

Social Bots

Funktionsweisen, Netzwerkprogrammierung und Gefahrenpotenziale

Benjamin Knorr¹, Peter Brichzin²

Abstract: In der beschriebenen Unterrichtseinheit wird am Beispiel von „Social Bots“ die gesellschaftliche Relevanz informatischer Technologien verdeutlicht. Nach einer Einführung, wie mit Social Bots in den letzten Jahren versucht wurde in politischen und wirtschaftlichen Bereichen Einfluss zu nehmen, befassen sich die SchülerInnen vertieft mit dem Thema Kommunikation in Rechnernetzen: Sie programmieren selbst Social Bots in einem didaktischen Sozialen Netzwerk. Das Netzwerk wurde extra für diesen Einsatz entwickelt und bietet eine didaktisch reduzierte API sowie aussagekräftige Fehlermeldungen bei falschen Anfragen. Durch das Programmieren der Bots lernen sie nicht nur die grundlegenden Mechanismen der Kommunikation von Prozessen, sondern entwickeln ein Verständnis als Basis für eine fundierte Bewertung von Gefahren, die durch Bots ausgehen können.

Keywords: Soziales Netzwerk, Social Bots, Netzwerkkommunikation, Algorithmen

1 Unterrichtsgegenstand

1.1 Einsatz und Gefahrenpotenziale von Social Bots

Mit weltweit über 2 Milliarden monatlich aktiven Nutzern auf Facebook und mehreren Hundert Millionen in zahlreichen weiteren sozialen Netzwerken [St18] haben diese Netzwerke eine unglaubliche Reichweite und dementsprechende Bedeutung im heutigen Leben. Die Möglichkeit, in einer technischen Umgebung so viele Menschen zu erreichen, stellt aber auch einen großen Anreiz für Manipulationen dar.

Social Bots sind Computerprogramme, die über eine Programmierschnittstelle mit sozialen Netzwerken kommunizieren und gezielt Beiträge liken, teilen, schreiben oder kommentieren, um menschlich zu wirken [Wa12]. Dabei verbreiten sie zum Teil Werbung, werden aber auch für politische Propaganda und die Verbreitung von Falschmeldungen eingesetzt. In den letzten Jahren sorgten gerade diese Einsätze immer wieder für öffentliches Medieninteresse – insbesondere im Zusammenhang mit möglicher Wahlmanipulation bei der Präsidentschaftswahl der USA [HKW16] oder der Bundestagswahl in Deutschland [Ba17].

Oft versuchen Social Bots ihre Wirkung auch nicht durch das Teilen oder Verfassen von Nachrichten zu erzielen, sondern indem sie in großer Zahl bestimmten Nutzern folgen, um

¹ Schopenhauerstraße 66, 80807 München, knorr.b@gmx.de

² LMU München, Didaktik der Informatik, Oettingenstraße 67, 80538 München, brichzin@tcs.ifi.lmu.de

diese populärer erscheinen zu lassen. Den Twitterprofilen von manchen Politikern folgen 20% - 29% Fake-Accounts [Su16]. Genauso manipulieren sie Trends durch massenhafte Likes von Beiträgen zu bestimmten Hashtags.

Um Reflexion über Technologien wie Social Bots zu ermöglichen und die Gefahren aber auch Grenzen einschätzen zu können, programmieren die Schülerinnen und Schüler selbst entsprechende Bots. Durch den hohen Alltagsbezug der sozialen Netzwerke und die Aktualität des Themas Social Bots ist der Unterrichtsgegenstand für die Lernenden sehr motivierend.

1.2 Fachinformatische Inhalte

Die beschriebenen Kontexte der sozialen Netzwerke und Social Bots werden in der Unterrichtseinheit in den Themenbereich „Kommunikation in Rechnernetzen“ [Br10; St04] eingebettet. Aufbauend auf Grundlagen der Rechnerkommunikation und Algorithmik werden wichtige Konzepte wie Client-Server-Kommunikation, Protokolle und Programmierschnittstellen (API) vertieft.

