



hoax.ly

Endbericht | Call 11 | Projekt ID 2035

Inhalt

Einleitung	3
Projektbeschreibung	3
Verlauf der Arbeitspakete	3
Arbeitspaket 1 - Projektmanagement	3
Arbeitspaket 2 - Konzept	4
Arbeitspaket 3 - Datenbank	4
Arbeitspaket 4 - Webserver einrichten	5
3.5 Arbeitspaket 5 - Browser Plugin programmieren	5
3.6 Arbeitspaket 6 - Chatbots	6
3.7 Arbeitspaket 7 - Webseiten Installationspaket	10
3.8 Arbeitspaket 8 - Datenqualität	13
3.9 Arbeitspaket 9 - Datenquellen	13
3.10 Arbeitspaket 10 - Tests & Dokumentation	14
3.11 Arbeitspaket 11 - Öffentlichkeitsarbeit	15
Liste Projektendergebnisse	19
Verwertung der Projektergebnisse in der Praxis	20
Öffentlichkeitsarbeit/ Vernetzung	20
Geplante Aktivitäten nach netidee-Projektende	21
Anregungen für Weiterentwicklungen durch Dritte	21

1 Einleitung

Falschmeldungen und Gerüchte verbreiten sich aktuell wie verrückt. Mehrere tausend Shares oder Retweets von objektiv falschen Informationen sind an der Tagesordnung. Angstmache und das Streuen von falschen Gerüchten ist nicht nur im Zusammenhang mit der Flüchtlingskrise ein Problem. Leider teilen viele Informationen im Internet ungeprüft bzw. werden zu Opfern von Betrugereien. Hoaxly schärft das Bewusstsein für dieses Problem und hilft solche problematischen Inhalte besser zu erkennen.

2 Projektbeschreibung

Hoaxly unterstützt die journalistische Arbeit zur Richtigstellung und Dekonstruktion von Falschmeldungen, Gerüchten und Betrugereien im Internet und besteht aus folgenden Tools:

- 1) Eine Browser-Erweiterung, mit dessen Hilfe problematische Inhalte direkt im Browser markiert werden und die einen Link zur Richtigstellung hinzufügt.
- 2) Einen Chat Bot für die gängigen Plattformen (Facebook, Slack, ..), der warnt, wenn problematische Inhalte geteilt werden bzw. Antworten auf oft gestellte Fragen gibt.
- 3) Eine zentrale Datenbank mit Schnittstelle (API), in der Metainformationen zu Richtigstellungen von Hoaxes gespeichert werden. Dient als Datenquelle für obige Tools. Ein Kriterienkatalog wird dafür ausgearbeitet, welche Quellen aufgenommen werden (zB mimikama, Watchlist Internet).
- 4) Webseiten Installationspaket für die Meldung/Bearbeitung von Hoaxes. Ermöglicht Privatpersonen, Vereinen und Communities Falschmeldungen redaktionell zu verarbeiten und über die Hoaxly-DB mit der Allgemeinheit zu teilen.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - Projektmanagement

Projektmanagement einrichten (Jira), Team definieren, Userstories und Backlog schreiben, regelmäßige Sprint-Planung, regelmäßige Retrospektiven

Die Userstories wurden sehr gewissenhaft erstellt. Durch das genaue Durchdenken aller Features sind nicht wenige Punkte aufgetaucht, die für das Konzept und die technische Umsetzung relevant waren.

Als Backlog wurden Tasks im Jira angelegt, mit jeweils einem Verweis auf die relevante Userstory. Die Arbeiten wurden in monatlichen Sprints organisiert. Regelmäßig gab es ein

Retrospektiv-Meeting, wo die Arbeit präsentiert wurde und Probleme sowie Fortschritte diskutiert wurden. Während der Sommermonate wurden diese Retrospektiv-Meetings aufgrund der wenigen Fortschritte jedoch nur spärlich abgehalten.

Die agile Herangehensweise war schon alleine deswegen notwendig, weil es doch einige Bereiche gab, in denen erst während der produktiven Arbeit erkennbar wurde, mit welchen Themen oder gar Schwierigkeiten man zu tun hat.

3.2 Arbeitspaket 2 - Konzept

Erstellung technisches Konzept, Erstellung inhaltliches Konzept, Erstellung grundlegendes Design

Neben dem technischen Konzept wurde konzeptionell sehr viele Überlegungen angestellt. Macht ein Tool wie hoaxly überhaupt weiterhin Sinn? Es schießen ja aktuell sehr viele Tools und Projekte rund um Fake-News aus dem Boden. Wir konnten diese Frage für uns jedoch mit einem klaren Ja beantworten. Oberstes Ziel muss sein mit hoaxly die Medienkompetenz zu stärken, hier sollten die hoaxly Anwendungen also vor allem eine pädagogische Rolle spielen. Außerdem sehen wir auch einen steigenden Bedarf bei der Zielgruppe der JournalistInnen, vor allem an einer Meta-Suchmaschine wie hoaxly es ist.

