

# Entwicklung eines KI-basierten FAQ-Chatbots für die Hochschule im Bereich Prüfungsangelegenheiten

Prof. Dr. Harald Ritz  
Technische Hochschule Mittelhessen  
Fachbereich MNI  
Wiesenstraße 14  
35390 Gießen  
[harald.ritz@mni.thm.de](mailto:harald.ritz@mni.thm.de)

Dogus Tansel (B.Sc.)  
Technische Hochschule Mittelhessen  
Fachbereich MND  
Wilhelm-Leuschner-Straße 13  
61169 Friedberg  
[dogus.tansel@mnd.thm.de](mailto:dogus.tansel@mnd.thm.de)

## ABSTRACT

Es ist davon auszugehen, dass die Nutzung von Chatbots in den nächsten Jahren enorm ansteigen wird. Seit zum ersten Mal der KI-Chatbot „ChatGPT“ Ende 2022 vorgestellt wurde [1, p. 423], ist dieser Trend auch in der breiten Öffentlichkeit angekommen. Es wird erwartet, dass im Jahr 2027 der weltweite Chatbot-Markt einen Umsatz von 454,8 Millionen US-Dollar erzielen wird, gegenüber 40,9 Millionen US-Dollar im Jahr 2018 [2]. Als Hochschule, bei der das Leitbild auf das Bewältigen von Herausforderungen unter Verwendung neuester Erkenntnisse ausgerichtet ist [3], lag es nahe, zur Verbesserung der studentischen Beratungssituation, diesen Trend frühzeitig aufzugreifen.

Der vorliegende Artikel stellt das vom Prüfungsausschuss-Vorsitzenden der Wirtschaftsinformatik an der Technischen Hochschule Mittelhessen im Verlauf des Jahres 2021 initiierte Projekt vor, das die Entwicklung eines KI-basierten FAQ-Chatbots im Bereich Prüfungsangelegenheiten zum Ziel hat.

Der Chatbot wurde entworfen, um Studierende bei häufig gestellten Fragen (FAQs) im Zusammenhang mit Prüfungsangelegenheiten im Bereich Wirtschaftsinformatik zu unterstützen. Es ist vorgesehen, dass mit Hilfe von Machine-Learning-Algorithmen der KI-basierte Chatbot kontinuierlich dazulernt und seine Antwortgenauigkeit und -effektivität verbessert. Der Chatbot bietet den Studierenden eine einfache Möglichkeit, schnell und unkompliziert Antworten auf ihre Fragen zu erhalten und kann auch außerhalb der regulären Bürozeiten genutzt werden. Die Integration des Chatbots in die Website der Hochschule trägt dazu bei, die Effizienz der Studierendenberatung zu erhöhen und das Nutzererlebnis für die Studierenden zu verbessern.

## SCHLÜSSELWÖRTER

Adam Optimizer, Angular, Chatbot, Deep Learning, Docker, FAQ-Bot, Flask-Framework, Unicorn, Hochschulberatung, Keycloak, Künstliche Intelligenz (KI), Machine Learning (ML), MongoDB, Natural Language Processing (NLP), Natural Language Toolkit (NLTK), Neuronale Netze, NGINX-Webserver, Python, PyTorch, Reinforcement Learning, REST-API, Sentence Similarity Model, spaCy, Transformers

### I. AUSGANGSSITUATION

Studierende an einer Hochschule sind oft mit vielen Fragen und Unsicherheiten im Zusammenhang mit Prüfungsangelegenheiten konfrontiert. Dies kann zu einem hohen Beratungsbedarf führen, der Ressourcen und Zeit des ohnehin eingeschränkten Hochschulpersonals in Anspruch nimmt. Darüber hinaus können sich Studierende aufgrund von begrenzten Büro- und Wartezeiten frustriert und unzufrieden fühlen. Um diese Herausforderungen zu bewältigen, wird im Rahmen eines internen Hochschulprojektes an der Technischen Hochschule Mittelhessen (THM) ein KI-basierter FAQ-Chatbot entwickelt,

der den Studierenden eine schnelle und effektive Möglichkeit bietet, Antworten auf ihre Fragen zu erhalten und dadurch Unsicherheiten zu beseitigen. Der Chatbot soll die Effizienz der Studierendenberatung erhöhen und das Nutzererlebnis der Studierenden verbessern sowie optimieren.

### II. VERWANDTE ARBEITEN

In der Forschung gibt es inzwischen vielfältige Arbeiten, die sich mit der Entwicklung von Chatbots und virtuellen Assistenten für die Hochschulberatung beschäftigen. Einige dieser Arbeiten haben sich auf die Verwendung von Natural Language Processing (NLP) und Machine-Learning (ML)-Algorithmen zur Verbesserung der Antwortgenauigkeit und -effektivität von Chatbots konzentriert [4]. Andere Arbeiten setzen ihren Schwerpunkt auf die Integration von Chatbots in bestehende Hochschul-Websites und -Plattformen, um eine nahtlose Benutzererfahrung für die Studierenden zu gewährleisten [5]. Insgesamt haben diese Arbeiten gezeigt, dass Chatbots ein großes Potenzial für die Verbesserung der Hochschulberatung und der Studierendenerfahrung haben, insbesondere in Bezug auf die Beantwortung häufig gestellter Fragen (FAQs) und die Entlastung des Hochschulpersonals.

### III. ANFORDERUNGEN

Das Ziel des Hochschulprojektes „Winfy“ ist es, einen KI-basierten Chatbot und eine entsprechende administrative Verwaltungsplattform für den Bereich Prüfungsangelegenheiten für die gemeinsam von den Fachbereichen MND und MNI angebotenen Studiengänge B.Sc. und M.Sc. Wirtschaftsinformatik an der THM zu entwickeln. Die Umsetzung eines solchen Vorhabens erfordert eine sorgfältige Analyse der Anforderungen, die von den Stakeholdern des Systems gestellt werden. Um dieses Ziel zu erreichen, sind insbesondere folgende Anforderungen zu erfüllen [6] [7]:

#### Benutzerfreundlichkeit

Der Chatbot sollte unkompliziert zu bedienen sein und eine intuitive Benutzeroberfläche bieten. Die Benutzer sollten in der Lage sein, ihre Fragen schnell und einfach zu stellen und die Antworten auf ihre Fragen in kurzer Zeit zu erhalten.

#### Zuverlässigkeit

Der Chatbot sollte in der Lage sein, zuverlässige und genaue Antworten basierend auf den Fragen der Benutzer zu liefern. Dabei sind die gestellten Fragen effektiv und effizient zu verarbeiten und Antworten auf klare und verständliche Weise zu präsentieren.

#### Flexibilität

Der Chatbot sollte flexibel genug sein, um auf die verschiedenen Bedürfnisse der Benutzer eingehen zu können. Hierbei soll realisiert sein, auf eine Vielzahl von Fragen zu antworten und eine breite Palette von Themen abzudecken. Der Chatbot sollte auch in der Lage sein, sich an neue Situationen und Anforderungen anzupassen und schnell auf Veränderungen zu reagieren.

#### Sicherheit und Datenschutz

Aufgrund der Möglichkeit mit der Plattform im öffentlichen Netz zu interagieren, ist es wichtig sicherzustellen, dass die Daten auf den zugrundeliegenden Server sicher und geschützt sind. Daher sollten alle notwendigen Sicherheitsmaßnahmen implementiert werden, um zu gewährleisten, dass das System vor unbefugten Zugriff und Missbrauch geschützt ist.

#### Barrierefreiheit

Der Chatbot sollte barrierefrei und allen Benutzern zugänglich sein, unabhängig von ihrer Fähigkeit oder Behinderung. Es sollte in der Lage sein, die Fragen der Benutzer in einer Vielzahl von Formaten zu verarbeiten, einschließlich Sprache und Text, und die Antworten auf eine Weise zu präsentieren, die für alle Benutzer leicht verständlich ist.

