Richtlinie zum „Software-Sprint“

OpenResearch – Open Research on COVID-19

Schlussbericht

Zuwendungsempfänger:

OpenResearch COVID19 - de Araújo, Hentschel & Blonska GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS20S28 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)?

Obwohl SARS-CoV-2 ein neues Virus darstellt, wurden schon mehr als 1 Mio. wissenschaftliche Publikationen weltweit hierzu und zu COVID-19 verfasst. Daher besteht ein dringender Bedarf an automatisierten Lösungen, die der wissenschaftlichen Gemeinschaft helfen können, relevante Informationen zu finden und ineinander verflochtene Terminologien in Dokumenten sinnvoll zu verwenden. In dieser Hinsicht zielt dieses Projekt (Förderkennzeichen: 01|S20S28) darauf ab, eine Infrastruktur (RESTful API) aufzubauen - die sogenannte OpenResearch API - um Experten aufschlussreiche Ergebnisse über die Inhalte der wissenschaftlichen Artikel zu SARS-CoV-2 und COVID-19 zu liefern Hierzu werden auf der API KI-Dienste, welche auf Sprachmodellen basieren, verfeinert und können mit Drittsystemen gekoppelt werden. Die Anwendungsfälle umfassen die Verbesserung von Aggregatoren, Chatbots, Dashboards usw., die mit den, von der OpenResearch-API, bereitgestellten Diensten verknüpft sind.

Die anfängliche Einrichtung der Plattform beinhaltete 3 Hauptdienstleistungen, die in der Lage waren das Training von Modellen und die Analyse von Datensätzen mit spezifischen NLP-Ansätzen durchzuführen. Das für die Entwicklung der OpenResearch API verantwortliche Team hat während der Laufzeit der Förderung nicht nur von wertvollen Rückmeldungen aus dem Solution Enabler Network[1] profitiert, sondern auch mit der ZB-MED[2] zusammengearbeitet.

Während des 6-monatigen Zeitrahmens wurden die folgenden Arbeitspakete durchgeführt:

AP1) Softwarearchitektur & -infrastruktur

[1] https://wirvsvirus.org/solution-enabler/
[2] Deutsche Zentralbibliothek für Medizin (ZB MED) - Leibniz-Informationszentrum Lebenswissenschaften, Bereich: Anwendungsorientierte Forschung & Wissensmanagement.

AP2) Content Management System API
AP3) Event-Streaming System
AP4) AI Servicemodul 1: Topic Modeling mit Latent Dirichlet Allocation;
AP5) AI Servicemodul 2: Q&A mit BiotBERT;
AP6) AI Servicemodul 3: NER mit BioBERT;
AP7) Etablierung Socialmedia Kanäle (7.1) & Webseite (7.2)
AP8) Projektmanagement

Die Fertigstellung der Arbeitspakete (APs) wurde wie folgt gestaltet:

A. Mai - Gesamte Softwarearchitektur und -infrastruktur; Social Media
- AP1.1, AP1.2, AP1.3, AP7.1
B. Juni - Content Management System API; Task Queue System; Webseite
- AP2.1, AP2.2, AP3.1, AP7.2
C. Juli - AI Servicemodul 1
- AP4.1, AP4.2, AP4.3, AP4.4
D. August - AI Servicemodul 2
- AP5.1, AP5.2, AP5.3, AP5.4
E. September - AI Servicemodule 3
- AP6.1, AP6.2, AP6.3, AP6.4
F. Oktober - Tests & Optimierung; Veröffentlichung der API
- AP4.5, AP5.5, & AP6.5

Beitrag des Projektes zu den Zielen der Förderinitative „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?

Unsere Zielgruppe sind vor allem Institutionen mit sogenannten Domainexperten, die einen schnellen Überblick über die vorhandene Fachliteratur erstellen und wissenschaftliche Texte zu SARS-CoV-2 und COVID-19 analysieren möchten. Wie zum Beispiel das ZB-MED. Mit welchem wir - im Speziellen dem Bereich der Anwendungsorientierten Forschung und Wissensmanagement - kooperiert haben.

Wir sehen auch die Möglichkeit, die Nützlichkeit der Plattform auf andere Sektoren zu erweitern, indem eine große Vielfalt von Diensten zur API hinzugefügt wird, die über den Themenkomplex SARS-CoV-2 und COVID-19 hinausgehen.

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?