Eine Liste von Richtigstellungsplattformen, mit u.a. Analyse der Datenstruktur, Auszeichnungssprache (zB. ClaimReview), verwendetes CMS, Bewertungssystematik und Verpflichtung zu einem "Code of Principles" ist hier veröffentlicht:

https://drive.google.com/open?id=1-jm8Rs-O_I9-dgBqB0Xz05Ajv2T4Bl6bi9g-aR1Zzh4

Diese Liste ist für uns Ausgangspunkt dafür zu entscheiden, welche Daten welcher Plattformen in die hoaxly Datenbank aufgenommen werden bzw. welche Quellen von uns für diesen Zweck genauer überprüft werden und wird laufend erweitert.

3.3 Arbeitspaket 3 - Datenbank

Basis unserer technischen Lösung ist Elasticsearch und Kibana. Das Scraping erfolgt über Scrapy (<https://scrapy.org/>) und Portia. Portia (<https://scrapinghub.com/portia/>) ermöglicht die Erstellung von Scraping Templates ohne Programmieren zu müssen.

Wir importieren erfolgreich HTML, RSS-Feeds und strukturiertes Markup wie schema.org. Der für Richtigstellungen vorhandene ClaimReview schema.org Standard wird leider aktuell nur von sehr wenigen Seiten unterstützt, vor allem nicht von europäischen Plattformen. Das würde den Import der Daten extrem erleichtern.

Berechtigungslayer für die Datenbank-Abfragen: Für den Zugriff auf den Index durch externe Personen dürfen diese nur Lesezugriff haben und es sollten nur ausgewählte Endpoints zur Verfügung stehen.

Hierfür wurde eine eigene API-Applikation geschrieben, die auf den Index in ElasticSearch zugreift und eine Swagger-Dokumentation bereit stellt.

Die API ist unter <https://api.hoax.ly/api/check> zugänglich, die Swagger Doku unter <https://api.hoax.ly/api>.

Lizenziert sind die Inhalte der Datenbank (Index) unter ODbL 1.0:
<https://opendatacommons.org/licenses/odbl/summary/>

3.4 Arbeitspaket 4 - Webserver einrichten

Ein Webserver wurde bei hetzner.de gemietet und eingerichtet. Aktuell laufen u.a. eine ElasticSearch Instanz mit Kibana und die vorhandenen Scraping Tools (Scrapy, Portia).

<https://kibana.hoax.ly>
<https://scrapyd.hoax.ly/>

Zwei weitere Instanzen wurden eingerichtet, eine für die API (<https://api.hoax.ly>) und eine für die Chatbot-Applikation (<https://chat.hoax.ly>).

Für das Deployment und Packaging verwenden wir Docker bzw. Ansible. Lokal können damit Entwicklungsumgebungen leicht aufgesetzt werden und Updates einfach auf den Server deployed werden.

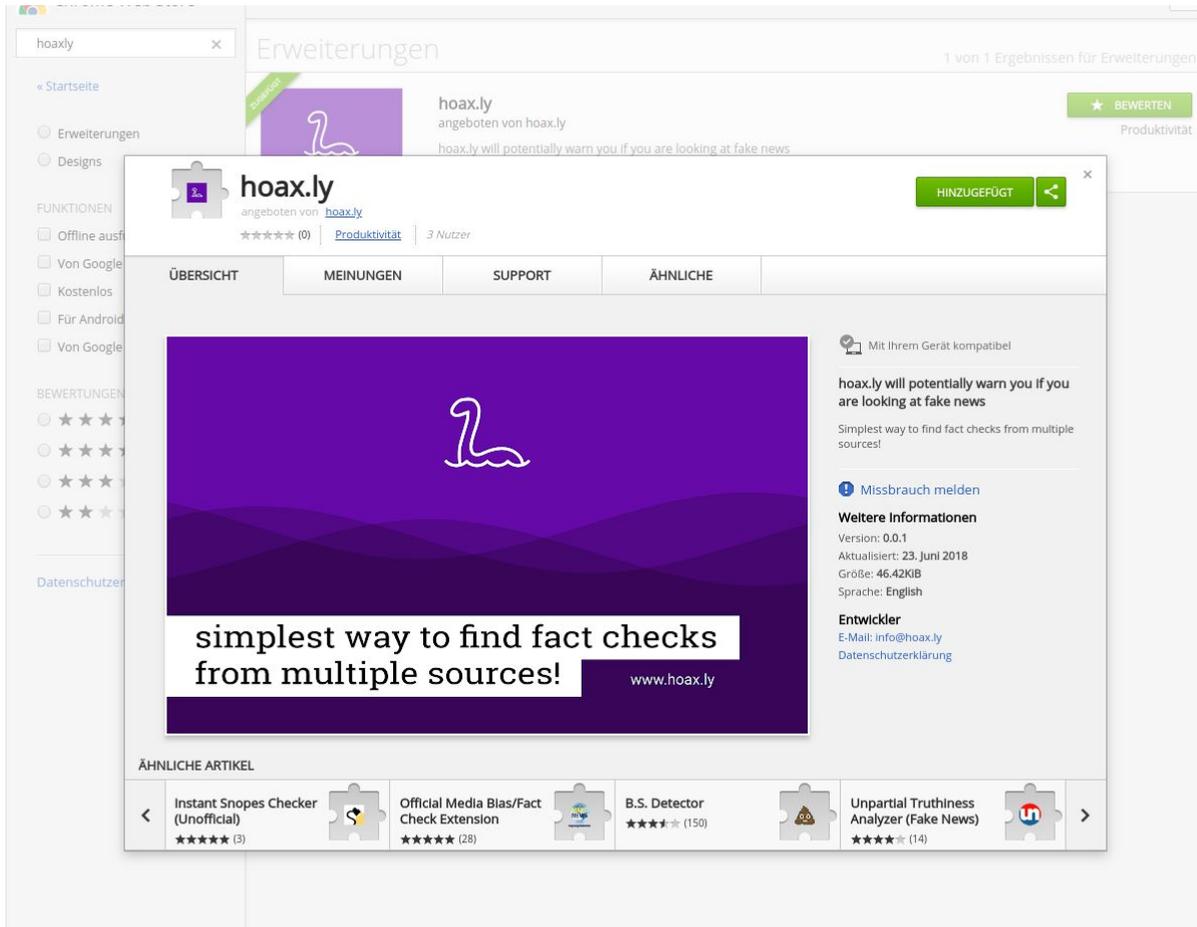
Jedes Repository hat entsprechende Docker Files bzw. stehen auch eigene Docker Images in einer gitlab registry zur Verfügung. Eigene Deployment-Prozesse erstellen diese Docker-Images automatisiert bei jeder Änderung.

