

DOI: <https://doi.org/10.11588/ip.2019.1.57941>

**Christian ERLINGER**

## Instant-Messenger Bots als alternative Suchoberfläche

### Zusammenfassung

Instant Messenger sind fixer Bestandteil in der alltäglichen Kommunikation über unsere Smartphones. Im bibliothekarischen Informationsdienst finden solche Medien teilweise Eingang im klassischen Auskunftsdienst, doch bieten diese Services durch API-Programmierung die Möglichkeit, sie als alternative Suchoberfläche mit integriertem Volltextservice zu nutzen. Der hier präsentierte Telegram-Bot erlaubt die Recherche und Speicherung von Open Access Artikeln über die REST-API von [openaccessbutton.org](https://openaccessbutton.org).

### Schlüsselwörter

Discovery Service; Instant Messaging; Open Access

## Instant Messenger Bots as alternative Search Screen

### Abstract

Instant Messenger are widely common in the daily communication. In the information services of libraries these medias are used as an additional communication channel. In combination with API-Engineering Instant Messenger could be applied as an alternative search screen or document delivery system. This article presents a chatbot for the Instant Messenger Telegram, which uses the REST-API of [openaccessbutton.org](https://openaccessbutton.org) to search for Open Access Articles and deliver the corresponding full-text files

### Keywords

Discovery Service; Instant Messaging; Open Access

## 1 Instant Messaging und Bibliotheken<sup>1</sup>

Instant Messaging hat sich von einem noch 2016 als Trendthema (Fingerle 2017) bezeichneten zu einem vollauf etablierten Kommunikationsmittel etabliert. Die Zahl der mittels Messengersoftware versandten Nachrichten hat dabei schon seit Längerem die Zahl der versandten SMS überholt (Sparkes 2015), was auch keineswegs zu verwundern braucht, denn die Einsatzfähigkeit eines Messengerdienstes – Text, Sprachnachricht, Grafik oder Video, Verwendung im Zweierdialog oder als Gruppenchat – übersteigt doch bei Weitem jene der klassischen SMS. Hinzu kommt, dass Messenger zunehmend als „Informationsmedium“ Anwendung finden. Die ersten Applikationen dabei sind reine Push-Dienste, bei denen Interessierte regelmäßig spezifische Nachrichten zugesandt bekommen. Die Stärke der Messengersoftware besteht aber darin, solche Informationsangebote mit einem Kommunikationselement zu versehen, das heißt, unmittelbar auf Eingabe der BenutzerInnen zu reagieren, und zwar in erster Linie automatisiert. „Ein oder zwei Tage auf eine Antwort zu warten, finden viele daher nicht mehr akzeptabel; sofortige Reaktionen innerhalb von drei bis fünf Minuten und On-Demand-Services werden immer mehr als Standard betrachtet.“ (Fingerle 2017)

Für Bibliotheken bieten Instant Messenger eine interessante Möglichkeit, ihre Verteilungs- und Kommunikationsformen an das aktuelle BenutzerInnenverhalten anzupassen. Bislang scheinen noch wenige Institutionen aktiv Instant Messaging in ihren Services bereitzuhalten. Winnacker (2016) beschreibt in ihrer Bachelorarbeit zwei Beispiele des Einsatzes von WhatsApp als zusätzlichen Kommunikationskanal im bibliothekarischen Auskunftsdienst. Die Stärke des Instant Messaging wird dadurch aber noch lange nicht ausgeschöpft. Die Vorteile liegen in der automatisierten Kommunikation mittels Chatbots sowie in der unmittelbaren Koppelung an die Bibliotheksbestandsdaten.

## 2 OpenAccessButton.Org – Chatbot für Telegram

Die vorliegende Arbeit präsentiert einen Chatbot für den Instant-Messenger Telegram<sup>2</sup>, der es ermöglicht, über die offene API der Seite OpenAccessButton.org<sup>3</sup> mittels Eingabe eines permanenten Identifiers (DOI) oder durch Titel-String-Suche nach wissenschaftlichen Open Access Artikeln zu recherchieren und diese direkt in den Messenger-Chat gesendet zu bekommen.