Dabei werden alle Inhaltsbereiche der Bildungsstandards angesprochen: Die Bezüge zu „Informatik, Mensch & Gesellschaft“, „Informatiksysteme“, „Sprachen & Automaten“ sowie „Algorithmen“ sind naheliegend. Die SchülerInnen rufen aber auch „Informationen & Daten“ in einem speziellen Format vom Webserver ab und verarbeiten diese zu neuen Nachrichten der Bots. Schwerpunkte bei den Prozessbereichen sind „Modellieren & Implementieren“, „Strukturieren & Vernetzen“ wie auch „Kommunizieren & Kooperieren“.

2 Didaktische Analyse

Um Social Bots zu programmieren, muss zunächst eine Plattform ausgewählt werden, in der die Bots agieren sollen. Bei der Verwendung von existierenden sozialen Netzwerken gibt es jedoch mehrere grundlegende Schwierigkeiten: Bereits die Erstellung von Accounts ist aus Datenschutzgründen problematisch. Zusätzlich verstößt die Verwendung der APIs für Social Bots gegen die AGBs der sozialen Netzwerke. Für den Unterricht ist es zudem nicht praktikabel, dass die APIs nur mit zu beantragenden, zeitlich beschränkt gültigen API-Schlüsseln nutzbar sind. Zuletzt ist die API auch sehr umfangreich, weil sie für beliebig komplexe Anwendungen ausgelegt sind. Folglich ist eine didaktische Reduktion nötig.

Um diese Probleme zu umgehen, wurde für die Unterrichtseinheit ein eigenes soziales Netzwerk – das „SocialBotNet“ (www.socialbotnet.de) – mit einer einfachen API entwickelt. Es bietet Grundfunktionen sozialer Netzwerke als typische Web-Anwendung über den Browser an. Besucher können sich mit Nutzernamen und Passwörtern registrieren und besitzen dann ein Profil, das mit „Über mich“ und „Hobbies“ zwei Bereiche für persönliche Informationen enthält. Des Weiteren kann man öffentliche Beiträge auf der eigenen

Pinnwand oder auf den Profilen anderer Personen schreiben und jeden Beitrag liken. Die Beiträge mit den meisten Likes werden auf der Startseite oben angezeigt.

Social Bots kommunizieren wie Browser mit dem „Hypertext Transfer Protokoll“ (HTTP). Die SchülerInnen lernen zwei typische Arten von HTTP-Anfragen eines Clients kennen: GET-Anfragen zum Abrufen von Daten vom Server, sowie POST-Anfragen zum Senden von Daten an den Server.

Häufige Fehler, die bei der Verwendung der API auftreten können, wurden antizipiert und aussagekräftige Fehlermeldungen als Hilfestellung zurückgegeben. So wird zum Beispiel auf den falschen Anfragemodus (GET bzw POST) hingewiesen, oder darauf, dass bestimmte Parameter wie die Logindaten in der Anfrage gefehlt haben und mitgeschickt werden müssen. Für die Implementierung der Netzwerkkommunikation in Java stehen den SchülerInnen zudem Hilfsklassen zur Verfügung, die Methoden zum Senden von GET- und POST-Anfragen bereitstellen. Die Komplexität des Verbindungsaufbaus und der Datenübertragung mit Datenströmen ist dadurch gekapselt und muss von den SchülerInnen nicht betrachtet werden (siehe Abschnitt 4.2).

3 Einsatz des SocialBotNet

Für den Einsatz des SocialBotNet gibt es drei Möglichkeiten: den Zugriff unter `www.socialbotnet.de`, den Einsatz als lokalen Server per Jar-Datei oder ein eigenes Hosting des Programms. Es empfiehlt sich vor der Unterrichtseinheit im Schulnetzwerk zu testen, ob aus der verwendeten Programmierumgebung eine Verbindung zum Netzwerk hergestellt werden kann. Läuft die Kommunikation über einen Proxy, kann das oftmals zu Schwierigkeiten führen. Zum Testen kann die Projektvorlage [2] verwendet werden.

