

Dokumentation & Projekttagbuch

Innovation Lab 2
Jahr 2025

Projekt: AI Info –
FH Technikum AI Info Chatbot

Team: **04**

1. Allgemeine Informationen

Projektname: AI Info – FH Technikum AI Info Chatbot

Supervisor: Knapp Bernhard

Innovation Lab 2, Sommersemester 2025

Projektteam

Brandtner Niklas, if23b270@technikum-wien.at, Projektleitung,

Alexander Kevin, if23b277@technikum-wien.at,

Lechner Sonja, if23b228@technikum-wien.at,

Vural Zeliha, if22b126@technikum-wien.at

Management-Summary des Projektes

Das Projekt „AI Info – FH Technikum AI Info Chatbot“ verfolgt das Ziel, einen intelligenten Chatbot zu entwickeln, der Studierenden und Mitarbeitenden der FH Technikum Wien hilft, Fragen zu offiziellen organisatorischen Dokumenten wie Prüfungsordnungen, Satzungen und Hausordnungen in natürlicher Sprache zu beantworten. Der Chatbot wird in der Lage sein, präzise Antworten zu liefern, dabei jedoch auch die genaue Quelle der Information (z. B. Seitenzahl und Abschnitt) anzugeben. Dies stellt sicher, dass Nutzer*innen die Antworten nachvollziehen und verifizieren können. Im ersten Semester lag der Fokus auf der organisatorischen Planung und Dokumentation, um eine solide Grundlage für die Umsetzung in den kommenden Semestern zu schaffen. Im zweiten Semester soll der Prototyp verbessert werden, eine Chatfunktion hinzugefügt werden und Word-Embedding verwenden.

Rahmenbedingungen und Projektumfeld

Für das Projekt gelten besondere Anforderungen hinsichtlich der Verlässlichkeit und Genauigkeit der Informationen. Der Chatbot muss nicht nur in der Lage sein, auf eine Vielzahl von offiziellen Dokumenten zuzugreifen, sondern auch Antworten zu liefern, die den aktuellen Regelungen der FH Technikum Wien entsprechen. Die Benutzerfreundlichkeit (Usability) ist essenziell, da der Chatbot von Studierenden und Personal genutzt wird. Das Projekt erfordert zudem die Einhaltung von Datenschutzstandards, insbesondere beim Umgang mit personenbezogenen Daten.

Semester-Roadmap

1. Semester

Organisation und Planung des Projekts, Erstellung des Management-Summarys, Definition der Anforderungen und Qualitätsstandards, Auswahl der technischen Tools, Erstellung eines Prototyps (Hardcode) für erste Usability-Tests und eine zweite, überarbeitete Version des Prototypen, welcher Text bereinigt und Entitäten/Sätze erkennt.

2. Semester

Im Rahmen des Projekts wurden zahlreiche User Stories definiert, die den Funktionsumfang des Chatbots strukturiert abbilden. Zu Beginn steht die Grundimplementierung im Fokus, bei

der eine Schnittstelle geschaffen wird, über die Nutzerinnen Fragen eingeben können. Der Chatbot soll dabei einfache Fragen aus einer vordefinierten Liste beantworten und in einer separaten Testumgebung getestet werden können. Sobald die Basisfunktionen stabil laufen, wird eine intelligente Suche mittels Word Embeddings integriert, um relevante Informationen präziser aus den Dokumenten zu extrahieren. In einem weiteren Schritt wird die Suchfunktion optimiert, sodass relevante Textstellen schneller gefunden werden und die Antworten angenehm formatiert sowie mit verständlichen Icons dargestellt werden. Um maximale Transparenz zu gewährleisten, werden alle Antworten mit einer exakten Quellenangabe inklusive Seitenzahl versehen. Abschließend wird ein Fallback-Mechanismus implementiert, der bei unklaren oder nicht eindeutigen Fragen eine neutrale Antwort liefert.