Das Erfüllen dieser (geplanten) Anforderungen wird dazu beitragen, dass der Chatbot effektiv und nützlich für die Studierenden der Wirtschaftsinformatik ist und dazu

beiträgt, die Effizienz und Genauigkeit der Prüfungsverwaltung an der Hochschule zu verbessern.

### IV. ENTWICKLUNG VERSION 1.0

#### Methodik

In einer ersten Entwicklungsstufe des Hochschul-Chatbots „Winfy“ wurde ein regelbasiertes System in Python implementiert, um die Fähigkeit zu verbessern, menschliche Sprache zu verstehen und auf Fragen ordnungsgemäß zu antworten [8].

Für die Entwicklung des FAQ-Bots wurden verschiedene Bibliotheken und Technologien verwendet. Zum einen wurde die Python-Bibliothek Transformers genutzt, die eine breite Palette von vortrainierten Modellen für NLP anbietet. Insbesondere wurde das „German Semantic“ (Sentence Similarity Model) [9] eingesetzt, das es dem Chatbot ermöglicht, Texte semantisch zu vergleichen. Des Weiteren wurde die Bibliothek spaCy [10] genutzt, die ebenfalls NLP-Technologien bereitstellt. Hierbei wurde das spaCy (Sentence Similarity Model) eingesetzt, um die Ähnlichkeit zwischen verschiedenen Eingabemessages und Antworten des Chatbots zu bestimmen.

Zur Datenhaltung wurde mit MongoDB eine NoSQL-Datenbank verwendet, die eine flexible und skalierbare Speicherung von Daten ermöglicht. Diese ist mit den häufig gestellten Fragen und Antworten ausgestattet, um dem System, als auch dem Benutzer, schnelle und präzise Antworten zur Verfügung zu stellen.

Um eine RESTful-Webanwendung zu ermöglichen, wurde eine REST-API mit Hilfe des Flask-Frameworks [11] in Python erstellt, das in der Lage ist, Nachrichten zu empfangen und zu senden. Das Flask-Framework ist ein beliebtes Python-Web-Framework, das für seine Einfachheit und Flexibilität bekannt ist. Durch die Implementierung des RESTful-Web Services ist der Chatbot in der Lage, mit anderen Anwendungen und Services zu kommunizieren. Mit Hilfe von HTTP-Anfragen kann der Chatbot Nachrichten von außerhalb empfangen und darauf antworten. Das RESTful-Design macht den Chatbot skalierbar, da er auf einfache Weise in andere Anwendungen integriert werden kann.

Zur Bereitstellung des Webservers wurde der NGINX-Webserver [12] in der Funktion eines Reverse Proxys eingesetzt, der die Anfragen an den Chatbot weiterleitet. Unicorn [13] wurde als Webserver Gateway Interface HTTP Server genutzt, um den Service bereitzustellen.

Für das Backend des Chatbots wurde das TypeScript-basierte Webapplikationsframework Angular [14] genutzt. Die Landingpage wurde mit einem HTML-Grundgerüst, einer JavaScript-Datei für die Funktionalitäten des Chatbots und einer CSS-Datei für das Styling der Oberfläche erstellt. Die Benutzeroberfläche des Chatbots wurde so gestaltet, dass sie einfach und intuitiv zu bedienen ist. Eine Kompatibilität mit verschiedenen Endgeräten wie

Mobiltelefon und Tablet wurde zunächst nicht realisiert. Zum Zeitpunkt des Starts der Entwicklung bestand der Fokus auf die Umsetzung einer Version für den Desktop-Computer (siehe Abbildung 1).



Abbildung 1: Landingpage - Startbildschirm

Zusätzlich zu den oben genannten Bibliotheken und Frameworks wird der Chatbot in einer Docker-Container-Umgebung betrieben. Hierfür wurden verschiedene Services, wie MongoDB, Keycloak, PostgreSQL und NGINX, als separate Container aufgebaut und über Docker Compose orchestriert. Dies erleichtert die Bereitstellung und Skalierung des Chatbots sowie die Integration in die vorhandene IT-Infrastruktur der Hochschule. Durch die Containerisierung können auch Abhängigkeiten zwischen den verschiedenen Komponenten des Chatbots einfach verwaltet werden. Weiterhin wird dadurch eine hohe Flexibilität in Bezug auf die Infrastruktur erreicht, da so der Chatbot problemlos auf verschiedenen Servern und Cloud-Plattformen betrieben werden könnte.

### Funktionalitäten

Nach dem Aufruf der Webseite des Frontends wird der Anwender vom Chatbot in einer Dialogbox aufgefordert zu entscheiden, ob er Fragen bezüglich des Bachelor- oder Masterstudiums hat. Hierfür muss der Nutzer die entsprechende Fläche auf der Webseite anklicken. Erst nach dieser Selektion steht es dem Chatbot frei, Fragen des Nutzers zu beantworten. Diese Fragen können in Form eines üblichen Chat-DIALOGS gestellt werden (siehe Abbildung 2). Der Chatbot bietet dem Nutzer auch vordefinierte Fragen in Form einer Autovervollständigung an, die aufgrund der vorselektierten Studienrichtung ausgewählt werden können.

Aktuell ist im Produktivbetrieb das „German Semantic“-Sprachmodell eingestellt. Erreichbar für Studierende ist der FAQ-Chatbot „Winfy“ auf der Hochschulseite über die URL: <https://www.thm.de/mnd/studium/service/chatbot-wirtschaftsinformatik>.

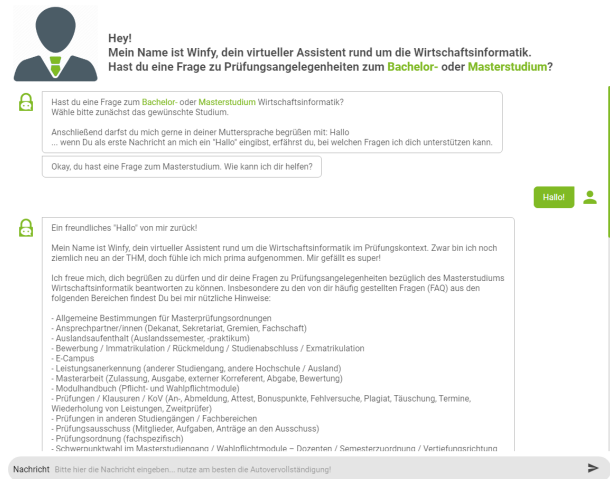


Abbildung 2: Landingpage - Dialogfenster

Beim Aufruf der Oberfläche erfolgt eine Anmeldung im REST-Service des Backends über das Authentifizierungs- und Autorisierungssystem Keycloak, das in den Docker-Containern des Backends integriert ist. Nach erfolgreicher Anmeldung werden notwendige Ressourcen, wie zum Beispiel die Antwort auf eine gestellte Frage, geladen. Über die in der JavaScript-Datei hinterlegten Methoden werden Inhalte an die entsprechenden TypeScript-Funktionen gesendet. Diese Funktionen rufen wiederum HTTP-Requests im Python-Code des Chatbots auf, um die notwendigen Ressourcen anzufordern. Dieser Prozess ermöglicht eine schnelle und effiziente Kommunikation zwischen dem Front- und Backend des Chatbots (siehe Abbildung 3).



Abbildung 3: Auszug REST-Aufrufe

Das Backend ist für Administratoren und Verantwortliche zugänglich und bietet eine Weboberfläche zur Verwaltung des Chatbots und dessen Inhalte. Innerhalb des Backends haben berechtigte Benutzer die Möglichkeit, neue Inhalte wie Fragen und Antworten in die Umgebung einzuspielen, Sprachmodelle zu ändern oder auf neue FAQ-Inhalte zu trainieren, Benutzerkonten zu erstellen und Log-Dateien bzw. Chat-Protokolle zu analysieren sowie die errechneten Score-Werte für Anfragen einzusehen. Dies ermöglicht eine einfache Verwaltung des Chatbots und eine schnelle Anpassung an neue Anforderungen und Inhalte.