3.5 Arbeitspaket 5 - Browser Plugin programmieren

Für die technische Umsetzung wurde schließlich Typescript mit Vue.js verwendet.

Das Browser Plugin ist aktuell nur für Chrome erhältlich. Wenn das Plugin entsprechend genutzt wird werden wir auch eine Version für Firefox erstellen.

hoax.ly im Chrome Webstore: <https://chrome.google.com/webstore/search/hoaxly>



Das Browser Plugin prüft die aktuelle URL und gleicht diese mit dem hoax.ly index ab. Sofern ein Treffer erfolgt wird dies angezeigt und entsprechende Richtigstellungen verlinkt. Werden Richtigstellungen mit negativer Bewertung gefunden leuchtet das Browser Extension Icon rot.

Momentan ist die Verwendung aufgrund der noch eher kleinen Anzahl an Richtigstellungen im hoaxly Index noch eher eingeschränkt, das soll sich im Laufe der Zeit jedoch ändern.

Außerdem sind weitere Maßnahmen geplant, um Feedback zu einer größeren Anzahl von Inhalten geben zu können: Warnung auch wenn Richtigstellungen für eine besuchte Domain existieren bzw. zusätzliche Einbindung von <https://github.com/BigMcLargeHuge/opensources> als Datenquelle für problematische Domains.

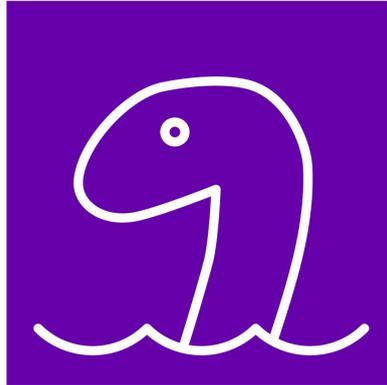
3.6 Arbeitspaket 6 - Chatbots

In der Konzeption wurden die Zielgruppe(n) definiert und Personas erstellt. Anschließend wurde darauf aufbauend folgende Vorgaben für den Chatbot definiert:

- Soll humorvoll und unterhaltsam sein, besonders auf Facebook.

- Soll Vertrauen vermitteln/fördern (-> Transparenz, auf Risiken hinweisen)

Als Avatar wurde schließlich Nessie ausgewählt, das Seeungeheuer von Loch Ness.



Nessie, das Avatar für die chatbots

Inhaltlich wurden folgenden Bereiche definiert und umgesetzt:

- Urls überprüfen lassen (über den hoaxly Index) und Ergebnisse entsprechend verlinken
- hoaxly FAQs anbieten
- Über Fact-Checking informieren (Was ist das, Wie kann man selber nachforschen)
- Kritisches Denken fördern und zum selber Nachforschen (“fact-checking”) animieren

