Richtlinie zum „Software-Sprint“

DocDialog - Unterhalte dich mit deinen (vertraulichen) Dokumenten auf Deutsch

Schlussbericht

Zuwendungsempfänger:

Johannes Twiefel

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS24S21 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)?

Ist man als Nicht-Expert*in mit einem neuen Thema konfrontiert, z.B. Bauvorschriften, recherchiert man zunächst im Internet – und erhält innerhalb kürzester Zeit unzählige Dokumente. Nun aber all diese zu lesen, inhaltlich zu verstehen und in Beziehung zueinander zu setzen, erfordert einen hohen Arbeitsaufwand. Mit dem Projekt DocDialog sollte es durch Künstliche Intelligenz ermöglicht werden, beliebige Textdokumente (wie hier Normen und Gesetze) durchsuchbar zu machen und Fragen, insofern die Antwort in den Dokumenten liegt, automatisch beantwortet zu bekommen. Dazu sollten lokal verfügbare Technologien zum Einsatz kommen, um das System auch auf vertrauliche Dokumente anwenden zu können. Es gab folgende wichtige Meilensteine (zum Zeitpunkt der Beantragung):

1. Es sollte eine Webplattform entwickelt werden, die selbst betrieben werden kann. (Flask)
2. Diese sollte mit einer Datenbank verknüpft werden, sodass eigene Dokumente (wie Gesetze, Normen oder andere) hochgeladen und gespeichert werden können. (Postgres)
3. Die Normen mussten aus PDFs als Rohtext extrahiert werden. (PyPDF2) Die Webplattform sollte ein Interface enthalten, welches Fragen zu Dokumenten beantwortet.
4. Das KI-System sollte auf LeoLM basieren, dem ersten wirklich nutzbaren deutschem openSource Large Language Model (https://laion.ai/blog/leo-lm/ erschienen am 28.09.2023)
5. Mit LeoLM sollten Embeddings (Maschinenlesbare Vektor-Repräsentationen eines Textes) für jedes Dokument erzeugt werden.

6. Die Dokumente sollten zusammen mit den Embeddings in einer Open Source Vektordatenbank wie Milvus abgelegt.
7. Fragen an die Dokumente sollten zunächst in Embeddings umgewandelt und die passenden Dokumente zu der Frage aus der Vektordatenbank geholt werden.
8. Die relevanten Dokumente sollten zusammen mit der Frage als Prompt an LeoLM geschickt werden.
9. Die Antwort von LeoLM sollte in der Webplattform dargestellt werden.

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?

Die Zielgruppe besteht aus Bürgern, welche Fragen zu vorhandenen Dokumenten haben und Firmen/Behörden, welche durch das Befragen der eigenen Dokumente entlastet werden wollen. Die entwickelte Software soll DSGVO-konform auf eigener Hardware betrieben werden können, ohne eine Internetverbindung zu benötigen. Das Github-Projekt enthält eine Beschreibung, welche Hardware benötigt wird und wie die Software installiert und benutzt werden kann.

Es bestehen folgende Bezüge zu den Themenfeldern und Zielen des PrototypeFund (Software-Sprint):

1. Civic Tech: Es handel sich um ein digitales Werkzeug, welches den Zugang zu Informationen unter Bewahrung der Bürger*innenrechte.
2. Data Literacy: Das Werkzeug hilft dabei, eigene Daten schneller zu interpretieren und zu verstehen.
3. Data Security: Das Tool ermöglicht es, Fragen an eigene (möglicherweise vertrauliche) Daten zu stellen und zu beantworten unter der Einhaltung des Rechts auf informationelle Selbstbestimmung. Daten müssen nicht in die Cloud / das Internet übertragen werden und folgt dem Credo “Privacy by Design”.

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?

Es konnte ein lauffähiger Prototyp eines Question Answering Systems entwickelt werden, welches in akzeptabler Zeit (~10s) Antworten auf Fragen zu eigenen Dokumenten erzeugen kann. Außerdem und besonders hervorzuheben ist die Darstellung der Quellen aus den eigenen Dokumenten. Dabei werden Textpassagen dargestellt, welche die Informationen beinhalten, die als Quelle für die Erzeugung der Antwort genutzt wurden. Die Antwort kann, wie bei allen Systemen, welche auf Large Language Models basieren, Halluzinationen und möglicherweise falsche Antworten enthalten. Durch sehr restriktive Prompts wird dies weitestgehend eingeschränkt. Die Quellen helfen dabei, die Richtigkeit der Antwort zu überprüfen.