Die Idee einen solchen Chatbot zu entwickeln, stammt daher, dass Sci-Hub für Telegram einen Bot mit der genannten bzw. bekannten Funktionalität anbietet (vgl. Passig 2018:74 f).

*„Bei Sci-Hub kommt noch etwas anderes hinzu. Viele Nutzer\*innen rufen auch Publikationen ab, die sie über ihre Heimateinrichtungen abrufen könnten, da diese Lizenzen erworben hat. Das ist eher eine Sache der Usability. Bei Sci-Hub bekommt man ein PDF direkt nach Eingabe der DOI. Die kann man leicht aus den Referenzen eines anderen Papers herauskopieren oder sogar abtippen. In zwei Schritten hat man die Quelle. Bei Bibliothekskatalogen und Verlagsdatenbanken muss man in der Regel längere Wege gehen. Arbeitet man zum Beispiel im Home-Office, muss man sich zumeist erst umständlich einwählen bzw. identifizieren.“ (Seer 2017)*

---

1 Dieser Text basiert auf einem am 33. Österreichischen Bibliothekartag am 15. September 2017 in Linz gehaltenen Vortrag. (Votragfolien verfügbar unter <https://nbn-resolving.org/urn:nbn:de:0290-opus4-32693>, abgerufen am 17. Jänner 2019)

2 <https://telegram.org/>, abgerufen am 17. Jänner 2019

3 <https://openaccessbutton.org>, abgerufen am 17. Jänner 2019

Warum sollten Bibliotheken, die mit viel Engagement gerade im Bereich des Open Access Akzente setzen, eine illegale Plattform wie Sci-Hub obsolet zu machen, nicht aber dort Anleihe in technologischer Praxis bzw. hinsichtlich der Usability nehmen? Die gerade am Smartphone in Kombination mit einem Messenger intuitive und einfache Bedienung kann im Rahmen wissenschaftlicher Recherche und Arbeit von großem Nutzen sein.

Open Access Button ist eine Plattform, die verschiedene Open Access Verzeichnisse oder Repositorien (wie oaDOI<sup>4</sup> oder BASE<sup>5</sup>) gebündelt durchsuchbar macht. Ein wesentliches Ziel dabei ist die direkte Verlinkung zu den jeweiligen Volltexten. Darüber hinaus bietet die Plattform die Möglichkeit an, einen „Request“ abzusetzen, wenn gesuchte Artikel nicht in einer Open Access Version verfügbar sind. Dieser Request soll den AutorInnen des Artikel zugestellt werden, damit diese nach Maßgabe der rechtlichen Optionen gegebenenfalls eine freie Version des Artikels publizieren. Open Access Button bietet auch eine gut dokumentierte und einfach strukturierte REST-API an, die zur Nachnutzung in eigenen Entwicklungsprojekten, so wie eben in dem hier gezeigten, verwendet werden kann.

### Funktionalität des Chatbots

Der Chatbot wird nach Registrierung bzw. Installation der Telegram Messenger App durch Aufrufen der URL [https://telegram.me/bOAt\\_oabot](https://telegram.me/bOAt_oabot) in die Messenger-Chatliste geladen. Durch Eingabe eines Start- bzw. Hilfefkommandos (/start oder /help) bzw. durch „Hi“ erscheint eine kurze Information die die Benutzung des Bots erläutert. (vgl. Abb. 1)

