index,
        y="Tweets",
        size="Accounts",
        hue="community_interaction",
        palette = color_map,
        sizes=(10, 1000),
        legend=None
    ).set(
        title=f"Twittervielfalt April {year}",
        ylim=(-50000, 2700000),#2700000)
        xlabel='Interaktionscommunity',
        xticklabels=[]
    )
    plt.axhline(y=500000)
    plt.show()
```


Twittervielfalt April 2021
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Tweets
Interaktionscommunity


Twittervielfalt April 2022
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Tweets
Interaktionscommunity


Twittervielfalt April 2023
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Tweets
Interaktionscommunity

5. Zeitverlauf - Tweets

5.1. Aprilstichprobe

Der April wurde gewählt, weil er 2023 der Monat mit den vollständigsten Daten ist. April 2021 war lange bevor Elon Musk klare Absichten gezeigt hat Twitter übernehmen zu wollen. Im April 2022 wurde erstmals bekannt, dass Elon Musk ein größeres Aktienpaket gekauft hat und einen Sitz im Aufsichtsrat angeboten bekam. Die Ablehnung des Sitzes sowie Ankündigung, dass er Twitter kaufen möchte, kamen ebenfalls im April 2022. Der April 2023 hat ein halbes Jahr Abstand zur rechtlichen Übernahme am 27. Oktober 2022, sodass sowohl die Nutzer*innen Zeit hatten die Übernahme mitzubekommen als auch Entscheidungen von Musk von Twitter umgesetzt werden konnten.

5.1.1. Weniger Tweets insgesamt

Der Median für den April 2023 liegt bei 1.537.203 deutschsprachigen Tweets pro Tag. Das sind 14% weniger als im April 2022 und 17% weniger als im April 2021. Zwischen April 2021 und April 2022 liegt der Rückgang bei 4%.

Für den Median wurden die Tweets nicht bereinigt. Es handelt sich um die Gesamtanzahl, wie sie Twitter über die API-Schnittstelle ausgegeben hat.

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import seaborn
seaborn.set()

figsize=(1,2)

ax = pd.DataFrame([{"Tweets":38645678},{"Tweets":37685804},{"Tweets":32271081}], index=[20
    ).plot(
        kind='bar',
        figsize=figsize,
        title='Tweets April',
```

```
        legend=False
        )
```

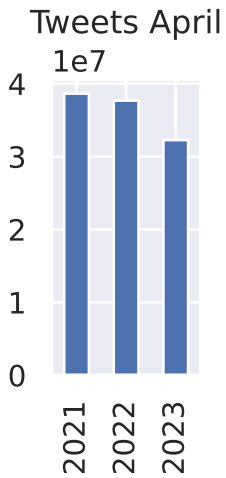


Abbildung 5.1.: Tweets gesamt

5.1.2. Laute Gruppe

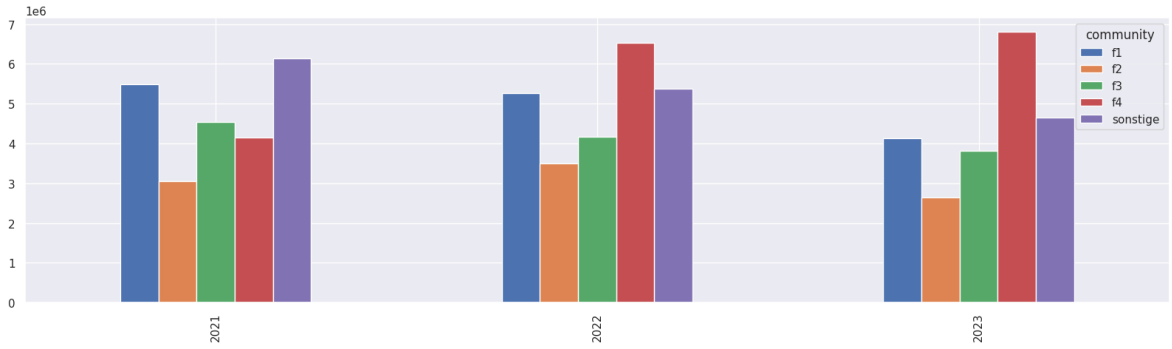


Abbildung 5.2.: “April Tweets”

Wie in Abbildung 5.2 zu sehen ist, hat das Tweetvolumen bei fast allen Gruppen über die letzten drei Jahre abgenommen. Einzige Außnahme ist der Cluster 74409, aus dem das Tweetvolumen von 2021 auf 2023 um 64% gestiegen ist. Eine nähere Betrachtung hat ergeben, dass dem Cluster vor allem rechte Accounts zugeordnet wurden. Mehr zur Erstellung der Cluster im Kapitel 3. 2021 kamen von drei anderen Clustern mehr Tweets, 2023 ist er mit Abstand der aktivste. 6,8 Millionen Tweets wurden von den enthaltenen Accounts in den 22 betrachteten Tagen veröffentlicht.