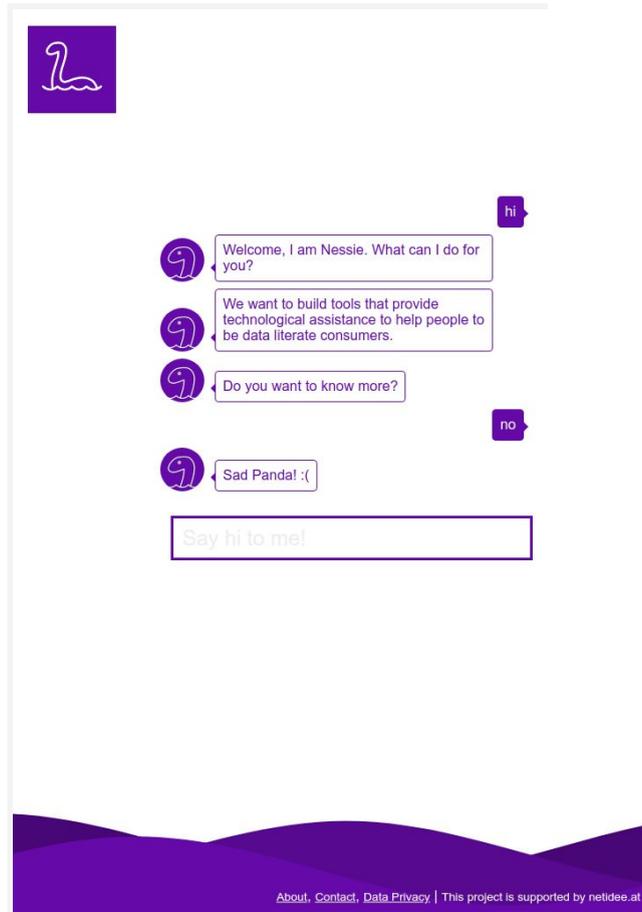
Statt auf externe Services zur Entitäten-Extrahierung und “Intent-Classification” zu vertrauen haben wir uns dazu entschlossen, eine Open-Source Software diesbezüglich selber zu hosten und zu “trainieren. Wir verwenden hierfür [Rasa NLU - the Open Source Conversation AI](#).

Das dazugehörige github Repo ist hier zu finden:

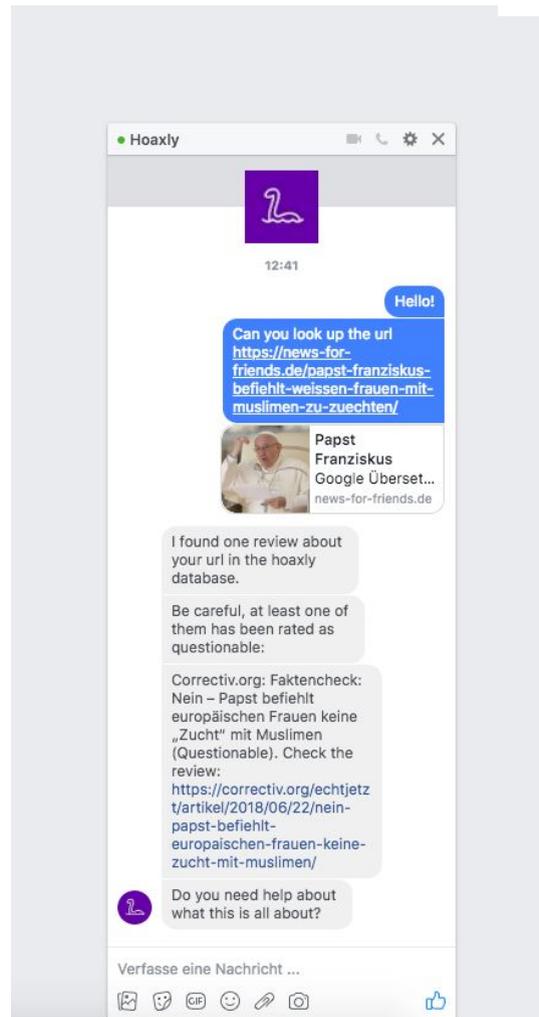
<https://github.com/hoaxly/hoaxly-rasa-nlu-container>

Für die chatbots selber haben wir uns für das [botman framework](#) entschieden. Vor allem deshalb, weil Integrationen für die für uns wichtigen Plattformen bestehen und wir bereits Erfahrung mit PHP und Symfony haben.

Nessie ist aktuell nicht nur auf Telegram, Facebook und Slack zugänglich, sondern auch über das Web: <https://chat.hoax.ly>. Auf Facebook ist der Bot aktuell noch im Review-Status, sollte aber hoffentlich bald freigeschaltet werden.



hoaxlybot für das web (<https://chat.hoax.ly>)



Hoaxlybot für Facebook

Das github-Repo für die chatbots ist <https://github.com/hoaxly/hoaxlybot-container>.

In der Dokumentation für die chatbots sind weitere Bilder und Informationen zu den Chatbots vorhanden:

<https://hoaxly.gitbook.io/documentation/en-user-documentation/using-the-hoax.ly-chatbots>

3.7 Arbeitspaket 7 - Webseiten Installationspaket



Unter dem Namen “debunkCMS” wurde ein Webseiten-Installationspaket erstellt, das auf die beliebte Drupal 8 Distribution [Thunder](#) aufbaut. Thunder wurde von Hubert Burda Media initiiert, und wird von vielen Firmen und Organisationen unter der GPL2 Lizenz weiterentwickelt sowie für kleine und großen Kunden im digitalen Publishing eingesetzt.