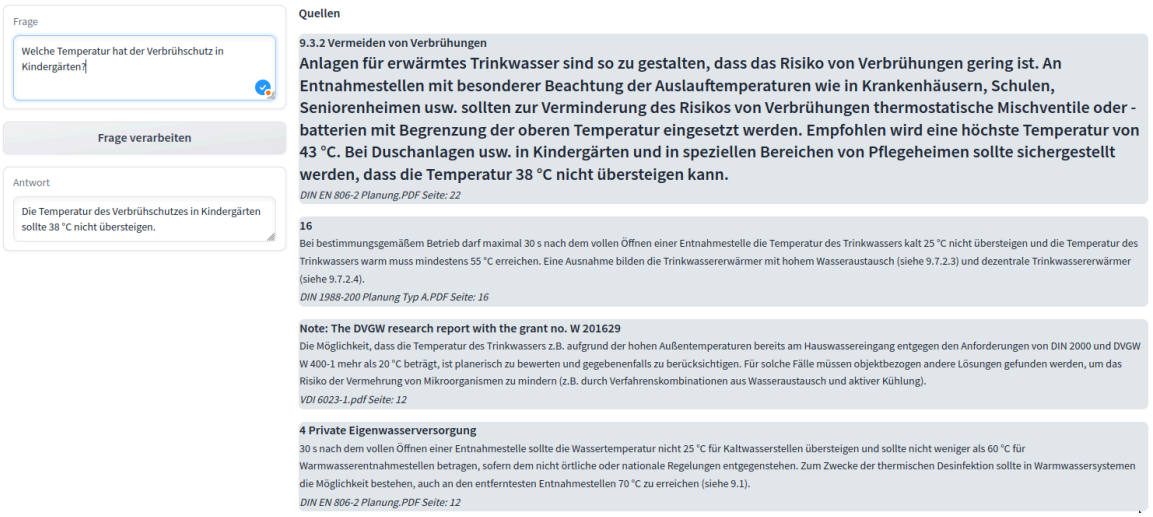
Somit ist der Nutzer in der Lage, Fragen zu eigenen Dokumenten auf Basis der Antworten des Systems und den angezeigten Quellen erheblich schneller als zu beantworten, als wenn die Dokumente

mühselig durchgearbeitet werden müssen. Das System benötigt eine NVIDIA-Grafikkarte mit mindestens 10GB VRAM und ist vollständig ohne Internetverbindung und Cloud-Dienste nutzbar.

Es konnten alle Meilensteine erreicht werden:

- Es wurde eine funktionsfähige Webplattform mithilfe von gradio entwickelt.
- Die hochgeladenen PDF-Dateien werden lokal gespeichert.
- Der Rohtext wird mit PyMuPDF und PyMuPDF4LLM als Markdown extrahiert und in Paragrafen unterteilt.
- Die Webplattform enthält ein Interface, auf dem Fragen eingegeben werden können, Antworten angezeigt werden können, sowie die Quellen angezeigt werden können, auf denen die Antwort basiert. Außerdem können eigene PDF Dateien hochgeladen und verwaltet werden.
- Das verwendete Large Language Model besteht aus einer INT4-quantisierten Version von Llama3.1.
- Mit dem Embeddings Model bge_m3 können Embeddings für jedes hochgeladene PDF-Dokument erzeugt werden. Dies geschieht paragrafweise.
- Die Paragrafen werden zusammen mit den Embeddings in einer Open Source Vektordatenbank (FAISS) abgelegt.
- Fragen an die Dokumente werden zunächst in Embeddings umgewandelt und die passenden Paragrafen zu der Frage aus der Vektordatenbank geholt.
- Die relevanten Dokumente werden zusammen mit der Frage als Prompt an Llama3.1 geschickt.
- Die Antwort von Llama3.1 sowie die Quellen aus bge_m3 werden in der Webplattform dargestellt.

Die Reihenfolge der Meilensteine wurde teilweise verändert und es wurden teilweise andere Technologien verwendet, welche zum Zeitpunkt der Implementierung neuer und geeigneter waren als die bei der Beantragung vorgeschlagenen Technologien. Die Open Knowledge Foundation hat sehr dabei geholfen, die oft sehr bürokratischen Vorgänge in Zusammenhang mit Fördermitteln zu vereinfachen. Im Großen und Ganzen war das Projekt sehr erfolgreich (lauffähiger Prototyp) und lief wie geplant (alle Meilensteine erreicht, teilweise andere Technologien verwendet als geplant). Das System ist über Browser erreichbar, die grafische Oberfläche sieht folgendermaßen aus:



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?