5.2. Gesamtverlauf

Über den gesamten Zeitraum betrachtet, werden die Erkentnisse aus der Aprilstichprobe bestätigt. Eine Community dominiert Twitter immer stärker. Zumindest was das Tweetvolumen betrifft.

5.2.1. Perspektive Followwnetzwerk

In der folgenden Abbildung ist der relative Anteil des Tweetvolumens je Follow-Community und Monat abgebildet. Die Tweets sowohl über die Auswahl (mindestens 100 deutschsprachige Tweets im Untersuchungszeitraum) als auch über die Cluster (Entfernung von zwei Spam-Clustern) bereinigt. Da die Tweets clusterunabhängig gesammelt wurden, kann das relative Volumen auch für unvollständig erfasste Zeiträume betrachtet werden.

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import seaborn
seaborn.set()

figsize=(10,5)

ax = pd.read_csv(
    'data/tweets_pro_monat_nach_communities.csv',
    index_col='created_at'
    ).plot(
        figsize=figsize,
        title='Relativer Anteil der Tweets je Follow-Community',
        legend=False
        )

ax.yaxis.set_major_formatter(mtick.PercentFormatter())
```

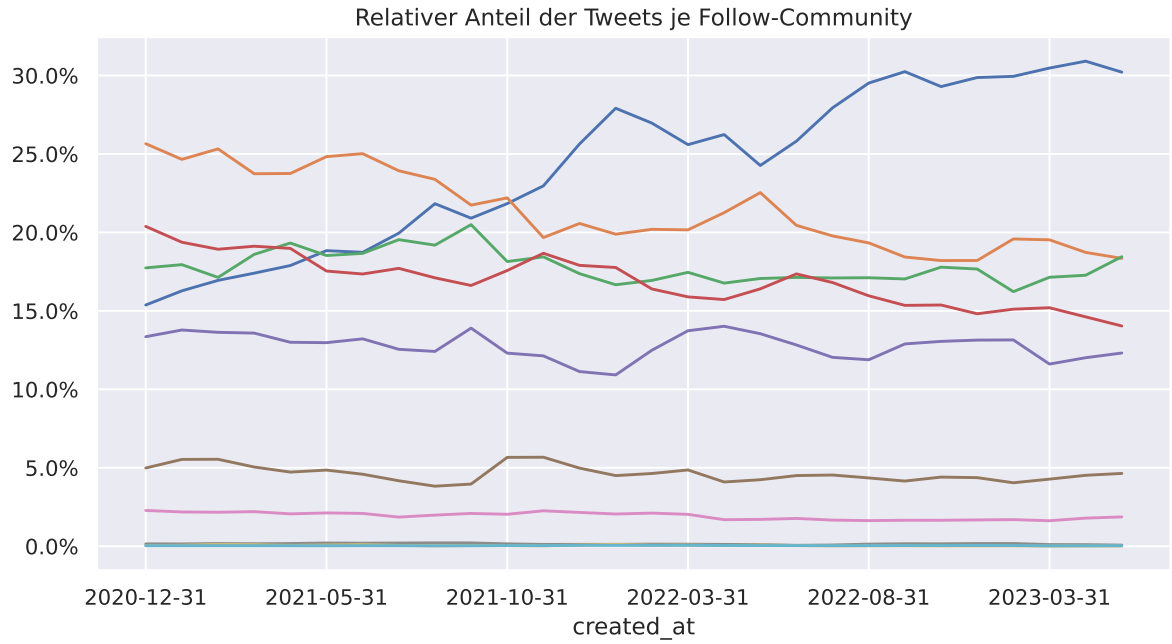


Abbildung 5.3.: Relativer Anteil Tweets je Followcommunity

Wie schon in der Aprilstichprobe, sieht man auch im Gesamtverlauf wie die Community im Laufe von 2021 zur lautesten Gruppe auf Twitter geworden ist. Das relative Volumen der anderen Communities nahm entsprechend ab.

5.2.2. Perspektive Interaktionsnetzwerk

Werden statt des Follow-Netzwerks die dynamischeren und deshalb meist kleineren Interaktionsnetzwerke betrachtet, zeigt sich auch dort eine Community, die weiter mehr Tweets veröffentlicht als alle anderen Communities. Dies liegt vor allem an der Größe der Community. Während der Großteil der Follow-Communities in kleinere Interaktions-Communities zerfallen, ist diese Community auch über Interaktionen eng verknüpft.

