

Masterarbeit

**Entwicklung eines Chat-Bots zur  
interaktiven Problemeinschränkung**  
Am Beispiel von Facility-Management

Sebastian Klatte, B.Sc.





Heinz Nixdorf Institut  
Fakultät für Wirtschaftswissenschaften  
Universität Paderborn

## Masterarbeit

# **Entwicklung eines Chat-Bots zur interaktiven Problemeinschränkung**

## Am Beispiel von Facility-Management

vorgelegt von

Sebastian Klatte  
66·18·227  
Vlothoer Str. 8a  
32108 Bad Salzuflen  
[sebastian.klatte@posteo.de](mailto:sebastian.klatte@posteo.de)

Universität Paderborn

Fakultät Wirtschaftswissenschaften

Fachgebiet Wirtschaftsinformatik

Wirtschaftsinformatik, insb. Semantische Informationsverarbeitung

betreut von:

Frederik Simon Bäumer, M.Sc.

Jun.-Prof. Dr. Michaela Geierhos

Paderborn, April 2017



**Abstract – english**

This master thesis provides an overview of building natural-language and text-based chat bots by using frameworks.

The investigation is carried out using an example in Facility Management. In practice, the reports on an emergency or malfunction are received by telephone. This is to be supported interactively by a chat bot.

Furthermore, the domain-specific problem area is used to compare and evaluate the chat-bot frameworks in order to determine the suitability, including performance and reliability.

The elaborated and potentially most appropriate frameworks use machine learning and the recognition of proper names to extract the information relevant to the incident. Using a developed rule set that also uses decision trees, the extracted data were categorized to narrow the problem, the priority, and the related fields of activity.

**Tags:** <chat bot, dialogue system, machine learning, named entity recognition, decision tree>

**Abstract – deutsch**

Diese Masterarbeit gibt einen Überblick über die Erstellung von natürlichsprachlichen und textbasierten Chat-Bots mithilfe von Frameworks.

Die Untersuchung wird anhand eines Beispiels im Facility-Management durchgeführt. Im Praxisalltag werden dort die Meldungen zu einem Not- oder Störfall telefonisch entgegengenommen. Mithilfe eines Chat-Bots soll dies interaktiv unterstützt werden.

Der domänenspezifische Problembereich dient ebenfalls dazu, die Chat-Bot Frameworks zu vergleichen und zu bewerten um die Eignung, samt Leistungsfähigkeit und Zuverlässigkeit, festzustellen.

Die herausgearbeiteten und potenziell geeignetsten Frameworks verwenden maschinelles Lernen und die Erkennung von Eigennamen, um die für den Vorfall relevanten Informationen zu extrahieren. Mittels eines entwickelten Regelwerks, das auch Entscheidungsbäume nutzt, wurden die extrahierten Daten kategorisiert, um das Problem, die Priorität und die damit verbundenen Tätigkeitsfelder einzuschränken.

**Stichworte:** <Chat-Bot, Dialogsystem, maschinelles Lernen, Eigennamenerkennung, Entscheidungsbaum>



## Inhaltsverzeichnis

<b>Abbildungsverzeichnis .....</b>	<b>ix</b>
<b>Tabellenverzeichnis .....</b>	<b>xi</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Problemstellung .....	3
1.2 Zielsetzung.....	4
1.3 Vorgehensweise .....	4
1.4 Motivation.....	5
<b>2 Problembereich .....</b>	<b>7</b>
2.1 Chat-Bots .....	7
2.1.1 Allgemeines.....	7
2.1.2 Aufbau und Komponenten .....	12
2.2 Facility-Management.....	14
2.2.1 Strukturen.....	14
2.2.2 Aufgabenbereiche und Prioritäten.....	16
2.2.3 Aktivitäten im Vorfallmanagement.....	19
<b>3 Stand der Technik .....</b>	<b>21</b>
3.1 Algorithmen und Methoden .....	21
3.1.1 Textvorverarbeitung .....	22
3.1.2 Textanalyse.....	24
3.1.3 Textproduktion .....	26
3.1.4 Klassifikation .....	26
3.2 Frameworks .....	28
3.2.1 Facebook .....	28
3.2.2 Microsoft .....	29
3.2.3 Google .....	31
3.2.4 IBM.....	32
3.2.5 Amazon.....	33
3.2.6 Recast.ai .....	34
3.2.7 ChatScript .....	35
3.2.8 Pandorabots.....	36
3.2.9 RiveScript.....	37