Collaboration & Tooling

Zur Zusammenarbeit werden GitHub ([GithubLink](#)) für das Versionierungssystem und ein GitHub Project für das Projektmanagement genutzt. Für die Dokumentation und das Protokollieren der Meetings wird Typora verwendet. Links zu den Tools:

- GitHub: https://github.com/asy0/AI_InfoChatbot.git
- Typora: Protokoll wird auf GitHub hochgeladen
- Python: Als primäre Programmiersprache für die Entwicklung

Besprechungen und Kommunikation

Wie bereits im letzten Semester nutzen wir die Videokommunikationsplattform Zoom für unsere Meetings mit unserem Betreuer. Für die interne Kommunikation nutzen wir die Mehrzweckplattform Discord.

- <https://zoom.us>
- <https://discord.com>

Anmerkungen

Das Projektteam besteht aus 4 Mitgliedern, und der Projektplan wird regelmäßig auf Basis der Sprint-Reviews angepasst. Der Fokus im ersten Semester liegt vor allem auf organisatorischen Aufgaben, während die technische Umsetzung ab dem zweiten Semester erfolgt.

2. Projekt-Kurzbeschreibung

1. Projektziel und Lösung

Das Projekt „AI Info – FH Technikum AI Info Chatbot“ hat das Ziel, einen intelligenten Chatbot zu entwickeln, der Studierenden und Mitarbeitenden der FH Technikum Wien bei Fragen zu organisatorischen Dokumenten wie Prüfungsordnungen, Satzungen und der Hausordnung effizient und präzise weiterhelfen kann. Das Besondere an diesem Chatbot ist seine Fähigkeit, nicht nur Antworten zu liefern, sondern auch die genaue Quelle der Informationen anzugeben. Dies erhöht die Transparenz und Nachvollziehbarkeit der Antworten.

Die Lösung wird folgende Funktionen bieten:

- Natürliche Sprachverarbeitung (NLP): Der Chatbot kann Anfragen in natürlicher Sprache verstehen und verarbeiten.
- Zitierfunktion: Jede Antwort wird mit einer genauen Quellenangabe versehen, sodass die Benutzer*innen die Informationen selbst verifizieren können.
- Zugriff auf organisatorische Dokumente: Der Chatbot wird auf offizielle Dokumente der FH Technikum Wien trainiert, um relevante Informationen abzurufen und bereitzustellen.

2. Größte Herausforderungen

Die größten Herausforderungen in diesem Projekt bestehen in der korrekten Verarbeitung und Analyse von Textdaten aus den vielfältigen organisatorischen Dokumenten der FH. Das Training des Chatbots auf juristischen Texten erfordert eine sorgfältige Handhabung, um sicherzustellen, dass der Bot relevante und präzise Informationen liefert. Zudem muss die Usability des Systems gewährleistet werden, da der Chatbot von einem breiten Benutzerkreis, einschließlich Studierenden und Mitarbeitenden, genutzt wird. Ein weiterer komplexer Aspekt ist die Sicherstellung der Datenschutzbestimmungen beim Umgang mit Anfragen und potenziell personenbezogenen Daten. Auch der Prozess der kontinuierlichen Verbesserung und Anpassung des Chatbots an die Bedürfnisse der Nutzer*innen wird ein wichtiger Punkt sein.

3. Mehrwert für die Anwender*innen

Der größte Mehrwert dieses Chatbots liegt in der Zeitersparnis und Vereinfachung von Suchprozessen. Statt durch umfangreiche Dokumente zu navigieren, können Benutzer*innen ihre Fragen direkt stellen und erhalten unmittelbar eine passende, rechtssichere Antwort mit Quellenangabe. Zudem wird die Barrierefreiheit erhöht, da die Informationen in einer leicht verständlichen Form zugänglich gemacht werden.

4. Scope des Projekts

- Ziel des Projekts

Die Entwicklung und Implementierung eines AI-basierten Chatbots, der auf organisatorische Dokumente der FH Technikum Wien zugreift und Fragen dazu beantwortet. Der Chatbot

wird über eine Schnittstelle zugänglich gemacht und auf spezifische Anfragen reagieren.

- Nicht-Ziele

Der Chatbot wird keine generelle Beratung zu Themen außerhalb der definierten Dokumente liefern, z.B. individuelle Studienberatung oder allgemeine technische Hilfestellungen. Außerdem wird der Chatbot keine interaktive Nutzung über externe Plattformen (z.B. soziale Netzwerke) ermöglichen. Die Integration in hochspezialisierte Systeme wie das FH-interne Prüfungsverwaltungssystem ist nicht Teil des Projekts.

