



Telegram als Datenquelle für das Erfassen von Migrationsbewegungen und Informationsbedarfen von Migrant*innen

Untersuchung der technischen Möglichkeiten der Telegram-API für Datenanalysen

Hanno Skowronek
Dezember 2022

Fem.OS

Das Projekt „Fem.OS – Aufsuchendes Orientierungs- und Beratungs-System in den sozialen Medien für Migrantinnen“ wird gefördert von der Beauftragten der Bundesregierung für Migration, Flüchtlinge und Integration zugleich der Beauftragten der Bundesregierung für Antirassismus.



Die Beauftragte der Bundesregierung
für Migration, Flüchtlinge und Integration
Die Beauftragte der Bundesregierung für Antirassismus

In Kooperation mit



**Bundesagentur
für Arbeit**

Um zugewanderten und geflüchteten Frauen aus Drittstaaten die beste Unterstützung und Beratung anbieten zu können, ist es notwendig, deren Informationsbedarfe in Erfahrung zu bringen.

Telegram ist eine der meist-genutzten Messenger-Apps und sozialen Netzwerke weltweit und wird insbesondere unter Geflüchteten aus der Ukraine intensiv genutzt. Mithilfe der Telegram-API lassen sich Bedarfe aus öffentlichen Telegram-Gruppen extrahieren, um so gezielt Beratungsarbeit leisten zu können.

Dieses Working Paper geht auf die technischen Möglichkeiten der Telegram-API ein, um zu untersuchen, inwieweit sich Telegram als Datenquelle für das Erfassen von Migrationsbewegungen und Informationsbedarfen von zugewanderten Frauen aus Drittstaaten eignet.

Hierzu wurde ein Prototyp entwickelt, anhand dessen beispielhaft aufgezeigt wird, wie Daten durch die Telegram-API angefragt und verarbeitet werden können.

Mit den Erkenntnissen, die wir aus Telegram als Datenquelle gewinnen können, erhoffen wir Angebote für zugewanderte Frauen aus Drittstaaten in Deutschland mit Daten zu unterstützen,

Um effektiv Hilfe anbieten zu können und Menschen ein würdiges Leben in Deutschland zu ermöglichen, ist es notwendig diese Informationsbedarfe einschätzen und entsprechende Angebote zur Deckung dieser Bedarfe machen zu können.

Inhaltsverzeichnis

1. Einleitung	3
2. Hintergrund	4
2.1. Telegram-API	5
2.2. Telethon (Telethon 2022a)	5
2.2.1. Authentifizierung	6
2.2.2. Auslesen von Chats	6
2.2.3. Auslesen von Mitgliedern eines Chats	7
2.2.4. Auslesen von Nachrichten eines Chats	9
2.3. Nutzungsbedingungen von Telegram	10
2.4. Datenschutz und personenbezogene Daten	11
2.5. Einschränkungen der Telegram-API	12
2.6. Telegram-Gruppen und Administratoren	15
3. Prototyp: Telegram-Research-App	15
3.1. Architektur	16
3.2. Implementation	17
3.2.1. Benutzereingaben	17
3.2.2. Generierte Daten	19
4. Verwendung der Telegram-Research-App	24
4.1. Technische Voraussetzungen	24
4.2. Installation und Vorbereitung der Telegram-Research-App	24
4.3. Erste Ausführung	25
4.4. Langfristige Verwendung	26
4.4.1. Aktualisierung der Telegram-Research-App	27
5. Analyse und Auswertung der initialen Daten	27
5.1. Betrachtete Gruppen	27
5.2. Verwendete Benutzereingaben	27
5.3. Informationsbedarfe	27
5.3.1. Stichwort-Analyse von einzelnen Wörtern	27
5.3.2. Verwendete Sprachen	28
5.3.3. Wörter ohne eigenständigen Inhalt	28
5.3.4. Konjugation und Synonyme	31
5.3.5. Stichwort-Analyse von Wortpaaren	31
5.4. Aktivität der Gruppen	32
5.5. Gender-Verteilung	34
5.6. Migrationsbewegungen	37
5.6.1. Entwicklung von Mitgliederzahlen	37
5.6.2. Netzwerkanalyse Möglichkeiten	39
6. Fazit	41

Literaturverzeichnis	42
Abbildungsverzeichnis.....	43
Codeverzeichnis	44
Tabellenverzeichnis.....	44

1. Einleitung

In Europa findet aktuell die größte Fluchtbewegung seit dem Zweiten Weltkrieg statt (Brücker et al. 2022). Allein in Deutschland wurden z.B. seit Kriegsbeginn in der Ukraine über 1 Mio. Geflüchtete registriert. Bereits im Jahr 2021 war die Anzahl der Asylanträge bis November in Deutschland mit 189.998 auf einem hohen Niveau, vor allem von Menschen aus Syrien, Afghanistan und der Türkei. Im Vergleich zum Vorjahr über denselben Zeitraum bedeutet dies eine Zunahme von 43,2 % (BAMF - Bundesamt für Migration und Flüchtlinge 2022).

Menschen, die in Deutschland ankommen, haben diverse Informationsbedarfe und Themen, welche für sie von hoher Relevanz sind wie z. B. die Unterstützung für Menschen mit Kindern, Arbeitssuche, finanzielle Grundsicherung, Sprachkurse und Weiterbildung. Die Erfassung der konkreten Bedarfe zu diesen Themen unter der Zielgruppe ist eine wichtige Grundlage, um bedarfsgerechte Hilfe anbieten und zielgerichtete Angebote entwickeln zu können.

Auch wenn staatlich finanzierte und private Einrichtungen bereits viel Unterstützung und Beratung für Hilfesuchende anbieten, werden durch diese Anlaufstellen nicht alle Menschen erreicht (Tunjić et al. 2022). Darüber hinaus können Sprachbarrieren und generelle Unsicherheit in einem neuen Land dazu führen, dass nicht alle Informationsbedarfe und Anliegen zur Sprache kommen. Viele dieser Bedarfe werden dementsprechend niedrigschwellig in sozialen Netzwerken geäußert und beantwortet (Schulz-Kersting et al. 2022; Pfeffer-Hoffmann 2022; Brücker et al. 2022). Soziale Netzwerke bieten Zugewanderten die Möglichkeit, sich mit anderen Menschen über ihre Erfahrungen auszutauschen, Fragen zu stellen und voneinander zu profitieren. Darüber hinaus gibt es in sozialen Netzwerken Gruppen von Migrant*innen und Geflüchteten, die in den jeweiligen Muttersprachen kommunizieren, sodass Informationsbedarfe ohne Sprachbarriere kommuniziert und im besten Fall aufgelöst werden können.

Schätzungsweise nutzten weltweit im Jahr 2022 knapp 4,59 Milliarden Menschen soziale Netzwerke (Statista 2022), im Juni 2022 verzeichnete allein die Messenger-App Telegram weltweit 700 Millionen monatlich aktive Nutzer*innen (Telegram 2022a). Es ist dementsprechend anzunehmen, dass ein großer Teil der Geflüchteten in Deutschland soziale Netzwerke wie zum Beispiel Telegram für private Zwecke nutzt, um Fragen zu stellen, Erfahrungsberichte von anderen Menschen in ähnlichen Situationen zu hören und um sich über alle Aspekte des Lebens in Deutschland informieren (Pfeffer-Hoffmann 2022).

Vor diesem Hintergrund ist insbesondere die Arbeit mit Telegram interessant: Telegram bietet eine umfangreiche API (Application Programming Interface), die genutzt werden kann, um Software auf Basis der Telegram-App zu schreiben (Telegram 2022j). Diese kann beispielsweise verwendet werden, um Inhalte und diskutierte Themen in öffentlichen Gruppen auszulesen und zu verarbeiten. Somit kann mithilfe der Telegram-API Telegram effektiv als Datenquelle genutzt werden, um Informationsbedarfe, die innerhalb von Telegram-Gruppen zum Ausdruck gebracht werden, zu analysieren.

Diese Veröffentlichung zeigt auf, welche technischen Möglichkeiten und Einschränkungen es gibt, die Telegram-API zu nutzen. Darüber hinaus beschreibt dieser Text, welche Daten über die Telegram-API bezogen werden können und Ansätze wie sich diese Daten für die Analyse von

Migrationsbewegungen und Informationsbedarfe von Zugewanderten nutzen lassen. Um die Verwendbarkeit von Telegram als Datenquelle zu bewerten zu können, wurde prototypisch eine Applikation auf Basis der Telegram-API implementiert - im Folgenden als Telegram-Research-App bezeichnet. Die Erkenntnisse und Ergebnisse, die sich aus den ersten von der Telegram-Research-App erzeugten Daten gewinnen lassen, verdeutlichen die Möglichkeiten der Telegram-API.

Zunächst geht der Text auf das notwendige Hintergrundwissen für die Verwendung der Telegram-API ein und führt, anhand ein paar simpler Beispiele, in die Thematik ein. Im Anschluss wird die Architektur und Implementierung der Telegram-Research-App beispielhaft erklärt, sowie die Verwendung und Ausführung dieses Prototyps. Zuletzt gehen wir auf exemplarisch erfasste Daten ein und zeigen Datenanalysepotenziale, welche die Verwendung von Telegram als Datenquelle, ermöglicht. Die Kapitel *3 Prototyp: Telegram-Research-App* und *4 Verwendung der Telegram-Research-App* haben einen technischen Fokus und setzen teilweise geringe Programmierkenntnisse voraus. Der Abschnitt *5 Analyse und Auswertung der initialen Daten* hat hauptsächlich einen Fokus auf die Ergebnisse und Daten der Telegram-Research-App.

2. Hintergrund

Mit 700 Millionen monatlichen aktiven Nutzer*innen ist die Telegram-App eines der am meisten genutzten sozialen Netzwerke der Welt (Telegram 2022a). Die Telegram-App bietet die Möglichkeit, Nachrichten mit privaten Kontakten auszutauschen. Darüber hinaus können innerhalb der Telegram-App bis zu 200.000 Nutzer*innen eine Gruppe bilden. Mit einer Suchmaschine lassen sich schnell öffentliche Gruppen zu vielen verschiedenen Themen wie Dating, Finanzen, Musik oder Sport finden¹. Im Gegensatz zu Facebook oder Instagram besitzt Telegram keine zentrale Startseite. Daraus ergibt sich eine dezentrale Plattform mit vielen diversen und teilweise schwierig zu findenden öffentlichen, von Nutzenden initiierten Gruppen, die keiner konkreten Struktur oder Ordnung unterliegen. Der Beitritt zu Gruppen funktioniert über persönliche Einladung, öffentliche Links oder anderen Gruppen. Das Beitrittsmanagement wird durch die Administrator*innen der Gruppen individuell gehandhabt.

Der Quellcode von Telegram ist öffentlich (DrKLO 2022) ebenso wie die von Telegram bereitgestellt Telegram-API (Telegram 2022j). Die Telegram-API ist in zwei verschiedene APIs unterteilt, eine für die Entwicklung von Telegram-Bots und eine für die Entwicklung von Messenger-Applikationen. Auf Basis der Telegram-API wurden viele weitere Software-Bibliotheken und Packages in verschiedenen Programmiersprachen von Dritten veröffentlicht².

¹ Siehe beispielsweise: <https://bestoftelegram.com/>, <https://telegramchannels.me/groups/>, <https://telegramguide.com/telegram-group-link/> (zuletzt besucht am 30.12.2022)

² Software-Bibliotheken und Packages sind Sammlungen von Software, die bereits Bausteine und Lösungswege für thematisch zusammengehörende Anwendungsfälle anbieten, wie in diesem Fall die Verwendung der Telegram-API. Um nur ein paar dieser Bibliotheken für die Verwendung der Telegram-API zu nennen: TDLib (TDLib 2022), Telethon (Telethon 2022b), python-telegram-bot (python-telegram-bot 2022), Java Telegram Bot API (Parshin 2022) und @mtproto/core (Gasymov 2022). Viele dieser Projekte werden sehr regelmäßig auf dem aktuellen Stand gehalten und haben eine Community von aktiven Nutzer*innen, die die Entwicklung dieser Projekte weiter vorantreibt.

Für die Verwendung der Telegram-API ist es notwendig, ausreichend Programmierkenntnisse zu haben. Der Prototyp dieser Veröffentlichung wurde in der Programmiersprache Python geschrieben, aber grundsätzlich lässt sich die Telegram-API mit vielen Programmiersprachen verwenden. Die Software-Bibliotheken helfen bei der Verwendung der Telegram-API, die alleinstehend aufwändig und anspruchsvoll zu verwenden ist.

In der Entwicklung der Telegram-Research-App wurde ausschließlich die *Telethon*-Bibliothek genutzt. Dementsprechend gehen wir im Folgenden nur auf die Telegram-API und die Verwendung von *Telethon* ein, thematisieren einige Einschränkungen der Telegram-API und betrachten notwendige Datenschutz-Richtlinien, die für die Entwicklung einer Applikation mit der Telegram-API wie der Telegram-Research-App sowie den daraus entstehenden Daten wichtig sind.

2.1. *Telegram-API*

Die Telegram-API bietet eine Vielzahl an verschiedenen Methoden, um mit Telegram zu interagieren, darunter Authentifizierung, das Versenden und Empfangen von Nachrichten, das Auslesen von Informationen über Chats, in denen sich der/die Nutzer*in befindet, und vieles mehr (Telegram 2022h). Um die Telegram-API zu verwenden, wird in erster Linie nur eine Handynummer und ein Gerät für die Ausführung benötigt. Grundsätzlich bietet sich für die Ausführung ein Computer oder Server an, aber auf Basis der Telegram-API können auch Applikationen für ein Smartphone oder andere mobile Geräte geschrieben werden. Pro Handynummer kann eine Applikation erstellt werden, für die der/die Nutzer*in einen *api_hash* und eine *api_id* erhält (Telegram 2022c). Mit Hilfe des *api_hash* und der *api_id* kann ein*e Nutzer*in sich authentifizieren (Telegram 2022f) und auf die ausführliche Bibliothek von Methoden zugreifen.

Die Telegram-API bietet Dokumentation über alle existierenden Methoden (Telegram 2022h) und darüber hinaus auch eine Anleitung (Telegram 2022g).

2.2. *Telethon*

Telethon (Telethon 2022a) ist eine Python-Bibliothek, die auf der Telegram-API basiert und das Nutzen dieser in Python vereinfachen soll. In *Telethon* sind alle Methoden der Telegram-API in Funktionen der *Client*-Klasse verarbeitet. Dadurch ist der/die Nutzer*in nicht darauf angewiesen, direkte Anfragen an die Telegram-API selbst zu formulieren. *Telethon* nutzt die *asyncio*-Bibliothek (Python 2022a) um asynchrone Implementierung und Anfragen zu erlauben. Vereinfacht bedeutet das, dass auf Funktionen, die Anfragen der Telegram-API nutzen, mithilfe von *await* gewartet werden muss, bevor die Ergebnisse der Funktion verwendet werden können.

Im Folgenden beschreiben wir beispielhaft ein paar ausgewählte Anwendungsfälle, die für die Telegram-Research-App und die Nutzung von Telegram als Datenquelle relevant sind. Grundsätzlich ist zu beachten, dass dies nur eine beschränkte Auswahl der Funktionalitäten von *Telethon* darstellt. Um mehr über die einzelnen Funktionen und die verschiedenen Datentypen zu erfahren, empfiehlt es sich, sowohl die Dokumentation von *Telethon* (Telethon 2022b) als auch die Dokumentation der Telegram-API (Telegram 2022j) zurate zu ziehen.

2.2.1. Authentifizierung

Die Authentifizierung und die Anmeldung kann mit *Telethon* mithilfe der Methode *start()* abgeschlossen werden.

```
from telethon import TelegramClient

api_id = 12345678
api_hash = 'fcbde8d1f296b4633c65f3f63836e2a4'
client = TelegramClient('anonymous', api_id, api_hash)
await client.start()
```

Code 1: Authentifizierung und Anmeldung mit *Telethon*

Die Methode *client.start()* ist eine interaktive Funktion und übernimmt die Authentifizierung, und leitet den/die Nutzer*in durch den Login-Prozess, falls es sich um einen neuen *TelegramClient* handelt, für den noch keine *.session*-Datenbank angelegt wurde. Diese Datenbank wird automatisch von *Telethon* angelegt, um bei erneuter Anmeldung den Prozess zu vereinfachen.

2.2.2. Auslesen von Chats

Telethon kann mithilfe einer Funktion die gesamte Liste aller Chats, Channels und Gruppen, in denen ein Telegram-Konto Mitglied ist, beziehungsweise denen es angehört, angefragt werden. Diese Liste besteht aus *Dialog*-Objekten. Ein *Dialog*-Objekt ist die Repräsentation eines beliebigen Chats, Channel oder Gruppe innerhalb von *Telethon*. In einem *Dialog*-Objekt sind unter anderem Informationen über den Namen der Gruppe gespeichert und wie viele Mitglieder sie hat, aber auch Account-abhängige Daten wie die Anzahl an ungelesenen Nachrichten und die letzte Nachricht, die innerhalb des Chats verschickt wurde.

```
dialogs = await client.get_dialogs()
```

Code 2: Dialog Anfrage - Anfrage einer Liste mit allen Chats, Channels und Gruppen für einen spezifischen Account.


```

{
  "date": "2022-11-26 10:37:36+00:00",
  "entity":
  {
    "date": "2021-01-23 16:21:55+00:00",
    "participants_count": 4133,
    "...": "..."
  },
  "id": -1001108795430,
  "is_channel": false,
  "is_group": true,
  "is_user": false,
  "message": "...",
  "name": "Beispiel Gruppe",
  "unread_count": 144,
  "...": "..."
}

```

Code 3: Dialog-Objekt Beispiel - Beispielhaftes Dialog-Objekt als eines der Elemente der Antwort auf eine `get_dialogs()`-Anfrage.

