Hochschule Offenburg
offenburg.university

Fakultät für Medien

Studiengang Medien- und Informationswesen, 7. Semester

Jana Wiegert
Hochschule Offenburg
Badstraße 24
77652 Offenburg

Bachelorthesis

Konzeption und prototypische Umsetzung eines kontextsensitiven Recommender Systems am Beispiel eines Chatbots aus dem Gesundheitswesen

---

Sommersemester 2022
Abgabedatum: 29. Juli 2022

Professor: Volker Sänger
Betreuer: Tobias Ostertag
Firma: CAS Software AG

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Quellen und Hilfsmittel angefertigt habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegen.

Karlsruhe, den 29. Juli 2022

Kurzfassung

In den letzten Jahren haben Recommender Systeme zunehmend an Bedeutung gewonnen. Diese Systeme sind meist für Bereiche des E-Commerce konzipiert und berücksichtigen oftmals nicht den aktuellen Kontext der nutzenden Person. Recommender Systeme können allerdings nicht nur im E-Commerce zum Einsatz kommen, sondern finden ihren Anwendungszweck auch im Gesundheitswesen. Ziel dieser Bachelorarbeit ist es, ein Recommender System zu entwickeln, das den aktuellen Kontext der nutzenden Person (Chatverlauf, demografische Daten) besser berücksichtigen kann. Dazu befasst sich diese Arbeit mit der Konzeption und prototypischen Umsetzung eines kontextsensitiven Recommender Systems für einen bereits existierenden Chatbot aus dem Gesundheitswesen. Das in dieser Arbeit konzipierte und entwickelte Recommender System soll Mitarbeitende aus dem Gesundheits- und Sozialwesen entlasten und ihnen hilfreiche sowie thematisch sinnvolle Informationen zur Verfügung stellen. Basierend auf festgelegten Anforderungen wurde ein Konzept für das Recommender System entwickelt und zu Teilen als Prototyp umgesetzt. Abschließend wurde der Prototyp im Hinblick auf die Anforderungen evaluiert. Zudem fand eine technische Evaluation und eine Evaluation mithilfe von Anwendenden statt, welche den implementierten Prototypen bereits existierenden Systemen gegenüberstellte. Die von dem Prototyp empfohlenen Textausschnitte erzielten in der Evaluation mit nutzenden Personen eine thematisch signifikant höhere Übereinstimmung mit den Chatdaten.

Schlüsselwörter: Chatbot, Recommender System, Gesundheitswesen, Kontextsensitivität, Natural Language Processing, BERT, Zero-Shot-Klassifikation, Topic Modeling, Cold-Start-Problem

Abstract

In recent years, recommender systems have become increasingly important. These systems are predominantly designed for e-commerce and often do not consider the current context of the person using them. However, recommender systems can be applied not only in e-commerce but also in healthcare. The intention of this bachelor thesis is to develop a recommender system that can improve to consider the current context of users (chat history, demographic data). Therefore this thesis approach the conception and prototypical implementation of a context-sensitive recommender system for an already existing healthcare chatbot. The recommender system designed and developed in this thesis is expected to relieve employees from the health and social care sector and to provide them with helpful as well as thematically meaningful information. Based on defined requirements, a concept for the recommender system is developed and partially implemented as a prototype. Finally, the prototype was evaluated concerning the requirements. In addition, a technical evaluation and a user evaluation were conducted, which compared the implemented prototype with already existing systems. The text excerpts recommended by the prototype achieved a thematically significantly higher agreement with the chat data in the evaluation with users.

Keywords: Chatbot, Recommender System, Healthcare, context sensitivity, Natural Language Processing, BERT, Zero-Shot-Classification, Topic Modeling, Cold-Start-Problem

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

API	Programmierschnittstelle
BERT	Bidirectional Encoder Representations from Transformers
BGW	Berufsgenossenschaft für Gesundheitsdienst und Wohlfahrtspflege
CA	Conversational Agent
DSR	Design Science Research
GBERT	German-Bert
HRS	Health Recommender System
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
NLTK	Natural Language Toolkit
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
SBERT	Sentence-BERT
TF-IDF	Term Frequency Inverse Document Frequency
UI	User Interface
UX	User Experience

1 Einleitung

Fachkräfte in Gesundheits- und Sozialwesen kämpfen stetig mit neuen Herausforderungen, nicht zuletzt mit der COVID-19-Pandemie. Wo schon zuvor ein enormer Fachkräftemangel bestand, macht sich dieser durch die momentane Situation noch signifikanter bemerkbar. Mehrere Studien sind sich einig, dass dieser Zustand zu erhöhten psychischen Folgen der Mitarbeitenden führt. [1] Um die Mitarbeitenden der Sozialwirtschaft zu entlasten, ist es möglich, ihnen den Arbeitsalltag mithilfe digitaler Technologien zu erleichtern. Hierfür können KI-gestützte Anwendungen in Betracht gezogen werden.

1.1 Problemstellung und Motivation

In den letzten Jahren wurden im Gesundheitswesen zunehmend Anwendungen mit Künstliche Intelligenzen (KIs) wie beispielsweise Chatbots eingesetzt und es wurde in deren Forschung investiert [2]. Bei einem richtigen Einsatz bieten diese die Möglichkeit, sowohl Zeit als auch Kosten zu sparen [3]. Im Gesundheitswesen können Chatbots eingesetzt werden, um Erkrankte oder Mitarbeitende zu unterstützen, zu beraten und zu informieren. Da Chatbots im Allgemeinen meist eher in kürzeren Sätzen antworten und lediglich die Frage der anwendenden Person beantworten sollen, hat die Person keine Möglichkeit, schnell und einfach an für sie hilfreiche weiterführende Informationen zu gelangen. Um der Person diese Informationen zu liefern, können Chatbots aus dem Gesundheitswesen um ein kontextsensitives Recommender System ergänzt werden. Recommender Systeme werden meist im E-Commerce genutzt, um nutzenden Personen weitere Artikel zu empfehlen. [4] Aber auch in Chatbots werden sie zunehmend eingesetzt, um den Nutzenden Inhalte oder Produkte zu empfehlen [5]. In der Gesundheitsbranche können Chatbots Recommender Systeme nutzen, um der anwendenden Person gesundheitsbezogene Inhalte wie Artikel oder Videos zu empfehlen. Diese Inhalte können für jede nutzende Person auf der Grundlage ihrer bisherigen Interaktionen mit dem Chatbot personalisiert werden. Recommender Systeme können Chatbots dabei helfen, der nutzenden Person relevantere und individuellere Inhalte zu bieten, was zu einer besseren Bindung und Zufriedenheit führen kann.

In dieser Bachelorarbeit wird an diesem Thema weitergeforscht, indem ein solches kontextsensitives Recommender System für einen Chatbot im Gesundheitsbereich konzipiert und prototypisch implementiert werden soll.

1.2 Zielsetzung

Hauptziel dieser Arbeit ist die Modellierung eines kontextsensitiven Recommender Systems, welches prototypisch in die vorhandene Systemumgebung des Chatbots „Impuls“ integriert werden soll. Dadurch soll der Chatbot in der Lage sein, kontextsensitive Empfehlungen auszusprechen, indem er Metadaten wie beispielsweise demografische Daten miteinbezieht. Er soll der anwendenden Person daraufhin passende Dokumente und Links zu ihrer Anfrage liefern. In der Arbeit sollen zudem die folgenden vier Forschungsfragen beantwortet werden.

Forschungsfrage 1: *Welche Arten von Recommender Systemen gibt es und welche Vorteile bringen diese jeweils mit sich?*
Um diese Frage zu beantworten, sollte eine umfassende Literaturanalyse stattfinden, in welcher Arten von Recommender Systemen analysiert und evaluiert werden. Hierzu wird angemerkt, dass die Entscheidung der Wahl des Recommender Systems, trotz einer allgemeinen Auflistung der Vor- und Nachteile, für jedes System individuell getroffen werden sollte, um bestmögliche Ergebnisse zu erzielen.

Forschungsfrage 2: *Wie sollte ein Recommender System im Gesundheitswesen mit sensiblen Daten umgehen?*
Die zweite Forschungsfrage wird zu Beginn der Konzeption aufgegriffen, indem zunächst die Herausforderungen eines Recommender Systems mit sensiblen Daten behandelt werden. Der Umgang mit sensiblen Daten ist ein sehr herausforderndes Thema, welches in der vorliegenden Arbeit nicht vollumfänglich geklärt werden kann. Trotz dessen soll ein Ausblick darauf geboten werden, da es vor allem im Gesundheitswesen von besonderer Relevanz ist.

Forschungsfrage 3: *Welche Möglichkeiten gibt es zur Implementierung eines kontextsensitiven Recommender Systems?*
Mit dieser Forschungsfrage wird sich zunächst in der Konzeption genauer befasst, dabei werden verschiedene Use Cases aufgestellt, welche unterschiedliche Kontextdaten zur Verfügung haben und mit diesen demnach auch unterschiedlich umgehen müssen. In der Implementierung wird eine Möglichkeit zur Umsetzung aufgezeigt, mit welcher der Prototyp geschaffen wurde.

Forschungsfrage 4: *Welche Kontextdaten erweisen sich am nützlichsten für die Verwendung von kontextabhängigen Empfehlungen und wie kann dies am besten umsetzt werden?*
Diese Forschungsfrage ist die zentrale Frage dieser Arbeit, welche über alle Kapitel hinweg beantwortet werden soll. Hierfür muss entschieden werden, welche Metadaten miteinbezogen werden sollen. Hierfür sollte der Kontext abgegrenzt werden und nach jeweiligem Use Case entschieden werden, welcher Bezugsrahmen sinnvoll für die Empfehlung sein könnte. Trotz dieser Begrenzung sollte das System so gebaut werden, dass es um neue Kontextdaten erweiterbar ist. Vor allem der Konversationsverlauf wird genauer als Kontextinformation betrachtet. Dabei wird untersucht, was bereits gesagt wurde und welche Informationen sich daraus ableiten lassen.

1.3 Kontext der Arbeit

Die vorliegende Arbeit wurde in Kooperation mit der CAS Software AG verfasst. Hierbei wurde sie im Rahmen des Projekts „Pulsnetz“ ausgeschrieben. „Pulsnetz“ hat sich zum Ziel gesetzt, verschiedene Einrichtungen aus dem Sozial- und Gesundheitswesen mit Angeboten und Projekten zu den Themen Arbeits- und Gesundheitsschutz zu unterstützen. [6] In deren KI-Garage wurde der Chatbot „Impuls“ konzipiert und implementiert, welcher stetig erweitert und optimiert wird. „Impuls“ ist noch im Lernprozess, kann aber schon nach Informationen auf der Homepage von „Pulsnetz“ und in zur Verfügung gestellten PDF-Dokumenten suchen, sowie Kontakte zu verschiedenen Themen ausfindig machen. Die Spracheingabe und -ausgabe des Chatbots erfolgt schriftlich. Der Chatbot soll durch ein kontextsensitives Recommender System erweitert werden. Dadurch soll der nutzenden Person die Möglichkeit geboten werden, sich mit Informationen zu beschäftigen, welche zu ihren momentanen Kontextdaten passen.

2 Methodik und Aufbau der Arbeit

Um eine wissenschaftliche Arbeit zu verfassen und zu strukturieren, wird ein Rahmen benötigt, in welchem die Forschung durchgeführt werden kann. Deshalb wird die vorliegende Arbeit nach dem Design Science Research (DSR) Ansatz durchgeführt.

2.1 Design Science Research Methodik

Die DSR Methodik ist ein gängiger Ansatz im Bereich der Softwareentwicklung auf wissenschaftlicher Basis. [7] Ziel dieser Methodik ist es, die Grenzen von menschlichen und organisatorischen Fähigkeiten zu erweitern, indem neue und innovative Artefakte geschaffen werden, um eine kontinuierliche Anpassung des vorliegenden Problems zu garantieren. [8] Artefakte stehen in diesem Zusammenhang für ein Objekt, welches Wissen enthält, wodurch neue wissenschaftliche Erkenntnisse erlangt werden können. Dieses Wissen umfasst die Designlogik, die Konstruktionsmethoden, das Werkzeug und den Kontext, in dem das Artefakt funktionieren soll. Beispiele für Artefakte aus dem Bereich der Softwareentwicklung sind Algorithmen oder Prototypen. Design beschreibt im Kontext der DSR Methodik die Modifikation an einem bestehenden System, mit dem Ziel, dadurch eine Verbesserung zu erreichen. [9, 10] Research wird dabei als eine Tätigkeit beschrieben, welche zum Verständnis eines Phänomens beiträgt. In diesem Kontext kann das Phänomen als Bereich, indem allgemeines oder persönliches Interesse besteht, definiert werden. [10]

2.2 DSR Modell nach Kuechler und Vaishnavi

Inzwischen gibt es unzählige Modelle des DSR, welche sich grundlegend unterscheiden. Die vorliegende Arbeit konzentriert sich auf das Modell von Kuechler und Vaishnavi. Wesentlich für dieses Modell ist der Forschungszyklus, welcher fünf Schritte beinhaltet. Diese Schritte werden in Abbildung 2.1 genauer dargestellt.

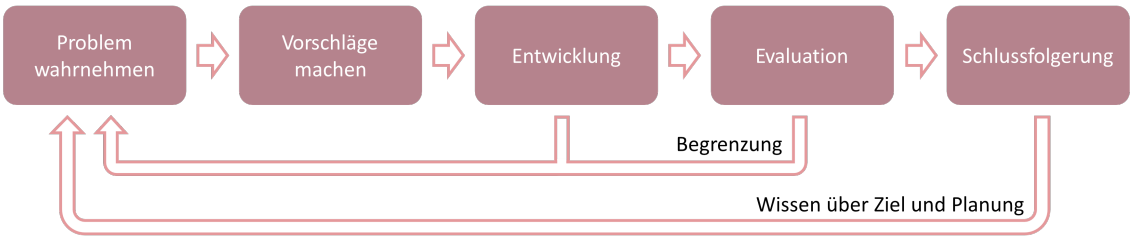


Abbildung 2.1: Prozess des DSR-Modells nach Kuechler und Vaishnavi nach [11]

Es lässt sich an den Pfeilen erkennen, dass der Prozess einen zyklischen Aufbau darstellt, welcher immer wieder zum Verständnis des Problems zurückspringen kann. Der Forschungszyklus wird dann in mehreren Iterationen durchlaufen, bis das Artefakt so weit vorangeschritten ist, dass es als endgültige Version vorliegt. [7] Im Folgenden wird genauer auf die einzelnen Phasen des Prozesses nach Kuechler und Vaishnavi eingegangen.

Problem wahrnehmen und verstehen Der erste Schritt ist die Wahrnehmung und das Verständnis der Problematik. Um sich einen Überblick zu verschaffen, kann eine Literaturanalyse in dem entsprechenden Wissensgebiet und dessen aktuellen Forschungen durchgeführt werden. Zudem ist es möglich, das nötige Wissen über Interviews und Gespräche mit Experten und Expertinnen zu erlangen. Das Ergebnis dieser Phase vermittelt eine ungefähre Vorstellung der bereits vorhandenen Forschung. [11]

Vorschläge machen Nachdem eine Wissensbasis geschaffen und das Problem identifiziert wurde, können erste Vorschläge eingebracht werden, um ein Konzept zu entwickeln. Hierfür wird meist eine weitergehende und spezifischere Literaturanalyse durchgeführt. Die Ergebnisse dieser Phase sollten die Anforderungen für das Artefakt, welches im Folgenden erschaffen wird, enthalten. [7]

Entwicklung Auf Basis des zuvor entwickelten Konzeptes kann das Artefakt in diesem Schritt des Prozesses entwickelt werden. Dafür muss ein klar definiertes Problem, eine solide Wissensbasis und genaue Anforderungen an das Artefakt vorliegen. Das Ergebnis dieser Phase stellt dann entweder das Artefakt als Prototypen oder die finale Version des Artefakts dar. In den ersten Iterationen liegt hier meist nur ein Prototyp vor, welcher in den folgenden Iterationen ausgebaut wird. [7]

Evaluation Wenn ein Artefakt, als Prototyp oder finale Version deklariert wurde, muss dieses evaluiert werden. Die Evaluation findet anhand von ausgewählten Kriterien statt. Es wird dabei vor allem auf die Erfüllung der Anforderungen aus der Konzeptionsphase geprüft. Anhand der Ergebnisse der Evaluation kann das Artefakt weiter verbessert oder ausgebaut werden. [11]

Schlussfolgerung Die Schlussfolgerung stellt die letzte Phase des Prozesses dar. Hier werden Ergebnisse nochmals reflektiert. Es wird versucht, das Artefakt nochmals hinsichtlich seines Designs zu hinterfragen. Wenn sich in dieser Phase Schwächen herauskristallisieren, kann eine neue Iteration begonnen werden, welche das neue Bewusstsein in die folgende Iteration mit einbringt. [7]

2.3 Anwendung des Modells auf die vorliegende Arbeit

Aufgrund des begrenzten Zeitraumes wurde ausschließlich eine Iteration des Prozesses durchgeführt. Die Anwendung des Modells auf diese Arbeit wird in Abbildung 2.2 visualisiert. Auf die eigene Anwendung wird im Folgenden Bezug genommen.

Abbildung 2.2: Erste Iteration von dem Prozess des DSR-Modells nach [11]

Literaturanalyse und Befragungen von Experten und Expertinnen Um das Problem zu identifizieren, wird zunächst Rücksprache mit Mitarbeitenden der CAS Software AG gehalten, welche vielfältiges Wissen auf diesem Gebiet mitbringen und in dem Projektteam „Pulsnetz“ mitarbeiten. Um ein Verständnis für das Problem zu entwickeln, wird dann eine umfassende Literaturanalyse zu den Themen Natural Language Processing (NLP), Chatbots und Recommender Systeme durchgeführt. Damit ein Überblick über den momentanen Forschungsstand im Bereich Recommender Systeme bei Chatbots im Gesundheitswesen garantiert werden kann, werden verwandte Arbeiten analysiert und aufgearbeitet.

Konzeption mit einer Anforderungsanalyse Um eine konzeptionelle Architektur des Recommender Systems zu erarbeiten, werden zunächst Personas erstellt. Diese leiten sich aus den vorhandenen Unterlagen des Projekts „Pulsnetz“ ab. Aus diesen Personas werden dann Use Cases entwickelt, mit welchen Anforderungen an das Recommender System definiert werden können. Mithilfe dieser Anforderungen kann die konzeptionelle Architektur des Recommender Systems aufgestellt werden. Anhand dieser können die Grenzen für die prototypische Implementierung festgelegt werden.

Prototypische Implementierung Für die Umsetzung der prototypischen Implementierung werden die konzeptionelle Architektur und die Grenzen der Implementierung aus dem vorherigen Schritt verwendet. Mit diesen kann eine Architektur des Prototyps in der tatsächlichen Umsetzung aufgestellt werden. Diese Architektur wird dann technisch umgesetzt.

Technische Evaluation gestützt durch eine Evaluation der Anwendenden Um zu evaluieren, wie gut das prototypische Recommender System arbeitet, wird eine technische Evaluation durchgeführt. Bei dieser wird der Prototyp mit schon vorhandenen Recommender Systemen anhand einer Metrik, welche mit Labels und zugehörigen Scores arbeitet, verglichen. Um diese Evaluation zu stützen wird zusätzlich eine Umfrage mit Anwendenden durchgeführt, in welcher der Prototyp den schon vorhandenen Systemen gegenübergestellt wird und anhand der Likert-Skala[1] von Testpersonen bewertet wird.

Fazit aus der Evaluation Aus den Ergebnissen der Evaluation werden dann Schlüsse in Bezug auf den Prototyp gezogen. Hierbei werden Erkenntnisse, sowie Grenzen aufgegriffen. Aus den Erkenntnissen und Grenzen kann letztlich ein Ausblick folgen, wie an der Thematik weitergeforscht werden kann.

[1] Die Likert-Skala ist ein psychometrisches Instrument, mit welchem Eigenschaften, Fähigkeiten und Qualitäten quantifiziert werden können. Die Likert-Skala ist eine lineare Skala, bei welcher die nutzende Person eine Auswahl zwischen den Werten eins und fünf treffen kann. [12]

3 Grundlagen

Im folgenden Kapitel werden die Grundlagen erläutert, welche für die Konzeption und die Umsetzung des prototypischen Recommender Systems benötigt werden. Die Grundlagen werden in die drei für die vorliegende Arbeit fundamentalen Themen NLP, Chatbots und Recommender Systeme aufgeteilt.

3.1 Natural Language Processing

Sprache ist für Menschen etwas Selbstverständliches, sie verwenden sie jeden Tag, um mit Mitmenschen zu kommunizieren. Ein Computerprogramm kann mit der natürlichen Sprache nicht arbeiten, weshalb numerische Daten benötigt werden, um Sätze und Wörter interpretieren und verstehen zu können. Da die Hauptaufgabe eines Recommender Systems die Verarbeitung und Interpretation von Textdaten ist, ist NLP der Grundbaustein für den Aufbau und die Funktionsfähigkeit eines kontextsensitiven Recommender Systems. Die Forschung, welche sich derzeit mit der Verarbeitung von Sprache beschäftigt, wird NLP genannt. Im Folgenden wird NLP definiert und abgegrenzt. Nachfolgend werden Methoden des NLP vorgestellt, welche für die vorliegende Arbeit von Relevanz sind.

3.1.1 Definition und Abgrenzung von Natural Language Processing

NLP ist ein Teilbereich der KI und kombiniert diesen mit Teilen der Linguistik. KI befasst sich mit der Programmierung eines Systems, welches daraufhin selbstständige Entscheidungen treffen können soll [13]. NLP ist für die Umwandlung von unstrukturierten Textdaten in numerische oder strukturierte Daten zuständig. [14] Es beinhaltet Natural Language Understanding (NLU) und Natural Language Generation (NLG), welche meist mithilfe von Machine Learning[2] oder Deep Learning[3] durchgeführt werden. NLU ist der Teilbereich von NLP, welcher dafür zuständig ist, Texte zu verstehen und zu interpretieren. Dafür müssen unter anderem Absichten und Ziele aus dem Text herausgelesen werden. [13] NLG hingegen erzeugt natürliche Sprache, welche an die anwendende Person zurückgegeben wird. [16] Deep Learning ist ein Teilbereich des Machine Learning, welches wiederum in den Bereich der KI fällt. Die Abbildung 3.1 verdeutlicht das Zusammenspiel dieser Komponenten.

[2] Machine Learning ist ein Oberbegriff für Methoden der Datenanalysen. Es beruht auf der Idee, dass Systeme aus Daten lernen und Muster in ihnen erkennen können. [13]

[3] Deep Learning beschreibt eine Methode des Machine Learning, welche, mit künstlichen neuronalen Netzen, unbeaufsichtigt aus unstrukturierten Daten lernen kann. [15]

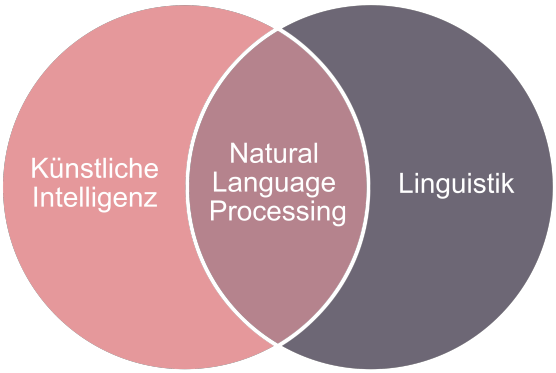


Abbildung 3.1: Einordnung und Abgrenzung von Natural Language Processing nach [14]

3.1.2 Methoden des Natural Language Processing

Ein grundlegender Schritt zur Verarbeitung von Textdaten ist die Normalisierung von Texten, welche Tokenisierung genannt wird [16]. Es handelt sich hierbei um eine syntaktische Methode, welche das Ziel verfolgt, die Eingabe in einzelne Wortteile zu zerlegen. Danach wird versucht, das Wort auf die wesentliche Bedeutung zu reduzieren. [17] Dafür gibt es zwei verschiedene Techniken. Stemming, welches das Wort auf seinen eigentlichen Wortstamm reduziert, wird beispielsweise bei verschiedenen Zeitformen verwendet. Da dies nicht für jedes Wort funktioniert, gibt es eine weitere Technik, welche Lemmatisierung genannt wird. Sie versucht die Wurzel des Wortes wiederzugeben, beispielsweise indem sie Superlative wieder in ihre Grundform bringt. Dies kann mithilfe von Wörterbüchern für das Training realisiert werden. [14]

Die Tokenisierung kann allerdings keine Wörter in Bezug zueinander setzen. Da Wörter aber des Öfteren erst in Sätzen ihre vollständige Bedeutung offenbaren, wird im Folgenden eine Methode vorgestellt, welche Wörter im Zusammenhang betrachten kann. Dies spielt vor allem bei Recommender Systemen in Chatbots eine wichtige Rolle, da der Kontext aus der Zusammensetzung der Wörter erschlossen werden kann. Eine Methode, welche den Kontext zwischen den Wörtern erfassen kann, nennt sich Word Embeddings. Dabei wird das Vokabular mithilfe von Vektoren abgebildet, welche durch flache neuronale Netze erzeugt werden. Jeder Vektor stellt dabei ein Wort dar. Somit kann anhand der Abstände zwischen den Wörtern abgelesen werden, wie ähnlich sich diese sind. [18] Die Vektoren stellen außerdem die Beziehung zwischen unterschiedlichen Themengruppen dar. Die vereinfacht dargestellte Abbildung 3.2 soll dies nochmal verdeutlichen. Hier zeigt sich, dass der Abstand von „König“ zu „Königin“ der gleiche ist, wie von „Mann“ zu „Frau“. Des Weiteren ist der Abstand zwischen „Frau“ und „Königin“ der gleiche, wie zwischen „Mann“ und „König“. [19]

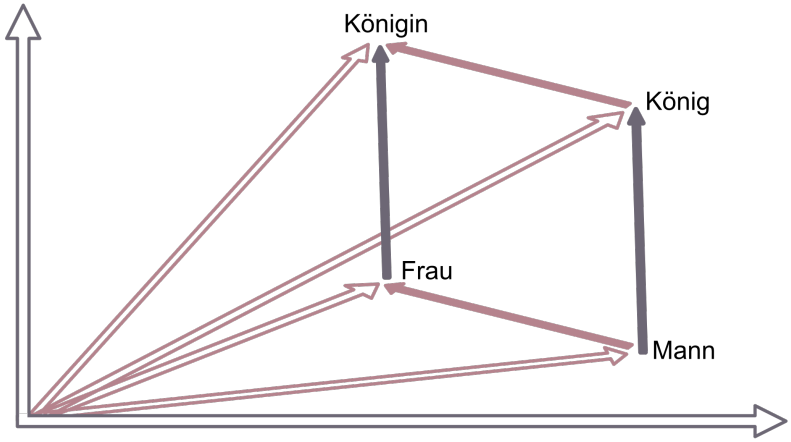


Abbildung 3.2: Vereinfachte Darstellung von Word Embeddings nach [19]

3.1.3 Sprachmodell BERT

Das Sprachmodell Bidirectional Encoder Representations from Transformers (BERT) wird für NLP-Tasks genutzt und dient dabei vor allem zur Beantwortung von Fragen. Er arbeitet dabei wesentlich schneller als seine Vorgänger, da Wörter simultan verarbeitet werden können. Der größte Vorteil von BERT ist, dass er Kontext aus zwei verschiedenen Richtungen simultan erlernen kann, was ihn bidirektional macht. Dies bedeutet, dass das Modell den Kontext eines Wortes auf Basis seiner Umgebung links und rechts des Wortes lernen kann. [20, 21] Um bidirektional arbeiten zu können, nutzt BERT zwei Sequenzen. Um die zweite Sequenz zu erhalten, wird die normale Sequenz gespiegelt. Damit beide Sequenzen simultan kodiert werden können, werden zudem zwei Encoder benötigt. Durch diese Näherung aus zwei verschiedenen Richtungen kann der Kontext und die Beziehungen zwischen Wörtern besser erfasst werden. Das Modell verwendet außerdem Transformers, um die Beziehung eines Wortes in einem Satz zu begreifen. [20] Das Prinzip der Transformers wurde erstmals in dem Paper „Attention is all you need“ [22] vorgestellt. Ein Transformer basiert auf einer Encoder-Decoder-Architektur. Die Komponenten bestehen dabei aus mehreren Encoder bzw. Decoder, welche gestapelt werden. [22] Der Encoder liest die gesamte Sequenz und der Decoder versucht Vorhersagen für die gewünschte Aufgabe zu treffen. [20] Es gibt ein BERT Base-Modell und ein BERT Large-Modell, wobei das Base-Modell aus zwölf Stapeln der Encoder und Decoder besteht und das Large-Modell aus 24 Stapelungen. BERT wird vor allem für Classification- und Question-Answering-Tasks genutzt. [21] Für Chatbots ist das Question-Answering von großer Bedeutung. Bei Recommender Systemen spielt allerdings der Classification-Task eine größere Rolle.

Im Folgenden werden zwei für die Implementierung dieser Arbeit grundlegenden Techniken erläutert. Beide werden meist mit einem Modell, welches Transformers nutzt, umgesetzt. Die erste für diese Arbeit relevante Technik ist Topic Modeling. Topic Modeling ist ein statistisches Verfahren, mit welchem Daten auf Basis ihrer Beziehungen ausgewertet werden können. Damit können The-

men in einer Sammlung von Daten ausfindig gemacht und klassifiziert werden. [23, 24] Auf Basis der Daten werden dann Topics erzeugt, diese sollen häufige Gemeinsamkeiten widerspiegeln. Bei der zweiten Technik handelt es sich um die Zero-Shot-Classification. Die Zero-Shot-Classification wird dadurch definiert, dass Daten mit einem Modell klassifiziert werden können, welche nicht für das Training dieses Modells verwendet wurden. [25]

3.2 Chatbots

Der folgende Abschnitt beschäftigt sich mit dem Themengebiet der Chatbots. Es erfolgt zunächst eine Definition der Begrifflichkeiten. Anschließend wird die grundlegende Architektur eines Chatbots genauer beschrieben, um eine Idee davon zu vermitteln, wie Chatbots funktionieren. Abschließend erfolgt eine Kategorisierung der Chatbots nach verschiedenen Gebieten.

3.2.1 Definition eines Chatbots

Um einen besseren Blick auf die Architektur und die Kategorisierung von Chatbots zu bekommen, wird zunächst der Begriff Chatbot genauer definiert. Der Begriff Chatbot setzt sich aus den Wörtern Unterhaltung (Chat) und Roboter zusammen, da er die Aufgabe hat, mit einer anwendenden Person zu kommunizieren und daraufhin möglichst autonom Aufgaben auszuführen. Oft wird der Begriff Chatbot als ein Synonym für einen Conversational Agent (CA) oder eine virtuelle Assistenz verwendet. [26] In verwandter Literatur wird allerdings zwischen diesen drei Konstrukten unterschieden, weshalb im Folgenden kurz die Unterschiede dargestellt werden. Die Tabelle 3.1 zeigt auf, welche Eigenschaften für den jeweiligen Begriff zwingend notwendig sind und welche nicht.

	Aufgabe	natürliche Sprache
Chatbot	nicht zwingend notwendig	zwingend notwendig
virtuelle Assistenz	zwingend notwendig	nicht zwingend notwendig
Conversational Agent	zwingend notwendig	zwingend notwendig

Tabelle 3.1: Definition des Begriffs Chatbot nach [27]

Ein Chatbot muss der nutzenden Person nicht zwingend eine bestimmte Aufgabe abnehmen. Er kann beispielsweise auch nur als Gesprächspartner dienen. Die virtuelle Assistenz hingegen ist vorrangig dafür zuständig, der anwendenden Person eine bestimmte Aufgabe abzunehmen, dies muss nicht in natürlicher Sprache geschehen. Der CA bildet eine Schnittstelle zwischen diesen beiden Definitionen. Er muss sowohl einen Dialog in natürlicher Sprache führen, als auch der nutzenden Person eine bestimmte Aufgabe abnehmen können. [27]

Da die vorliegende Arbeit in deutscher Sprache verfasst wird, bezieht sich diese Arbeit zukünftig auf die Definition aus dem Buch „Chatbots gestalten mit Praxisbeispielen der Schweizerischen Post: HMD Best Paper Award 2018“ [27]. Der Chatbot in der vorliegenden Arbeit führt einen Dialog in natürlich-sprachlicher Form und nimmt der anwendenden Person bestimmte Aufgaben

ab. Deshalb trifft jede der drei Begriffsdefinitionen auf ihn zu. Die Definition des CA ist die am engsten gefasste zutreffende Definition, da der Chatbot dieser Arbeit beide Kriterien erfüllt. Der Verständlichkeit und Konsistenz wegen wird im Folgenden trotz dessen nur noch der Begriff Chatbot verwendet, da dieser ebenfalls zutrifft und in verwandter Literatur am verbreitetsten ist.

3.2.2 Architektur eines Chatbots

Die Aufgabe eines Chatbots besteht darin, eine Eingabe der anwendenden Person anzunehmen, diese zu verarbeiten und anschließend eine angemessene Reaktion als Antwort zu liefern. [27] Im Folgenden wird die Architektur, welche es dem Chatbot ermöglicht zu antworten, genauer erläutert. Diese wird in Abbildung 3.3 abgebildet. hierbei ist zu erkennen, welche verschiedenen Stufen eine Eingabe durchläuft und welche Methoden dafür notwendig sind.

