„IT mit KI“ – Crawlen von Website-Inhalten für KI-Chatbots

Eine Anleitung für Nicht-IT-ler

Uwe Dierolf

Abstract

Seit 2025 betreibt die KIT-Bibliothek den KI-Service-Chatbot BibKI. Er kann über die Homepage der KIT-Bibliothek oder die URL https://chatbot.bibliothek.kit.edu aufgerufen werden. Seine Funktionsweise wurde in der Zeitschrift „Information, Wissenschaft und Praxis“ (IWP) (https://degruyter.com/document/doi/10.1515/iwp-2025-2002/html) beschrieben.

Beim Start des Dienstes im Januar 2025 bestand das Datenmaterial für die lokale Wissensbasis des Bots ausschließlich aus selbst erstellten, mit Markdown versehenen Textdokumenten.

Inzwischen liegen weitere Erfahrungen mit Web-Crawling vor. Dieser Beitrag erklärt den Einsatz von Web-Crawlern exemplarisch am Beispiel des github-Projekts GPT-Crawler, mit dem der gesamte Content einer Website als Grundlage für einen solchen KI-Service-Chatbot in einem Lauf ermittelt werden kann. Dieser Beitrag demonstriert, wie man diesen einfachen Crawler nutzen kann. Weiterhin wird gezeigt, wie man auch als Nicht-IT-ler einen solchen Crawler aufsetzen und lokal inkl. aller benötigten Tools mit Hilfe von ChatGPT & Co installieren kann. Damit sind die Voraussetzungen geschaffen, Datenmaterial für den eigenen Chatbot regelmäßig zu aktualisieren.

The KIT library has been operating the AI service chatbot BibKI since 2025. It can be accessed via the KIT library homepage or the URL https://chatbot.bibliothek.kit.edu. Its functionality was described in the journal “Information, Wissenschaft und Praxis” (IWP) (https://degruyter.com/document/doi/10.1515/iwp-2025-2002/html).

When the service was launched in January 2025, the data material for the bot's local knowledge base consisted exclusively of self-created text documents in Markdown. Further experience with web crawling is now available. This article explains the use of web crawlers using the example of the github project GPT-Crawler, with which the entire content of a website can be determined in one run as the basis for such an AI service chatbot.

This article demonstrates how to use this simple crawler. It also shows how non-IT specialists can set up such a crawler and install it locally, including all the necessary tools, with the help of ChatGPT & Co. This creates the prerequisites for regularly updating data material for your own chatbot.

Die Idee des KI-Service-Chatbots BibKI

BibKI hat die Aufgabe, Nutzerfragen in beliebigen Sprachen rund um die Uhr zu beantworten. Er eignet sich nicht, um Literaturrecherchen im Katalog zu unterstützen. In der Regel antwortet der Chatbot dann in der Sprache, in der die Frage gestellt wurde.

Zur Beantwortung dient öffentlich zugängliches Datenmaterial, das man dem Bot hochgeladen hat.

Diese Art von Anwendung nennt man RAG (Retrieval Augmented Generation). Sowohl die semantische Suche nach dem passenden Dokument zur Beantwortung der Frage als auch die Erstellung der Antwort erledigt ein OpenAI-Assistant. Es kommt das kostengünstige Sprachmodell 4o-mini zum Einsatz.

KI-Chatbot BibKI

Die Aufgabe von BibKI beschränkt sich dabei auf die Vermittlung der Fragen an den OpenAI-Assistant und der Darstellung der Antwort, die dieser liefert. BibKI kann man sich als Agent vorstellen, der im Hintergrund mit OpenAI kommuniziert. Das geschieht datenschutzkonform, da OpenAI keinerlei personenbezogene Daten erhält. BibKI schreibt alle Fragen in Logfiles, so dass das Bibliothekspersonal diese auswerten kann.

Die nicht beantworteten Fragen werden dabei in einem separaten Logfile „unanswered_questions“ gespeichert. So kann man sehr schnell Lücken im Datenmaterial erkennen und umgehend nachbessern, damit ähnliche Fragen beim nächsten Mal vom Bot beantwortet werden können. Dieses Logfile stellt eine sehr große Hilfe beim Betrieb von BibKI dar. Beim Betrieb einer Website zeigen Zugriffsstatistiken nur, wie oft eine Seite aufgerufen wurde, man erkennt so aber noch lange nicht, ob diese Seite hilfreich war.

