Richtlinie zum „Software-Sprint“

ARST – Analyse- und Recherche-Software für Telegram

Schlussbericht

Zuwendungsempfänger:

Grischa Stanjek und Gregor Weichbrodt GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S33 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Unsere Motivation war und ist, eine Software zu entwickeln, die eine Untersuchung und Auswertung öffentlicher Inhalte auf dem Messenger Telegram vereinfacht. Denn trotz seiner Popularität gab es für Telegram bis dato keine frei verfügbaren Werkzeuge. Bisher war dies nur mit hohem technischen Aufwand und Wissen möglich.

Die wichtigsten Meilensteine bei diesem Projekt waren:

1. **Scraper**
   Als Erstes wurde ein Programm (*Scraper)* entwickelt, der Telegram-Inhalte in regelmäßigen Zeitabständen herunterlädt und archiviert. Zusätzlich werden die Inhalte für quantitative Untersuchungen aufbereitet und verarbeitet (statistische Daten, Entwicklung von Trends, Wachstumsrate in Telegram-Kanälen u. v. m.), bevor sie in der Datenbank gespeichert werden.
2. **API**
   Als Nächstes wurde eine *API* (*Programmierschnittstelle*) entwickelt, über die die Inhalte in der Datenbank programmatisch abgerufen werden können. Eine API bietet mehrere Vorteile, z.B. dass man Anwender*innen nicht vorschreibt, mit welcher Software die Daten abzurufen sind. Wer an die Daten muss für maschinelle Auswertungen, braucht keine Weboberfläche. Wer eine Weboberfläche braucht, nutzt unser Frontend.
3. **Frontend**
   Das Frontend ist die Weboberfläche, die man im Browser ansteuern kann. Es kommuniziert mit der API und verfügt über eine Benutzerverwaltung, ein Zugang ist daher nur mit Login möglich. Hierüber lassen sich die Inhalte aus der Datenbank einsehen, durch Telegram-Kanäle navigieren, Inhalte suchen und recherchieren, Statistiken einsehen u. v. m.

Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts? Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?

Die Software richtet sich an Journalist*innen, Forscher*innen und zivilgesellschaftliche Akteur*innen, die z. B. die Verbreitung von rechtsextremen und verschwörungsideologischen Inhalten auf Telegram untersuchen (wollen). Der Code dieses Projektes ist öffentlich einsehbar und die Entwicklung findet öffentlich statt. Interessierte Personen können die Software unter Anleitung selbst aufsetzen oder sich an der Entwicklung beteiligen.

Sowohl die Bedienung als auch die Installation soll dabei für Personen ohne tiefgehendes technisches Verständnis durch eine Weboberfläche möglich sein. Gleichzeitig sollen Entwickler*innen die Möglichkeit haben, einzelne Funktionen je nach Bedarf nachzurüsten bzw. den Funktionsumfang der Software zu erweitern.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Ein oder mehrere Telegram-Accounts können mit der Software verknüpft werden. Anschließend werden die Inhalte in regelmäßigen Abständen heruntergeladen und verarbeitet. Mithilfe einer Weboberfläche (Frontend) lassen sich übergreifend alle Kanäle, Gruppen und Chats durchsuchen. Möchte man die Suche einschränken, kann man mit verschiedenen Parametern die Suche feinjustieren. Hier können bspw. Nachrichten aus einem bestimmten Zeitraum in bestimmten Kanälen gesucht werden. Oder Nachrichten von bestimmten Telegram-Nutzer*innen.
Mittels automatisierter Texterkennung wird Text auf Bildern erkannt und gespeichert. Darüber hinaus werden Sprachnachrichten automatisiert im Hintergrund transkribiert und als durchsuchbarer Text gespeichert. So werden Inhalte aus Medien durchsuchbar, die ursprünglich nicht in Textform vorlagen.
Die Software erhebt regelmäßig statistische Daten zur Aktivität und zum Wachstum von Kanälen und Gruppen, wodurch quantitative Analysen ermöglicht werden. So lassen sich beispielsweise Trends frühzeitig erkennen und der Zuwachs von Nutzer*innen im Verlauf der Zeit abbilden.
Medien wie Videos, Fotos und Sprachnachrichten können automatisiert heruntergeladen und in einem Cloud-Storage abgelegt werden. Die Weboberfläche zeigt alle Medien als Vorschau an und stellt diese für den Download bereit.
Alle erhobenen Daten sind auch über eine Programmierschnittstelle (API) abrufbar. Die API-Endpunkte sind mit verschiedenen Parametern ausgestattet, mit denen alle Inhalte gefiltert ausgegeben werden können. Die REST-API eignet sich z. B. für eine Weiterverarbeitung der Inhalte durch Software Dritter. Alle API-Endpunkte können im Browser mittels *Swagger* getestet werden. Darüber hinaus können alle Datensätze mittels Shell-Skripte als CSV oder JSON exportiert werden.