Auf der Startseite des Backends wird der Nutzer aufgefordert, seine Anmeldedaten einzugeben. Dabei handelt es sich um einen Benutzernamen und ein Passwort, die vordefiniert sind und innerhalb der Plattform geändert werden können. Nach der Eingabe und dem Bestätigen

mit dem Login-Button sendet die Webanwendung eine REST-Anfrage an den Host-Server. Der Host-Server prüft daraufhin, ob die Benutzerdaten gültig sind, und gleicht sie in einer separaten HTTP-Anfrage mit dem Keycloak-Authentifizierungsserver ab. Wenn der Benutzer vorhanden ist und die eingegebenen Daten mit dem Nutzerprofil übereinstimmen, wird der Benutzer erfolgreich angemeldet und erhält ein Session- bzw. Bearer-Token zugewiesen. Dieser Token ist für eine bestimmte Laufzeit innerhalb der Umgebung gültig und ermöglicht dem Benutzer den Zugriff auf die verschiedenen Funktionen des Backends.

Diese Funktionsbereiche, z.B. das Management des Chatbots mit Verwalten der Anfragen und Antworten, der Einstellungsbereich mit Auswahl der Sprachmodelle und Verwalten der Log-Einträge, das Dashboard zur Analyse der Ergebniswerte des Chatbot-Verhaltens und die Benutzerverwaltung, sind aus der Abbildung 4 ersichtlich.

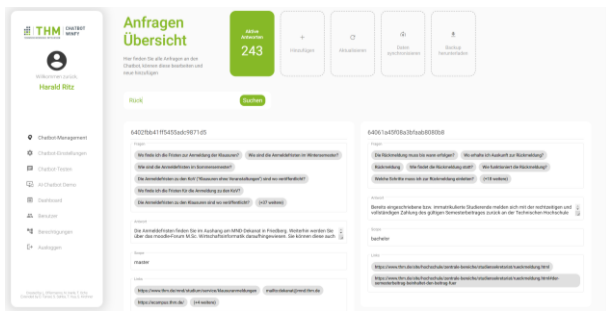


Abbildung 4: Backend-Oberfläche

## V. THEORETISCHER RAHMEN

Um auf der Grundlage der entwickelten Plattform eine Erweiterung im Rahmen eines KI-Chatbots zu realisieren, wird in diesem Abschnitt zunächst auf den theoretischen Hintergrund und dann im darauffolgenden Kapitel auf die Implementierung eingegangen.

Die einfachste Form computergesteuerter Dialoge bilden regelbasierte Chatbots, die auch als Entscheidungsbaum-Chatbots bezeichnet werden. Diese Art der Chatbots lernen im Laufe der Zeit nicht dazu und arbeiten lediglich in den vordefinierten Szenarien, für die sie konzipiert wurden. In diese Kategorie ist die aktuell schon im Hochschulbetrieb befindliche Version 1.0 des FAQ-Chatbots „Winfy“ einzuordnen. Ein Beispiel für fortschrittlichere Chatbots sind Chatbots mit maschinellem Lernen, auch als Chatbots mit künstlicher Intelligenz (KI) bezeichnet. Sie ermöglichen den Benutzern, offene Fragen zu stellen und bieten die natürlichsten Antworten. Im Gegensatz zu einfachen regelbasierten Chatbots lernen sie aus Gesprächen und verbessern ihre Antworten im Laufe der Zeit [15].

Im Folgenden werden einige grundlegende Begriffe und Definitionen erläutert, die für das Verständnis von KI-Chatbots und deren Funktionsweise relevant sind.

## Machine Learning

Machine Learning ist ein Teilbereich der künstlichen Intelligenz, bei dem Algorithmen entwickelt werden, die aus Daten lernen können. Es geht darum, automatisch Muster in Daten zu erkennen, um Vorhersagen treffen zu können. Dabei werden verschiedene Techniken wie Klassifikation, Regression, Clustering und Reinforcement Learning eingesetzt [16, pp. 1-14].

## Deep Learning

Deep Learning ist ein Teilbereich des maschinellen Lernens, der sich mit künstlichen neuronalen Netzen beschäftigt. Hierbei werden mehrere Schichten von Neuronen hintereinandergeschaltet, um komplexe Funktionen zu approximieren. Durch das Lernen aus Daten können neuronale Netze komplexe Aufgaben wie Bild- oder Spracherkennung ausführen [17].

## Neuronale Netze

Neuronale Netze sind ein Konzept aus der Neurobiologie, welches in der Informatik adaptiert wurde. Ein neuronales Netz ist eine Verbindung von Neuronen, die in Schichten organisiert sind und Signale durch das Netzwerk weitergeben. Ein Vorwärtsnetz ist eine spezielle Art von neuronalem Netzwerk, in dem die Signale nur in eine Richtung fließen, von der Eingangsschicht zur Ausgangsschicht [18].

## Natural Language Toolkit

Natural Language Toolkit (NLTK) ist eine Open-Source-Bibliothek für die Sprachverarbeitung mit Python. NLTK bietet Funktionen wie Tokenisierung, Lemmatisierung und POS-Tagging, um Texte automatisch zu analysieren und zu verarbeiten [19]. Es ist eine wichtige Bibliothek für die Entwicklung von KI-Chatbots.

## Lemmatisierung

Lemmatisierung ist ein Prozess der Textnormalisierung, bei dem die Grundform eines Wortes, das Lemma, ermittelt wird. Es ist eine wichtige Technik der Sprachverarbeitung, um verschiedene Formen eines Wortes in eine einheitliche Form zu bringen [19]. Durch Lemmatisierung können KI-Chatbots besser verstehen, was der Nutzer meint, da verschiedene Formen desselben Wortes als gleichwertig behandelt werden (siehe Abbildung 5).

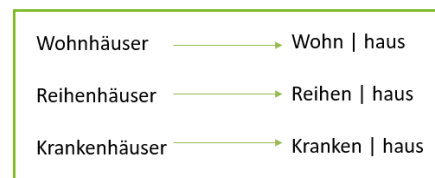


Abbildung 5: Lemmatisierung

## Tokenisierung

Tokenisierung ist der Prozess der Aufteilung eines Textes in sinnvolle Einheiten, genannt Tokens. Ein Token

kann ein Wort, Satzzeichen oder eine Phrase sein. Durch die Tokenisierung kann ein Text automatisch analysiert werden, um Muster zu erkennen [19]. Dies ist eine essenzielle Technik für KI-Chatbots, um die Absichten des Nutzers zu verstehen.

### Bag of Words

Bag of Words ist ein Konzept der Textklassifizierung, bei dem jeder Text als eine Sammlung von Wörtern betrachtet wird, ohne Berücksichtigung ihrer Reihenfolge oder Grammatik. Jedes Wort wird als Merkmal angesehen und ein Modell wird trainiert, um Texte auf Basis dieser Merkmale zu klassifizieren [20]. Bag of Words wird verwendet, um die Absichten des Nutzers zu verstehen und passende Antworten zu generieren.

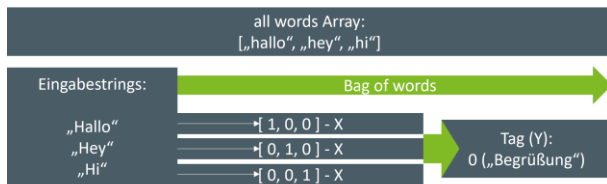


Abbildung 6: Bag of Words

In der Abbildung 6 wird Bag of Words eingesetzt, um jedes Muster aus den Tokens der gegebenen Strings als Vektor abzubilden, damit diese zum Training und Klassifizierung eines Modells genutzt werden können. Dabei bildet jedes Pattern eine Reihung mit der Größe des „all words“-Array, um zu überprüfen, ob die Tokens der Eingabe sich im „all words“-Array befinden, um diese einem Tag zuzuweisen und die Eingabe zu klassifizieren. Jeder Tag wird in einem Array mit einem Label versehen und erhält eine Nummer, die ebenfalls als Vektor agiert und das Y der Trainingsdaten darstellt.