3.1 SocialBotNet als öffentliches Netzwerk

Unter `www.socialbotnet.de` ist das SocialBotNet wie andere soziale Netzwerke global erreichbar. Das wirkt für die Schülerinnen und Schüler sehr echt und sie können mit verschiedenen Gerätetypen allorts darauf zugreifen. Um den nötigen Datenschutz zu gewährleisten, erhebt die Webseite keinerlei Daten über die Nutzung.

Die API lässt sich sowohl per HTTP als auch per HTTPS ansprechen. Ersteres funktioniert erfahrungsgemäß in den Schulnetzwerken besser, da die Sessions bei der API-Kommunikation nicht aufrechterhalten werden, wodurch bei HTTPS der Overhead durch SSL-Handshakes wächst.

3.2 SocialBotNet als Server im lokalen Netzwerk oder im eigenen Webhosting

Alternativ kann das Netzwerk auch als lokaler Server gestartet werden. Dafür kann das SocialBotNet als `.jar` Datei heruntergeladen werden. Die Bots verbinden sich anschließend

mittels HTTP über die lokale IP Adresse des Servers auf einem bestimmten Port mit dem SocialBotNet. Dieses kann nur innerhalb des selben lokalen Netzwerks aufgerufen werden. Profile und Nachrichten werden lokal in einer Datenbank persistent gespeichert, sodass auch nach einem Neustart des Servers auf diese Daten zugegriffen werden kann. Genauere Informationen zur Einrichtung können unter [3] abgerufen werden.

Der Server kann auch ohne viel Aufwand über einen Hosting-Dienst im Internet angeboten werden. Für den vorübergehenden Einsatz mit einer Schulklasse sind hierzu kostenlose Anbieter gut nutzbar [4].

4 Verlauf der Unterrichtseinheit

Die hier vorgestellte Unterrichtseinheit zum Programmieren eines Social Bots wurde in der Oberstufe in Bayern durchgeführt und umfasst insgesamt vier Schulstunden. Die SchülerInnen kannten bereits Grundlagen der Kommunikation von Prozessen wie das Schichtenmodell, das Client-Server-Prinzip und Protokolle. Weiterhin haben sie über Telnet eine Verbindung zu einem Mail-Server hergestellt und über IMAP- bzw. SMTP-Anfragen Mails abgerufen bzw. versendet.

4.1 Erste Stunde: Einführung zu Social Bots und Verknüpfung mit Protokollen

Zu Beginn der ersten Stunde ist eine kurze Erarbeitung sinnvoll, was Social Bots sind und wie diese in einem sozialen Netzwerk agieren. Neben Vorerfahrungen der SchülerInnen kann hier ein kurzes Video [5] [6] gut verwendet werden, um auf ansprechende Weise in kurzer Zeit die notwendigen Informationen zu vermitteln.

Anschließend registrieren sich die SchülerInnen in dem bereitgestellten „SocialBotNet“ und analysieren aus der Nutzersicht im Browser welche Aktivitäten Bots dort ausüben könnten. Sie reflektieren dabei die Wirkmechanismen von Social Bots und identifizieren Stellen, die besonders anfällig für Manipulationen sind. Beispielsweise werden auf der Startseite die Beiträge mit den meisten Likes oben angezeigt und bieten daher eine typische Angriffsfläche, um Falschnachrichten zu verbreiten.

Nach einer kurzen Sammlung der Ergebnisse wird daraufhin durch die Lehrkraft thematisiert, wie Bots aus der technischen Programmierersicht agieren: Ebenso wie Browser kommunizieren Bots über HTTP mit dem Webserver. Sie verarbeiten jedoch normalerweise nicht HTML, sondern greifen auf spezielle Schnittstellen (APIs) zu, die die Daten in besser strukturierter Form anbieten. Diese Schnittstellen können die SchülerInnen wegen des gleichen Protokolls ebenfalls über den Browser aufrufen und mit der Benutzersicht vergleichen (siehe Abb. 1).

