UNIVERSITÄT
LEIPZIG

Software engineering for AI systems

Retrieval enhanced generative QA chatbot system based on GPT-3.5 Turbo

June 22, 2023

Moritz Beyer, Max Matkowitz, Elias Messner, Felix Vogel

Institut für Informatik
Fakultät für Mathematik und Informatik
Universität Leipzig

Outline

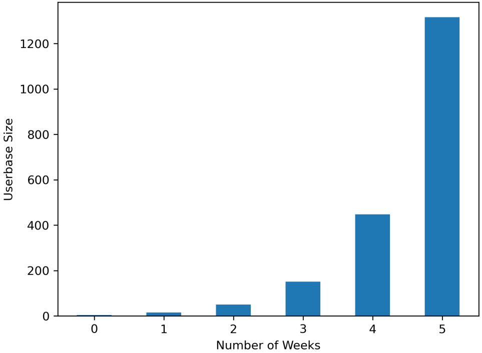
1. Businessplan
   1.1 Cost Estimation
   1.2 Risk Register
2. Pipeline Implementation
3. Reflection
   3.1 Challenges
   3.2 Work Method
   3.3 Further Steps
4. Discussion

Businessplan

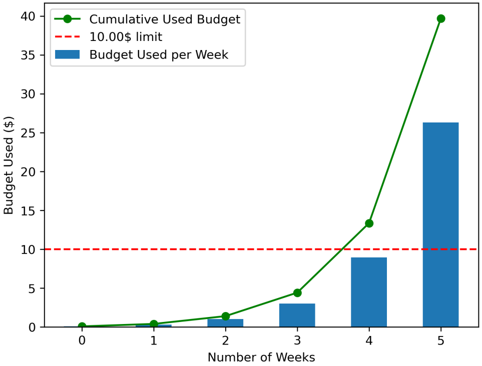
Cost Estimation

- Average request length ≈ 350 tokens
- Average weekly token usage per user ≈ 4100 tokens
- $\mathit{userbase}(n) = \mathit{userbase}(n-1) + r \cdot (\mathit{userbase}(n-1) - \mathit{userbase}(n-2))$

Cost Estimation



Cost Estimation



Risk Register I: Budget runs out

Development Team

Raised 09.05.2023 | Likelihood High | Impact High | Severity High

Risk Register II: Copyright infringement

Development Team

Raised 09.05.2023 | Likelihood Medium | Impact High | Severity High

Risk Register III: Database fills up

Development Team

Raised 16.06.2023 | Likelihood Low | Impact High | Severity Medium

Risk Register IV: Deadlines are not met

Development Team

Raised 16.06.2023 | Likelihood Low | Impact High | Severity Medium

Risk Register V: Pinecone removes index

Development Team

Raised 16.06.2023 | Likelihood Medium | Impact Low | Severity Low

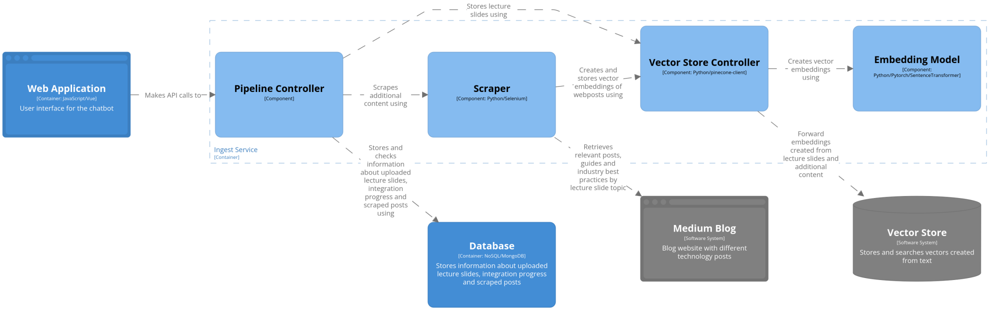
Risk Register VI: Production servers fail

Chair of Software Systems

Raised 16.06.2023 | Likelihood Low | Impact Low | Severity Low

Pipeline Implementation

Retrieval enhanced QA Chatbot | Pipeline Implementation



Web Application
[Container: JavaScript/Vue]
User interface for the chatbot

Makes API calls to

Ingest Service
[Container]

Pipeline Controller
[Component]

Scrapes additional content using

Scraper
[Component: Python/Selenium]

Stores lecture slides using

Creates and stores vector embeddings of webposts using

Vector Store Controller
[Component: Python/pinecone-client]

Creates vector embeddings using

Embedding Model
[Component: Python/Pytorch/SentenceTransformer]

Stores and checks information about uploaded lecture slides, integration progress and scraped posts using

Retrieves relevant posts, guides and industry best practices by lecture slide topic