## VI. ENTWICKLUNG VERSION 2.0

Nachfolgend wird der Prozess beschrieben, wie der Chatbot mit Machine-Learning-Technologien weiterentwickelt wurde. Dazu werden die genutzten Technologien, die Architektur und die Implementierungsschritte erläutert [21].

### Technologien

Der KI-Chatbot „Winfy“ in Version 2.0 wurde auf Basis von PyTorch entwickelt und verwendet gängige Verarbeitungsmethoden sowie Bibliotheken. PyTorch überzeugt als Open-Source-Bibliothek im Rahmen von Deep-Learning-Anwendungsfällen mit einer hohen Geschwindigkeit und enormer Flexibilität zur Entwicklung und Implementierung von künstlichen neuronalen Netzen. Während der Nutzung von PyTorch kann ohne Komplikationen auf etablierte Python-Bibliotheken wie NumPy, SciPy und Cython zugegriffen werden. Modelle werden mittels des Compilers und Torch-Skripten erstellt [22]. Ein zentrales Konzept ist der Aufbau eines neuronalen Netzes zur Vorhersage der Absicht des Benutzers und des Erstellens der Antwort des Chatbots.

### Architektur

Die Architektur des KI-Chatbots (siehe Abbildung 7) besteht aus einem zentralen Knoten, dem Chatbot selbst, der durch PyTorch umgesetzt wurde. Das Modelltraining ist in zwei Bereiche unterteilt: das Trainingsdatenset und die Ergebnisse.

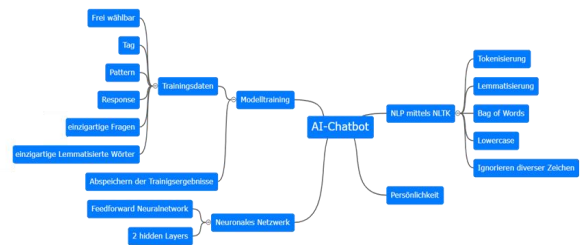


Abbildung 7: KI-Chatbot-Architektur

Das neuronale Netzwerk wird als Feedforward Neural Network mit zwei Hidden-Layern erstellt (siehe Abbildung 8), um die Verarbeitung von Informationen zu ermöglichen.

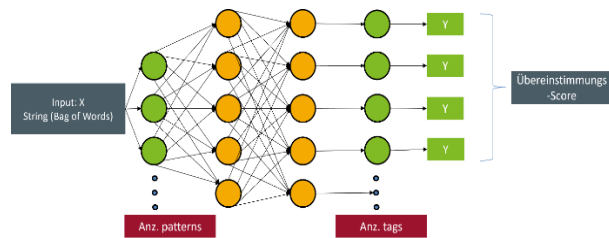


Abbildung 8: Feedforward-Netz

Zum Verstehen und Verarbeiten der menschlichen Sprache wird NLP und das NLTK eingesetzt. Um Textdaten in eine strukturierte Form umzuwandeln, werden die Methoden der Tokenisierung, Bag of Words und Lemmatisierung verwendet. Diese Technologien bilden die Grundlage für das Verarbeiten der Benutzeranfragen und das Erstellen von Antworten durch den Chatbot, um eine effektive Sprachverarbeitung zu gewährleisten.

### Datensatz

Der Trainingsdatensatz basiert auf dem Inhalt der MongoDB. Diese Daten beziehen sich auf die Studiengänge Bachelor und Master Wirtschaftsinformatik der THM. Im Bereich Datensatz bildet eine FAQ-Liste der Hochschule die Grundlage für die Entwicklung des KI-Chatbots. Um diesen Datensatz in PyTorch verarbeiten zu können, wird er im Rahmen einer JSON-Datei strukturiert. Dabei wird ein "intents" als Fragenkatalog erstellt, der Fragen einer bestimmten Kategorie einem "tag" zuordnet. Innerhalb jedes "intents" werden "patterns" definiert, die mögliche Fragen darstellen, die von einem Nutzer gestellt werden könnten. Weiterhin werden "responses" definiert, die eine mögliche Antwort des Chatbots auf die jeweiligen Fragen inklusive relevante URLs beinhalten.

```

{
  "intents": [
    {
      "tag": "Anerkennung von Leistungen",
      "patterns": [
        "Kann ich Prüfungen aus einem anderen Studiengang anrechnen lassen?",
        "Ich benötige eine Auskunft zum Anrechnen von Leistungen",
        "Wie kann ich meine bereits bestandenen Prüfungen anrechnen lassen?",
        "...",
      ],
      "responses": [
        "Sie müssen zunächst selbst tätig werden und prüfen, was anerkennbar sein könnte. Welche Module ggf. anerkennbar sind, können Sie dem Modulhandbuch entnehmen, das Sie auf der WWW-Seite des Prüfungsausschusses Wirtschaftsinformatik finden. (...)"
      ]
    }
  ]
}

```

## Modellaufbau

Der Modellaufbau des KI-Chatbots basiert, wie erwähnt, auf einem Feedforward-Neuronalen-Netzwerk, welches mit PyTorch erstellt wurde. Das Netzwerk besteht aus drei Schichten: Input-, Hidden- und Output-Layer. Der Input-Layer erhält die Eingabedaten und leitet diese an die Hidden-Layer weiter, welche die Daten verarbeiten und transformieren. Der Output-Layer gibt schließlich das Ergebnis aus. Die Anzahl der Hidden-Layer und die Größe der Neuronen in diesen Schichten wurden experimentell ermittelt und optimiert. Der Modellaufbau wird durch die verwendete Activation Function „ReLU“ unterstützt, der zur Beschleunigung des Trainings beiträgt.

```

class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        return out

```

Das neuronale Netzwerk des KI-Chatbots ist in der Lage, die Daten im Training zu verarbeiten und dadurch Muster zu erkennen, um die korrekten Antworten auf Benutzeranfragen zu generieren. Der Input-Layer besteht aus einer

Anzahl von Neuronen, die der Anzahl der Token entspricht, die als Eingabe erwartet werden. Der Output-Layer enthält so viele Neuronen, wie es Tags gibt, die von der FAQ-Liste abgedeckt werden. Jeder Knoten im Output-Layer entspricht einem Tag und ist mit einer ReLU-Aktivierungsfunktion verknüpft, die eine Wahrscheinlichkeitsverteilung über die möglichen Tags ausgibt.

## Datenverarbeitung

In diesem Abschnitt wird die Datenverarbeitung für den Chatbot beschrieben. Hierbei wird die Eingabe der Fragen und Antworten in einzelne Wörter (Tokens) zerlegt und auf den Wortstamm zurückgeführt (Lemmatisierung). Dies dient dazu, Synonyme und verschiedene Formen eines Wortes als identisch zu behandeln, und somit die Trefferquote bei der Erkennung von Nutzeranfragen zu erhöhen. Anschließend werden diese Wörter in eine Bag-of-Words-Repräsentation umgewandelt, die angibt, welche Wörter in der Eingabe vorhanden sind und welche nicht. Dies ermöglicht es dem Chatbot, die Intention des Nutzers zu erkennen und die passende Antwort darauf zu generieren.

```

def tokenize(sentence):
    return nltk.word_tokenize(sentence)

def lem(word):
    return lemmatizer.lemmatize(word.lower())

def bag_of_words(tokenized_sentence, words):
    sentence_words = [lemmatizer.lemmatize(word.lower()) for word in tokenized_sentence]

    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1
    return bag

```

## Modelltraining

Das Trainieren des Modells beginnt mit dem Erstellen der Trainingsdaten. Hierzu werden die Sätze aus der intents.json-Datei durchlaufen und jedes einzelne Wort wird in einem Array gespeichert. Die Tokens werden lemmatisiert und in Kleinbuchstaben umgewandelt, um Duplikate entfernen und eine Sortierung durchführen zu können. Anschließend werden die Trainingsdaten erstellt, indem jedes Muster in eine Bag-of-Words-Repräsentation umgewandelt wird, die es dem Modell ermöglicht, die Ähnlichkeit zwischen verschiedenen Mustern zu berechnen.