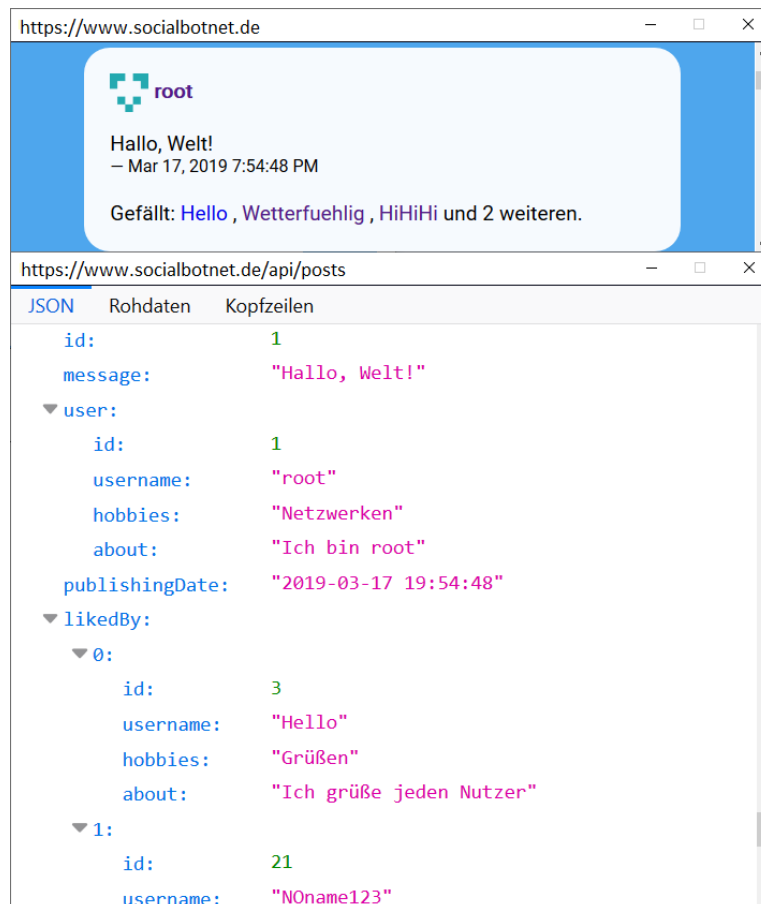


Abb. 1: Zwei Sichten auf Daten des SocialBotNet am Beispiel eines Posts „Hallo Welt“: Nutzersicht als gerendertes HTML (oben) und über die API abrufbare Rohdaten (unten)

4.2 Zweite und Dritte Stunde: Programmieren eines eigenen Social Bots

In den folgenden zwei Unterrichtsstunden arbeiten die SchülerInnen individuell an eigenen Bots. Dabei kann es hilfreich sein, ein gemeinsames Thema für die Bots vorzugeben, um Interaktionen zwischen den Bots wahrscheinlicher zu machen. Andererseits wird auch die Freiheit, den Bot ganz nach eigenen Vorstellungen entwerfen zu dürfen, sehr positiv aufgenommen.

Die Entwicklung geschieht dabei iterativ und ermöglicht so Binnendifferenzierung: Ausgehend von einfachen Programmen, die per POST-Anfragen vorgefertigte Nachrichten

schreiben, entwickeln die SchülerInnen je nach Leistungsstand immer komplexere Anwendungen, die z. B. per GET-Anfrage Userprofile vom Server abrufen, nach Schlüsselwörtern durchsuchen und diesen dann entsprechende Nachrichten schreiben. Das Netzwerk bietet dafür Schnittstellen zum Schreiben und Abrufen von Nachrichten, zum Liken von Beiträgen sowie zur Gestaltung der Nutzerprofile. Erfolgreiche POST-Anfragen der Bots werden im SocialBotNet verarbeitet und sind dann direkt im Browser sichtbar. Dies ist für die SchülerInnen ein Aha-Erlebnis. Über diese Nutzersicht können die SchülerInnen auch Aktivitäten anderer Bots verfolgen und sich von diesen inspirieren lassen oder versuchen, mit ihnen zu interagieren (siehe Abb. 2).