5. Umsetzung und Vorgehensweise

Im ersten Semester wird das Projektteam die organisatorische Planung und Dokumentation des Projekts durchführen. Dazu gehören die detaillierte Definition der Projektziele, das Sammeln relevanter Dokumente und die Ausarbeitung einer Datenstruktur für den Chatbot. Eine Evaluierung von bestehenden NLP-Modellen (z.B. BERT, GPT) wird durchgeführt, um das geeignetste Modell für die Sprachverarbeitung auszuwählen.

Im zweiten Semester folgt die Implementierung der Chatbot-Logik, basierend auf den dokumentierten Anforderungen. Der Fokus liegt hierbei auf der Integration der organisatorischen Dokumente und der Entwicklung eines funktionalen Prototyps. Erste Tests und Optimierungen werden durchgeführt, um sicherzustellen, dass der Chatbot die Anforderungen erfüllt.

Das dritte Semester widmet sich der Optimierung des Chatbots, einschließlich Benutzerfreundlichkeit und Performanz. Es werden abschließende Tests durchgeführt und Feedback von Benutzer*innen eingeholt, um das System vor der endgültigen Bereitstellung zu verfeinern.

6. Ergebnisse (Deliverables) pro Semester

1. Semester

Ausführliche Planung, Dokumentation, Datenstruktur, Sprintplanung, Erstellung eines Hardcoded-Prototyps zur Validierung der Usability und eine zweite, überarbeitete Version des Prototypen, welcher Text bereinigt und Entitäten/Sätze erkennt.

2. Semester

Funktionsfähiger Prototyp des Chatbots, inklusive Integration der Dokumente und Zitierfunktion, Training eines Modells auf Basis der gegebenen Files.

3. Semester

Finaler Chatbot, inklusive detaillierter Verfeinerungen der NLP-Modelle, umfangreicher Benutzerfreundlichkeitstests und Optimierung der Antwortqualität.

3. Spezifikation der Lösung

Systemumfeld

- ❖ Der AI Info Chatbot dient Studierenden und Mitarbeitenden der FH Technikum Wien als zentrale Anwendung zur schnellen Beantwortung von Fragen zu organisatorischen Dokumenten.
- ❖ Die Verarbeitung erfolgt lokal und greift auf offizielle PDF-Dokumente der FH zu.

Features (Funktionale Anforderungen)

1. PDF-Verarbeitung

- Möglichkeit, PDF-Dokumente zu lesen und deren Text zu extrahieren.
- Bereinigung des extrahierten Texts, indem Seitenzahlen und Sonderzeichen entfernt werden.

2. Textanalyse

- Extraktion und Kennzeichnung wichtiger Begriffe (z. B. „Prüfungsordnung“, „Hausordnung“) als Entitäten.
- Segmentierung des Texts in einzelne Sätze für gezielte Verarbeitung.

3. Verarbeitung mehrerer Dokumente

- Alle PDFs in einem Ordner werden durchsucht, Texte extrahiert und analysiert.

4. Frage-Antwort-System

- Nutzer können Fragen stellen, und der Bot liefert passende Antworten aus den Dokumenten.
- Quellenangabe mit Seitenzahl und Abschnitt für maximale Transparenz.

5. Benutzeroberfläche

- Eine einfache Oberfläche, auf der Nutzer PDFs hochladen und Fragen stellen können.

6. Erweiterte Suche

- Vergleich von Nutzerfragen mit Textinhalten zur Bestimmung der relevantesten Antwort.

Schnittstellen

- ❖ Benutzerschnittstelle: Weboberfläche zur Interaktion mit dem Chatbot.
- ❖ Datenverarbeitung: Funktionen zur PDF-Analyse und Entitätenerkennung.
- ❖ Backend: Python-basierte Verarbeitung mit spaCy für NLP.