3.2.10	Motion AI und andere .....	37
3.2.11	Botkit .....	38
3.2.12	NLP-Toolkits .....	38
<b>4</b>	<b>Lösungskonzept .....</b>	<b>39</b>
4.1	Anforderungen.....	39
4.2	Lösungssegmente .....	40
4.2.1	Auswahl und Bewertung der Frameworks .....	40
4.2.2	Erstellung eines Katalogs an Vorfällen .....	43
4.2.3	Deutungsmöglichkeiten der Aussagen .....	45
4.2.4	Beschreibung des Dialogflusses.....	46
4.2.5	Datenmodell eines Vorfalls und dessen Meldungen.....	51
4.2.6	Bestandteile der zu erfassenden Daten.....	53
4.2.7	Kategorisierung des Vorfalls.....	58
<b>5</b>	<b>Implementierung.....</b>	<b>63</b>
5.1	Architektur .....	63
5.1.1	Komponenten .....	64
5.1.2	Pakete .....	66
5.1.3	Programme.....	66
5.2	Absichten und Entitäten .....	67
5.2.1	Erfassung der Adresse .....	67
5.2.2	Erfassung des Geschehens und des lokalen Ortes.....	70
5.2.3	Erfassung von Zeit, Dringlichkeit und persönlichen Daten ....	71
5.3	Abläufe und Zustände .....	72
5.4	Datenmodell und Datenverarbeitung.....	75
5.5	Benutzeroberfläche .....	77
<b>6</b>	<b>Evaluation.....</b>	<b>79</b>
6.1	Natürlichsprachlichkeit.....	79
6.2	NLU-Dienste.....	80
6.3	Klassifikation .....	82
<b>7</b>	<b>Fazit und Ausblick .....</b>	<b>85</b>
	<b>Literaturverzeichnis .....</b>	<b>89</b>
	<b>Anhang .....</b>	<b>97</b>



## Abbildungsverzeichnis

Abbildung 2.1: Beispiel einer kontextfreien Grammatik in Backus-Naur-Form	8
Abbildung 2.2: Komponenten eines textbasierten Dialogsystems (vgl. [CFL13], S. 444)	12
Abbildung 2.3: Prozessmodell für FM (vgl. [GEFMA-100-1], S. 2)	15
Abbildung 2.4: Operatives Geschäft	15
Abbildung 3.1: Beispiel einer ChatScript Regel	35
Abbildung 3.2: Beispiel einer AIML Regel	36
Abbildung 3.3: Beispiel einer RiveScript Regel	37
Abbildung 4.1: Dialogfluss	47
Abbildung 4.2: Beispiel eines Dialogverlaufes	49
Abbildung 4.3: Datenmodell eines Vorfalles und Meldungen	52
Abbildung 4.4: Aussagenklassen eines Vorfalles	54
Abbildung 4.5: Lokale Ortsbezeichnungen und Entitäten	55
Abbildung 4.6: Informationsabhängigkeiten	59
Abbildung 4.7: Typklassifikation der Vorfälle und relevanten Berufe	61
Abbildung 4.8: Beispiel eines Entscheidungsbaums für Priorität der Sauberkeit	62
Abbildung 5.1: Komponentendiagramm	65
Abbildung 5.2: Paketdiagramm der Node.js Pakete	66
Abbildung 5.3: Klassendiagramm der Programme	67
Abbildung 5.4: Zuweisung der Entitäten in LUIS	70
Abbildung 5.5: Zuweisung der Entitäten in Api.ai	70
Abbildung 5.6: Zuweisung der Entitäten in Recast.ai	71
Abbildung 5.7: Aktivitätsdiagramm zur Erkennung der Adresse	72
Abbildung 5.8: Aktivitätsdiagramm zur Erkennung und Klassifikation des Vorfalles	74
Abbildung 5.9: Klassendiagramm der Konversationsdaten	76
Abbildung 5.10: Beispiel der Meldung eines Vorfalles im Emulator (1)	77
Abbildung 5.11: Beispiel der Meldung eines Vorfalles im Emulator (2)	78