Eigene Dokumente vs. gecrawlter Content

Die Qualität der Antworten, die ein Service-Chatbot auf Nutzerfragen geben kann, hängt von mehreren Faktoren ab. Ein wesentlicher Aspekt ist die Qualität der verwendeten Retrieval-Augmented-Generation (RAG)-

Pipeline. Dabei spielt insbesondere die Güte der Ähnlichkeitssuche (Retrieval) eine Rolle, mit der relevante Dokumente zur Beantwortung einer Frage ausgewählt werden. Nicht alle Dokumenttypen eignen sich gleichermaßen gut für RAG-Anwendungen.

Auch die Struktur und Lesbarkeit der zugrunde liegenden Dokumente beeinflussen die Qualität der generierten Antworten. Klar strukturierte Texte – zum Beispiel solche, die mit Markdown-Syntax formatiert sind (Wikipedia: Markdown https://de.wikipedia.org/wiki/Markdown) – können dem System helfen, relevante Inhalte einfacher zu erkennen und zu verarbeiten. Solche Formate führen in der Praxis häufig zu besseren Ergebnissen.

Wer bereits eine eigene Website betreibt und diese aktuell hält, stellt sich zu Recht die Frage, ob eine zusätzliche, redundante Datenhaltung tatsächlich durch einen möglichen Qualitätsgewinn gerechtfertigt ist. Aus diesem Grund wird in vielen Fällen eine redundanzfreie Lösung bevorzugt.

Ein vielversprechender Ansatz ist daher der Einsatz von Web-Crawlern, um Inhalte direkt von einer bestehenden Webpräsenz automatisiert zu extrahieren.

Web-Crawler-Projekt „GPT-Crawler"

Auf Github findet sich unter der URL https://github.com/BuilderIO/gpt-crawler das Projekt „GPT-Crawler". Es handelt sich dabei um einen sehr einfach zu nutzenden Crawler, der alle Seiten einer Web-Präsenz auf einmal abrufen und deren Inhalte in einer Datei abspeichern kann. Diese Datei liegt im JSON-Format vor (https://de.wikipedia.org/wiki/JSON).

Es handelt sich dabei zwar um ein Format, das von Menschen nicht sehr gut gelesen werden kann, ein OpenAI-Assistant bzw. eine RAG-Anwendung im Allgemeinen kann solche Daten jedoch gut verarbeiten. Für jede vom Crawler gelesene Web-Seite wird Titel, URL und das HTML erfasst und in der JSON-Datei als ein Datensatz gespeichert.

Wir schauen uns nun im folgenden Kapitel an, wie man selbst als Endanwender und Nicht-IT-ler einen solchen Crawler aufsetzen und benutzen kann. Dessen Output kann man auch z.B. für den Aufbau eines eigenen Custom GPT verwenden, sofern man über einen kostenpflichtigen ChatGPT-Account verfügt.

Installation des GPT-Crawlers – eine Anleitung für IT-Laien

Wie wir alle wissen, kann KI sehr gut mit Sprache umgehen. Außerdem kann sie Sachverhalte für Nutzer mit unterschiedlichen Vorkenntnissen so aufbereiten, dass auch komplizierte Sachverhalte gut nachvollzogen werden können.

Warum sollte man also nicht auch als Endnutzer KI einsetzen, um sich erklären zu lassen, wie Software – in diesem Fall der GPT-Crawler und die dafür nötigen Basis-Tools – installiert und benutzt werden kann.

GPT-Crawler (https://github.com/BuilderIO/gpt-crawler)

Wer die github-Seite besucht, findet unter „Get started" den schönen, für die meisten jedoch völlig unverständlichen Hinweis, dass man zuerst einmal das Repository mit „git clone" herunterladen soll. Spätestens beim Lesen dieses Satzes dürften 99% aller Endanwender abgehängt werden.