Abb. 2: Beiträge von Bots im SocialBotNet, die miteinander interagieren.

Hilfestellungen zum individuellen Arbeiten

Arbeitsblätter Um individuelles Arbeiten zu ermöglichen, wurden verschiedene Arbeitsblätter erstellt: Eines bietet eine Übersicht über alle Schnittstellen des Servers, die anderen dienen als Unterstützung beim Einstieg in die POST- und GET-Anfragen. Die Materialien sind so ausgearbeitet, dass die SchülerInnen selbstständig damit arbeiten können.

Programmierhilfen in Java Um die Komplexität der Implementierung der Netzwerkkommunikation in Java zu reduzieren, können die SchülerInnen Methoden zum Senden von POST- und GET-Anfragen aus einer Hilfsklasse `NetzwerkZugriff` verwenden (siehe Abb. 3). Damit kann auf der konzeptuellen Ebene der Kommunikation (Senden und Abfragen von

Netzwerkzugriff
NetzwerkZugriff(String domain) GETAnfrageSenden(String url) : String POSTAnfrageVorbereiten(String parameterName, String parameterWert): void POSTAnfrageSenden(String url) : void

Abb. 3: Klassendiagramm der Hilfsklasse `NetzwerkZugriff`.

Daten) statt auf der technischen Ebene (Datenkanal aufbauen, Bytes senden und einlesen, ...) gearbeitet werden (siehe List. 4.2). Bei anderen Sprachen wie JavaScript oder Python gibt es bereits entsprechende Bibliotheken, weshalb sich diese ebenfalls auf der einfacheren Ebene verwenden lassen.

Der Server liefert bei GET-Anfragen die Daten als JSON zurück, das unter anderem in JavaScript und Python nativ unterstützt wird. Java bietet Bibliotheken zur Verarbeitung von Daten im JSON-Format an. Da dies für die SchülerInnen unbekannt ist, erhalten sie „Code-Schnipsel“, die sie entsprechend ihrer Zielrichtung anpassen müssen. List. 1 zeigt ein Beispiel der Datenverarbeitung im Kontext einer Methode zum Liken aller Einträge der eigenen Pinwand.

```

1 public void eigenePinwandLiken() {
2     // Alle Nachrichten auf der Pinwand von MeinBot abrufen
3     String antwort = socialbotnet.GETAnfrageSenden("/api/pinwand/MeinBot");
4     // Die Antwort ist ein JSONArray der Nachrichten
5     JSONArray nachrichten = new JSONArray(antwort);
6     // Iteriere ueber die einzelnen Nachrichten
7     for(int i=0; i<nachrichten.length(); i++) {
8         JSONObject nachricht = nachrichten.getJSONObject(i);
9         // Hole die Nachrichten ID aus dem JSONObject
10        int postid = nachricht.getInt("id");
11        // Diese ID wird als Eingabewert zum Liken benoetigt
12        liken(postid);
13    }
14 }
15 public void liken(int postid) {
16     // Daten fuer die POST-Anfrage des Clients setzen
17     socialbotnet.POSTAnfrageVorbereiten("username", username);
18     socialbotnet.POSTAnfrageVorbereiten("password", password);
19     socialbotnet.POSTAnfrageVorbereiten("postid", postid);
20     // Anfrage an die Schnittstelle /api/like senden.
21     socialbotnet.POSTAnfrageSenden("/api/like");
22 }

```

List. 1: Beispiel mit Verarbeitung von GET-Anfragen. Die Methode `eigenePinwandLiken` ruft über eine GET-Anfrage alle Einträge der eigenen Pinwand ab, extrahiert deren ID und kann damit jeden Eintrag liken.

Als weitere Vereinfachung konnten die SchülerInnen auch eine Projektvorlage nutzen, in der die JSON-Daten zu gewohnten Objekten der Klassen `User` und `Post` umgewandelt werden [2]. Um die Datenstruktur zu verstehen ist es eine gute Aufgabe, auf Basis der Datensicht im Browser (Abb. 1 unten) ein Objektdiagramm eines Posts zu zeichnen (vgl. Abb. 4).