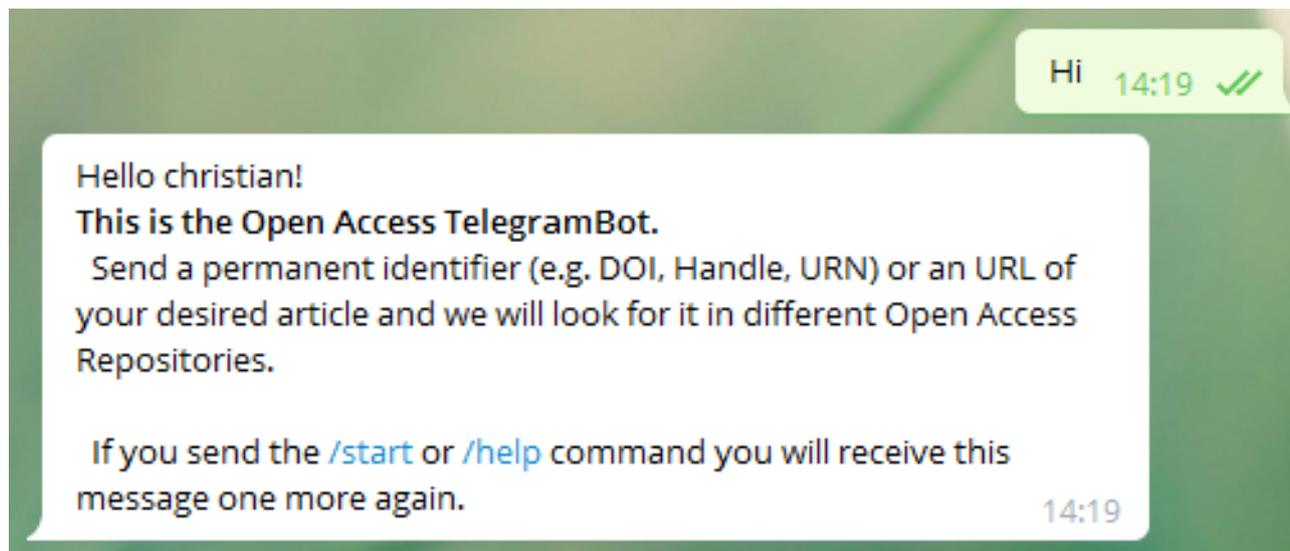


Abb. 1: Start- bzw. Hilfetext

Durch Eingabe einer DOI, einer URL oder eines Titel-Strings, wobei hier aktuell über die API Treffer nur bei exakter Angabe des Titels erzeugt werden können, wird eine Suche im Open Access Button Datensatz ausgelöst. (vgl. Abb. 2) Bei Open Access Verfügbarkeit eines Artikels erscheint im Chat eine Kurzmitteilung, die zumindest den Link zur Fundstelle beinhaltet. Falls ein Link zum Volltext maschinell ermittelbar ist, wird auch der Artikel im Chatdialog integriert.

