

Expose: Seminar „Innovative Ideen“ – Jasper Slowik

Einleitung

Mein Name ist Jasper Slowik, Bachelorstudent der Wirtschaftsinformatik. Meine Leidenschaft liegt in den Bereichen Künstliche Intelligenz (KI) und Machine Learning, insbesondere in deren praktischer Anwendung. Durch Online-Kurse zu neuronalen Netzen (z. B. CNNs) und praktische Erfahrungen mit Python, TensorFlow und Flutter habe ich mir fundierte Grundlagen angeeignet. Diese Kenntnisse möchte ich im Seminar nutzen, um ein innovatives KI-Tool für Universitäten zu entwickeln.

Aktueller Stand

Mein Projekt ist ein intuitives Werkzeug, das Universitäten ermöglicht, spezialisierte Chatbots für Studierendenfragen zu erstellen. Über eine einfache Oberfläche können Universitäten Dokumente wie Studienpläne oder Modulhandbücher hochladen und im Anschluss werden diese automatisch ausgewertet. Der Chatbot kann auf die bereitgestellten Inhalte zugreifen und präzise Antworten auf ganz spezifische Fragen wie „Wann ist die Anmeldefrist für Modul X?“ Aber auch breitere Fragen wie „Ich studiere X möchte in Zukunft in Job Y arbeiten, welche Module kannst du mir empfehlen?“ geben. Ziel ist es, den Zugang zu Informationen in einer simplen und interaktiven Weise zu ermöglichen.

Technische Umsetzung

Grafische Oberflächen

Die Benutzeroberfläche, entwickelt mit dem Flutter-Framework, umfasst zwei intuitive, plattformübergreifende Web-/App-Schnittstellen. Die erste ermöglicht das einfache Erstellen von Projekten und das Hochladen von Dateien. Die zweite bietet eine benutzerfreundliche Interaktion mit dem Chatbot, über die Studierende ihre Fragen stellen und präzise Antworten erhalten können.

Firebase

Firebase dient als Backend zur zuverlässigen Speicherung hochgeladener Dateien und deren Zuordnung zu Projekten, wodurch eine strukturierte Datenverwaltung gewährleistet wird.

Python-Server

Ein zentraler Python-Server verbindet die Benutzeroberflächen mit den Backend-Diensten. Er verwaltet eine Vektordatenbank, die für das Vektorisieren, Hochladen, Löschen und die Ähnlichkeitssuche der vom Nutzer hochgeladenen Dateien zuständig ist. Darüber hinaus übernimmt der Server die Retrieval Augmented Generation (RAG), um den Kontext sowie das zugrunde liegende Wissen des Chatbots zu erweitern. Er steuert die gesamte Kommunikation mit dem Chatbot, empfängt Nutzerfragen und generiert die entsprechenden Antworten, die in der Benutzeroberfläche bereitgestellt werden.

Ziele für das Seminar

Ziel ist es, einen funktionsfähigen Prototypen zu erstellen, der im Rahmen der Abschlusspräsentation vorgestellt werden kann.

Über einen Simulator werden beide Benutzeroberflächen gestartet, und der gesamte Prozess, vom Datei-Upload bis zur Antwortgenerierung, wird vollständig durchgespielt und getestet. Dabei sollen folgende Eigenschaften bzw. Charakteristiken erfüllt sein:

Technische Umsetzung

Die technische Umsetzung wie oben beschrieben: Eine stabile Architektur mit einem zentralen lokalen Python-Server, der die beiden Benutzeroberflächen mit dem Backend verbindet. Vektordatenbank, moderne Suchalgorithmen und RAG ermöglichen eine flexible und skalierbare Funktionalität.

Zugänglichkeit für Nutzer ohne Vorkenntnisse

Das Ziel ist eine intuitive Bedienung, sodass auch Nutzer ohne KI-spezifische Erfahrung das Tool intuitiv bedienen können. Der Fokus liegt auf einer benutzerfreundlichen Oberfläche, weshalb Einstiegshürden minimiert werden.

Interaktive Anleitung zur Dokumentvorbereitung und Nutzung

Um die Nutzung des Chatbot-Erstellungssystems sehr benutzerfreundlich und intuitiv zu gestalten, aber dennoch qualitativ hochwertige Ergebnisse zu ermöglichen, wird eine interaktive Anleitung in Form eines modalen Fensters bereitgestellt. Diese erscheint beim Öffnen des Chatbot-Moduls und kann jederzeit über ein Hilfesymbol erneut angezeigt werden. Die Anleitung erklärt, welche Dateiformate unterstützt werden, wie diese strukturiert sein sollten und gibt weitere allgemeine Empfehlungen und Richtlinien zur Erstellung eines Chatbots, um optimale Ergebnisse zu erzielen. Darüber hinaus vermittelt sie ein klares Verständnis der Fähigkeiten und Grenzen des erstellten Chatbots, wie etwa die Arten von Fragen, die der Bot effektiv beantworten kann, sowie Beschränkungen bei der Verarbeitung komplexer Anfragen. Diese Funktion stellt sicher, dass auch ohne technisches Vorwissen ein sinnvoller und effektiver Einsatz des Tools möglich ist.

Nahtloser Workflow

Der gesamte Prozess, vom Datei-Upload über die automatische Verarbeitung bis hin zur Interaktion mit dem Chatbot, verläuft flüssig und weitgehend im Hintergrund. Wartezeiten werden so kurz wie möglich gehalten, um den Nutzerfluss nicht zu unterbrechen. Ziel ist eine durchgängig reibungslose Nutzung.

Mehrwert für den Nutzer

Das Tool zielt darauf ab, eine nutzerorientierte Anwendung bereitzustellen, die den Endnutzern durch kontextbezogene Suche und intelligente Antwortgenerierung einen relevanten Mehrwert bietet. Es wird anerkannt, dass Large Language Models in seltenen Fällen ungenaue oder erfundene Inhalte liefern können, etwa durch Halluzinationen, bei denen das Modell plausible, aber falsche Informationen generiert. Durch RAG-Techniken, die sich strikt auf die bereitgestellten Dokumente stützen, sowie gezielte Steuerung des Modells, werden solche Phänomene minimiert, können jedoch nicht vollständig ausgeschlossen werden.

Innovation und Alleinstellungsmerkmal

Mein Ansatz hebt sich durch folgende Punkte ab:

- No-Code-Lösung: Universitäten können den Chatbot ohne technische Kenntnisse per File-Upload nutzen.
- Ressourceneffizienz: Distilled Models ermöglichen kostengünstige und nachhaltige KI-Nutzung.
- Spezialisierung: Der Fokus auf universitäre Anforderungen unterscheidet das Tool von generischen Chatbots.
- Sofortige Nutzbarkeit: Nach dem Upload ist der Chatbot direkt einsatzbereit.