KI kann technik-affinen Menschen jedoch helfen, auch solche Hürden zu meistern. Sofern man die Berechtigung hat, auf seinem PC Software installieren zu dürfen und ein wenig Mut mitbringt, etwas gänzlich Neues auszuprobieren, kann man daher z.B. mit folg. Prompt starten (da KI die Schreibweise egal ist, wird auf Groß-Klein-Schreibung kein Wert gelegt):

> *„ich möchte, dass du mir bei der installation des gpt-crawler projekts von github unter URL https://github.com/BuilderIO/gpt-crawler hilfst. Ich bin kein IT-ler, daher musst du mir genau erklären, wie ich die notwendigen tools für die installation auf meinem windows pc installieren kann"*
> *[ggfs. ist windows pc durch mac zu ersetzen]*

Die KI erläutert dann, dass man zuerst einige Basis-Komponenten (Tools) installieren muss:

> *Installation von Node.js = https://nodejs.org/en*
> *Installation von Git = https://git-scm.com/downloads/win*

Wenn man zum ersten Mal überhaupt Software installiert, kann z.B. folg. Frage an die KI helfen:

> *„ich habe die oben genannten tools node und git heruntergeladen. was muss ich nun tun, damit ich sie installieren kann und wie nutze ich sie? in der anleitung steht, dass ich git clone eintippen soll. dann soll ich „npm i" aufrufen und an der config. ts veränderungen vornehmen. bitte erkläre es mir so, dass ich auch als nicht it-ler verstehe, was ich tun muss"*

Mit Glück erstellt ChatGPT am Ende der Antwort sogar eine Zusammenfassung wie diese hier:

Zusammenfassung:

Schritt	Was du tust
1. Tools installieren	Git und Node.js Doppelklick → Installieren
2. Projekt klonen	`git clone https://...`
3. Projektordner öffnen	`cd gpt-crawler`
4. Abhängigkeiten laden	`npm install`
5. Konfiguration anpassen	In VS Code Datei `config.ts` bearbeiten
6. Crawler starten	`npm start`

Schritte zur Installation des GPT-Crawlers

Optional kann man sich noch erklären lassen, mit welchem Programm man die Konfigurationsdatei des Crawlers editiert. Hier kennen die meisten Endanwender nur Textverarbeitungsprogramme wie Word oder den Editor „notepad". Mit folgendem Prompt kann man sich jedoch auch dafür ein besser geeignetes Werkzeug empfehlen lassen:

> *„gibt es ein gutes programm, mit dem ich mir den source code des gpt-crawlers schön anzeigen lassen kann?"*

Die KI empfiehlt dann meistens die Installation von Visual Studio Code https://code.visualstudio.com/
So gerüstet, kann man nun versuchen, seine eigene Webpräsenz herunterzuladen. Dazu muss lediglich in der config.ts die eigene Website-URL eingetragen werden.
Eine sinnvolle, minimalistische config.ts sieht dann wie folgt aus.

```
export const defaultConfig: Config = {
  url: „https://meine-domaine.de/",
  match: „https:// meine-domaine.de/**",
  maxPagesToCrawl: 10,
  outputFileName: „crawler-output.json",
  maxTokens: 200000,
  resourceExclusions: [‚png',‚jpg',‚jpeg',‚gif',‚css',‚js',‚pdf']
};
```

Sollten Sie Teile davon nicht verstehen, lassen Sie sich die Bedeutung von KI erklären.

Bitte beachten Sie beim Testen folgenden wichtigen Hinweis!

Da in den letzten Wochen und Monaten die Belastung der Webserver von öffentlich zugänglichen Websites durch z.T. sehr aggressive LLM-Crawler stark zugenommen hat, sollte man moderat vorgehen und sich dessen bewusst sein, dass der Einsatz eines Crawlers, der (rekursiv) alle Seiten einer Webpräsenz herunterlädt eine – wenn auch nur temporär – starke Belastung für den Webserver bedeutet. Tests sollten daher tunlichst nur zu Randzeiten durchgeführt werden. Sie riskieren ansonsten ggfs. eine befristete Sperrung Ihrer IP durch eine Firewall.