## Tabellenverzeichnis

Tabelle 2.1: Not- und Störfälle im FM aus dem StLV (vgl. [GEFMA-520], S. 23f)	17
Tabelle 2.2: Störungs- und Prioritätenkategorien der Not- und Störfälle im FM	18
Tabelle 4.1: Vergleich der Chat-Bot Frameworks	41
Tabelle 4.2: Verteilung der Vorfälle im Katalog	44
Tabelle 4.3: Deutungsmöglichkeiten der Informationstypen	45
Tabelle 4.4: Begrüßung, Fragen und Rückfragen des Chat-Bots	50
Tabelle 4.5: Absichten und deren Entitäten	57
Tabelle 4.6: Vorfalltypen und relevante Berufe	60
Tabelle 6.1: Vergleich der erkannten Entitäten	81
Tabelle 6.2: Vergleich der Vorfalltyp-Klassifikation	83
Tabelle 6.3: Vergleich der Priorität-Klassifikation	83
Tabelle A.1: Vorfallkatalog - Trainingsset	97
Tabelle A.2: Vorfallkatalog - Testset	107
Tabelle A.3: Irrelevante Aussagen und Aussagen zum Zeitpunkt	112



## 1 Einleitung

Dialogsysteme<sup>1</sup> sollen dem Menschen den Alltag erleichtern; unterwegs mit dem Smartphone oder Automobil, auf der Arbeit oder im Smart Home. Statt eine Benutzeroberfläche zu bedienen oder physische Schalter zu betätigen soll ein Dialogsystem die Aufgaben, Fragen und Wünsche des Menschen verstehen und umsetzen.

Das bietet den Anreiz informationstechnische Systeme verwenden zu können, ohne die Oberfläche zu benutzen beziehungsweise nur Teile davon. Die menschliche Kommunikationsmöglichkeit wird dementsprechend zur natürlichen Eingabeoption. Folglich lässt sich erkennen, dass zukünftig die Bedienung einer Oberfläche zumindest wahlfrei realisiert wird. Ein Praxisbeispiel ist Amazon Echo.

Die ersten Chat-Bots<sup>1</sup> wurden vor dem Personal Computer (PC) und der Gründung von Apple an Großrechnern entwickelt. Darüber hinaus wurde mit dem Turing-Test ein Verfahren zur Überprüfung der Denkfähigkeit einer Maschine im Vergleich zu Menschen definiert. Mit der Einführung von PCs gewannen sprachbasierte Systeme an Bekanntheit, anfänglich ausschließlich zum Diktieren von Aufsätzen. Mit zunehmender Verbreitung des Internets entstanden Community-Foren und Chat-Systeme, die virtuelle Butler, ausgestattet mit simplen Funktionen, basierend auf der Erkennung von Stichworten, einsetzen.

Anschließend etablierten sich kommerzielle Chat-Bots für die Kundenberatung von Webshops. Diese werden auch heute noch, zum Teil in Kombination mit menschlichen Akteuren, eingesetzt. Telefonhotlines wurden mit Sprachdialogsystemen ausgestattet, bei denen der Kunde sein Anliegen durch das Antworten auf vordefinierte Stichwörter oder die Eingabe von Ziffern über die Telefontastatur spezifiziert, bevor er mit einem Sachbearbeiter verbunden wird. Vollständige Automatisierungen wurden etwa beim Telebanking durchgeführt.

Für mediale Beachtung sorgte zuletzt ein von Microsoft in Twitter integrierter Chat-Bot, der sich via Benutzernachrichten Wissen aneignete. Jedoch wurde der Chat-Bot mit menschenverachtender Propaganda trainiert, sodass Microsoft den zunehmend rassistischeren und sexistischeren Chat-Bot vom Netz nehmen musste.