Qualitätseigenschaften, technische Anforderungen (Nicht-Funktionale Anforderungen)

- ❖ Seitenangaben speichern: Seitenzahlen werden beim Extrahieren beibehalten.
- ❖ Textkategorisierung: Strukturierung in Themenbereiche wie „Prüfungsordnung“ und „Hausordnung“.
- ❖ Performance: Effiziente Verarbeitung auch bei mehreren Dokumenten.
- ❖ Skalierbarkeit: System muss mit steigender Anzahl von Dokumenten und Fragen umgehen können.
- ❖ Benutzbarkeit: Intuitives Design für Studierende und Mitarbeitende.

Sonstige Merkmale

- ❖ Tests und Optimierung: Regelmäßige Tests mit verschiedenen Dokumenten zur Verbesserung der Antwortgenauigkeit.
- ❖ Schrittweise Entwicklung: Prototyping (Hardcoded) im ersten Semester, Modelltraining im zweiten Semester und Optimierung im dritten Semester.

4. Aufwandschätzung

Für die Aufwandschätzung der im Projekt definierten Epics und User Stories wurde die Delphi-Methode in Kombination mit PERT (Program Evaluation and Review Technique) angewendet. Ziel war es, eine möglichst realistische Aufwandseinschätzung für die Umsetzung unseres Chatbot-Projekts im Rahmen von InnoLab 2 zu erhalten.

Zusammenfassung der Ergebnisse (nach Epics)

Epic 1 – Grundimplementierung des Chatbots

Dieser Bereich stellt die Basis des Chatbots dar. Die fünf zugehörigen User Stories erfordern einen geschätzten Gesamtaufwand von 50,5 Story Points, was hauptsächlich durch die Implementierung der Kernfunktionen wie Schnittstelle, Basisantworten und Testumgebung begründet ist.

T-Shirt-Größe: L

Epic 2 – Bessere Verarbeitung von Dokumenten & präzisere Quellenangaben

Die User Stories hier fokussieren sich auf Suchfunktion, Antwortdarstellung und Sprachumschaltung. Der Gesamtaufwand beläuft sich auf 20,5 Story Points, wobei insbesondere die Suche nach relevanten Stellen in Dokumenten komplexer ist.

T-Shirt-Größe: M

Epic 3 – Interaktive Funktionen für bessere Nutzererfahrung

Diese Story umfasst ein einfaches Feedbacksystem für falsche Antworten und hat einen geringeren geschätzten Aufwand von 2,83 Story Points.

T-Shirt-Größe: M

Epic 4 – Verarbeitung der Suche für kürzere Ladezeiten

Diese User Stories betreffen Optimierungen wie das Erkennen von Synonymen, Fallback-Mechanismen und das Caching von häufigen Fragen. Der Aufwand liegt hier bei insgesamt 18,7 Story Points, wobei einzelne Stories wie das Word Embedding-Caching einen höheren Aufwand darstellen.

T-Shirt-Größe: M

Gesamtaufwand (alle Epics/User Stories)

Kategorie	Erwarteter Aufwand (Sp)
Epic 1	50,5
Epic 2	20,5
Epic 3	2,83
Epic 4	18,7
Gesamt	~92,5 Story Points

Link zur Aufwandschätzung (Excel)

<https://cloud.technikum-wien.at/s/XccJpEnkH2KPXLJ>

5. Auslieferung

Lieferumfang der Lösung

Im Rahmen dieses Projekts haben alle Teammitglieder (4 Personen) individuell an eigenen Implementierungen gearbeitet. Ziel war es, verschiedene Ansätze zur Lösung derselben Aufgabenstellung zu entwickeln. Dadurch sind vier unterschiedliche Versionen entstanden, die sich teils in ihrer Systemarchitektur, Funktionsweise und technologischen Umsetzung unterscheiden.

Da die Projekte im kommenden Semester weiterentwickelt und ggf. zusammengeführt werden sollen, haben wir entschieden, nur die zwei stabilsten und ausbaufähigsten Versionen im GitHub-Repository zu dokumentieren. Diese enthalten jeweils eine vollständige Zusammenfassung und technische Dokumentation, um eine spätere Weiterverwendung oder Integration zu ermöglichen.