Forward embeddings created from lecture slides and additional content

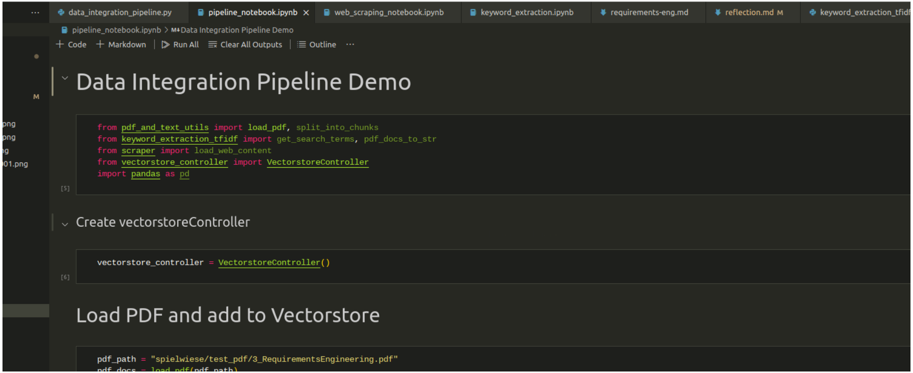
Database
[Container: NoSQL/MongoDB]
Stores information about uploaded lecture slides, integration progress and scraped posts

Medium Blog
[Software System]
Blog website with different technology posts

Vector Store
[Software System]
Stores and searches vectors created from text

[Component] Chatbot - Ingest Service
Tuesday, June 20, 2023 at 9:26 AM Central European Summer Time

Practical Demonstration



Data Integration Pipeline Demo

```
from pdf_and_text_utils import load_pdf, split_into_chunks
from keyword_extraction_tfidf import get_search_terms, pdf_docs_to_str
from scraper import load_web_content
from vectorstore_controller import VectorstoreController
import pandas as pd
```

Create vectorstoreController

```
vectorstore_controller = VectorstoreController()
```

Load PDF and add to Vectorstore

```
pdf_path = "spielwiese/test_pdf/3_RequirementsEngineering.pdf"
pdf_docs = load_pdf(pdf_path)
```

Reflection

Challenge I: Embeddings

Usage of Open AI Embeddings not possible through API Wrapper

- Switch to local embedding model based on sBERT

Challenge II: Load and structure of system

Load and structure of system is not manageable and scalable, when all components are running in a monolithic manner

- Separation into Core service (Handling chat questions and responses, searching of vector database) and Ingest service (Handling of file uploads and scraping, calculation of embeddings and ingestion into vector database)

Challenge III: Synchronization of services regarding upload status and loading duplicate slides/posts

Both services need to influence or access uploads. It must be ensured that duplicate slides and posts are not embedded and ingested into the vector database

- Usage of a database to track uploaded files and scraped posts (hashes) and status of uploads

Challenge IV: Topic of uploaded slides

It is needed to retrieve the topic of uploaded slides to scrape blog posts that are fitting to the lecture slides topic

- KeyBERT - Delivered poor results, because words that appear only once inside the text get assigned high relevance for the predicted keywords/topic
- Tfidf - Statistical approach to reduce text to keywords/topic - delivered good results, even though keywords tend to be very general
- Decision was taken to force users to enter a topic for slide uploads to increase scraping quality

Challenge V: Length of embedded text

The documents (slides, posts) can be splitted with different lengths (e.g. sentence, paragraphs, fixed size). Splitting posts by paragraphs delivered poor results, as there were many short paragraphs with no meaning.

- For posts, it was decided to split by fixed length (1000 characters) with NLTK (takes care of natural splitting, by adjusting the fixed length slightly)

Work Method

- 2 week sprints
- Combined sprint review and planning in between
- In plannings, it was already slightly discussed on who wants to take over which tasks and a soft assignment was made
- All in all: slight changes in agile development to fit project to student life and take into account that no one is working full time on the project
- Whatsapp Group for quick communication

Further Steps

- Focus on structuring and containerizing the code into components
- Definition of service interfaces
- Implementation of APIs, Frontend and interoperability
- Rollout of application components on Kubernetes cluster
- Implementation of A/B Testing
- Testing of metrics (e.g. response time)

Discussion

Risk Register I: Budget runs out

Development Team

Raised 09.05.2023 | Likelihood High | Impact High | Severity High

- Run calculations on the number of tokens sent to the API before sending the request
- Run tests with the API only when all the other infrastructure is working as intended so that the length of the request is minimal
- Regularly check the remaining budget

Risk Register II: Copyright infringement

Development Team

Raised 09.05.2023 | Likelihood Medium | Impact High | Severity High

- Restrain user access to files uploaded by other users
- Regularly check stored files for copyright infringement
- Talk to study senate to get blanko permission to store all lectures
- Only store lecture slides provided by professors and not allow upload by users

Risk Register III: Database fills up

Development Team

Raised 16.06.2023 | Likelihood Low | Impact High | Severity Medium

- Separately store already scraped links to avoid duplicates
- Delete oldest scraped content when database is close to full

Risk Register IV: Deadlines are not met

Development Team

Raised 16.06.2023 | Likelihood Low | Impact High | Severity Medium

- Create and check milestones
- Regularly have meetings as Development Team

Risk Register V: Pinecone removes index

Development Team

Raised 16.06.2023 | Likelihood Medium | Impact Low | Severity Low

- Regularly use the index to avoid deletion
- Keep code modular to allow implementing a different vector datebase later
- Possibly switch to a paid index if more budget becomes available

Risk Register VI: Production servers fail

Chair of Software Systems

Raised 16.06.2023 | Likelihood Low | Impact Low | Severity Low

- Communicate with Development Team in case of updates or failures