Nutzer können das System auf eigener Hardware betreiben, eigene Dokumente hochladen und diese befragen. Der Prototyp ist voll funktionsfähig. Während der Projektlaufzeit sind neue KI-Modelle (Large Language Models, Embedding Models) erschienen, welche in das System integriert wurden und so die Performance steigern konnten. Die Entwicklung im Bereich KI ist sehr rasant, es ist sehr nützlich, dass das Projekt als Open Source erscheint, sodass in Zukunft auch weiterhin neue Modelle eingebunden werden können. Das Projekt hat mir sehr geholfen, meine fachliche Expertise im Bereich Retrieval Augmented Generation weiterzuentwickeln.

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?

Anstelle des kurz nach Beantragung erschienenen Large Language Models LeoLM wurde LLama3.1 verwendet. Es wurde als Embedding Model zunächst mit LeoLM, später cross-en-de-roberta-sentence-transformer verwendet. In der finalen Version wurde bge_m3 als Embedding Model verwendet. Da nicht vorher ersichtlich ist, welche konkrete Modell das am besten geeignete ist, wurden verschiedene Modelle eingesetzt und über einen Benchmarktest die Performance gemessen. Es wurden verschiedene Model Loader getestet, etwa ctransformers, text-generation-webui und vLLM. Diese arbeiten sehr eng mit der Hardware (Grafikkarte) und sind schwer zu konfigurieren bzw. funktionieren oft nicht wie gewünscht, sind etwa aus unbekannten Gründen langsam. Letztendlich wurde vLLM lauffähig gemacht. Die PDF-Dateien wurden zunächst seitenweise eingelesen. Dies führte zu einer akzeptablen, wenn auch nicht optimalen Performance. Wenn auf einer Seite viele irrelevante und wenige relevante Informationen vorhanden waren, war die Qualität der Embeddings wesentlich schlechter. Daher wurde das Einlesen der PDF-Dateien als Rohtext erweitert, sodass Paragrafen mit Überschriften einzeln eingelesen wurden. Dies hat die Performance massiv gesteigert. Zunächst wurde versucht, nachdem die relevanten Seiten für die Frage aus der Vektordatenbank geladen wurden, für jede dieser Seiten durch ein LLM zu entscheiden, ob diese relevant für die Beantwortung der Frage ist oder nicht. Dieses Vorgehen wurde wieder verworfen, weil dies mindestens eine Sekunde Bearbeitungszeit je Seite erforderte, was bei 60 Seiten bereits eine Antwortzeit von mindestens 60 Sekunden bedeuten würde. Die Qualität der Embeddings mit bge_m3 ist so gut, dass für die Testdaten alle korrekten Quellen innerhalb der ersten 20 Ergebnisse des Retrieval-Systems erschienen. Dies reicht aus, um sämtliche erhaltenen Texte in den Prompt zu geben, um die Frage zu beantworten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Der Code und eine Installationsanleitung sind auf GitHub vorhanden:
https://github.com/jtwiefel/doc_dialog

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?

Es wurden insgesamt nur wenig Urlaub gemacht, daher sind mehr Stunden gearbeitet worden als vom Projektträger getragen werden können. Diese sind dann unbezahlt. Die Stunden wurde dafür verwendet, um die Performance des Systems zu steigern.

Es wurden teilweise andere Bibliotheken benutzt, als vorher geplant:

- Anstelle von Flask wurde gradio für die Webplattform verwendet, da so sehr einfach eine grafische Oberfläche entwickelt werden konnte
- Anstelle eine Postgres Datenbank wurden die Daten als pickle gespeichert, da nur ein Nutzer vorgesehen ist und dies den Umfang vereinfacht
- Anstelle von PyPDF2 wurde PyMuPDF und PyMuPDF4LLM verwendet, da der Text in Paragraphen geteilt wurde. Dafür wurde die PDF zunächst in Markdown umgewandelt, was nicht durch PyPDF2 unterstützt wurde
- Anstelle von LeoLM wurde für die finale Version Llama3.1 verwendet, da die Performance besser war
- Anstelle von Milvus wurde FAISS als Vektordatenbank verwendet
- Anstelle von LeoLM wurde für die Embeddings bge_m3 aufgrund besserer Performance für die finale Version verwendet

Von den Meilensteinen wurde nicht abgewichen, es wurden teilweise nur andere Technologien verwendet, um eine bessere Performance zu erhalten. Einige der Technologien wurden erst während der Projektlauffzeit veröffentlicht.

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?

nein