Mittlerweile sind Chat-Bots und das Thema der natürlichsprachlichen Datenverarbeitung für die Industrie so wichtig, dass neben Microsoft alle großen Softwarehersteller wie

---

<sup>1</sup> Dialogsystem meint die natürlichsprachliche Kommunikation über eine auditive, textbasierte oder visuelle Schnittstelle beziehungsweise Kombinationen daraus. Ein Chat-Bot ist beispielsweise ein textbasiertes Dialogsystem. Abzugrenzen ist dieser Begriff von Systemen, die Dialoge auf einer grafischen Benutzeroberfläche bereitstellen und Eingaben in vordefinierten Dialogfeldern erwarten. Kapitel 2.1 enthält weitere Erläuterungen.

Amazon, Apple, Facebook, Google, IBM und SAP entsprechende Produkte (Siri, HANA und andere, die im Verlauf dieser Arbeit vorgestellt werden) im Portfolio bereitstellen. Darüber hinaus hat das Thema seit der NSA-Affäre auch eine politische Dimension bekommen. Moralische, ethische und soziale Fragen, wie auch der Umgang mit dem Datenschutz sind bislang weitgehend ungeklärt.<sup>2</sup> Daher sollte bei der Entwicklung und Verwendung von natürlichsprachlichen Systemen bedacht werden, dass diese prinzipiell auch für die personalisierte Auswertung und zu militärischen Zwecken verwendet werden könnten.

Im Facility-Management, werden Vorfälle, die in der Infrastruktur auftreten, bislang dem Telefonsupport gemeldet. Die Meldung erfolgt durch Nutzer<sup>3</sup> dieser Einrichtungen, beispielsweise in einem Kaufhaus durch das Personal. Die Annahme der Meldungen wird durch Supportmitarbeiter, welche sich durch langjährige Berufserfahrung im Facility-Management beziehungsweise der Baubranche auszeichnen, abgewickelt. Erkrankte Mitarbeiter oder Tage mit hoher Anzahl an Meldungen stellen eine Gefahr für die Abwicklung des vollständigen Meldungsempfangs dar. Auch der Verlust von Mitarbeitern birgt ein hohes Risiko, denn die Mitarbeiter sind für die Einschätzung der Vorfälle des Meldenden unverzichtbar. Aufgrund des benötigten Know-hows, das im Regelfall nur durch Schulungen und Praxiserfahrung vermittelt wird, sind Personalfluktuationen zu vermeiden. Zudem sind Personalwechsel aufwendig, da namentlich zu benennen inklusive Nachweis über persönliche und fachliche Qualifikationen, welche den Anforderungen im Vorfallmanagement genügen müssen (vgl. [GEFMA-520], S. 22f).

Durch ein zu entwickelndes Dialogsystem soll geprüft werden, ob dies eine praktikable Ergänzung zum Telefonsupport ist, der je nach vertraglicher Vereinbarung eine lückenlose Erreichbarkeit und sichere Kommunikationswege an 365 Tagen im Jahr rund um die Uhr gewährleisten muss. Zum einen kann die Gefährdung des Betriebes durch ausfallende Mitarbeiter eingeschränkt werden, zum anderen ist voraussichtlich weniger Personal bereitzustellen. Weiterhin lässt sich der Arbeitstag entlasten und die Schulung der Mitarbeiter durch die Problemeinschränkung und Hinweise seitens des Dialogsystems vereinfachen. Insgesamt ermöglicht dies die systematische Verbesserung der Qualität.

---

<sup>2</sup> Als Beispiel sei an dieser Stelle erwähnt, dass Julio Ceja eine Klage gegen Apple eingereicht hat, in welcher er das Softwareunternehmen auffordert, die Benutzung des Smartphones während der motorisierten Teilnahme am Straßenverkehr zu unterbinden. (Fall BC647057). Folglich steht die Frage im Raum ob ein Chat-Bot, dem beispielsweise ein Mordfall gestanden wird, dazu verpflichtet ist, die Information weiterzuleiten.