Für das debunkCMS hat das den großen Vorteil, dass die in Thunder enthaltenen Features einfach auch zur Verfügung stehen und nicht extra entwickelt werden müssen. Außerdem können Neuentwicklungen regelmäßig in das Projekt einfließen.

debunkCMS bietet folgende Funktionalitäten:

- Einfache Installation
- Mehrsprachigkeit (kann optional hinzugeschaltet werden)
- Vorkonfigurierter Inhaltstyp (review) mit konfigurierten Metatags (og:*, twitter:card, usw), Pfaden, Anzeigen (Teaser und volle Darstellung) und allen relevanten Feldern für Richtigstellungen:
 - field_review_content
 - field_review_item_media
 - field_review_metatags
 - field_review_item_publisher
 - field_review_rating
 - field_review_item_claim
 - field_review_item_authors
 - field_review_item_country
 - field_review_item_language
 - field_review_item_pub_date
 - field_review_item_tags

- field_review_item_title
- field_review_item_type
- field_review_item_url
- field_seo_title
- field_tags
- field_teaser_text
- field_teaser_media
- Vereinfachte Übersichten und Verwaltungsseiten im Backend
- Automatische Ausgabe der Daten im ClaimReview schema.org Standard

```
<script type="application/ld+json">{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "ClaimReview",
      "url": "http://debunk.docksal/review/drupal-and-wikipedia-have-same-birthday",
      "itemReviewed": {
        "@type": "CreativeWork",
        "name": "Tweet by acolono GmbH about the birthday of Drupal and Wikipedia",
        "datePublished": [
          "Mon",
          "01/15/2018 - 12:00"
        ]
      },
      "claimReviewed": "Drupal and Wikipedia have both birthday on Januar 15",
      "author": {
        "@type": "Organization",
        "@id": "http://debunk.docksal/",
        "name": "My Drupal 8 Site",
        "url": "http://debunk.docksal/"
      },
      "datePublished": "2018-06-18T20:19:25+0000",
      "reviewRating": {
        "@type": "Rating",
        "ratingValue": "3",
        "bestRating": "3",
        "worstRating": "1",
        "alternateName": "Positive (+)"
      }
    }
  ]
}</script>
```

- Aktivierte REST API, um auf die Daten bei Bedarf einfach zugreifen zu können (zB zur Integration in die hoaxly Datenbank).
- Halbwegs ansprechende Darstellung der Richtigstellungen im Frontend. Momentan wird das Standard-Theme von Thunder benutzt, mit nur kleinen Anpassungen. Eine schöne Erweiterung wäre ein eigenes Theme, das den Look&Feel einer aktuellen News-Seite per default besser erfüllt.
- Vordefinierte Benutzerrollen und Berechtigungen.
- Beispiel-Content: Ein review Inhalt mit allen relevanten Daten, plus Kategorien, Blöcke, usw.
- Alle Features, die auch von Thunder bereitgestellt werden:
 - story telling,
 - user management,
 - publishing workflows,

- multilingual,
- scheduling content,
- improved media handling,
- Facebook Instant Articles,
- embedding social media posts,
- liveblog,
- device specific previews



debunkCMS mit Beispiel-Content auf iPhone

Wie unsere Analysen der von den Plattformen eingesetzten technischen Lösungen zeigen (fast keine strukturierten Inhalte oder Einsatz von schema.org), gibt es hier großen Bedarf. Das DebunkCMS kann hier sehr gut Abhilfe schaffen.

3.8 Arbeitspaket 8 - Datenqualität

Bei den Kriterien, welche Plattformen eingebunden werden sollen, haben wir uns entschieden, den Standard IFCN (<http://www.poynter.org/fact-checkers-code-of-principles/>) so gut wie möglich zu verwenden. Wie unser Analyse jedoch zeigt, gibt es eher wenige europäische Richtigstellungsplattformen, die sich diesen Richtlinien verbunden fühlen bzw. das auch kommunizieren. Vor allem im deutschsprachigen Raum ist uns mit <https://correctiv.org/echtjetzt/> weiterhin nur eine einzige Plattform bekannt.

Das bedeutet, dass wir entweder Plattformen bitten müssen, dieses Audit durchzuführen. Oder wir führen selber Prüfungen durch, um transparent begründen zu können, dass eine Datenquelle unseren Anforderungen gerecht wird. Das ist ein gehöriger Mehraufwand, für den Erfolg der hoaxy Datenbank jedoch sicher von großer Bedeutung.

Für die Prüfung der Plattformen werden wir das Assessment Dokument von IFCN verwenden: https://drive.google.com/file/d/1oM9hZvCNk8jirwJ2madpCYY09o_igZQE/view?usp=sharing

3.9 Arbeitspaket 9 - Datenquellen

Die Optimierung der Scraper und der für die Anwendungen nötigen Objektstruktur in Elasticsearch hat uns viel Mühe bereitet (bzw. tut es noch immer). Die Importer geben sehr viele Daten zurück, vieles davon ist nicht brauchbar und muss gefiltert bzw. weiterverarbeitet werden.

Bei den Importern ist das folgende Stichwort aktuell zentral: "Friendly scraping". Für da Entwickeln haben wir immer nur einzelne Artikel importiert, da wir sonst riskieren von den Plattformen gesperrt zu werden (bzw. unsere IP). Am Server gilt es hier entsprechende Vorkehrungen zu treffen bzw. zumindest die Best Practices umzusetzen:

Robots.txt Einstellungen der Plattformen überprüfen und akzeptieren, nur so viele Zugriffe wie unbedingt notwendig und Kontaktinformationen mitsenden.