Beispielhaft zeigt Code 3, wie ein *Dialog*-Objekt aussehen könnte und welche Daten dieses enthält, z.B. das Datum der letzten Aktivität sowie den Zeitpunkt am 23.01.2021, an dem diese Telegram-Gruppe erstellt wurde. Des Weiteren ist die Anzahl der Teilnehmer*innen des Chats vermerkt. Darüber hinaus besitzt jeder *Dialog* einen einzigartigen Identifikator *id* und Informationen darüber, ob es sich um einen Channel, Gruppe oder einen Chat mit einer anderen Nutzer*in handelt. In diesem Beispiel wird eine Gruppe abgebildet. Außerdem werden die letzte Nachricht (hier gekürzt), der Name und die Anzahl an ungelesenen Nachrichten angezeigt.

2.2.3. Auslesen von Mitgliedern eines Chats

Wenn über die Anzahl der Teilnehmenden hinaus weitere Informationen über die Teilnehmenden gebraucht werden, kann mithilfe der `get_participants()`-Funktion eine vollständige Liste aller Teilnehmenden eines Chats angefragt werden.

```

participants = await client.get_participants(dialog)

```

Code 4: Beispielhafte Mitglieder-Anfrage - Anfrage einer Liste mit allen Mitgliedern eines Chats.

```

{
  "bot": false,
  "fake": false,
  "first_name": "Alice",
  "id": 188951969,
  "last_name": "Bob",
  "participant":
  {
    "date": "2021-01-23 16:23:55+00:00",
    "inviter_id": 795939944,
    "...": "..."
  },
  "phone": null,
  "photo": "...",
  "scam": false,
  "status": "userStatusRecently",
  "username": "alice",
  "...": "..."
}

```

Code 5: Participant-Objekt Beispiel - Beispielhaftes *Participant*-Objekt als eines der Elemente der Antwort auf eine *get_participants(dialog)*-Anfrage.

Über einzelne Mitglieder sind im *Participant*-Objekt einige Informationen zu finden, beispielhaft hierfür Code 5. Jedes *Participant*-Objekt ist die Repräsentation eines Mitglieds in einer Gruppe. Neben den freiwillig angegebenen und gewählten Vor- und Nachnamen, Username und Handynummer, gibt es auch Attribute, die beschreiben, ob der/die Nutzer*in ein Telegram-Bot ist oder als *fake* oder *scam* von anderen Nutzer*innen gemeldet wurde. Auch das Profilbild des/der Nutzer*in ist verzeichnet und kann über eine weitere Anfrage heruntergeladen werden.

Die Eigenschaften *fake* und *scam* werden Nutzer*innen von Telegram zugewiesen, wenn diese von vielen anderen Nutzer*innen gemeldet wurden. Grundsätzlich bedeutet dies eine Warnung von Telegram, wenn man mit diesen Nutzer*innen interagiert, da bereits andere Nutzer*innen negative Erfahrungen mit ihnen gemacht haben. Die genaue Richtlinie, wie Telegram diese Eigenschaften vergibt und wie Telegram mit diesen Nutzer*innen umgeht, ist nicht öffentlich bekannt. Telegram-Bots sind Mitglieder in Gruppen, die in der Regel für die Moderation genutzt werden. Sie können von den Admins der Gruppen nach eigenem Ermessen implementiert und hinzugefügt werden. Beispielsweise bieten einige öffentlich verfügbaren Telegram-Bots Funktionalität wie Spam-Schutz durch automatische *timeouts* oder *bans*. Telegram-Bots sind für die Analyse von Informationsbedarfen nicht interessant und können dementsprechend ignoriert werden.

Der Online-Status des Mitglieds gibt Auskunft darüber, wann ein Mitglied zuletzt online war: *recently*, *last week*, *last month*. Das *Participant*-Objekt enthält außerdem noch Informationen darüber, wann ein Mitglied in eine Gruppe beigetreten ist und auch wie, beziehungsweise von wem, dieses Mitglied eingeladen wurde. Das Attribut *id* ist genau wie für die *Dialog*-Objekte ein einzigartiger Identifikator für ein Mitglied.

2.2.4. Auslesen von Nachrichten eines Chats

Um Nachrichten eines spezifischen Chats auszulesen, gibt es ein paar Parameter und Einschränkungen zu beachten. Die Nachrichten eines Chats in einer Liste zu erhalten ist möglich mit der Anfrage `TelegramClient.get_messages(dialog)`.

```
messages = await client.get_messages(dialog, limit=10)
```

Code 6: Beispielhafte Nachrichten-Anfrage - Anfrage einer Liste mit den 10 letzten Nachrichten eines Chats.

In Code 6 ist ein Ausschnitt der Parameter zu sehen, mit denen die Anfrage spezifiziert werden kann. Grundsätzlich umfasst die Antwort der Anfrage sowohl die Anzahl aller Nachrichten, die je in dem Chat verschickt wurden, und eine Liste aller angefragt Nachrichten. In diesem Beispiel würde diese Liste aufgrund des Parameters `limit=10` die zehn letzten Nachrichten enthalten. Bei Nachrichten kann es sich sowohl um, von Mitgliedern verschickte Nachrichten handeln, als auch `MessageService`-Objekte. Diese werden beispielsweise verschickt, wenn ein Mitglied der Gruppe beitrifft oder sie verlässt, wenn die Beschreibung des Chats angepasst wird, wenn ein Anruf gestartet wird oder auch wenn Punkte für ein Spiel innerhalb des Chats vergeben werden.

Grundsätzlich kann ein*e Nutzer*in mithilfe der Parameter sehr spezifisch wählen, welche Nachrichten angefragt werden. Mit den Parametern `min_id` und `max_id` kann eine Spanne von Nachrichten auf Basis ihrer im Chat einzigartigen und aufsteigend sortierten `id` ausgewählt werden. Der Parameter `offset_date` erlaubt die Anfrage von Nachrichten vor einem spezifischen Datum und mit dem Parameter `reverse` kombiniert, können auch Fragen nach einem spezifischen `offset_date` angefragt werden. Dasselbe ist auch mit dem Parameter `offset_id` für Identifikatoren möglich. Wenn der Parameter `limit=None` gesetzt wird, werden sogar alle verschickten (und ungelöschten) Nachrichten angefragt.

```
import datetime
from telethon.types import InputMessagesFilterGif

date = datetime.datetime(2022, 12, 31, 23, 59, 59)
user = participants[0]

messages_gifs_before_year_end = await client.get_messages(dialog, limit=5,
    offset_date=date, filter=InputMessagesFilterGif)

messages_from_user = await client.get_messages(dialog, limit=14,
    offset_date=date, from_user=user)
```

Code 7: Beispielhafte komplexe Anfragen - Anfrage einer Liste mit den fünf letzten GIFs vor Ende des Jahres 2022 eines Chats und Anfrage der letzten 14 Nachrichten eines spezifischen Mitglieds im Jahr 2022.

Mit den hier beschriebenen Parametern hat der/die Nutzer*in sehr differenzierte Möglichkeiten Nachrichten anzufragen. Weitere Parameter, die über die hier im Beispiel gezeigten Parametern hinaus gehen, enthalten Möglichkeiten nach Inhalt zu suchen, nach Nachrichtentyp zu filtern

(beispielsweise nur Sprachnachrichten, nur Bilder oder nur Hyperlinks) oder nur Nachrichten von einem bestimmten Mitglied anzufragen. Code 7 zeigt ein etwas ausführlicheres Beispiel, bei dem sowohl die fünf letzten GIFs des Jahres 2022 als auch die letzten vierzehn Nachrichten eines bestimmten Mitglieds im Jahr 2022 angefragt werden.

```
{
  "date": "2022-12-14 14:35:17+00:00",
  "from_id":
  {
    "user_id": 136820589,
    "...": "..."
  },
  "id": 51374,
  "is_reply": true,
  "media": null,
  "mentioned": false,
  "message": "Wow, mit Telethon geht ja viel.",
  "reactions": null,
  "...": "..."
}
```

Code 8: Message-Objekt Beispiel - Beispielhaftes *Message*-Objekt als eines der Elemente der Antwort auf eine *TelegramClient.get_messages(dialog)*-Anfrage.

Code 8 zeigt einen Ausschnitt der Attribute, die ein *Message*-Objekt aufweist. Neben den zu erwartenden Attributen wie dem Zeitpunkt der Nachricht, von wem die Nachricht verschickt wurde und dem Text der Nachricht, enthält das Objekt außerdem Informationen über mögliche Media-Inhalte wie Bilder oder Sprachnachrichten. Darüber hinaus ist in der Nachricht auf vermerkt, ob der/die Nutzer*in erwähnt wurde, ob und welche Reaktionen es auf die Nachricht bisher gab und auch ob es sich bei der Nachricht um eine Antwort handelt sowie den Identifikator der beantworteten Nachricht.

2.3. Nutzungsbedingungen von Telegram

Die Nutzungsbedingungen von Telegram (Telegram 2022e) und insbesondere der Telegram-API (Telegram 2022i) sowie die Datenschutzrichtlinie (Telegram 2022d) sind sehr deutlich in der Aussage, dass sie Spam und jegliche Form von Missbrauch von Telegram und der Telegram-API ahnden und die Accounts von Nutzer*innen bannen, die solche Verhaltensweisen auf der Messenger-Plattform aufweisen. Um zu gewährleisten, dass Nutzer*innen mit unlauteren Absichten sich nicht auf die Maßnahmen, die von Telegram genutzt werden, einstellen können, hält Telegram diese Maßnahmen zur Abwehr von Spam und anderem Missbrauchsverhalten geheim und optimiert und ändert sie regelmäßig. Für das Funktionieren einer Telegram-Forschungs-App ist es essenziell, dass diese Nutzungsbedingungen von Telegram vollständig erfüllt werden. Wenn eine Applikation die Nutzungsbedingungen verletzt, wird der zugehörige Telegram-Account zeitweise oder permanent gebannt. Daher ist es notwendig Anfragen zeitlich zu verteilen und zwischen Anfragen künstliche Pausen einzubauen, um nicht zu viele Anfragen zu schicken und als Spam eingestuft zu werden und um die Telegram-API nicht unnötig zu belasten. Außerdem sollte nach Möglichkeit die Anzahl an

Anfragen so gering wie möglich gehalten werden und selbstverständlich sollten unnötige doppelte Anfragen, die denselben Inhalt erfragen, vermieden werden. Darüber hinaus drosselt sowohl die Telegram-API als auch die *Teleton*-Bibliothek die Anzahl der verschickten Anfragen bei zu hoher Frequenz.

Aus diesen Nutzungsbedingungen ergibt sich eine Begrenzung über die Anzahl der Gruppen, Nachrichten und die verfügbaren Daten, die mithilfe der Telegram-API analysiert werden können.

2.4. *Datenschutz und personenbezogene Daten*

Bei den in Telegram verschickten Nachrichten und Informationen über Nutzer*innen handelt es sich in vielen Fällen potenziell um personenbezogene Daten. Freiwillige Angaben von Nutzer*innen wie beispielsweise Telefonnummer und Vor- und Nachname sind personenbezogene Daten, die bei der Verarbeitung mit besonderer Vorsicht behandelt werden müssen, um mit der Datenschutz-Grundverordnung (DSGVO) konform zu sein. Weitergehend, sind auch Parameter, die potenziell eindeutig auf Nutzer*innen zurückführen können, als personenbezogene Daten zu beachten, beispielsweise der einzigartige Identifikator einer Nachricht und von welchem/welcher Nutzer*in diese Nachricht verschickt wurde. Sogar die Nachricht selbst ist potenziell einzigartig, sodass von der Nachricht auf den/die Nutzer*in geschlossen werden kann, insbesondere wenn die Gruppe, in welcher die Nachricht geschickt wurde, bekannt ist. Obwohl es sich hierbei um personenbezogene Daten handelt, stellt Telegram den Zugang zu diesen Daten über die API zur Verfügung. Hierauf haben externe Entwickler*innen keinen Einfluss. Es ist allerdings möglich, die abgerufenen Daten so einzuschränken, dass die Daten vor einer weiteren Analyse anonymisiert werden.


Um die Konformität einer Telegram-Forschungs-App mit den rechtlichen Vorgaben der Datenschutzgrundverordnung zu gewährleisten, werden in der Telegram-Research-App keine personenbezogenen Daten gespeichert. Darüber hinaus empfiehlt es sich, nur die für Analyse relevanten Daten zu nutzen und beispielsweise Telefonnummer und Nachname zu verwerfen, falls diese von Nutzer*innen angegeben wurden. Wenn für weitergehende Analyse Daten gespeichert werden müssen, muss darauf geachtet werden, dass potenziell personenbezogene Daten so abgewandelt werden, dass sie nicht mehr als personenbezogene Daten verwendet werden können, wie beispielsweise durch Einweg-Hashing³.

³ Hashing beschreibt die Transformation eines Datensatzes mithilfe einer Funktion. Dadurch können Verknüpfungen zwischen Datenpunkte aufrechterhalten und gleichzeitig die Anonymität gewährleistet werden. Einweg-Hashing bedeutet, dass derselbe Datensatz unwiderruflich transformiert wird und dass die Transformation des Datensatzes nicht eindeutig sein muss. Es kann also mit demselben Datensatz und derselben Einweg-Hashing-Methode zu unterschiedlichen transformierten Datensätzen kommen, die alle weiterhin die korrekten Verknüpfungen und Beziehungen zwischen Datenpunkten erfüllen. Tabelle 1 stellt dies anhand eines Beispiels eines Netzwerk-Datensatzes mit Identifikatoren und Gruppenzugehörigkeit dar.

Tabelle 1: Beispiel einer Einweg-Hashing-Transformation

In diesem simplen Beispiel wird die personenbezogenen Mitglied-Id anonymisiert. Die Tabelle zeigt in welchen Telegram-Gruppen sich Mitglieder befinden. Die Verknüpfungen werden korrekt aufrechterhalten, sodass auch in der anonymisierten Tabelle weiterhin erkennbar ist, wie die Gruppen miteinander vernetzt sind.

Mitglied-Id	Telegram-Gruppe-Id
48371475	-1001589772550
19284782	-1001589772550
90918233	-1004493818492
19284782	-1004493818492
90918233	-1001029403871
48371475	-1003192849813



Hashed Mitglied-Id	Telegram-Gruppe-Id
1	-1001589772550
2	-1001589772550
3	-1004493818492
2	-1004493818492
3	-1001029403871
1	-1003192849813

2.5. Einschränkungen der Telegram-API

Die Telegram-API hat weitreichendes Potenzial um als Datenquelle für die Analyse von transnationalen Migrations- und Fluchtbewegungen und die damit einhergehenden Informationsbedarfe genutzt zu werden. Dennoch gibt es Einschränkungen, die dem Schutz der Privatsphäre und der Person der Teilnehmer*innen dienen. Manche Einschränkungen lassen sich mithilfe von Approximationen ausgleichen. Grundsätzlich ist die Telegram-API aber als Möglichkeit für externe Entwickler*innen geeignet, eigene Messenger-Applikationen auf Basis von Telegram zu entwickeln. Das Nutzen von Telegram als Datenquelle ist in der Telegram-API nicht in größerem Ausmaß vorhergesehen.