<sup>3</sup> Im Folgenden wird aus Gründen der besseren Lesbarkeit ausschließlich die männliche Form benutzt. Es können dabei aber sowohl männliche als auch weibliche Personen gemeint sein.

## 1.1 Problemstellung

Die Themengebiete der Dialogsysteme als auch des Facility-Managements sind jeweils interdisziplinär. Daher sind erforderliche Grundlagen und Lösungen für die verschiedenen Fachbereiche nötig.

Die Erstellung eines Chat-Bots erfordert die Entwicklung einer künstlichen Intelligenz, welche nach Piepenburg folgende Probleme umfasst:

- das Problem der maschinellen Sprach- und/oder Textrezeption;
- das Problem von Wissensbasen in Bezug auf Sprachregelwerke, sowie auf semantische Netze;
- das Problem der Kontexterfassung, speziell Probleme der Ellipsen-Bildung; und
- das Problem der Lernfähigkeit eines Rechnersystems. (vgl. [NuRö88], S. 136)

Die Problembehandlung des Vorfalls durch den Telefonsupport im Facility-Management beruht auf Erfahrungswerten. Die Vorgehensweise oder das Schema, nach dem ein Supportmitarbeiter den Vorfall bearbeitet und klassifiziert, ist in keiner Literatur dokumentiert.<sup>4</sup> Insbesondere die zu erfassenden Merkmale einer Vorfallmeldung und Entscheidungsregeln, nach welchen ein Störfall bewertet wird, fehlen. Einzig domainübergreifend definierte Prozessmodelle (siehe Kapitel 2.2.3) sowie die unternehmensspezifisch angepassten Aktivitäten geben Aufschluss über das Vorgehen bei einer telefonischen Erfassung.

Die Vorfälle im Facility-Management reichen von einem Ausfall der Klimaanlage bis hin zum Wasserrohrbruch. Nicht immer ist die Ursache trivial, beispielsweise bei einer feuchten Wand, sodass gegebenenfalls nur Hinweise aufgenommen werden können. Ein Mitarbeiter hat gelernt, die richtigen Fragen in einer sinnvollen Reihenfolge zu stellen und relevante Informationen von irrelevanten zu unterscheiden, damit auch für den Meldenden das Gespräch in einer hektischen Situation so professionell wie möglich verläuft. Im Übrigen hat auch der Meldenden ein Interesse daran, das Gespräch kurz zu halten, um sich dem Tagesgeschehen zu widmen.

---

<sup>4</sup> Basierend auf einer Suche in der Universitätsbibliothek Paderborn, Google Scholar und nach Aussagen der synexs GmbH gibt es für die Vorfälle im Facility-Management kein Schema, nach welchem die Probleme klassifiziert und priorisiert werden. Bei der synexs GmbH werden unternehmensintern Störungskategorien benutzt, welche den Vorfall in ein Prioritätenschema einordnen. Die Details, wie die zu erfassenden Merkmale und Entscheidungsregeln, welche beispielsweise die Auswahl der benötigten Dienstleister beschreiben, erfolgt individuell durch den Supportmitarbeiter. Verglichen dazu ist die Diagnose-Literatur im medizinischen Bereich umfangreich.

## 1.2 Zielsetzung

Das Ziel dieser Arbeit ist ein domänenspezifisches Dialogsystem zu entwickeln. Dies soll Vorfälle im Facility-Management, die bislang vom Telefonsupport entgegengenommen werden, auf einer textbasierten Schnittstelle in natürlicher Sprache entgegennehmen. Daraufgehend ist eine Problemeinschränkung und weitere Klassifizierung des Vorfalls, angereichert mit strukturierten Informationen, vorzunehmen. Diese Klassifizierung soll dazu dienen den Vorfall einzuschätzen und Lösungsmaßnahmen aufzuzeigen, um damit die Mitarbeiter im Telefonsupport zu entlasten.

Ziel ist es, geeignete Lösungen für die Verarbeitung von natürlicher Sprache zu finden, welche die in 1.1 beschriebenen Probleme überwinden. Diese Lösungsansätze sind zu vergleichen und zu bewerten.