Enthaltene Bestandteile

- **Lösungskomponenten**

Zwei vollständige Python-Projekte, jeweils mit funktionierendem Source-Code und den folgenden wichtigen Bestandteilen:

- **Version 1:** Eine Implementierung als Python-Skript, das in der Kommandozeile ausgeführt wird.
- **Version 2:** Eine erweiterte Version, die mit Streamlit eine Web-Oberfläche für die Interaktion mit der Anwendung bietet.

- **Systemarchitektur & Datenhaltung**

Beide Projekte basieren auf einer Client-Server-Architektur, jedoch mit unterschiedlichen Client-Komponenten:

- Web-basierter Client: Eine der Versionen nutzt eine Web-Oberfläche, die mit Streamlit entwickelt wurde. Hier können die Nutzer*innen Fragen an den Chatbot stellen und Antworten auf der Webseite erhalten.
- Terminal-basierter Client: Die andere Version des Projekts nutzt ein einfaches Terminal-Interface, über das Nutzer*innen ebenfalls Anfragen an den Chatbot stellen können, jedoch über die Kommandozeile.

In beiden Fällen fungiert der Server als zentraler Verarbeitungspunkt, der Anfragen entgegennimmt, Text aus den Dokumenten extrahiert und analysiert und dann die passenden Antworten zurückgibt. Für die Datenhaltung haben wir einen zentralen Ordner in der FH Technikum Cloud erstellt, in dem alle relevanten Dokumente und Projektdateien abgelegt werden. Dies ermöglicht den sicheren und strukturierten Zugriff auf die Daten durch alle Teammitglieder.

- **Lizenzen & Copyrights**

- Alle verwendeten externen Bibliotheken und Frameworks sind Open-Source und unter folgenden Lizenzen verfügbar:

- **Streamlit** (MIT-Lizenz),
- **Flask** (BSD-Lizenz),
- **Pandas** (BSD-Lizenz),
- **NumPy** (BSD-Lizenz).
- Die vollständige Liste der verwendeten Bibliotheken ist in der requirements.txt enthalten.
- **Hardwareanforderungen**
 - Es sind keine speziellen Hardwareanforderungen notwendig. Die Anwendung funktioniert auf handelsüblichen Laptops mit mindestens 8 GB RAM und einer aktuellen Python-Version (ab 3.8).
 - Für die Streamlit-Oberfläche ist ein moderner Webbrowser erforderlich (z.B. Chrome oder Firefox).

Installationsanleitung

Die Installation erfolgt über die folgenden Schritte:

1. Klonen des Repositories oder Herunterladen der ZIP-Datei.
2. Erstellen eines virtuellen Python-Umfeldes (z.B. mit venv oder conda).
3. Installieren der notwendigen Abhängigkeiten:

```
pip install -r requirements.txt
```

4. Bei Verwendung der Streamlit-Version kann die Anwendung mit folgendem Befehl gestartet werden:

```
streamlit run app.py
```

5. Wenn keine Web-Oberfläche benötigt wird, kann das Python-Skript direkt ausgeführt werden:

```
python main.py
```

Weiteres Vorgehen

Im kommenden Semester werden wir beide Versionen weiterentwickeln. Unser Ziel ist es, eine vollständige Integration der beiden Ansätze zu erreichen. Wir werden den Fokus auf die Verbesserung der Performance legen und eine benutzerfreundliche Oberfläche schaffen. Die agile Entwicklung folgt einem iterativen Ansatz (Forschung > Entwicklung > Feedback), wobei wir regelmäßige Tests und Anpassungen vornehmen werden, um ein robustes Endprodukt zu liefern.

6. Unser Projekt-Tagebuch

1. Sprint & Meeting

In der ersten Woche haben wir uns darauf konzentriert, wie wir Fragen und Antworten semantisch vergleichen können. Zunächst haben wir uns mit der SentenceTransformer-Bibliothek beschäftigt und herausgefunden, wie wir Sätze in Vektoren umwandeln können. Wir haben die Modelle SBERT und T5 getestet und verglichen, um herauszufinden, welches Modell am besten funktioniert.

Was wir gemacht haben

- Jeder von uns hat Recherchen angestellt, um die Modelle besser zu verstehen.
- GitHub wurde für die Code-Verwaltung genutzt, während Discord für regelmäßige Meetings und den Austausch untereinander genutzt wurde.