In der Dokumentation wurde eine eigene Seite hierfür erstellt:

<https://hoaxy.gitbook.io/project/polite-scraping>

Außerdem wurde eine eigene E-Mail Adresse eingerichtet, die beim Scraping für Fragen und Beschwerden mitgesandt wird: bot@hoax.ly

Aktuell sind Importer vorhanden für die folgenden Richtigstellungs-Plattformen:

- correctiv.org (Deutschland)
- factcheck.org (Großbritannien)
- politifact.com (USA)
- snopes.com (USA)

- climatefeedback.org/ (USA, Audit noch ausstehend)
- unfake.us (USA, Audit noch ausstehend)

Die Importer (technisch werden diese als Spider bezeichnet) sind in folgendem GitHub Repository einzusehen: <https://github.com/hoaxly/hoaxly-scraping-container>

Da fast jede Plattform eigene Bewertungsmuster hat werden diese Bewertungen in ein normalisiertes Raster eingeordnet. Folgende Tabelle zeigt die vereinfachte Einordnung von hoaxly:

value	label	description / example
3	No issues: Content is rated as not problematic. Credibility high or very high.	The reviewed item is rated as mostly true or true.
2	Neutral: Whether positive nor negative rating. Neutral credibility.	The reviewed item is not rated as true, mostly true, mostly false or false.
1	Questionable: Content is rated as questionable (at least for some aspects). Credibility low or very low.	The reviewed item is rated as mostly false or false.
n/a	No rating	There is no rating.

Wichtig dabei ist dass wir keine Inhalte als wahr oder falsch bezeichnen. Wir geben nur an dass bei einem Check negative Aspekte gefunden wurden und deshalb der Inhalt mit Vorsicht zu behandeln ist.

Das Erstellen von weiteren Importern wird ein Hauptaugenmerk in der nächsten Zeit sein. Je mehr qualitativ gute Inhalte desto besser.

3.10 Arbeitspaket 10 - Tests & Dokumentation

Durchführung von Tests und Feedback einholen, Programmierung der definierten Verbesserungen, Dokumentation finalisieren und überarbeiten

User- und Usability Tests wurden durchgeführt und daraus resultierende Erkenntnisse für Verbesserungen der Software bzw. Benennungen/Anweisungen verwendet. Wir werden mit der Veröffentlichung der Applikationen sicher noch einiges an Feedback erhalten, das wir noch einarbeiten werden.



Die Dokumentation steht sowohl in den jeweiligen Repositories, als auch über gitbook zur Verfügung.

Dokumentation für EntwicklerInnen:

<https://hoaxly.gitbook.io/documentation/developer-documentation>

Dokumentation für AnwenderInnen:

<https://hoaxly.gitbook.io/documentation/en-user-documentation>

hoaxly github Seite mit allen Reposities: <https://github.com/hoaxly>

Darüber hinaus wurden entsprechende Dokumente bzgl. Data Privacy und Nutzungsbedingungen erstellt:

<https://hoaxly.gitbook.io/documentation/en-user-documentation/terms-of-use>

https://hoax.ly/privacy_policy.html

3.11 Arbeitspaket 11 - Öffentlichkeitsarbeit

Die Webseite ist online und enthält alle wichtigen Infos bzw. Links: <https://hoax.ly>

Darüber hinaus ist hoaxly auch auf Facebook und Twitter vertreten:

https://twitter.com/hoax_ly

<https://www.facebook.com/Hoaxly-162897680835820>

Bezüglich Öffentlichkeitsarbeit ist darüber hinaus vor allem unsere Präsenz auf der Ars Electronica mit dem Ars Electronica Fake News Game zu nennen, sowie Vorträge und Diskussionen auf Barcamps als auch das Ö1-Interview zum Thema letztes Jahr.

Beim netidee OpenSource Community Camp gewannen wir mit unseren Pitch den dritten Preis und erhielten dadurch eine Erwähnung auf <https://www.derbrutkasten.com/>.



Preisverleihung netidee OpenSource Community Camp (Foto: Arman Rastegar)

Besonders gefreut hat uns die darauf folgende Einladung zur Ars Electronica im September 2017. Diese Chance mussten wir nutzen, um Feedback zu sammeln und Kontakte zu knüpfen. Hierfür haben wir eine eigene Anwendung programmiert, das hoaxly Fake News Game:

<https://www.aec.at/ai/en/hoaxly/>

Das hoaxly Fake news Game ist eine interaktive Installation, in der Schlagzeilen nach dem vermuteten Wahrheitsgehalt bewertet werden. Die SpielerInnen lesen Schlagzeilen, treffen Bewertungen und erhalten umgehend Rückmeldung über den Ursprung der Nachricht. Die Analysen von Plattformen, die Falschmeldungen im Internet aufdecken, zum Beispiel Mimikama oder Snopes, erscheinen auf dem Bildschirm, inklusive Statistiken, wie oft Meldungen in sozialen Medien geteilt wurden. Das Game ist eine Kombination aus zwei netidee Projekten: OpenTrigger (Voting über kabellose Buttons) und hoaxly.