Erste Erfahrungen

Das Crawlen von Websites ist dank freien Crawlern wie dem GPT-Crawler einfach.
Es kann jedoch je nach verwendetem Content Management System (CMS) mehr oder weniger schwierig sein, dass man alle relevanten Inhalte erfasst bzw. ständig wiederkehrende Texte wie z.B. Headerinhalte, die auf jeder Seite einer Website auftauchen, auszublenden. Auch das Entfernen von unnötig vielen Leerzeichen oder Zeilenumbrüchen kann sinnvoll sein, damit die ermittelten Dokumente überschaubar groß bleiben und für Menschen besser lesbar sind.
In den von der KIT-Bibliothek durchgeführten Tests mit dem GPT-Crawler wurden daher einige wenige Anpassungen am Code des GPT-Crawlers durchgeführt.
Den geänderten Code dazu hat in vielen Fällen die KI erstellt. Wir haben lediglich das Problem in Form von Prompts geschildert und dann die Datei des Projektes, die für das Laden von Inhalten zuständig ist (core.ts) von der KI ändern lassen.
Da es sich um Open-Source-Software handelt, konnte einfach der Code der core.ts hochgeladen und im Prompt angegeben werden. Diese Vorgehensweise hat in allen Fällen geklappt. Besonders gute Ergebnisse lieferte dabei das Reasoning-Modell o3-mini, das von ChatGPT bzw. OpenAI angeboten wird. Jedoch kann auch Claude 3.x oder Perplexity etc. genutzt werden.
Das Anpassen des Source Code mit Hilfe von KI stellt für Endanwender die Königsklasse dar. Solche Änderungen sollten Sie daher anfangs am besten gemeinsam mit Ihren Kollegen aus der IT durchführen.
Sofern Sie Mitglied des DFN sind, können Sie auch die Sprachmodelle, die von der Gesellschaft für wiss. Datenverarbeitung (GWDG) im Rahmen der Academic Cloud mit dem Service „Chat AI" angeboten werden (https://chat-ai.academiccloud.de/) nutzen. Diese LLMs können sehr gute Dienste leisten.
Kleine, speziell für Coding optimierte Modelle wie Qwen 2.5 Coder können sogar lokal mit Ollama (https://ollama.com/) benutzt werden und liefern zügig Ergebnisse.
Ein anderes sehr vielversprechendes Crawler-Projekt ist Crawl4AI (https://github.com/unclecode/crawl4ai). Dieser Crawler bietet wesentlich mehr Einstellungen und Konfigurationsmöglichkeiten. Zugunsten der Verständlichkeit für Endanwender wurde Crawl4AI in diesem Beitrag nicht berücksichtigt.

Fazit und Ausblick

Die Beschaffung von Datenmaterial in Form von Content einer gepflegten Webpräsenz mit freier Crawler-Software stellt eine brauchbare Alternative zur redundanten Erfassung und Pflege von Dokumenten dar. Diese meist maschinenlesbaren Daten bilden eine solide Basis für den Aufbau eines eigenen Service-Chatbots.

Sie können auch die Grundlage für BibKI-Hosting sein. Diesen Weg haben in Karlsruhe sowohl die Badische Landesbibliothek als auch die Stadtbibliothek Karlsruhe gewählt. Beide verwenden das weit verbreitete CMS „Typo3“, wenn auch in unterschiedlichen Versionen. Das Crawlen erforderte nur geringe Anpassungen am GPT-Crawler. Es wurden zwei BibKI-Instanzen zum Testen aufgesetzt und beide Einrichtungen fanden die Ergebnisse so überzeugend, dass die KIT-Bibliothek nun BibKI für beide Partnerbibliotheken hostet.

Auch hybrides Vorgehen kann angewendet werden. So können die JSON-Dateien, die der Crawler erzeugt z.B. durch eigene Text-, Word- oder PDF-Files ergänzt werden.

Wichtig ist beim Einsatz von externen KI-Hostern wie OpenAI, die im Fall von BibKI im Backend eingesetzt werden, dass nur öffentlich zugängliches Material, das von anderen Bots auch erreicht und abgezogen werden kann, benutzt werden darf. Das muss auch beim Crawlen beachtet werden. Sollten Sie sich nicht sicher sein, dann crawlen Sie am besten von einem externen Netzwerk und nicht aus dem eigenen LAN und bedenken Sie immer die Last, die beim eigenen Webserver in diesem Moment entsteht.

Der Beitrag hat vielleicht dem ein oder anderen Leser eine Möglichkeit aufgezeigt, wie man mit Hilfe von KI und dem Schreiben von Prompts auch technisch komplizierte IT-Aufgaben, in dem Fall die Installation von Software meistern kann.

Fragen beantworte ich gerne auch telefonisch unter 0721-608-46076 oder per Mail an Uwe.Dierolf@kit.edu

Anlage:

Mein ChatGPT-Chatverlauf „GPT-Crawler Installation“ https://chatgpt.com/share/67e12436-63f8-8002-8268-1bdd40ce118d

Uwe Dierolf
ist Diplom-Informatiker und ist seit 2001 Leiter der IT-Abteilung an der KIT-Bibliothek in Karlsruhe. Er beschäftigt sich seit dem Erscheinen von ChatGPT mit KI und hat einige Praxis-Workshops u.a. bei der DGI zu diesem Thema gegeben.