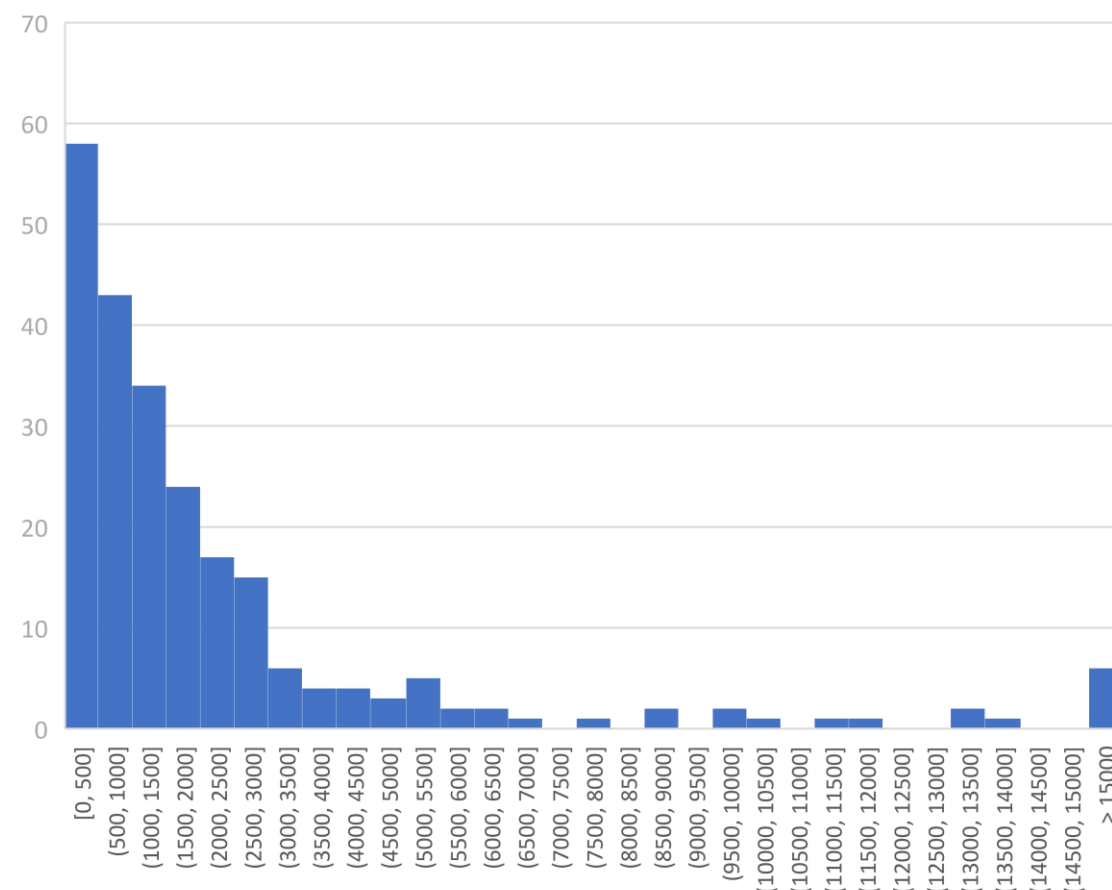


Abbildung 1: Histogramm der Größe von Gruppen

Insgesamt sind die Größen von 235 Telegram-Gruppen in diesem Histogramm abgebildet. Quelle: Telegram © Minor

Wie bereits aus Kapitel 2.4 hervorgeht, ist eine der größten Einschränkungen die Anzahl der Datenabfragen, die mithilfe der Telegram-API, innerhalb eines Zeitfenster verschickt werden dürfen. Darüber hinaus ist auch die Geschwindigkeit, mit der Nachrichten von größeren und aktiven (Anzahl an Nachrichten pro Woche) Gruppen, angefragt werden können, ein Hindernis. Datenabfragen zu größeren Gruppen mit vielen Nachrichten können teilweise länger als eine Stunde dauern. Bereits bei einer Gruppengröße von etwa 40000 Mitgliedern, brauchte eine Anfrage über die Nachrichten aus dem letzten Jahr etwa 70 Minuten. In den Abbildung 1 und Abbildung 2 sind jeweils die Größen der Telegram-Gruppen, die hier untersucht wurden, und die Anzahl der Nachrichten über eine Woche in einem Histogramm dargestellt. Außerdem sind in Tabelle 2 die acht größten ukrainischsprachigen Gruppen abgebildet und deren Anzahl an Nachrichten über eine Woche.

Für die Nutzung der Telegram-API als Datenquelle bedeutet dies, dass effektiv nur eine begrenzte Menge an Daten angefragt und verarbeitet werden kann. Das Ausmaß der Analyse ist also beschränkt.

Tabelle 2: Die acht größten ukrainischsprachigen Telegram-Gruppen nach Mitgliedern

Die größten Gruppen können potenziell jede Woche wechseln, ebenso die Anzahl verschickter Nachrichten. Dies ist also nur eine Momentaufnahme. Quelle: Telegram

Mitgliederanzahl	Nachrichten
38.559	2.189
24.779	1.663
21.811	3.997
20.796	1.366
20.708	2.247
20.216	1.955
13.703	4.851
13.362	624

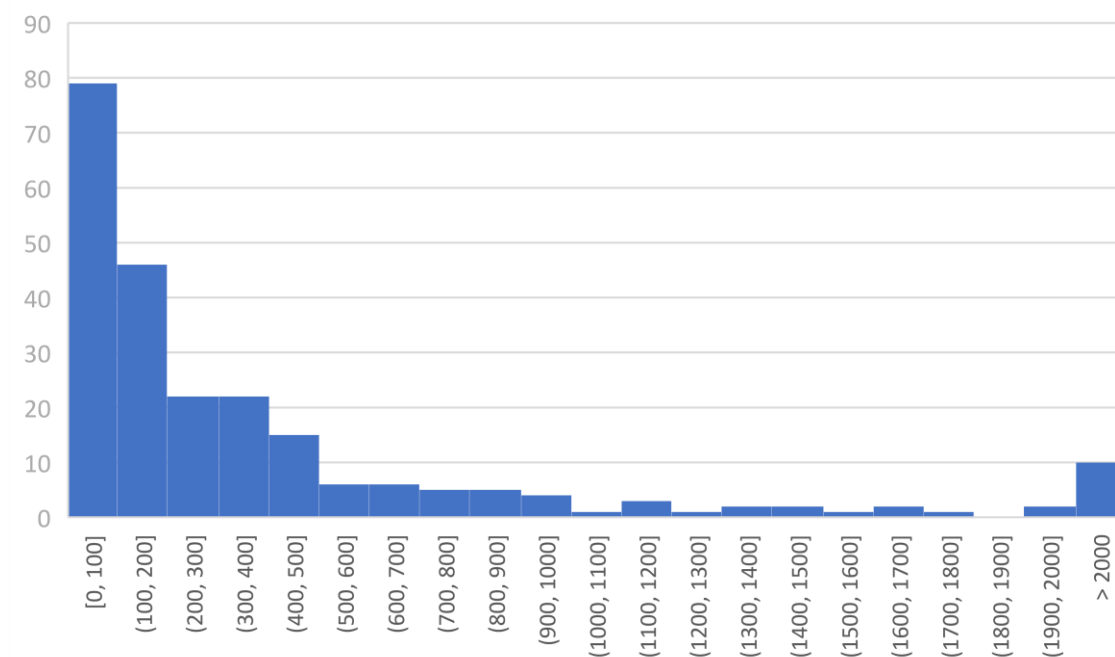


Abbildung 2: Histogramm der Anzahl an verschickten Nachrichten in einer Woche

Insgesamt ist die Nachrichtenanzahl von 235 Telegram-Gruppen über eine Woche in diesem Histogramm abgebildet. Quelle: Telegram © Minor

Die Telegram-API erlaubt keine rückwirkenden Informationen über Mitglieder von Gruppen. Das heißt retrospektiv lässt sich beispielsweise nicht sagen, wie sich die Größe von Gruppen entwickelt hat und damit auch nicht, ob und wie sich das Interesse an einer Gruppe geändert hat. Um diese Einschränkung zu umgehen, beziehungsweise um Informationen über spezifische Bedarfe auch retrospektiv zu erhalten, gibt es zwei Möglichkeiten. Sobald die Applikation über einen gewissen Zeitraum Daten analysiert und diese Ergebnisse speichert, existieren die Daten retrospektiv in der eigenen Datenbank. Wenn es darum geht in Zukunft neue Migrationsbewegungen zu analysieren, werden neu entstehende Gruppen früh in ihrer Entstehungsphase der Analyse hinzugefügt werden und somit wird eine fast lückenlose Analyse möglich sein.

Darüber hinaus wurde in der Telegram-Research-App eine Approximation des Wachstums der Gruppen implementiert, welche in vielen Fällen auch retrospektiv funktioniert. Mithilfe der Benachrichtigung, die versandt wird, immer wenn ein Mitglied der Gruppe beitrifft, ist es möglich aus den Nachrichten zu schließen, wie sich das Wachstum der Gruppe verhält.

2.6. Telegram-Gruppen und Administratoren

Telegram-Gruppen sind öffentliche Räume, die von und für Nutzer*innen geschaffen werden, um gemeinsam in den Austausch zu kommen. Administratoren sind häufig selbst aktiver Teil der Gruppe und haben dementsprechend ein Interesse daran, dass die Gruppen geordnet und nach den gemeinschaftlichen Vorstellungen funktionieren. Dementsprechend ist es bei der Erfassung von Daten ungemein wichtig, auf die Bedürfnisse und Richtlinien der Gruppen zu achten, sodass nicht unbeabsichtigterweise Bedürfnisse der Telegram-Gruppen übergangen werden.

Einige Administratoren von Telegram-Gruppen schränken den Zugriff, den Mitglieder auf die Daten anderer Mitglieder haben, ein, um die Privatsphäre ihrer Mitglieder auch in einem offenen Forum zu schützen. Unter den zehn größten ukrainischen Gruppen befinden sich beispielsweise drei, die keine Informationen über die einzelnen Mitglieder veröffentlichen.

Häufig werden Telegram-Bots zum Einsatz gebracht, um bei der Moderation zu unterstützen. Es kann dementsprechend sein, dass das Verhalten einer Applikation wie der Telegram-Research-App zu Komplikationen, Timeouts oder sogar Bans führt. Im Falle eines automatischen Timeouts oder Bans, sollten umgehend die Administratoren der Telegram-Gruppe kontaktiert werden, damit es nicht zu Missverständnissen über die Beweggründe der Datenerfassung kommt.

3. Prototyp: Telegram-Research-App

Die von uns entwickelte Telegram-Research-App ist eine Python-Applikation auf Basis der *Telethon*-Bibliothek (Telethon 2022a), welche auf der Telegram-API (Telegram 2022j) basiert. Ziel der Implementation war es, sich mit der Verwendung der Telegram-API vertraut zu machen und zu überprüfen, inwieweit sich Telegram als Datenquelle für die Analyse von Migrationsbewegung und Informationsbedarfen von Migrant*innen eignet.

Es hat sich gezeigt, dass es mit Hilfe einer Applikation wie der Telegram-Research-App möglich ist automatisiert Informationen über Telegram-Gruppen wie beispielsweise Anzahl der Mitglieder, Aktivität über den Tag und über die Woche verteilt, Anzahl an einzigartigen Mitgliedern über eine Menge von Gruppen sowie eine approximierte Gender-Verteilung auf Basis des gewählten Vornamens zu generieren. Darüber hinaus ist es möglich, auf Basis von Schlagwortlisten zu zählen, welche Wörter in Gruppen am häufigsten verwendet werden und/oder zu betrachten welche Wortpaare am häufigsten vorkommen. Zudem können die Daten aller Gruppen einer Sprache aggregiert und als eine Gesamtübersicht einer Sprachgruppe bereitgestellt werden. Neben all diesen Statistiken wird zudem eine anonymisierte Liste aller Mitglieder aller Gruppen einer Sprachgruppe erstellt, die weitere Netzwerk-Analysen ermöglicht.

3.1. Architektur

Der von uns entwickelte Prototyp Telegram-Research-App verwendet dazu eine recht simple Architektur aus insgesamt sechs relevanten Python-Skripts. Diese Architektur ist eine beispielhafte Repräsentation für die Implementation einer Applikation auf Basis der *Telethon*-Bibliothek. Der Einstiegspunkt der Applikation findet sich in der *main.py*-Datei.

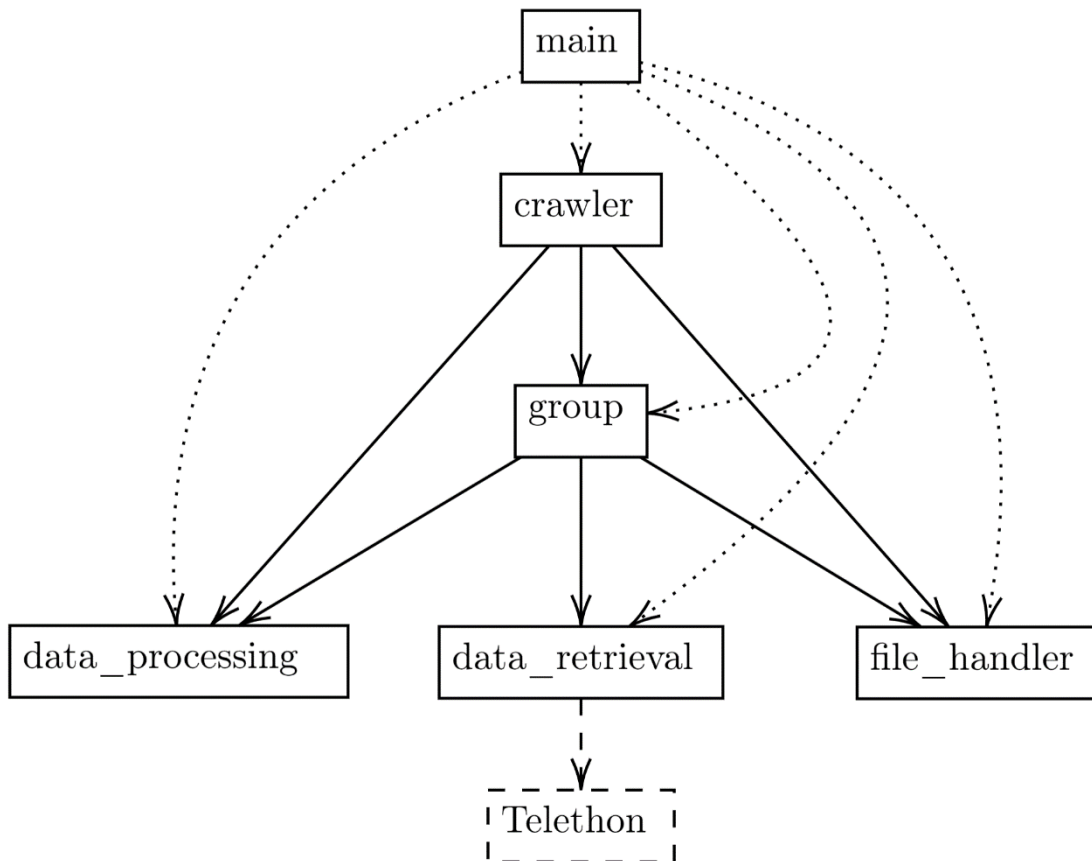


Abbildung 3: Vereinfachte Architektur der Telegram-Research-App

Pfeile repräsentieren eine Verwendungshierarchie, beispielsweise verwendet das Skript *crawler* das Skript *file_handler*. Dateiendungen wurden für bessere Lesbarkeit weggelassen. Manche der Pfeile sind gestrichelt für bessere Lesbarkeit. Die *Telethon*-Bibliothek ist zusätzlich ergänzt und ist nicht von uns geschrieben (Telethon 2022b).

In Abbildung 3 ist die Architektur der Telegram-Research-App verbildlicht. Die Applikation wird aus der *main.py* gestartet, welche die Applikation initialisiert und ausführt. Jede Sprache wird durch eine Instanz der *Crawler*-Klasse repräsentiert, welche in *crawler.py* definiert wird. Jede Gruppe wird repräsentiert durch eine Instanz der *Group*-Klasse, welche in *group.py* definiert wird. Alle Gruppen werden in der Initialisierung ihren Sprachen zugeordnet, sodass Sprachen-spezifische Statistiken erstellt werden können. Darüber hinaus gibt es den *file_handler.py*, für das Schreiben und Auslesen von lokalen Dateien, das *data_processing.py* für die Verarbeitung der Daten und das *data_retrieval.py* für das Senden und Empfangen von Anfragen an die Telegram-API über die *Telethon*-Bibliothek.

main.py

Die Datei *main.py* enthält den Einstiegspunkt der Applikation. Alle Accounts werden angemeldet, die *Crawler*-Instanzen der verschiedenen Sprachen werden für alle Sprachen und Gruppen initialisiert und ausgeführt und zum Schluss werden alle Accounts wieder abgemeldet. Darüber hinaus sind in der *main.py*-Datei die Verwaltung der Kommandozeile-Argumente zu finden.

group.py

Die Datei *group.py* enthält die Implementierung der *Group*-Klasse. Jede Telegram-Gruppe wird innerhalb von der Telegram-Research-App durch eine *Group*-Instanz repräsentiert. Neben dem Identifikator, Namen und anderen Attributen der Gruppe, enthält diese Klasse die Methode *run(self, week_datetime, is_present, words_to_ignore, words_of_interest)*, mit der die Analyse für die jeweilige Gruppe ausgeführt werden kann. Dabei muss über die Parameter festgelegt werden, welche Woche analysiert werden soll, ob es sich dabei um die zuletzt vergangene Woche handelt, welche Wörter bei der Schlagwort-Analyse ignoriert werden sollen und welche Wörter von besonderem Interesse für die Analyse sind. Die Ergebnisse der Analyse werden in *.csv*-Dateien auf der Festplatte gespeichert, sodass die *Group*-Objekte nicht zu groß werden und den Arbeitsspeicher unnötigerweise belegen.

crawler.py

Die Datei *crawler.py* enthält die Implementierung der *Crawler*-Klasse. Innerhalb jeder Instanz der *Crawler*-Klasse werden alle Gruppen (*Group*-Instanzen), die dieser Sprache zugeordnet sind, während der Initialisierung in einer Liste hinzugefügt. Die Methode *run(self)* führt wiederum die *run(self, ...)*-Methode aller Gruppen der jeweiligen Sprache aus. Im Anschluss werden alle Statistiken der Gruppen aggregiert und ebenfalls als *.csv* abgespeichert.

3.2. Implementation

Um die Applikation sinnvoll zu verwenden, ist es wichtig zu verstehen, welche Benutzereingaben für die Applikation notwendig sind und wie die generierten Daten zu interpretieren sind.

3.2.1. Benutzereingaben

Damit die Applikation wie erwartet funktioniert, sind ein paar Benutzereingaben notwendig. Andere Benutzereingaben verbessern ausschließlich Lesbarkeit und die Verwendung der Ergebnisse. Alle Benutzereingaben sind entweder im *.csv*- oder im *.json*-Format. Sie lassen sich also beispielsweise in *LibreOffice Calc*, *Microsoft Excel* oder mit einer beliebigen Textbearbeitungs-Anwendung wie *Notepad*, *Notepad++* oder *Sublime Text* bearbeiten.