```
ax = pd.read_csv(
    'data/tweets_pro_monat_nach_interaction_communities.csv',
    index_col='created_at'
    ).plot(
        figsize=figsize,
        title='Relativer Anteil der Tweets je Interaktions-Community',
        legend=False
        )
```

```
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
```

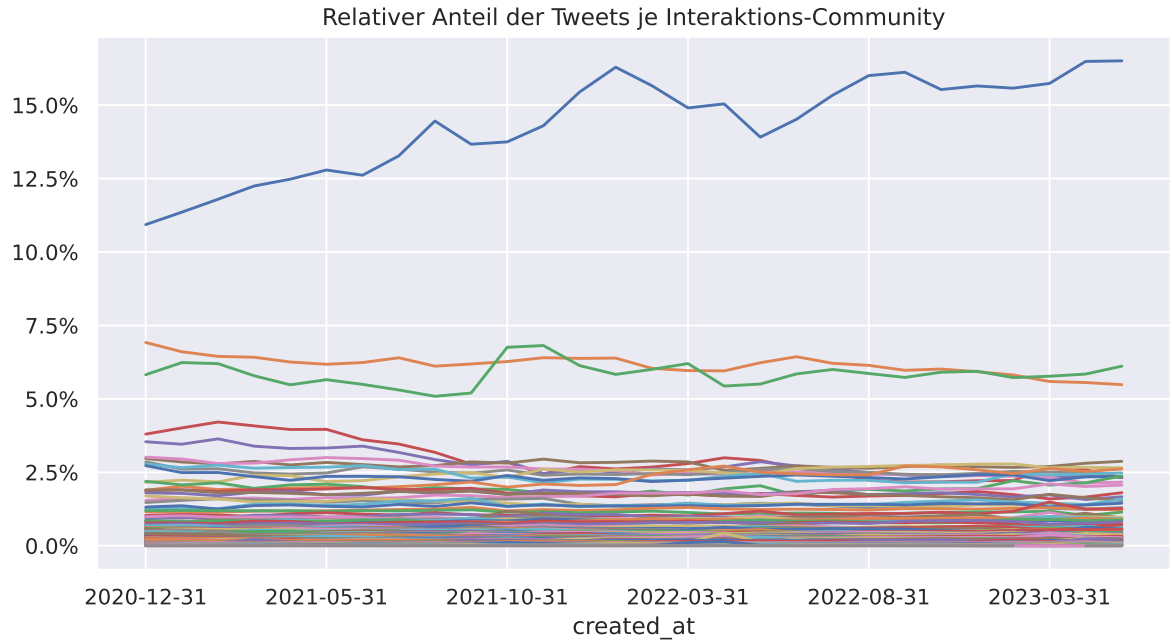


Abbildung 5.4.: Relativer Anteil Tweets je Interaktionscommunity

Im gesamten Datensatz befinden sich 80 Millionen Tweets, die dieser Community zugeordnet werden. Für die nächste Community sind es weniger als die Hälfte (35 Millionen).

5.3. Fehlerrate

Um einschätzen zu können, wie viele deutschsprachige Tweets durch die Mindestanzahl von 100 Tweets im Datensatz ausgeschlossen wurden, haben wir jeweils eine Zufallsstichprobe von 100 Accounts aus den eingeschlossenen und 100 Accounts aus den ausgeschlossenen Tweets genommen und per Hand codiert.

Von den 108.602.563 Tweets von 9.787.791 Accounts aus den Aprilstichproben kommen 38.399.751 Tweets von 9.310.335 Accounts, die mit weniger als 100 Tweets im Datensatz vertreten sind und 70.202.812 von 477.456 Accounts mit mehr als 100 Tweets im Datensatz.

5.3.1. Stichprobe weniger als 100 Tweets

14 Accounts (29 Tweets) gab es nicht mehr, 78 ware nicht deutschsprachig (212 Tweets), 3 privat (3 Tweets) und 5 deutschsprachig (7 tweets).

5.3.2. Stichprobe mehr als 100 Tweets

1 Account (162 Tweets) nicht mehr erreichbar, 12 Accounts nicht deutschsprachig (192 Tweets), 87 deutschsprachig (15052 Tweets).

6. Fazit

Die deutschsprachige Twittersphäre verändert sich. In Kapitel 4 konnten wir sichtbar machen, wie die Interaktionscommunity aus überwiegend rechten Accounts über die letzten Jahre lauter geworden ist, während die meisten anderen Interaktionscommunities immer weniger Tweets veröffentlicht haben. Warum das so ist, können uns die Daten jedoch nicht verraten. Es kann an Elon Musks Aussagen liegen, von denen sich rechte Gruppen angesprochen fühlen. Es kann mit einem gesellschaftlichen Wandel zusammenhängen. Es kann an Algorithmen liegen, die die Tweets aus dem vorwiegend rechten Cluster mehr Accounts anzeigen, wodurch sich die Wahrscheinlichkeit von Interaktionen erhöht. Auch Feedbackschleifen können eine Rolle spielen: Wenn Menschen sich auf Twitter nicht mehr wohl fühlen und die Plattform verlassen, wird es für die, die noch dort sind leerer und unangenehmer, sodass die Wahrscheinlichkeit steigt, dass sie ebenfalls gehen.

Neben dem allgemeinen Trend von weniger deutschsprachigen Tweets, haben wir in Kapitel 5 gezeigt, dass die steigende Dominanz der Gruppe aus vorwiegend rechten Accounts sowohl über den gesamten Untersuchungszeitraum als auch bei der Betrachtung von Followcommunities sichtbar ist.