Die OpenResearch API ist ein erweiterbarer Cloud-Service, der Forschungseinrichtungen die Möglichkeit bietet, Machine-Learning (ML)-Modelle zu trainieren, fein anzupassen und einzusetzen. Die anfänglichen Meilensteine, die zu Beginn des Projekts gesetzt wurden, wurden insgesamt erfolgreich erreicht. Im Laufe der Entwicklung haben wir jedoch beschlossen, einen der ML-Dienste, namentlich Doc2Vec, durch Named-Entity-Recognition mit BioBERT zu ersetzen. Die Änderung wurde unternommen, um besser auf

die Bedürfnisse unserer Zielgruppe eingehen zu können, welche sich durch die Zusammenarbeit mit unserem Kooperationspartners herauskristalisierte[3].

In dieser Hinsicht bietet die API zur Zeit Topic Modeling mit Latent Dirichlet Allocation (LDA), Named-Entity-Recognition mit BioBERT (NER + BioBERT) und Frage & Antwort mit BioBERT (F&A + BioBERT). Sobald diese Modelle online veröffentlicht sind, bietet die API Möglichkeiten zur nahtlosen Integration mit Leistungen von Drittanbietern, so dass die folgenden Anwendungsfälle ermöglicht werden:

- Klassifizierung von Tausenden von Dokumenten in automatisch identifizierte Kategorien;
- Semantische Informationssuche in natürlicher Sprache;
- Extraktion von Schlüsselwörtern (z.B. Gen/Protein; Krankheit; Wirkstoff/Chemikalie; Spezies; Mutation).

Die verschiedenen API-Module sind als Microservices angeordnet, welche die folgenden Vorteile bieten:

- **Hohe Erweiterbarkeit:** Alle API-Komponenten sind unabhängig voneinander einsetzbar, eigenverantwortlich, dezentralisiert und lose gekoppelt. In dieser Hinsicht lässt sich die API leicht erweitern, indem neue ML-Dienste, die ähnliche Merkmale aufweisen, bereitgestellt und registriert werden.
- **Hohe Skalierbarkeit:** Da sie auf einem Container-Orchestrierungssystem läuft, werden neue Instanzen von ML-Services in Abhängigkeit vom Ressourcenverbrauch schnell bereitgestellt.
- **Hohe Prozesskoordination:** Die gesamte Kommunikation zwischen den verschiedenen API-Komponenten erfolgt über eine Reihe von Event-Streaming-Systemen(Queues und Push-Notifikationen).