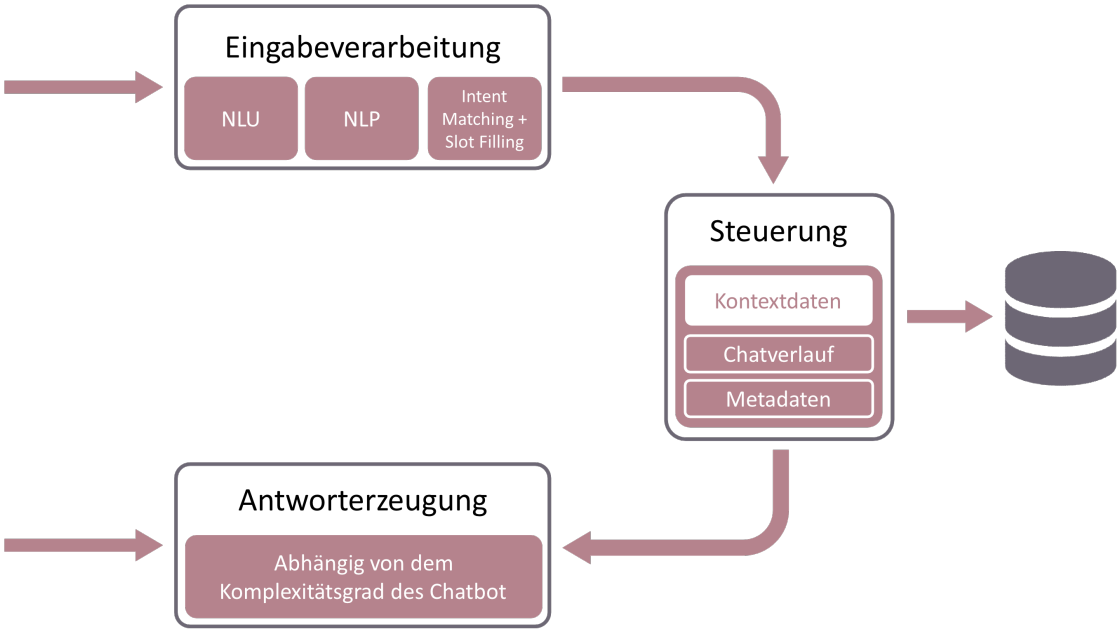


Abbildung 3.3: Architektur eines Chatbots nach [27]

Wenn die nutzende Person eine Eingabe tätigt, wird diese an die Eingabeverarbeitung weitergeleitet, welche versucht, die Eingabe zu verstehen. Dieser Prozess wird NLU genannt. Die Eingabeverarbeitung versucht, die Absicht der nutzenden Person herauszufiltern. Diese Absicht wird im Bereich der Chatbots Intent genannt und beschreibt dabei die Informationsbasis, aus welcher Kontext geschaffen wird. Um den Intent herauszufiltern, wird der Text zunächst bereinigt, indem unter anderem Satzzeichen und Groß- und Kleinschreibung entfernt werden. [27] Um die Eingabe aufzubereiten, werden Methoden des NLP verwendet. Näheres zu diesen Methoden wird im Abschnitt 3.1.2 aufgearbeitet. Der Text liegt nach diesem Vorgang in einer strukturierten Form vor. Mit diesen strukturierten Textdaten kann das Intent Matching beginnen. Hierbei wird versucht, den Intent der anwendenden Person herauszufinden. [27] Das Intent Matching kann entweder regelbasiert oder mithilfe von maschinellem Lernen stattfinden. Der erhaltene Input wird mit bereits vorhandenen Input-Output-Paaren verglichen. Daraufhin wird die Kombination, welche den

höchsten Vertrauenswert besitzt, herausgefiltert und ausgewählt. [17] Neben dem Intent Matching findet auch das Slot Filling statt. Hierbei wird versucht, Elemente aus den Textdaten zu ziehen, welche auf eine bestimmte Entität passen könnten. Eine Entität liefert damit erweiterte Informationen, um den Kontext zu spezifizieren. [16] Der Intent und die Entitäten werden dann an die Steuerung übergeben, diese bewertet folglich, wie mit dem Intent umgegangen werden soll. Bei einem simpleren Frage-Antwort-Chatbot bezieht die Steuerung damit Informationen, beispielsweise aus einer Datenbank. Wenn der Chatbot allerdings kontextsensitiv handeln soll, müssen die genutzten Metadaten auf dieser Ebene miteinbezogen werden. [27]

Wenn die Steuerung alle benötigten Informationen für eine Antwort hat, gibt sie diese an die Antworterzeugung zurück. Die Antworterzeugung erzeugt daraufhin aus den Informationen wieder natürlich-sprachlichen Text. Dieser Vorgang wird NLG genannt. Bei simplen Chatbots gibt es für die jeweilige Rückgabe der Steuerung vordefinierte Aussagen, dadurch entstehen immer statische Antworten. Bei fortgeschritteneren Chatbots werden die Antworten noch parametrisiert und können mit Informationen aus der Steuerungskomponente ergänzt werden, um eine spezifischere Antwort auszugeben. Auf der höchsten Komplexitätsstufe kann die Antworterzeugung selbstständig Antwortsätze dynamisch erzeugen. [27]

3.2.3 Kategorisierung von Chatbots

Da Chatbots in den letzten Jahren immer vielfältiger geworden sind, gibt es verschiedenste Ansätze, nach denen sie klassifiziert werden können. Chatbots können anhand ihrer Funktion, ihrem Einsatzgebiet oder anhand von technischen Eigenschaften klassifiziert werden. [28] Da der Fokus dieser Arbeit auf den technischen Komponenten von Chatbots liegt, wird explizit eine Unterscheidung dieser durchgeführt. Hierbei gibt es verschiedene Klassifikationsansätze. Auf den Großteil dieser wird im Folgenden nur kurz Bezug genommen. Da die Klassifikation nach Designansätzen für diese Arbeit eine wichtige Rolle spielt, wird diese ausführlicher beschrieben. [29] Die Grafik 3.4 veranschaulicht in einer Übersicht die möglichen Klassifikationsachsen.

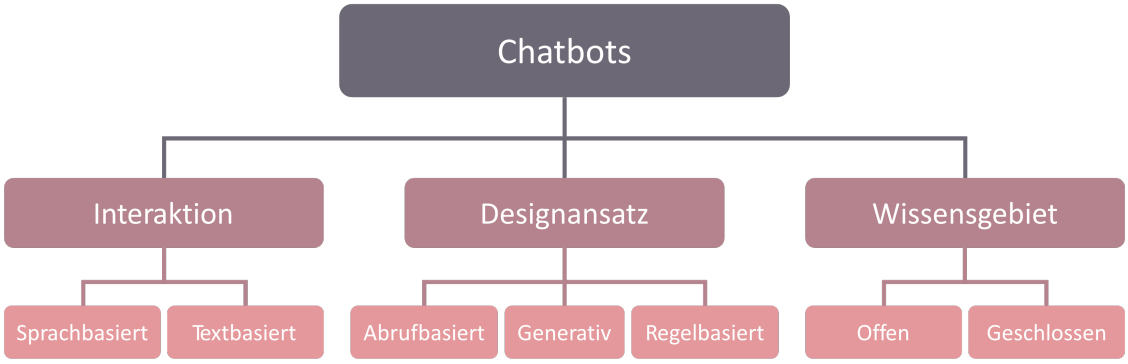


Abbildung 3.4: Kategorisierung von Chatbots nach [29]

Klassifizierung nach Interaktion

Chatbots lassen sich danach klassifizieren, wie mit ihnen interagiert wird. Hier wird zwischen textbasierten und sprachbasierten Chatbots unterschieden. Die sprachbasierten Chatbots müssen einen zusätzlichen Schritt in ihrer Architektur durchlaufen, da sie die Verfahren Speech-to-Text und Text-to-Speech benötigen, um die Daten richtig zu verarbeiten. [16] Da in Kapitel 6 kein sprachbasierter Chatbot umgesetzt wird, wird im Folgenden nicht genauer auf dieses Verfahren eingegangen.

Klassifizierung nach Designansätzen

Bei der Klassifizierung nach Designansätzen wird zwischen einem regelbasierten (Rule-Based), einem abrufbasierten (Retrieval-Based) und einem generativen (Generative-Based) Chatbot unterschieden. [29]

Die simpelste Art eines Chatbots ist ein regelbasierter Chatbot. Der erste Chatbot, welcher entwickelt wurde, war ein regelbasierter Chatbot, der nach vordefinierten Regeln antwortet. [16] Der vordefinierte Satz an Regeln basiert auf der Erkennung der lexikalischen Form des Inputs. Der regelbasierte Chatbot erzeugt somit keine eigenen Antworten, sondern verwendet ausschließlich von Menschen vordefinierte Sätze. [28] Die Antworten können lediglich verbessert werden, indem die Regeldatenbank umfassender gestaltet wird. [26]

Die abrufbasierten Chatbots besitzen einen Pool an vordefinierten Antworten, aus welchem sie die passenden Antworten nach einer Heuristik auswählen. Sie können den regelbasierten Chatbots sehr ähneln, wenn sie die regelbasierte Ausdrucksübereinstimmung als Heuristik verwenden. Ein abrufbasierter Chatbot kann aber auch wesentlich komplexer aufgebaut sein, indem er eine Kombination aus Machine Learning Klassifikatoren verwendet. Abrufbasierte Chatbots erzeugen jedoch niemals neue Antworten, sondern wählen lediglich passende Antworten aus dem vorgegebenen Pool aus. [29] Ein Vorteil des abrufbasierten Chatbots gegenüber des regelbasierten Chatbots ist außerdem, dass er verfügbare Ressourcen auch über Programmierschnittstellen (APIs) abfragen und analysieren kann, was die Menge an Daten, welche genutzt werden können und damit den Umfang und die Flexibilität, in welchem der Chatbot antworten kann, erhöht. [30] Ein weiterer Unterschied zu einem regelbasierten Chatbot ist, dass mehrere Antwortkandidaten zur Auswahl stehen und daraufhin der passende ausgewählt und angewendet wird. [31] Regelbasierte und abrufbasierte Chatbots haben den Vorteil, dass sie weniger grammatikalische Fehler machen, da die Antworten nur ausgewählt werden. Allerdings scheitern sie, sobald ihnen eine Frage gestellt wird, auf die keine der vordefinierten Antworten passt. Außerdem beziehen sie oft nicht den Kontext von vorherigen Antworten und Metadaten mit ein, was sie in diesen Fällen unflexibel macht. [29, 26]

Im Gegensatz zu abrufbasierten Chatbots sind generative Chatbots nicht abhängig von vordefinierten Antworten. Sie erzeugen neue Antworten mithilfe von verschiedenen Techniken der maschinellen Übersetzung. Die Eingabe wird dabei in eine Antwort statt in eine andere Sprache

übersetzt. [29] Generative Chatbots haben die Möglichkeit kontextsensitiv zu antworten, was ihnen den Vorteil verschafft, persönlicher und gezielter antworten zu können. [31] Allerdings sind sie schwerer zu trainieren und durch das selbstständige Erzeugen von Antworten auch anfälliger für grammatikalische Fehler. Zudem sind sie deutlich schwieriger zu kontrollieren, da sie sich unvorhersehbare Kommunikationsstrategien aneignen. Deshalb erfordert ein generativer Chatbot eine große Menge an Trainings- und Testdaten. [29]

Klassifizierung nach Wissensgebiet

Eine weitere Art der Klassifizierung bei Chatbots kann das Wissensgebiet sein, welches ihnen zur Verfügung steht oder mit welchem sie trainiert wurden. [29] Es wird zwischen einem offenen und einem geschlossenen Wissensgebiet unterschieden. Chatbots mit einem offenen Wissensgebiet haben den Vorteil, dass sie über einen breit gefächerten Bereich an Themen informiert sind. Da das Wissensgebiet so breit gefächert ist, ist ihr Wissen in den einzelnen Bereichen eher allgemein und oberflächlich. Chatbots mit einem geschlossenen Wissensbereich hingegen konzentrieren sich auf bestimmte Themen und haben in diesen auch eine gewisse Tiefe. Sie kommen jedoch schnell an ihre Grenzen, sobald sich die Frage aus diesem Wissensbereich entfernt. [32]

3.3 Recommender Systeme

Recommender Systeme werden heutzutage immer öfter eingesetzt, um trotz großer Mengen an Informationen relevante und neue Inhalte zu erhalten. [33] Ein verbreiteter Anwendungszweck von Recommender Systemen liegt im kommerziellen Online Shopping oder in der Empfehlung von Musik und Film. [4] Recommender Systeme können jedoch auch als Bestandteil eines Chatbots genutzt werden, um der anwendenden Person für sie hilfreiche Informationen zu liefern. [5]

3.3.1 Definition und Abgrenzung von Recommender Systemen

In der vorliegenden Arbeit wird der Begriff Recommender System verwendet, statt der deutschen Übersetzung Empfehlungssystem. Dieser Begriff hat sich im deutschen Sprachgebrauch mittlerweile durchgesetzt. Ein Recommender System wird in der vorliegenden Arbeit als ein System definiert, welches einer nutzenden Person Items vorschlägt, welche für die nutzende Person von Interesse sein könnten. [34] Ein Item beschreibt dabei den Gegenstand der Empfehlung eines Recommender Systems. Ein Recommender System arbeitet mit verschiedenen Techniken und Werkzeugen, um Informationen zu filtern. [35] Der grundlegende Zweck eines Recommender Systems besteht deshalb darin, die nutzende Person auf andere Items aufmerksam zu machen, welche diese auf Basis ihrer bisherigen Daten auch interessieren könnten. Bei einer passenden Empfehlung steigt deshalb die Zufriedenheit der anwendenden Person. [36]

3.3.2 Kontextsensitivität eines Recommender Systems

Es gibt eine Vielzahl an Definitionen, welche versuchen den Begriff Kontext zu beschreiben. Diese fallen zum Teil sehr unterschiedlich aus, weshalb im Folgenden mehrere Definitionen aufgegriffen

werden. Eine oftmals genutzte Definition für Kontext beschreibt diesen als Information, welche genutzt werden kann, um eine Situation zwischen einer anwendenden Person und einer Anwendung zu charakterisieren. Ein System ist somit kontextsensitiv, wenn es der anwendenden Person für sie wichtige Informationen zur Verfügung stellt, welche das System durch den Kontext erarbeiten konnte. [37]

Kontext wird oft auf zwei unterschiedliche Arten definiert. Zum einen kann Kontext die Einflüsse im Zusammenhang mit einer bestimmten Situation beschreiben. Zum anderen wird Kontext als der Text, welcher vor oder nach einem bestimmten Textabschnitt steht, beschrieben. Wenn der Textausschnitt unabhängig von seinem Kontext betrachtet wird, kann es vorkommen, dass die Bedeutung von diesem unvollständig oder gar verfälscht wird.[38] Vor allem um Fehlinterpretationen zu vermeiden, spielt Kontext eine essenzielle Rolle, da viele Wörter eine Mehrdeutigkeit aufweisen. Indem ein Modell den Kontext, in welchem ein Wort steht, beachtet, ist es ihm möglich, die Mensch-Maschine-Interaktion zu verbessern. Ein kontextsensitives Recommender System muss somit eine Art Gedächtnis besitzen, um flexibel reagieren zu können. [39, 40]

Aus den oben gelisteten Definitionen lässt sich schließen, dass es verschiedene Kontextarten gibt, welche im Folgenden vorgestellt werden. Der Domainkontext beschreibt das konzeptionelle Wissen über die Umgebung und die Welt im Allgemeinen. Der Dialogkontext beschreibt das Wissen darüber, was in einem Gespräch oder in einem Text gesagt wurde. Der Kontext der nutzenden Person beschreibt das Wissen über die Gesprächsbeteiligten. Der situative Kontext beschreibt das Wissen, zu welcher Zeit und an welchem Ort etwas stattgefunden hat. [41] Für die vorliegende Arbeit sind mehrere der Kontextarten relevant. Der Fokus liegt allerdings vor allem auf dem Dialogkontext. Auch der Kontext der nutzenden Person, der Domainkontext und der situative Kontext sollen miteinbezogen werden. Auf das Recommender System bezogen, kann der Grad an Kontextsensitivität eines Recommender Systems berücksichtigt werden. Ein Recommender System kann kontextfrei handeln, indem es nur Keywords wahrnimmt und diese mit in die Wahl der Empfehlung miteinbezieht. Ein Recommender System kann auch eine simple Form der Kontextsensitivität anstreben, indem es kontextsensitiv innerhalb eines isolierten Kontexts handelt. Beispielsweise könnte das innerhalb eines Satzes der Fall sein. Damit ein Recommender System als kontextsensitiv bezeichnet werden kann, sollte es neben dem Kontext der Konversation auch den Dialogkontext beachten. Das heißt, es sollte in der Lage sein, vorherige Nachrichten, Daten der nutzenden Person oder situative Daten miteinzubeziehen. Da ein in einen Chatbot integriertes Recommender System nur miteinbeziehen kann, was zuvor schon gesagt wurde, wird der Dialogkontext in diesem Zusammenhang nur durch vorgehende Dialoge gekennzeichnet. Im Folgenden wird ein Recommender System als kontextsensitiv beschrieben, wenn die vorhergehende Konversation miteinbezogen wird oder wenn Metadaten, wie Daten der nutzenden Person oder situative Daten, bei der Wahl der Empfehlung miteinbezogen werden.

3.3.3 Arten von Recommender Systemen

Es gibt verschiedene Arten von Recommender Systemen, im Folgenden wird ein Ausschnitt dieser aufgezeigt. Grundsätzlich können Recommender Systeme aufgrund ihrer Datengrundlage unterschieden werden. Dabei wird zwischen inhaltsbasierten Recommender Systemen und kollaborativen Recommender Systemen unterschieden. Der Fokus wird im Folgenden vor allem auf diesen zwei Unterscheidungen liegen. Es wird in dieser Arbeit aber auch auf andere Typen von Recommender Systemen eingegangen, sowie auf hybride Systeme. [36] Die Tabelle 3.2 zeigt eine Übersicht der wichtigsten Recommender Systeme mit ihren Zielen und Inputs, auf welche im Folgenden einzeln eingegangen wird.

Recommender System	**Ziel**	**Input**
inhaltsbasiert	Empfehlungen sollen, aufgrund der zuvor bewerteten oder gekauften Artikel ausgesprochen werden	Bewertung der anwendenden Person + Artikeleigenschaften
wissensbasiert	Empfehlungen sollen aufgrund der zuvor von der anwendenden Person festgelegten Spezifikation ausgesprochen werden	Spezifikation der anwendenden Person + Artikeleigenschaften + Domainkontext
kollaborativ	Empfehlungen sollen aufgrund der Bewertung und des Kaufes von ähnlichen Anwendenden ausgesprochen werden	Bewertung der anwendenden Person + Gemeinschaftsbewertungen
demografisch	Empfehlungen sollen aufgrund des demografischen Profils einer anwendenden Person ausgesprochen werden	Bewertung der anwendenden Person + Spezifikation der anwendenden Person

Tabelle 3.2: Abgrenzung von Recommender Systemen nach [35, 42]

Inhaltsbasiertes Recommender System

Ein inhaltsbasiertes Recommender System betrachtet nur den Anwendenden selbst, das heißt es stellt keinen Bezug zu anderen anwendenden Personen her. Es lernt beispielsweise ihnen Items zu empfehlen, die Ähnlichkeiten mit Items haben, welche der nutzenden Person in der Vergangenheit gefallen haben. Meist gleichen inhaltsbasierte Recommender Systeme die Attribute des Kontos einer nutzenden Person mit den Attributen der Artikel ab, um Empfehlungen zu geben. [36] Die Ähnlichkeit wird durch die Merkmale, welche die Artikel verbinden, berechnet. [43] Der Begriff Inhalt in inhaltsbasierten Recommender Systemen bezieht sich auf diese Merkmale, welche aus Keywords bestehen, die aus den Beschreibungen der Artikel entstammen. Inhaltsbasierte Recommender Systeme verwenden die Artikelbeschreibungen und ihre Bewertung der anwendenden Per-

son als Trainingsdaten und erstellen damit ein für eine nutzende Person spezifisches Regressions- oder Klassifikationsmodellierungsproblem. Die Trainingsdokumente von jedem Nutzenden entsprechen damit den Beschreibungen der Artikel, die er bewertet oder gekauft hat. Das für die nutzende Person spezifische Modell, welches daraus entsteht, wird verwendet, um vorherzusagen, ob diese Person einen Artikel mögen wird. [35] Inhaltsbasierte Recommender Systeme sind sinnvoll, wenn noch kein Zugang zu Informationen von anderen Anwendenden besteht. Denn ohne Informationen von anderen Anwendenden ist es nicht möglich, kollaborative Filterung einzusetzen. [35]

Kollaborative Filterung

Kollaborative Filterung empfiehlt einer nutzenden Person einen Artikel, welche andere Nutzende kauften, die ähnliche Merkmale wie die ursprüngliche Person aufweisen. Es wird also die Annahme getroffen, dass zwei Nutzende, welche sich bei einem Artikel einig sind, sich eher auch bei anderen Artikeln einig sind. [44] Die kollaborative Filterung teilt sich meist in zwei größere Gruppen ein. Zum einen die speicherbasierten (memory-based) und zum anderen die modellbasierten (model-based) Algorithmen. Die erste Definition, in welcher diese beiden Algorithmen unterschieden wurden, besagt, dass speicherbasierte Algorithmen über den gesamten Datensatz operieren. Modellbasierte Algorithmen hingegen verwenden den Datensatz, um ein Modell zu bewerten, damit es dann Vorhersagen treffen kann. [45] Diese Definition lässt sich jedoch nicht auf Recommender Systeme übertragen, weshalb die Unterscheidung in diesem Fall genutzt wird, um die Designunterschiede eines Algorithmus darzustellen. Demnach verwenden modellbasierte Algorithmen verschiedene Techniken des maschinellen Lernens, um ein parametrisiertes Modell anzupassen. Speicherbasierte Algorithmen hingegen durchsuchen die Trainingsdaten, um ähnliche Nutzende oder Elemente zu finden. Wenn die speicherbasierten Algorithmen diese gefunden haben, werden sie aggregiert, um Empfehlungen zu berechnen. [46] Neben dieser Definition gibt es allerdings auch Quellen, welche die grundlegende Unterteilung in modellbasierte und speicherbasierte Algorithmen kritisieren. Diese differenzieren die Algorithmen stattdessen nach ihrer mathematischen Struktur und Motivation. [47] Da die Definition nach [46] momentan noch die gängigste ist, wird im Folgenden diese Unterscheidung verwendet.

Wissensbasiertes Recommender System

Eine weitere Art eines Recommender Systems ist ein wissensbasiertes Recommender System. Sie geben Empfehlungen auf Basis der Ähnlichkeiten zwischen den Anforderungen einer nutzenden Person und den Artikelbeschreibungen. Dieser Prozess wird durch eine Wissensdatenbank unterstützt, welche hinter dem System steht und die Daten über Regeln und Ähnlichkeitsfunktionen enthält. Da die Kundschaft bei wissensbasierten Recommender Systemen angeben muss, welche Präferenzen sie bezüglich Items besitzen, sind diese Systeme hoch interaktiv und setzen voraus, dass die anwendende Person eine ungefähre Selbsteinschätzung hat, was ihr gefällt und was nicht. [35]

Die wissensbasierten Recommender Systeme beziehen sich vor allem auf Domainwissen, welches in Abschnitt 3.3.2 spezifiziert wurde. Sie stehen den inhaltsbasierten Recommender Systemen sehr nahe und werden teilweise als eine Kategorie zusammengefasst. [48] In der vorliegenden Arbeit werden sie trotzdem getrennt gelistet aufgrund des unterschiedlichen Wissens, mit welchem sie jeweils arbeiten. Wissensbasierte Recommender Systeme werden aufgrund des Wissens, welches sie benötigen, vor allem für Artikel mit wenigen Bewertungen oder Käufen eingesetzt. Es ist somit ein passendes System, um das Cold-Start-Problem zu überbrücken. [35] Das Cold-Start-Problem beschreibt die Problematik der Datenlücke, welche auftritt, sobald ein neuer Artikel oder eine neue nutzende Person hinzugefügt wird. Für die anwendende Person oder den Artikel gibt es noch keine Bewertungen oder Käufe, wodurch es zu Datenlücken kommt. [49] Wissensbasierte Recommender Systeme können nochmals anhand ihrer Schnittstelle unterschieden werden. Bei einschränkungsbasierten (constraint-based) Systemen geben die Nutzenden Beschränkungen für die Merkmale der Artikel vor, wie beispielsweise eine Ober- und Untergrenze. Diese Beschränkungen werden als Regeln gespeichert, welche dann mit den Artikelmerkmalen abgeglichen werden. [50] Die zweite Art eines wissensbasierten Recommender Systems ist das fallbasierte Recommender System. Hier muss die anwendende Person bestimmte Fälle als Ziele angeben, welche erreicht werden sollen. [48, 36]

Demografisches Recommender System

Die demografischen Recommender Systeme nutzen die demografischen Informationen der Anwendenden, um Klassifikatoren zu erlernen, die dann demografische Merkmale auf Bewertungen oder Kaufneigungen abbilden können. Oftmals werden die demografischen Informationen mit zusätzlichem Kontext kombiniert, um den Empfehlungsprozess besser zu steuern. [35]

Hybride Recommender System

Hybride Recommender Systeme verknüpfen verschiedene Arten von Recommender Systemen. Da durch diese Kombinationen sehr viele verschiedene hybride Arten von Recommender Systemen entstehen, werden diese nicht einzeln unterschieden. Stattdessen werden die hybriden Recommender Systeme im Folgenden nach Art der Verknüpfung unterschieden. Das Modell, welches vor allem genutzt wird, um das Cold-Start-Problem zu lösen, ist das wechselnde hybride Recommender System. Dabei wird zwischen den verschiedenen Empfehlungstechniken immer dann gewechselt, wenn sich ein anderes System zu einem bestimmten Zeitpunkt als effektiver erweist. Ein weiteres Modell ist das gewichtete hybride Recommender System. Dabei werden die Erkenntnisse aus allen genutzten Recommender Systemen mit ihrer jeweiligen Gewichtung kombiniert. Danach wird entweder die Schnittmenge oder die Vereinigung der Gruppe verwendet. [42] Zuletzt gibt es noch das gemischte hybride System. Diese bilden zunächst Gruppen aus verschiedenen Recommender Systemen, welche zusammen gewichtet werden und aus dieser Gewichtung ein Ergebnis erzeugen. Zwischen diesen verschiedenen hybriden Recommender Systemen kann dann gewechselt werden. [35]

3.3.4 Bewertung der Arten von Recommender Systemen

Im Folgenden werden die zwei grundlegenden Arten von Recommender Systemen, die inhaltsbasierten Recommender Systeme und die kollaborative Filterung, gegenübergestellt und bewertet.

Inhaltsbasierte Recommender Systeme sind sinnvoll, wenn keine ausreichende Datengrundlage für neue Artikel vorliegt, da sie die Keywords, mit welchen sie die Artikel mit anderen Artikeln abgleichen, aus der Beschreibung der Artikel ziehen. Wenn der neue Artikel also noch keine Historie an Bewertungen hat, kann er trotzdem einer nutzenden Person aufgrund seiner Merkmale empfohlen werden. [35] Sobald allerdings statt eines neuen Artikels eine neue anwendende Person angelegt wird, werden inhaltsbasierte Recommender Systeme nachteilig. Denn um Vorhersagen zu treffen benötigt das Trainingsmodell eine Historie an Bewertungen und Käufen von der neuen nutzenden Person, welche noch nicht existiert. Die Vorhersagen bei neuen Anwendenden sind also wenig robust. Eine Lösung, die sich hierfür anbietet, wäre die nutzende Person beim Anlegen ihres Profils verschiedene Keywords angeben zu lassen, um direkt eine Datenbasis zu schaffen, welche verwendet werden kann. Diese Art der Datenerhebung und -verarbeitung fällt allerdings eher unter die Kategorie der wissensbasierten Chatbots, auf welche in Abschnitt 3.3.3 näher eingegangen wurde. Daraus lässt sich ein hybrides System schaffen, welches erst dann inhaltsbasiert arbeitet, sobald genug Daten zur Verfügung stehen. [42] Inhaltsbasierte Modelle bringen aber auch noch einen anderen Nachteil mit sich. Artikel mit Keywords, die nicht in anderen von der nutzenden Person gekauften oder empfohlenen Artikeln verwendet werden, haben keine Chance dieser Person empfohlen zu werden. Das könnte zur Folge haben, dass die Person sich in einer Filterblase befindet und ganze Segmente nicht wahrnehmen kann, die ihm aber auch gefallen könnten. Dies liegt unter anderem daran, dass kein Gemeinschaftswissen genutzt werden kann, was zu weniger Vielfalt in den empfohlenen Artikeln führen kann. [35]

Kollaborative Filterung hat den Vorteil, dass oftmals eine große Menge an Daten vorliegt, welche miteinbezogen werden können. Die große Datenmenge begründet sich darin, dass kollaborative Filterung sich nicht auf die Daten einer einzelnen nutzenden Person bezieht, sondern die Daten von allen Nutzenden miteinbeziehen kann. Die großen Datenmengen können allerdings auch zu Problemen führen, da bei den meisten speicherbasierten Algorithmen der kollaborativen Filterung eine Verbindung von jeder nutzenden Person zu jedem Artikel berechnet werden muss. Bei vielen Artikeln kann dies zu längeren Laufzeiten und erhöhtem Speicherverbrauch führen. Das Cold-Start-Problem, welches bei einem inhaltsbasierten Recommender System besteht, verstärkt sich hier sogar noch. Dieses tritt in diesem Fall ein, wenn ein neues System aufgesetzt wird oder wenn neue Nutzende oder neue Artikel hinzukommen. Dann gibt es keine bis wenige Bewertungen oder Käufe, weshalb sich auch keine kollaborative Filterung anwenden lässt. Eine Lösung hierfür wäre ein hybrides System, wie es zuvor erläutert wurde. Je nach Datenlage und Datenknappheit lässt sich dieses auch in Kombination mit einem inhaltsbasierten System umsetzen. Wie bereits beim inhaltsbasierten System erwähnt, besteht auch in diesem Fall die Gefahr der Bildung einer Filterblase. Dies kann eintreten, da ihnen nur Artikel vorgeschlagen werden, welche ähnliche Personen

gekauft haben. Dieses Problem lässt sich durch die Veränderung der Algorithmen abschwächen. Hierzu können absichtlich kontroverse oder generische Empfehlungen mit eingebunden werden. Dies ist allerdings vor allem ein gesellschaftspolitisches Thema, welches in der vorliegenden Arbeit nicht näher behandelt werden kann. Ein weiteres Problem, welches sich bei kollaborativer Filterung ergeben kann ist, dass es Nutzende gibt, welche nicht mit der Masse übereinstimmen. Für diese Personen sind Empfehlungen, welche auf einer kollektiven Gruppe basieren, weder sinnvoll noch hilfreich.

Abschließend lässt sich sagen, dass beide Arten von Recommender Systemen Vor- und Nachteile mit sich bringen. Die kollaborative Filterung ist jedoch die fortgeschrittenere Technik, welche meist bessere Ergebnisse hinsichtlich der Empfehlungen erzielen kann. Um ein kollaboratives Recommender System mit allen Vorteilen zu nutzen, wird jedoch eine große Basis an Daten benötigt, ohne welche ein kollaboratives Recommender System nicht zu empfehlen ist. Um die genannten Probleme der kollaborativen Filterung zu umgehen, könnte auf hybride Systeme zurückgegriffen werden, welche erst mit kollaborativer Filterung arbeiten, wenn die Datengrundlage dafür geschaffen wurde. Zusammenfassend wird angemerkt, dass die Wahl des Recommender Systems immer nach dem spezifischen Anwendungsfall ausgewählt und angepasst werden muss. Welches Recommender System sich für die in dieser Arbeit vorliegende Implementierung in Kapitel 6 am besten eignet, wird deshalb in der Konzeption in Kapitel 5.3 erarbeitet.

4 Verwandte Arbeiten

Im Folgenden werden verwandte wissenschaftliche Arbeiten in Bezug zu der vorliegenden Arbeit gesetzt. Ziel des Kapitels ist es, aufzuzeigen, inwiefern sich die aktuelle Forschung mit den für diese Arbeit relevanten Themen auseinandergesetzt hat und welche Lösungswege schon bekannt sind. Außerdem soll die Verbindung zur eigenen Arbeit deutlich werden. Es wird analysiert, in welchen Bereichen schon vorhandenes Wissen genutzt werden kann und wo noch Forschungsbedarf besteht. Daraufhin wird die Verknüpfung der Gebiete „Chatbots“ und „Recommender Systeme“ im Zusammenhang mit dem Gesundheitssektor betrachtet. Dabei werden Ähnlichkeiten zu verwandten Arbeiten aufgedeckt und Zusammenhänge hergestellt. Die Grafik 4.1 zeigt die Verbindung der Gebiete, welche für diese Arbeit von besonderer Relevanz sind.

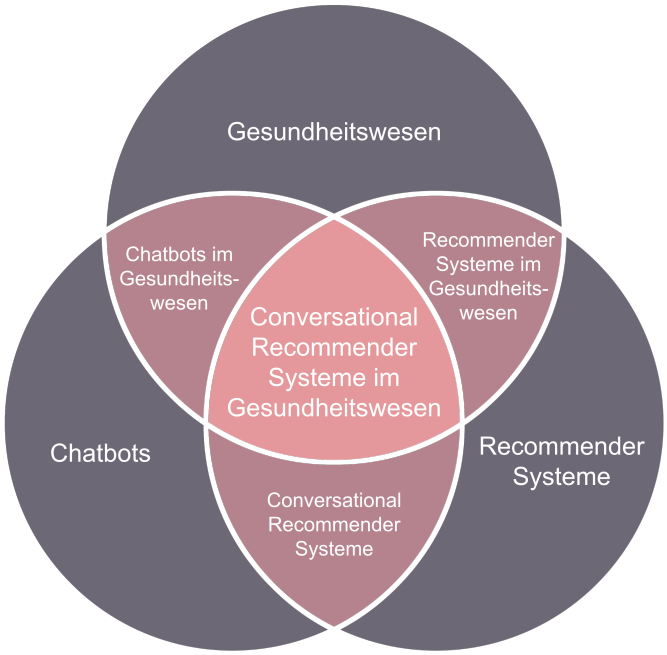


Abbildung 4.1: Überblick der Themen der verwandten Arbeiten nach

Die Kreise zeigen die verschiedenen Themengebiete auf. In der Mitte befinden sich die dialogorientierten Recommender Systeme im Gesundheitswesen, welche die Themengebiete miteinander kombinieren und die zentrale Thematik der Arbeit darstellen. Zunächst wird im Abschnitt 4.1 auf Chatbots im Gesundheitswesen eingegangen, danach wird im Abschnitt 4.2 das Thema Health Recommender Systeme (HRS) betrachtet. Die Verknüpfung von Recommender Systemen mit Chatbots wird im Abschnitt 4.3 hergestellt. Abschließend wird im Abschnitt 4.4 der aktuelle Forschungsstand der Recommender Systeme bei Chatbots im Gesundheitswesen beleuchtet.