Die Auswertung von Vielfalt auf struktureller Ebene und Tweetvolumen im Zeitverlauf basieren auf der Identifikation von Communities. Das Vorgehen und einzelne Communities haben wir in Kapitel 3 ausgeführt. Indem wir uns angeschaut haben, wie sich Followcommunities mit Interaktionscommunities überschneiden, konnten wir sehen, welche Communities vor allem untereinander interagieren, als auch welche Communities aufgrund der Interaktionen weiter unterscheiden lassen. Der Prozess der Clusteridentifizierung bringt jedoch immer eine Unschärfe mit sich. Das Netzwerk ist bereits eine starke Reduktion der Realität. In den Daten befindet sich keine Bedeutung der jeweiligen Folgeverbindungen und Interaktionen. Der Algorithmus probiert dann die in sich am stärksten vernetzten Cluster zu identifizieren. Kann dabei aber gar nicht alle Möglichkeiten durchrechnen, weil das zu lange dauern würde. Bei Netzwerken mit vielen Verbindungen und klaren Clustern ähneln sich die Ergebnisse bei jedem Durchgang. Je geringer die Modularität des Netzwerks, desto höher sind die Veränderungen aufgrund der Reihenfolge, in der gerechnet wird. Auch die Datenbasis spielt eine wichtige Rolle. Wir haben in Kapitel 2 die uns bekannten Einschränkungen aufgrund von fehlenden Tagen in der Erhebung, den Erhebungszeitpunkten, die ungenaue Sprachklassifizierung und das Filtern des Datensatzes beschrieben. Trotz intensiver Beschäftigung mit den Daten, ist es nie möglich unbekannte Einschränkungen auszuschließen. Durch die Beschreibung des Vorgehens und Teilen des Codes, haben wir nachvollziehbar gemacht, wie wir zu den Ergebnissen gekommen sind.

Mit Ausnahme der Beschreibungen der Communities und der Überprüfung der Fehlerrate, stützt sich die Auswertung auf Strukturen, nicht Inhalten. Um zu verstehen, warum sich die deutschsprachige Twittersphäre verändert hat, wie sie sich verändert hat, ist eine inhaltliche Auseinandersetzung mit Tweets und Akteuren notwendig, die wir im Rahmen dieser Datenexploration nicht umsetzen konnten. Unsere Ergebnisse sind ein guter Ausgangspunkt, um sich mit inhaltlich mit den Veränderungen auseinanderzusetzen.

References

Alshaabi, Thayer, David Rushing Dewhurst, Joshua R. Minot, Michael V. Arnold, Jane Lydia Adams, Christopher M. Danforth, und Peter Sheridan Dodds. 2020. „The growing echo chamber of social media: Measuring temporal and social contagion dynamics for over 150 languages on Twitter for 2009-2020“. *CoRR* abs/2003.03667. https://arxiv.org/abs/2003.03667.

Blondel, Vincent D, Jean-Loup Guillaume, Renaud Lambiotte, und Etienne Lefebvre. 2008. „Fast unfolding of communities in large networks“. *Journal of Statistical Mechanics: Theory and Experiment* 2008 (10): P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008.

Hammer, Luca. 2020. „Vermessung der deutschsprachigen Twittersphäre“. https://lucahammer.com/ba2020.

Pfeffer, Juergen, Angelina Mooseder, Jana Lasser, Luca Hammer, Oliver Stritzel, und David Garcia. 2023. „This Sample seems to be good enough! Assessing Coverage and Temporal Reliability of Twitter's Academic API“. https://arxiv.org/abs/2204.02290.

A. Code Tweetsammlung