Die Trainingsdaten werden dann in eine PyTorch-Dataset-Klasse verpackt, um sie in einem DataLoader zu laden und den Trainingsprozess zu beschleunigen. Anschließend wird das Modell definiert, das aus Input-, Hid-

den- und Output-Layer besteht. Es wird ein Cross-Entropy-Loss als Verlustfunktion und Adam als Optimizer verwendet. Mit Blick auf die Hyperparameter wurde eine Trainingsdauer von 1000 Epochen mit einer Lernrate von 0.001 festgelegt.

Das Modell wird auf der GPU oder CPU trainiert, je nachdem, was verfügbar ist. Der Trainingsprozess wird für die festgelegte Anzahl von Epochen durchlaufen. In jeder Epoche werden die Daten in Batches von bestimmter Größe geladen und für jedes Batch die Vorhersage, der Verlust und die Gradienten berechnet. Nachdem die Gradienten berechnet wurden, werden sie zurückpropagiert, um die Gewichte des Modells zu aktualisieren. Wenn die Anzahl an angegebenen Epochen durchlaufen ist, wird das Modell gespeichert. Nach der erfolgreichen Anbindung an das Frontend steht dem Nutzer eine neue Version des erweiterten KI-Chatbots zur Verfügung.

## VII. ERGEBNISSE

Innerhalb des Projektes wurde zunächst mit „Winfy v1.0“ ein regelbasierter FAQ-Chatbot konzipiert und implementiert. Dieser kann verschiedene vortrainierten NLP-Modellen nutzen und liefert mit dem eingestellten „German Semantic“-Modell sehr gute Ergebnisse, wie die ersten Erfahrungen und eingegangenen Rückmeldungen der studentischen Nutzer seit der Freischaltung zum Sommersemester 2023 im Hochschulalltag zeigen.

Anschließend wurde mit der Version 2.0 von „Winfy“ ein KI-Chatbot mithilfe von Deep Learning erstellt. Der Chatbot ist in der Lage, Texteingaben zu interpretieren und passende Antworten zu generieren. Dazu wurden die Python-Bibliotheken PyTorch und NLTK verwendet. Die Daten für das Training des Modells wurden aus einer JSON-Datei gelesen. Nach der Vorverarbeitung der Daten wurden sie in Trainings- und Testdaten aufgeteilt und in Form von Batches an das Modell übergeben.

Es kann im Prototyp festgestellt werden, dass „Winfy v2.0“ vielversprechende Antworten liefert und mit einer Vielzahl von Nutzeranfragen umgehen kann. Natürlich gibt es noch Verbesserungsmöglichkeiten, insbesondere in Bezug auf die Handhabung von unerwarteten oder unbekanntem Nutzereingaben. Ein Beispiel dafür bildet das Verarbeiten von Buchstabendreher beim Versenden der Anfrage an den Chatbot. Um diese Probleme anzugehen, könnte der Chatbot mit erweiterten Funktionen zur Textnormalisierung ausgestattet werden. Dies würde sicherstellen, dass Buchstabendreher automatisch korrigiert werden.

Auf unbekanntem Nutzereingaben reagiert der Chatbot derzeit abweisend, da diese nicht in seinem Trainingsdatensatz enthalten sind. Ist die Eingabe der Plattform nicht bekannt, verweist diese auf das Umformulieren der Frage oder unzutreffenden Antworten. Dies könnte der Chatbot durch die Implementierung einer Intent-Erkennung verbessern, die es dem System ermöglicht, unbekanntem Anfragen zu erkennen und angemessen zu reagieren. Dies

könnte durch die Integration eines Fallback-Mechanismus oder einer Fehlerbehandlungsfunktion erreicht werden.

Ein weiteres Problem bildet die Misskommunikation mit dem System ab. Der Chatbot hat Schwierigkeiten, Anfragen zu interpretieren, die mehrdeutig oder missverständlich sind. Dies führt zu ungenauen oder fehlerhaften Antworten, da der Chatbot möglicherweise nicht die Absicht des Benutzers erkennt oder falsche Annahmen trifft.

Aufgrund der Datenbasis und der jungen Implementierung der KI-Variante hat der Chatbot Schwierigkeiten, Sarkasmus oder Ironie in den Nutzereingaben zu erkennen und angemessen darauf zu reagieren. Diese Formen der sprachlichen Nuancierung verwirren den Chatbot und führen zu unangemessenen oder unpassenden Antworten.

Derzeit bietet der Chatbot keine Fähigkeiten der kontextbezogenen Konversation an, um darauf basierend angemessene Antworten an den Nutzer zurückzugeben. Dies schränkt die Möglichkeiten der Antwortgenerierung erheblich ein, da sich dadurch Ausgaben nur auf die letzte Anfrage beziehen. Angemessener wäre es, dass sich das System auf den Kontext früherer Fragen oder Aussagen bezieht, die der Chatbot möglicherweise nicht ausreichend berücksichtigt oder erfasst hat.

Letztlich *fühlt* sich der Chatbot mit Fachsprache oder Jargon konfrontiert. Hierbei weist er Schwierigkeiten in dem Verständnis nicht-trainierter Fachsprache auf. Dies kann zu Verwirrung oder fehlerhaften Antworten führen, da die Datenbasis bzw. der Chatbot nicht über das erforderliche Wissen in spezifischen Domänen verfügt.

Das Training von Chatbots basierend auf neuronalen Netzen erfordert eine solide und umfangreiche Datengrundlage, um genaue Ergebnisse zu erzielen. Je größer die Datenmenge ist, desto besser kann das Modell Muster und Zusammenhänge erkennen und somit präzise Vorhersagen treffen.

Wichtig ist zu betonen, dass Chatbots im Laufe der Zeit und mit der kontinuierlichen Erweiterung der Datengrundlage besser werden können. Durch das ständige Training und die Verarbeitung von Benutzerinteraktionen kann das Modell seine Fähigkeit zur Beantwortung von Fragen und zur Erbringung von Dienstleistungen verbessern. Daher ist es sinnvoll, die Datenbank des Chatbots für eine bessere Leistung kontinuierlich zu erweitern und das Modell regelmäßig zu trainieren.

Das Aufnehmen von Informationen für eine mögliche Erweiterung der Datenbank mit neuem und verwertbarem Inhalt für das Training des Modells kann wie folgt aussehen:

Der Chatbot kann so konzipiert werden, dass er Interaktionen und Feedback von Benutzern erfasst. Durch die

Analyse dieser Interaktionen können neue Fragen, Muster oder Themen identifiziert werden, die der Chatbot bisher nicht abdeckt. Diese neuen Informationen können genutzt werden, um die Datenbank des Chatbots zu erweitern und ihn auf häufig auftretende Nutzeranfragen besser vorzubereiten. Dies kann in Form von Active Learning stattfinden. Mit der Implementierung von Active Learning kann gezielt nach Rückmeldungen von Benutzern gefragt werden, um sein Verständnis und seine Antworten zu verbessern. Durch die Integration von Mechanismen des aktiven Lernens kann der Chatbot seine Fähigkeiten kontinuierlich weiterentwickeln und sich an die spezifischen Bedürfnisse der Benutzer anpassen.