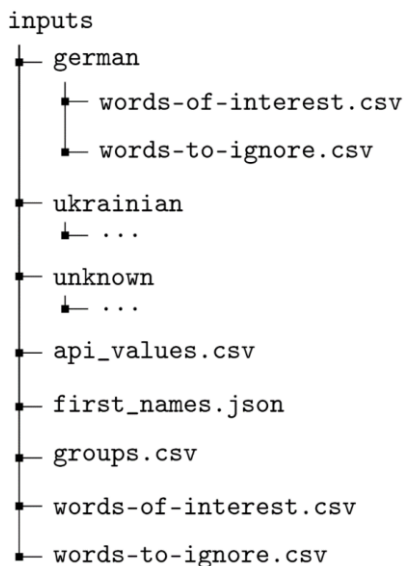


Abbildung 4: Beispiel der Ordner-Struktur der Benutzereingaben

Eine beispielhafte Darstellung der Ordner-Struktur der Benutzereingaben für die Sprachen Deutsch und Ukrainisch.

In Abbildung 4 ist eine beispielhafte Ordner-Struktur der Benutzereingaben zu sehen. Für jede Sprache gibt es einen Ordner, der zwei .csv-Dateien enthält. Diese Ordner und die entsprechenden Dateien werden automatisch erstellt, sobald die Applikation mit einer Gruppe in der jeweiligen Sprache ausgeführt wird. Alternativ lässt sich diese Ordner-Struktur auch mit dem `--setup`-Parameter erstellen. Beispielhaft für genau diese Ordner-Struktur: `python main.py --setup german ukrainian`.

Neben den sprachspezifischen Benutzereingaben gibt es außerdem einige Dateien, die sprachübergreifend als Benutzereingaben genutzt werden. Hierzu gehören die `api_values.csv`, `first_names.json`, `groups.csv`, `words-of-interest.csv` und `words-to-ignore.csv`.

api_values.csv

Diese Datei ist notwendig und muss mindestens einen Eintrag enthalten. Sie enthält drei Spalten: `label`, `api_id` und `api_hash`. Hier werden die Accounts eingetragen, die für die Nutzung der Applikation bereitgestellt werden. Das `label` dient als Identifikator und sollte einzigartig gewählt werden, `api_id` und `api_hash` werden von Telegram pro Account und Handynummer bereitgestellt (Telegram 2022b). Diese Parameter sind für jeden Account einzigartig und werden benötigt, damit sich die Applikation in dem gewünschten Account anmelden kann.

first_names.json

Diese Datei enthält eine Liste von Namen, die innerhalb einer Sprache typischerweise als weiblich oder männlich gelesen werden. Diese Liste wird verwendet, um die Verteilung der Gender von Mitgliedern in Gruppen zu approximieren. Zum Zeitpunkt der Veröffentlichung besteht diese Liste ausschließlich aus ukrainischen und deutschen Namen. Je nach Sprache, die mit der Applikation analysiert werden soll, lohnt es sich diese Datei, um zusätzliche Namen zu erweitern.

groups.csv

Bei der Ausführung der Telegram-Research-App wird diese Datei erstellt und aktualisiert. Sie enthält alle Gruppen, in denen sich einer der Accounts befindet. Neben dem Identifikator und dem Namen der Gruppe, enthält diese Datei auch die Parameter *language* und *active*, die von dem/der Nutzer*in eingegeben werden kann. In der Spalte *language* kann eine Sprache angegeben werden, um diese für die Analyse gemeinsam mit anderen Gruppen dieser Sprache zu aggregieren. Falls keine Angabe gemacht wird, wird bei der Analyse angenommen, dass die Sprache nicht bekannt ist und die Gruppe gemeinsam mit allen anderen Gruppen, deren Sprachen nicht bekannt sind, analysiert und aggregiert. Wenn bestimmte Gruppen von der Analyse ausgeschlossen bleiben sollen, kann die Spalte *active* von *True* auf *False* gesetzt werden.

words-of-interest.csv

Pro Sprache und auch sprachübergreifend können hier Wörter gesammelt werden, die für die Analyse von besonderem Interesse sind.

words-to-ignore.csv

Pro Sprache und auch Sprach-übergreifend können hier Wörter gesammelt werden, die für die Analyse ausgeschlossen werden sollen. Beispielsweise können hier Artikel, Konnektoren oder Begriffe wie *http* oder *www* gesammelt werden, die potenziell häufig vorkommen, aber einzeln betrachtet keine Bedeutung haben.

3.2.2. Generierte Daten

Die Ausgabe der Applikation ist ähnlich wie die Eingabe in einer Ordner-Struktur nach Sprachen unterteilt. Für alle Gruppen werden die Statistiken als Ergebnis der Applikation hier gespeichert. Darüber hinaus sind dort außerdem die aggregierten Ergebnisse pro Sprache zu finden. Das Ausgabe-Format ist ausschließlich *.csv* und *.png*. Die Ergebnisse können also ähnlich wie die Eingaben beispielsweise mit *LibreOffice Calc*, *Microsoft Excel* oder mit einer beliebigen Textbearbeitungs-Anwendung wie *Notepad*, *Notepad++* oder *Sublime Text* eingesehen werden. Die Bild-Dateien mit der Endung *.png* können einfach als Bilder betrachtet werden.



Abbildung 5: Beispiel der Ordner-Struktur der Programmausgaben

Beispielhafte Darstellung der Ordner-Struktur der Programmausgaben für die Sprachen Deutsch und Ukrainisch.

Ein beispielhafter Ausschnitt der Ausgabe kann in Abbildung 5 betrachtet werden. In der ersten Ebene ist die Ausgabe nach Sprachen sortiert. Für jede analysierte Woche wird für jede Sprache ein Ordner mit den jeweiligen Ergebnissen erzeugt. Innerhalb dieser Ordner befinden sich dann jeweils die Ergebnisse für jede Gruppe und zudem die aggregierten Statistiken. Darüber hinaus gibt es noch die Datei *Total-Overview.csv*, welche für jede Sprache pro Woche einen Eintrag erhält, um unter anderem Mitgliederzahlen zu dokumentieren.

Total-Overview.csv

Diese Datei wird pro Sprache einmal erstellt und pro Durchlauf wird dieser Tabelle eine Zeile hinzugefügt. Enthalten sind Informationen über die gesamte Anzahl an Mitgliedern in allen Gruppe, die Anzahl an einzigartigen Mitgliedern über alle Gruppen, die Anzahl aller betrachteten Gruppen und eine Abschätzung über die Gender-Verteilung aller einzigartigen Mitglieder. Bei dieser Tabelle ist sehr wichtig zu beachten, dass sich die Ausgabe Woche zu Woche massiv unterscheiden kann, wenn die Benutzereingaben angepasst werden. Wenn beispielsweise in einer Woche 100 Gruppen

analysiert werden und in der nächsten 300, verändert sich dementsprechend natürlich die Anzahl an Mitgliedern und möglicherweise auch die Gender-Verteilung.

Group-Overview.csv

Diese Datei wird jede Woche für jede Sprache erstellt. Sie enthält Gruppen-spezifische Statistiken über Mitgliederanzahl, Anzahl an verschickten Nachrichten über die analysierte Woche und eine Abschätzung über die Gender-Verteilung pro Gruppe.

Group-Participant-Network.csv

In dieser Datei sind alle Mitglieder aller Gruppen anonymisiert aufgelistet und es ist vermerkt, in welchem Gruppen sie sich jeweils befinden. Die hier enthaltenen Daten bieten Grundlage für Netzwerk-Analyse.

Weekly-Joins (.csv und .png)

In diesen Dateien ist in tabellarischer und grafischer Form aufgezeichnet, wie viele Mitglieder über die analysierte Woche den Gruppen insgesamt beigetreten sind. Diese Statistik ist insbesondere dann hilfreich, wenn Interesse an Entwicklung der Mitgliederzahlen von Gruppen besteht. Die Telegram-API bietet leider keine Möglichkeit die Mitgliederzahlen retrospektiv anzufragen. Daten wie diese stellen eine Möglichkeit dar, trotzdem eine Approximation über die Mitgliederzahlen zu erhalten. Wichtig zu beachten ist, dass einige Administratoren (vorwiegend großer Gruppen) in vielen Fällen die Beitritts-Nachricht im Chat blockieren oder automatisch löschen lassen. Daher ist diese Statistik für manche Gruppen unbrauchbar. Diese Statistik wird sowohl für alle Gruppen als auch aggregiert erstellt. Ein Beispiel ist in Abbildung 6 zu sehen.

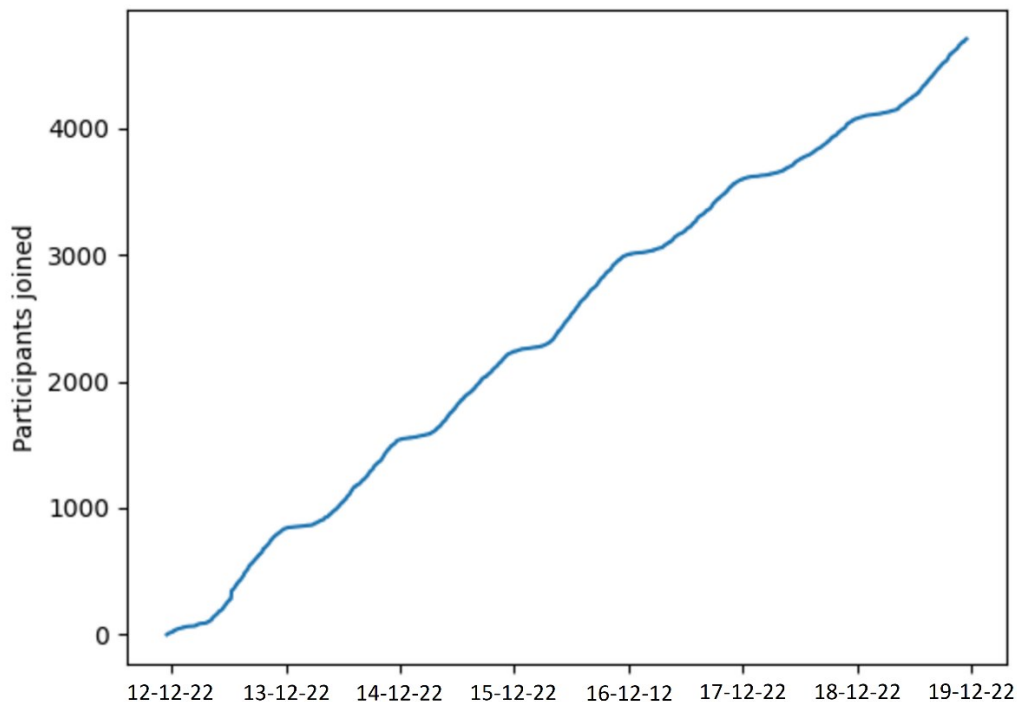


Abbildung 6: Mitglieder-Beiträge in untersuchten Telegram-Gruppen über eine Woche

Beispielhafte Beiträge über eine Woche vom 12. bis zum 18. Dezember 2022 aus 238 Gruppen mit insgesamt 244.822 einzigartigen Mitgliedern. Quelle: Telegram © Minor

Word-Frequencies.csv

In dieser Datei wird tabellarisch die Häufigkeit *aller* Wörter aufgelistet, die in Nachrichten verwendet und nicht von dem/der Nutzer*in ignoriert werden. Hierbei wird ein Wort mehrfach gezählt, wenn es mehrfach in einer Nachricht vorkommt. Diese Statistik wird sowohl für alle Gruppen als auch aggregiert erstellt.

Words_of_Interest-Frequencies.csv

In dieser Datei wird tabellarisch die Häufigkeit *aller ausgewählten* Wörter aufgelistet, die in Nachrichten verwendet werden. Hierbei wird ein Wort mehrfach gezählt, wenn es mehrfach in einer Nachricht vorkommt. Diese Statistik wird sowohl für alle Gruppen als auch aggregiert erstellt.

Words_of_Interest-Pair-Frequencies.csv

Diese Datei enthält die Häufigkeit von Wort-Paaren, die in Nachrichten verwendet werden. Die Paare müssen mindestens ein von dem/der Nutzer*in ausgewähltes Wort enthalten. Ein Paar wird pro Nachricht maximal einmal gezählt, auch wenn das Wort-Paar mehrfach in einer Nachricht vorkommt. Die Reihenfolge, in der die Wörter vorkommen, wird nicht beachtet. Das heißt die Position der Wörter innerhalb der Nachricht ist für diese Statistik unwichtig. Diese Statistik wird sowohl für alle Gruppen als auch aggregiert erstellt.

Daily-Activity

Diese Dateien enthalten Daten darüber, wie sich die verschickten Nachrichten *nach Wochentag* für die analysierte Woche verteilen in tabellarischer und grafischer Form. Diese Statistik wird sowohl für alle Gruppen als auch aggregiert erstellt. Ein Beispiel ist in Abbildung 7 zu sehen.

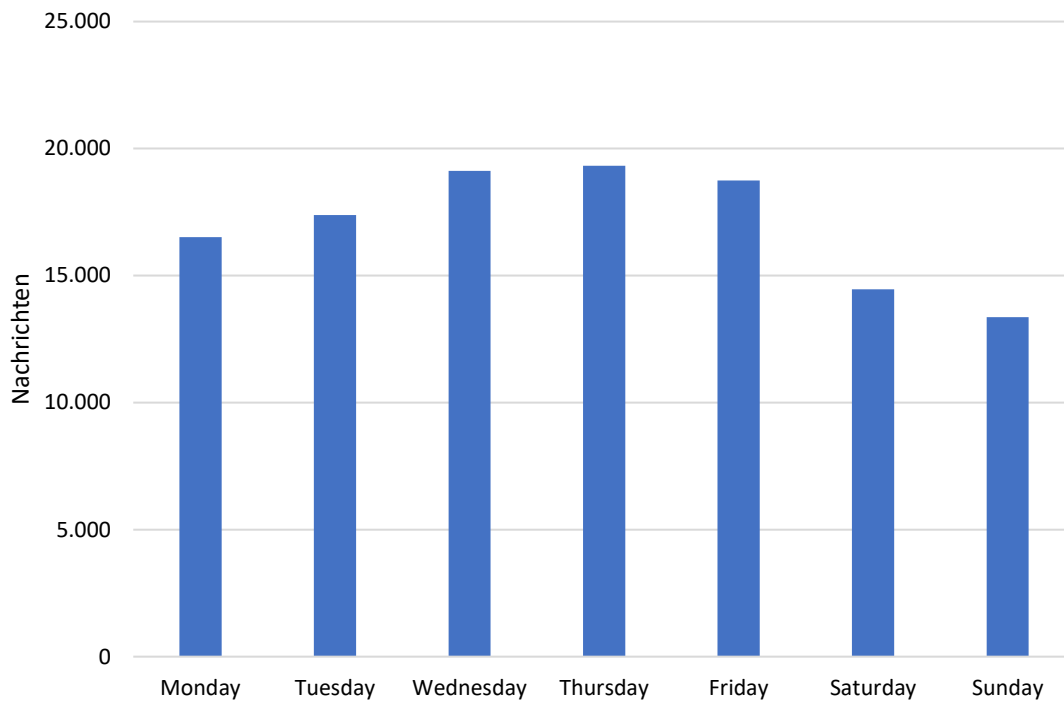


Abbildung 7: Nachrichtenverteilung nach Wochentag

Beispiel für die Nachrichtenaktivität von 238 ausgewählten Gruppen nach Wochentag. Quelle: Telegram © Minor

Hourly-Activity (.csv und .png)

Diese Dateien enthalten Daten darüber, wie sich die verschickten Nachrichten *nach Uhrzeit* für die analysierte Woche verteilen in tabellarischer und grafischer Form. Diese Statistik wird sowohl für alle Gruppen als auch aggregiert erstellt. Ein Beispiel ist in Abbildung 8 zu sehen.

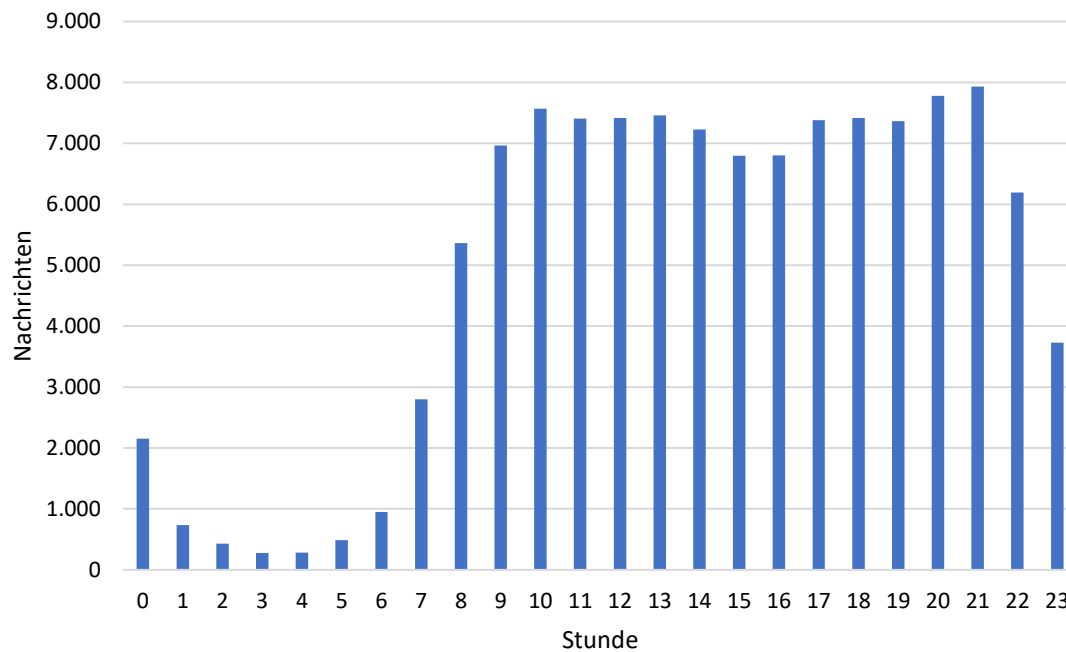


Abbildung 8: Nachrichtenverteilung nach Uhrzeit

Beispiel für die Nachrichtenaktivität von 238 ausgewählten Gruppen nach Uhrzeit. Quelle: Telegram © Minor

4. Verwendung der Telegram-Research-App

Zunächst ging es vor allem um den technischen Hintergrund und wie die Telegram-Research-App implementiert wurde. Dieses Kapitel legt den Fokus auf die Verwendung der Telegram-Research-App und betrachtet technische Voraussetzungen, Installation und Vorbereitung sowie erste Ausführung und langfristige Verwendung der App.