```
import os
scriptdir =  os.path.dirname(os.path.abspath(__file__))
os.chdir(scriptdir)

from datetime import datetime, timedelta
import json
import json_lines
from TwitterAPI import TwitterAPI, TwitterPager

with open("config.json", 'r') as configfile:
    config = json.load(configfile)

api = TwitterAPI(config['twitter']['api_key'],
                 config['twitter']['api_secret_key'],
                 auth_type='oAuth2'
                )

def collect_tweet_ids(search_term, filename='', since='', since_id='', max_id=''):
    log_message(
        f'Starting. {search_term}, {filename}, {since_id}'
    )

    if since != '':
        query = '{0} since:{1}'.format(search_term, since)
    else:
        query = search_term

    r = TwitterPager(api, 'search/tweets', {'q': query,
                                            'count':100,
                                            'since_id':since_id,
                                            'max_id':max_id,
                                            'result_type':'recent'})
    n = 0
```

```
    if filename == '':
        filename = search_term +'.jsonl'

    with open(filename, 'a', encoding='utf-8') as f:
        for item in r.get_iterator(wait=2.2):
            n += 1
            if 'text' in item:
                max_id = item['id']
                json.dump({'created_at' : item['created_at'],
                           'id' : item['id'],
                           'user_id' : item['user']['id'],
                           'at_id' : item['in_reply_to_status_id'] if 'in_reply_to_status_
                           'rt_id' : item['retweeted_status']['id'] if 'retweeted_status'
                           'qt_id' : item['quoted_status']['id'] if 'quoted_status' in ite
                           'text': item['text']
                          }, f)
                f.write('\n')
            elif 'message' in item and item['code'] == 88:
                log_message(f'Broke at {max_id}')
                log_message('SUSPEND, RATE LIMIT EXCEEDED: %s\n' % item['message'])
                break
    log_message(
        f'Finished. Collected {n} Tweets. Max ID: {max_id}'
    )
    return ()

def log_message(message):
    with open('collection_log.txt', 'a') as f:
        f.write(
            f'{datetime.now().strftime("%Y-%m-%d %H:%M:%S")} '
            f'{message}'
            '\n'
        )

def load_tweet_ids_from_jsonl(files):
    ids = set()

    for file in files:
        with open(file, 'rb') as f:
            for tweet in json_lines.reader(f, broken=True):
                ids.add(tweet['id'])
```

```
    return (ids)

def get_highest_id(filename):
    ids = load_tweet_ids_from_jsonl([filename])
    idslist = list(ids)
    idslist.sort()
    return(idslist[-1])

def get_first_id_in_file(filename):
    with open(filename) as f:
        first_line = f.readline()
    tweet = json.loads(first_line)
    return(tweet['id'])

def create_filename(date):
    prefix = 'lang_de'
    postfix = 'IDs.jsonl'
    filename = f'{prefix}-{date.strftime("%Y-%m-%d")}_{postfix}'
    return (filename)

if __name__ == '__main__':
    search_term = 'lang:de'
    yesterdays_file = create_filename(datetime.now() + timedelta(days=-1))
    todays_file = create_filename(datetime.now())
    highest_id = get_first_id_in_file(yesterdays_file)
    collect_tweet_ids(search_term,filename=todays_file, since_id=highest_id)
```

B. Code Followsammlung