Hierfür wurden etwa 40 unterhaltsame Meldungen ausgewählt. Das Spiel ist bei den BesucherInnen sehr gut angekommen, insgesamt wurden bei der Ars Electronica 1,234 Einschätzungen abgegeben. Nur rund zwei drittel (63,45%) davon waren richtig eingeschätzt. Ein Indiz dafür, dass es nicht leicht ist, den Wahrheitsgehalt von Informationen einzuordnen.



Das hoaxly fake news game bei der Ars Electronica 2017 (Foto: Nico Griener)

Ein Blogpost über die Auswertung der Ergebnisse wurde veröffentlicht, hier eine Übersicht über die am meisten falsch oder richtig eingeschätzten Meldungen:

true: votecorrect: Descending		false: votecorrect: Descending	
articleheading.keyword: Descending	Count	articleheading.keyword: Descending	Count
AfD-Politiker Podeswa fordert Frauenverbrennungen gegen Klimawandel	36	Fleischallergie nach Zeckenbiss	33
Volvo: Kängurus lassen selbstfahrende Autos verzweifeln	31	Frau täuschte Blindheit vor - 28 Jahre lang	23
Richter lobt Drogendealer für Geschäftsmodell	31	Kondome mit Luthers Segen	22
Polizei belagert 7 Stunden lang leeres Haus	30	Drug-sniffing lizard joins Arizona police force	21
Mehrere Waffen in Versteck auf Donauinsel gefunden	28	3 gleichnamige Kandidaten bei Rumänischer Wahl	17
3 gleichnamige Kandidaten bei Rumänischer Wahl	26	Mehrere Waffen in Versteck auf Donauinsel gefunden	15
Drug-sniffing lizard joins Arizona police force	21	Polizei belagert 7 Stunden lang leeres Haus	13
Kondome mit Luthers Segen	21	Volvo: Kängurus lassen selbstfahrende Autos verzweifeln	12
Frau täuschte Blindheit vor - 28 Jahre lang	20	Richter lobt Drogendealer für Geschäftsmodell	12
Fleischallergie nach Zeckenbiss	9	AfD-Politiker Podeswa fordert Frauenverbrennungen gegen Klimawandel	7

Abb.: hoaxly fake news game Auswertung: Die am meisten falsch oder richtig eingeschätzten Meldungen

Der Code hierfür wurde auf github veröffentlicht: <https://github.com/piccaso/HoaxBuzzer>

Die Daten über die Einschätzungen der TeilnehmerInnen sind hier einsehbar:

<https://drive.google.com/open?id=1IQq6o8qjCJKqJnnxDfWfmD9q9VxnXICo>

Blogposts und Socialmedia-Posts wurden mehr oder weniger regelmässig erstellt. Mit der der aktuellen Veröffentlichung der Anwendungen werden wieder mehr Inhalte veröffentlicht werden können, erstens über Vorstellungen der einzelnen Anwendungen und zweitens über vorgenommene Updates, die neue Features beinhalten.

Generell wurde in diesem Arbeitspaket weniger Zeit aufgewandt als geplant, alleine deswegen weil die Fertigstellung der Anwendungen so lange auf sich warten hat lassen und alle Ressourcen in die Entwicklung geflossen sind.

Mit der erstmaligen Veröffentlichung der Anwendungen wird es auch einfacher potentielle Partner anzusprechen, weil Ziele und Technologie einfacher zu erklären sind, wenn diese bereits zur Verfügung stehen. Ideen für mögliche Partnerschaften gibt es jedenfalls genug.

Es wird sich in diesem Bereich also sicher noch viel tun. So werden wir auch dieses Jahr wieder auf der Ars Electronica vertreten sein.

4 Liste Projektergebnisse

1	Projektendbericht	CC-BY-3.0 AT	https://www.netidee.at/hoaxly
2	Entwickler-DOKUMENTATION	CC-BY-3.0 AT	https://www.netidee.at/hoaxly bzw. https://hoaxly.gitbook.io/documentation/developer-documentation
3	Anwender-DOKUMENTATION	CC-BY-3.0 AT	https://www.netidee.at/hoaxly bzw. https://hoaxly.gitbook.io/documentation/en-user-documentation
4	Veröffentlichungsfähiger Einseiter	CC-BY-3.0 AT	https://www.netidee.at/hoaxly
5	Datenbank mit Metainformationen zu vorhandenen Richtigstellungen	ODbL 1.0	https://api.hoax.ly/api bzw. https://hoax.ly/api.html
6	Browser-Plugin für Chrome	MIT	https://github.com/hoaxly/hoaxly_browser_extension
7	Webseiten-Installationspaket	GPL2	https://github.com/hoaxly/debunkCMS
8	Kriterienkatalog für Quellen	CC-BY-3.0 AT	https://hoaxly.gitbook.io/documentation/developer-documentation/adding-new-sites-to-the-database/sources-assessments
9	Chatbots für Facebook, Telegram und Slack: RASA NLU Integration (Apache License) Hoaxlybot Container (MIT License)	Apache bzw. MIT	https://github.com/hoaxly/hoaxlybot-container und https://github.com/hoaxly/hoaxly-rasa-nlu-container

10	<i>Importer für die wichtigsten Richtigstellungs-Plattformen.</i>	MIT	https://github.com/hoaxly/hoaxly-scraping-container
----	---	-----	---

5 Verwertung der Projektergebnisse in der Praxis

Die Browserextension kann installiert werden, um Warnmeldungen zu erhalten, wenn man eine Seite mit zweifelhaften Informationen ansurft. Die Qualität der Rückmeldungen ist direkt proportional zur Menge der in der hoaxly Datenbank gespeicherten Falschmeldungs-Urls mit dazugehörigen ClaimReviews. Aus diesem Grund wird das Hauptaugenmerk in der öffentlichen Startphase darauf liegen, möglichst viele Menschen mithilfe einer leicht verständlichen Doku dazu zu motivieren, Scraper zu bauen, um den Datenbestand zu erweitern.