4.1. Technische Voraussetzungen

Die Telegram-Research-App braucht nur geringe Ressourcen für die Ausführung. Während der Entwicklung und für die Tests wurde ein Linux-basiertes Debian-System mit einem *Intel Xeon Processor (Skylake, IBRS)* Prozessor und 4 GB Arbeitsspeicher verwendet. Je nach Anzahl der Gruppen, versandten Nachrichten und Anzahl an Mitgliedern ist der Bedarf an Arbeitsspeicher variabel. Für 302 Gruppen, mit insgesamt ca. 450.000 Mitgliedern, brauchte die Telegram-Research-App nach unseren Messungen etwas weniger als einen Gigabyte Arbeitsspeicher.

4.2. Installation und Vorbereitung der Telegram-Research-App

- 1) Die Applikation muss als erstes von dem Git-Repository (Minor-Kontor 2023) heruntergeladen werden. Hierfür gibt es zwei Möglichkeiten, entweder per *git clone* (git-scm 2022) oder per Download und entpacken. Wenn man sich bereits mit *git* auskennt, bietet sich dieser Weg an, da dadurch die Einbindung zukünftiger Updates und Verbesserungen einfacher wird. Abbildung 9 zeigt, wo sich der Download für die Applikation auf der Github-Seite befindet.

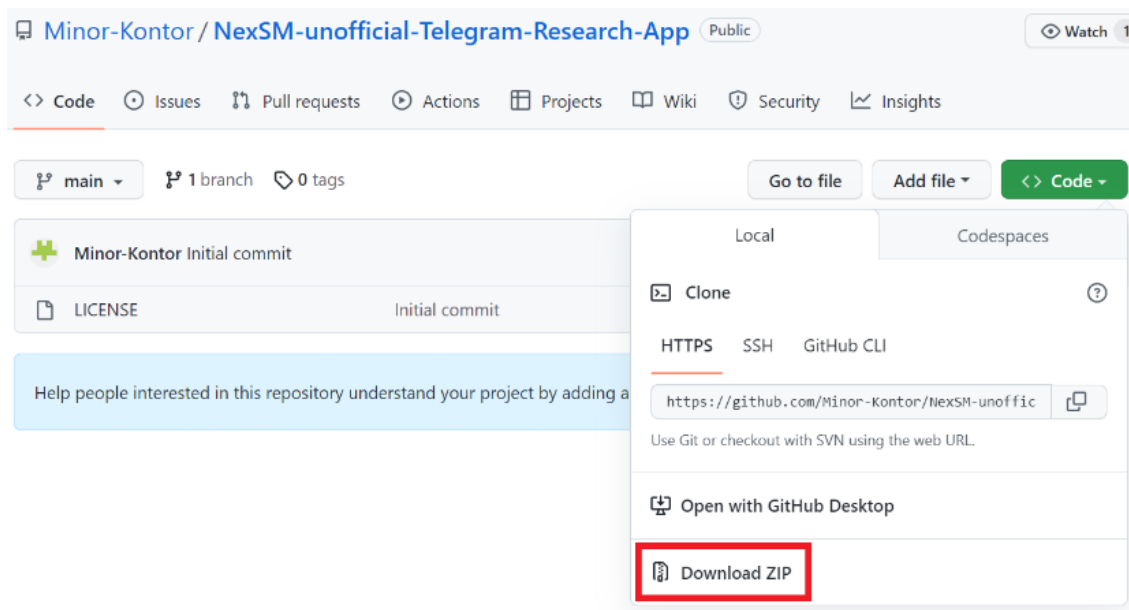


Abbildung 9: Download der Telegram-Research-App von Github (Minor-Kontor 2023)

Der Download der ZIP-Datei ist im Bild rot hervorgehoben.

- 2) Die Telegram-Research-App ist auf Basis der Programmiersprache Python implementiert. Für die Ausführung ist notwendig, dass ein korrekter Python-Interpreter installiert (Python 2022b). Die Telegram-Research-App wurde in Python 3.9.15 implementiert und getestet.
- 3) Zunächst muss eine Konsole (Windows: Eingabeaufforderung, Linux: Terminal) geöffnet und in den heruntergeladenen Ordner navigiert werden.
- 4) Für die Installation von Bibliotheken wie beispielsweise *Teleton* ist es notwendig, dass *pip* installiert ist. Dafür muss in der Konsole folgender Befehl ausgeführt werden:
python -m ensurepip --upgrade
- 5) Folgender Befehl installiert alle von der Telegram-Research-App benötigten Bibliotheken:
pip install -r requirements.txt
- 6) Alle Voraussetzungen für die Verwendungen der Telegram-Research-App sind jetzt erfüllt.

4.3. Erste Ausführung

Bevor die Applikation zum ersten Mal ausgeführt und für Datenanalyse verwendet werden kann, muss sie zuerst korrekt konfiguriert werden. Dafür ist es notwendig die Eingabe-Datei *api-values.csv* wie in Kapitel 3.2.1 beschrieben auszufüllen. Außerdem ist es wichtig zu beachten, dass die Telegram-Research-App ausschließlich die Gruppen der angegebenen Accounts betrachtet.

Nachdem die korrekten Werte in der Datei *api-values.csv* eingetragen wurden, kann die Telegram-Research-App mit folgendem Befehl ausgeführt werden, um relevante Eingabe-Dateien für die relevanten Sprachen zu erzeugen:

```
python main.py --setup [Sprache1] [Sprache2] [...]
```

Hierbei kann der/die Nutzer*in beliebig viele Sprachen mit Leerzeichen getrennt angeben. Beispielsweise kann die Applikation mit folgendem Befehl für Ukrainisch und Deutsch konfiguriert werden:

```
python main.py --setup deutsch ukrainisch
```

Während der ersten Ausführung der Applikation fragt die Telegram-API nach einer Bestätigung, dass die Applikation auf die jeweiligen Telegram-Accounts Zugriff haben darf. Zuerst muss die, zu dem Account gehörende, Telefonnummer angegeben werden. Im Anschluss wird an diesen Telegram-Account ein Code geschickt, der in die Konsole eingegeben werden muss. Dies muss für jeden angegebenen Telegram-Account nur einmal gemacht werden, ab der zweiten Ausführung ist das Gerät bekannt.

Nachdem die Konfiguration erfolgreich abgeschlossen wurde, existiert der *inputs*-Ordner für die Benutzereingaben wie in Kapitel 3.2.1 beschrieben und kann genutzt werden, um die Datenbeschaffung und -analyse den Anforderungen des/der Nutzer*in anzupassen. Insbesondere ergibt es Sinn, die Datei *inputs/groups.csv* wie beschrieben anzupassen.

4.4. Langfristige Verwendung

Nachdem die Applikation installiert und konfiguriert wurde, kann es einfach mit folgendem Befehl aus dem Installationsordner ausgeführt werden:

```
python main.py
```

Die Telegram-Research-App wird immer für die letzte vollständige Woche ausgeführt, wenn die Applikation also am 13. Dezember 2022 gestartet wird, würde es die Nachrichten aus der Woche vom 12. bis zum 18. Dezember 2022 anfragen und verarbeiten. Wenn der Wunsch besteht die Telegram-Research-App für eine spezifische vergangene Woche auszuführen, kann dies einfach mit folgendem Befehl ausgeführt werden (hier beispielsweise für die Woche vom 05. bis 11. Dezember 2022):

```
python main.py --week 2022-12-05
```

Dabei ist egal welcher Tag der jeweiligen Woche angegeben wird, es wird immer die volle Woche von Montag bis Sonntag analysiert. Mitgliederzahlen können retrospektiv leider nicht von Telegram angefragt werden, aber mithilfe der wöchentlichen Beitritte zu den Gruppen (*Weekly-Joined.csv*) approximiert werden.

Wenn eine Applikation für die Nutzung von Telegram als Datenquelle, wie die Telegram-Research-App, längerfristig für die Bereitstellung von Daten verwendet werden soll, lohnt es sich über eine Automatisierung der Ausführung der Applikation nachzudenken. Für Unix-Systeme bietet sich hierfür beispielsweise das Programm *crontab* an (Kerrisk 2022).

Um die Ausgabe iterativ zu optimieren, kann der/die Nutzer*in beispielsweise die Ausgabe der Programmausgabe-Datei *Weekly-Word-Frequencies.csv* nutzen, um Wörter, welche für die Analyse als nicht relevant erachtet werden, mithilfe der Eingabe-Dateien *words-to-ignore.csv* ignorieren zu

lassen. So kann die Ausgabe mit jeder Ausführung optimiert werden und so langfristig die Nützlichkeit der Applikation verbessert werden.

4.4.1. Aktualisierung der Telegram-Research-App

Um die Telegram-Research-App zu aktualisieren, gibt es grundsätzlich zwei Möglichkeiten. Entweder hat man bereits bei der ersten Installation *git* verwendet und kann mithilfe von dem Befehl *git pull* die neuste Version der Telegram-Research-App herunterladen und automatisch integrieren. Alternativ kann auf der Github-Seite (Minor 2023) die neueste Version heruntergeladen werden und alle Dateien bis auf den *inputs*- und *outputs*-Ordner ersetzt werden.

5. Analyse und Auswertung der initialen Daten

Im Rahmen der Entwicklung der Telegram-Research-App wurden bereits initiale Daten untersucht und werden im folgenden Kapitel diese Ergebnisse beispielhaft präsentieren und interpretieren. Außerdem gehen wir auf einige Herausforderung in den Analysen ein, versuchen diese zu erklären und betrachten Möglichkeiten, wie diese Probleme angegangen und verbessert werden könnten. Diese Daten basieren auf von Fem.OS für die Beratungsarbeit ausgewählten Gruppen und wurden über einen kurzen Zeitraum erstellt. Dementsprechend handelt es sich bei den Ergebnissen um einen ersten Einblick und keine repräsentative Studie.

5.1. Betrachtete Gruppen

Insgesamt wurden für die initiale Analyse 302 Telegram-Gruppen von zugewanderten Personen in Deutschland ausgewählt. Dies sind Gruppen, die über die Telegram-Accounts von Fem.OS im Rahmen der aktiven Beratungsarbeit mit zugewanderten Frauen aus Drittstaaten in Deutschland erfasst wurden. Von diesen Gruppen wurden anhand der genutzten Sprachen und der Selbstidentifikation der Mitglieder anhand der Gruppentitel 238 als Ukrainisch, 28 als Persisch und 36 als unbekannt eingeordnet. In diesem Kapitel werden vor allem die Daten der ukrainischsprachigen Gruppen beschrieben, da diese den Großteil der analysierten Gruppen ausmachen.

5.2. Verwendete Benutzereingaben

Dazu haben wir uns in der Auswahl der Benutzereingaben auf ukrainische Wörter und Namen fokussiert und für die Auswertung der initialen Daten die *words-of-interest.csv* und die *words-to-ignore.csv* Liste auf Ukrainisch angelegt. Die verwendeten Vornamen für die *first_names.json*-Datei kann im Github-Repository angeschaut werden (Minor 2023).

5.3. Informationsbedarfe

Die Häufigkeit von verwendeten Wörtern zu beobachten ist ein einfacher Weg, die Informationsbedarfe der Mitglieder von verschiedenen Gruppen herauszufinden. Die grundsätzliche Idee besteht darin, aus häufig verwendeten Wörtern Bedarfe abzuleiten. Wenn beispielsweise Wörter wie *Arbeitsamt*, *Arbeit* oder *Beschäftigung* in einer Gruppe häufig verwendet werden, ist eine mögliche

Schlussfolgerung, dass Mitglieder dieser Gruppe nach Arbeit oder Unterstützung bei berufsbezogenen Angelegenheiten suchen.

5.3.1. *Verwendete Sprachen*

Die Unterteilung der Telegram-Gruppen nach Sprachen passiert nach Ermessen der Nutzer*in von der Telegram-Research-App. Grundsätzlich brauchen Gruppen eine sprachliche Grundlage, über die sich alle Mitglieder unterhalten können. Die Annahme, dass innerhalb einer Telegram-Gruppe eine spezifische Sprache gesprochen wird, ist allerdings eine vereinfachte Darstellung der Realität und führt dementsprechend zu Fehlern. Innerhalb verschiedener Länder werden Dialekte gesprochen und grundsätzlich sind nicht alle Bewohner*innen eines Landes auf demselben Sprachniveau, so dass es in manchen Fällen Sinn ergibt, eine andere Sprache zu nutzen, um sich auszutauschen.

Unter den in Tabelle 3 aufgezählten Wörtern befinden sich vor allem russischsprachige Begriffe, obwohl es sich bei den betrachteten Chats laut unserer Analyse um Telegram-Gruppen handelt, die hauptsächlich durch ukrainische Geflüchtete genutzt werden. Insbesondere bei ostslawischen Sprachen sind aufgrund historischer Zusammengehörigkeit Sprachgruppen nicht immer an heutige nationale Grenzen gebunden. Im Ukrainischen gibt es für Mischsprachen, die beispielsweise sowohl ukrainisch und russisch enthalten, sogar ein spezifisches Wort „суржик“ beziehungsweise suržik⁴ (Gagarina 2014).

Normalerweise haben Gruppen Regeln diesbezüglich und spezifizieren erlaubte Sprachen innerhalb der Gruppenbeschreibung. In einer der von uns untersuchten Gruppen für und von ukrainischen Geflüchteten, steht in der Beschreibung: „ЯЗЫКИ группы: немецкий и русский! Украинский не понимают 70 % местных участников“. Zu Deutsch: „GRUPPENSPRACHEN: Deutsch und Russisch! Ukrainisch wird von 70 % der lokalen Mitglieder nicht verstanden“⁵. Auch wenn dies nur eine ausgewählte Stichprobe ist, zeigt dies, dass die jeweils gesprochenen Sprachen nicht zwangsweise mit der Herkunft der Mitglieder übereinstimmt.

Transliteration zwischen verschiedenen Sprachen erweist sich als eine weitere noch ungelöste Herausforderung. Es gibt verschiedene Normen, für die Transliteration von Sprachen, die auf dem kyrillischen Alphabet basieren, um sie in ein lateinisches Alphabet zu transliterieren. Neue Wortschöpfungen im deutschen, die für ukrainische Geflüchtete von großer Relevanz sind, werden sehr unterschiedlich in die eigene Sprache übernommen. Beispielsweise wird das deutsche Wort Jobcenter teilweise Джобцентр, Джопцентр oder Жобцентр geschrieben, für das Wort Kindergeld finden sich Schreibweise wie Кіндергельт und Кіндергельд.

5.3.2. *Wörter ohne eigenständigen Inhalt*

Präpositionen, Konjunktionen und Pronomen enthalten ohne Kontext keinen Inhalt. Dasselbe gilt für viele Adjektive, manche Verben und einzelne Nomen. Der von uns gewählte Ansatz Stichwörter

⁴ Transliteration nach ISO 9.

⁵ Übersetzt mit deepl.

zu zählen hat eindeutig erkennbare Herausforderungen und Einschränkungen in dieser Hinsicht. Dies ist ein grundsätzliches Problem der Methodik und lässt sich nur bis zu einem gewissen Grad bereinigen.

Tabelle 3: Auswahl an ignorierten Wörtern

Beispielhafte Auswahl an 10 Wörtern, die in **Fehler! Verweisquelle konnte nicht gefunden werden.** ignoriert wurden und dementsprechend nicht in der Analyse vorkamen. Quelle: Telegram © Minor

Wort	Übersetzung ⁶
i	und
та	und
в	in
також	auch
ні	keine
як	wie
так	ja
не	nicht
тільки	nur
а	und

Um dieses Problem anzugehen, gibt es in der Telegram-Research-App die Möglichkeit mithilfe von der Eingabe-Datei *words-to-ignore.csv* aufkommende Wörter, die nicht von Interesse für die Analyse sind, zu ignorieren. Das ermöglicht eine Bereinigung der Stichwortlisten, wie sie beispielsweise in **Fehler! Verweisquelle konnte nicht gefunden werden.** zu sehen ist. In

Tabelle 3 ist ein Ausschnitt einer möglichen Liste von Wörtern zu sehen, die für die Analyse keinen Wert haben.