4 <http://unpaywall.org/>, abgerufen am 17. Jänner 2019

5 <https://de.base-search.net/>, abgerufen am 17. Jänner 2019

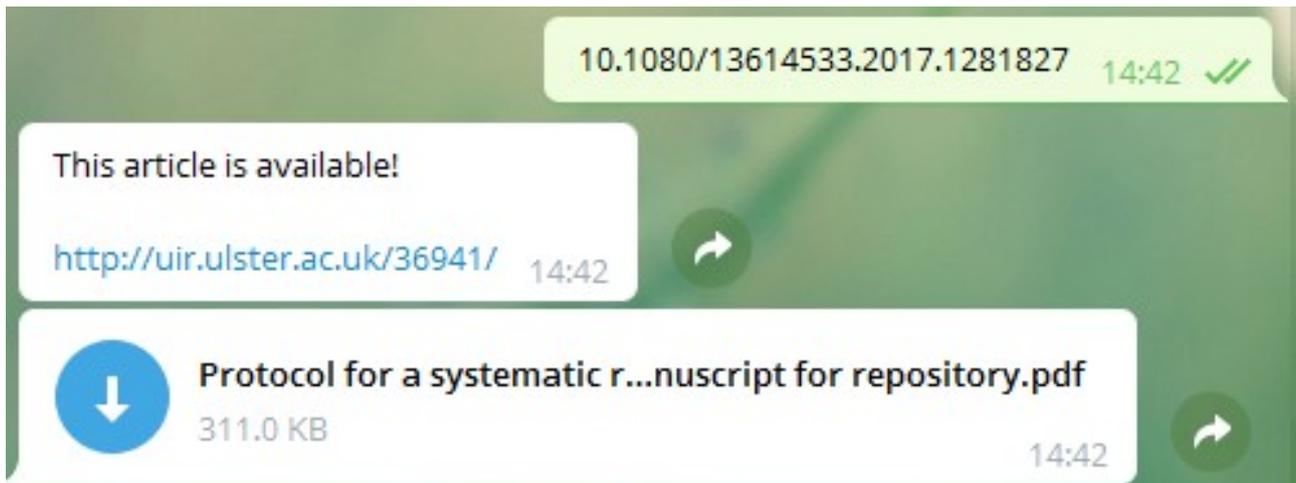


Abb. 2: Antwort bei Verfügbarkeit eines Artikels

Wird nach einem Artikel gesucht, der nicht mit einer der in Open Access Button verwendeten Ressourcen gefunden werden kann, so erscheint im Chatdialog eine Nachricht, die einen Link beinhaltet, um auf der Open Access Button Website einen bereits erwähnten Request zur Publikation einer freien Artikelversion enthält. (vgl. Abb. 3)