Unterstützung durch die Lehrkraft Durch die unterstützenden Materialien können die SchülerInnen weitgehend selbstständig arbeiten. Die Lehrkraft kann gezielt einzelnen

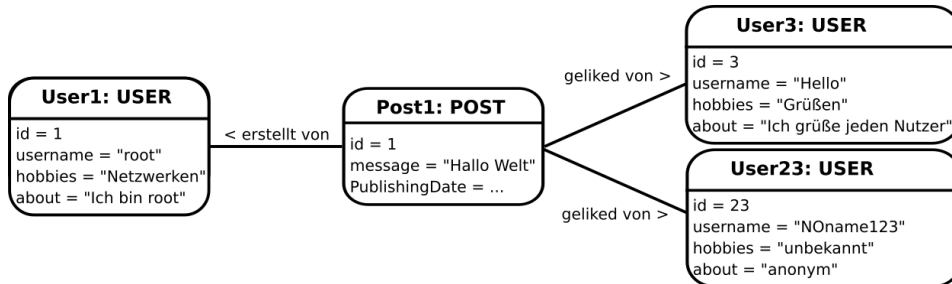


Abb. 4: Objektdiagramm eines Posts

SchülerInnen Hilfestellung und Anregungen für mögliche Weiterentwicklungen geben. Typische Anwendungen sind:

- Posten vorgefertigter Texte auf der eigenen Pinnwand. (*einfach*)
- Liken aller Beiträge auf der eigenen Pinnwand. (*einfach*)
- Posten von Texten, die aus mehreren Satzteilen zufällig kombiniert werden. (*mittel*)
- Posts nach Schlüsselwörtern durchsuchen und dann liken. (*anspruchsvoll*)
- Nutzerprofile nach Schlüsselwörtern durchsuchen und dann vorgefertigte Texte schreiben. (*anspruchsvoll*)

Fortgeschrittene Anwendungen

Manche Funktionen bietet das SocialBotNet nicht über die API an. So ist die Registrierung nur über die normale Browser-Schnittstelle verfügbar. Da jedoch der Browser das selbe Protokoll verwendet, können die Bots diese POST-Schnittstelle ebenfalls benutzen und so automatisiert neue Accounts erstellen. Erfahrungsgemäß versuchen einzelne SchülerInnen von alleine solche Tricks. Ein Erfolg kann hier stark motivieren und spiegelt ein herausragendes Verständnis des Unterrichtsinhalts wider. Solche „Sicherheitslücken“ regen auch an, über Schutzmaßnahmen zu diskutieren. Üblich ist beispielsweise der Einsatz von CAPTCHAs bei Schnittstellen, die für automatisierte Programme nicht zugänglich sein sollen.

Leistungsstarke SchülerInnen können auch andere Webservices anbinden, die über eine JSON-API verfügen. Verknüpfungen mit Nachrichten oder Wetterdaten sind so beispielsweise möglich.

4.3 Vierte Stunde: Sicherung und Reflexion

In der abschließenden Unterrichtsstunde findet die Sicherung und Reflexion des Gelernten statt. Anhand eines konkreten Schülerprogramms und dem zugehörigen Kommunikationsverlauf wird die Netzwerkkommunikation zwischen Client und Server wiederholt. Abgerundet wird das Thema mit einer Diskussion der gesellschaftlichen Auswirkungen von Social Bots, insbesondere der möglichen Einflussnahme auf politische Debatten.

5 Erfahrungen aus der gehaltenen Unterrichtseinheit

Die Thematik allgemein, aber insbesondere das Ziel, einen eigenen Bot zu programmieren, war für die SchülerInnen sehr motivierend. Die Differenzierungsangebote wurden gut angenommen, beispielsweise haben ca. 60% die Projektvorlage auf Objektbasis genutzt, 10% die Projektvorlage bei der JSON direkt verarbeitet werden musste und der Rest hat ohne Vorlage seine persönlichen Bots implementiert. Einzelne SchülerInnen führten die Programmierung sogar nach den Unterrichtseinheiten in ihrer Freizeit fort und entwickelten z. B. Botarmeen mit generierten Personennamen.