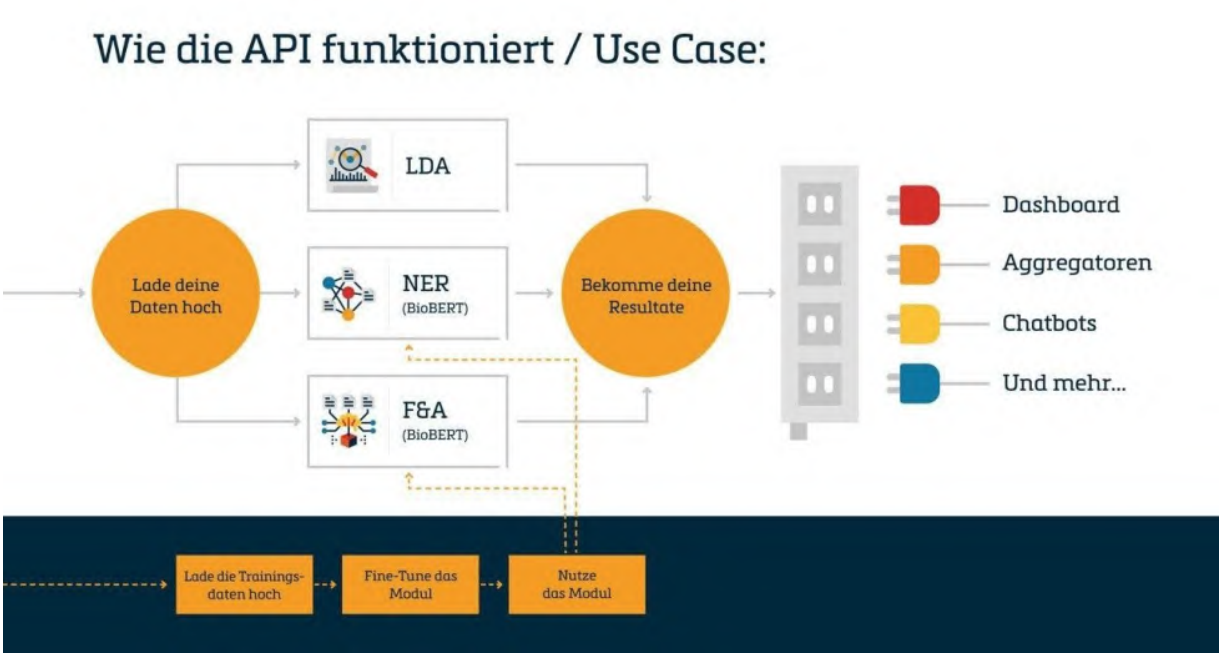


Abbildung 1: Workflow mit der OpenResearch API

[3] ZB MED – Informationszentrum Lebenswissenschaften.

Auf dem GitHub-Repository werden die folgenden API-Komponenten zur Verfügung gestellt:

Job-Service-Component

Der Job-Service-Component ist für die Verwaltung von Jobs zuständig. In Bezug auf die OpenResearch API ist ein Job ein in sich geschlossenes Datenobjekt, das für Folgendes verwendet wird:

- die Konfiguration, wie ein Modell trainiert oder verfeinert werden soll;
- die Einstellung von Parametern für die Analyse eines Datensatzes;
- die Speicherung der Ergebnisse einer Trainings- oder Analyseaufgabe;
- die Registrierung der verschiedenen Zustände eines Jobs, die sind: cancelled, waiting, training, analysing, completed.

Nachstehend ein Beispiel für eine API-Abfrage, bei der der Benutzer einen F&A-Analysis-Job registriert, um Antworten unter Verwendung des BioBERT-Sprachmodells auf einen bestimmten Datensatz zu erhalten:

```
curl --location --request POST 'https://api-gateway.openresearch.cloud/jobs' \
--header 'label: Prediction of answers with BioBert' \
--header 'kind: qna' \
--header 'task: analyse' \
--header 'description: Game of Thrones in ElasticSearch' \
--header 'model: 761dcb25-47e1-41e8-a5a8-651f0a916ba8' \
--header 'dataSample: ea13ebc0-18bf-4dfe-8750-61641fdbb00b' \
--header 'taskParams: { "questions": ["Can humans become infected with the COVID-19
from an animal source?", "Can pregnant women pass coronavirus to unborn children?",
"Can the COVID-19 virus spread through drinking water?"], "top_k_retriever":  5,
"top_k_reader": 5, "details": "minimal",  "selectTargetTextColumn": "abstract",
"selectTargetIdColumn": "cord_uid"}'
```

Data-Service-Component

Der Daten-Service-Component ist verantwortlich für die Registrierung von Modellen, Datensätzen (die von den Jobs verwendet werden) und Ergebnissen (die aus den Analysen generiert werden). Dieses Modul ist auch für die Verwaltung von Uploads großer Dateien in das Cloud-Storage-System verantwortlich.

Communication-Service-Component

Der Communication-Service-Component ist verantwortlich für den Empfang von Push-Notifikationen vom Job-Service-Component und für das Hinzufügen von Jobs zu einer Queue, sodass sie sich in Stand-by befinden, um ausgeführt zu werden, sobald Rechenressourcen zur Verfügung gestellt werden. Beim derzeitigen Aufbau der Architektur nutzt die API das Folgende:

- 1 Topic in AWS SNS, von dem der Communication-Service-Component subskribiert;

- 6 SQS FIFO-Queues, in denen Jobs zur Training und Analyse der 3 verfügbaren ML-Dienste gesammelt werden.

Task-Service-Component

Der Task-Service-Component ist für die Durchführung der ML-Aufgaben verantwortlich. Während der Laufzeit können die, von dem Task-Service-Component instanziierten, Container je nach Startup-Einrichtung für die Verarbeitung verschiedener und spezifischer ML-Aufgaben vorgesehen werden. Alle verfügbaren ML-Services (LDA, NER+BioBERT, Q&A+BioBERT) müssen zusätzlich zu den beiden unterschiedlichen Hauptaufgaben, die sie ausführen sollen (Training und Analyse), in einzelnen spezialisierten Instanzen enthalten sein. Daher erstellt die API mit den derzeit verfügbaren Services insgesamt sechs Task-Service-Komponenten-Container. Die Task-Service-Component-Instanzen sind dafür verantwortlich, Jobs aus den Queues zu ziehen und auszuführen, sobald sie in den Idle-Zustand gehen.