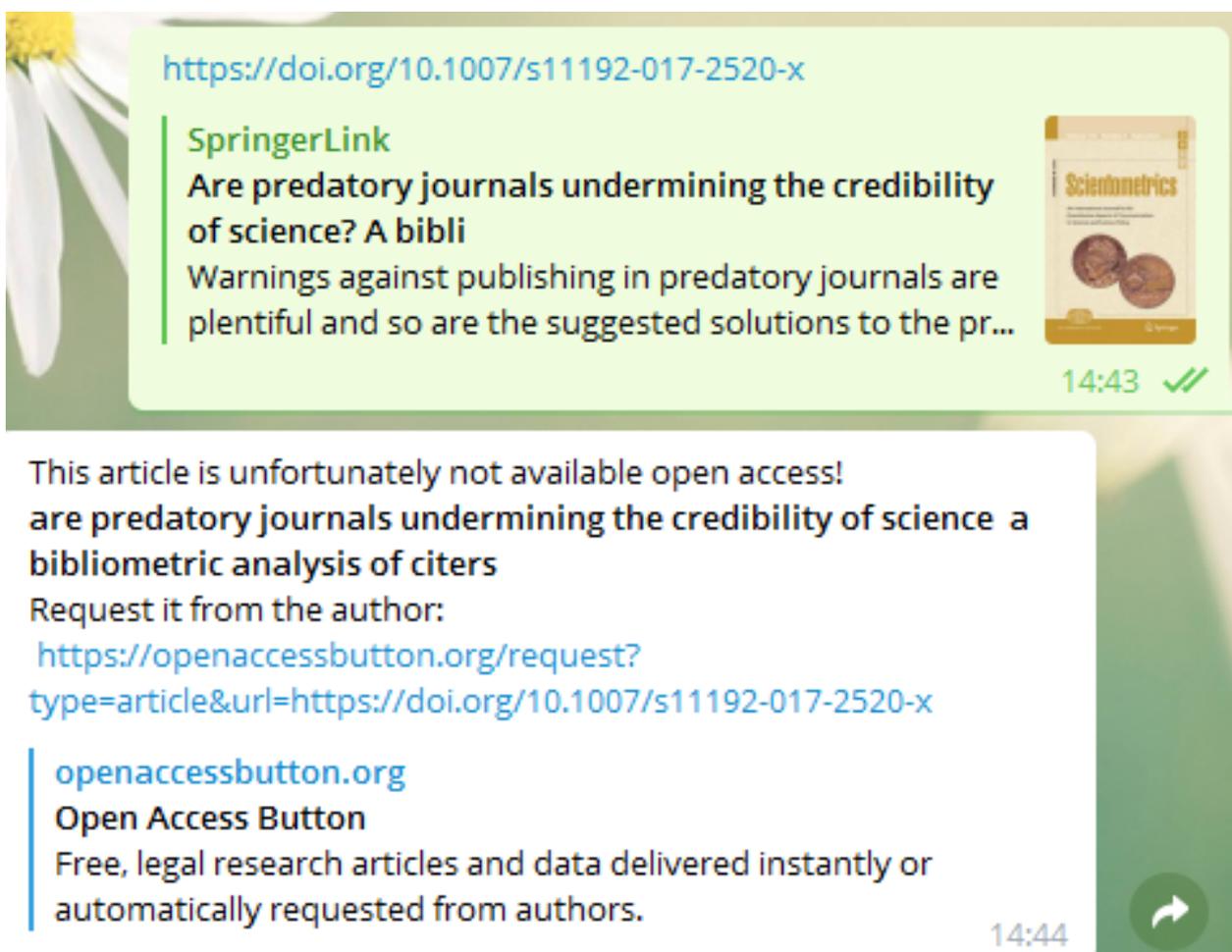


Abb. 3: Antwort bei Nichtverfügbarkeit eines Artikels

## 2.2 Technische Bemerkungen zur Implementation des Chatbots

Um einen Telegram-Chatbot zu initialisieren, ist die Generierung eines API-Keys Voraussetzung. Ein solcher API-Key wird von einem eigenen Telegram-Chatbot namens Boffather erzeugt. Somit ist für die Einrichtung eines Bots eine einmalige Registrierung in Telegram mittels verfügbarer Telefonnummer notwendig. Die Telegram-Bot-API<sup>6</sup> ist sehr gut entwickelt und dokumentiert, dazu sind zahlreiche, quelloffene Programmbibliotheken in verschiedensten Programmiersprachen vorhanden, die es mit wenig Aufwand erlauben, einen in Grundzügen funktionsfähigen Bot zu erstellen.

Der Quellcode des hier gezeigten Open Access Button Chatbots ist auf GitHub inklusive der verwendeten Telegram-API-Library verfügbar.<sup>7</sup> Der vorliegende Chatbot ist in PHP geschrieben und nutzt eine dafür entwickelte Telegram-API-Library<sup>8</sup>. Die Open Access Button API<sup>9</sup> wird über die REST-Schnittstelle von Open Access Button genutzt, um somit einen stets aktuellen Datenbestand abfragen zu können. Alternativ, zur Steigerung der Abfragegeschwindigkeit bzw. um die Titelstrings indiziert zu durchsuchen, könnte auch ein Abzug der Open Access Button Daten lokal genutzt werden, der aber klarerweise regelmäßig aktualisiert werden müsste.

Mittels Webhook<sup>10</sup> wird bei jeder Eingabe im Chatfenster ein Aufruf an eine im Boffather hinterlegte Webadresse, den Speicherort des individuellen Bot-Programmes, gestartet, der neben einer User-ID auch den Inhalt der Anfrage übermittelt. Theoretisch könnten diese Informationen dann lokal gespeichert werden, um beispielsweise Push-Nachrichtendienste gezielt an BenutzerInnen des Chatbots übermitteln zu können. Der hier beschriebene Bot speichert aber lokal keine Daten sondern reagiert nur auf die übermittelten Daten (DOI, URL, Chat-Text, Chat-Kommando). Der Bot überprüft den übermittelten Textstring, gibt entweder den gezeigten Hilfstext aus oder setzt eine Suche ab. Je nach Antwort der Open Access Button REST-Schnittstelle wird eine der gezeigten Antwort generiert.

## 3 Maschinenlesbare Metadaten in Repositorien als Basis zum Abgreifen der Volltexte

Die Transferierung eines Volltextdokuments direkt in den Chatdialog, setzt allerdings die maschinell eindeutig auffindbare Adresse des Dokuments voraus. Im Idealfall befinden sich auf den Landingpages der Repositorien im HTML-Header standardisierte bibliographische Informationen, die auch einen eigenständigen Tag enthalten, der die Adresse zum Volltextdokument, egal in welchem Format, ausweist (vgl. Listing 1). Bei der Entwicklung des hier präsentierten Bots zeigte sich bereits beim Besuch weniger Repositorien, dass der Umgang mit der standardisierten Darstellung bibliographischer Informationen im HTML-Header höchst unterschiedlich gehandhabt wird.