6 Weiterentwicklung

Das SocialBotNet ist in Java mit dem Spark Framework programmiert und auf Github veröffentlicht [3]. Aktuelle Pläne zur Weiterentwicklung können dort eingesehen und weitere Vorschläge eingebracht werden. Derzeit geplant sind folgende Weiterentwicklungen:

- Antworten auf Posts: Derzeit ist es nur möglich, auf dem Profil eines Nutzers einen Kommentar als Antwort zu schreiben, aber nicht, sich auf eine bestimmte Nachricht zu beziehen und dadurch direkt zu antworten. Die geplante Erweiterung bietet daher neue Möglichkeiten für die Social Bots mit anderen zu interagieren.
- Getrennte Bereiche für Schulklassen: Statt alle Bots und Kommentare im selben globalen Bereich zu handhaben, ist es für viele Schulklassen auch interessant, einen eigenen abgetrennten Bereich zu besitzen.
- API-Explorer: Im Browser lassen sich nur GET-Anfragen direkt ausprobieren und ein selbstständiges Entdecken der API ist nicht möglich. Es soll eine Seite eingerichtet werden, in der verschiedene Anfragen simuliert werden können. Dies bietet einen geeigneten Zwischenschritt zwischen der Konzeption und der tatsächlichen Programmierung.
- Weiterentwicklung der Materialien für den Einsatz mit anderen Programmiersprachen: Geplant ist eine Erweiterung der Materialien für JavaScript und Python, bei denen die Netzwerkkommunikation sogar leichter ist als in Java.

Weiterführende Links

- [1] Gehostetes Netzwerk: <https://www.socialbotnet.de>
- [2] Arbeitsblätter und Projektvorlagen: <https://www.socialbotnet.de/material>
- [3] Quellcode: <https://github.com/Knorrke/socialbotnet>
- [4] Eigenes Setup: https://github.com/Knorrke/socialbotnet#build_instructions
- [5] Video „Social Bots“ - ZDF heuteplus: <https://youtu.be/HVuB1QPxdT0>
- [6] Video „Fake News & Social Bots in 3 Minuten erklärt“ - explain-it: <https://youtu.be/j14s00N3c1g>

Literatur

- [Ba17] Balzer, I.: Social Bots können kleine Gruppen wie eine Bewegung aussehen lassen, *Zeit ONLINE*, Sep. 2017, URL: <https://www.zeit.de/digital/internet/2017-09/soziale-medien-bundestagswahl-manipulation-social-bots-trolle>.
- [Br10] Brichzin, P.; Freiberger, U.; Reinold, K.; Wiedemann, A.: *Informatik Oberstufe 2 – Maschinenkommunikation – Theoretische Informatik*. Oldenbourg Schulbuchverlag, München, 2010.
- [HKW16] Howard, P. N.; Kollanyi, B.; Woolley, S.: *Bots and Automation over Twitter during the US Election*. Computational Propaganda Project: Working Paper Series/, 2016.
- [St04] Staatsinstitut für Schulqualität und Bildungsforschung (ISB): *Lehrplan des achtjährigen Gymnasiums, Informatik 11/12, 2004*, URL: http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/id_26193.html.
- [St18] Statista: *Most famous social network sites 2018, by active users*, Jan. 2018, URL: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>.
- [Su16] Suárez-Serrato, P.; Roberts, M. E.; Davis, C.; Menczer, F.: *On the Influence of Social Bots in Online Protests*. In (Spiro, E.; Ahn, Y.-Y., Hrsg.): *Social Informatics*. Springer International Publishing, Cham, S. 269–278, 2016, ISBN: 978-3-319-47874-6.
- [Wa12] Wagner, C.; Mitter, S.; Körner, C.; Strohmaier, M.: *When social bots attack: Modeling susceptibility of users in online social networks*. *Making Sense of Microposts (# MSM2012) 2/4*, S. 1951–1959, 2012.