Um die verwandten Arbeiten zu finden, wird vor allem nach den Suchbegriffen „Conversational Recommender System“, „Health Recommender System“, „Chatbots in Healthcare“ und „Conversational Health Recommender System“ gesucht. Dies geschah primär auf der Plattform „Google Scholar“, dem Katalog der Hochschulbibliothek Offenburg und dem Katalog der Bibliothek der Hochschule Karlsruhe. Bei den dort gefundenen Ergebnissen werden vor allem Paper von „IEEE“, „Springer Link“, „Research Gate“ und „ScienceDirect“ verwendet.

Aufgrund des stetigen und schnellen Wandels in den relevanten Themengebieten NLP, Chatbots und Recommender Systeme, wird darauf geachtet, die Literatur in diesem Kapitel in der Jahreszahl einzugrenzen. Im Jahr 2017 standen Chatbots kurz vor dem Peak auf dem Hype Cycle[4] von [51]. Der Peak beschreibt im Zusammenhang mit dem Hype Cycle den Zeitpunkt, an welchem die Kurve ihren höchsten Punkt erreicht. An diesem Punkt ist der Hype um die entsprechende Technologie am höchsten. [51] Deep Learning und Machine Learning waren 2017 auf ihrem Peak in dem Hype Cycle. 2017 wurde zudem das Paper „Attention is all you need“ [22] veröffentlicht, welches den neuen Ansatz der Transformers[5] auf dem Gebiet der neuronalen Netze vorstellte. Jedoch war das Themengebiet der KI auch 2016 schon weiter vorangeschritten, dies lässt sich auch an dem Hype Cycle aus dem Jahr 2016 erkennen. Die Themen KI und Machine Learning zählten in diesem Jahr zu einem der vier Megatrends des Hype Cycle. [52] Deshalb wird sich im Folgenden ausschließlich auf Literatur bezogen, welche im Jahr 2016 oder später veröffentlicht wurde.

4.1 Chatbots im Gesundheitswesen

Chatbots werden zunehmend in Bereichen des Gesundheitswesens eingesetzt. Sie werden dort für verschiedene Zwecke entwickelt, entweder um Mitarbeitende zu entlasten oder um Erkrankte zu unterstützen. Vor allem die COVID-19 Pandemie hat die Forschung in diesem Bereich beschleunigt, da auch hierfür Lösungen mit Chatbots entwickelt wurden. Ein Beispiel dafür ist ein Chatbot, der die Mitarbeitenden des Gesundheitswesens täglich befragt und mit diesen Informationen prüft, ob sie arbeiten können oder ob sie sich aufgrund von Symptomen krankschreiben lassen sollten. [3]

In dem Paper „Survey of conversational agents in health“ [53] wird die Taxonomie von Chatbots im Gesundheitswesen genauer betrachtet. Chatbots im Gesundheitswesen werden dabei in wiederkehrende zentrale Konzepte kategorisiert. Die drei Knoten, die dabei entstehen, sind „Interaktionen“, „Dialoge“ und „Architekturen“. Diese Knoten werden daraufhin einzeln in Form von Attributen weiter spezifiziert, um den Stand der Technik auf dem jeweiligen Gebiet aufzuzeigen. Der erste Knoten „Interaktionen“ definiert den Kontext für Chatbots im Gesundheitswesen

[4] Der Hype Cycle von Gartner ist ein grafisches Modell, welches den Reifegrad und die Akzeptanz von bestimmten Technologien abbildet. Es zeigt außerdem, wie sich die Technologien im Laufe der Zeit entwickeln werden. [51]

[5] Ein Transformer ist eine Art künstliches neuronales Netz, welches dazu dient Eingabedaten in eine neue Form zu übersetzen

und bezieht sich dabei nicht auf die Art des Vertreters. Dieser Knoten enthält die Attribute „Gesundheitsziele“, „Gesundheitskontexte“ und „Bereiche des Gesundheitswesens“. Der zweite Knoten „Dialoge“ definiert die Art des Chatbots und dessen Kommunikationsmodell. Er besteht aus den Merkmalen „Dialogtypen“, „Vertretertypen“ und den „Kommunikationsmodellen“. Der letzte Knoten „Architekturen“ definiert die technischen Komponenten des Chatbots. Darin enthalten sind die Attribute „Techniken“ und „Systeme“, welche ein Chatbot im Gesundheitswesen verwenden kann. Die folgende Abbildung 4.2 verdeutlicht diese Unterteilung in die einzelnen Knoten mit ihren jeweiligen Attributen.

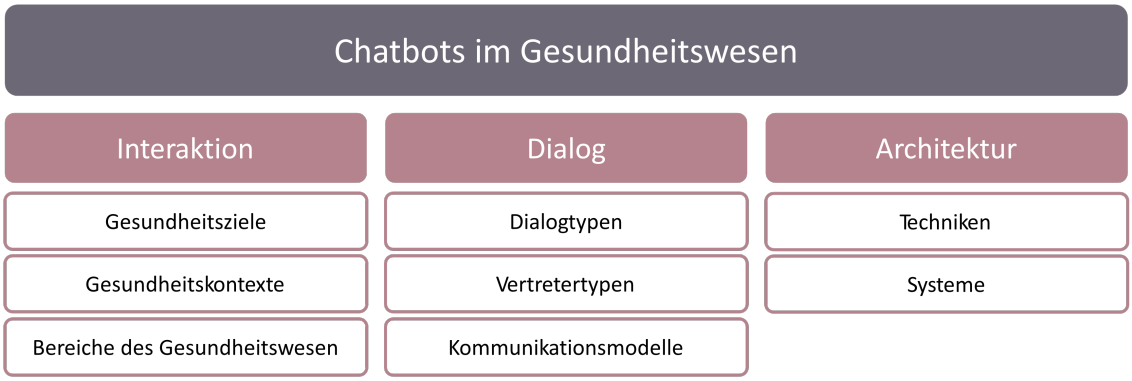


Abbildung 4.2: Taxonomie eines Chatbots im Gesundheitswesen nach [53]

In dem Paper „Chatbot breakthrough in the 2020s? An ethical reflection on the trend of automated consultations in health care“ [54] wird erforscht, wie Chatbots als Beratungssysteme Teile des Gesundheitswesens beeinflussen können. Dabei stellte sich heraus, dass der Einsatz von Chatbots die Rationalität und Automatisierung im Gesundheitswesen vorantreiben kann. Es wurde herausgefunden, dass Chatbots eine essenzielle Rolle für die Unterstützung und Motivation von Betroffenen, sowie für organisatorische Aufgaben spielen. Des Weiteren wurde herausgefunden, dass Chatbots vor allem einen Ersatz für nicht-medizinisches Pflegepersonal darstellen können. Da das Projekt der vorliegenden Arbeit zum Ziel hat, genau diese Mitarbeitende des Gesundheitswesens zu entlasten, kann daraus geschlossen werden, dass ein Chatbot hierfür die richtige Ansatzweise darstellt. Es soll möglich sein, der zu behandelnden Person schnell gewünschte Informationen zur Verfügung zu stellen. Je nach Anwendungsfall sollen diese Informationen der anwendenden Person bei den zuvor beschriebenen gesundheitlichen Problemen weiterhelfen und Klarheit vermitteln. Die Forschung der Verfassenden hat ergeben, dass dies mit einem Chatbot möglich ist und umgesetzt werden kann.

Um einen Chatbot auf der technischen Seite genauer zu beleuchten, wird das Paper „Contextual Chatbot for Healthcare Purposes (using Deep Learning)“ [55] näher betrachtet. Darin wird erforscht, wie ein kontextsensitiver Chatbot im Gesundheitswesen funktioniert und angewandt werden kann und welche zukünftigen Anwendungsbereiche für ihn vorgesehen werden könnten. Chatbots werden darin vorgesehen, um prädiktive Diagnosen auszustellen und um organisatorische Aufgaben zu verrichten, wie beispielsweise das Buchen von Terminen. Es wird deutlich gemacht,

dass Chatbots im Gesundheitswesen das Personal entlasten und diesen deshalb eine hohe Priorität in diesem Bereich zugeschrieben werden sollte. Die Forschenden beschreiben außerdem die Herausforderung, dass Chatbots oftmals falsche Entscheidungen treffen und deshalb ihre Architektur weiter verbessert werden sollte. Im Folgenden wird daher untersucht, ob ein Recommender System bei einem Chatbot auch in diesem Punkt zu Verbesserungen führen kann, beispielsweise durch die Möglichkeit direkt mit der anwendenden Person über den Chat in Kontakt zu treten und Rückfragen stellen zu können.

4.2 Health Recommender Systeme

Recommender Systeme haben sich vor allem in der Wirtschaft bereits durchgesetzt. Sie können aber auch im Bereich des Gesundheitswesens angewendet werden, um Prozesse zu beschleunigen oder die Lebensqualität und Gesundheit von Menschen und vor allem Mitarbeitenden im Gesundheitswesen zu verbessern.

Das Paper „DeepReco: Deep Learning Based Health Recommender System Using Collaborative Filtering“ [56] schlägt eine Umsetzung eines intelligenten HRS unter Verwendung der Restricted-Boltzmann-Machine-Convolutional-Neural-Network-Deep-Learning-Methode vor. Diese Methode zeigt auf, inwiefern Big-Data-Analysen für die Implementierung eines HRS genutzt werden können. In dem Paper hat sich herausgestellt, dass diese Methode weniger fehleranfällig ist als andere Ansätze. Zudem liefert die Methode der Forschenden eine höhere Genauigkeit der Antworten, als andere Methoden, mit welchen sie verglichen wurde.

4.3 Dialogorientierte Recommender Systeme

Zu dem Ansatz einen Chatbot mit einem Recommender System zu verbinden, gibt es bereits Forschungen, allerdings sind diese immer noch sehr begrenzt. Auf einen Ausschnitt der bereits vorhandenen Forschungen wird im Folgenden genauer eingegangen.

In dem Paper „Towards Conversational Recommender Systems“ [44] wird die Kombination dieser zweier Systeme genauer untersucht. Der gängige Begriff für diese Kombination lautet „Conversational Recommender System“. Das Ziel der Forschenden ist dabei, Recommender Systeme dem Menschen ähnlicher zu gestalten, da sich Personen meist anders verhalten, wenn sie Empfehlungen von anderen Personen erhalten. Zudem befasst sich die Arbeit mit dem schon zuvor in Abschnitt 3.3.3 behandelten Cold-Start-Problem. Dieses kann von Menschen meist schneller und besser bewältigt werden, als von einem Computer. Da in dieser Arbeit ein dialogorientiertes Recommender System umgesetzt wird, sind die folgenden Erkenntnisse wichtig für die Konzeption und Entwicklung eines dialogorientierten Recommender Systems. Es wurde herausgefunden, dass die beste Performance mit absoluten Fragen für das Feedback der nutzenden Person erreicht werden kann. Absolute Fragen für das Feedback beschreiben in Bezug auf Recommender Systeme eine Form von Feedback, bei welcher die nutzende Person nach ihrer Meinung zu einem bestimm-

ten Artikel gefragt wird, ohne diesen in Relation zu anderen Artikeln zu setzen. Die Rückmeldungen der Nutzenden werden daraufhin verwendet, um das System zu verbessern. Eine weitere Erkenntnis ist, dass die Strategie der Frageauswahl, um Feedback zu erhalten, entscheidend für ein effektives Lernen ist. In der Arbeit kamen die Forschenden zu dem Ergebnis, dass eine Banditen-Strategie die meisten Vorteile bei einem dialogbasierten Recommender System mit sich bringt. Dieser wurde allerdings nicht wie der klassische mehrarmige Bandit umgesetzt, welcher erst konvergiert, wenn alle Arme erforscht sind. Stattdessen ermöglicht eine kollaborative Struktur, wie sie hier verwendet wurde, eine schnellere Konvergenz.

Die Forschenden des Paper „Conversational Recommender System“ [57] haben sich zum Ziel gesetzt, diese zwei Systeme zu einem System zusammenzuführen, um sitzungsbasierte Nutzenfunktionen zu optimieren. Der Gesprächsverlauf wird als semistrukturierte Anfrage einer anwendenden Person dargestellt und analysiert. Diese Daten werden immer dann aktualisiert, wenn die nutzende Person mit dem dialogorientierten Recommender System kommuniziert oder es neue Informationen gibt, welche gesammelt werden können. Das Modell der Forschenden bezieht sowohl frühere Bewertungen der nutzenden Person als auch die Anfragen der aktuellen Sitzung in die Wahl der passenden Empfehlung mit ein. Auch das Belohnungssystem, welches von den Forschenden für das auf Reinforcement Learning basierenden dialogbasierten Recommender System eingesetzt wird, ist ein interessanter Ansatz für die in dieser Bachelorthesis gestellten Forschungsfragen. In der Arbeit der Autoren sind nur zwei Arten von Interaktionen möglich, die Abfrage eines Facettenwertes von einer nutzenden Person und die Abgabe von Empfehlungen an die anwendende Person. Die vorliegende Arbeit soll das dialogbasierte Recommender System konzeptionell erweitern, indem die nutzende Person proaktiv Fragen stellen kann. Hierzu wäre es auch möglich, dass das System aktiv Feedback von der anwendenden Person zu bestimmten Artikeln erfragen kann.

4.4 Dialogorientierte Health Recommender Systeme

In der vorliegenden Arbeit soll ein dialogbasiertes Recommender System für die spezifische Anwendung im Gesundheitswesen entworfen werden. Für diesen speziellen Anwendungsbereich müssen Dinge, wie beispielsweise der Umgang mit sensiblen Daten, anders gehandhabt werden. Die aktuelle Forschung in diesem Bereich ist noch sehr marginal, weshalb nur sehr wenig Forschung gefunden wurden, welche sich mit diesem Thema auseinandersetzt. Daraus wird erkennbar, dass hier noch ein großer Forschungsbedarf besteht, welcher durch diese Arbeit vorangetrieben werden soll.

Ein Paper, welches im Folgenden vorgestellt wird, beschäftigt sich bereits mit dialogbasierten HRS. Die Forschenden aus dem Paper „A conversational recommender system for diagnosis using fuzzy rules“ [58] nutzen ein dialogbasiertes Recommender System für medizinische Diagnosen, welches auf der Fuzzy-Logik beruht. Die Fuzzy-Logik kann Wissen über Krankheiten speichern und mit diesem durch die Gespräche mit Anwendenden eine Diagnose durchführen. Auch hier

wird mit Feedback der anwendenden Person gearbeitet, um bessere Ergebnisse bei den Empfehlungen zu erzielen. Das Modell arbeitet bisher noch nicht mit kollaborativer Filterung, es wird aber im Ausblick des Papers darauf hingewiesen, dass in diese Richtung weiter geforscht werden könnte.

In Kapitel 6 wird versucht, ein dialogbasiertes HRS zu implementieren, welches diesem Modell ähnlich ist. Die Taxonomie des zu entwickelnden Chatbots im Gesundheitswesen dieser Arbeit stützt sich dabei auf die Ergebnisse des Literaturvergleichs von [53].

5 Konzeption

Die Konzeption legt den Grundstein dafür, wie der spätere Prototyp aussehen soll. Zunächst werden dafür die grundlegenden Herausforderungen eines Recommender Systems analysiert. Um ein Recommender System zu entwickeln und dieses anschließend in den vorhandenen Chatbot von Pulsnetz zu integrieren, ist es zudem wichtig, zuvor die Anforderungen und Zielsetzungen zu definieren. Damit wird sichergestellt, dass die Implementierung im Rahmen des Möglichen liegt und auch im zeitlichen Rahmen dieser Arbeit umsetzbar ist. Zu Beginn werden einige Herausforderungen eines dialogorientierten Recommender Systems im Gesundheitswesen definiert. Um ein strukturiertes Vorgehen zu garantieren, wird eine Anforderungsanalyse durchgeführt. Hierfür werden zuerst verschiedene Personas erstellt, welche sich an den schon vorhandenen Personas des Projekts „Pulsnetz“ orientieren. Mithilfe dieser Personas werden Use Cases aufgestellt, welche analysiert und evaluiert werden. Damit werden anschließend die finalen Anforderungen an das Recommender System definiert. Nach Abschluss der Anforderungsanalyse wird aus dieser das konzeptionelle Design des Recommender Systems festgelegt und ausgearbeitet. Letztlich werden noch die Grenzen der Umsetzung festgelegt, welche für die Implementierung des Prototyps benötigt werden.

5.1 Herausforderungen eines Recommender Systems

Ein Recommender System bringt viele Herausforderungen mit sich, welche im Folgenden behandelt und analysiert werden sollen. Dies dient dazu, einen Überblick zu behalten und besser mit den Herausforderungen umgehen zu können. Zuerst werden zwei allgemeine Herausforderungen von Recommender Systemen aufgegriffen. Zum einen das Cold-Start-Problem bei Recommender Systemen und zum anderen die begrenzten Möglichkeiten der Evaluation von Recommender Systemen. Letztlich wird noch Bezug auf den Anwendungsfall des umzusetzenden Recommender System genommen. Da der Prototyp und auch das Konzept auf das Gesundheitswesen ausgerichtet sind, muss auch ein Bezug zu dem Problem der Verarbeitung von sensiblen Daten im Gesundheitswesen hergestellt werden. Es gibt noch zahlreiche weitere Herausforderungen, welche Recommender Systeme mit sich bringen, hierzu zählt unter anderem die Serendipität, die Skalierbarkeit und das Problem der Überspezialisierung. [4, 34] Auf diese wird im Folgenden nicht genauer eingegangen, da es den zeitlichen Rahmen der vorliegenden Arbeit überschreiten würde.

5.1.1 Cold-Start-Problem eines Recommender Systems

Ein fundamentales Problem, welches bei Recommender Systemen auftritt, befasst sich damit, wie eine Empfehlung für neue anwendende Personen oder neue Items ausgesprochen werden soll. Das

System hat in diesem Szenario noch keine Kontextdaten, um Empfehlungen auszusprechen. [59] Im Fall, dass die anwendende Person bisher unbekannt war, fehlt das Wissen, was die Person interessieren könnte, was ihr gefallen könnte und womit sie sich oft befasst. Im Fall eines neuen Items hingegen ist unbekannt, wie gut das Item bei anderen anwendenden Personen bewertet wurde und für welche Interessen es empfohlen werden soll. [60] Für den ersten Fall werden deshalb oft Items empfohlen, welche bei vielen anderen nutzenden Personen eine hohe Popularität aufgewiesen haben. Es gibt jedoch auch den Ansatz aus dem Paper [49], welcher aufzeigt, dass die Präferenzen von neuen Nutzenden eher auf weniger populäre Items ausgerichtet sind. Es gibt auch den Ansatz, soziale Netzwerke zur Entschärfung von dem Cold-Start-Problem zu nutzen. Zu diesem gibt es zahlreiche Literatur, welche in dem Paper „Social network data to alleviate cold-start in recommender system: A systematic review“ [61] verglichen und aufgearbeitet wurde. Grundsätzlich lässt sich feststellen, dass es zu dem momentanen Zeitpunkt noch keine allgemeingültige Lösung für dieses Problem gibt. Dies hängt mit zwei fundamentalen Fragen, welche essenziell für Recommender Systeme sind, zusammen. Diese sind komplex zu beantworten, da bei diesen Fragen festgestellt werden muss, was der nutzenden Person gefällt und wieso es ihr gefällt. [62] Um die Fragen beantworten zu können, bedarf es einer angemessenen Evaluation der Empfehlungen. Auf diese Herausforderung wird im nächsten Abschnitt Bezug genommen.

5.1.2 Evaluation von Recommender Systemen

Wie zuvor beschrieben hängt die Relevanz einer Empfehlung von den Präferenzen, den Intentionen und dem Kontext der nutzenden Person ab. Damit kann keine allgemeingültige Antwort auf die Frage gefunden werden, ob eine Empfehlung für die nutzende Person von Relevanz ist. Dies macht die Evaluation schwieriger, da die Empfehlungen unüberwacht sind und somit nicht bekannt ist, ob die Empfehlung „richtig“ oder „falsch“ ist. Die Forschung im Bereich der Evaluation von Recommender Systemen ist zu marginal, um eine allgemeingültige Antwort darauf finden zu können. [62] Es steht fest, dass die Vorhersage der Vorlieben und Interessen einer Person nicht ausreicht, um eine passende Empfehlung auszusprechen. Zusätzlich sollte die Menge der Eigenschaften identifiziert werden, die den Erfolg im Kontext beeinflussen. Dies ist herausfordernd, da zuerst die Menge der relevanten Eigenschaften identifiziert werden muss und diese dann mit in die Evaluation einbezogen werden sollten. [63] Bei Recommender Systemen wird zwischen drei Arten der Evaluation unterschieden. Die Offline-Evaluation, die Online-Evaluation und die Studie mit Anwendenden. Die simpelste Option ist dabei die Offline-Evaluation, bei dieser wird zuvor ein Datensatz mit hypothetischen Nutzenden angelegt. Damit wird ihr Verhalten mit dem Recommender System simuliert. Es muss dafür davon ausgegangen werden, dass reale Nutzende sich simultan zu den hypothetischen Nutzenden verhalten. [63, 64] Die Online-Evaluation verhält sich ähnlich zur Offline-Evaluation. Ihr Vorteil ist jedoch, dass Nutzende in Echtzeit evaluiert werden können. Dafür muss das Recommender System allerdings schon veröffentlicht sein. Die letzte Option ist einer Studie mit Anwendenden. Diese ist meist sehr aufwändig, da eine realistische Simulation für die Probanden erzeugt werden muss, in welcher die Evaluation stattfindet. Zudem wird eine große Menge an Probanden benötigt, damit die Evaluation aussagekräftig ist. [63]

5.1.3 Sensible Daten im Gesundheitssektor

Zuletzt stellt der Anwendungsfall auf den Gesundheitssektor noch eine Herausforderung bezüglich des Umgangs mit sensiblen Daten dar. Optimierte Empfehlungen benötigen möglichst viele Kontextdaten, auf Basis derer die Empfehlung ausgesprochen werden kann. Dazu werden vor allem im Gesundheitssektor allerdings auch sensible Daten benötigt, um genaue Empfehlungen aussprechen zu können. [65] Sensible Daten beschreiben in der vorliegenden Arbeit alle Daten, welche Bezug auf den Gesundheitszustand der anwendenden Person nehmen, sowie deren demografische Daten, wie beispielsweise das Alter, der Wohnort und das Geschlecht. Der Kompromiss zwischen Privatsphäre und Personalisierung wirft im Gesundheitssektor ethische Bedenken auf. Zunächst herrscht zwischen der anwendenden Person und dem Recommender System nicht die gleiche Vertrauensbasis, wie zwischen der anwendenden Person und einem Gesundheitsdienstleister. Dies kann dazu führen, dass die anwendende Person nicht bereit ist, vertrauliche Daten mit dem System zu teilen. Eine weitere Herausforderung stellt auch die Rechtslage dar, welche vor allem bei Gesundheitsdaten genau geprüft werden sollte. [65, 66] Konzeptionell wird deshalb nicht genauer auf die Frage des Datenschutzes eingegangen, bei einer Umsetzung und Veröffentlichung eines Recommender System im Gesundheitswesen sollten diese Fragen jedoch erneut aufgearbeitet und geklärt werden. Da dies den Umfang der vorliegenden Arbeit überschreiten würde, wird hierauf nicht weiter Bezug genommen.

5.2 Anforderungsanalyse für ein Recommender System

Um ein konzeptionelles Design eines Recommender Systems aufzubauen, wird im Folgenden eine Anforderungsanalyse durchgeführt. Diese baut auf erstellten Personas auf, aus welchen Use Cases gebildet wurden. Anhand dieser Use Cases werden dann funktionale und nicht-funktionale Anforderungen definiert.

5.2.1 Personas

Um die Zielgruppe des Recommender Systems genauer zu spezifizieren, wird die Persona-Methode verwendet. Dabei wird versucht, aus den Daten der Zielgruppen fiktive Personen zu formen. [67] Für diesen Prozess wird zwischen realen und realistischen Personas unterschieden. Um reale Personas umzusetzen, müssen Untersuchungen an der Zielgruppe vorgenommen werden, aus welchen qualitative und quantitative Daten für die Personas gewonnen werden können. Realistische Personas hingegen können aus Gesprächen zwischen Projektteilnehmenden entstehen. [68] Die vorliegenden Personas fallen unter die Kategorie der realistischen Personas, da die genutzten Informationen zur Erstellung hauptsächlich aus schon vorhandenen Informationen des Projekts „Pulsnetz“, sowie aus Befragungen von Projektpartnern stammen. Aus diesen Informationen sind drei Personas entstanden, die nach Berufsfeldern klassifiziert wurden (siehe Anhang A.1). Jede Persona repräsentiert dabei eine bestimmte Gruppe nutzender Personen. Diese Gruppen umfassen Pflegekräfte, Sozialarbeitende und Pflegedienstleitungen. Diese Einteilung in nutzende Gruppen wurde aus den Unterlagen des Projekts „Pulsnetz“ übernommen. Die entwickelten Personas sind zudem

auf die Lebensstile in [69] und die Internet-Milieus in [70] bezogen. Bei den Lebensstilen in [69] werden Menschen in prototypische Gruppen unterteilt, welche die Gesellschaft anhand von Einstellungen, Motiven und Vorlieben abbilden sollen. Diese waren bei der Erstellung der Personas vor allem hilfreich, um einen ersten Rahmen der Lebenseinstellung der Persona festzulegen. Um diesen zu verfeinern, wurden nachfolgend die Internet-Milieus miteinbezogen. Diese unterteilen die Personen einer Gesellschaft in verschiedene Gruppen. Die Unterteilung findet dabei nach der sozialen Lage, nach der Grundorientierung und nach der Haltung gegenüber dem Internet statt. Diese Gruppenzuweisung war vor allem hilfreich, um die Technikaffinität der Personas einschätzen zu können. Im Folgenden wurde deshalb jeder Persona ein Internet-Milieu und ein Lebensstil zugeordnet. Nachfolgend wird auf die einzelnen Personas genauer eingegangen und es werden die wichtigsten Merkmale, Werte und Einstellungen in Bezug auf das Projekt „Pulsnetz“ aufgegriffen.

Pflegekraft Annabell

Annabell ist 26 Jahre alt, sie hat die Realschule absolviert und danach eine Ausbildung zur Pflegekraft gemacht. Die größten beruflichen Herausforderungen stellen für sie die berufliche Weiterentwicklung, die Problemlösung, die Entscheidungsfindung und das Zeitmanagement dar. Für ihren Beruf benötigt sie neben medizinischer Kompetenz auch Durchhaltevermögen, Einfühlsamkeit und Zuverlässigkeit. Bei den Internet-Milieus kann sie den „unbekümmerten Hedonisten“ zugeordnet werden. Diese nutzen die angebotenen Möglichkeiten online ausgiebig und sind vor allem von Social Media begeistert. Sie kümmern sich weniger um Datenschutz und haben im Allgemeinen einen eher lockeren Umgang mit dem Internet. Dies begründet sich darin, da sie, trotz ausgiebiger Nutzung, unsicher im Umgang mit Technik sind. [70] Die Einordnung dieser Persona in einen Lebensstil gestaltete sich etwas komplizierter. Annabell stellt hier eine Mischung aus dem Lebensstil „Digital Creative“ und dem Lebensstil „Forever Youngster“ dar. Der „Digital Creative“ stellt einen Lebensstil dar, bei welchem sich die Welt hauptsächlich online abspielt. Personen mit dem Lebensstil „Digital Creative“ sind gerne vernetzt und in Gesellschaft. All dies trifft auch auf Annabell zu. Allerdings ist das Knowhow im Bereich der Technik bei einem „Digital Creative“ meist höher als bei Annabell. Der Lebensstil „Forever Youngster“ hingegen setzt den Fokus vor allem auf das Thema Gesundheit. Menschen mit diesem Lebensstil ist es besonders wichtig, sich gesund zu ernähren und viel Sport zu treiben. Sie informieren sich zudem in den Medien regelmäßig über Gesundheitsthemen, diese können sowohl psychische als auch physische Gesundheitsinformationen umfassen. [69]

Sozialarbeiter Markus

Markus ist 55 Jahre alt und hat sein Diplom in einem sozialwissenschaftlichen Studiengang absolviert. Danach ist er als Sozialarbeiter in der Kinder- und Jugendarbeit eingestiegen, wo er auch bis heute geblieben ist. Beruflich stellen vor allem Veränderung und Zeitmanagement Herausforderungen für Markus dar. Er benötigt in seinem Beruf eine hohe Frustrationstoleranz, Kommunikationsfähigkeit, sowie Fachkompetenz. Bei den Internet-Milieus fällt er unter die „vorsichtigen Skeptiker“. Diese sind meist überfordert im Umgang mit dem Internet und meiden es deshalb.

Sie haben starke Bedenken in Bezug zur Digitalisierung und wollen möglichst wenig persönliche Daten im Internet preisgeben. Sie nutzen das Internet zwar für gewohnte Prozesse, jedoch stehen sie digitalen Neuerungen kritisch gegenüber. [70] So auch Markus, dieser nutzt im beruflichen sporadisch seinen Computer mit für ihn gewohnter Software. Vom Lebensstil lässt sich Markus als „Neo-Biedermeier“ einstufen. Diese Gruppe strebt nach Sicherheit in allen Lebensbereichen, vor allem für ihre Familie. Zudem fühlen sie sich unsicher im Umgang mit Neuem und legen Wert auf Routinen. [69]

Pflegedienstleitung Chiara

Chiara ist 45 Jahre alt und hat eine abgeschlossene Berufsausbildung zur Pflegekraft absolviert. Durch diverse Weiterbildungen ist sie inzwischen Pflegedienstleitung. Vor allem mangelnde Ressourcen und die Motivation ihrer Mitarbeitenden stellen sie vor Herausforderungen. Ein großer Teil ihres Berufs ist das Personalmanagement, wofür sie eine ausgeprägte Kommunikationsfähigkeit und Durchsetzungsvermögen benötigt. In den Internet-Milieus in [70] fällt sie unter die „verantwortungsbedachten Etablierten“. Diese zeichnen sich durch einen besonnen und abwägenden Umgang mit dem Internet aus. Sie finden Digitalisierung wichtig, stehen ihr aber nicht euphorisch gegenüber. Sie nutzen das Internet weniger zur Unterhaltung, sondern mehr zur Informationsgewinnung. In den Lebensstilen lässt sich Chiara dem „Golden Mentor“ zuordnen. Personen mit diesem Lebensstil legen viel Wert darauf, informiert zu sein und nutzen hierfür auch verschiedene Kanäle. Sie sind diszipliniert und ihre Selbstständigkeit ist ihnen sehr wichtig. Trotz dessen legen sie viel Wert auf ein Familienleben. Ihre Mediennutzung ist zwar sehr hoch, jedoch ist die Konsumfreude bei dieser Gruppe eher niedrig ausgeprägt. Sie nutzen zudem viele klassische Kanäle, wie die Zeitung und das Radio. Das Internet wird von ihnen ausschließlich zur Informationsgewinnung genutzt. [69]

5.2.2 Mögliche Kontextdaten

Um die Use Cases zu definieren, müssen zuvor die verschiedenen vorhandenen Kontextdaten festgelegt werden. Mögliche Kontextdaten sind:

A: Items Die Items beschreiben in diesem Fall die Dokumente, welche später als Empfehlung ausgesprochen werden sollen.

B: Aktuelle Frage Die aktuelle Frage der anwendenden Person muss vorhanden sein, um nachfolgend eine Empfehlung aussprechen zu können.

C: Vorheriger Chatverlauf Sollte die anwendende Person zuvor mit dem Chatbot interagiert haben, kann der dabei entstandene Chatverlauf miteinbezogen werden. Sollte die anwendende Person ein Profil erstellt haben, wird der Chatverlauf darin gespeichert.

D: Bewertung der Items Sollten zuvor schon Empfehlungen an die nutzende Person ausgesprochen worden sein, können diese von der nutzenden Person bewertet werden. Diese Bewertungen können somit auch als Kontextdaten miteinbezogen werden.

E: Präferenzen Die Voraussetzung dafür, dass Präferenzen der anwendenden Person vorhanden sind, ist, dass diese sich ein Profil erstellt und bei der Erstellung von diesem ihre Präferenzen hinterlegt hat.

F: Demografische Daten Die demografischen Daten können ebenfalls von der anwendenden Person bei Erstellung eines Profils angegeben werden. Unter demografische Daten fallen das Alter, der Wohnort, das Geschlecht und der Beruf der anwendenden Person. Sollte die anwendende Person kein Profil erstellt haben, besteht die Möglichkeit demografische Daten aus dem Chat zu extrahieren und als Kontextdaten zu nutzen.

G: Kontextdaten anderer Nutzenden Alle bisher aufgezählten Kontextdaten können auch von anderen Nutzenden vorliegen. In diesem Fall können die Daten von anderen Nutzenden in die Wahl der Empfehlung miteinbezogen werden. Dies kann beispielsweise aufgrund von Ähnlichkeiten zwischen zwei Personen stattfinden.

5.2.3 Use Cases

Aus den obigen Personas und möglichen Kontextdaten lassen sich im Folgenden verschiedene Use Cases ableiten. Diese unterscheiden sich anhand der möglichen Kontextdaten, die dem Recommender System zur Verfügung stehen. Use Cases sind vor allem von Relevanz, um ein Bewusstsein für die Anforderungen und Funktionalitäten an das spätere System zu entwickeln. Aufgrund der vielfältigen Kombinationen von Kontextdaten gibt es sehr viele verschiedene Ausgangslagen. Ein detailliertes Diagramm mit allen möglichen Use Cases befindet sich deshalb im Anhang. (siehe Anhang A.2)

Aus diesen Use Cases werden drei Fälle ausgewählt, auf welche im Folgenden näher eingegangen wird. Es wird versucht, damit die Randfälle abzudecken. Es wird ein Fall berücksichtigt, in welchem keine Kontextdaten außer die Items vorhanden sind und ein Fall, in dem alle möglichen zuvor definierten Kontextdaten zur Verfügung stehen. Zudem wird auf ein Use Case genauer eingegangen, bei welchem nur bestimmte Kontextdaten zur Verfügung stehen. Die folgende Tabelle 5.1 verschafft einen Überblick darüber, welche Kontextdaten bei welchem Use Case zur Verfügung stehen. Die verschiedenen Kontextdaten sind dabei alphabetisch nach der Liste aus Abschnitt 5.2.2 nummeriert.