```
from TwitterAPI import TwitterAPI, TwitterPager
import yaml
import json
import os
from tqdm.auto import tqdm
import pandas as pd
import json_lines
import time

with open("config.yaml", 'r') as ymlfile:
    config = yaml.safe_load(ymlfile)

api = TwitterAPI(config['twitter']['api_key'],
                 config['twitter']['api_secret_key'],
                 auth_type='oAuth2'
                )

projectname = 'api_end'

DIRECTORY = 'local_data/'
filename = '{0}{1}.jsonl'.format(DIRECTORY, projectname)
FDAT_DIR = '{0}fdat/'.format(DIRECTORY)

token = pd.read_csv('token.csv')

df_users = pd.read_parquet('counted_users.parquet')
user_ids = df_users[df_users['id'] > 100].index.to_list()

def save_accounts_data(filename, account_names=[], account_ids=[]):
    account_count = 0
    if len(account_names) > 0:
        bags = [account_names[x:x+100] for x in range(0, len(account_names), 100)]
        for bag in tqdm(bags,
```

```
                            desc='Bags of names'):
            with open(filename, 'a', encoding='utf-8') as f:
                r = api.request('users/lookup', {'screen_name': str.join(',', bag), 'tweet
                for account in r:
                    account_count += 1
                    json.dump(account, f)
                    f.write('\n')
    if len(account_ids) > 0:
        account_ids = list(set(account_ids))
        bags = [account_ids[x:x+100] for x in range(0, len(account_ids), 100)]
        for bag in tqdm(bags,
                            desc='Bags of IDs'):
            with open(filename, 'a', encoding='utf-8') as f:
                r = api.request('users/lookup', {'user_id': str.join(',',map(str, bag)), '
                for account in r:
                    account_count += 1
                    json.dump(account, f)
                    f.write('\n')
    return(account_count)

def collect_friends(account_id, cursor = -1, over5000 = False):
    ids = []
    r = api.request('friends/ids', {'user_id': account_id, 'cursor': cursor})

    if 'errors' in r.json():
        if r.json()['errors'][0]['code'] == 34:
            return(ids)
        elif r.json()['errors'][0]['code'] == 326:
            pass
        elif r.json()['errors'][0]['code'] == 88:
            time.sleep(60*15)
        else:
            print (r.json()['errors'])

    for item in r:
        if isinstance(item, int):
            ids.append(item)
        elif 'message' in item:
            print ('{0} ({1})'.format(item['message'], item['code']))

    if over5000:
```

```
        if 'next_cursor' in r.json():
            if r.json()['next_cursor'] != 0:
                ids = ids + collect_friends(account_id, r.json()['next_cursor'], over5000

    return(ids)

def save_friends(user, ids):
    with open('{0}{1}.f'.format(FDAT_DIR, user), 'w', encoding='utf-8') as f:
        f.write(str.join('\n', (str(x) for x in ids)))

def collect_and_save_friends(user, refresh = False):
    if not refresh and os.path.exists('{0}{1}.f'.format(FDAT_DIR,user)):
        return('Already saved: {}'.format(user))
    else:
        friends = collect_friends(user)
        save_friends(user,friends)
        return('Friends saved: {}'.format(user))

def get_friends(friend_id):
    friends = []
    try:
        with open('{0}{1}.f'.format(FDAT_DIR, friend_id)) as f:
            for line in f:
                friends.append(int(line))
    except:
        pass
    return (friends)

def lazy_collect_from_ids(all_ids):
    '''
    collect followings of all accounts
    '''
    print(len(all_ids))
    all_ids = [uid for uid in all_ids if not os.path.exists(f'{FDAT_DIR}{uid}.f')]
    print(len(all_ids))

    bags = [all_ids[x:x+15] for x in range(0, len(all_ids), 15)]
    for bag in tqdm(bags,
                        desc='Bags of IDs'):
        for user_id in tqdm(bag,
                                desc = 'IDs in bag',
```