Herausforderungen

- Zu Beginn war es schwierig, das richtige Modell zu wählen und die Funktionsweise von Cosine Similarity zu verstehen.
- Das Erstellen einer einfachen Frage-Antwort-Mapping-Struktur stellte uns vor einige Herausforderungen.

Erfolge

- Wir haben einen funktionierenden ersten Prototyp entwickelt, der die semantische Ähnlichkeit zwischen Fragen und Antworten bewertet.
- Die Grundstruktur des Projekts wurde festgelegt und der erste Schritt erfolgreich abgeschlossen.

2. Sprint & Meeting

Im zweiten Sprint haben wir die verschiedenen Methoden vertieft und uns auf die Verbesserung des Antwort-Matchings konzentriert. Jeder von uns hat sich mit einer eigenen Methode beschäftigt, wie z.B. Bi- oder Cross-Encodern oder der Integration von FAISS für die performante Ähnlichkeitssuche. Wir haben außerdem verschiedene Sprachmodelle getestet, um herauszufinden, welches Modell am besten zu unserem Projekt passt.

Was wir gemacht haben

- Jeder hat sich intensiv mit einer eigenen Methode auseinandergesetzt und diese in den Prototyp integriert. Einige haben FAISS verwendet, andere haben Bi- und Cross-Encoder getestet.
- Sprachmodelle wie deepset/gbert-base und google/mt5-base wurden getestet, um die beste Leistung zu ermitteln.
- Jeder hat 5 Fragen erstellt, die unterschiedliche Schwierigkeitsgrade abdeckten, von einfachen bis hin zu komplexeren, mehrdeutigen Fragen.

Herausforderungen

- Die Implementierung von FAISS war anfangs herausfordernd, da einige Anpassungen nötig waren, um es korrekt zu integrieren.
- Da jeder eine andere Methode verwendete, war es schwierig, die verschiedenen Ansätze zu koordinieren und die besten auszuwählen.

Erfolge

- Wir konnten die Leistung des Antwort-Matchings durch die Anwendung verschiedener Methoden erheblich steigern.
- Die Integration von FAISS und Cross-Encodern hat die Geschwindigkeit und Genauigkeit des Matching-Prozesses deutlich verbessert.

Zusätzliche Details

Am Ende dieses Sprints haben wir die Ergebnisse zusammengefasst und per E-Mail an unseren Prof geschickt, um sicherzustellen, dass wir auf dem richtigen Weg sind.

3. Sprint& Meeting

Zu Beginn des Projekts haben wir mit einer einzelnen PDF-Datei gearbeitet. Im dritten Sprint haben wir jedoch begonnen, mehrere PDFs zu integrieren und deren Daten zu extrahieren, um die Leistung des Prototyps weiter zu testen und zu verbessern. Diese Erweiterung ermöglichte es uns, die Verarbeitung von verschiedenen Dokumenten und deren semantische Analyse weiter zu optimieren

Was wir gemacht haben

- Wir haben mehrere PDFs verarbeitet und relevante Daten extrahiert.
- Unsere Testumgebung wurde weiter optimiert, um sicherzustellen, dass die extrahierten Daten korrekt verarbeitet und validiert werden.

Herausforderungen

- Die PDFs hatten unterschiedliche Strukturen, was die Extraktion des Textes und die weitere Verarbeitung erschwerte.
- Die Daten mussten vorverarbeitet werden, um sie korrekt zu analysieren und zu integrieren.

Erfolge

- Die erfolgreiche Extraktion und semantische Einbettung von Daten aus mehreren PDFs.
- Wir konnten eine stabile Testumgebung einrichten, die uns half, die Qualität der extrahierten Daten zu validieren.

4. Sprint & Meeting

In diesem Sprint lag der Fokus auf der Durchführung systematischer Tests. Jeder von uns erstellte 10 Testfälle, die verschiedene Schwierigkeitsgrade abdeckten. Wir ließen alle Testfälle durch die Modelle laufen und analysierten die Ergebnisse, um festzustellen, welches Modell am besten funktionierte.