	A	B	C	D	E	F	G
Keine zusätzlichen Kontextdaten	X	X					
Bestimmte zusätzliche Kontextdaten	X	X	X	X			
Alle Kontextdaten	X	X	X	X	X	X	X

Tabelle 5.1: Vorhandene Kontextdaten der verschiedenen Use Cases

Keine zusätzlichen Kontextdaten

Der erste Use Case, welcher genauer betrachtet wird, beschreibt das Szenario, in welchem keine zusätzlichen Kontextdaten der nutzenden Person vorliegen. Die nutzende Person stellt hierbei eine Frage an den Chatbot, diesem stehen anfänglich somit nur die Items und die Frage zur Verfügung. Die nutzende Person hat kein Profil angelegt, weswegen Präferenzen und demografische Daten zunächst nicht berücksichtigt werden können. Da die nutzende Person zuvor noch nicht mit dem Chatbot interagiert hat, liegen auch keine vorherigen Chatverläufe vor. Es besteht eventuell die Möglichkeit, demografische Daten aus der gestellten Frage zu ziehen.

Das hier beschriebene Szenario orientiert sich an der Persona von Markus, da dieser ein typischer Kandidat für einen solchen Ablauf wäre. Dies begründet sich damit, dass er dem Internet eher skeptisch gegenübersteht und neue Anwendungen nicht direkt ausprobieren möchte. Somit wird er sich wahrscheinlich kein eigenes Profil erstellen, in welchem er Daten von sich preisgibt. Dieses Szenario beschreibt das schon im Vorfeld aufgegriffene Cold-Start-Problem. Hier soll versucht werden, mit den vorhandenen Daten so gut wie möglich zu arbeiten. Der simpelste Schritt hierfür wäre es, anhand der gestellten Frage Items zu suchen und drei von diesen zusätzlich an die anwendende Person auszuspielen. Dies könnte beispielsweise mit einer Text Similarity umgesetzt werden. Um die Ergebnisse präziser zu gestalten, bestünde die Möglichkeit, Rückfragen an die anwendende Person zu stellen. Dieser Vorteil, welcher sich durch die Kombination des Recommender Systems mit dem Chatbot bietet, sollte vor allem in Fällen mit wenigen Daten ausgenutzt werden. Durch diese Rückfragen lässt sich beispielsweise das Alter der nutzenden Person ermitteln, wodurch neue Möglichkeiten gewonnen werden können, um Empfehlungen auszusprechen. Dann könnten Items empfohlen werden, welche für das spezifische Alter eher geeignet sind.

Bestimmte zusätzliche Kontextdaten

Dieses Szenario beschreibt den Mittelweg zwischen den zwei Extremen. In diesem Use Case werden zwar Daten von der nutzenden Person zur Verfügung gestellt, jedoch geht sie streng mit demografischen Daten um. Es besteht eine solide Datenbasis, welche für Empfehlungen genutzt werden kann. Wie zuvor bei der skeptischen nutzenden Person stehen die Items und die einhergehende Frage zur Verfügung und können verarbeitet werden. Die Wahrscheinlichkeit in diesem Szenario aus dem Chatverlauf demografische Kontextdaten extrahieren zu können, ist größer, da hierfür auch der vorherige Chatverlauf genutzt werden kann. Zusätzlich wird in diesem Szenario die Feedback-Funktion genutzt, durch welche sich Chatverläufe gewichten lassen.

Dieser Use Case basiert auf der Persona von Chiara, welche wahrscheinlich ähnlich eines solchen Use Cases handeln würde. Sie steht neuen Anwendungen und Funktionen grundlegend offen gegenüber und beschäftigt sich mit ihnen. Hierunter fallen beispielsweise die Feedback-Funktion und das Anlegen eines Profils für die anwendende Person. Wenn es um sensible Daten geht, welche sie selbst angeben kann, reagiert sie allerdings zurückhaltender, weswegen ihr Profil anfangs relativ wenig Aussagekraft besitzt.

Alle möglichen Kontextdaten

In diesem Szenario sind alle zuvor definierten möglichen Kontextdaten vorhanden. Es existieren, wie bei jedem der Szenarien die Items und die aktuelle Frage im Chatverlauf. Zudem bestehen auch vorherige Chatverläufe, welche teilweise durch Feedback gewichtet sind. Zusätzlich ist ein Profil vorhanden, welches bei der Erstellung mit Kontextdaten zu den persönlichen Präferenzen und demografischen Daten befüllt wurde. Somit bietet dieses Szenario die am breitesten aufgestellte Datenbasis.

Das hier beschriebene Szenario basiert auf der Persona von Annabell. Sie würde sich ähnlich der im Szenario beschriebenen anwendenden Person verhalten, da sie offen für neue Anwendungen und Funktionen ist und kein Problem damit hat sensible Daten von sich preiszugeben. Demnach würde sie alle Daten, die für eine breit aufgestellte Empfehlung nötig sind, zur Verfügung stellen. Auch in diesem Szenario können alle Methoden und Daten der zuvor beschriebenen Use Cases miteinbezogen werden. Zusätzlich können die Empfehlungen durch Präferenzen und demografische Daten, wie beispielsweise das Alter, beeinflusst werden. Eine weitere Möglichkeit, welche in Betracht gezogen werden kann, wäre es, die Daten anderer Nutzenden miteinzubeziehen. Somit könnten Nutzende miteinander verglichen werden und es können der anwendenden Person Items empfohlen werden, welche von einer ihr ähnlichen Person gut bewertet wurden. Die Grundlage für die Umsetzung der kollaborativen Filterung stellt eine breit gefächerte Datenbasis dar.

5.2.4 Basisanforderungen an das Recommender System

Mithilfe der Use Cases können im Folgenden Anforderungen an das Recommender System aufgestellt werden. Diese Anforderungen werden in funktionale und nicht-funktionale Anforderungen unterschieden. Funktionale Anforderungen legen dabei fest, welche Aufgaben mit der Anwendung realisiert werden sollen. Die Qualität der Lösung der Aufgabe spielt dabei keine Rolle. Nichtfunktionale Anforderungen hingegen stellen Anforderungen an die Qualität dieses Vorgehen.

Funktionale Anforderungen

Im Nachfolgenden werden die funktionalen Anforderungen an das Recommender System definiert und festgelegt.

F1 - Grundlage der Empfehlung Es sollen Empfehlungen für die anwendende Person auf Basis von vorhandenen Kontextdaten ausgesprochen werden.

F2 - Zielsetzung der Empfehlung Die Empfehlung soll der anwendenden Person weitere Themeneinblicke aufzeigen. Sie kann sich so weiterbilden und andere nützliche Informationen in Erfahrung bringen.

F3 - Zeitpunkt der Empfehlung Es soll immer dann eine Empfehlung ausgesprochen werden, sobald genug Kontextdaten vorhanden sind, um eine sinnvolle Empfehlung auszusprechen.

F4 - Kontextsensitivität Das Recommender System kann alle Kontextdaten, welche vorhanden sind, mit in die Entscheidung der Empfehlungsgebung einbeziehen und damit auch alle Use Cases abbilden.

F5 - Empfehlungsmenge Das Recommender System soll bei jeder Empfehlung drei Dokumente an die anwendende Person ausspielen.

F6 - Feedback Das Recommender System soll die Möglichkeit bieten, die ausgespielten Empfehlungen zu bewerten. Diese Rückmeldung kann daraufhin in die folgenden Empfehlungen miteinbezogen werden.

Nicht-Funktionale Anforderungen

Im Folgenden werden die nicht-funktionalen Anforderungen an das Recommender System aufgelistet.

N1 - Robustheit Das Recommender System sollte stabil sein und keine Ausfälle bei einer großen Menge an Daten aufweisen.

N2 - Geschwindigkeit Das Recommender System sollte eine schnelle Antwortzeit aufweisen, da die nutzende Person sonst nicht mehr damit rechnet, dass sie noch weitere Empfehlungen erhält. Die Antwortzeit sollte dabei unter 10 Sekunden liegen.

N3 - Konsistenz Die Empfehlung des Recommender System sollte thematisch zu den Kontextdaten der anwendenden Person passen.

N4 - Generisch Das Recommender System sollte auch für andere Themen funktionieren, wenn die Items entsprechend angepasst werden.

Abgrenzung

Die grundlegende Funktionsweise des zu entwickelnden Prototypen wird durch F1 beschrieben. Das Ziel des Recommender Systems wird dabei durch F2 beschrieben. F4 ist als Schwerpunkt der vorliegenden Arbeit aufzufassen, während F3, F5 und F6 Anforderungen an die Ein- und Ausgabe abbilden. Die nicht-funktionalen Anforderungen sind so definiert, dass sie ein Mindestmaß an Robustheit und Geschwindigkeit garantieren. Der Fokus bei den nicht-funktionalen Anforderungen liegt allerdings auf N3, da diese Anforderung grundlegend für das Recommender System ist. Da die Arbeit nur einen begrenzten Zeitraum umfasst, werden die Anforderungen an die Sicherheit und das Design des User Interface (UI) vernachlässigt und aufgrund dessen auch nicht näher beschrieben.

5.3 Konzeptionelles Design des Systems

Der folgende Abschnitt soll ein konzeptionelles Design eines Recommender System im Gesundheitswesen aufzeigen. Um dieses aufzustellen wurden die Personas, Use Cases und Anforderun-

gen miteinbezogen. Zunächst wird das Recommender System konzeptionell aufgearbeitet und im Anschluss wird noch kurz Bezug auf eine mögliche Darstellung in der Oberfläche genommen.

5.3.1 Konzeptionelles Design des Recommender Systems

Im Folgenden wird eine konzeptionelle Architektur eines Recommender System für Chatbots im Gesundheitswesen aufgezeigt. Zunächst wird festgelegt, welche Art von Recommender System hierfür am sinnvollsten wäre. Es ist aufgrund der verschiedenen Kontextdaten, welche vorhanden sein können, sinnvoll, ein hybrides Recommender System zu konzipieren. Dabei wird spezifisch ein gewichtetes hybrides Recommender System gewählt, sodass die Erkenntnisse aller Kontextdaten, die vorhanden sind, mit unterschiedlichen Gewichten versehen werden und aus der Gesamtheit dieser eine Empfehlung entstehen kann. Dabei werden immer die Kontextdaten, welche schon vorhanden sind, miteinbezogen. Sollten zu wenig Kontextdaten vorhanden sein, soll der Chatbot eine Rückfrage an die anwendende Person stellen und sie zu ihrem Beruf, Alter und Geschlecht befragen, um Informationen zu sammeln. Das vorliegende Konzept wird zunächst ohne kollaborative Filtering aufgestellt, da dieses verschiedenste neue Möglichkeiten aufbringt, auf welche aufgrund des begrenzten Rahmens der vorliegenden Arbeit nicht spezifisch genug eingegangen werden könnte. Die folgende Abbildung 5.1 zeigt die konzeptionelle Architektur auf und verdeutlicht, welche Kontextdaten der anwendenden Person miteinbezogen werden sollen.

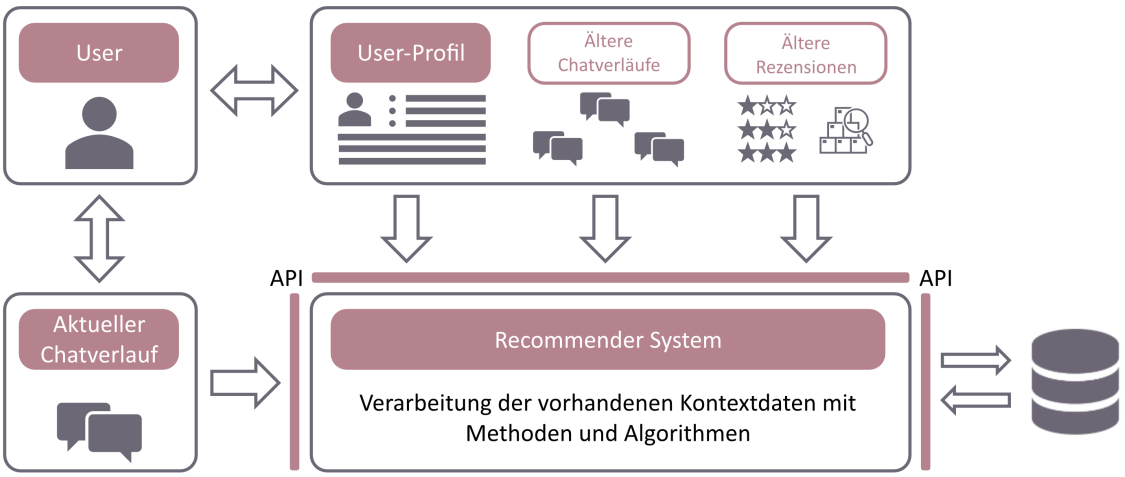


Abbildung 5.1: Architektur des konzeptionellen Recommender Systems

In der folgenden Konzeption wird davon ausgegangen, dass alle zuvor definierten Kontextdaten der anwendenden Person vorhanden sind. Somit erstellt sich die anwendende Person bei der ersten Nutzung des Systems ein Konto, in welchem sie ihre Präferenzen und demografischen Daten hinterlegt. In diesem werden ab diesem Zeitpunkt alle Chatverläufe gespeichert. Zudem werden darin die schon bewerteten Items, welche der anwendenden Person empfohlen wurden, gespeichert. Sollte die anwendende Person dann eine Frage an den Chatbot stellen, wird diese Frage zusammen mit den Kontextinformationen an das Recommender System geschickt. Die Items liegen in einer Datenbank, auf welche das Recommender System zugreift, sobald eine Anfrage gesendet wird. Wichtig hierbei ist, dass die Items zusätzlich zu ihrem Textinhalt auch Metadaten benötigen.

Diese Metadaten sollten sich möglichst mit den Präferenzen und den demografischen Daten der anwendenden Person decken. Für diesen Anwendungsfall sollten die Items vorrangig die Informationen zu den darin behandelten Themen enthalten. Zusätzlich wäre es hilfreich, wenn den Items Informationen zu den bevorzugt interessierten Berufs-, Alters-, Geschlechts- und Orts-Gruppen beigefügt sind. Abbildung 5.2 zeigt auf, welche Kontextdaten der anwendenden Person, mit welchen Meta-Informationen der Items verglichen werden.

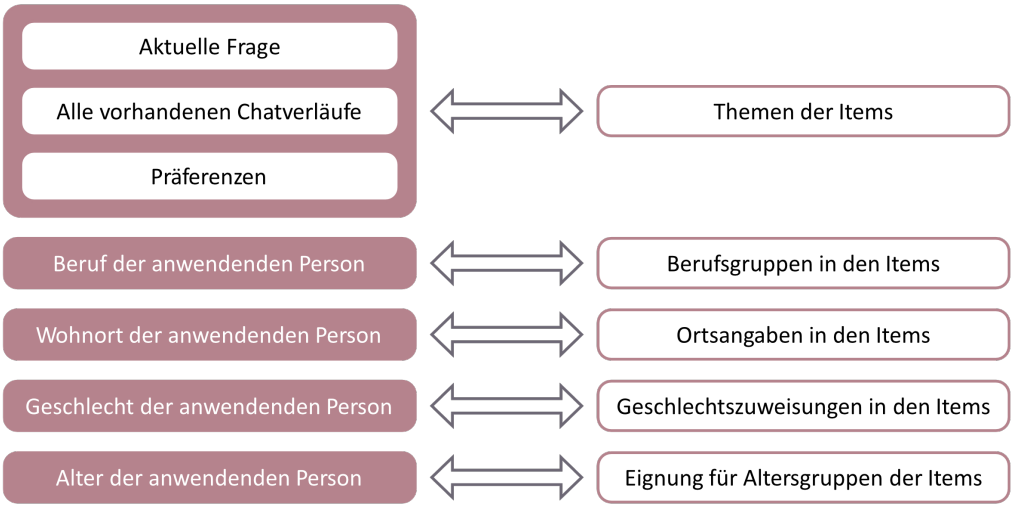


Abbildung 5.2: Vergleich der Kontextdaten zwischen Person und Items

Diese Vergleiche können Ergebnisse liefern, welche durch eine festzulegende Gewichtung kombiniert werden. Letztlich werden bereits empfohlene Items, welche von der anwendenden Person bewertet wurden, miteinbezogen. Beispielsweise können die Themen und Meta-Informationen von gut bewerteten Items für die neue auszusprechende Empfehlung höher gewichtet werden.

5.3.2 Konzeptionelle Darstellung in der Oberfläche

Die folgende Abbildung 5.3 zeigt auf, wie die Empfehlungen an die nutzende Person zurückgegeben werden könnten.

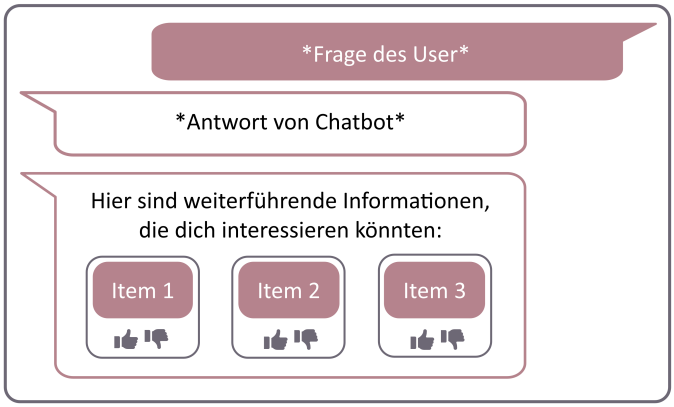


Abbildung 5.3: Integration des Recommender Systems in den Chatbot

Dabei werden drei Items in einer Chat-Nachricht ausgespielt und mit einem Text versehen, der die anwendende Person darauf hinweisen soll, dass es sich hierbei um weiterführende Empfehlungen handelt und was die Thematik der jeweiligen Empfehlung ist. Jedes der drei Items kann dann individuell von der anwendenden Person mit einer Bewertung versehen werden, ob sie das Item hilfreich fand. Zusätzlich kann getrackt werden, welche der drei Items angeklickt und geöffnet wurden. Dies hat wiederum eine Aussagekraft, wie nützlich die anwendende Person das Item fand. Vor allem die Feedbackfunktion für das Recommender System muss in der Oberfläche des Chatbots angelegt werden, sollte diese umgesetzt werden.

5.4 Limitationen

Im Folgenden wird festgelegt, welche Teile der konzeptionellen Architektur des Recommender System in dem Kapitel 6 prototypisch umgesetzt werden sollen. Aufgrund des begrenzten Rahmens der vorliegenden Arbeit, werden zunächst die grundlegenden Funktionalitäten des Recommender System umgesetzt, um die in Abschnitt 5.2.4 festgelegten Anforderungen zu erfüllen. Das prototypische Recommender System sollte die in den Items behandelten Themen mit der einhergehenden Frage der anwendenden Person und dem vorherigen Chatverlauf abgleichen können, sodass ein inhaltsbasiertes Recommender System garantiert werden kann. Um ein gewichtetes hybrides Recommender System zu bauen, werden zusätzlich, falls von der anwendenden Person preisgegeben, noch Berufe miteinbezogen, welche aus dem Chatverlauf extrahiert werden können. Ein Profil der nutzenden Person wird in dem Prototyp nicht umgesetzt, weshalb auch keine Präferenzen erfasst werden können. Auch die Bewertung schon empfohlener Items wird nicht mitberücksichtigt. Das prototypische Recommender System wird jedoch so gebaut, dass der Vergleich anderer Kontextdaten und Meta-Informationen durch wenige Schritte ergänzt werden kann. Weiterführend wird zunächst auch die Funktionalität, welche es dem Chatbot ermöglichen würde, Rückfragen an die anwendende Person zu stellen, außen vor gelassen, da diese sehr viele Korrekturen an dem Chatbot mit sich bringen würde.

6 Prototypische Implementierung eines Recommender Systems

Das folgende Kapitel beschreibt die prototypische Umsetzung des Recommender System. Zunächst werden die in diesem Kapitel verwendeten Technologien kurz beschrieben und es wird genauer darauf eingegangen, inwiefern sie in dem Prototyp verwendet wurden. Danach wird die Architektur des umgesetzten Prototyps und des Systems als Gesamtheit beschrieben. Im Anschluss wird auf die technische Ausgangslage eingegangen, auf welcher die Implementierung stattgefunden hat. Letztlich wird die Umsetzung des Prototyps technisch genauer beschrieben und erläutert.

6.1 Verwendete Technologien

im Folgenden werden die verwendeten Technologien dieses Kapitels aufgelistet. Dabei werden sie kurz erklärt und es wird erläutert, wofür sie genutzt wurden.

Anaconda Anaconda ist ein Paket- und Umgebungsmanager, mit welchem zahlreiche Open-Source-Pakete installiert und verwaltet werden können. [71] In der vorliegenden Arbeit wurde es verwendet, um mehrere Conda-Umgebungen zu erstellen. Eine Conda-Umgebung kann bestimmte Sammlungen an Paketen mit verschiedenen Versionen enthalten.

Cuda Cuda ist eine Plattform, welche parallele Berechnungen durchführen kann. In der vorliegenden Arbeit wird es genutzt, um mit einem Grafikprozessor zu arbeiten, da bestimmte Models, welche für den Prototyp benötigt wurden, sehr viel Rechenleistung erfordern. [72]

Docker Docker ist eine Plattform für die Entwicklung und Ausführung von Anwendungen, mit welcher Anwendungen von ihrer Infrastruktur getrennt werden können, um Software schneller bereitzustellen. [73] In der vorliegenden Arbeit wurde Docker Desktop genutzt, um schnell auf Elasticsearch zuzugreifen und darin die Dokumente zu verwalten.

Elasticsearch Elasticsearch ist eine verteilte Such- und Analysemaschine, in welcher JavaScript Object Notation (JSON)-Dokumente in einem NoSQL-Format gespeichert werden können. Zudem können Informationen mit Elasticsearch aggregiert werden, um Muster zu erkennen. In der vorliegenden Arbeit wurde Elasticsearch genutzt, um vorverarbeitete Dokumente zu speichern und zu suchen. [74]

Flask Flask ist ein Micro Framework, welches es ermöglicht Webanwendungen mit Python zu programmieren. In der vorliegenden Arbeit wird es genutzt, um Daten in Echtzeit zwischen

zwei Projekten zu transferieren. Flask hat einen einfachen Kern, welcher sich leicht erweitern lässt, wodurch viele Entscheidungen von der anwendenden Person selbst getroffen werden können. [75]

Git Git ist eine Open-Source-Software zur Versionsverwaltung von Dateien. Es ermöglicht mehrere lokale Verzweigungen, welche voneinander unabhängig sein können. Diese können dann wieder zusammengeführt oder separat gelöscht werden. [76] Git wurde in der vorliegenden Arbeit genutzt, um die verschiedenen Projekte mit ihrem Quellcode zu verwalten.

Haystack Haystack ist ein Open-Source-Framework von deepset, welches mithilfe von NLP Dokumentensammlungen auf unterschiedliche Weise verarbeiten kann. Grundlegend kann die Funktionsweise von Haystack in zwei Kategorien eingeteilt werden. Haystack kann zunächst als Indexing Pipeline[6] genutzt werden, hierbei werden Daten eingelesen, konvertiert und können dann vorverarbeitet werden. Die zweite Funktionsweise von Haystack ist die Search Pipeline. Bei dieser werden die vorverarbeiteten Daten zunächst in einem Retriever[7] verarbeitet und danach in einen Reader[8] gegeben. [78] In diesem Kapitel wird vor allem die Indexing Pipeline von Haystack verwendet.

Hugging Face Hugging Face ist eine Sammlung an verschiedensten Funktionen zu den Themenbereichen Machine Learning, NLP und Deep Learning. In der vorliegenden Arbeit wird ein Transformer-Modell von Hugging Face verwendet, welches eine Zero-Shot-Classification durchführt. [79]

Natural Language Toolkit Das Natural Language Toolkit (NLTK) ist eine Plattform für die Erstellung von Programmen, welche mit menschlicher Sprache arbeiten. Dazu werden Schnittstellen zu lexikalischen Ressourcen zur Verfügung gestellt. [80] In der vorliegenden Arbeit wurde es genutzt, um Stop Words aus den Items zu entfernen, um diese passend vorzuverarbeiten.

RASA Rasa ist ein Open-Source-Framework für maschinelles Lernen, welches es ermöglicht, automatisierte text- und sprachbasierte Konversationen zu führen. Das Framework versteht Nachrichten und kann Unterhaltungen führen. [81] Es schneidet im Vergleich zu anderen Frameworks, welche sich mit NLU für die Chatbot-Entwicklung befassen, gut ab. [82] In der vorliegenden Arbeit basiert die Chatbot-Komponente auf RASA, welche mit dem prototypischen Recommender System interagieren soll.

[6] Eine Pipeline beschreibt eine Abfolge von Prozessen, welche miteinander verkettet sind. Meistens wird die Ausgabe des vorherigen Prozesses als Eingabe für den nächsten Prozess verwendet. [77]

[7] Der Retriever führt eine Suche nach Dokumenten durch. Dabei durchsucht er eine Datenbank und gibt daraufhin eine Reihe an relevanten Dokumenten zurück. [78]

[8] Der Reader erhält die von dem Retriever ausgewählten Dokumente als Eingabe und wählt innerhalb dieser Dokumente Textausschnitte aus, welche die Frage der anwendenden Person beantworten sollen. [78]

6.2 Architektur des Prototyps

Bevor der Prototyp auf technischer Ebene betrachtet wird, wird die Architektur des entwickelten Prototyps vorgestellt. Diese unterscheidet sich von der konzeptionellen Architektur aus Abschnitt 5.3, da bei der praktischen Umsetzung verschiedene Umgebungsfaktoren miteinbezogen werden müssen. Bei diesen Faktoren handelt es sich um die verfügbaren Daten, der zur Verfügung stehende Umsetzungszeitraum und das Projekt Pulsnetz als Anwendungsfall.

6.2.1 Systemüberblick

Das gesamte System besteht aus vier Hauptkomponenten, welche zusammen einen funktionierenden Chatbot mit integriertem Recommender System darstellen. Wie diese zusammenspielen, wird in der Abbildung 6.1 dargestellt.

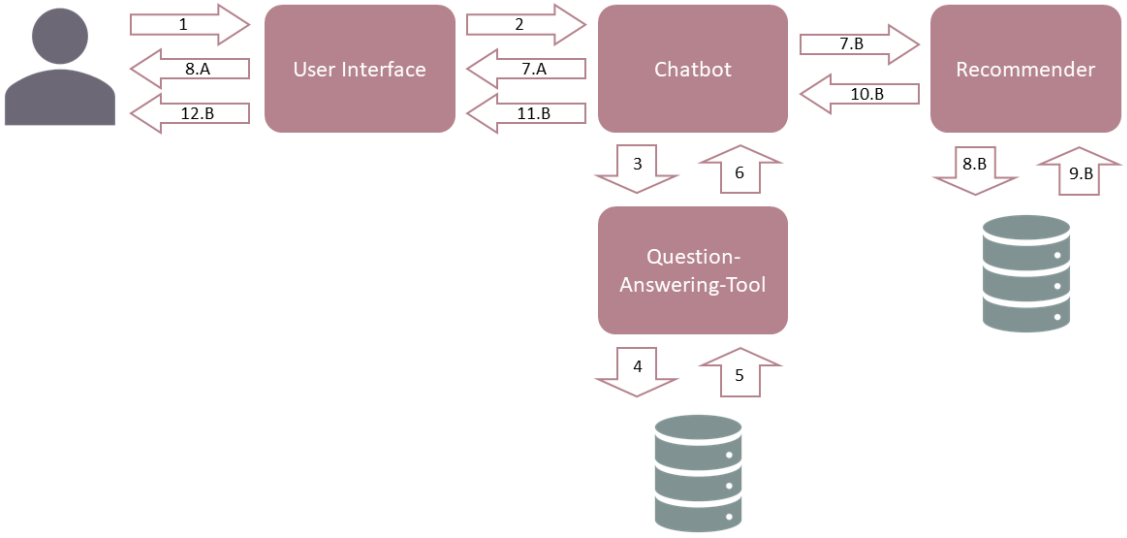


Abbildung 6.1: Systemüberblick des Prototyps

Das UI wurde von der CAS Software AG entwickelt und gestaltet sich als einfaches Chat-Fenster, welches es der anwendenden Person ermöglicht, mit dem System zu kommunizieren. „CASy“ bietet neben der Chatfunktion die Möglichkeit Sprachaufnahmen einzusprechen und gibt zudem Informationen aus dem Backend des Chatbots zurück. Diese Informationen beinhalten beispielsweise die Intents und Entities des Chatverlaufs, sowie die Konversation im Allgemeinen. In der Mitte dieser Architektur steht der Chatbot-Server „Impuls“ des Projektes Pulsnetz. Dieser ist an eine Question-Answering Komponente angebunden. Die Chatbot-Komponente ist dafür zuständig, die Eingaben der nutzenden Person zu analysieren und daraufhin die gewünschte Aktion auszuführen. Das Recommender System hat die Aufgabe, aus den erhaltenen Kontextdaten eine möglichst passende Empfehlung an die nutzende Person auszuspielen.

Der Ablauf beginnt damit, dass die anwendende Person eine Nachricht in dem UI sendet. Das UI leitet diese an den Chatbot-Server weiter. Dieser verarbeitet den erhaltenen Text mithilfe von NLU und bringt ihn in eine Form, die von dem Computer verstanden werden kann. Dabei werden

auch die Intents und Entities erkannt und extrahiert. Sollte die Anfrage der anwendenden Person ein Backend erfordern, um den Task richtig zu erfüllen, leitet der Chatbot-Server die Intents an das Question-Answering-Tool weiter. Dieses analysiert die Intents und sucht die passende Antwort auf die Anfrage. Dazu leitet das Question-Answering-Tool die Daten an eine Datenbank weiter, welche die passende Antwort zurückliefert. Die Antwort wird daraufhin zum Chatbot-Server gesendet. Dieser wandelt die Antwort mit Natural Language Generation wieder in natürliche Sprache um und leitet sie an das UI weiter, wodurch die Antwort bei der nutzenden Person erscheint. Während diesem Prozess wird wiederum entschieden, ob genug Kontextdaten für eine passende Empfehlung vorliegen. Falls dies der Fall sein sollte, schickt der Chatbot-Server alle Daten, welche ihm zur nutzenden Person vorliegen, an das Recommender System. Auch dieses arbeitet mit einer Datenbank, in welcher die enthaltenen Dokumente hinterlegt sind. Die erhaltenen Kontextdaten werden hier mit verschiedenen Methoden verarbeitet, auf die genaue Architektur des Recommender System wird in Abschnitt 6.2.3 eingegangen. Das Recommender System sendet daraufhin die Empfehlung zurück zum Chatbot-Server, welcher die Antwort wiederum zum UI weiterleitet.

6.2.2 Architektur des Chatbot-Servers

Da der Chatbot grundlegend dafür ist, dass das gesamte System richtig zusammenspielen kann, wird im Folgenden auch seine Architektur spezifischer erläutert, welche in Abbildung 6.2 aufgezeigt wird.

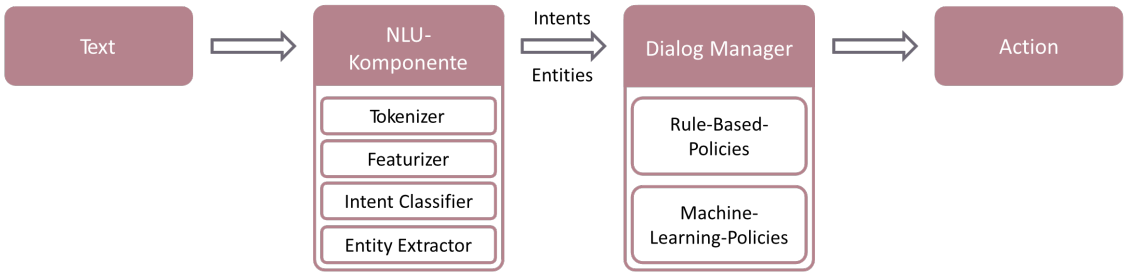


Abbildung 6.2: Architektur des Chatbots nach [26]

Der Chatbot-Server lässt sich in zwei Hauptkomponenten aufgliedern. Diese sind das NLU-Model und der Dialog-Manager. Die Text-Rohdaten werden zunächst an das NLU-Model gesendet. Das Ziel von diesem ist es, den Text so zu bearbeiten, dass Intents und Entities extrahiert werden können. Dazu durchläuft der Text im NLU-Model vier verschiedene Schritte, welche im Folgenden genauer erklärt werden.