Auch das debunkCMS kann einen Beitrag leisten, um den Datenbestand der hoaxly-db zu verbessern. denn wenn es von einer Seite benutzt wird, die Richtigstellungen veröffentlicht, dann können die Claim Reviews dieser Seite wiederum einfach in die hoaxly DB integriert werden.

Auch die hoaxly chatbots haben viel Potential um einen Beitrag zu leisten BenutzerInnen über fact-checking zu informieren und User anzuspornen, kritisch zu sein und selber Nachforschungen zu betreiben.

6 Öffentlichkeitsarbeit/ Vernetzung

Zusammenarbeit mit möglichst vielen Richtigstellungs-Seiten im deutschsprachigen Raum. Wenn es möglich ist, auf einfachem Weg die Daten einer Seite strukturiert aufbereitet zu bekommen, dann wird neben dem Rating einer URL auch der Link zu der Seite / den Seiten mit der Richtigstellung angezeigt. hoaxly soll nicht als Konkurrenz empfunden werden, sondern als technische Unterstützung um die eigenen Artikel zu verbreiten und das gemeinsame Anliegen (weniger Verbreitung von eindeutig falschen Informationen) zu unterstützen.

Nicht zuletzt um möglichst viel gutes Datenmaterial für hoaxly zu bekommen, wird Seiten, die Richtigstellungen ins Netz stellen (zuerst im deutschsprachigen Raum), das debunkCMS angeboten, um ihre Daten freier verfügbar zu machen.

Geplant sind weitere Kooperation mit Projekten wie mimikama.at, kobuk.at, Watchlist Internet u.ä., nicht nur um deren Datensätze in die Datenbank aufzunehmen, sondern auch um die Tools an ihre Zielgruppen zu kommunizieren.

Relevante Journalisten werden kontaktiert und über das Projekt informiert.

Auf der Ars Electronica sind wir auch dieses Jahr wieder präsent und werden u.a. hoaxly promoten.

7 Geplante Aktivitäten nach netidee-Projektende

- Weitere Integration von qualitativ hochwertigen Quellen
Der hoaxly Index ist noch am Wachsen. Je mehr guter Content desto besser.
- Deutsche Sprache
Momentan sind die Tools (Chatbots, Browser Extension) und die Dokumentation in englischer Sprache gehalten. Es wäre jedoch absolut sinnvoll die angebotene Funktionalität auch in deutscher Sprache anzubieten.
- Laufende Verbesserung der Chatbot-Antworten
Wir möchten die Inhalte des Chatbots noch ausweiten (konkretere Infos über das eigenhändige Erstellen von fact-checks, Allgemeine Infos zum Thema fact-checking, usw.). Laufendes Training des chatbots um immer besser passende Antworten zu geben.
- Ausweitung der API auf Abfrage nach Zitaten
Aktuell können nur URLs abgefragt werden. Eine Abfrage von Zitaten oder Stichwörtern wäre sehr brauchbar.
- Browser Extension ausbauen
Warnung wenn Richtigstellungen für eine besuchte Domain existieren bzw. zusätzliche Einbindung von <https://github.com/BigMcLargeHuge/opensource>, damit mehr Suchtreffer erfolgen. Auch für Firefox veröffentlichen.
- Ausweitung der Suche und Nutzung semantischer Technologien (optional)
Nicht alle Falschmeldungen haben eine eindeutige URL. Sehr oft verbreiten sich falsche Gerüchte über unterschiedlichste Formate und Kanäle. Ein automatischer Check von Inhalten ist allerdings eine völlig andere Herausforderung, die nicht am Einsatz semantischer Technologien vorbeikommt. hoaxly um diese inhaltliche Komponente zu erweitern wäre durchaus ein Plan für die Zukunft.

8 Anregungen für Weiterentwicklungen durch Dritte

Der erste und dringendste Aufruf an Dritte ist der, mehr scraper zu bauen und die hoaxly-DB mit möglichst vielen URLs von Falschmeldungen und den ClaimReviews der dazugehörigen Richtigstellungen zu füttern.

Das debunkCMS kann von allen genützt werden, die redaktionell an Richtigstellungen arbeiten und wollen, dass die von Ihnen erarbeiteten Informationen von möglichst vielen Menschen (und Bots!) genützt werden können.

Alle Komponenten von hoaxly können unter Einhaltung der OS-Lizenzen frei weiterentwickelt werden und möglichst oft und vielfältig eingesetzt werden.