Was wir gemacht haben

- Wir haben alle Testfälle durchlaufen lassen und die Ergebnisse miteinander verglichen.
- Jeder hat eine eigene Methode zur Fehlererkennung entwickelt, die anschließend verglichen und auf ihre Wirksamkeit getestet wurde.
-

Herausforderungen

- Der manuelle Aufwand war hoch, da viele Testfälle bearbeitet werden mussten, was Zeit in Anspruch nahm.
- Einige Modelle erzielten bei bestimmten Fragetypen bessere Ergebnisse als bei anderen.

Erfolge

- Wir konnten die Leistung der Modelle unter verschiedenen Testfällen gut vergleichen und wichtige Schwächen erkennen.

5. Sprint & Meeting

In Sprint 5 haben wir den Feinschliff vorgenommen und uns auf das finale Modell geeinigt. Ein wichtiger Teil dieses Sprints war die Implementierung einer automatisierten Testführung, mit der wir alle Testfälle ohne manuelle Eingriffe durchlaufen lassen konnten. So konnten wir einfach nachverfolgen, wie viele Fragen korrekt beantwortet wurden und welche Schwächen noch bestehen.

Was wir gemacht haben

- Jeder hat 10 neue Fragen erstellt, sodass insgesamt 40 Fragen für die Tests zur Verfügung standen.
- Wir haben die automatisierte Testführung implementiert, um die Leistung der Modelle effizient zu überprüfen und die Fehlerquellen schnell zu identifizieren.

Herausforderungen

- Die Entwicklung der automatisierten Testführung war technisch anspruchsvoll, da es eine präzise Nachverfolgung ohne manuelle Eingriffe erforderte.
- Es war auch schwierig, die „Vorher-Nachher“-Ergebnisse eindeutig zu vergleichen und die Fortschritte genau zu dokumentieren.

Erfolge

- Die automatisierte Testführung funktionierte gut, sodass wir jetzt jederzeit nachvollziehen konnten, wie viele Antworten richtig waren und welche Schwächen noch bestanden.
- Wir haben eine klare Vorstellung davon, welches Modell in welchen Szenarien am besten funktioniert und konnten die Ergebnisse dokumentieren.

6. Sprint & Meeting

Im letzten Sprint lag der Fokus auf der Analyse der bisherigen Ergebnisse, der gezielten Behebung von Schwächen aus den Prototyp-Tests und der Vorbereitung für die Präsentation. Wir haben die besten zwei von vier Prototypen ausgewählt, die wir weiterverfolgen wollten. Es wurde besprochen, wie das Projekt weitergeführt wird, welche Teile des Codes hochgeladen werden müssen und welche nächsten Schritte anstehen.

Was wir gemacht haben

- Wir haben die Ergebnisse unserer Tests analysiert und die besten Methoden für die Präsentation zusammengefasst.
- Wir haben die vier Prototypen miteinander verglichen und die besten zwei ausgewählt.
- Es wurde besprochen, wie das Projekt weitergeführt wird, welche Teile des Codes hochgeladen werden müssen und wie die Dokumentation strukturiert wird.
- Ein Video wurde erstellt, das die Ergebnisse und die Funktionsweise unseres Projekts zeigt.
- Eine PowerPoint-Präsentation wurde ebenfalls erstellt, in der wir unsere Arbeit und die wichtigsten Ergebnisse anschaulich zusammengefasst haben.

Erfolge

- Wir hatten eine erfolgreiche Präsentation, bei der wir unsere Methoden und Ergebnisse vorgestellt haben. Das Video und die PowerPoint-Präsentation haben dabei eine wichtige Rolle gespielt.
- Wir konnten sicherstellen, dass wir auf dem richtigen Weg sind und die nächsten Schritte gut geplant haben.

7. Ausblick: Fortsetzung im nächsten Semester

Im kommenden Semester wird das Projekt im Rahmen von InnoLab 3 fortgesetzt. Wir möchten die beste Matching-Methode weiterentwickeln, das Antwortverhalten optimieren und neue Features wie interaktive Nutzerabfragen und adaptives Feedback einbauen. Ein weiterer Schwerpunkt wird auf der Evaluation und Usability-Tests liegen, um den Prototypen fit für den produktiven Einsatz zu machen.