1. **Tokenizer** Der Tokenizer ist dafür zuständig, die Eingabe in Sequenzen zu zerlegen, diese Sequenzen bestehen bei dem hier vorliegenden Chatbot aus einzelnen Wörtern, da in diesem Fall immer nach Leerzeichen getrennt wird. [13]

2. **Featurizer** Der Featurizer hat die Aufgabe, aus den Tokens verschiedene Merkmale für die Entity Extraction und den Intent Classifier zu extrahieren und diese weiterzugeben. Dabei wird für jeden Ausdruck ein Merkmal gesetzt, welches angibt, ob der Ausdruck in der Nachricht der anwendenden Person existiert. [13, 81]

3. **Intent Classifier** Der Intent Classifier weist der Nachricht der nutzenden Person einen der vordefinierten Intents zu. Der zugewiesene Intent wird daraufhin an den Dialog-Manager weitergeleitet. [83]

4. **Entity Extractor** Zuletzt extrahiert der Entity Extractor Entities, welche in der Nachricht der nutzenden Person enthalten sind. Der Entity Extractor des Chatbots „Impuls“ implementiert dazu ein bedingtes Zufallsfeld, welches die Entitäten erkennen soll. Ein bedingtes Zufallsfeld ist ein statistisches Modell, welches zur Modellierung von Datenfolgen verwendet wird. In diesem Fall stellen die Zeitschritte Wörter dar und die Zustände werden als Entitätsklassen beschrieben. Damit kann berechnet werden, welche Entität zutrifft. [81, 13]

Die identifizierten Intents und Entities werden daraufhin an den Dialog-Manager weitergegeben. Der Dialog-Manager entscheidet auf Basis der Intents und Entities, welche Aktion der Chatbot durchführen soll, um eine Antwort zu erzeugen. Um die Antwort zu erzeugen, kann je nach Aktion das Question-Answering-Tool angesprochen werden. Der Dialog-Manager unterteilt sich bei dem hier vorliegenden Chatbot in zwei Komponenten.

Rule Based Policies Regelbasierte Richtlinien werden verwendet, wenn die Konversation einer festen Ablauflogik folgt. Auf Basis dieser wird dann eine passende Aktion ausgewählt.

Machine-Learning Policies In dem vorliegenden Chatbot wurden zwei Policies angewandt. Die erste arbeitet mit Transformers, in welche eine Verkettung von Eingaben, der anwendenden Person als Eingabevektor eingespeist wird. Die Ausgaben des Transformer werden genutzt, um Dialog- und System-Einbettungen für jeden Zeitschritt zu erhalten. Zwischen diesen Einbettungen können daraufhin Ähnlichkeiten berechnet werden. Die zweite Richtlinie ist dafür zuständig, sich die vorherigen Dialoginformationen zu merken, um Kontext berücksichtigen zu können. [81]

6.2.3 Architektur des Recommender Systems

Im Folgenden wird die Architektur des Recommender System im Detail betrachtet, da der Fokus der Arbeit auf dieser Komponente liegt, diese wird in Abbildung 6.3 dargestellt.

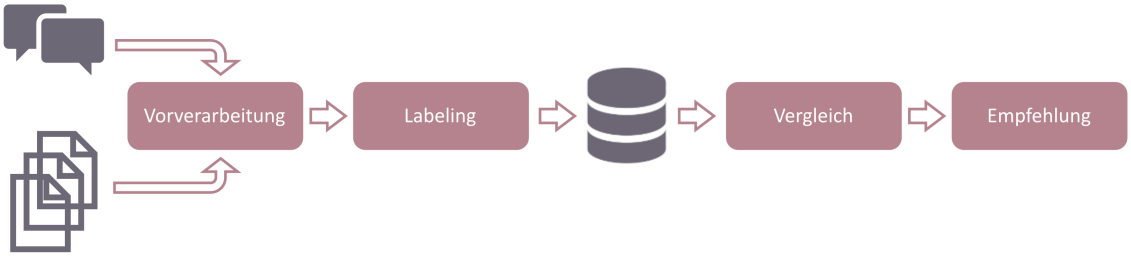


Abbildung 6.3: Architektur des Recommender Systems

Das Recommender System hat zwei verschiedene Eingaben, welche asynchron zueinander eingespielt werden. Im Folgenden wird die grobe Vorgehensweise kurz erklärt, bevor dann auf die einzelnen Inputs genauer eingegangen wird. Beide Inputs werden zunächst auf unterschiedliche

Weise vorverarbeitet und mit zuvor festgelegten Labels annotiert. Die daraus resultierenden Daten werden in eine Datenbank eingespeist und darauf folgend verglichen. Aus diesem Vergleich entsteht dann die Empfehlung, welche auf den Chat der nutzenden Person zugeschnitten ist.

Der erste Input, welcher näher erläutert wird, sind die Dokumente. Diese werden für zwei Prozesse genutzt, welche einander bedingen. Diese sind in Abbildung 6.4 detaillierter dargestellt.

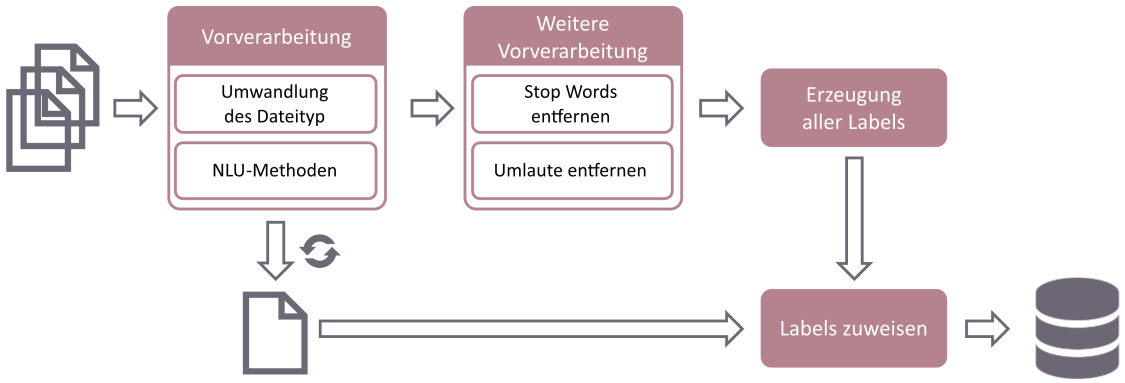


Abbildung 6.4: Architektur des Recommender Systems mit Fokus auf die Dokumente

Die Prozesse starten zunächst gemeinsam damit, dass die Dokumente in das System gegeben werden. Sie durchlaufen zunächst einen Prozess der Vorverarbeitung. In diesem werden die verschiedenen Dateitypen in einen einheitlichen Datentyp umgewandelt. Danach werden verschiedene Methoden des NLU angewandt, um den Rohtext in eine Form zu bringen, mit welcher besser weitergearbeitet werden kann. Dazu wird der Text unter anderem zerteilt. In diesem Fall wurde immer nach einer bestimmten Anzahl an Absätzen ein neues Item angelegt. Zudem werden Leerräume, leere Reihen, Header und Footer entfernt. Nach diesem Schritt teilt sich der Prozess auf, hierbei muss beachtet werden, dass die entstandenen Subprozesse nicht zeitgleich ablaufen, sondern der erste Subprozess vor dem zweiten ausgeführt wird. Keiner der Subprozesse wird jedoch in Echtzeit während des Chats mit der nutzenden Person ausgeführt, sondern schon zuvor. Das Ziel des ersten Prozesses ist es, aus der Gesamtheit der Dokumente Labels zu erzeugen. Dafür erfolgt eine weitere Vorverarbeitung aller Dokumente. Hierbei ist besonders wichtig, dass Stop Words entfernt werden, damit die zu erzeugenden Labels mehr Aussagekraft haben. Zudem werden auch Umlaute und Sonderzeichen entfernt. Mit diesen vorverarbeiteten Dokumenten können dann mithilfe von Topic Modeling Labels generiert werden, welche die behandelten Themen in allen Items widerspiegeln sollen. Der zweite Prozess, wofür die Dokumente genutzt werden, ist der grundlegendere für die Funktionalität des Recommender Systems. Hierbei wird jedes Item einzeln aufgerufen, damit ihm die zuvor generierten Labels zugewiesen werden können. Dafür wird eine Zero-Shot-Classification angewandt. Nachdem die zu den Themen des Items passenden Labels zugewiesen wurden, können sie an das Item geschrieben werden. Daraufhin wird das neue Objekt aus Item und Label in die Datenbank geschrieben, aus welcher es später wieder leicht abgerufen werden kann.

Der zweite Input, welcher in das Recommender System gegeben wird, ist der aktuelle Chatverlauf der nutzenden Person. Der damit folgende Prozess findet in Echtzeit statt, da die nutzende Person direkt eine Antwort erhalten soll. Aus dem Chatverlauf werden die Nachrichten der nutzenden Person und des Chatbots extrahiert. Dem Chatverlauf werden daraufhin passende Labels aus den zuvor erzeugten Labels zugewiesen. Nachdem dem Chatverlauf passende Labels zugewiesen wurden, kann der Vergleich mit den Labels der Items stattfinden.

6.3 Technische Ausgangslage

Nachdem die Architektur des Systems definiert und erläutert wurde, wird die technische Ausgangslage des Systems miteinbezogen. Damit soll aufgezeigt werden, welche Komponenten, mit welchen Eigenschaften schon vorhanden sind, um im Abschnitt 6.4 darauf eingehen zu können, wie die neue Komponente in das vorhandene System integriert wird. Im Folgenden wird deshalb auf den technischen Stand des Projekts vor der prototypischen Implementierung des Recommender System eingegangen. Die Abbildung 6.5 zeigt, welche Komponenten der Architektur bereits vorhanden sind, indem dieser Part farbig hinterlegt wurde.

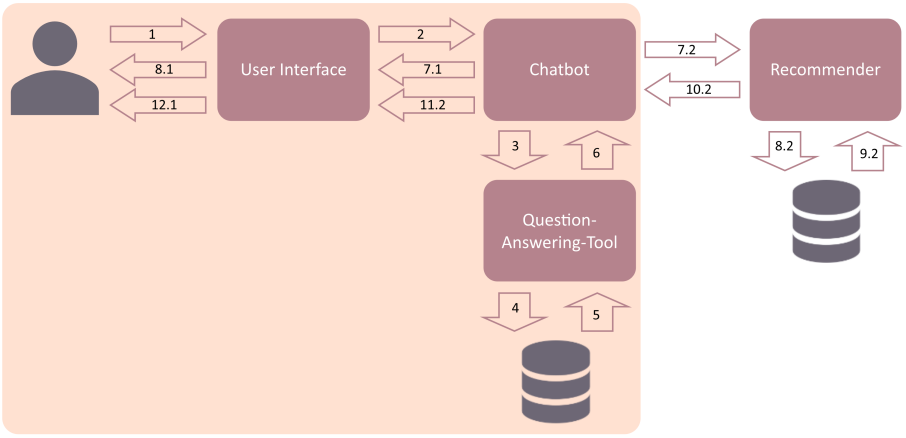


Abbildung 6.5: Technische Ausgangslage des Projekts „Pulsnetz“ nach Projektunterlagen

Zunächst wird definiert, unter welche Art von Chatbot „Impuls“ fällt. Dafür wird nach in den Grundlagen unterschiedenen Klassifizierungen in Abschnitt 3.2.3 vorgegangen. Im Bereich Interaktion fällt „Impuls“ unter die textbasierten Chatbots, da er nicht in der Lage ist auf gesprochene Wörter zu reagieren oder diese zu verarbeiten. Er hat ein geschlossenes Wissensgebiet, da er sich auf bestimmte Themen konzentriert und auf diese spezialisiert ist. Beispielhafte Themen, mit welchen der Chatbot umgehen kann, sind Arbeitnehmerschutz und psychische Gesundheit. Zuletzt kann „Impuls“ noch nach Designansätzen klassifiziert werden. Hierbei fällt er unter die Kategorie der abrufbasierten Chatbots, da er keine neuen Antworten erzeugt, sondern diese aus einem Informationspool auswählt. Er ist trotz dessen komplexer aufgebaut als regelbasierte Chatbots, da er verfügbare Ressourcen über eine API abrufen und verarbeiten kann. Zudem basiert die Response Selection des Chatbots auf Machine Learning.

Die Technologie hinter dem Chatbot ist RASA. RASA filtert die Intents der nutzenden Person heraus und leitet diese an Haystack weiter. Haystack verarbeitet daraufhin die Intents von RASA. Dazu arbeitet es mit der Datenbank Elasticsearch, in welcher PDF-Dokumente zu allen relevanten Themen des Chatbots hinterlegt sind. Sobald Haystack einen passenden Ausschnitt zur Frage der anwendenden Person in einem Dokument gefunden hat, gibt es diesen an RASA zurück. RASA spielt die Antwort daraufhin an die nutzende Person aus. Momentan kann der Chatbot „Impuls“ keine Empfehlungen ausspielen, weshalb diese Funktion in dem in dieser Arbeit entwickelten Prototypen implementiert und integriert werden soll.

6.4 Integration in die Systemumgebung

Um das Recommender System in Echtzeit an einem Chatbot nutzen zu können, muss es in die bereits vorhandene Systemumgebung des Chatbots integriert werden. Die Integration in das vorhandene System lässt sich dabei in drei Schritte aufgliedern, welche im Folgenden näher beschrieben werden.

6.4.1 Setup des Chatbot-Servers mit einem passenden User Interface

Zunächst findet ein Setup des schon vorhandenen Systems statt. Da das Recommender System nur als Prototyp dienen soll und auch Änderungen an der Komponente des Chatbots vorgenommen werden müssen, wird das gesamte Projekt in einem neuen Git-Repository aufgebaut, um eine bessere Testumgebung zu schaffen. Der Chatbot „Impuls“ wird für das Testen des Prototyps zusammen mit der „CASy“ Weboberfläche genutzt, da diese für die entwickelnde Person zusätzlich hilfreiche Backend-Informationen zur Verfügung stellt. Sowohl das UI, als auch die Chatbot-Komponente werden geklont und lokal gespeichert. Da für das Projekt viele Packages mit den richtigen Versionen benötigt werden, wurde mit Anaconda für jede Komponente ein eigenes Environment erzeugt. In dieses wurden dann die jeweiligen Packages installiert, sodass keine Konflikte zu anderen Projekten entstehen können. Der „CASy“ Webserver kann damit leicht gestartet werden und ist über den Localhost abrufbar. Der Chatbot-Server muss an zwei Stellen gestartet werden. Es wird zunächst der standardmäßige Start-Befehl von Rasa aufgerufen. Mit diesem kann dann über die Konsole mit dem Chatbot interagiert werden. Sobald allerdings durch eine Nachricht der nutzenden Person eine Action getriggert wird, wird der Rasa Action-Server benötigt, welcher separat gestartet werden muss. Um den Chatbot-Server mit dem UI „CASy“ zu verbinden, wird in den Endpoints des Chatbot-Servers die Adresse des Localhost hinterlegt. Auf diesem Stand lässt sich der Chatbot mit dem UI mit seinen vollständigen Funktionen nutzen.

6.4.2 Extrahieren der Kontextdaten aus dem Chatbot-Server

Um das Recommender System mit Daten zu versorgen, werden Kontextdaten aus dem Chatbot-Server benötigt. Es muss festgelegt werden, welche Kontextdaten extrahiert werden sollen und zu welchen Zeitpunkten dies geschehen soll. Die Kontextdaten, welche gespeichert werden sollen,

sind die Uhrzeit, das Datum und die jeweiligen Chat-Nachrichten der bisherigen gesamten Konversation von dem Chatbot und der nutzenden Person mit ihren jeweiligen Intents und Entities. Dazu werden diese aus der Konversation extrahiert und in ein JSON-File geschrieben, damit sie leicht abgerufen und versendet werden können. Um festzulegen, zu welchen Zeiten Konversationsdaten gespeichert werden sollen, müssen die einzelnen Rules und Actions genauer betrachtet werden. Der Chatverlauf soll nur gespeichert werden, wenn die Action thematisch auch zu den Items passt, welche im Recommender System hinterlegt sind. Somit kann eine erste Filterung stattfinden. Zudem soll die Konversation nur dann gespeichert werden, wenn die Chat-Nachrichten einen Inhalt aufweisen. Somit darf der typische Smalltalk die Logging-Funktion, welche die Konversationsdaten speichert, nicht triggern, da auf diesen auch keine sinnvolle Empfehlung folgen können.

6.4.3 Aufbauen einer Schnittstelle zu dem Recommender System

Um die gespeicherten Konversationsdaten in Echtzeit an das Recommender System zu übermitteln, muss eine Schnittstelle zwischen den beiden Projekten hergestellt werden. Diese muss Daten von dem Chatbot-Server zu dem Recommender System schicken und im gleichen Zug eine Antwort von dem Recommender System an den Chatbot-Server zurücksenden können, sobald dieser die Daten verarbeitet und eine passende Empfehlung gefunden hat. Hierfür wird die REST-Architektur verwendet, da sie flexibel und schlank anwendbar ist. Um die REST-Architektur aufzubauen wird das Framework Flask verwendet, welches es ermöglicht Daten zwischen zwei Projekten zu transferieren. Die Flask-App wird dabei im Recommender System erzeugt und mit dem standardmäßigen Start-Befehl von Flask gestartet. In der Route kann dann die URL festgelegt werden, auf welcher die Daten empfangen werden sollen. Zusätzlich wird noch die Methode angegeben, mit welcher die Daten empfangen werden sollen. Hierbei handelt es sich im Recommender System um ein GET-Request mit einem direkt darauf folgenden POST-Request, da der Recommender zuerst Daten empfangen will, diese in Echtzeit verarbeitet und daraufhin in derselben Route direkt eine Antwort zurücksenden will. Diese URL, über welche die Daten transferiert werden, muss dann in dem Chatbot-Server beim Versenden der Konversationsdaten wieder angegeben werden. Im Chatbot-Server wird außerdem der Content-Typ angegeben, damit ersichtlich ist, welches Dateiformat versendet werden soll. Dann wird ein POST-Request gestartet, in welchem die passende URL, die zu versendenden Daten und die Headers angegeben werden. Letztlich gibt das Recommender System die drei passendsten Empfehlungen an den Chatbot zurück, welche im Chatbot-Server nachfolgend direkt ausgespielt werden.

6.5 Umsetzung des Recommender Systems

Im folgenden Abschnitt wird die Umsetzung des Recommender System erläutert. In diesem finden mehrere Prozesse mit unterschiedlichen Komponenten statt, welche, wie in Abschnitt 6.3 aufgezeigt, zusammenspielen. In diesem Abschnitt wird der Fokus auf die technische Komponente hinter dieser Architektur gelegt.

6.5.1 Setup des Recommender Systems

Das Recommender System ist in einem Projekt zusammengefasst, welches alle Komponenten und Prozesse beinhaltet, um eine Empfehlung auf Basis eines Chatverlaufs auszusprechen. Das Projekt wurde in der Programmiersprache Python geschrieben und wird über Git synchronisiert. Beim Setup des Projekts wurde zudem ein neues Anaconda Environment angelegt, um die notwendigen Packages zu installieren und zu verwalten. Da das Projekt an mehreren Stellen die Leistung eines Grafikprozessors benötigt, wurde Cuda installiert. Das Projekt benötigt zudem eine Datenbank, in welcher die Items gespeichert und später wieder abgerufen werden können. Hierfür wurde Elasticsearch verwendet, da es viele Funktionen zum Vorverarbeiten, Suchen und Ordnen von Dokumenten zur Verfügung stellt, was vor allem bei NLP von großem Vorteil sein kann. Elasticsearch wurde mit Docker verknüpft und kann somit durch einen Docker Container gestartet werden. Um eine Verbindung zu dem Chatbot-Server herzustellen, muss letztlich noch ein Setup von Flask stattfinden. Nach der Installation von Flask wird der Pfad der Flask-App auf das File, in welchem die Flask App geschrieben ist, angepasst, um diese später starten zu können.

6.5.2 Vorverarbeitung der Items

Um Daten vorzuverarbeiten, müssen diese zunächst beschafft werden. Die Daten, welche für den vorliegenden Prototypen verwendet wurden, wurden von der Webseite der Berufsgenossenschaft für Gesundheitsdienst und Wohlfahrtspflege (BGW) entnommen. Dabei wurden zum größten Teil auf der Webseite verfügbare PDF-Dokumente entnommen, aber auch Webseiten-Inhalte als Text-Datei verwendet. Es wurde sich für die Inhalte der BGW-Webseite entschieden, da BGW in dem Projekt „Pulsnetz“ mitwirkt.

Für die Vorverarbeitung der Dokumente wurde im Prototypen Haystack verwendet, da es sowohl einen Converter, als auch einen Präprozessor besitzt, mit welchem gearbeitet werden kann. Um die verschiedenen vorliegenden Dokumente alle auf die gleiche Weise weiterzuverarbeiten, müssen sie zunächst in ein einheitliches Dokumentenformat umgewandelt werden. Hierzu wird der Converter von Haystack verwendet, welcher die Dokumententypen .pdf, .word und .txt in einen Dokumententyp von Haystack umwandelt. Diese Dokumententypen können dann mit dem Präprozessor von Haystack in eine geeignete, vorverarbeitete Form gebracht werden. Der Präprozessor von Haystack verwendet hierzu die NLU-Methoden Cleaning und Splitting.

Cleaning Beim Cleaning geht darum, bestimmte Stellen aus dem Text zu entfernen, welche keinen Mehrwert an Informationen für die Verarbeitung liefern. In diesem Fall werden drei oder mehr leere Linien entfernt. Außerdem werden Leerzeichen am Anfang und am Ende von Zeilen entfernt. Letztlich werden noch Header und Footer entfernt, was vor allem bei PDF-Dokumenten sehr nützlich ist. [78]

Splitting Splitting ist dafür zuständig, lange Dokumente in Wörter, Sätze oder Absätze aufzuteilen. In diesem Fall wird nach 50 Absätzen ein neues Dokument begonnen. Somit werden Sätze nie in der Mitte aufgeteilt, sondern bleiben zusammen in einem Item. [78]

6.5.3 Erzeugung der Labels

Um die Dokumente und Chatdaten mit Labels versehen zu können, werden passende Labels benötigt. Diese werden mit BERTopic erzeugt. BERTopic ist eine Technik für das Topic Modeling, welches mithilfe von Transformers und Term Frequency Inverse Document Frequency (TF-IDF), Themen aus einer Gesamtheit an Dokumenten repräsentieren kann. Dabei werden Wortmengen extrahiert, welche aus einer Liste von Keywords bestehen, welche die Themengebiete der Dokumente abbilden sollen. TF-IDF beschreibt ein statisches Maß, welches genutzt werden kann, um die Bedeutung von Sätzen darzustellen. BERTopic verwendet das Framework Sentence-BERT (SBERT). SBERT erlaubt der anwendenden Person, mit vortrainierten transformer-basierten Sprachmodellen Sätze als Word Embeddings in einer Vektor-Repräsentation darzustellen. Diese Einbettungen werden daraufhin in Cluster umgewandelt, aus welchen repräsentative Themen mithilfe eines klassenbasierten TF-IDF-Verfahren generiert werden können. [24] Ein Ausschnitt der finalen erzeugten Labels wird in Abbildung 6.6 aufgezeigt.

```
{
    "schwangerschaft": [
        "schwangerschaft", "muetter", "mutterschutz", "arbeitgeber",
"gefaehrdungsbeurteilung", "stillende", "werdende"
    ],
    "mobbing": [
        "mobbing", "vertrauensperson", "person", "konflikt",
"ursachen", "betroffene", "betrieb", "konflikte"
    ],
    "ruecken": [
        "patienten", "arbeitsweise", "wirbelsaeulenbelastung",
"koerperschaft", "rueckenschmerzen", "ruecken"
    ],
}
```

Abbildung 6.6: Ausschnitt der verarbeiteten Informationen von BERTopic

Im ersten Versuch wurde eine Pipeline von BERTopic erzeugt, welche mehrere Sprachen erkennen soll. In diese wurden dann alle mit Haystack vorverarbeiteten Items eingespielt und Labels aus der Gesamtheit dieser erzeugt. Diese Labels waren allerdings nicht aussagekräftig über die Gesamtheit aller Items. Um die Labels zu verbessern, wurden verschiedene Ansätze verfolgt, welche im folgenden kurz vorgestellt werden.

Da die Pipeline auf Englisch zuverlässiger funktioniert, als wenn sie deutsch oder mehrere Sprachen erkennen muss, wurde zunächst versucht, die Items in die englische Sprache zu übersetzen. Hierzu wurden verschiedene Bibliotheken zur Übersetzung von längeren Dokumenten gefunden und verglichen. Die Python-Client-Bibliothek für die DeepL API wurde aufgrund der Begrenzung der Wörter verworfen, da die Dokumente sehr umfangreich sind und die Möglichkeit geboten werden soll, weitere Dokumente hinzuzufügen. Die Bibliothek für die API von Google Translate

war zwar für größere Übersetzungen geeignet, jedoch funktionierte sie nicht richtig aufgrund von Versionskonflikten zu anderen Packages, welche für das Recommender System benötigt wurden. Das gleiche Problem trat bei einer neueren Bibliothek, welche mit der API von Google Translator arbeitet, auf. Die letzte Bibliothek, welche getestet wurde, war die Bibliothek Deep Translator. Mit dieser konnten die schon vorverarbeiteten Items übersetzt werden, allerdings mussten sie dafür in kleinere Ausschnitte aufgeteilt werden, da sie sonst nicht von der API akzeptiert wurden. Mit den übersetzten Dokumenten wurden dann wieder Labels mit einer englischen Pipeline von BERTopic erzeugt. Die erzeugten Labels waren dadurch zwar passender zu den Themen in den Dokumenten, allerdings sind vor allem Fachwörter durch die Übersetzungen verloren gegangen, welche vor allem im Gesundheitssektor eine hohe Relevanz aufweisen. Deshalb wurde der Ansatz bessere Labels durch Übersetzungen zu erzeugen nachfolgend verworfen.

Ein weiterer Ansatz, der daraufhin verfolgt wurde, sieht eine weitere Vorverarbeitung der Dokumente vor. Dazu wurden zunächst alle Stop Words mit dem NLTK aus den Dokumenten entfernt. Dazu wird die Sammlung der Stop Words auf Deutsch des NLTK aufgerufen. Diese Liste kann daraufhin mit weiteren Wörtern ergänzt werden, welche aus den Dokumenten ausgeschlossen werden sollen. Die Liste an Stop Words wird nachfolgend mit einer einfachen Schleife aus den Dokumenten entfernt. Um bessere Labels zu erhalten, wurden zusätzlich alle Sonderzeichen, sowie alle URLs entfernt. Im letzten Schritt der zweiten Vorverarbeitung wurden alle Umlaute aus den Dokumenten ersetzt und alle Buchstaben zu Kleinbuchstaben konvertiert.

Durch diese weitere Vorverarbeitung wurden die mit BERTopic erzeugten Labels wesentlich treffender und es wurden weniger Worte erzeugt, welche nicht brauchbar waren. Die Labels werden von BERTopic selbstständig in Kategorien unterteilt. Eine dieser Kategorien zeigt Abbildung 6.7. Es ist erkennbar, dass diese Kategorien nicht mit einem Topic als Oberbegriff versehen sind.

```
[
    ('gesundheit', {0.03274648262627894}),
    ('fuehrung', 0.021878326712491916),
    ('beschaeftigten', 0.020271802578368508),
    ('fuehrungskraft', 0.017616246242685676),
    ('fuehrungskraefte', 0.017502531199692653),
    ('zusammenhang', 0.01708954263585925),
    ('mitarbeiter', 0.01654767923426038),
    ('psychische', 0.01624615182671019),
    ('ressourcen', 0.015463801329438805),
    ('stress', 0.015120637040401329)
]
```

Abbildung 6.7: Labels eines erzeugten Topics mit BERTopic

Dieser muss manuell nachträglich gewählt werden. Die folgenden manuellen Schritte, die getätigt wurden, sind optional und dienen lediglich dazu, die Labels weiter zu verbessern, haben aber keinen Einfluss auf die Funktionalität des Recommender System. Es wurden zuerst Labels, welche keine sinnvollen Wörter beinhalteten, gelöscht. Danach wurden einzelne Kategorien, wie in Abbildung 6.7 gezeigt, zusammengefasst und mit einem Oberbegriff versehen. Letztlich wurde noch eine weitere Kategorie manuell hinzugefügt, welche verschiedene Berufsgruppen enthält.

6.5.4 Labeling der Items und der Chatdaten

Die erzeugten Labels, welche aus der Gesamtheit aller Items gewonnen wurden, können im Folgenden genutzt werden, um den Chatdaten und den Items daraus passende Labels zuzuweisen. Für diesen Prozess wird eine Zero-Shot-Classification genutzt. Diese wird mit dem vortrainierten Modell Sahajtomar/German_Zeroshot, welches auf Hugging Face zur Verfügung gestellt wurde, durchgeführt. Dieses Modell nutzt ein German-Bert (GBERT) Large Modell von deepset, als ein Basis-Modell. Für das Finetuning wurde dann der deutsche Datensatz von XNLI verwendet. [79, 84]

Das Labeling hat für die Chatdaten, sowie für die Items denselben Ablauf, weswegen für beide dieselbe Methode aufgerufen werden kann. Da die Methode einen vergleichbaren Input erhalten muss, werden zunächst auch die Chatdaten vorverarbeitet. Dieser Prozess muss nicht so umfangreich, wie zuvor bei den Items durchgeführt werden, da die Chatdaten weniger unbrauchbare Informationen enthalten, wie beispielsweise Header oder Footer. Zunächst werden die Events für die anwendende Person und den Chatbot aus der JSON-Datei extrahiert und in eine Liste geschrieben. Der Inhalt dieser Liste wird dann in Kleinbuchstaben umgewandelt und Umlaute werden entfernt. Damit kann der Prozess des Labelings gestartet werden. Für das Labeling mit dem Modell aus [79] von Hugging Face wird sehr viel Leistung benötigt, weshalb ein Grafikprozessor zum Einsatz kam. Zuerst kann eine Pipeline mit dem entsprechenden Modell von Hugging Face aufgebaut werden, welche später dann mehrmals verwendet werden kann. Der gesamte Ablauf von dem Labeling ist in Abbildung 6.8 in abstrahierter Form dargestellt.

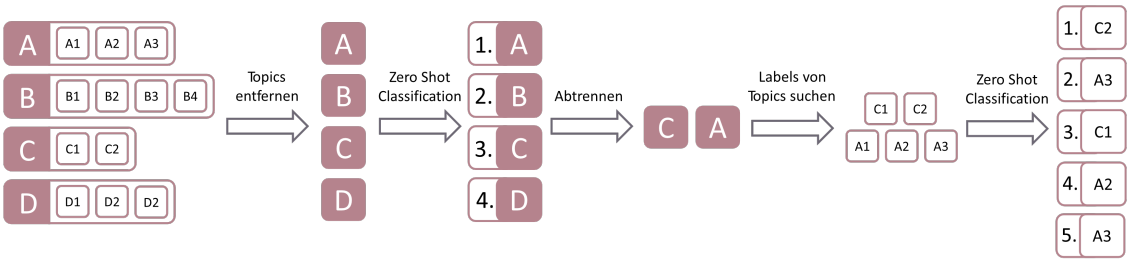


Abbildung 6.8: Abstrahierter Prozess von dem Labeling der Daten

Zur besseren Veranschaulichung wurde er mit Buchstaben dargestellt. Die Buchstaben stehen dabei für die Topics und die Buchstaben, welche mit Zahlen versehen wurden, für das jeweilige Label zu dem Topic, dessen Buchstabe verwendet wurde. Zu Beginn des Prozesses wird die JSON-Datei mit den zuvor mit BERTopic erzeugten Topics mit ihren jeweiligen Labels geladen. Aus dieser

werden dann die Topics extrahiert und in eine neue Liste geschrieben. Aufgrund von besseren Ergebnissen und aus Performance-Gründen durchlaufen zunächst die Topics der Labels die aufgebaute Pipeline. Die Topics werden daraufhin nach ihrem Score absteigend aufgelistet. Um die Anzahl der Labels zu verringern, wird nur die erste Hälfte der Topics, welche besser zu den Daten passt, verwendet. Darauffolgend durchlaufen die Labels der übrigen Topics die Pipeline von Hugging Face. Damit ergibt sich eine absteigend sortierte Liste mit Labels und ihren jeweiligen Scores. Im Anschluss werden noch die Labels mit den Berufen zugewiesen. Dazu durchlaufen diese zusammen mit den Textdaten die Pipeline, sodass auch hierbei eine absteigende Sortierung mit jeweiligen Scores vorliegt. Die folgende Abbildung 6.9 zeigt die Struktur der zugewiesenen Labels und Berufe mit den Scores. Zu beachten ist, dass die Abbildung nicht die Gesamtheit der Labels abbildet, sondern nur einen Ausschnitt der zugewiesenen Labels von einem Item darstellt. Ein ausführlicher Ausschnitt eines gelabelten Chatverlaufs befindet sich im Anhang A.3.


```
{
    "automatic_labels": [
        {
            "label": "rueckenschmerzen",
            "score": 0.1213025376200676
        },
        {
            "label": "ruecken",
            "score": 0.08589255809783936
        }
    ],
    "profession": [
        {
            "label": "krankenpflege",
            "score": 0.17861826717853546
        },
        {
            "label": "pflegekraft",
            "score": 0.17535577714443207
        }
    ],
}
```


Abbildung 6.9: Ausschnitt der zugewiesenen Labels und Berufe für ein Item

Es ergab sich ein Problem mit den Scores der Labels, da diese relativ über die Anzahl der Labels verteilt werden. Wenn die Anzahl der Labels verändert wird, was durch den Prozess aus Abbildung 6.8 eintritt, verfälscht dies die Scores, da diese in Summe nicht mehr 100 % ergeben. Damit kann keine Vergleichbarkeit mehr garantiert werden. Um diese trotzdem garantieren zu können, wurde die Formel 6.1 aufgestellt, welche die Scores einzeln anpasst.

$$S_{neu}(x) = \frac{1}{\sum_{1}^{n} S_{alt}(x)} \cdot S_{alt}(x) \tag{6.1}$$

Im Folgenden wird die Formel kurz erklärt. S (x) beschreibt den Score des jeweiligen Labels. S_{alt} steht dabei für den Score vor der Anwendung der Formel und S_{neu} für den neuen Score. Im Nenner wird die Summe aller vorher existierenden Scores von eins bis n gebildet. Das Ergebnis der Division wird letztlich noch mit S_{alt} multipliziert.

Das n beschreibt das Minimum der möglichen Labels, die ein Item oder ein Chatverlauf besitzen kann. Die minimale Anzahl an Labels, die einem Chatverlauf oder einem Item zugewiesen werden können, hängt von der Anzahl der Labels unter der jeweiligen Überschriften ab, welche in Abbildung 6.6 beispielhaft dargestellt wurden. Ein weiterer Faktor, der n bestimmt, ist die Anzahl an Topics, welche für die endgültige Zuweisung verwendet wird. Diese Anzahl wird als Parameter in die Methode zum Berechnen von n gegeben und wird im folgenden als a bezeichnet. Sie muss individuell festgelegt werden, da der Wert von a auf einer explorativen Analyse beruht.