Durch die Analyse von Chatprotokollen und statistischen Auswertungen der Nutzerinteraktionen können Muster, häufig gestellte Fragen oder Problembereiche identifiziert werden. Diese Erkenntnisse können genutzt werden, um die Datenbank des Chatbots gezielt zu erweitern und das Training zu verbessern, um genau auf die Bedürfnisse der Benutzer einzugehen.

Die Integration von Multimodalität ermöglicht dem Chatbot die Verarbeitung und Interpretation von mehreren Modalitäten wie Text, Bildern, Sprache oder Videos. Durch die Einbeziehung zusätzlicher Modalitäten kann der Chatbot ein tieferes Verständnis der Nutzeranfrage erlangen und präzisere Antworten liefern.

Das Lernverhalten der Chatbot-Komponente ist abhängig von der eingesetzten Technologie und den verwendeten Lernalgorithmen. Im Allgemeinen wird der Chatbot durch das maschinelle Lernen trainiert, um aus den vorhandenen Daten zu lernen und seine Antworten entsprechend anzupassen. Mit einem Blick auf die Laufzeit des Trainingsprozesses kann festgehalten werden, dass sich die Ergebnisse in Abhängigkeit des Modellaufbaus variieren. In der Regel erfordert das Training eines Sprachmodells wie German Semantic eine beträchtliche Rechenleistung und kann mehrere Stunden oder sogar Tage und in unserem Konstrukt über eine Viertelstunde in Anspruch nehmen.

Ähnlich handhabt es das spaCy-Modell, welches ebenfalls hohe Rechenleistung erfordert und je nach Umfang des Trainingsdatensatzes und der spezifischen Konfiguration des Modells variiert.

Die Laufzeit des Trainingsprozesses für die KI-Version des Chatbots hängt von der Architektur des neuronalen Netzwerks, wie bei den anderen Modellen von der Größe des Trainingsdatensatzes und den verwendeten Trainingsstrategien ab. Das Training des KI-Modells kann dadurch in unserem Konstrukt bis zu 20-30 Minuten in Anspruch nehmen, insbesondere wenn große Datensätze und komplexe Modelle verwendet werden. Das Antwortzeitverhalten der KI-Version hängt von der Größe und Komplexität des Modells sowie von der Rechenleistung des Systems ab. Dieses erweist sich jedoch verlässlich

und bringt mit einer geringen Latenz die Antwort an die Oberfläche des Chatbots.

Um das Lernverhalten des Modells positiv zu beeinflussen, können nachfolgende Maßnahmen ergriffen werden:

- Qualität des Trainingsdatensatzes: Ein qualitativ hochwertiger und vielfältiger Trainingsdatensatz ist entscheidend, um eine gute Lerngrundlage für den Chatbot zu schaffen.

- Kontinuierliche Aktualisierung der Datenbank: Durch regelmäßige Updates und Erweiterungen der Datenbank kann der Chatbot kontinuierlich neue Informationen und Muster aufnehmen.

- Berücksichtigung von Benutzerfeedback: Rückmeldungen und Bewertungen von Benutzern sind wertvolle Informationen, um das Lernverhalten des Chatbots zu verbessern.

- Anpassung der Modellarchitektur: Die Wahl der richtigen Modellarchitektur kann einen großen Einfluss auf das Lernverhalten haben. Es kann erforderlich sein, verschiedene Architekturen auszuprobieren und zu vergleichen, um diejenige zu finden, die am besten zu den Anforderungen des Chatbots passen.

- Erweiterte NLP-Techniken: Durch die Integration fortgeschrittenerer NLP-Techniken wie Named Entity Recognition (NER), Sentiment-Analysen oder Kontextverständnis kann der Chatbot ein tieferes Verständnis der Nutzeranfragen entwickeln und präzisere Antworten liefern.

Im Rahmen der Anfragenverwertung haben das German Semantic und spaCy-Modell ähnliche Verhaltensmuster. Ihnen fällt es leicht Fragen zu beantworten, die eine große Ähnlichkeit zum Trainingsdatensatz aufweisen. Das Beantworten von verschachtelten Fragen fällt wiederum dem KI-Modell einfacher. Eine Herausforderung, welche sich alle Modelle teilen, liegt in der Beantwortung von größeren und dadurch komplexeren Anfragen. Hierbei kann sich die Antwortzeit auf mehrere Sekunden erhöhen. Zusammenfassend kann gesagt werden, dass alle Modelle ähnliche Antwort-Zeit-Verhalten nachweisen.

## VIII. FAZIT UND AUSBLICK

Zusammenfassend lässt sich sagen, dass das Modellkonzept hinter dem Chatbot „Winfy“ vielversprechend ist. Die Verwendung von noch mehr Daten und ein regelmäßiges Training werden dazu beitragen, dass der Chatbot noch präzisere und vollständigere Antworten liefert. Insgesamt ist dieses Hochschulprojekt jedoch ein gutes Beispiel dafür, wie Deep Learning verwendet werden kann, um ein nützliches Software-Werkzeug für den täglichen Gebrauch zu erstellen.



Ein vielversprechender Ansatz zur Verbesserung der Leistung des Chatbots wäre, ihn durch die Integration von Reinforcement Learning dazu zu befähigen, im Laufe der Interaktion mit Benutzern kontinuierlich zu lernen und seine Leistung zu verbessern (Session-Konzept). Dabei soll der Nutzer auf gegebene Antworten eine Bewertung geben, die im Nachhinein vom Administrator oder dem Chatbot selbst ausgewertet werden und dadurch die Usability und Performance verbessern [23]. Eine Herausforderung, der sich dabei besonders zu widmen sein wird, ist, dass der KI-Bot von Nutzern zu einem falschen Antwortverhalten verleitet werden könnte.

Darüber hinaus könnten neue Funktionen, wie die Integration von Sprach- oder Bilderkennung, entwickelt werden, um dem Chatbot ein noch breiteres Anwendungsspektrum zu ermöglichen. Hierbei sind Sprachverarbeitungstechnologien (Speech-to-Text) zu erforschen, um dem Nutzer die Möglichkeit zu geben, mit dem Chatbot mündlich zu interagieren, anstatt nur textbasiert zu kommunizieren [24].

Eine andere Möglichkeit wäre die Verwendung von fortgeschrittenen Deep-Learning-Techniken wie beispielsweise Transformer-Netzwerken, die in der Lage sind, lange und komplexe Sätze besser zu verstehen und zu interpretieren. Die Einbettung neuer Sprachmodelle wie GPT-4 oder BERT können dazu beitragen, dass Konversationen noch natürlicher und menschenähnlicher wirken. GPT-4 findet bereits innerhalb von ChatGPT sowie in der Suchmaschine Bing in der Text- und Bilderkennung Anwendung [25].

Ein weiterer interessanter Forschungsbereich ist die Integration von KI-gestützten Emotionserkennungs-Tools, um den Chatbot in der Lage zu versetzen, Emotionen des Nutzers zu erkennen und dementsprechend zu reagieren. Dies könnte dazu beitragen, dass der Chatbot auf eine noch persönlichere Art und Weise mit dem Nutzer kommunizieren kann [26].

Insgesamt gibt es eine große Anzahl von Möglichkeiten zur Weiterentwicklung des KI-basierten Chatbots, sowohl in Bezug auf die Datenbank, die verwendeten Technologien als auch die Funktionen des Chatbots selbst. Aktuell wird u.a. eine plattformunabhängige Frontend-Anwendung zur Nutzung von „Winfy“ auf verschiedenen Endgeräten implementiert.