---

6 <https://core.telegram.org/bots/api>, abgerufen am 17. Jänner 2019

7 <https://github.com/LibrErl/OpenAccessButtonTelegramBot>, abgerufen am 17. Jänner 2019

8 <https://github.com/Elleirbag89/TelegramBotPHP>, abgerufen am 17. Jänner 2019

9 <https://openaccessbutton.org/api>, abgerufen am 17. Jänner 2019

10 Webhook ist ein Verfahren, das bei Eintreten eines Ereignisses ein definiertes Server-Skript aufruft und die Daten des Ereignisses übermittelt. (vgl.

<https://web.archive.org/web/20120413121142/http://wiki.webhooks.org/w/page/13385124/FrontPage>, abgerufen am 17. Jänner 2019)

```

<meta name="eprints.document_url" content="http://uir.ulster.ac.uk/36941/1/Protocol
%20for%20a%20systematic%20review%20-%20Manuscript%20for%20repository.pdf" />
<link rel="schema.DC" href="http://purl.org/DC/elements/1.0/" />
<meta name="DC.relation" content="http://uir.ulster.ac.uk/36941/" />
<meta name="DC.title" content="Developing and applying a protocol for a systematic
review in the social sciences" />
<meta name="DC.creator" content="Campbell, Allison" />

```

*Listing 1: HTML Meta-Tags mit bibliographischen Angaben*

Um auch ein sehr praktisches Beispiel anzuführen, inwieweit standardisierte Metadaten in HTML-Seiten zur erhöhten Sichtbarkeit von Informationen in den Social-Media oder Messengerdiensten bereits Eingang gefunden haben, sei auf Abb. 3 verwiesen. Bei der eingegeben Suche des DOI-URL eines spezifischen Artikels, entsteht in der Suchanfrage darunter eine Kurzinformation mit Artikel-Titel, Abstract-Ausschnitt und Cover. Diese Informationen werden selbständig vom Chatprogramm aus Metadaten des HTML-Headers geparkt. Für diese automatischen Daten werden beispielsweise Metadaten, die im Open Graph Format<sup>11</sup> hinterlegt sind, eingesetzt. Dieses sehr schlanke von Facebook entwickelte Format hat sich mittlerweile als Standard in den Sozialen Medien bzw. im Webdesign entwickelt und sollte daher durchaus auch in den fachlichen Repositorien mit den entsprechend korrekten Informationen befüllt sein. Für Publikationswebseiten wurden im Open Graph Protocol eigene Objekte für article, book und profile definiert.<sup>12</sup>

Die vorhandenen bibliographischen Informationen lassen sich aber auch in anderen Formaten gewinnbringend darstellen. Beispielsweise in Dublin Core oder in jenen für die Auswertung durch Google Scholar verwendeten citation\_<sup>13</sup> Schema. EPrints-Repositorien bieten übrigens sogar eigenständige eprints.<sup>14</sup> Metadatenchema an.

Bibliotheken betreiben einen nicht zu unterschätzenden Aufwand mit der Erstellung formaler und inhaltlicher Metadaten der unselbständigen Literatur, die in institutionellen Repositorien gespeichert wird. Daher wäre es ein einmaliger und sich sicherlich lohnender Aufwand, Mappings in einige der genannte Metadatenformate für die HTML-Header zu erstellen.<sup>15</sup> Denn gerade durch die Verlinkung der vorhandenen Daten und den damit möglichen maschinellen Austausch, bspw. durch Webscraping abseits der standardisierten Schnittstellen, wird auch die Sichtbarmachung sowie die bessere Messung der Publikationen erleichtert (Pekala 2018) bzw. überhaupt erst ermöglicht.

## 4 Ausblick

Instant Messaging bzw. Social Media Applikationen bieten vor allem durch die teils sehr gut ausgebauten Möglichkeiten automatischer Informationsverarbeitung interessante Distributions-

<sup>11</sup> <http://ogp.me>, abgerufen am 17. Jänner 2019

<sup>12</sup> [http://ogp.me/#type\\_article](http://ogp.me/#type_article), abgerufen am 17. Jänner 2019

<sup>13</sup> <https://gist.github.com/hubgit/f3e359ab51da6d15118b>, abgerufen am 17. Jänner 2019