Der Algorithmus dieser Methode wird im Folgenden kurz erläutert. Zunächst wird die Anzahl der Labels von jeder Überschrift in eine neue Liste geschrieben. Diese Liste wird dann aufsteigend sortiert und nachfolgend an der Stelle a abgeschnitten, sodass alle Werte hinter a nicht weiter betrachtet werden. Letztlich wird die Summe aus allen Werten der Liste gebildet. Diese Summe beschreibt n. Nachdem die oben stehende Formel auf alle Scores angewandt wurde, besitzt der Text noch n Labels mit ihren jeweiligen Scores, welche in der Summe wieder 100 % ergeben.

6.5.5 Abgleich der Labels und Scores

Nachdem sowohl der Chatverlauf als auch die Items mit den von BERTopic erzeugten Labels und Berufen annotiert wurden, können die Items mit dem Chatverlauf abgeglichen werden, um die drei passendsten Items ranglistig zu finden und empfehlen zu können. Das Ziel des folgenden Algorithmus ist es, jedem Item ein Score zuzuweisen, welcher aussagt, wie gut das Item zu dem Chatverlauf passt. Somit werden die Items mit den drei höchsten Scores der Reihenfolge nach empfohlen.

Im Folgenden wird der Algorithmus hinter dem Scoring genauer erläutert, um transparent zu machen, inwiefern die zu empfehlenden Items genau ausgewählt wurden. An diesem Punkt wird die Unterscheidung zwischen den Themen und den Berufen besonders wichtig, denn diese werden für das Scoring der Recommendation separat betrachtet. Zunächst werden Limits für die Scores der Themen und Berufe festgelegt. Um das Limit der Scores der Themen festzulegen, wird der durchschnittliche Score x = 1 / n berechnet. Eine Exploration der Empfehlungen ergab, dass die Empfehlungen an Qualität gewinnen, wenn das Limit 5 % über dem Durchschnitt angesetzt wird. Der Score der Berufe beruht ebenfalls auf einer explorativen Analyse. Diese Limits wurden als

Parameter eingegeben und sollten individuell angepasst werden. Dies sollte geschehen, da je nach Anwendungsfall des Recommender Systems die Berufe eine unterschiedliche Wichtigkeit für die Empfehlung mit sich bringen. Da jedes Item einen eigenen Score erhalten soll, wurden die einzelnen Items nacheinander betrachtet. Zunächst wurden nur die Themen betrachtet, dabei wurde jedes Label des Chatverlaufs mit den Labels des Items abgeglichen. Sollte ein Label in dem Item und in dem Chat vorhanden sein, wurde geprüft, ob der Score des Labels bei dem Item und bei dem Chatverlauf über dem festgelegten Limit liegt. Sollte dies der Fall sein, wird das Label bei der Berechnung des Scores des Items mitaufgenommen. Die Berechnung, mit welcher Gewichtung das Label in den endgültigen Score des Items eingeht, wird in der Formel 6.2 beschrieben.

$$\underset{fi}{S} += \underset{li}{(S)^2} \cdot \underset{lc}{(M \cdot S)} \tag{6.2}$$

Das Formelzeichen S_{fi} vor dem Gleichheitszeichen steht für den Score des momentan behandelten Items. Dieser wird immer, wenn ein ihm zugewiesenes Label die obigen Anforderungen erfüllt, um die hinten stehende Formel erhöht, was durch das Additionszeichen vor dem Gleichheitszeichen ausgedrückt wird. Die Formel quadriert S_{li} und multipliziert es mit den Werten M und S_{lc}. S_{li} steht dabei für den Score des Labels bei dem Item, dieser wird quadriert, weil er zu einem geringeren Teil Einfluss auf die Empfehlung nehmen soll, als der Score des Labels an dem Chat. Dies begründet sich darin, dass der Chatverlauf essenzieller für die Empfehlung ist als die Items. S_{lc} steht für den Score des Labels bei dem Chatverlauf und M steht für den Multiplikator, welcher mit S_{lc} und S_{li} verrechnet wird. Die Berechnung von M wird in der Formel 6.3 dargestellt.

$$M = \frac{(n - i_c)^2}{n} \tag{6.3}$$

Der Multiplikator M ist abhängig von der Ranghöhe des Labels und der im Abschnitt 6.5.4 beschriebenen Anzahl n. Der Index i_c beschreibt dabei den Rang des Labels von dem Chatverlauf. Das am besten passendste Label des Chats hat dabei den Index 1 und das am schlechtesten passendste Label den Index n. Umso niedriger der Index eines Labels ist, umso kleiner wird der Multiplikator und damit die Relevanz des Labels auf den endgültigen Score des Items.

Im nächsten Schritt wird entschieden, ob die dem Chat zugewiesenen Berufe einen Einfluss auf den Score des Items nehmen sollen. Diese sollen nämlich erst dann miteinbezogen werden, wenn der Score des Items hoch genug ist, um garantieren zu können, dass die behandelten Themen thematisch zu dem Chatverlauf passen. Dieser Wert kann als Parameter übergeben werden und muss explorativ je nach Anwendungsfall festgelegt werden. Sollte der Score des Items über dem festgelegten Wert liegen, wird der Beruf mit in die Berechnung des Scores einbezogen. Dieser Ablauf ähnelt dem der Themen stark. Zunächst werden wieder die Berufe des Chatverlauf mit denen der Items abgeglichen, um zu ermitteln, ob der jeweilige Beruf bei dem Chatverlauf und dem Item über dem zuvor festgelegten Limit liegt. Sollte dieser Fall eintreten, wird die obige Formel auch

auf das Label des Berufs angewandt und fließt damit auch in den Score des Items mit ein. Es gibt hier jedoch eine Besonderheit. Sollte ein Beruf mit in den Score einfließen, wird er doppelt gewichtet, da das Limit für die Berufe höher angesetzt wurde, als für die Labels und es somit eine höhere Gewichtung für den endgültigen Score erfordert. Im letzten Schritt müssen die drei Items mit den höchsten Scores ermittelt und absteigend sortiert werden.

7 Evaluation

Im Folgenden wird der Prototyp auf verschiedene Arten evaluiert, um seine Funktionalität und die Qualität der Empfehlungen zu prüfen. Zunächst wird erläutert, wie die genutzten Daten für die Evaluation gesammelt und aufbereitet wurden. Darauffolgend werden die drei verschiedenen Ansätze zur Evaluation vorgestellt. Die erste Evaluation erfolgt auf Basis der Erfüllung der im Abschnitt 5.2 festgelegten funktionalen und nicht-funktionalen Anforderungen. Im Anschluss erfolgt eine technische Evaluation, bei welcher der umgesetzte Prototyp mit zwei schon vorhandenen Recommender Systemen anhand verschiedener Metriken verglichen wird. Die technische Evaluation wird letztlich noch mit einer Evaluation durch Nutzende gestützt.

7.1 Daten und Systeme für die Evaluation

In diesem Abschnitt werden kurz und prägnant die Daten und Systeme vorgestellt, welche für die nachfolgende Evaluation verwendet wurden.

7.1.1 Grundlage der Daten für die Evaluation

Um das Recommender System evaluieren zu können, werden Chatdaten benötigt, für welche Empfehlungen ausgesprochen werden können. Für eine gut fundierte Evaluation ist es dafür maßgeblich, dass auch die Antworten des Chatbots miteinbezogen werden und der Chatbot dabei die Fragen der nutzenden Person zumindest thematisch passend beantwortet. Da dies der Chatbot Impuls noch nicht gewährleisten kann, kann dieser nicht für die Evaluation verwendet werden. Deshalb wurden 40 Chatverläufe für die Evaluation erzeugt, welche aus verschiedenen Quellen entstammen, auf welche im Folgenden genauer eingegangen wird.

Im Projekt vorhandene Chatdaten Vier der 40 Chatdaten konnten aus vorhandenen exemplarischen Chats aus dem Projekt „Pulsnetz“ übernommen werden. Diese wurden von Mitarbeitenden aus dem Projektteam in Kooperation mit Fachkräften im Rahmen eines Workshops erarbeitet. Als Fachkräfte werden in diesem Kontext Personen definiert, welche in einem medizinischen oder sozialen Bereich tätig sind.

Erstellung durch Fachkräfte Weitere vier Chatverläufe wurden von Fachkräften erzeugt. Hierzu haben zwei Fachkräfte zusammen einen Chatverlauf erarbeitet, indem eine Person die nutzende Person simuliert und die andere den Chatbot simuliert und somit auf die Frage der ersten Fachkraft antwortet.

Erstellung durch Nicht-Fachkräfte 15 Chats wurden, mit derselben Vorgehensweise wie bei den Fachkräften, von Personen erstellt, welche nicht im medizinischen oder sozialen Bereich tätig sind.

Extraktion aus Fachkräfte-Foren Die restlichen 17 Chatverläufe wurden mithilfe von Forum-Beiträgen erzeugt. Hierzu wurden Foren aufgesucht, in welchen sich Personen, welche im medizinischen oder sozialen Bereich arbeiten, austauschen können. Da ein Forums-Beitrag sich ähnlich einem Chatverlauf verhält, in welchem einer Person geholfen wird, konnten Konversationen aus diesen extrahiert werden. Die für die Evaluation erzeugten Chatverläufe stammen aus verschiedenen Foren für medizinische oder soziale Berufe[9].

Die gesammelten Chat-Daten wurden in ein JSON-Format gebracht, welches von dem Recommender System verarbeitet werden kann. Wichtig ist, anzumerken, dass es sich hierbei um unüberwachte Daten handelt. Das bedeutet, dass es keine Musterempfehlung als Vergleich gibt, anhand welcher eine Evaluation stattfinden könnte. Dies ist ein bekanntes Problem im Bereich der Evaluation von Recommender Systemen, welches in Abschnitt 5.1 aufgegriffen wurde.

7.1.2 Andere Systeme zum Vergleich

Um einen Vergleichswert zu schaffen, an welchem sich orientiert werden kann, wurde der umgesetzte Prototyp mit zwei schon bestehenden System verglichen. Die technische Evaluation in Abschnitt 7.4 und die Evaluation mit Nutzenden in Abschnitt 7.5 setzen den Prototyp und die zwei bestehenden Systeme in Bezug. Im Folgenden werden die zwei schon bestehenden Systeme kurz näher erläutert.

Das erste System, welches für den Vergleich herangezogen wird, ist ein TF-IDF-Retriever von Haystack. Dieser stellt eine Grundlage für die Informationssuche dar. Er sucht dabei nach Dokumenten, die lexikalische Überschneidungen mit der Suchanfrage aufweisen. Zudem weist er Wörtern, welche in der Gesamtheit in weniger Dokumenten vorkommen, eine größere Bedeutung zu, als Wörtern, welche in vielen Dokumenten vorkommen. Um eine Entscheidung zu treffen, welches Dokument empfohlen werden soll, wird für jedes Dokument ein TF-IDF-Score berechnet. [78] Der TF-IDF-Retriever wurde als Vergleichswert herangezogen, weil er eine gut fundierte Basis der Empfehlung von Dokumenten darstellt, welche in der Vergangenheit oftmals genutzt wurde.

Das zweite System, welches in den späteren Vergleich miteinbezogen wird, ist der Dense Passage Retriever von Haystack mit dem Modell „deepset/gbert-base-germandpr-question_encoder“ von deepset, welches von Hugging Face zur Verfügung gestellt wird. Das Dense Passage Retrieval ist eine Methode zum Suchen von Dokumenten, welche die Relevanz eines Dokumentes mithilfe von dichten Darstellungen ermittelt. Es nutzt jeweils ein BERT-Base-Modell zur Kodierung von Dokument und von Suchanfragen. Der Dense Passage Retriever nutzt zudem getrennte Kodierer

[9] Forum für Erziehende: https://www.forum-fuer-erzieher.de; Forum für medizinische Berufe: https://www.medi-jobs.de; Forum für medizinische Fragen: https://www.medizin-forum.de

für Dokumente und Abfragen. Diese Kodierer stellen die Modelle dar. Das Modell „deepset/gbert-base-germandpr-question_encoder“ wurde verwendet, da es auf die deutsche Sprache optimiert ist und in der vorliegenden Arbeit PDF-Dokumente in deutscher Sprache verarbeitet werden sollen. Der Dense Passage Retriever wurde als Vergleichswert verwendet, da er mit einem sehr neuen Stand der Technik arbeitet, indem er beispielsweise Modelle von BERT nutzt, welche mit Transformers arbeiten. Zudem wird er von Haystack, unter allen von deepset angebotenen Retrievern, empfohlen. [78, 85]

7.2 Evaluation anhand der Anforderungen

In dem vorliegenden Abschnitt wird geprüft, ob alle funktionalen und nicht-funktionalen Anforderungen, welche in Abschnitt 5.2 definiert wurden, erfüllt worden sind. Dies hat auch eine Aussagekraft darüber, ob die Anwendung korrekt und anhand der Vorgaben umgesetzt wurde. Die Tabelle 7.1 stellt eine Zusammenfassung der Ergebnisse der Evaluation nach Anforderungen zusammen, welche im Folgenden diskutiert werden.

Anforderung	**Grad der Erfüllung**
F1 - Grundlage der Empfehlung	vollständig erfüllt
F2 - Zielsetzung der Empfehlung	vollständig erfüllt
F3 - Zeitpunkt der Empfehlung	vollständig erfüllt
F4 - Kontextsensitivität	bedingt erfüllt
F5 - Empfehlungsmenge	vollständig erfüllt
F6 - Feedback	nicht erfüllt
N1 - Robustheit	bedingt erfüllt
N2 - Geschwindigkeit	nicht erfüllt
N3 - Konsistenz	siehe Abschnitt 7.5
N4 - Generisch	vollständig erfüllt

Tabelle 7.1: Zusammenfassung der Evaluation nach Anforderungen

7.2.1 Evaluation der funktionalen Anforderungen

Im Folgenden werden zunächst die funktionalen Anforderungen an den Prototypen aus Abschnitt 5.2.4 überprüft. Dafür werden diese nachfolgend mit ihren Überschriften aufgelistet, um sie direkt zu evaluieren.

F1 - Grundlage der Empfehlung Der implementierte Prototyp bezieht die Kontextdaten von allen ihm zur Verfügung gestellten Chatdaten mit ein. Aus diesen kann er, falls vorhanden, Themen und Berufe extrahieren und als Kontextdaten der anwendenden Person weiterverwenden. F1 ist damit vollständig erfüllt.

F2 - Zielsetzung der Empfehlung Der Prototyp spielt der anwendenden Person Empfehlungen aus, welche ihr weitere Themeneinblicke aufzeigen. F2 ist somit vollständig erfüllt.

F3 - Zeitpunkt der Empfehlung Der Prototyp spielt nur dann eine Empfehlung aus, wenn ein gewisser Grenzwert überschritten ist. Um diesen Grenzwert zu überschreiten, müssen genügend Themen und optional Berufe eine hohe Übereinstimmung mit den Textausschnitten aufweisen, damit die Empfehlung sinnvoll bleibt. F3 ist somit vollständig erfüllt.

F4 - Kontextsensitivität Der implementierte Prototyp ist nicht in der Lage alle Kontextdaten miteinzubeziehen und kann somit nicht alle Use Cases abbilden. Trotz dessen agiert er kontextsensitiv bei den Kontextdaten, mit welchen er umgehen kann. F4 ist damit nur bedingt erfüllt, da der Prototyp nur mit den bestimmten Kontextdaten umgehen kann.

F5 - Empfehlungsmenge Der Prototyp spielt bei Überschreitung des Grenzwertes genau drei Dokumente als Empfehlung an die anwendende Person aus. F5 ist somit vollständig erfüllt.

F6 - Feedback In den Prototypen wurde bisher keine Feedback-Funktion für die einzelnen ausgespielten Empfehlungen eingebaut. Somit ist F6 nicht erfüllt.

7.2.2 Evaluation der nicht-funktionalen Anforderungen

In diesem Abschnitt werden die nicht-funktionalen Anforderungen aus Abschnitt 5.2.4 evaluiert. Hierzu werden auch diese mit ihren Überschriften aufgelistet, um sie direkt evaluieren zu können.

N1 - Robustheit Der Prototyp weist bei der momentanen Datenmenge von circa 500 Dokumenten keine Ausfälle auf. Um diese Aussage auch für größere Datenmengen treffen zu können, müsste der Prototyp hierfür getestet werden. N1 ist somit bedingt erfüllt.

N2 - Geschwindigkeit Es wurden zehn Werte erfasst bei verschiedenen Längen an Chatverläufen, welche die Zeit bis zur Aussprache der Empfehlung widerspiegeln. Der Durchschnitt dieser Werte liegt bei 13,42 Sekunden. Somit ist N2 nicht erfüllt.

N3 - Konsistenz Um zu evaluieren, ob die Empfehlungen thematisch zu den Kontextdaten der anwendenden Person passen, wird im Folgenden eine technische Evaluation durchgeführt, welche zuletzt noch durch eine Evaluation mit Nutzenden gestützt wird. Ob N3 erfüllt ist, wird somit in den nächsten Abschnitten beantwortet.

N4 - Generisch Der Prototyp wurde so gebaut und konzipiert, dass es auch möglich ist, ihn für jegliche andere Themen, Items und Chatbots zu nutzen. Somit müssen immer nur kleine Änderungen vorgenommen werden, wie die Anpassung der URL in der Schnittstelle oder das Hinzufügen anderer Items. N4 ist somit vollständig erfüllt.

7.3 Metriken für die weiteren Evaluationen

Die technische Evaluation und die Evaluation mit Nutzenden werden mit denselben Metriken evaluiert. Deshalb werden diese im Folgenden kurz vorgestellt und es wird erläutert, wozu sie verwendet werden.

Arithmetischer Mittelwert Der arithmetische Mittelwert beschreibt die Addition aller n Variablen, welche letztlich durch n geteilt werden. In der vorliegenden Arbeit liegen die Daten als Urliste vor, somit ist jeder Ausprägungswert einzeln aufgeführt. Das arithmetische Mittel gibt jedoch keine Auskunft über die Streuung der Werte. [86]

Standardabweichung Die Standardabweichung stellt ein geeignetes Streuungsmaß dar, um die durchschnittliche Abweichung vom Durchschnitt anzugeben. [86] Die Standardabweichung ist die positive Wurzel aus der Varianz und wird in der gleichen Einheit gemessen, wie die tatsächlichen Werte. [87]

Box-Plot Der Box-Plot stellt eine grafische Methode zur Abbildung verschiedener Maßzahlen dar. Ein beispielhafter Box-Plot mit seinen einzeln Begrifflichkeiten wird in Abbildung 7.1 dargestellt. Der Median wird aus einer nach der Größe geordneten Liste gewonnen und beschreibt die Mitte dieser Liste. [86, 87] Das untere Quartil beschreibt die Grenze, unter welcher 25 % der Werte liegen. Äquivalent dazu beschreibt das obere Quartil die Grenze, oberhalb welcher sich 25 % der Werte befinden. Somit gibt der Interquartilsabstand die Spannweite an, in welcher sich die mittleren 50 % der Werte befinden. Der untere und der obere Whisker stellen das Minimum, bzw. das Maximum der Werte dar. Sie schließen Ausreißer allerdings aus. Ein Wert gilt als Ausreißer, sobald der Wert mehr als das 1.5-fache der Boxlänge unter- bzw. oberhalb der Quartilsgrenzen liegt. [86, 87]

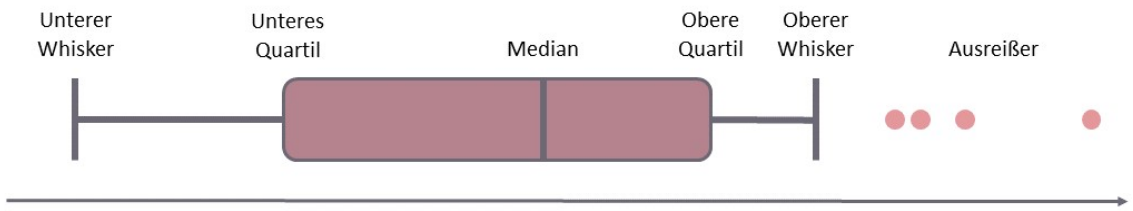


Abbildung 7.1: Aufbau eines Box-Plot nach [86]

Um die Ergebnisse hinsichtlich ihrer Signifikanz zu prüfen, wurde ein Wilcoxon-Mann-Whitney-Test für mehrere Ergebnisse durchgeführt. Dieser stellt einen nicht-parametrisierten Test dar. Der Test wird verwendet, um die aufgestellte Nullhypothese zu beweisen oder zu widerlegen. Die Nullhypothese für die folgenden Anwendungen des Wilcoxon-Mann-Whitney-Tests lautet H_0 = „Die Werte sind gleich.“ Der zuvor verwendete Begriff „Werte“ bezieht sich dabei immer auf das Ergebnis, welches geprüft werden soll. Das Signifikanzniveau α wird in dieser Arbeit auf 0.05 festgelegt. Diese Festlegung führt dazu, dass das Ergebnis statistisch signifikant ist, sobald $p < 0.05$ beträgt. [88, 89]

7.4 Technische Evaluation

Im Folgenden wurde eine technische Evaluation mit denen in Abschnitt 7.1.1 beschriebenen Evaluationsdaten durchgeführt. Die im Folgenden aufgeführte Evaluation macht den Hauptteil dieses Kapitels aus und wird im nächsten Abschnitt durch eine Evaluation mit Nutzenden gestützt.

Zunächst wird der Aufbau der technischen Evaluation beschrieben, um auf Basis dessen die verschiedenen Metriken anwenden zu können. Letztlich werden die Ergebnisse und die Auswertung der technischen Evaluation vorgestellt.

7.4.1 Aufbau der technischen Evaluation

Um Metriken anwenden zu können, wird zunächst ein Datensatz benötigt, welcher im richtigen Format vorliegen muss. Um diesen Zustand zu erreichen, wird ein Prozess durchlaufen, welcher in Abbildung 7.2 dargestellt wird.

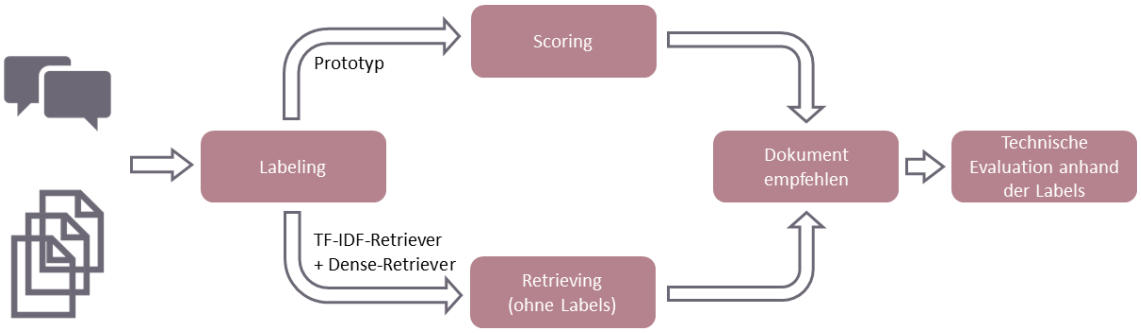


Abbildung 7.2: Ablauf des Aufbaus der technischen Evaluation

Hierzu wurden die Evaluationsdaten, welche in Abschnitt 7.1.1 beschrieben wurden, zusammen mit den Items, also den PDF-Ausschnitten, in das System gegeben. Die Items und die Chatdaten für die Evaluation wurden anschließend, wie in Abschnitt 6.5.4 beschrieben, gelabelt. Daraufhin wurden die gelabelten Daten in die drei verschiedenen Systeme, welche in Abschnitt 7.1.2 behandelt wurden, gegeben. Der implementierte Prototyp geht dabei wie in Abschnitt 6.5.5 ausgeführt vor und vergleicht die Chatdaten und Dokumente anhand der Labels und Scores und trifft damit die Entscheidung, welches Dokument empfohlen werden soll. Die beiden anderen Systeme berechnen ihre Empfehlung dabei unabhängig von den zugewiesenen Labels, das heißt sie beziehen diese nicht in ihre Entscheidung mit ein. Der Output zeigt daraufhin drei Empfehlungen, welche jeweils von jedem System ausgespielt wurden und welche im Folgenden anhand verschiedener Werte durch die zugewiesenen Labels und Berufe evaluiert werden können.

Die technische Evaluation gliedert sich in drei Bestandteile. Zunächst werden die Labels, welche dem Chatverlauf zugewiesen wurden, mit den Labels des jeweiligen Systems verglichen. Bei der Zuweisung der Labels, welche in Abschnitt 6.5.4 erläutert wurde, wurde festgelegt, dass 26 Labels jeweils dem Chatverlauf, sowie den empfohlenen Items angehängt werden. Somit lässt sich prüfen, wie viele der 26 Labels aus dem Chatverlauf auch bei dem empfohlenen Item annotiert wurden. Dabei wird zwischen den ersten drei Empfehlungen unterschieden und letztlich noch die Gesamtheit der drei Empfehlungen betrachtet. Der zweite Teil der Evaluation befasst sich mit der Annotation der Berufe an Chatdaten und Items. Dazu wird der Beruf des Chatverlaufs betrachtet, welcher am besten zu diesem passt und deshalb von der Zero-Shot-Classification an erster Stelle zugewiesen wurde. Nun wird geprüft, ob dieser Beruf auch an erster Stelle bei dem empfohlenen

Item liegt. Es wird sich hierbei absichtlich nur auf den ersten Beruf fokussiert, da die Annahme getroffen wird, dass Personen aus dem Gesundheits- und Sozialwesen, welche sich an den Chatbot wenden, vorrangig Fragen zu ihrem eigenen Beruf stellen, wodurch der Chatverlauf meist nur einen Beruf behandelt. Zuletzt wird noch nach dem erreichten Score bei dem Scoring, auf welches in Abschnitt 6.5.5 genauer eingegangen wurde, verglichen. Der Prototyp besitzt bereits einen Score, da durch diesen die Wahl der Empfehlung getroffen wurde. Für die drei ausgesprochenen Empfehlungen des TF-IDF-Retriever und des Dense-Retriever wird dieser Score nachträglich berechnet, um einen Vergleichswert zu schaffen.

7.4.2 Ergebnisse und Auswertung der technischen Evaluation

Im Folgenden wurden die zuvor in Abschnitt 7.4.1 genannten Bestandteile der Evaluation mit den Metriken aus Abschnitt 7.3 evaluiert.

Übereinstimmung der zugewiesenen Labels

Zunächst werden die Ergebnisse für die Übereinstimmung der annotierten Labels zwischen den Chatdaten und den empfohlenen Items vorgestellt. Die Tabelle 7.2 zeigt die Mittelwerte mit ihren Standardabweichungen für die ersten drei Empfehlungen der verschiedenen Systeme.

Empfehlung	**System**	**Mittelwert**	**Standardabweichung**
1. Empfehlung	Prototyp	13.28	3.71
	Dense-Retriever	11.20	4.50
	TF-IDF-Retriever	11.28	4.91
2. Empfehlung	Prototyp	13.25	3.95
	Dense-Retriever	10.80	4.89
	TF-IDF-Retriever	11.03	4.35
3. Empfehlung	Prototyp	13.40	3.74
	Dense-Retriever	10.18	4.67
	TF-IDF-Retriever	10.63	4.66
Alle Empfehlungen	Prototyp	13.31	3.77
	Dense-Retriever	10.73	4.67
	TF-IDF-Retriever	10.98	4.61

Tabelle 7.2: Mittelwerte und Standardabweichungen für die Anzahl an übereinstimmenden Labels mit einer Spanne von 0 bis 26 übereinstimmenden Labels

Die ersten drei Empfehlungen beschreiben dabei die drei Empfehlungen, welche den höchsten Score hatten und deshalb an den Chatverlauf annotiert wurden, bzw. bei dem Prototyp an die nutzende Person ausgespielt werden. Die erste Empfehlung passt laut dem jeweiligen System immer am besten zu dem Chatverlauf. Die Zeile „Alle Empfehlungen“ beschreibt den Mittelwert und die Standardabweichung, wenn alle drei Empfehlungen des jeweiligen Systems zusammen betrachtet werden. Die maximale Übereinstimmung der Labels, welche erreicht werden kann, beträgt 26 Labels. Der Mittelwert und die Standardabweichung für die erste, zweite und dritte Empfehlung setzt sich aus jeweils 40 Werten zusammen, was daraus resultiert, dass 40 Chatverläufe zur Eva-

luation zur Verfügung stehen. Demnach setzen sich der Mittelwert und die Standardabweichung der Zeilen „Alle Empfehlungen“ aus 120 Werten zusammen. Der Prototyp hat im Durchschnitt bei allen Empfehlungen einen Wert von 13.31 erreicht und hat damit im arithmetischen Mittel die höchste Übereinstimmung der Labels von den drei Systemen. Der Dense-Retriever und der TF-IDF-Retriever schnitten bei den Mittelwerten in der Zeile mit allen Empfehlungen ähnlich ab, wobei der TF-IDF-Retriever im Schnitt einen etwas höheren Mittelwert aufweist. Die Standardabweichung fällt bei dem Prototyp immer am niedrigsten aus, mit einer Abweichung von 3.77 bei allen Empfehlungen. Daraus lässt sich schließen, dass die Anzahl der übereinstimmenden Labels bei dem Prototyp am stabilsten sind.

Der Box-Plot in Abbildung 7.3 gibt einen grafischen Überblick über die Übereinstimmung der Labels bei den verschiedenen Systemen.

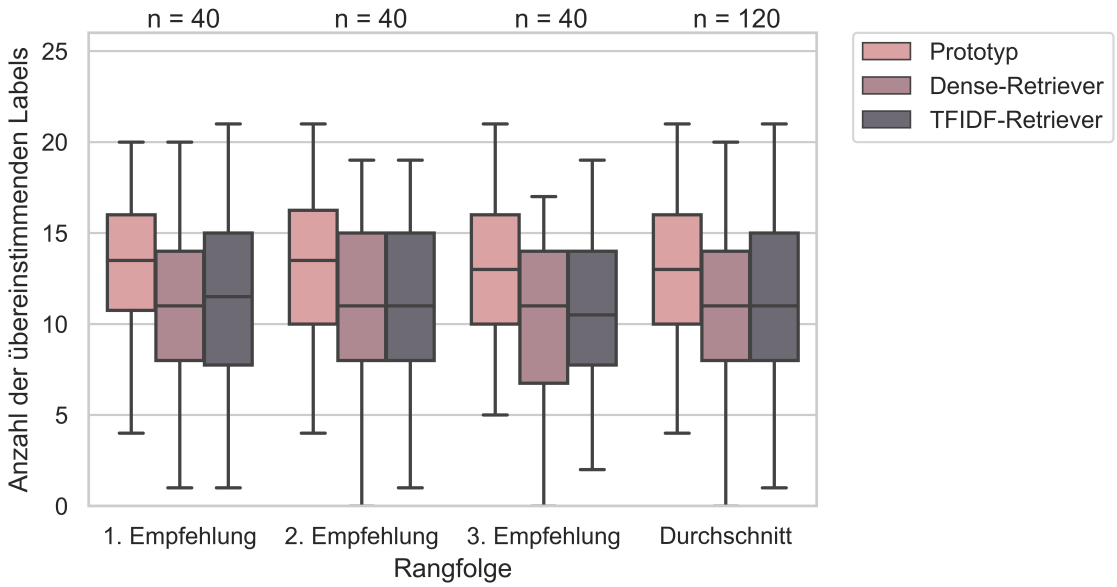


Abbildung 7.3: Box-Plot für die Anzahl der übereinstimmenden Labels

Auf der y-Achse ist die erreichte Anzahl der 26 maximal möglichen Labels dargestellt, wobei die x-Achse die einzelnen Empfehlungen aufgliedert. Das n über den gebündelten Balken steht für die Anzahl der Werte dieser. Der schwarze Strich in den farbig gefüllten Balken stellt den Median jedes Balkens dar. Dieser ist bei dem Prototyp bei allen Konstellationen am höchsten und fällt bei dem Dense-Retriever und bei dem TF-IDF-Retriever, mit einer leichten Schwankung in der dritten Empfehlung, sehr ähnlich aus. Das Diagramm zeigt zudem, dass der Interquartilsabstand bei dem Prototyp am geringsten ausfällt, was bedeutet, dass die mittleren Werte sich auf einen kleineren Bereich bündeln. Keines der drei Systeme erreichte zudem mehr als 22 Labels als Maximum. Es gibt außerdem bei keinem der Systeme einen Ausreißer, welcher aus dem unteren oder oberen Whisker fällt.

Um die Nullhypothese H_0 „Die Verteilungen der Werte, welche die Anzahl der übereinstimmenden Labels abbilden, sind gleich“ zu prüfen, wird der Wilcoxon-Mann-Whitney-Test durchgeführt. Da in diesem Kapitel der Prototyp anhand von vergleichbaren Systemen evaluiert werden soll, werden die Werte des Prototyps jeweils mit den zwei anderen Systemen verglichen. So entsteht ein p-Wert für Prototyp-Dense und ein p-Wert für Prototyp-TF-IDF. Da die Hypothese auch für die einzelnen Empfehlungen geprüft werden soll, wird für diese der Wilcoxon-Mann-Whitney-Test einzeln durchgeführt. Die Tabelle 7.3 stellt die p-Werte der jeweiligen Empfehlung, mit dem jeweiligen System, dar.

System/Empfehlung	**Prototyp-Dense**	**Prototyp-TF-IDF**
1. Empfehlung	$p < 0.05$	$p >= 0.05$
2. Empfehlung	$p < 0.05$	$p < 0.05$
3. Empfehlung	$p < 0.05$	$p < 0.05$
Alle Empfehlungen	$p < 0.05$	$p < 0.05$

Tabelle 7.3: P-Werte des Wilcoxon-Mann-Whitney-Tests für die Übereinstimmung der Labels

Die P-Werte sagen dabei aus, wie wahrscheinlich die Nullhypothese H_0 ist. Da α auf 0.05 festgelegt wurde, wird H_0 verworfen, wenn p unter dem Wert 0.05 liegt. In diesem Fall greift die Alternativhypothese H_A, welche besagt, dass die Werte unterschiedlich sind. H_A ist in diesem Fall für alle Werte außer für die erste Empfehlung zwischen dem Prototyp und dem TF-IDF-Retriever statistisch signifikant.