```
                                leave = False):
            try:
                collect_and_save_friends(user_id)
            except:
                print('Error while collecting')
                time.sleep(60*15)

lazy_collect_from_ids(user_ids)
```

C. Code Followwetzwerk

```
from tqdm.auto import tqdm
import networkx as nx
import time
import glob
import re
from msgspec.json import decode
from msgspec import Struct


def get_followings(following_id):
    global error_count
    followings = []
    try:
        with open(f"local_data/fdat/{following_id}.f") as f:
            for line in f:
                followings.append(int(line))
    except:
        error_count += 1
        pass
    return followings


def create_networkxgraph(account_ids, users):
    start_time = time.time()

    G = nx.DiGraph()

    nbar = tqdm(total=len(account_ids), desc="Nodes")
    for account in users:
        nbar.update(1)
        G.add_node(
            account["id"],
            Label=account["screen_name"],
            followers_count=account["followers_count"],
```

```
            friends_count=account["friends_count"],
            statuses_count=account["statuses_count"],
            favourites_count=account["favourites_count"],
            created_at=account["created_at"],
            created_at_year=account["created_at"][-4:],
            verified=account["verified"],
        )
    nbar.close()

    relevant_ids = set(account_ids)
    ebar = tqdm(total=len(account_ids), desc="Edges")
    for account in account_ids:
        ebar.update(1)
        local_edges = []
        followings = get_followings(account)
        relevant_followings = list(filter(lambda x: x in relevant_ids, followings))
        for following in relevant_followings:
            local_edges.append((following, account))
        G.add_edges_from(local_edges)
    ebar.close()

    print(f"Created graph with networkX in {(time.time() - start_time)} seconds")

    return G


class User(Struct):
    id: int
    created_at: str
    screen_name: str
    followers_count: int
    friends_count: int
    statuses_count: int
    favourites_count: int
    verified: bool


def load_users(files):
    users = []
    rel_users = set(relevant_ids)
```

```
    for file in tqdm(files):
        with open(file, "rb") as f:
            for line in tqdm(f, total=691465):
                user = decode(line, type=User)
                if user.id in rel_users:
                    users.append(
                        {
                            "id": user.id,
                            "created_at": user.created_at,
                            "screen_name": user.screen_name,
                            "followers_count": user.followers_count,
                            "friends_count": user.friends_count,
                            "statuses_count": user.statuses_count,
                            "favourites_count": user.favourites_count,
                            "verified": user.verified,
                        }
                    )
    return users


start_time = time.time()
relevant_ids = [
    int(re.search(r"\d+", userid)[0]) for userid in glob.glob("local_data/fdat/*.f")
]
print(f"Gefunden in {(time.time() - start_time)} Sekunden")

users = load_users(["userinfo.jsonl"])
print("Users loaded.")

error_count = 0
start_time = time.time()
networkxgraph = create_networkxgraph(relevant_ids, users)
print(f"Fehler mit {error_count} Accounts")

start_time = time.time()
nx.write_graphml(networkxgraph, "followNetzwerk.graphml")
print(f"Saved graph with networkX in {(time.time() - start_time)} seconds")
```

D. Code Interaktionsnetzwerk

```
# create interaction network

def create_edges(tweets_df, edges):
    # only if target tweet is available
    '''
    {"user_id": {
            "user1id": 1,
            "user5id": 1}
            }}
    '''

    columns = ['at_id','rt_id','qt_id']

    tweets_df.reset_index(inplace=True)
    tweets_df.set_index('id', inplace=True)
    for column in columns:
        print(f"{time.strftime('%Y-%m-%d %H:%M:%S')} Converting {column}")
        tweets_df[column] = tweets_df[column].map(tweets_df["user_id"])

    def add_edge(source, target):
        if source == target:
            return
        if source not in edges:
            edges[source] = {}
        if target not in edges[source]:
            edges[source] = {target: 1}
        else:
            edges[source][target] += 1

    for column in columns:
        print(f"{time.strftime('%Y-%m-%d %H:%M:%S')} Creating {column} edges")
        for row in tqdm(tweets_df[tweets_df[column].notna()][['user_id',column]].to_dict('
            add_edge(int(row['user_id']), int(row[column]))
```

```
    return (edges)

def create_networkxgraph(user_ids, users_df, edges, min_weight=0):
    start_time = time.time()

    G = nx.DiGraph()