<sup>14</sup> [https://wiki.eprints.org/w/Category:Eprints\\_Metadata\\_Fields](https://wiki.eprints.org/w/Category:Eprints_Metadata_Fields), abgerufen am 17. Jänner 2019

<sup>15</sup> Es könnte dies auch eine durchaus lohnenswerte Forschungsaufgabe sein, den aktuellen Stand der unterschiedlich im Informationsbereich eingesetzten Softwarepakete in deren Verarbeitung und Sichtbarmachung bibliographischer und technischer Metadaten zu analysieren und zu vergleichen.

und Kommunikationskanäle für Bibliotheken. Bots bieten dabei, abgesehen von einem reduzierten Ressourcenbedarf im Betrieb, in folgenden Einsatzszenarien praktische Dienste:

- Als anonyme Schnittstelle zu Discovery Services oder auch im personalisierten Austausch mit dem OPAC zur Steuerung der Entlehnungen etc.
- Als Werkzeug für eine automatisierte Vorstufe der Fernleihe und des elektronischen Dokumentenlieferservices, das, wie gezeigt, in Kombination mit Open Access Datenbeständen aufgebaut wird.
- Als Push-Nachrichtendienste für personalisierte Suchanfragen („Alert-Messaging“) oder Neuerscheinungen/Neuzugänge.

Grundvoraussetzung für den Aufbau automatischer Kommunikations- und Rechercheinstrumente ist das Vorhandensein offener und standardisierter Datenquellen und Programmierschnittstellen sowie letztlich auch offen zugänglicher Literatur. Unter den Bedingungen einer offenen Wissensgesellschaft ist es leicht, spielerisch und mit geringem personellen Einsatz<sup>16</sup> innovative Distributionsformen auszuprobieren und nutzbar zu machen.

## Quellen

Fingerle, Birgit 2017. *Trends die 2016 auf die Agenda kommen*. ZBW Mediatalk. <https://www.zbw-mediatalk.eu/2016/01/7-trends-die-2016-auf-die-agenda-kommen/> [Stand 2019-01-4].

Passig, Kathrin 2018. Schattenbibliotheken. [https://docs.google.com/presentation/d/1wF41nNijMQ4-MI0ERAIkYcq0\\_ySDIBJWyiL9ITHo6dg/edit](https://docs.google.com/presentation/d/1wF41nNijMQ4-MI0ERAIkYcq0_ySDIBJWyiL9ITHo6dg/edit) [Stand 2019-01-4].

Pekala, Shayna 2018. Microdata in the IR: A Low-Barrier Approach to Enhancing Discovery of Institutional Repository Materials in Google. *The Code4Lib Journal* 39, . <http://journal.code4lib.org/articles/13191> [Stand 2019-01-4].

Seer, Veil 2017. Von der Schattenbibliothek zum Forschungskorpus. Ein Gespräch über Sci-Hub und die Folgen für die Wissenschaft. *LIBREAS. Library Ideas* 32, . <https://libreas.eu/ausgabe32/scihub/> [Stand 2019-01-4].

Sparkes, Matthew 2015. WhatsApp overtakes text messages. *The Telegraph* . <https://www.telegraph.co.uk/technology/news/11340321/WhatsApp-overtakes-text-messages.html> [Stand 2019-01-4].

Winnacker, Laura 2016. *Eine Analyse über die Integration von WhatsApp in den Auskunftsdienst von öffentlichen und wissenschaftlichen Bibliotheken in Deutschland*. Suttgart. [https://hdms.bsz-bw.de/files/6014/01\\_BA\\_WhatsApp\\_Final.pdf](https://hdms.bsz-bw.de/files/6014/01_BA_WhatsApp_Final.pdf) [Stand 2019-01-4].

---

16 Der hier präsentierte Bot wurde in ca. acht Arbeitsstunden entwickelt.

## Autor

Christian ERLINGER

Büchereien Wien

Urban-Loritz-Platz 2a

1070 Wien

[christian.erlinger-schiedlbauer@wien.gv.at](mailto:christian.erlinger-schiedlbauer@wien.gv.at)

<https://orcid.org/0000-0001-7872-9617>