Übereinstimmung des ersten zugewiesenen Berufs

Im Folgenden werden die Ergebnisse des zweiten Teils der Evaluation betrachtet. Dabei wurde verglichen, ob der am besten zu dem Chatverlauf passende Beruf auch an erster Stelle bei der jeweiligen Empfehlung steht. Abbildung 7.4 stellt das arithmetische Mittel für die Optionen dar.

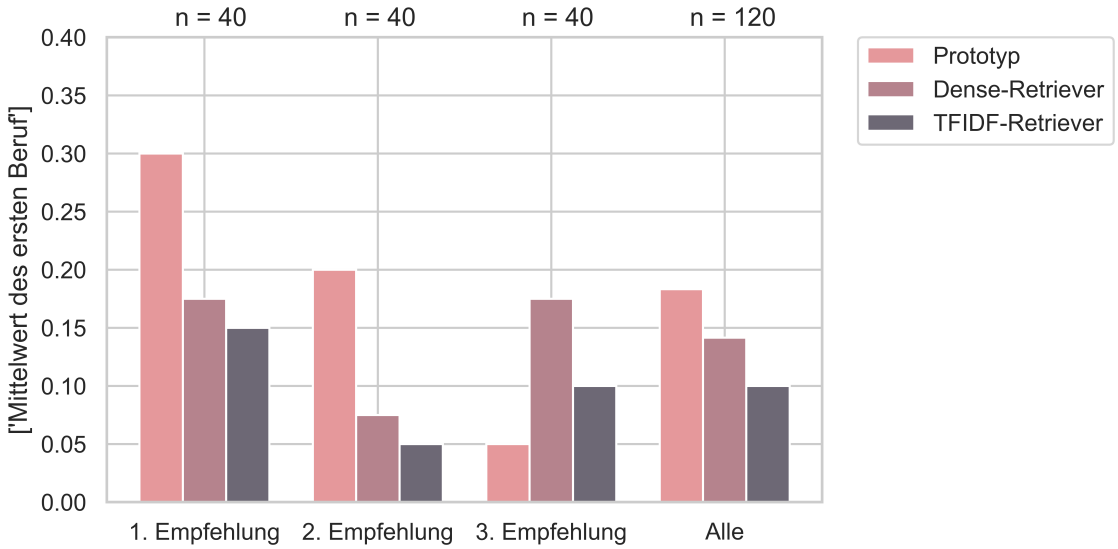


Abbildung 7.4: Mittelwert für die Übereinstimmung des ersten Berufs

Der Mittelwert kann theoretisch zwischen null und eins liegen, da maximal der erste Beruf bei beiden übereinstimmen kann. Da der arithmetische Mittelwert in der Abbildung den Wert 0.3 allerdings nicht überstiegen hat, wurde die Skala auf der y-Achse dementsprechend angepasst. Die Betrachtung zeigt, dass der Prototyp bei der ersten, der zweiten und bei allen Empfehlungen insgesamt die höchste Übereinstimmung des Berufes aufweist, gefolgt von dem Dense-Retriever. Die dritte Empfehlung fällt dabei aus dem Schema, bei dieser weist der Prototyp die geringste Übereinstimmung des Berufes auf. Eine Theorie, weshalb dies der Fall sein könnte, ist der Algorithmus hinter dem Scoring des Prototyps, welches in Abschnitt 6.5.5 transparent aufgeführt wurde. Dabei wird der Beruf nur miteinbezogen, sollte das Thema des Items auch zu dem Thema des Chatverlaufs passen, da dieses eine wichtigere Rolle für die Empfehlung spielt, als der Beruf. Sollten also zu wenig Items vorliegen, bei welchen sowohl Thema als auch Beruf gut passen, wird der Beruf gewollt nicht mehr beachtet. Dieses Szenario tritt vor allem bei der dritten Empfehlung ein, da die Items, bei welchen Thema und Beruf übereinstimmen, schon vergeben sind.

Die Standardabweichungen zu den Werten für die Übereinstimmung des ersten Berufs, fallen bei allen Empfehlungen bei dem Prototyp am höchsten aus, gefolgt von dem Dense-Retriever. Eine mögliche Begründung hierfür könnte auch die obige Theorie sein, um dies zu beweisen, müsste die Evaluation mit mehr Items durchgeführt werden. Damit könnte dann verglichen werden, ob die Standardabweichung des Prototyps dann sinkt. Die Standardabweichung für „Alle“ beträgt bei dem Prototypen 0.38, bei dem Dense-Retriever 0.35 und bei dem TFIDF-Retriever 0.3. Es lässt sich auch bei den Standardabweichungen ein abweichendes Muster bei der dritten Empfehlung erkennen, bei dieser hat der Dense-Retriever die höchste Standardabweichung gefolgt von dem TF-IDF-Retriever. Daraus lässt sich erkennen, dass die Standardabweichung dann höher ist, wenn auch der Mittelwert höher ist. Eine Vermutung wieso dem so ist, könnte sein, dass der Spielraum bei konstant niedrigen Werten geringer ist, als bei niedrigen Werten gemischt mit hohen Werten.

Im Folgenden wird die Nullhypothese H_0 „Die Verteilungen der Werte, welche die Übereinstimmung des ersten Berufs abbilden, sind gleich.“ wieder mit dem Wilcoxon-Mann-Whitney-Test geprüft. Die Tabelle 7.4 zeigt wieder die p-Werte der einzelnen Kombinationen auf.

System/Empfehlung	**Prototyp-Dense**	**Prototyp-TF-IDF**
1. Empfehlung	p >= 0.05	p >= 0.05
2. Empfehlung	p >= 0.05	p < 0.05
3. Empfehlung	p >= 0.05	p >= 0.05
Alle Empfehlungen	p >= 0.05	p >= 0.05

Tabelle 7.4: P-Werte des Wilcoxon-Mann-Whitney-Tests für die Übereinstimmung des ersten Berufs

Hieraus lässt sich erkennen, dass fast keiner der p-Werte bei dem paarweisen Vergleich von dem Prototyp und dem Dense-Retriever unter 0.05 liegt. Deshalb wird die Nullhypothese H_0 nicht verworfen und die Alternativhypothese H_A greift nicht, weshalb mit keiner statistischen Signifikanz behauptet werden kann, dass die Werte nicht gleich sind.

Scores der einzelnen Empfehlungen

Zuletzt werden noch die Ergebnisse des Vergleichs der Scores der Items betrachtet. Die Ergebnisse von diesem sind in dem Box-Plot in Grafik 7.5 dargestellt.

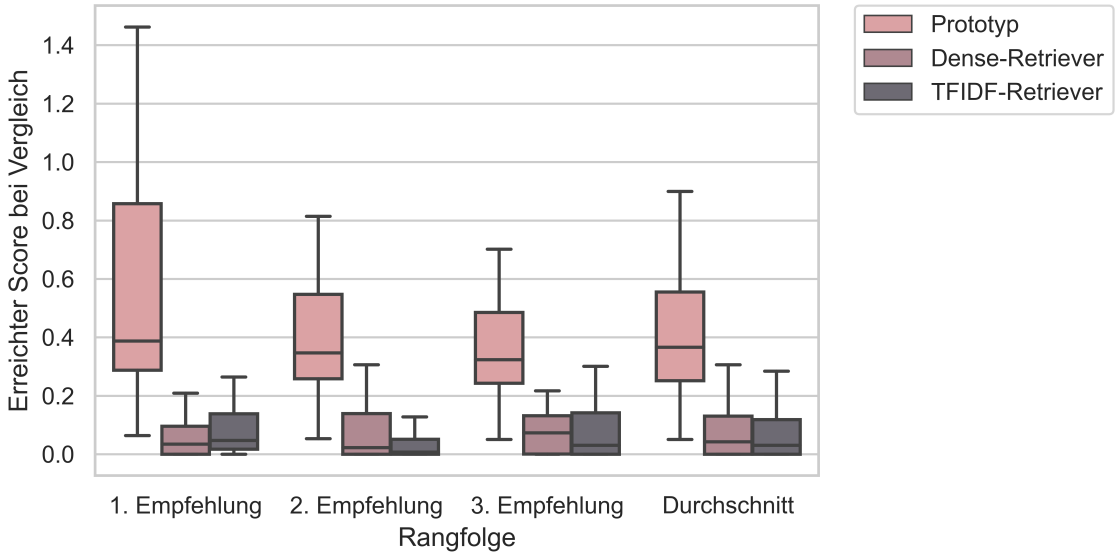


Abbildung 7.5: Box-Plot für die Scores der Empfehlungen

Der Box-Plot wird in der Grafik 7.5 ohne Ausreißer dargestellt, um eine Übersichtlichkeit zu gewährleisten. Der gleiche Box-Plot mit den Ausreißern befindet sich, um Transparenz und Vollständigkeit zu garantieren, im Anhang A.4. Auf dem Box-Plot wird deutlich, dass der Median des Prototyps deutlich größer ist als der Median des Dense-Retriever und des TF-IDF-Retriever. Es wird die Annahme getroffen, dass dies unter anderem der Fall ist, da der Algorithmus des Scorings für den Prototyp geschrieben und für diesen optimiert wurde. Dass der Interquartilsabstand des Prototyps so viel größer ist, als der der anderen zwei Systeme, kann damit begründet werden, dass die beiden anderen Systeme sich am unteren Rand der Werte-Skala bewegen und somit eine kleinere Streuung haben. Auch hier könnte wieder der kleine Datensatz, bei welchem nicht immer ein Item gefunden werden kann, bei welchem Thema und Beruf passen, eine Rolle spielen. Denn sollte der Fall eintreten, dass bei einem Item mit passendem Thema auch der Beruf passt, wird der Score stark erhöht, damit dieses Item auch empfohlen wird.

Auch hier wird die Nullhypothese H_0 geprüft. Diese lautet „Die Verteilungen der Werte, welche den Score der jeweiligen Empfehlung abbilden, sind gleich.“. Die Tabelle 7.5 zeigt wieder die p-Werte nach dem Wilcoxon-Mann-Whitney-Test auf. Da der Wert p für jede Berechnung unter 0.05 liegt, kann H_0 verworfen werden. Somit greift die Alternativhypothese H_A, welche besagt, dass die Werte nicht gleich sind. Die Alternativhypothese H_A ist statistisch signifikant.

Empfehlung/System	Prototyp-Dense	Prototyp-TF-IDF
1. Empfehlung	$p < 0.05$	$p < 0.05$
2. Empfehlung	$p < 0.05$	$p < 0.05$
3. Empfehlung	$p < 0.05$	$p < 0.05$
Alle Empfehlungen	$p < 0.05$	$p < 0.05$

Tabelle 7.5: P-Werte des Wilcoxon-Mann-Whitney-Tests für die Scores der Empfehlungen

7.5 Evaluation mithilfe von Nutzenden

Im Folgenden wird eine Evaluation mit Nutzenden vorbereitet, durchgeführt und ausgewertet. Diese dient dazu, die technische Evaluation zu stützen und im besten Fall ähnliche Ergebnisse zu erzielen.

7.5.1 Aufbau und Methode der Umfrage

Die Umfrage wurde in Form eines Fragebogens aufgebaut, welcher dann an Personen versendet wurde. Der Fragebogen wurde folgendermaßen aufgebaut:

Frage zu dem Beruf der befragten Person Zuerst wurde die Person nach ihrem Beruf gefragt. Sie hatte hierbei die Auswahl zwischen „Medizinischer/pädagogischer Bereich“ und „Keiner der genannten Bereiche“. Je nach Antwort kann die Person dann der Gruppe „Fachkräfte“ oder „Nicht-Fachkräfte“ zugeordnet werden.

Chatverlauf Im Folgenden werden acht Chatverläufe an die befragte Person ausgespielt, welche aus den 40 Chatdaten für die Evaluation stammen. Diese sollen von der befragten Person gelesen werden.

Empfohlene Textausschnitte Nach jedem Chatverlauf werden der befragten Person drei Empfehlungen ausgespielt. Diese setzen sich aus der jeweils ersten Empfehlung jedes Systems zu dem jeweiligen Chatverlauf zusammen. Es wurde sich dafür entschieden, für die Evaluation mit Nutzenden jeweils nur die erste Empfehlung zu evaluieren, da es sonst den zeitlichen Rahmen des Fragebogens überschritten hätte. Die drei Textausschnitte sollten wieder von der befragten Person genau gelesen werden.

Frage zur Bewertung der Textausschnitte Danach folgen drei Fragen zur Bewertung der drei Textausschnitte. Dazu wurde die Frage „Passt Textausschnitt [X] thematisch zu dem obigen Chatverlauf?“ für jeden Textausschnitt gestellt. Die Textausschnitte wurden dabei mit den Buchstaben A, B und C nummeriert. Das „[X]“ in der Frage soll als beispielhafter Platzhalter in der Frage dienen. Die Antwortmöglichkeiten für diese Frage wurden als Likert-Skala dargestellt. Die Likert-Skala stellt eine lineare Skala mit den Werten eins bis fünf dar und wird in dieser Arbeit als unipolare Skala verwendet. Das bedeutet, dass lediglich eine Emotion in unterschiedlichen Ausprägungen bewertet wird. [12] In den erstellten Fragebögen steht eins dabei immer für „passt überhaupt nicht“ und fünf für „passt voll und ganz“.

Um zu garantieren, dass die Befragten nicht erahnen können, welcher Textausschnitt von welchem System empfohlen wurde, wurden die Textausschnitte für jeden Chatverlauf in einer neuen zufälligen Reihenfolge ausgespielt. Es den Bearbeitungszeitraum eines Fragebogens zwischen zehn und 15 Minuten zu halten, damit möglichst viele Personen an der Umfrage teilnehmen. Deshalb wurden die 40 Chatdaten zur Evaluation auf fünf Fragebögen aufgeteilt. Jeder Fragebogen enthält somit acht Chatverläufe, zu welchen jeweils drei Textausschnitte bewertet werden müssen. Jede befragte Person hat nur einen Fragebogen erhalten, sodass letztlich jede Person, welche teilgenommen hat, acht Chatverläufe mit ihren jeweiligen Empfehlungen bewertet hat. Jeder der fünf Fragebögen wurde schlussendlich von drei Fachkräften und sechs Nicht-Fachkräften bearbeitet und ausgefüllt. Damit ergaben sich 360 Werte pro System.

7.5.2 Ergebnisse und Auswertung des Fragebogens

Die Ergebnisse der Fragebögen wurden, wie auch die technische Evaluation, mit den Metriken aus Abschnitt 7.3 ausgewertet. Zunächst wird der arithmetische Mittelwert betrachtet. Dieser wird in Abbildung 7.6 dargestellt.

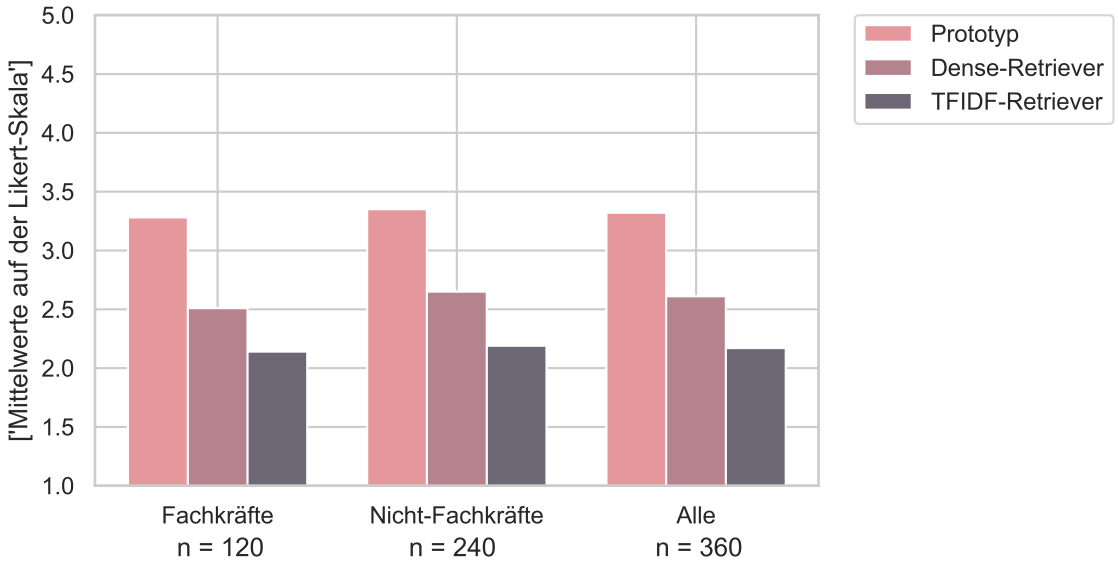


Abbildung 7.6: Mittelwerte der Bewertungen von Empfehlungen anhand der Likert-Skala

Auf der y-Achse werden die Werte null bis fünf der Likert-Skala dargestellt und es wird das arithmetische Mittel auf dieser Achse angegeben. Auf der x-Achse werden die Balken nach Personengruppe gebündelt. Diese unterscheiden sich in Fachkräfte und Nicht-Fachkräfte, und am Ende wird noch der Mittelwert von beiden Gruppen in ihrer Gesamtheit betrachtet. Unter den Personengruppen ist die jeweilige Stichprobengröße mit der Variablen n angegeben. Rechts befindet sich die Legende, welche nach Farben unterscheidet und die verschiedenen Systeme aufzeigt. Es wird deutlich, dass der Mittelwert des Prototyps auf der Likert-Skala bei allen Personengruppen am höchsten ist. Darauf folgt der Dense-Retriever, wobei der TF-IDF-Retriever bei allen Personengruppen am schlechtesten bewertet wurde. Es wird auch deutlich, dass sich die Werte zwischen

den verschiedenen Personengruppen kaum merkbar unterscheiden. Es wird die Theorie aufgestellt, dass dies der Fall ist, weil für eine Bewertung der Textausschnitte hauptsächlich Lesekompetenz gefordert wird, um zu erkennen, ob die Themen und Berufe übereinstimmen. Da diese Kompetenz keinem bestimmten Bereich an Berufen zugeschrieben werden kann, könnte dies eine mögliche Erklärung darstellen. Die Standardabweichungen sind ebenfalls sehr ähnlich zwischen den verschiedenen Personengruppen, was in der Tabelle 7.6 mit den Standardabweichungen für die einzelnen Gruppen und Systeme deutlich wird.

	Prototyp	**Dense**	**TF-IDF**
Fachkräfte	3.28	2.51	2.14
Nicht-Fachkräfte	3.35	2.65	2.19
Alle	3.32	2.19	2.17

Tabelle 7.6: Vollständige Standardabweichungen der Werte der Likert-Skala für die Systeme

Allerdings zeigt sich, dass die Standardabweichung der Fachkräfte bei allen drei Systemen knapp unter der Standardabweichung der Nicht-Fachkräfte liegt. Daraus kann die Vermutung entstehen, dass sich die Fachkräfte etwas sicherer in ihrer Bewertung waren, als die Nicht-Fachkräfte. Allerdings ist der Unterschied zu minimal, um hierzu eine statistisch signifikante Aussage zu treffen. Es zeigt sich jedoch, dass die Standardabweichung der Werte des Prototyps am höchsten ist, gefolgt von dem Dense-Retriever. Diese Reihenfolge spiegelte sich auch im arithmetischen Mittel wider und führt zu der Annahme, dass die Standardabweichung höher wird, wenn die Mittelwerte eher in der Mitte der Skala liegen, da dann mehr Spielraum nach oben und unten ist. Bei dem TF-IDF-Retriever, welcher eine geringe Standardabweichung aufweist, ist auch der Mittelwert sehr niedrig. Diese Theorie bekräftigt sich nochmals, bei Betrachtung des Box-Plots in Abbildung 7.7.

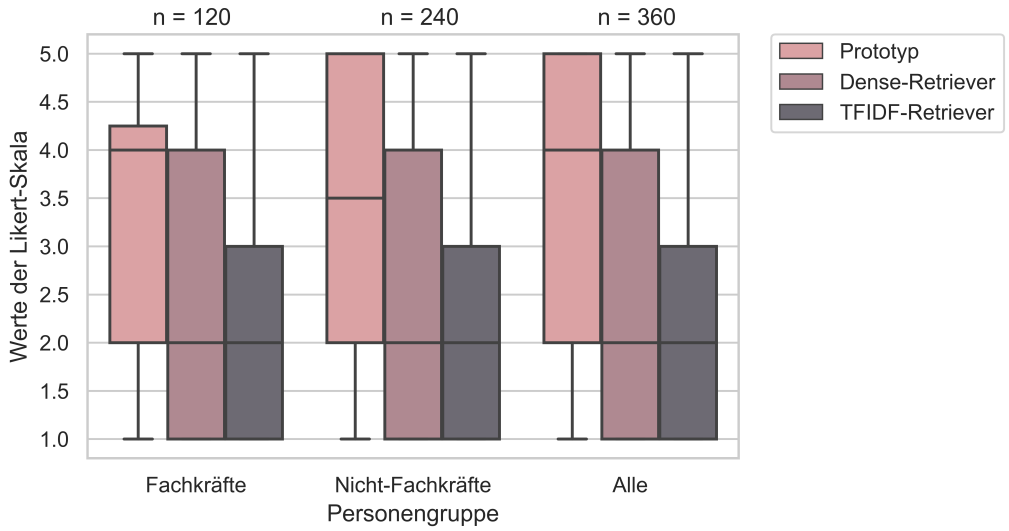


Abbildung 7.7: Box-Plot der Bewertungen von Empfehlungen anhand der Likert-Skala

Hier zeigt sich, dass sowohl bei dem Dense-Retriever, als auch bei dem TF-IDF-Retriever das untere Quartil an der niedrigsten Bewertung, welche für „passt überhaupt nicht“ steht, anliegt. Auch

hier unterscheiden sich die Ergebnisse kaum zwischen Fachkräften und Nicht-Fachkräften, lediglich bei dem Prototyp gibt es leichte Unterschiede. Der Median der Werte des Prototyps liegt bei den Fachkräften bei 4 wohingegen er bei den Nicht-Fachkräften bei 3.5 liegt. Dafür ist das obere Quartil bei den Fachkräften bei dem Prototyp niedriger, als bei den Nicht-Fachkräften. Generell weist das Ergebnis der Nicht-Fachkräfte bei dem Prototyp einen größeren Interquartilsabstand auf, was bedeutet, dass die mittleren Daten eine breitere Streuung haben, als bei den Fachkräften.

Um die Nullhypothese H_0 „Die Verteilungen der Werte auf der Likert-Skala, welche die Bewertung des Textausschnitts der Befragten widerspiegeln, sind gleich.“ zu prüfen, wird wieder ein Wilcoxon-Mann-Whitney-Test durchgeführt. Die daraus resultierenden p-Werte werden in Abbildung 7.7 abgebildet.

Personengruppe/System	**Prototyp-Dense**	**Prototyp-TF-IDF**
Fachkräfte	$p < 0.05$	$p < 0.05$
Nicht-Fachkräfte	$p < 0.05$	$p < 0.05$
Alle	$p < 0.05$	$p < 0.05$

Tabelle 7.7: P-Werte des Wilcoxon-Mann-Whitney-Tests für die Bewertungen auf der Likert-Skala

Es zeigt sich, dass jeder der p-Werte unter einem Wert von 0.05 liegt. Somit wird die Nullhypothese H_0 verworfen und die Alternativhypothese H_A „Die Verteilungen der Werte auf der Likert-Skala, welche die Bewertung des Textausschnitts der Befragten widerspiegeln, sind nicht gleich.“ greift. Die Alternativhypothese H_A ist statistisch signifikant.

Durch dieses Ergebnis kann die nicht-funktionale Anforderung N3 - Konsistenz aus Abschnitt 7.2 als vollständig erfüllt angesehen werden.

8 Diskussion

In diesem Kapitel werden die Ergebnisse dieser Arbeit kritisch reflektiert. Es werden die wichtigsten Erkenntnisse für das vorliegende Problem und die zuvor definierten Forschungsfragen aufgearbeitet. Zunächst wird dabei auf den theoretischen Beitrag der Arbeit eingegangen, welcher von dem praktischen Beitrag ergänzt wird. Zuletzt werden die Limitationen der vorliegenden Arbeit behandelt, woraufhin ein Ausblick für zukünftige Forschungen gegeben wird.

8.1 Theoretischer Beitrag

Die vorliegende Arbeit liefert Wissen für das Design eines Recommender System. Dieses ist spezialisiert auf Chatbots im Gesundheitswesen und kann kontextsensitiv konzipiert werden. Hierfür wurden mit den verschiedenen Use Cases alle relevanten Kontextdaten aufgearbeitet, welche auf verschiedene Weisen zusammenspielen und die Empfehlung damit beeinflussen können. Dieses konzeptionelle Design ist der theoretische Beitrag dieser Arbeit.

Diese Arbeit zielte darauf ab, einer nutzenden Person eine möglichst passende Empfehlung zu ihrem vorherigen Chatverlauf auszuspielen. Dazu konnten verschiedene Kontextdaten miteinbezogen werden, welche unterschiedlich verarbeitet wurden. Das konzeptionelle Design, welches die Kontextdaten und die einzelnen Komponenten mit ihren Schnittstellen beschreibt, ist das hauptsächliche Ergebnis des theoretischen Beitrags und leitet sich aus den vorherigen Arbeitsschritten ab. Zunächst wurden Fachkräfte befragt, um einen Überblick über den theoretischen Teil und die verschiedenen Themen und Komponenten zu schaffen. Daraufhin wurde eine umfangreiche Literaturanalyse durchgeführt. Auf Basis dieser und mithilfe der Unterlagen des Partner-Projekts „Pulsnetz“ konnten Personas erarbeitet werden, aus welchen Use Cases abgeleitet wurden. Anhand der Literaturanalyse und den Use Cases wurden letztlich noch Anforderungen definiert, welche zusammen mit den anderen Schritten die Basis für das konzeptionelle Design gebildet haben. Im Folgenden wird Bezug zu den in Kapitel 1 gestellten Forschungsfragen genommen.

Forschungsfrage 1: *Welche Arten von Recommender Systemen gibt es und welche Vorteile bringen diese jeweils mit sich?*
Diese Frage wurde in den Abschnitten 3.3.3 und 3.3.4 beantwortet. Dabei kristallisierte sich heraus, dass es verschiedenste Arten von Recommender Systemen gibt. Die zentralen Arten von Recommender Systemen wurden dabei in Tabelle 3.2 dargestellt. Es kann zusammenfassend festgehalten werden, dass vor allem zwischen inhaltsbasierten und kollaborativen Recommender Systemen unterschieden wird. Die inhaltsbasierten Recommender Systeme bieten sich vor allem dann

an, wenn wenige Daten zur Verfügung stehen, während die kollaborativen Recommender Systeme bei großen Datenmengen sehr effektiv sind. Prinzipiell ist die kollaborative Filterung die fortgeschrittenere Technik, welche bei großen Datenmengen bessere Ergebnisse erzielt. Sollte ein System entwickelt werden, welches nur anfangs wenig Daten zur Verfügung hat, kann auf ein hybrides System zurückgegriffen werden, welches durch den Wechsel zwischen verschiedenen Arten von Recommender Systemen besonders flexibel ist.

Forschungsfrage 2: *Wie sollte ein Recommender System im Gesundheitswesen mit sensiblen Daten umgehen?*
Der Umgang mit sensiblen Daten bei Recommender Systemen stellt vor allem im Gesundheitswesen eine große Herausforderung dar. Die Thematik wurde in Abschnitt 5.1.3 aufgegriffen und erläutert. Hierzu werden klare gesetzliche Vorgaben benötigt, welche der nutzenden Person transparent vermittelt werden müssen. Es sollte der nutzenden Person freistehen zu entscheiden, welche Daten sie von sich preisgibt, weshalb transparent gestaltet werden sollte, wann und welche Daten der nutzenden Person erfasst und gespeichert werden. Aufgrund des begrenzten Umfangs der vorliegenden Arbeit wurde diese Thematik nicht vollständig und umfänglich aufgegriffen.

Forschungsfrage 3: *Welche Möglichkeiten gibt es zur Implementierung eines kontextsensitiven Recommender Systems?*
Mit dieser Frage wurde sich in Kapitel 5 genauer befasst. Dabei wurden durch verschiedene Use Cases Möglichkeiten aufgezeigt, wie unterschiedliche Arten von Recommender Systemen umgesetzt werden können. Besonders flexibel war dabei das hybride, gewichtete Recommender System, bei welchem die Erkenntnisse aller Kontextdaten, gewichtet miteinbezogen werden konnten. Hierbei wurde auch die Möglichkeit der kollaborativen Filterung in Erwägung gezogen, jedoch wurde dies wieder verworfen, da die Datenlage hierfür zu marginal war. In der Implementierung wurde letztlich ein gewichtetes hybrides inhaltsbasiertes Recommender System umgesetzt, welches zunächst die Themen der Items miteinbezieht und, falls der Score hoch genug ist, auch die Berufe mit in die Entscheidung der Empfehlungsgebung miteinbezieht.

Forschungsfrage 4: *Welche Kontextdaten erweisen sich am nützlichsten für die Verwendung von kontextabhängigen Empfehlungen und wie kann dies am besten umsetzt werden?*
Diese Forschungsfrage wurde über die ganze Arbeit hinweg behandelt und stellt den zentralen Punkt der vorliegenden Arbeit dar. Es stellte sich heraus, dass sich konzeptionell vor allem demografische Daten, sowie Feedback der nutzenden Person als hilfreich erwiesen. In der Umsetzung waren vor allem die Themen der Items, sowie die aktuelle Frage der anwendenden Person und der vorhandene Chatverlauf von besonderer Bedeutung. Auch das Einbeziehen des Berufs der anwendenden Person hat die Empfehlungen maßgeblich verbessert. Andere demografische Daten, wie das Alter erwiesen sich zunächst als weniger hilfreich, da die meisten Items bei diesem Anwendungsfall keinen Bezug zu einem präferierten Alter der Leserschaft angaben.

8.2 Praktischer Beitrag

Der fundamentale praktische Beitrag dieser Arbeit ist der entwickelte Prototyp. Dieser zeigt auf, welche Möglichkeit es zur Implementierung eines kontextsensitiven Recommender Systems für einen Chatbot gibt. Ein weiterer wichtiger praktischer Beitrag sind die zugehörigen Ergebnisse der Evaluation, welche den implementierten Prototypen in Relation zu schon bestehenden System setzen. Die Ergebnisse dieser deuten darauf hin, dass die erste Empfehlung des Prototyps thematisch signifikant besser passt, als die der Vergleichssysteme. Auch in der technischen Evaluation schnitt der Prototyp für die ersten drei Empfehlungen signifikant besser bei der Übereinstimmung der Labels ab.

Das implementierte System dient als Prototyp, welcher für den spezifischen Use Case des Projekts „Pulsnetz“ entwickelt wurde. Er kann jedoch durch seine generischen Merkmale, sowohl für andere Chatbots, als auch für andere Wissensgebiete genutzt werden. Um ihn in den tatsächlichen Einsatz zu bringen, müssten Änderungen vorgenommen werden, auf welche im folgenden Abschnitt genauer eingegangen wird.

8.3 Limitationen und Ausblick

Die vorliegende Arbeit weist verschiedene Beschränkungen auf, welche in weiteren Studien und Forschungsarbeiten verbessert und optimiert werden könnten. Die Beschränkungen und Einbringungen beziehen sich im Folgenden auf die Recherche, den implementierten Prototypen sowie auf die Evaluation von diesem.

Zunächst werden die möglichen Ausblicke und Limitationen in der Recherche betrachtet. Hier könnten durch Interviews mit Fachkräften im sozialen und medizinischen Bereich weitere Anforderungen an das Recommender System gewonnen werden. Dadurch könnte es noch spezifischer auf das geschlossene Wissensgebiet angepasst werden, vor allem was die Wahl und den Umgang mit den zu sammelnden Kontextdaten betrifft. Dadurch könnte auch das konzeptionelle Design, welches in Abschnitt 5.3 vorgestellt wurde, noch weiter spezifiziert und ausgearbeitet werden. Dies könnte sich wiederum positiv auf einen möglichen Prototyp sowie dessen Evaluation auswirken. Hier könnte zudem mehr Forschung in die optimale konzeptionelle Darstellung in der Oberfläche investiert werden, da diese maßgeblich für eine gute User Experience (UX) ist. Dazu könnten beispielsweise verschiedene Prototypen umgesetzt, evaluiert und verglichen werden. Somit könnte beispielsweise geprüft werden, bei welcher Oberfläche am häufigsten auf Empfehlungen geklickt wird.

Im Folgenden werden die Beschränkungen des implementierten Recommender System betrachtet und es werden Vorschläge für weitere Forschungsmöglichkeiten dazu geliefert. Dieses weist vor allem noch Limitationen in der Ansicht des UI auf. Dieses könnte entsprechend der konzeptionellen Darstellung in der Oberfläche aus Abschnitt 5.3.2 umgesetzt werden. Zudem sollten die

Rohdaten, welche momentan von dem Recommender System zurückgegeben werden, wieder als PDF-Ausschnitte ausgespielt werden. Dadurch erfährt die nutzende Person eine bessere UX und beschäftigt sich womöglich eher mit den empfohlenen Textausschnitten, da diese für sie leichter lesbar sind. Es könnte zudem untersucht werden, an welchen Stellen die PDF-Dokumente, welche als Input in das Recommender System kommen, am besten geteilt werden, damit Ausschnitte weder zu lange, noch aus dem Kontext gerissen wirken. Bezüglich des entwickelten Algorithmus für das Zuteilen der Scores zu den einzelnen Items könnten verschiedene weitere Ansätze getestet und evaluiert werden. Ein solcher Ansatz wäre beispielsweise zu prüfen, ob die Ergebnisse an Qualität gewinnen, wenn Nachrichten, welche neuer sind, höher gewichtet werden, als ältere Chat-Nachrichten. Zudem könnten weitere Kontextdaten, welche konzeptionell aufgestellt wurden, in den Prototypen implementiert werden, da diese auf ähnliche Weise wie die schon implementierten Kontextdaten eingebaut werden können und es hierfür wenig Änderungen an dem Quellcode vorgenommen werden müssen. Vor allem die Möglichkeit, den Chatbot zu nutzen, um weitere Kontextdaten der nutzenden Person durch Rückfragen des Chatbots zu erhalten, könnte ein interessanter Ansatz sein, an welchem weitergeforscht werden sollte. Auch der Zeitpunkt der Ausspielung der Empfehlung könnte noch weiter spezifiziert und letztlich auch in Form einer Evaluation bewertet werden.