Tabelle 4: Auswahl von 22 für die Beratungsarbeit relevanten Wörter

Beispielhafte Auswahl an 22 Wörtern, welche dabei helfen soll, die Informationsbedarfe von ukrainischen Geflüchteten zu analysieren. Enthalten sind außerdem Synonyme, um nach Möglichkeit verschiedene Begriffe für den gleichen Informationsbedarf abzudecken. Quelle: Telegram © Minor

Wort	Übersetzung
Реєстрація	Anmeldung
Робота	Arbeit
Працювати	Arbeit
Здоров'я	Gesundheit
Пенсія	Rente
Самостійність	Selbstständigkeit
Незалежність	Selbstständigkeit
Податки	Steuern
Плоскі	Wohnung

⁶ Übersetzt mit deepl.

квартира	Wohnung
Освіта	Bildung
Навчання	Bildung
авто	Auto/Fahrzeug
Дискримінація	Diskriminierung
паспорт	Ausweisdokument
тварина	Haustier
вакцинація	Impfung
батьки	Eltern
дитина	Kind
больница	Krankenhaus
врач	Arzt
лікар	Arzt

Neben der Möglichkeit unerwünschte und inhaltslose Wörter zu ignorieren, gibt es auch die Möglichkeit, eine Stichwortliste von relevanten Wörtern anzugeben. Die Eingabe-Datei *words-of-interest.csv* fungiert als ein Filter, der genutzt wird, um die in ihr enthaltenen Wörter in der Programmausgabe-Datei *Words_of_Interest-Frequencies.csv* hervorzuheben. In Tabelle 4 ist eine beispielhafte Liste gezeigt, welche dabei helfen soll, die Informationsbedarfe von ukrainischen Geflüchteten zu analysieren.

Tabelle 5: Häufig verwendete Wörter für die Beratungsarbeit

Häufigkeit der am meisten verwendeten Worte, die vorher für die Analyse als relevant angegeben wurden.
Quelle: Telegram © Minor

Wort	Häufigkeit	Übersetzung
паспорт	1.267	Ausweisdokument
авто	733	Auto/Fahrzeug
врач	485	Arzt
квартира	373	Wohnung
школа	287	Schule
шукаю	243	suche
проживання	189	Unterkunft
дитина	172	Kind
навчання	168	Bildung
робота	143	Arbeit
лікар	125	Arzt
працювати	124	Arbeit
квиток	86	Ticket
реєстрація	81	Anmeldung
платити	69	bezahlen
батьки	67	Eltern
вокзал	65	Bahnhof
ковид	62	Corona Virus
здоров'я	57	Gesundheit
освіта	50	Bildung

In Tabelle 5 ist die Häufigkeit von verschiedenen Wörtern, die für Analyse als relevant angegeben wurden, dargestellt. Anhand der Reihenfolge lassen sich verschiedene Informationsbedarfe approximieren. Die am meisten verwendeten Wörter beschreiben einen Informationsbedarf bezüglich

Ausweisdokumenten, Fortbewegungsmitteln, ärztliche Versorgung, Wohnung/Unterkunft und Schule. Es scheint, als wären für viele der Mitglieder in den ausgewählten Telegram-Gruppen grundlegende Informationsbedarfe zu Aufenthalt und Ausweis, sowie medizinischer Versorgung und Unterkunft noch nicht ausreichend geklärt. Anschließend scheinen Themen zu Schule, Kinder und Bildung von hoher Relevanz zu sein. Zudem scheint auch ein wichtiges Thema Arbeit zu sein.

Grundsätzlich lässt sich aus diesen Schlagwörtern nicht immer eine klare Schlussfolgerung ziehen, was der konkrete Bedarf ist. Beispielsweise kommt das Wort Auto/Fahrzeug häufig vor, aber aus dem Kontext gerissen ist es schwierig zu beurteilen, ob es den Mitgliedern darum geht ein Auto in Deutschland zu finden, ob sie auf der Suche nach einem Gefährt für die Flucht nach Deutschland sind oder ob sie ein Auto verkaufen wollen. Dementsprechend ist es wichtig, diese Informationen nicht zu stark zu gewichten und darauf aufbauend weitere Recherchearbeit zu leisten.

5.3.3. Konjugation und Synonyme

Mithilfe einer einfachen Stichwort-Analyse ist es nicht möglich das Vorkommen verschiedener Konjugationen von Wörtern gemeinsam zu zählen. Dadurch wird die Analyse in jedem Fall nie die exakte Häufigkeit der verwendeten Wörter analysieren können und immer nur eine Approximation sein. Darüber hinaus bereiten Synonyme und Homonyme beim Zählen weitere Schwierigkeiten. Ähnlich wie Konjugationen werden Synonyme nicht korrekt gemeinsam gezählt, wodurch das Vorkommen weiter unterschätzt wird. Homonyme (Wörter mit mehreren Bedeutungen) wie im Deutschen beispielsweise *Bank* führen zu einer möglichen Überschätzung der semantischen Häufigkeit eines Begriffs.

Verschiedene Dialekte oder Änderungen in Sprache führen außerdem zu verschiedenen Schreibweisen für dasselbe Wort. Im Deutschen führte beispielsweise die Einführung der neuen deutschen Rechtschreibung zu vielen Veränderungen und daraus resultierend Diskrepanzen in Schreibweisen.

Um dieses Problem zu beseitigen, benötigt die Applikation ein *tieferes Verständnis* der Syntax und Semantik von verschiedenen Sprachen. Projekte wie die *nltk*-Bibliothek (Bird 2006; NLTK 2022) versuchen seit vielen Jahren die Entwicklung von natürlicher Spracherkennung voranzutreiben und haben bereits viele Erfolge damit erreicht. Grundsätzlich wird Sprache und Schrift in Chats und Gruppen anders verwendet als in Büchern oder Zeitschriften, dennoch gibt es sehr vielversprechende Ansätze, die sich in einer Applikation wie der Telegram-Research-App in Zukunft möglicherweise anwenden ließen. Berko et al. veröffentlichten bspw. 2021 eine Applikation, mit derer die *nltk*-Bibliothek auch auf Ukrainisch angewandt werden kann (Berko et al. 2021).

5.3.4. Stichwort-Analyse von Wortpaaren

Neben der Analyse von einzelnen Stichworten zählt die Telegram-Research-App auch die Häufigkeit von Wortpaaren. Damit die Anzahl der Paare nicht kombinatorisch explodiert, haben wir uns dazu entschieden nur Paare von Worten zu betrachten, in denen sich wenigstens ein Wort aus der Eingabe-Datei *words-of-interest.csv* befindet.

Tabelle 6: Häufig verwendete Wortpaare

Häufigkeit der am meisten verwendeten Wortpaare, von denen wenigstens eines der Wörter vorher für die Analyse als relevant angegeben wurden. Quelle: Telegram © Minor

1. Wort	2. Wort	1. Übersetzung	2. Übersetzung	Häufigkeit
паспорт	с	Ausweisdokument	mit	207
паспорт	загран	Ausweisdokument	im Ausland	180
паспорт	до	Ausweisdokument	zu	164
паспорт	или	Ausweisdokument	oder	151
паспорт	кто	Ausweisdokument	wer	135
паспорт	можно	Ausweisdokument	kann	132
авто	с	Auto/Fahrzeug	mit	126
квартира	с	Wohnung	mit	120
паспорт	как	Ausweisdokument	wie	120
паспорт	есть	Ausweisdokument	es gibt	115
паспорт	нужно	Ausweisdokument	müssen	115
паспорт	только	Ausweisdokument	nur	111
паспорт	пожалуйста	Ausweisdokument	bitte	110
врач	есть	Arzt	es gibt	108
паспорт	подскажите	Ausweisdokument	Sag mir	108
паспорт	германии	Ausweisdokument	Deutschland	105
паспорт	день	Ausweisdokument	Tag	105
паспорт	добрый	Ausweisdokument	Guten	105
врач	с	Arzt	mit	102
паспорт	без	Ausweisdokument	ohne	101

Tabelle 6 zeigt exemplarisch die häufigsten verwendeten Wortpaare. Ähnlich wie bei der Analyse der einzelnen Stichworte fällt schnell auf, dass ein Großteil der Kombinationen keine weitere Bedeutung haben und mehr Filterung notwendig wäre. Die letzte Zeile „Ausweisdokument, ohne“ ist möglicherweise ein Indikator für einen Bedarf nach Informationen über, wie damit umgegangen werden soll, wenn jemand kein aktuelles Ausweisdokument hat.

5.4. Aktivität der Gruppen

Die Aktivität der Gruppen kann unter anderem anhand der Anzahl verschickter Nachrichten verfolgt werden. Grundsätzlich kann die Aktivität einer Gruppe zurate gezogen werden, um für aufsuchende Beratung genau zu wissen, in welchen Gruppen viel geschrieben wird und zu welchen Uhrzeiten. Darüber hinaus können die Daten genutzt werden, um Kampagnen zu planen und herauszufinden, zu welchen Zeitpunkten und in welchen Gruppen man die größte Reichweite erzielt werden kann.

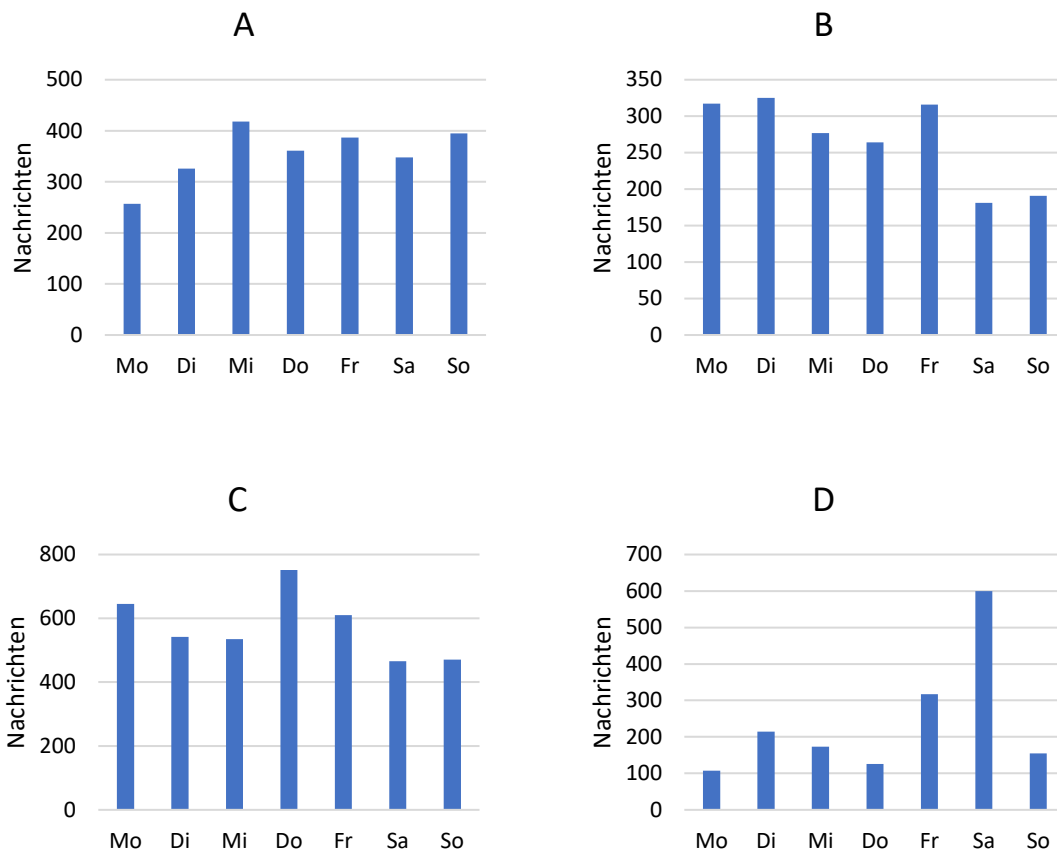


Abbildung 10: Vergleich verschiedener Aktivitäten nach Wochentag

Aktivität nach Anzahl verschickter Nachrichten der vier größten ukrainischen Gruppen über eine Woche. **A)** 38464 Mitglieder, 2492 Nachrichten; **B)** 25307 Mitglieder, 1871 Nachrichten; **C)** 21963 Mitglieder, 4021 Nachrichten; **D)** 21159 Mitglieder, 1692 Nachrichten. Quelle: Telegram © Minor

Abbildung 10 zeigt die vier größten ukrainischen Gruppen und deren Aktivitätsprofile für eine spezifische Woche. Auch wenn die Nachrichten über die Wochentage recht gleichmäßig verteilt sind, ist gut erkennbar, dass verschiedene Gruppen potenziell auch an verschiedenen Tagen aktiv sind. Auffällig ist, dass Gruppe D) sehr viel Aktivität am Samstag aufwies. Statistiken wie diese können beispielsweise genutzt werden, um den Einfluss von Öffentlichkeitsarbeit und Kampagnen auf die Aktivität einer Telegram-Gruppe zu steuern.

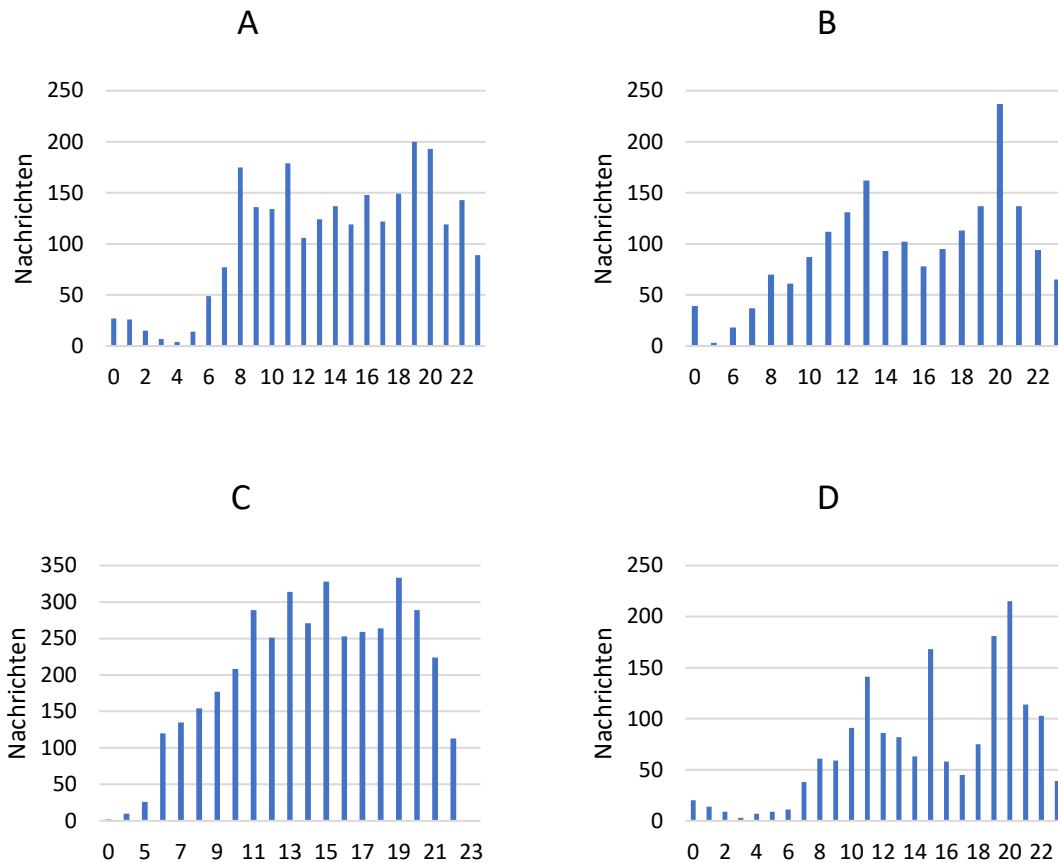


Abbildung 11: Vergleich verschiedener Aktivitäten nach Uhrzeit

Aktivität nach Anzahl verschickter Nachrichten der vier größten ukrainischen Gruppen über den Tag verteilt über eine Woche aggregiert. **A)** 38464 Mitglieder, 2492 Nachrichten; **B)** 25307 Mitglieder, 1871 Nachrichten; **C)** 21963 Mitglieder, 4021 Nachrichten; **D)** 21159 Mitglieder, 1692 Nachrichten. Quelle: Telegram © Minor

Abbildung 11 zeigt für dieselben vier größten ukrainischen Gruppen die Aktivität über eine Gruppe nach Tageszeit. Insgesamt ist bei allen Gruppen eine Konzentration der Nachrichten nachmittags/abends zu sehen.

5.5. Gender-Verteilung

Seit Beginn des Krieges in der Ukraine gilt in der Ukraine das Kriegsrecht, nach dem biologische Männer im Alter von 18 bis 60 das Land nicht mehr verlassen dürfen. Durch diesen Umstand ist zu erwarten, dass der Großteil der Geflüchteten aus der Ukraine nicht biologisch männlich ist. Um diese These zu prüfen und im Allgemeinen mehr über die jeweiligen Zielgruppen in Telegram-Gruppen herauszufinden, haben wir in der Telegram-Research-App eine Approximation des gewählten Gender implementiert.

Innerhalb von Telegram, können alle Nutzer*innen nach ihrem eigenen Ermessen einen Vor- und Nach- sowie einen Username wählen. Diese Namen werden von Telegram nicht geprüft und entsprechen dementsprechend nicht unbedingt den Namen, den die Nutzer*innen auf ihrem Ausweis stehen haben. Die Telegram-API erlaubt, diese Namen anzufragen. Um die gewählten Gender von © Minor – Projektkontor für Bildung und Forschung

Mitgliedern von Telegram-Gruppen zu approximieren, betrachtet die Telegram-Research-App die Vornamen der Mitglieder und vergleicht diese mit einer von uns zusammengestellten Liste von Namen, die in der Datei *first_names.json* aufgelistet sind. In der Liste haben wir insgesamt 2880 weiblich und 1707 männlich gelesene Namen gesammelt, die *first_names.json* kann auf der Github-Seite betrachtet werden (Minor2023).