    # Make info from follow network easier identifiable
    users_df.rename(columns={"Degree": "follow_degree",
                             "indegree": "follow_indegree",
                             "outdegree": "follow_outdegree",
                             "modularity_class" : "follow_modularity_class"},
                    inplace=True)



    local_users_df = users_df[users_df['id'].isin(user_ids)].copy()
    local_users_df['withheld_in_countries'] = local_users_df['withheld_in_countries'].appl
    nodes_info = local_users_df.fillna(0).to_dict('records')

    nodes = [(node_info['id'], node_info) for node_info in nodes_info]

    G.add_nodes_from(nodes)

    ebar = tqdm(total=len(edges), desc="Edges")
    for source, target in edges.items():
        ebar.update(1)
        local_edges = []
        for target_id, weight in target.items():
            if weight >= min_weight:
                local_edges.append((int(source), int(target_id), weight))
        G.add_weighted_edges_from(local_edges)
    ebar.close()

    print(f"Created graph with networkX in {(time.time() - start_time)} seconds")

    return G

df_users = pd.read_csv("follownetwork_user_info.csv")
user_info_df = pd.read_parquet("user_info.parquet")
tweets_df = pd.read_parquet('tweets.parquet')
```

```
user_ids = set(tweets_df.groupby('user_id').count().reset_index()['user_id'].to_list())

combined_edges = {}
edges = create_edges(tweets_df, combined_edges)
g = create_networkxgraph(user_ids, user_info_df.copy(), combined_edges)
nx.write_graphml(g, "interaktionsnetzwerk.graphml")
```

E. Code Auswertung

```
import pandas as pd

dfs = [pd.read_parquet("tweets-lang_de-April_2021.parquet"), pd.read_parquet("tweets-lang_
]

tweets_df = pd.concat(dfs)

f_communities = {18720: "f1",
                 71784: "f2",
                 11372: "f3",
                 74409: "f4",
                 51203: "f5",
                 10997: "f6",
                 35892: "f7",
                 40824: "f8",
                 62524: "f9",
                 55167: "f10"}

i_communities = {9713: 'i1', 8521: 'i2', 2901: 'i3', 9172: 'i4', 7816: 'i5', 9788: 'i6', 9


tweets_per_year = tweets_df[tweets_df['community'].isin(set(f_communities))].groupby(['yea

for i in range(0,3):
    print(i)
    users_in_community = len(dfs[i][~dfs[i]['community_all'].isna()].groupby('user_id')['i
    users_total = len(dfs[i].groupby('user_id')['id'].count())
    tweets_in_communities = len(dfs[i][~dfs[i]['community_all'].isna()]['id'])
    tweets_total = len(dfs[i])

    print(f"users_in_community: {users_in_community}")
    print(f"users_total: {users_total}")
    print(f"% in community: {users_in_community/users_total}")
    print(f"tweets_in_communities: {tweets_in_communities}")
```

```
    print(f"tweets_total: {tweets_total}")
    print(f"% tweets in communities: {tweets_in_communities/tweets_total}")

def scatter(df, year):
    df['Tweets'] = 1
    df_int = df.groupby('community_interaction')['Tweets'].count().to_frame()
    int_count = df.groupby(['community_interaction','user_id']).count()
    df_int['Accounts'] = df_int.index.map(lambda x: len(int_count.loc[x]))
    df_int['Tweets_je_Account'] = df_int['Tweets'] / df_int['Accounts']
    sns.scatterplot(
        df_int[(df_int['Accounts'] > 2) & (df_int['Tweets'] > 50000)].reset_index(),
        x=df_int[(df_int['Accounts'] > 2) & (df_int['Tweets'] > 50000)].reset_index().inde
        y="Tweets",
        size="Accounts",
        hue="community_interaction",
        palette = color_map,
        sizes=(500, 1000),
        #legend=None
    ).set(
        title=f"Twittervielfalt April {year}",
        ylim=(-50000, 2700000),#2700000)
        xlabel='Interaktionscommunity',
        #xticklabels=[]
    )
    plt.axhline(y=500000)
    #plt.savefig(f"twittervielfalt_april_{year}.svg")
    df_int[(df_int['Accounts'] > 2) & (df_int['Tweets'] > 5000)].to_csv(f'Twitteruntersuch

scatter(dfs[2], 2023)

tweets = pd.read_parquet('lang_de-tweets-2020to2023.parquet', columns=['created_at','user_

counted_month = tweets.groupby([pd.Grouper(freq='m'), 'community']).size().unstack()
```