Schlussendlich werden noch die Limitationen und der Ausblick in der Evaluation beleuchtet. Hierbei wäre es möglich, wie schon zuvor genannt, mehr Komponenten spezifischer in der Evaluation zu untersuchen. Beispielsweise könnte untersucht werden, ob ein Zusammenhang zwischen der Länge des Chatverlaufs und der Qualität der Empfehlung besteht. Des Weiteren könnte die Evaluation mit Nutzenden mit einer größeren Stichprobe durchgeführt werden, auch wenn die Ergebnisse statistisch schon signifikant waren. Es wäre interessant zu ermitteln, ob sich dann signifikante Unterschiede zwischen den Personengruppen zeigen. Zudem könnte die Evaluation mit Nutzenden realistischer durchgeführt werden, wenn pro Nutzendem nur ein Chatverlauf bewertet wird und dieser im besten Fall durch eine vorherige Konversation der nutzenden Person mit dem Chatbot entsteht. Dies war in der vorliegenden Arbeit aufgrund des momentanen Entwicklungsstands des Chatbots noch nicht möglich, könnte aber in Zukunft mit durchgeführt werden. Zudem könnte in zukünftiger Forschung der Prototyp mit weiteren Systemen verglichen und evaluiert werden, da der TFIDF-Retriever und der Dense-Retriever beide eher auf Question-Answering ausgelegt sind. Für die durchgeführte Evaluation war dieser Vergleich jedoch ausreichend, da es vor allem darum ging, thematisch passende Ausschnitte zu empfehlen. Damit leitet sich auch der letzte Punkt ein, nämlich die Untersuchung der Nützlichkeit und des Informationsgehalts des empfohlenen Textausschnitts. Diese Informationen wurden nicht mit in die Evaluation einbezogen, sollten aber in weiterer Forschung ebenso evaluiert werden, da auch diese Faktoren eine maßgebliche Rolle für die Qualität der Empfehlungen spielen.

9 Schlussbetrachtung

Recommender Systeme sind in der Lage, Empfehlungen an die nutzende Person auszuspielen, welche dieser im besten Fall weiterhelfen können oder interessant für sie sind. Im Gesundheitswesen können Chatbots Recommender Systeme nutzen, um den anwendenden Personen Informationen und Tipps zu ihrer Gesundheit zu geben.

Die vorliegende Arbeit hat sich mit dieser Thematik befasst, indem ein kontextsensitives Recommender System für einen Chatbot aus dem Gesundheitswesen konzipiert und dieses Konzept zu Teilen in Form eines Prototyps implementiert und evaluiert wurde. Auf der Basis von Befragungen von Experten und Expertinnen, Unterlagen des Partner-Projekts und einer Literaturanalyse wurden Personas, Use Cases und Anforderungen definiert, durch welche ein konzeptionelles Design eines Recommender Systems aufgestellt werden konnte. Das konzeptionelle Design konnte für die Architektur des Prototyps genutzt werden. Dieser Prototyp konnte anhand der zuvor definierten Anforderungen evaluiert werden. Er weist eine Kontextsensitivität in Bezug auf die implementierten Kontextdaten auf und bezieht diese mit in die Wahl der Empfehlung ein. Zudem wägt er ab, wann eine Empfehlung sinnvoll ist und spielt diese erst dann aus, wenn genug aussagekräftige Kontextdaten vorhanden sind. Um einen Vergleichswert zu schaffen, wurde der Prototyp gegen andere schon bestehende Systeme evaluiert und hat dabei bei der ersten Empfehlung eine signifikant höhere thematische Übereinstimmung mit den Chatdaten aufgewiesen, als die beiden anderen Systeme.

Es gibt noch einige Aspekte an Recommender Systemen für Chatbots im Gesundheitswesen, welche noch nicht ausreichend erforscht sind, wie beispielsweise die Nützlichkeit und der Informationsgehalt der Empfehlungen für die anwendende Person. Es sind deshalb weitere Studien und Forschung in diesem Bereich notwendig, um Recommender Systeme bei Chatbots im Gesundheitswesen zu etablieren. Abschließend lässt sich feststellen, dass ein Recommender System bei einem Chatbot im Gesundheitswesen in zukünftigen Anwendungen und Projekten großes Potenzial aufweisen kann, um Mitarbeitende des Gesundheits- und Sozialwesen zu unterstützen. Dies spiegelt sich in den Ergebnissen der Evaluation in Kapitel 7 wider, welche aufzeigt, dass Recommender Systeme der anwendenden Person einen thematischen Mehrwert durch weiterführende Informationen liefern können.

Literaturverzeichnis

[1] E. M. Giusti, E. Pedroli, G. E. D'Aniello, C. Stramba Badiale, G. Pietrabissa, C. Manna, M. Stramba Badiale, G. Riva, G. Castelnuovo, und E. Molinari, „The Psychological Impact of the COVID-19 Outbreak on Health Professionals: A Cross-Sectional Study," *Frontiers in psychology*, Vol. 11, S. 1684, 2020.

[2] MarketsandMarkets, „Healthcare Chatbots Market by Component (Software, Service), Deployment Model (Cloud, On-Premise), Application (Symptom Check, Medical Assistance, Appointment Booking), End User (Patient, Healthcare Providers, Insurance Companies) - Global Forecast to 2023," 2018.

[3] T. J. Judson, A. Y. Odisho, J. J. Young, O. Bigazzi, D. Steuer, R. Gonzales, und A. B. Neinstein, „Implementation of a digital chatbot to screen health system employees during the COVID-19 pandemic," *Journal of the American Medical Informatics Association : JAMIA*, Vol. 27, Nr. 9, S. 1450–1455, Juni 2020.

[4] M. H. Mohamed, M. H. Khafagy, und M. H. Ibrahim, „Recommender Systems Challenges and Solutions Survey," in *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*. IEEE, Februar 2019, S. 149–155.

[5] D. Jannach, A. Manzoor, W. Cai, und L. Chen, „A Survey on Conversational Recommender Systems," *ACM Computing Surveys*, Vol. 54, Nr. 5, Mai 2021.

[6] Pulsnetz. (25.07.2022) Voneinander lernen und gemeinsam vorankommen. [Online]. Zuletzt abgerufen am 25.07.2022. https://www.pulsnetz.de/

[7] B. Kuechler und V. Vaishnavi, „On theory development in design science research: anatomy of a research project," *European Journal of Information Systems*, Vol. 17, Nr. 5, S. 489–504, September 2008.

[8] A. R. Hevner, S. T. March, J. Park, und S. Ram, „Design Science in Information Systems Research," *Management Information Systems Quarterly*, Vol. 28, Nr. 1, S. 75–105, März 2004.

[9] A. Dresch, D. P. Lacerda, und Antunes, José A. V. Jr., *Design Science Research: A Method for Science and Technology Advancement*. Springer, 2015.

[10] A. Hevner und S. Chatterjee, *Design Research in Information Systems*. Boston, MA: Springer US, 2010, Vol. 22.

[11] W. Kuechler, V. K. Vaishnavi, und S. Petter, „The Aggregate General Design Cycle as a Perspective on the Evolution of Computing Communities of Interest in_ Computing Letters Volume 1 Issue 3 (2005)," *Computing Letters*, Vol. 1, Nr. 3, S. 123–128, Juli 2005.

[12] A. Joshi, S. Kale, S. Chandel, und D. K. Pal, „Likert Scale: Explored and Explained," *Current Journal of Applied Science and Technology*, Vol. 7, Nr. 4, S. 396–403, Februar 2015.

[13] K. R. Chowdhary, *Fundamentals of Artificial Intelligence*, 1. Aufl., Serie Springer eBook Collection. New Delhi: Springer India and Imprint Springer, 2020.

[14] A. Kulkarni, A. Shivananda, und A. Kulkarni, *Natural Language Processing Projects: Build Next-Generation NLP Applications Using AI Techniques*, 1. Aufl., Serie Springer eBook Collection. Berkeley, CA: Apress and Imprint Apress, 2022.

[15] M. Awad und R. Khanna, *Efficient Learning MachinesDesigners: Theories, concepts, and applications for engineers and system designers*, Serie The expert's voice in machine learning. New York: Apress Berkeley, 2015.

[16] D. Jurafsky und J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3. Aufl. Upper Saddle River, NJ: Prentice Hall PTR, Dezember 2020.

[17] J. Mackmood, D. Bammert Marty, und S. D'Onofrio, „Chatbot & Cognitive Services – Ein Schritt Richtung Automatisierung im User Help Desk der Schweizerischen Post," in *Cognitive Computing*, Serie Edition Informatik Spektrum, E. Portmann und S. D'Onofrio, Hrsgg. Springer Vieweg, 2020, S. 147–168.

[18] A. Kulkarni und A. Shivananda, *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning Using Python*, 2. Aufl., Serie Springer eBook Collection. Erscheinungsort nicht ermittelbar and Boston, MA: Apress and Safari, 2021.

[19] D. Yemelyanov. (August 2020) Word Embeddings by example.

[20] N. Sabharwal und A. Agrawal, *Hands-on Question Answering Systems with BERT: Applications in Neural Networks and Natural Language Processing*. Berkeley, CA: Apress, 2021.

[21] R. Horev. (November 2018) BERT Explained: State of the art language model for NLP. [Online]. Zuletzt abgerufen am 28.07.2022

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, und I. Polosukhin, „Attention Is All You Need," in *31st Conference on Neural Information Processing Systems*, Juni 2017.

[23] I. Vayansky und S. A. P. Kumar, „A review of topic modeling methods," *Information Systems*, Vol. 94, Juni 2020.

[24] M. Grootendorst, „BERTopic: Neural topic modeling with a class-based TF-IDF procedure," März 2022.

[25] F. Jurie, M. Bucher, und S. Herbin, „Generating Visual Representations for Zero-Shot Classification," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, März 2017, S. 2666–2673.

[26] E. Adamopoulou und L. Moussiades, „An Overview of Chatbot Technology," in *Artificial Intelligence Applications and Innovations*, Serie Springer eBook Collection, I. Maglogiannis, L. Iliadis, und E. Pimenidis, Hrsgg. Springer International Publishing and Imprint Springer, 2020, Vol. 584, S. 373–383.

[27] T. Stucki, S. D'Onofrio, und E. Portmann, *Chatbots gestalten mit Praxisbeispielen der Schweizerischen Post: HMD Best Paper Award 2018*, Serie Springer eBooks Computer Science and Engineering. Wiesbaden: Springer Vieweg, 2020.

[28] K. Ramesh, S. Ravishankaran, A. Joshi, und K. Chandrasekaran, „A Survey of Design Techniques for Conversational Agents," in *Information, Communication and Computing Technology*, S. Kaushik, D. Gupta, L. Kharb, und D. Chahal, Hrsgg. Springer Singapore, 2017, Vol. 750, S. 336–350.

[29] S. Hussain, O. Ameri Sianaki, und N. Ababneh, „A Survey on Conversational Agents/Chatbots Classification and Design Techniques," in *Web, Artificial Intelligence and Network Applications*, L. Barolli, M. Takizawa, F. Xhafa, und T. Enokido, Hrsgg. Springer International Publishing, 2019, Vol. 927, S. 946–956.

[30] H. T. Hien, P.-N. Cuong, L. N. H. Nam, H. T. K. Le Nhung, und L. D. Thang, „Intelligent Assistants in Higher-Education Environments," in *Proceedings of the Ninth International Symposium on Information and Communication Technology*, Serie ACM Other conferences, Unknown, Hrsg. ACM, 2018, S. 69–76.

[31] Y. Wu, W. Wu, C. Xing, M. Zhou, und Z. Li, „Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots," 2016.

[32] K. Nimavat und T. Champaneria, „Chatbots: An overview. Types, Architecture, Tools and Future Possibilities," in *International Journal for Scientific Research & Development*, Oktober 2017.

[33] A. Polonioli, „The ethics of scientific recommender systems," *Scientometrics*, Vol. 126, Nr. 2, S. 1841–1848, Februar 2021.

[34] D. Kotkov, J. Veijalainen, und S. Wang, „Challenges of Serendipity in Recommender Systems," in *Proceedings of the 12th International Conference on Web Information Systems and Technologies*. SCITEPRESS - Science and and Technology Publications, 2016, S. 251–256.

[35] C. C. Aggarwal, *Recommender systems: The textbook*, 1. Aufl. Cham: Springer, 2016.

[36] F. Ricci, L. Rokach, und B. Shapira, „Recommender Systems: Introduction and Challenges," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, und B. Shapira, Hrsgg. Springer, Januar 2011, S. 1–34.

[37] G. D. Abowd und A. K. Dey, „Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*, Serie Lecture Notes in Computer Science Ser, H.-W. Gellersen, Hrsg. Springer Berlin / Heidelberg, 1999, Vol. v.1707, S. 304–307.

[38] C. McIntosh, Hrsg., *Cambridge advanced learner's dictionary*. Cambridge, United Kingdom and New York, NY and Port Melbourne, VIC and Delhi and Singapore: Cambridge University Press, 2013.

[39] T. Mikolov, K. Chen, G. Corrado, und J. Dean, „Efficient Estimation of Word Representations in Vector Space," Januar 2013.

[40] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, und L. Zettlemoyer, „Deep contextualized word representations," März 2018.

[41] R. Porzel, *Contextual Computing: Models and Applications*, Serie Cognitive technologies. Berlin and Heidelberg: Springer, 2011.

[42] R. Burke, „Hybrid Web Recommender Systems," in *The adaptive Web*, Serie State-of-the-art survey, P. Brusilovsky, A. Kobsa, und W. Nejdl, Hrsgg. Springer, 2007, S. 377–408.

[43] P. Lops, M. de Gemmis, und G. Semeraro, „Content-based Recommender Systems: State of the Art and Trends," in *Recommender systems handbook*, F. Ricci, L. Rokach, B. Shapira, und P. B. Kantor, Hrsgg. Springer, 2011, S. 73–105.

[44] K. Christakopoulou, F. Radlinski, und K. Hofmann, „Towards Conversational Recommender Systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Serie ACM Digital Library, B. Krishnapuram, Hrsg. ACM, 2016, S. 815–824.

[45] J. S. Breese, D. Heckerman, und C. Kadie, „Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Redmond, WA, USA, 1998.

[46] X. Su und T. M. Khoshgoftaar, „A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, Vol. 2009, Nr. 421425, S. 1687–7470, Oktober 2009.

[47] D. Kluver, M. D. Ekstrand, und J. A. Konstan, „Rating-Based Collaborative Filtering: Algorithms and Evaluation," in *Social information access*, Serie Lecture Notes in Computer Science, P. Brusilovsky und D. He, Hrsgg. Springer, 2018, Vol. 10100, S. 344–390.

[48] B. Smyth, „Case-Based Recommendation," in *The adaptive Web*, Serie State-of-the-art survey, P. Brusilovsky, A. Kobsa, und W. Nejdl, Hrsgg. Springer, 2007, S. 342–376.

[49] N. Silva, D. Carvalho, A. C. Pereira, F. Mourão, und L. Rocha, „The Pure Cold-Start Problem: A deep study about how to conquer first-time users in recommendations domains," *Information Systems*, Vol. 80, S. 1–12, Februar 2019.

[50] A. Felfernig und R. Burke, „Constraint-based Recommender Systems: Technologies and Research Issues," in *ACM International Conference Proceeding Series*, Januar 2008.

[51] K. Panetta. (August 2017) Top Trends In The Gartner Hype Cycle For Emerging Technologies 2017. [Online]. Zuletzt abgerufen am 31.03.2022. https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017

[52] ——. (September 2016) Gartner 2016 Hype Cycles Reveal 4 Megatrends. [Online]. Zuletzt abgerufen am 31.03.2022. https://www.gartner.com/smarterwithgartner/gartner-2016-hype-cycles-reveal-4-megatrends

[53] J. L. Z. Montenegro, C. A. Da Costa, und R. Da Rosa Righi, „Survey of conversational agents in health," *Expert Systems with Applications*, Vol. 129, S. 56–67, September 2019.

[54] J. Parviainen und J. Rantala, „Chatbot breakthrough in the 2020s? An ethical reflection on the trend of automated consultations in health care," *Medicine, health care, and philosophy*, Vol. 25, Nr. 1, S. 61–71, 2022.

[55] P. Kandpal, K. Jasnani, R. Raut, und S. Bhorge, „Contextual Chatbot for Healthcare Purposes (using Deep Learning)," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, S. 625–634.

[56] A. K. Sahoo, C. Pradhan, R. K. Barik, und H. Dubey, „DeepReco: Deep Learning Based Health Recommender System Using Collaborative Filtering," *Computation*, Vol. 7, Nr. 2, S. 25, Mai 2019.

[57] Y. Sun und Y. Zhang, „Conversational Recommender System," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Serie ACM Conferences, K. Collins-Thompson, Hrsg. ACM, Juni 2018, S. 235–244.

[58] P. Cordero, M. Enciso, D. López, und A. Mora, „A conversational recommender system for diagnosis using fuzzy rules," *Expert Systems with Applications*, Vol. 154, Nr. 113449, 2020.

[59] B. Lika, K. Kolomvatsos, und S. Hadjiefthymiades, „Facing the cold start problem in recommender systems," *Expert Systems with Applications*, Vol. 41, Nr. 4, S. 2065–2073, 2014.

[60] L. Sharma und A. Gera, „A Survey of Recommendation System: Research Challenges," *International Journal of Engineering Trends and Technology (IJETT)*, Vol. 4, Nr. 5, S. 1989–1992, Mai 2013.

[61] L. A. G. Camacho und S. N. Alves-Souza, „Social network data to alleviate cold-start in recommender system: A systematic review," *Information Processing & Management*, Vol. 54, Nr. 4, S. 529–544, Juli 2018.

[62] C. Gao, W. Lei, X. He, M. de Rijke, und T.-S. Chua, „Advances and challenges in conversational recommender systems: A survey," *AI Open*, Vol. 2, S. 100–126, 2021.

[63] A. Gunawardana, G. Shani, und S. Yogev, „Evaluating Recommender Systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, und B. Shapira, Hrsgg. Springer, Januar 2011, S. 257–297.

[64] P. Castells und A. Moffat, „Offline recommender system evaluation: Challenges and new directions," *AI magazine*, Vol. 43, Nr. 2, S. 225–238, Juni 2022.

[65] L. Valentine, S. D'Alfonso, und R. Lederman, „Recommender systems for mental health apps: advantages and ethical challenges," *AI & SOCIETY*, S. 1–12, Januar 2022.

[66] A. C. Valdez und M. Ziefle, „The Users' Perspective on the Privacy-Utility Trade-offs in Health Recommender Systems," *International Journal of Human-Computer Studies*, Vol. 121, Nr. 1, S. 108–121, Januar 2019.

[67] M. Nickl, „Marken – Herausforderung für die Technische Dokumentation," in *Marke und Gesellschaft*, Serie VS research, N. Janich, Hrsg. VS Verlag für Sozialwissenschaften, 2009, S. 163–178.

[68] A. Beck, H. Eichstädt, W. Schweibenz, B. Gaiser, P. Savigny, und U. Schubert, „Personas in der Praxis," in *Tagungsband UP05*, M. Hassenzahl und M. Peissner, Hrsgg., August 2005, S. 94–100.

[69] H. Gatterer, V. Muntschick, P. Hofstätter, J. Seitz, L. Papasabbas, C. Schuldt, C. Kelber, M. Morrison, und C. Kristandl, „Lebensstile: Eine neue Sicht auf Kunden und ihre Bedürfnisse," Frankfurt, 2017.

[70] o. A., „DIVSI Internet-Milieus 2016: Die digitalisierte Gesellschaft in Bewegung," Hamburg, 2016.

[71] ——. (2017) Anaconda Documentation. [Online]. Zuletzt abgerufen am 31.05.2022. https://conda.io/en/latest/

[72] F. Oh. (September 2012) What Is CUDA? [Online]. Zuletzt abgerufen am 08.07.2022. https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/

[73] o. A. (o. A.) Docker overview. [Online]. Zuletzt abgerufen am 31.05.2022. https://docs.docker.com/get-started/overview/

[74] ——. (2022) Elastic Docs. [Online]. Zuletzt abgerufen am 16.05.2022. https://www.elastic.co/guide/index.html

[75] ——. (2010) Flask: web development, one drop at a time. [Online]. Zuletzt abgerufen am 27.05.2022. https://flask.palletsprojects.com/en/2.1.x/foreword/

[76] S. Chacon und J. Long. (31.05.2022) Git: About. [Online]. Zuletzt abgerufen am 31.05.2022. https://git-scm.com/about

[77] C. V. Ramamoorthy und H. F. Li, „Pipeline Architecture," *Computing Surveys*, Vol. 9, Nr. 1, S. 61–102, März 1977.

[78] o. A. (0. A.) Haystack: Overview. [Online]. Zuletzt abgerufen am 27.05.2022. https://haystack.deepset.ai/overview/intro

[79] S. Tomar. (Mai 2021) German Zeroshot. [Online]. Zuletzt abgerufen am 28.06.2022. https://huggingface.co/Sahajtomar/German_Zeroshot

[80] T. Aarsen, J. Nothman, S. Bird, A. Dimitradis, D. Sepler, D. Milajevs, F. Bond, I. Kurenkov, und L. Tan. (März 2022) NLTK: Documentation. [Online]. Zuletzt abgerufen am 08.07.2022. https://www.nltk.org/

[81] o. A. (Juli 2022) Introduction to Rasa Open Source. [Online]. Zuletzt abgerufen am 16.05.2022. https://rasa.com/docs/rasa/

[82] D. Braun, A. Hernandez-Mendez, F. Matthes, und M. Langen, „Evaluating Natural Language Understanding Services for Conversational Question Answering Systems," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, K. Jokinen, M. Stede, D. DeVault, und A. Louis, Hrsgg. Association for Computational Linguistics, August 2017, S. 174–185.

[83] T. Bunk, D. Varshneya, V. Vlasov, und A. Nichol, „DIET: Lightweight Language Understanding for Dialogue Systems," April 2020.

[84] A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, und V. Stoyanov, „XNLI: Evaluating Cross-lingual Sentence Representations," September 2018.

[85] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, und W.-t. Yih, „Dense Passage Retrieval for Open-Domain Question Answering," April 2020.

[86] T. Cleff, *Deskriptive Statistik und moderne Datenanalyse: Eine computergestützte Einführung mit Excel, SPSS und STATA*, Serie Springer eBook Collection. Wiesbaden: Gabler, 2008.

[87] H. Toutenburg und C. Heumann, *Deskriptive Statistik: Eine Einführung in Methoden und Anwendungen mit SPSS*, 5. Aufl., Serie Springer-Lehrbuch. Berlin and Heidelberg: Springer, 2006.

[88] M. Neuhäuser, „Wilcoxon–Mann–Whitney Test," in *International Encyclopedia of Statistical Science*, M. Lovric, Hrsg. Springer, 2011, S. 1656–1658.

[89] M. Hollander, D. A. Wolfe, und E. Chicken, *Nonparametric statistical methods*, 3. Aufl., Serie Wiley series in probability and statistics. Hoboken, New Jersey: John Wiley & Sons Inc, 2014.

Anhang

A.1 Personas

Die Unterteilung der Personengruppen werden anhand von drei Personas abstrahiert. Diese sind im Folgenden mit ihren einzelnen Merkmalen aufgeführt.

A.1.1 Persona von der Pflegekraft Annabell

Annabell

Stellenbezeichnung:
Pflegekraft

Alter: 26 Jahre

Höchster Schulabschluss: Realschulabschluss

Branche: Gesundheitswesen

Unternehmensgröße: 50-200 Mitarbeiter

DIVSI-Milieu: unbekümmerte Hedonisten

Lebensstil: Digitale Creative/Forever Youngster

Vorgesetzter: Pflegedienstleitung

Soziale Netzwerke:
- Instagram
- Twitter
- Pinterest
- Snapchat

Bevorzugte Kommunikationsmittel:
- Smartphone

Tools, die für die Arbeit erforderlich sind:
- Anrufsysteme
- Notfall-Pager

Zuständigkeiten:
- Betreuung, Beobachtung und Pflege von Patienten

Ziele:
- Menschen helfen

Maßstäbe für die Leistung:
- Zuverlässigkeit
- Durchhaltevermögen
- Einfühlsamkeit
- Fachliche medizinische Kompetenz

Informationsgewinnung durch:
- Online-Seminare
- Teilnahme an Mitarbeiterkonferenzen
- Google-Suche

Größte Herausforderungen:
- Berufliche Weiterentwicklung
- Problemlösung und Entscheidungsfindung
- Zeitmanagement

Privatleben:
- Kleine Wohnung mit ihrem Partner
- Unternimmt viel mit Freunden
- Geld ist zweitrangig

Werte:
- Menschlichkeit
- Unabhängigkeit
- Optimismus

Technikaffinität:
- Nutzt im privaten ihr Smartphone sehr ausgiebig, vor allem für Social-Media-Plattformen
- Hat einen privaten Laptop den sie vor allem zur Unterhaltung nutzt
- Ist im Job offen für mehr Digitalisierung

Gesundheit:
- das Thema Gesundheit ist ihr sehr wichtig
- ernährt sich gesund und treibt Sport
- überarbeitet sich meistens wegen Personalmangel

A.1.2 Persona von der Pflegedienstleitung Chiara

Chiara

Stellenbezeichnung:
Pflegedienstleitung

Alter: 45 Jahre

Höchster Schulabschluss: abgeschlossene Ausbildung und berufliche Weiterbildung

Branche: Gesundheitswesen

Unternehmensgröße: 50-200 Mitarbeiter

DIVSI-Milieu: Verantwortungsbedachte Etablierte

Lebensstil:
Vorwärtsmacher/Multiperformer/Golden Mentor

Vorgesetzter: Unternehmensleitung

Soziale Netzwerke:
- Facebook
- LinkedIn

Bevorzugte Kommunikationsmittel:
- E-Mail
- Persönlich

Tools, die für die Arbeit erforderlich sind:
- E-Mail-Programm
- Mitarbeiterplanungssoftware

Zuständigkeiten:
- Personalmanagement
- Organisationsmanagement
- Bedarfsplanung

Ziele:
- Hohe Mitarbeiterzufriedenheit
- Gute Organisationsstruktur
- Beruflicher Aufstieg

Maßstäbe für die Leistung:
- Personalmanagement
- Kommunikationsfähigkeit
- Durchsetzungsvermögen

Informationsgewinnung durch:
- Teilnahme an Konferenzen
- Teilnahme an Messen
- Teilnahme an Fortbildungen

Größte Herausforderungen:
- Ressourcen
- Mitarbeitermotivation

Privatleben:
- Hat ein Mann und drei Kinder
- Ist ein Familienmensch
- Ihr Job ist ihr sehr wichtig

Werte:
- Flexibilität
- Klarheit
- Zuversicht

Technikaffinität:
- Steht der Digitalisierung positiv gegenüber
- eher technikaffin
- Nutzt im Privaten einen PC und Laptop für Büroangelegenheiten
- Nutzt im beruflichen Umfeld ihren PC mit diverser Software

Gesundheit:
- Hat körperliche Beschwerden, wie Rückenschmerzen
- Macht keinen Sport aber achtet auf ihre Ernährung
- Ist oft gestresst von der Arbeit und schläft zu wenig

A.1.3 Persona von der Sozialarbeiter Markus

Markus

Stellenbezeichnung: Sozialarbeiter – Kinder- und Jugendarbeit

Alter: 55 Jahre

Höchster Schulabschluss: Bachelor-Abschluss

Branche: Gesundheitswesen

Unternehmensgröße: 11-50 Mitarbeiter

DIVSI-Milieu: vorsichtige Skeptiker

Lebensstil: Progressive Parent/Neo-Biedermeier/Nervösbürger

Vorgesetzter: Abteilungsleiter

Soziale Netzwerke:
- keine

Bevorzugte Kommunikationsmittel:
- Persönlich
- Telefon
- Post

Tools, die für die Arbeit erforderlich sind:
- Fax
- E-Mail-Programm

Zuständigkeiten:
- Beratung und Kommunikation mit Kindern, Jugendlichen und deren Familien

Ziele:
- Verbesserung der Lebensbedingungen von Kindern und Jugendlichen

Maßstäbe für die Leistung:
- Frustrationstoleranz
- Kommunikationsfähigkeit
- Fachkompetenz

Informationsgewinnung durch:
- Teilnahme an Konferenzen
- Teilnahme an Messen

Größte Herausforderungen:
- Änderungsmanagement
- Zeitmanagement

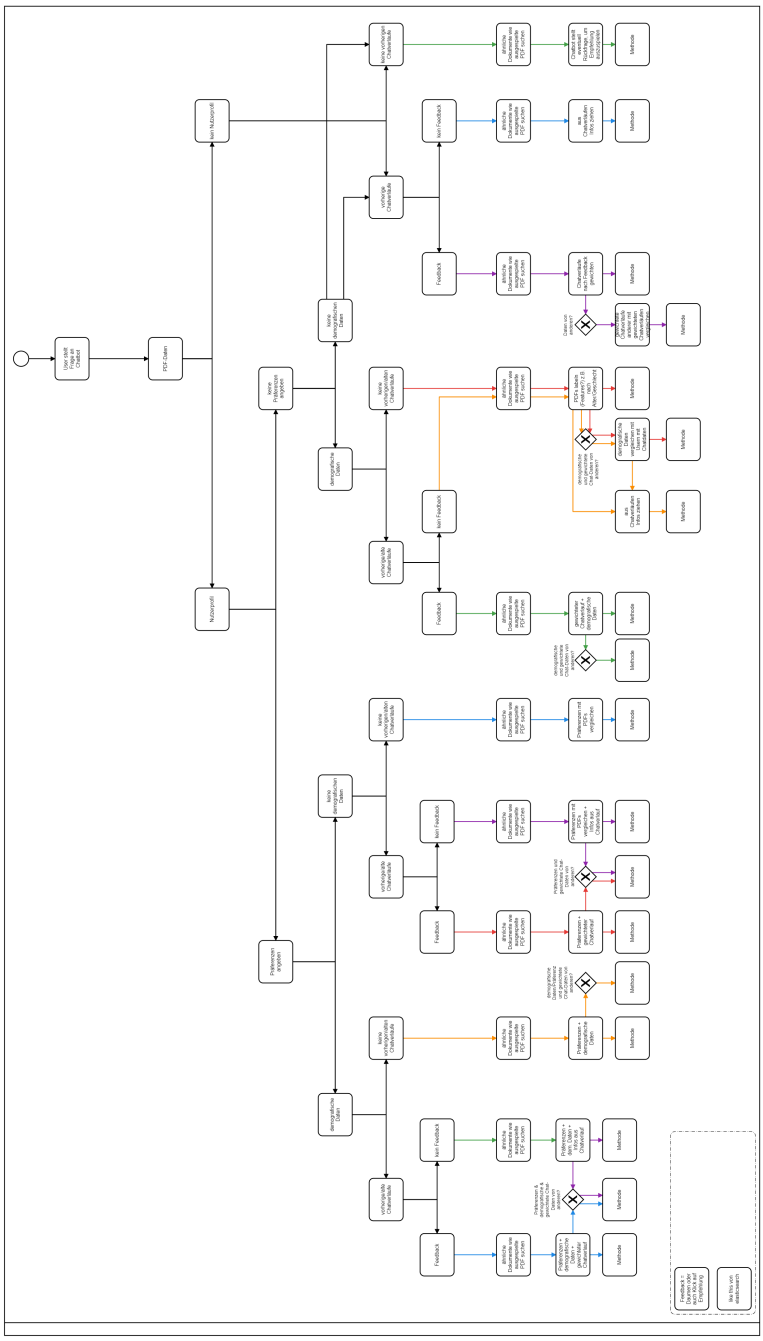
Privatleben:
- Haus in ländlicher Region
- Ist ein Familienmensch
- Hat eine Frau und zwei Kinder
- Ist gerne in der Natur

Werte:
- Klarheit
- Fairness
- Toleranz

Technikaffinität:
- Steht der Digitalisierung in Teilen kritisch gegenüber
- wenig technikaffin
- nutzt im Privaten kein Smartphone und selten einen Stand-PC
- Nutzt im beruflichen Umfeld seinen Stand-PC sporadisch

Gesundheit:
- Das Thema Gesundheit spielt bei ihm eher unterbewusst eine Rolle
- Sehr sportlich aber beschäftigt sich wenig mit Themen wie Ernährung
- Hat eine gute Work-Life-Balance

A.2 Vollständiges Use Case Diagramm



A.3 Gekürzter Ausschnitt eines gelabelten Chatverlaufs


```
{
    "chat": [
        {
            "event": "user",
            "text": "Ich arbeite in der Pflege und habe seitdem Rückenschmerzen. Was kann ich dagegen tun?"
        },
        {
            "event": "bot",
            "text": "Rückenschmerzen lassen sich auf eine hohe psychische und physische Belastung zurückführen."
        }
    ],
    "automatic_labels": [
        {
            "label": "rueckenschmerzen",
            "score": 0.1213025376200676
        },
        {
            "label": "ruecken",
            "score": 0.08589255809783936
        }
    ],
    "automatic_labels_formula": [
        {
            "label": "rueckenschmerzen",
            "score": 0.15361499806210088
        },
        {
            "label": "ruecken",
            "score": 0.10877254016791219
        }
    ],
    "profession": [
        {
            "label": "krankenpflege",
            "score": 0.17861826717853546
        },
        {
            "label": "pflegekraft",
            "score": 0.17535577714443207
        }
    ],
}
```

A.4 Box-Plot des Scorings der technischen Evaluation mit Ausreissern