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?

In erster Linie haben wir Forschungseinrichtungen als unsere Hauptzielgruppe identifiziert. Um zu veranschaulichen, wie diese von unserer API profitieren, können wir die ZB-MED als Anwendungsfall heranziehen. Das ZB-MED entwickelt derzeit einen Aggregator mit dem Namen “preVIEW: COVID-19”. Die Institution experimentiert mit neueren Sprachmodellen, um die Ergebnisse zu verbessern und mehr Erkenntnisse über Publikationen zu gewinnen. Die OpenResearch API kann eine Lösung für Institutionen wie die ZB-MED bieten, indem die OpenResearch API als Cloud-Lösung eine einsatzbereite Infrastruktur zur Verfügung stellt, die einfach in Drittsysteme, z.B. Aggregatoren, integriert werden kann, um Dokumente zu analysieren. Darüber hinaus können mehrere Sprachmodelle mit unterschiedlichen Konfigurationen der Öffentlichkeit zur Verfügung gestellt werden, sodass andere Forschungseinrichtungen und Domänenexperten diese Modelle wiederverwenden können.

Da die API in Bezug auf ihre Architektur einen hochgradig modularen Ansatz verfolgt, können verschiedene spezialisierte KI-Module leicht an Dienste der API gekoppelt werden. Durch die vollständige Verfügbarkeit des Projekts auf GitHub stehen Forschungseinrichtungen und Entwicklern grundsätzlich alle notwendigen Mittel zur Verfügung, um die Software nach Wunsch zu erweitern.

Obwohl die Software wie beabsichtigt funktioniert, müssen nichtsdestotrotz weitere Tests, insbesondere in Hinblick auf die hohe Auslastung des Systems, die durch eine große Anzahl von Benutzern erzeugt wird, durchgeführt werden. Außerdem müssen beim ML-Service NER+BioBERT noch einige Verbesserungen vorgenommen werden. Wie unser Kooperationspartner uns bewusst machte, wären die Ergebnisse eines solchen Dienstes nur dann nützlich und in Betracht zu ziehen, wenn die durch den NER-Ansatz extrahierten Entitäten normalisiert werden könnten. Leider konnte eine Normalisierungsstrategie (Named-Entity-Normalization - NEN) während der Zeit der Finanzierung nicht umgesetzt werden, da die Wichtigkeit der Normalisierungsstrategie erst zum Ende dieser aufkam.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

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

Wie bereits erläutert, haben wir während der Entwicklung beschlossen, einen der ML-Dienste, nämlich Doc2Vec, durch Named-Entity-Recognition mit BioBERT zu ersetzen. Die Änderung wurde umgesetzt, um besser auf die Bedürfnisse unseres Kooperationspartners eingehen zu können.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Homepage: https://openresearch.cloud
Dokumentation: https://app.swaggerhub.com/apis-docs/OR-API/OpenResearch/1.0.1
GitHub: https://github.com/leonardomra/openresearch-api.git
API: https://api-gateway.openresearch.cloud

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?

Das Projekt stellte viele technische Herausforderungen aufgrund der Komplexität der Systemanforderungen und der Systemarchitektur, in der viele Module koordiniert arbeiten mussten. Die zusätzliche Zeit, die zur Lösung dieser technischen Herausforderungen erforderlich war, wurde durch die Wiederverwendung externer Open-Source-Projekte kompensiert, was die Entwicklung der Module NER und Q&A beschleunigte. Darüber hinaus wurde die Arbeits- und Kostenplanung auch in allen anderen APs eingehalten.

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?

Von Anfang an erhielt das Projekt viele spezifische Rückmeldungen aus verschiedenen Quellen: z.B. von Paten und Experten aus dem Solution Enabler Programm sowie von unserem Kooperationspartner. Diese Rückmeldungen waren wichtig, um entweder sicherzustellen, dass wir auf dem richtigen Weg waren, oder, um Verbesserungsmöglichkeiten aufzuzeigen. Die Idee, einige der Dienste auf der Grundlage von BioBERT anstelle des BERT zu implementieren und Doc2Vec durch NER zu ersetzen, sind Beispiele für solche Einflüsse Dritter in unserem Projekt.