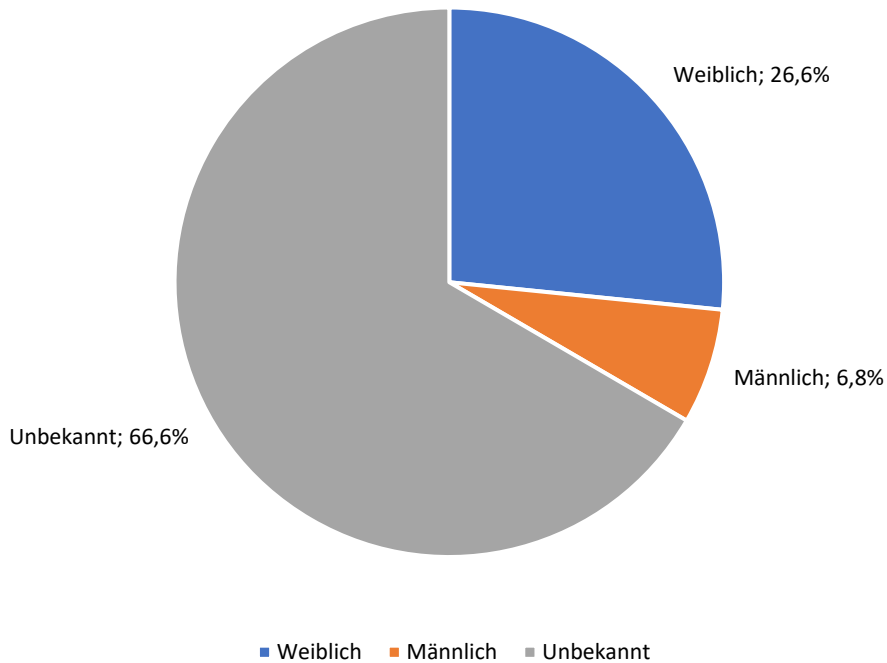


Abbildung 12: Gesamtverteilung des Gender über alle Telegram-Gruppen

Ergebnis der Approximation des gewählten Gender basierend auf angegebenen Vornamen. Unbekannt bedeutet, dass entweder kein Name gewählt wurde, der Name nicht bekannt ist oder dass der Name sowohl als männlich oder weiblich gelesen werden kann. Analysiert wurden 235 ukrainischsprachige Gruppen mit insgesamt 227.113 individuellen Mitgliedern. Quelle: Telegram © Minor

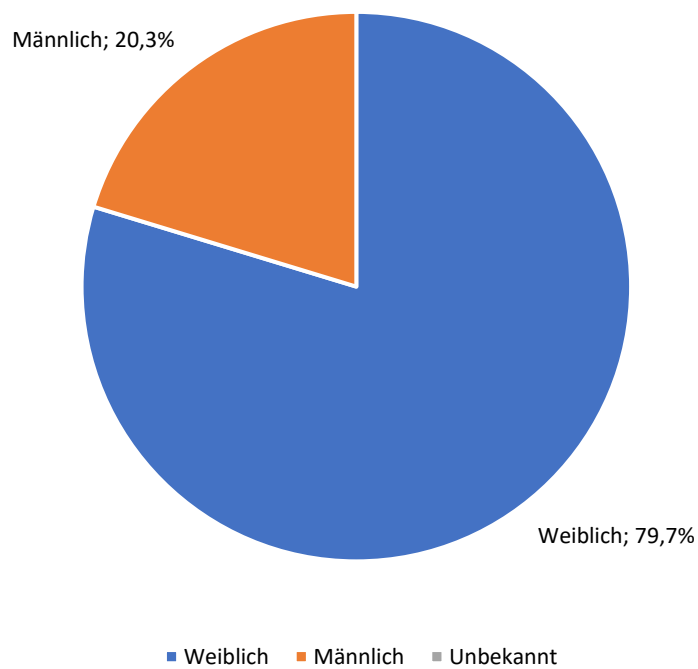


Abbildung 13: Verhältnis zwischen männlich und weiblich gelesenen Vornamen

Das Verhältnis zwischen weiblich und männlich gelesenen Vornamen, wenn die unbekannt und nicht gewählten Namen ausgelassen werden. Quelle: Telegram © Minor

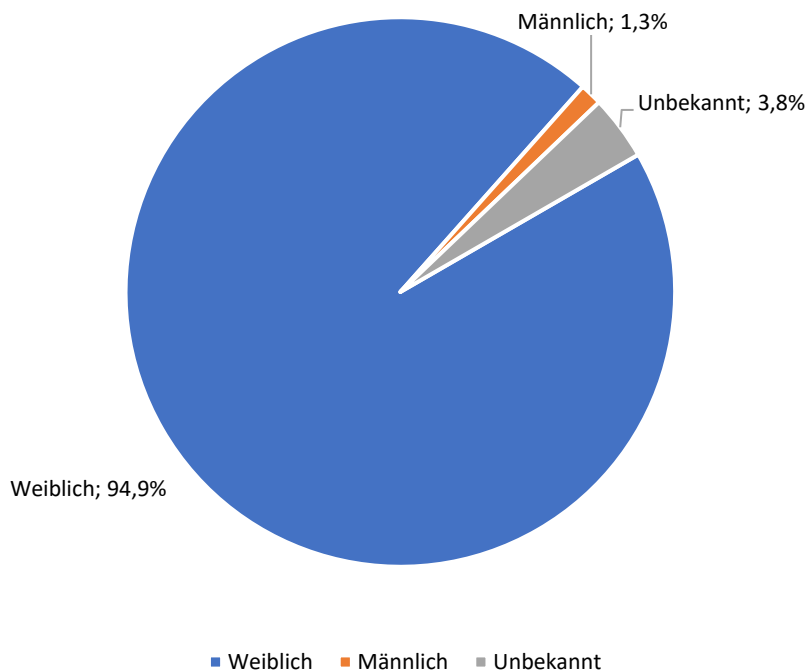


Abbildung 14: Verhältnis der Gender zwischen den Gruppen

Verhältnis der Anzahl Gruppen, die eine Mehrheit weiblich oder männlich gelesenen Vornamen haben. Quelle: Telegram © Minor

In den Abbildung 12, Abbildung 13 und Abbildung 14 ist eine beispielhafte Approximation der gewählten Gender von 235 ukrainischen Gruppen abgebildet. Dadurch, dass der Großteil der Mitglieder weibliche gelesene Vornamen gewählt hat, wird die Hypothese unterstützt, dass auf Telegram die Mehrheit der ukrainischsprachigen Nutzenden in Deutschland weiblich ist. Die Ergebnisse der Analyse stimmen mit den Umfrageergebnissen des BMI überein (Brücker et al. 2022). Es ist allerdings sehr wichtig anzumerken, dass, wie insbesondere in Abbildung 12 zu sehen ist, ca. 66 % der Vornamen nicht einem Gender zugeordnet werden konnten. Darüber hinaus ist die Liste der weiblich gelesenen Namen in *first_names.json* länger als die der männlichen.

Wenn gezielt die Informationsbedarfe nach Gender relevant sind, kann diese Statistik gut als Grundlage gewählt werden, um spezifische Gruppen zu fokussieren.

5.6. *Migrationsbewegungen*

Konkrete Migrationsbewegungen mithilfe der Telegram-API zu analysieren, stellt definitiv eine größere Herausforderung dar. Mithilfe von Mitgliederzahlen, kann Bewegung approximiert werden, diese kann jedoch nicht über die Telegram-API retrospektiv herausgefunden werden. Mithilfe von den Beitrittsnachrichten in Gruppen kann approximiert werden, wie viel neue Mitglieder Gruppen zu verzeichnen haben. Wenn diese in Gruppen nicht automatisch gelöscht werden, kann dies auch retrospektiv geschehen.

Darüber hinaus haben viele der ausgewählten Gruppen regionalen Bezug, sodass mithilfe von Netzwerkanalysen untersucht werden kann, inwieweit eine Verschiebung zwischen Städten stattfindet. Dieser regionale Bezug lässt sich häufig an den Gruppennamen und teilweise auch an den Inhalten der Gruppendiskussionen identifizieren.

5.6.1. *Entwicklung von Mitgliederzahlen*

Die Entwicklung von Mitgliederzahlen lässt sich nur über einen längeren Zeitraum analysieren. Um solch eine Analyse zu machen, lohnt es sich über mehrere Monate die Mitgliederzahlen zu betrachten. Dabei ist wichtig zu beachten, dass sich über Zeit potenziell die Anzahl an untersuchten Gruppen verändert und insbesondere wenn große Gruppen dazukommen oder weggelassen werden, dies einen großen Einfluss auf die Gesamtzahl der betrachteten Mitglieder hat.

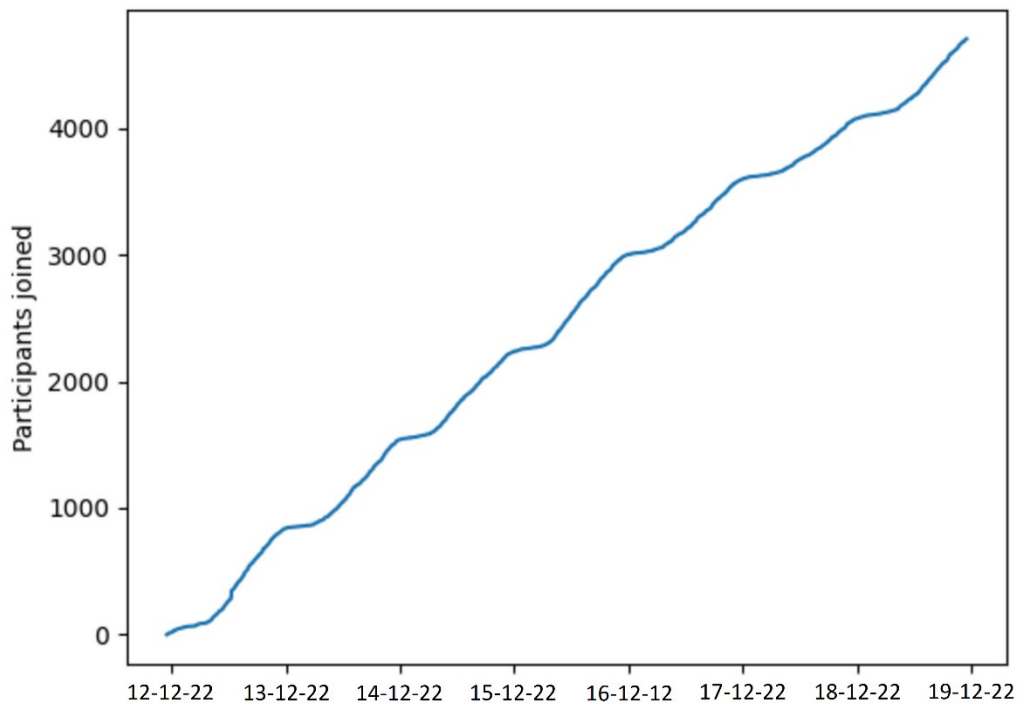


Abbildung 15: Mitglieder-Beiträge in untersuchten Telegram-Gruppen über eine Woche
 333.333. Quelle: Telegram © Minor

Um darüber hinaus eine Abschätzung zu haben, inwieweit neue Mitglieder den Gruppen beitreten, haben wir in der Telegram-Research-App außerdem eine Approximation der Beiträge auf Basis von Beitrittsnachrichten implementiert. Immer wenn ein neues Mitglied einem Chat beitrifft, wird eine automatische Nachricht in der Gruppe geschickt. In der Telegram-Research-App werden diese Nachrichten gezählt, sodass wöchentlich eine Übersicht entsteht, wie viele neue Mitglieder der Gruppe beigetreten sind. Darüber hinaus erlaubt diese Funktionalität auch, die Beiträge von Mitgliedern retrospektiv zu analysieren. In Abbildung 15 sind beispielhaft die Beiträge in 238 ukrainischsprachigen Telegram-Gruppen über eine Woche dargestellt. Diese Approximation unterschätzt die gesamte Zahl an neuen Mitgliedern, da viele, insbesondere große, Gruppen die Benachrichtigung automatisch löschen. Daher lässt sich mithilfe der Grafik zwar keine konkreten Aussagen über Zahlen machen, aber es ist dennoch möglich einen Trend zu erkennen.

5.6.2. Netzwerkanalyse Möglichkeiten

Die einzigartigen Identifikatoren der einzelnen User*innen in Telegram, erlaubt es über verschiedene Telegram-Gruppen Netzwerke zu betrachten. In Abbildung 16 ist ein Netzwerk-Graph dargestellt, der die Größe und die Konzentration der betrachteten ukrainischen Telegram-Gruppen verdeutlicht.

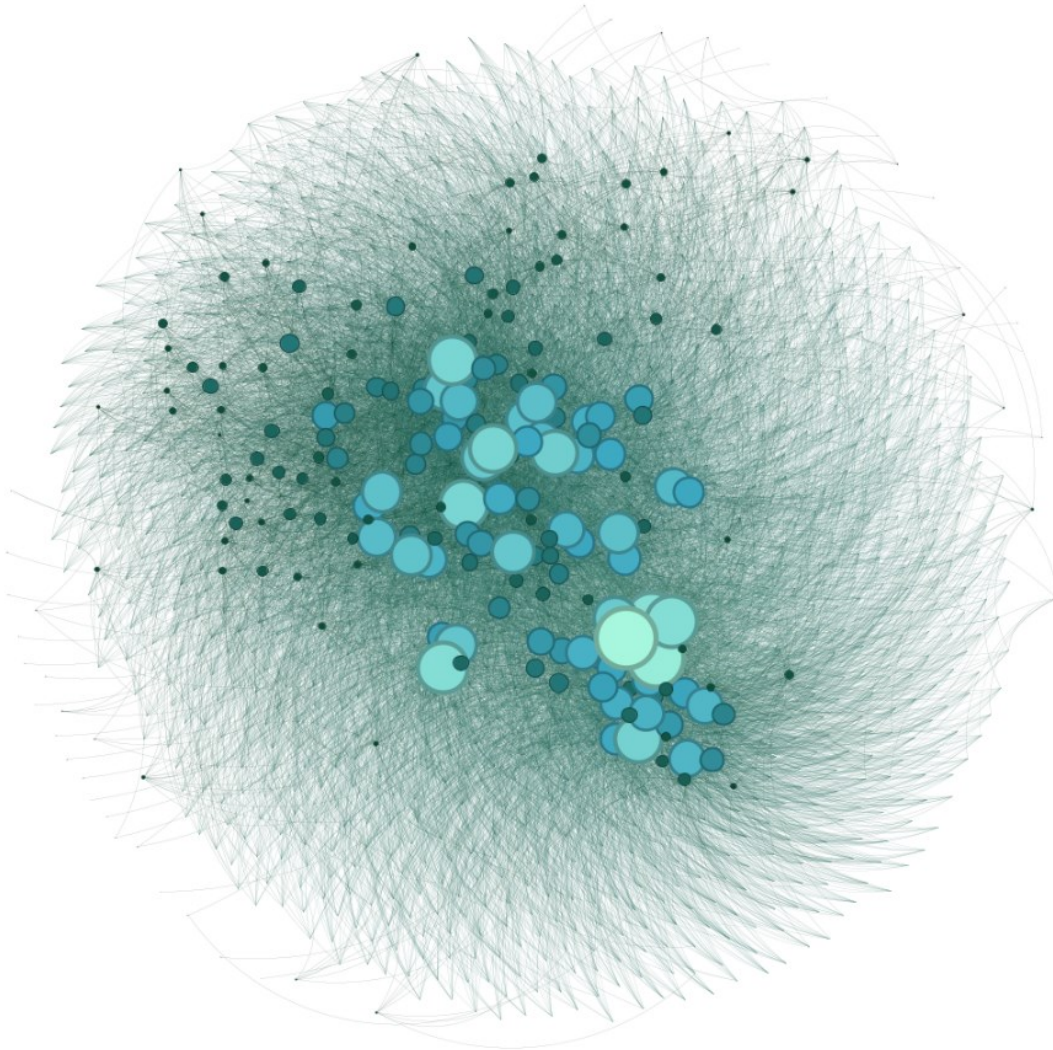


Abbildung 16: Netzwerk-Graph der analysierten ukrainischsprachigen Telegram-Gruppen mit ihren Mitgliedern

Die sichtbaren Knoten in diesem Netzwerk-Graphen sind Telegram-Gruppen. Desto größer eine Telegram-Gruppe ist, desto heller und größer erscheint der Knoten. In dieser Darstellung sind nur alle Gruppenmitglieder dargestellt, die in mindestens zehn betrachteten Telegram-Gruppen sind. Quelle: Telegram © Minor

Mithilfe von Netzwerk-Analysen kann beispielsweise beobachtet werden, ob über Zeit eine Konsolidierung stattfindet. Außerdem kann darüber graphisch dargestellt betrachtet werden, ob und wie sich das Größenverhältnis zwischen Gruppen verändert.