Bis auf **Frontend** konnten alle Meilensteine vollständig erreicht werden. Es gab ein paar Faktoren, die uns die Planung erschwert haben, weshalb die Entwicklung von **Scraper** länger gedauert hat, als wir geplant haben (siehe unten). Zudem blieb keine Zeit mehr für eine ausführliche Dokumentation.

Das Projekt ist mit dem Abschluss der Förderphase für uns aber nicht abgeschlossen und wir arbeiten nach wie vor daran. Ziel ist die Kooperation mit anderen Institutionen sowie eine Anschlussförderung. Dass das Projekt gefragt ist, erfahren wir bereits durch Zuschriften verschiedener Organisationen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weiter-gehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung? Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Ein großer Nutzen für die Zielgruppe besteht darin, dass die Beschaffung von Telegram-Inhalten nicht mehr an die Kompetenz und das Equipment einzelner weniger Menschen gebunden ist, die ihre Methoden aus diversen Gründen unter Verschluss halten. Da das Projekt Open Source ist, sind die Methoden zur Beschaffung und Auswertung von Inhalten transparent, ebenso die Weiterentwicklung. In Gesprächen mit anderen Organisationen und Einzelpersonen sind bereits Ideen für die Weiterentwicklung entstanden und sollen in der Zukunft umgesetzt werden. Einzelne Organisationen wollen die Software nutzen und ausführlich testen. Durch den Austausch mit anderen Entwickler*innen soll die Software verbessert werden und der Funktionsumfang ausgebaut werden. Die Veröffentlichung als Open-Source-Software ermöglicht es gemeinnützigen bzw. nicht profitorientierten Organisationen und Einzelpersonen mit wenig oder keinem Budget die Software einsetzen zu können.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hinter-gründe, und wie bist Du alternativ vorgegangen?

Einzelne Features wurden zurückgestellt, da ihre Implementierung aus unterschiedlichen Gründen schlecht umzusetzen war. Auf die Auswertung von Video-Inhalten wurde vorerst verzichtet, da dies zusätzliche Ressourcen wie Rechenleistung und Speicherplatz benötigt, die der Zielgruppe in der Regel nicht zur Verfügung stehen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GHitHub, Veröffentlichungen)?

Im GitHub-Repository https://github.com/democ-de/teledash wird die Funktionsweise der Software erläutert und auf die Repositories der einzelnen Bestandteile wie das Frontend bzw. Backend verlinkt. Dort ist der Quellcode abrufbar und die Konfiguration dokumentiert.
In einem deutschsprachigen Blogbeitrag auf der Seite des Protype Funds wird die Funktionsweise niedrigschwellig erklärt: https://demoweek.prototypefund.de/projects/25-teledash.html
Im Rahmen der Eröffnungsveranstaltung für die "DEMO WEEK Runde 10" zum Abschluss der 10. Förderrunde des Prototype Fund wurde in einer Videoübertragung die Software veranschaulicht und demonstriert: https://www.youtube.com/watch?v=WfzpYk8nsn8

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Einzelne Teilaufgaben waren komplexer als in der Planungsphase angenommen. So wurde die Umsetzung einzelner Bestandteile zeitaufwändiger als geplant. Beispielsweise musste die fehlende Dokumentation der Telegram-API insbesondere bezüglich Rate-Limits durch ausführliche Tests kompensiert werden. Um Zugriffssperren zu verhindern, mussten Vorrichtungen geschaffen werden, um auf Rate-Limits zu erkennen und Abfragen ggf. zu verlangsamen oder zu wiederholen.

Aufgaben wie Text- und Spracherkennung waren weitaus ressourcenaufwendiger, als geplant, da u. a. der Umfang unterschätzt wurde, was Implikationen für die Performance und notwendigen Hardware-Ressourcen hatte. Nach ausführlichen Tests bestehender Software-Bibliotheken für diese Aufgaben wurden die performantesten ausgewählt. Die zu untersuchenden Medieninhalte waren heterogener als angenommen. Dies erschwerte die automatisierte Erkennung von Inhalten dieser Medien wie Sprache und Text.

Unregelmäßig auftretende Verbindungsabbrüche in der Kommunikation mit der Telegram API führten zu Verzögerung durch langwieriges Debugging und zu Umstrukturierungen einzelner Software-Bestandteile. Darüber hinaus trat immer wieder unregelmäßiges Verhalten in der Kommunikation mit der API auf, speziell beim Abfragen von Gruppen mit einer hohen Anzahl von Teilnehmenden. Hier wurden vereinzelte Anfragen zu einzelnen Inhalten unregelmäßig und nicht nachvollziehbar abgewiesen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Es gab keine nennenswerten Einflüsse durch andere Institutionen, die den Verlauf des Projekts maßgeblich beeinflusst haben, da es bisher keine vergleichbaren Software-Lösungen gibt, die öffentlich zugänglich sind. Im Verlauf des Projektes wurden Wissenschaftler*innen und Journalist*innen bezüglich des Bedarfs einzelner Funktionen befragt, die unsere Vorannahmen bestätigt haben.