## LITERATUR

- [1] H. Else, „Abstracts written by ChatGPT fool scientists,“ *Nature*, 12 Januar 2023. [Online]. <https://www.nature.com/articles/d41586-023-00056-7>. [Zugriff: 06 Mai 2023].
- [2] B. Thormundsson, „Chatbot market revenue worldwide from 2018 to 2027,“ *The Insight Partners*, April 2019. [Online]. <https://www.statista.com/statistics/1007392/worldwide-chatbot-market-size/>. [Zugriff: 06 Mai 2023].
- [3] Technische Hochschule Mittelhessen, „Leitbild der TH Mittelhessen (THM),“ o.D. [Online]. <https://www.thm.de/site/hochschule/profil/leitbild-2019.html>. [Zugriff: 06 Mai 2023].
- [4] S. B. Sonawane, A. S. Badwar, R. H. Dalvi, G. N. More und S. Talekar, „Design of Chatbot System for Student,“ *International Journal of New Innovations in Engineering and Technology*, p. 6, 2020.
- [5] L. H. Su, T. Dang-Huy, T. Thi-Yen-Linh, N. Thi-Duyen-Ngoc, L. Bao-Tuyen und N. Ha-Phuong-Truc, „Development of an AI Chatbot to Support Admissions and Career Guidance for Universities,“ *International Journal of Emerging Multidisciplinary Research*, Nr. 4, p. 9, 2020.
- [6] A. Dietrich, „Chatbots – Wie müssen sie gestaltet sein, um den Anforderungen zu genügen?,“ Hochschule Luzern, 29. April 2019. [Online]. <https://hub.hslu.ch/retailbanking/chatbots-wie-muessen-sie-gestaltet-sein-um-den-anforderungen-zu-genuegen/>. [Zugriff: 30 April 2023].
- [7] Y. Altmann, „8 Funktionen, die jeder Chatbot haben sollte,“ Userlike UG, 21 Januar 2021. [Online]. <https://www.userlike.com/de/blog/chatbot-funktionen>. [Zugriff: 26 April 2023].
- [8] N. Inerle, T. Ochs und L. Offermanns, „Entwicklung eines FAQ-Chatbots zur Beratung im Hochschulbereich,“ Abschlussdokumentation des Master-Entwicklungsprojekts Wirtschaftsinformatik WS21/22 (intern), THM, Friedberg, März 2022.
- [9] S. Tomar, „Sahajtomar/German-semantic,“ hugging face, o.D. [Online]. <https://huggingface.co/Sahajtomar/German-semantic>. [Zugriff: 30 April 2023].
- [10] ExplosionAI GmbH, „Industrial-Strength Natural Language Processing,“ ExplosionAI GmbH, o.D. [Online]. <https://spacy.io/>. [Zugriff: 27 April 2023].
- [11] Pallets, „Flask,“ palletsprojects, o.D. [Online]. <https://flask.palletsprojects.com/en/2.3.x/>. [Zugriff: 27 April 2023].
- [12] I. Sysoev, „NGINX,“ F5 Inc., o.D. [Online]. <https://www.nginx.com/>. [Zugriff: 30 April 2023].
- [13] B. Chesneau, „Gunicorn,“ o.D. [Online]. <https://gunicorn.org/>. [Zugriff: 30 April 2023].
- [14] Google LLC, „Angular,“ Google LLC, o.D. [Online]. <https://angular.io/>. [Zugriff: 27 April 2023].
- [15] I. W. Advertising, „The ultimate guide to machine-learning chatbots and conversational AI,“ 28 September 2022. [Online]. <https://www.ibm.com/watson-advertising/thought-leadership/machine-learning-chatbot>. [Zugriff: 28 April 2023].
- [16] E. Alpaydm, *Introduction to Machine Learning*, Cambridge, Massachusetts: The MIT Press, 2004.
- [17] S. Bhattacharyya, V. Snasel, A. E. Hassanien, S. Saha und B. K. Tripathy, *Deep Learning: Research and Applications*, Berlin: De Gruyter, 2020.

- [18] L. Wuttke, „Künstliche Neuronale Netzwerke: Definition, Einführung, Arten und Funktion,“ datasolut GmbH, o.D. [Online]. <https://datasolut.com/neuronale-netzwerke-einfuehrung/>. [Zugriff: 30 April 2023].
- [19] S. Bird, E. Klein und E. Loper, *Natural Language Processing with Python*, Sebastopol, Kalifornien: O'Reilly Media, 2009.
- [20] J. Brownlee, „A Gentle Introduction to the Bag-of-Words Model,“ *Machine Learning Mastery*, 09 Oktober 2017. [Online]. <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>. [Zugriff: 29 April 2023].
- [21] T.-A. Hua, S. Kirchner, S. Sahba und D. Tansel, „KI-basierter FAQ-Chatbot für eine Hochschule im Bereich Prüfungsangelegenheiten,“ Abschlussdokumentation des Master-Entwicklungsprojekts Wirtschaftsinformatik SoSe22 (intern), THM, Friedberg, Oktober 2022.
- [22] Simplilearn, „What is PyTorch, and How Does It Work: All You Need to Know,“ simplilearn, 31 März 2023. [Online]. <https://www.simplilearn.com/what-is-pytorch-article>. [Zugriff: 30 April 2023].
- [23] D. Biswas, „Self-improving Chatbots based on Deep Reinforcement Learning,“ *towardsdatascience*, 14 September 2020. [Online]. <https://towardsdatascience.com/self-improving-chatbots-based-on-reinforcement-learning-75cca62debce>. [Zugriff: 02 Mai 2023].
- [24] A. Singh, P. Murugeswari, S. D. P. Ragavendiran, A. Kaur, G. Singh und S. Margabandu, „AI-based Chatbot for Physically Challenged People,“ in *International Conference on Edge Computing and Applications (ICECAA)*, Tamilnadu, India, 2022.
- [25] R. Spitko, „Was ist GPT-4? – Fähigkeiten, Limitierungen, Grenzen, Einstazmöglichkeiten, Kosten, Nutzen, API und vieles mehr,“ *onlinemarketing-mastermind*, 16 März 2023. [Online]. <https://onlinemarketing-mastermind.de/ki/gpt-4/>. [Zugriff: 03 Mai 2023].
- [26] M. Karna, D. S. Juliet und R. Joy, „Deep learning based Text Emotion Recognition for Chatbot applications,“ in *4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184)*, Tirunelveli, India, 2020.

## Anhang

**THM | CHATBOT WINFY**  
TECHNISCHE HOCHSCHULE MITTELHESSEN

**Herzlich willkommen!**

Ich bin Winfy, der Chatbot der Prüfungsausschüsse B.Sc. u. M.Sc. Wirtschaftsinformatik der beiden gemeinsamen MND/MNH-Studiengänge an der Technischen Hochschule Mittelhessen.  
Über ein Feedback freue ich mich!

**Impressum:**  
Vorsitzender Prüfungsausschuss Wirtschaftsinformatik (B.Sc. & M.Sc.)  
Prof. Dr. Harald Riz  
@pa\_wk@mnd.thm.de  
☎ +49/6031/604-7620  
📍 Technische Hochschule Mittelhessen  
Campus Friedberg  
Fachbereich MND  
Wilhelm-Leuschner-Str. 13  
D-61169 Friedberg  
Büro 82.0.02

**Prüfungsausschuss WK** | **Fachbereich MND**  
**THM ZPA** | **Feedback**

Disclaimer: Alle Angaben ohne Gewähr  
Für die Inhalte und Richtigkeit der Informationen ist stets der jeweilige Informationsanbieter der verlinkten Webseite verantwortlich. Zum Zeitpunkt der Verlinkung waren keine Rechtsverstöße erkennbar. Zugleich wird darauf hingewiesen, dass aktuelle und verbindliche Auskünfte lediglich dem Webauftritt des Zentralen Prüfungsausschusses der THM zu entnehmen sind.  
Bitte nutze hierzu den Button 'THM ZPA'.

**Hey!**  
Mein Name ist Winfy, dein virtueller Assistent rund um die Wirtschaftsinformatik.  
Hast du eine Frage zu Prüfungsangelegenheiten zum **Bachelor-** oder **Masterstudium**?