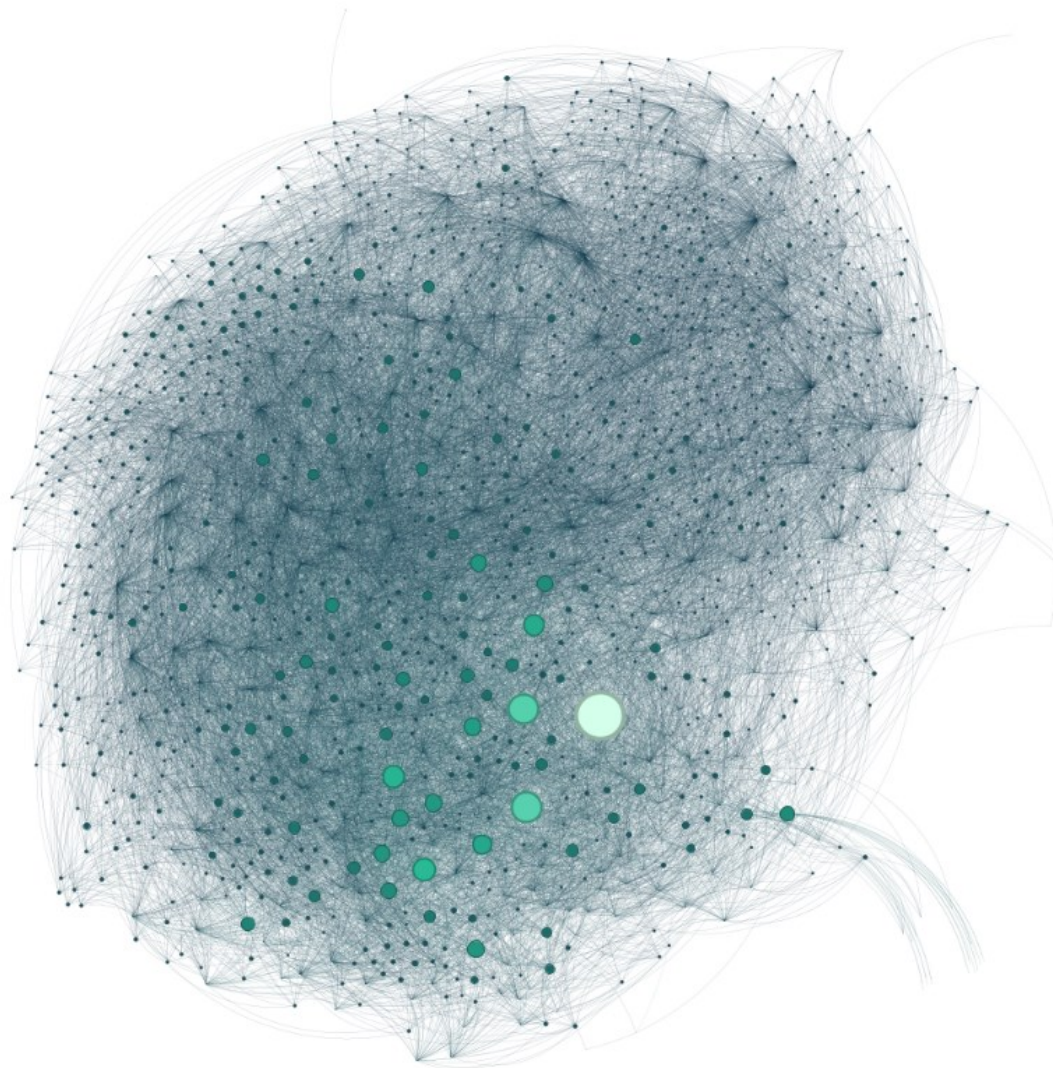


Abbildung 17: Netzwerk-Graph der Mitglieder ukrainischer Gruppen

Die sichtbaren Knoten in diesem Netzwerk-Graphen sind Mitglieder in Telegram-Gruppen. In Abhängigkeit der Anzahl an Gruppen, in denen ein*e Nutzer*in Mitglied ist, ist ein Knoten heller und größer. In dieser Darstellung sind nur alle Gruppenmitglieder dargestellt, die in mindestens zehn betrachteten Telegram-Gruppen sind. Quelle: Telegram © Minor

Abbildung 17 basiert auf denselben Daten wie Abbildung 16, jedoch ist der Fokus durch die Färbung und Größe der Knoten in dieser Instanz auf den Nutzer*innen und Mitglieder der Telegram-Gruppen. Der größte und hellste Knoten ist vermutlich einer der Accounts, die für die Analyse in der Telegram-Research-App verwendet wurden. Dennoch ist auffällig, dass es ein paar Nutzer*innen gibt, die sich in unzähligen Telegram-Gruppen befinden.

Grundsätzlich kann eine Netzwerk-Analyse insbesondere in Kombination mit anderen Daten, wie beispielsweise der Regionalität der Telegram-Gruppen, genutzt werden. Mithilfe dieser Daten können Bewegungen zwischen städtischen und ländlichen Gegenden nachvollzogen werden. Insgesamt weisen netzwerk-analytische Ansätze viel Potenzial auf, um gezielt dort Hilfe anzubieten, wo sie am meisten benötigt wird.

6. Fazit

Telegram kann mithilfe der Telegram-API als Datenquelle genutzt werden, um Migrations- und Fluchtbewegungen zu betrachten und damit einhergehende Informationsbedarfe von Migrant*innen und Geflüchteten, die Mitglieder in den Gruppen sind, zu erfassen und zu analysieren. Die hier vorgestellte Telegram-Research-App stellt eine Möglichkeit dar, wie die Telegram-API genutzt werden kann.

Die initialen exemplarischen Daten, auf Basis von Telegram-Gruppen für und von ukrainischen Geflüchteten in Deutschland, die im Zuge der FEM.OS Beratungsarbeit erfasst wurden, zeigen grundsätzlich weiteren wöchentlichen Zuwachs anhand von wöchentlichen Beitritten. Darüber hinaus ergibt die Stichwortanalyse Informationsbedarfe zu Ausweisdokumenten, Fortbewegungsmitteln, ärztliche Versorgung, Wohnung/Unterkunft und Schule. Eine Analyse der Genderverteilung innerhalb der analysierten Gruppen zeigt, dass fast alle Gruppen zu einem Großteil von weiblichen Mitgliedern genutzt werden.

Mithilfe einer Applikation, ähnlich wie der Telegram-Research-App, lassen sich also Datensätze erzeugen, die bei der Analyse von Informationsbedarfen bestimmter Gruppen sowie für ihre Beratung von großer Hilfe sein können. Die Anwendung kann zudem mit weiteren Entwicklungen kombiniert und so z.B. im Rahmen des *Natural Language Processing* (Berko et al 2011; Bird 2006) verfeinert werden. Außerdem kann potenziell mithilfe der Netzwerkdaten, die aus der Analyse verschiedener Gruppen hervorgehen, untersucht werden, ob und wie eine Konsolidierung der Gruppen stattfindet und inwieweit regionale Bezüge von Gruppen Auswirkungen hat.

Es gibt noch viele weitere Möglichkeiten, die Telegram-API als Datenquelle zu nutzen und damit Migrations- und Fluchtbewegungen genauer zu untersuchen. Eine Möglichkeit wäre beispielsweise, die Daten von Telegram-Gruppen von Geflüchteten und Migrant*innen in Deutschland aus verschiedenen Ländern und Regionen zu analysieren, um herauszufinden, welche Informationsbedarfe in verschiedenen Kontexten bestehen und wie sich diese Bedarfe im Laufe der Zeit verändern.

Ein weiteres Forschungsfeld könnte die Analyse von Emotionen und Stimmungen innerhalb der Gruppen sein. Hier könnte untersucht werden, welche Themen und Ereignisse bestimmte Emotionen auslösen und wie sich diese Emotionen im Laufe der Zeit verändern.

Eine weitere Möglichkeit wäre die Anwendung von Machine-Learning-Methoden, um automatisch bestimmte Muster und Zusammenhänge in den Daten zu erkennen. Hier könnte beispielsweise untersucht werden, ob sich bestimmte Wörter oder Phrasen häufig in bestimmten Kontexten wiederholen und ob es Zusammenhänge zwischen diesen Wörtern und bestimmten Ereignissen oder Themen gibt.

Insgesamt gibt es noch viele weitere Möglichkeiten, die Telegram-API als Datenquelle zu nutzen, um die Abbildung von Migrations- und Fluchtbewegungen in den Sozialen Medien genauer zu untersuchen und die Bedürfnisse von Migrantinnen und Geflüchteten, die sich auf diesen Plattformen manifestieren besser zu verstehen und zu adressieren.

Literaturverzeichnis

- BAMF - Bundesamt für Migration und Flüchtlinge. 2022: Aktuelle Zahlen (11/2022). <https://www.BAMF.de/SharedDocs/Anlagen/DE/Statistik/AsylinZahlen/aktuelle-zahlen-november-2022.html?nn=284722> (30.12.2022).
- Berko, Andrii, Yurii Matseliukh, Yurii Ivaniv, Lyubomyr Chyrun, und Vadim Schuchmann. 2021: The Text Classification Based on Big Data Analysis for Keyword Definition Using Stemming. In *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 1, 184–188.
- Bird, Steven. 2006: NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 69–72. Sydney, Australia: Association for Computational Linguistics <https://aclanthology.org/P06-4018>.
- Brücker, Herbert et al. 2022: *Geflüchtete aus der Ukraine in Deutschland: Flucht, Ankunft und Leben*. Bundesinstitut für Bevölkerungsforschung <http://www.bib.bund.de/ukraine-kurzbericht> (30.12.2022).
- DrKLO. 2022: DrKLO/Telegram. <https://github.com/DrKLO/Telegram> (30.12.2022).
- Gagarina, Natalia. 2014: Das Russische und das Ukrainische. In *Das mehrsprachige Klassenzimmer: Über die Muttersprachen unserer Schüler*, Hrsg. Manfred Krifka et al., 219–245. Berlin, Heidelberg: Springer.
- Gasymov, Ali. 2022: @mtproto/core. <https://github.com/alik0211/mtproto-core> (30.12.2022).
- git-scm. 2022: git-clone Documentation. <https://git-scm.com/docs/git-clone> (30.12.2022).
- Kerrisk, Michael. 2022: crontab - Linux manual page. <https://man7.org/linux/man-pages/man5/crontab.5.html> (30.12.2022).
- Minor-Kontor. 2023: Minor-Kontor/NexSM-unofficial-Telegram-Research-App. <https://github.com/Minor-Kontor/NexSM-unofficial-Telegram-Research-App> (14.2.2023).
- NLTK. 2022: NLTK :: Natural Language Toolkit. <https://www.nltk.org/> (30.12.2022).
- Parshin, Stas. 2022: Java Telegram Bot API. <https://github.com/pengrad/java-telegram-bot-api> (30.12.2022).
- Pfeffer-Hoffmann, Christian. 2022: Migrationsberatung in sozialen Medien. *Minor*, November <https://minor-kontor.de/migrationsberatung-in-sozialen-medien/> (30.12.2022).
- Python. 2022a: asyncio - Asynchronous I/O. <https://docs.python.org/3/library/asyncio.html> (30.12.2022).
- Python. 2022b: Download Python. <https://www.python.org/downloads/> (30.12.2022).
- python-telegram-bot. 2022: python-telegram-bot v20.0. <https://docs.python-telegram-bot.org/en/stable/> (30.12.2022).
- Schulz-Kersting, Odette, Tobias Stapf, und Taissiya Sutormina. 2022: Digital Streetwork für und mit ukrainischen Geflüchteten. April <https://minor-kontor.de/digital-streetwork-fuer-und-mit-ukrainischen-gefluechteten/> (30.12.2022).

Statista. 2022: Worldwide Social Network Users 2017-2027. <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/> (30.12.2022).

TDLib. 2022: TDLib. <https://github.com/tdlib/td> (30.12.2022).

Telegram. 2022a: 700 Million Users. <https://telegram.org/blog/700-million-and-premium> (30.12.2022).

Telegram. 2022b: Creating your Telegram Application. https://core.telegram.org/api/obtaining_api_id (30.12.2022).

Telegram. 2022c: My Telegram. <https://my.telegram.org/auth> (30.12.2022).

Telegram. 2022d: Telegram Privacy Policy. <https://telegram.org/privacy> (30.12.2022).

Telegram. 2022e: Telegram ToS. <https://telegram.org/tos> (30.12.2022).

Telegram. 2022f: Telegram-API auth.sendCode. <https://core.telegram.org/method/auth.sendCode> (30.12.2022).

Telegram. 2022g: Telegram-API Getting Started. <https://core.telegram.org/api#getting-started> (30.12.2022).

Telegram. 2022h: Telegram-API Methods. <https://core.telegram.org/methods> (30.12.2022).

Telegram. 2022i: Telegram-API ToS. <https://core.telegram.org/api/terms> (30.12.2022).

Telegram. 2022j: Telegram-APIs. <https://core.telegram.org/> (30.12.2022).

Telethon. 2022a: Telethon API. <https://tl.telethon.dev/> (30.12.2022).

Telethon. 2022b: Telethon's Documentation. <https://docs.telethon.dev/en/stable/> (30.12.2022).

Tunjić, Davor, Lia Zlateva, und Anna Bielicka. 2022: 4 Jahre Digital Streetwork für EU-Zugewanderte. *Minor*, Juli <https://minor-kontor.de/4-jahre-digital-streetwork-fuer-eu-zugewanderte/> (30.12.2022).

Abbildungsverzeichnis

Abbildung 1: Histogramm der Größe von Gruppen	13
Abbildung 2: Histogramm der Anzahl an verschickten Nachrichten in einer Woche	14
Abbildung 3: Vereinfachte Architektur der Telegram-Research-App	16
Abbildung 4: Beispiel der Ordner-Struktur der Benutzereingaben	18
Abbildung 5: Beispiel der Ordner-Struktur der Programmausgaben	20
Abbildung 6: Beitritte in untersuchten Gruppen über eine Woche	22
Abbildung 7: Nachrichtenverteilung nach Wochentag	23

Abbildung 8: Nachrichtenverteilung nach Uhrzeit	24
Abbildung 9: Download der Telegram-Research-App von Github (Minor-Kontor 2023).....	25
Abbildung 10: Vergleich verschiedener Aktivitäten nach Wochentag.....	33
Abbildung 11: Vergleich verschiedener Aktivitäten nach Uhrzeit.....	34
Abbildung 12: Gesamtverteilung des Gender über alle Gruppen	35
Abbildung 13: Verhältnis zwischen männlich und weiblich gelesenen Vornamen	36
Abbildung 14: Verhältnis der Gender zwischen den Gruppen	36
Abbildung 15: Beitritte in untersuchten Gruppen über eine Woche.....	38
Abbildung 16: Netzwerk-Graph der ukrainischen Gruppen mit ihren Mitgliedern	39
Abbildung 17: Netzwerk-Graph der Mitglieder ukrainischer Gruppen.....	40

Codeverzeichnis

Code 1: Authentifizierung und Anmeldung mit <i>Telethon</i>	6
Code 2: Dialog Anfrage - Anfrage einer Liste mit allen Chats, Channels und Gruppen für einen spezifischen Account.	6
Code 3: Dialog-Objekt Beispiel - Beispielhaftes Dialog-Objekt als eines der Elemente der Antwort auf eine <i>get_dialogs()</i> -Anfrage.	7
Code 4: Beispielhafte Mitglieder-Anfrage - Anfrage einer Liste mit allen Mitgliedern eines Chats. .	7
Code 5: Participant-Objekt Beispiel - Beispielhaftes <i>Participant</i> -Objekt als eines der Elemente der Antwort auf eine <i>get_participants(dialog)</i> -Anfrage.....	8
Code 6: Beispielhafte Nachrichten-Anfrage - Anfrage einer Liste mit den 10 letzten Nachrichten eines Chats.....	9
Code 7: Beispielhafte komplexe Anfragen - Anfrage einer Liste mit den fünf letzten GIFs vor Ende des Jahres 2022 eines Chats und Anfrage der letzten 14 Nachrichten eines spezifischen Mitglieds im Jahr 2022.	9
Code 8: Message-Objekt Beispiel - Beispielhaftes <i>Message</i> -Objekt als eines der Elemente der Antwort auf eine <i>TelegramClient.get_messages(dialog)</i> -Anfrage.....	10

Tabellenverzeichnis

Tabelle 1: Beispiel einer Einweg-Hashing-Transformation	12
---	----

Tabelle 2: Die acht größten ukrainischsprachigen Telegram-Gruppen nach Mitgliedern	14
Tabelle 3: Auswahl an ignorierten Wörtern	29
Tabelle 4: Auswahl von 22 für die Beratungsarbeit relevanten Wörter	29
Tabelle 5: Häufig verwendete Wörter für die Beratungsarbeit	30
Tabelle 6: Häufig verwendete Wortpaare	32

Impressum

Minor –Projektkontor für Bildung und Forschung gGmbH 2022

Text und Daten:
Hanno Skowronek

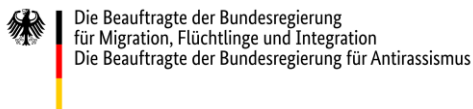
Fem.OS – Aufsuchendes Orientierungs- und Beratungs-System in den sozialen Medien für Migrantinnen ist ein Projekt von



Alt-Reinickendorf 25
13407 Berlin
Tel.: +49 30 – 457989500
E-Mail: minor@minor-kontor.de

www.minor-kontor.de
www.minor-kontor.de/fem-os/

Gefördert von



In Kooperation mit:



© Minor – Projektkontor für Bildung und Forschung