Hast du eine Frage zum **Bachelor-** oder **Masterstudium** Wirtschaftsinformatik?  
Wähle bitte zunächst das gewünschte Studium.

Anschließend darfst du mich gerne in deiner Muttersprache begrüßen mit: Hallo  
... wenn Du als erste Nachricht an mich ein "Hallo" eingibst, erfährst du, bei welchen Fragen ich dich unterstützen kann.

**Nachricht** Bitte hier die Nachricht eingeben... nutze am besten die Autovervollständigung!

Abbildung 1: Landingpage – Startbildschirm

**Hey!**  
Mein Name ist Winfy, dein virtueller Assistent rund um die Wirtschaftsinformatik.  
Hast du eine Frage zu Prüfungsangelegenheiten zum **Bachelor-** oder **Masterstudium**?

Hast du eine Frage zum **Bachelor-** oder **Masterstudium** Wirtschaftsinformatik?  
Wähle bitte zunächst das gewünschte Studium.

Anschließend darfst du mich gerne in deiner Muttersprache begrüßen mit: Hallo  
... wenn Du als erste Nachricht an mich ein "Hallo" eingibst, erfährst du, bei welchen Fragen ich dich unterstützen kann.

Okay, du hast eine Frage zum Masterstudium. Wie kann ich dir helfen?

Hallo!

Ein freundliches "Hallo" von mir zurück!

Mein Name ist Winfy, dein virtueller Assistent rund um die Wirtschaftsinformatik im Prüfungskontext. Zwar bin ich noch ziemlich neu an der THM, doch fühle ich mich prima aufgenommen. Mir gefällt es super!

Ich freue mich, dich begrüßen zu dürfen und dir deine Fragen zu Prüfungsangelegenheiten bezüglich des Masterstudiums Wirtschaftsinformatik beantworten zu können. Insbesondere zu den von dir häufig gestellten Fragen (FAQ) aus den folgenden Bereichen findest Du bei mir nützliche Hinweise:

- Allgemeine Bestimmungen für Masterprüfungsordnungen
- Ansprechpartner/innen (Dekanat, Sekretariat, Gremien, Fachschaft)
- Auslandsaufenthalt (Auslandssemester, -praktikum)
- Bewerbung / Immatrikulation / Rückmeldung / Studienabschluss / Exmatrikulation
- E-Campus
- Leistungsanerkennung (anderer Studiengang, andere Hochschule / Ausland)
- Masterarbeit (Zulassung, Ausgabe, externer Korreferent, Abgabe, Bewertung)
- Modulhandbuch (Pflicht- und Wahlpflichtmodule)
- Prüfungen / Klausuren / KoV (An-, Abmeldung, Attest, Bonuspunkte, Fehlversuche, Plagiat, Täuschung, Termine, Wiederholung von Leistungen, Zweitprüfer)
- Prüfungen in anderen Studiengängen / Fachbereichen
- Prüfungsausschuss (Mitglieder, Aufgaben, Anträge an den Ausschuss)
- Prüfungsordnung (fachspezifisch)
- Schwerpunktwahl im Masterstudiengang / Wahlpflichtmodule – Dozenten / Semesterzuordnung / Vertiefungsrichtung

**Nachricht** Bitte hier die Nachricht eingeben... nutze am besten die Autovervollständigung!

Abbildung 2: Landingpage – Dialogfenster

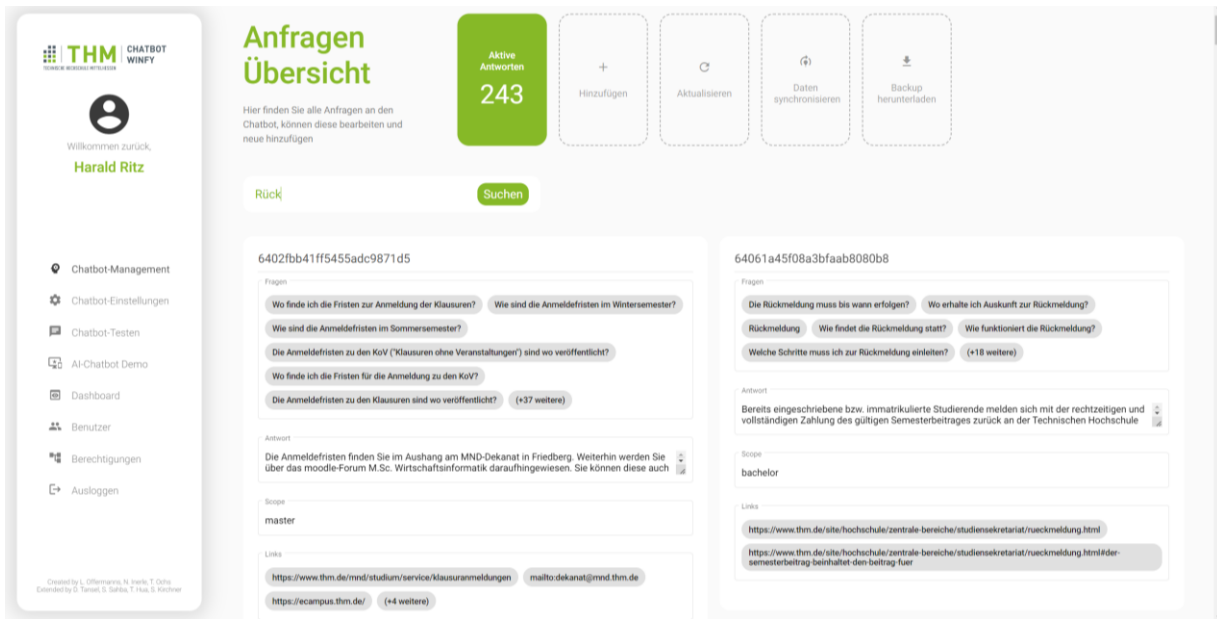


Abbildung 4: Backend-Oberfläche

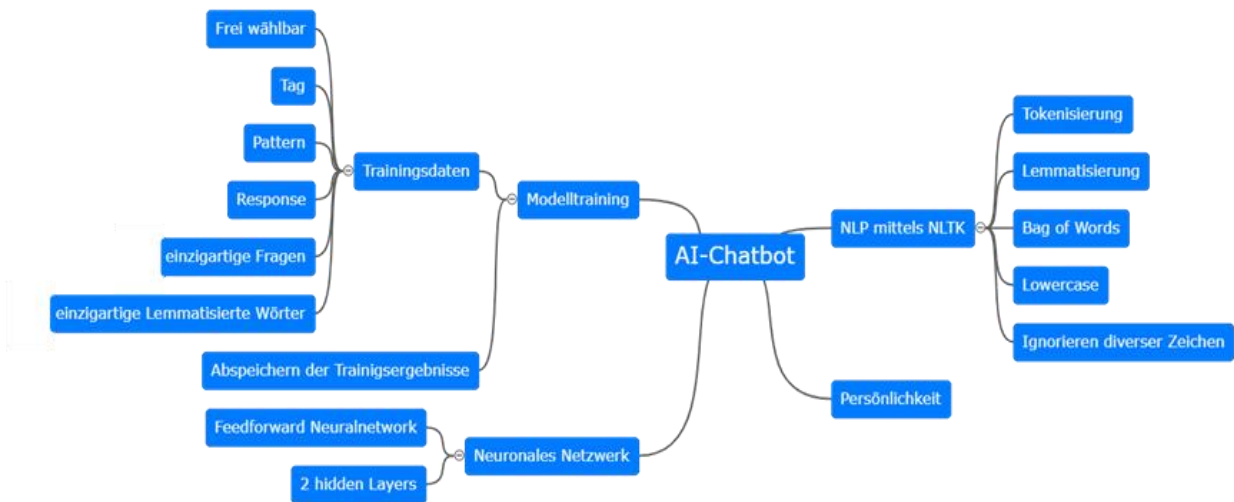


Abbildung 7: KI-Chatbot-Architektur