Des Weiteren ist ein Datenmodell über die Vorfälle im Facility-Management sowie Wege, um jene strukturiert zu erfassen, auszuarbeiten. Dazu wird domänenspezifisches Datenmaterial benötigt. Dieses ist in Form eines Vorfallkatalogs für die Analyse der Datenstruktur und zur Klassifizierung der Vorfälle zu erstellen.

Die textbasierte Schnittstelle ist mittels Kommunikationsdiensten umzusetzen, die auf eine hohe Akzeptanz bei den Meldenden stößt, sowie eine breite Verfügbarkeit bei den Endgeräten aufweist.

## 1.3 Vorgehensweise

Die Arbeit wird zielführend folgendermaßen gegliedert:

Im ersten Kapitel werden die Arbeit und die Zielsetzung genauer vorgestellt und erklärt. Die Grundlagen von Chat-Bots als auch des Facility-Managements werden im zweiten Kapitel erläutert um daraus den Problembereich aufzuzeigen. Das folgende dritte Kapitel stellt einen Überblick über den Stand der Technik im Bereich der Algorithmen und Methoden sowie Frameworks für Chat-Bots zusammen. Die Realisierung beginnt mit dem vierten Kapitel, in dem ein Lösungskonzept für die Konstruktion eines Chat-Bots am Beispiel des Facility-Managements präsentiert wird. Hier erfolgt eine Auswahl und Bewertung der vorhandenen Chat-Bot Frameworks, sowie die Definition von domänenspezifischen Informationen, um die Problemeinschränkung von Nutzern gemeldeten Vorfällen im Rahmen des Facility-Managements zu ermöglichen. Die Implementierung eines Chat-Bots sowie die Komponenten und Schnittstellen in den verschiedenen Frameworks werden in Kapitel Fünf beschrieben. Die Evaluation der Framework-Ergebnisse wird im Kapitel Sechs durchgeführt. Zum Schluss setzt das Kapitel Sieben ein Fazit und gibt einen Ausblick, wie mit den gewonnenen Erkenntnissen weiter geforscht werden kann.



## 1.4 Motivation

Die Entwicklung der letzten Jahre und insbesondere Monate hat gezeigt, dass Dialogsysteme an Akzeptanz und Marktreife gewonnen haben (vgl. [HUB-ol]; [TRA-ol]). Ebenfalls hat sich bestätigt, dass natürlichsprachliche Dialogsysteme notwendig sind, da im Gegensatz dazu niemand feststehende Einzelwortbefehle lernen möchte, um mit einem Dialogsystem zu interagieren.

Ob Amazon Alexa oder Apple Siri, Dialogsysteme werden unsere Gesellschaft nachhaltig verändern. Daher wird an dieser Stelle nicht vom Benutzer der Systeme, sondern vom Menschen allgemein gesprochen.

Die Auswirkungen betreffen den Menschen gleichermaßen wie Dialogsysteme einsetzende Unternehmen. Die Vorteile von Dialogsystemen werden im Folgenden motiviert, aber auch Nachteile und Grenzen von ebendiesen aufgezeigt.

Dialogsysteme aus dem Blickwinkel des Menschen:

- **ermöglichen** Menschen mit Behinderung, wie Sehbehinderung oder körperliche Behinderung, Teilhabe an der Gesellschaft;
- **assistieren** Menschen, die durch ihre Umwelt oder Tätigkeit eingeschränkt sind, wie Autofahrer oder Ärzte im Operationssaal, durch eine kontaktlose Interaktionsmöglichkeit über den akustischen Kanal;
- **verbessern** den Umgang mit digitalen und physischen Systemen, was zu einer persönlichen Leistungssteigerung durch mehr Effizienz führt;<sup>5</sup>
- **ersetzen** Arbeitsplätze und damit menschliche Arbeitskraft mehr, als für die Entwicklung von Dialogsystemen benötigt wird.<sup>6</sup>

Dialogsysteme aus Sicht der Unternehmen:

- **bilden eine kommunikative Wissensdatenbank**, da unternehmensinternes Wissen jederzeit innerhalb des Dialogsystems und verbundener Systeme abrufbereit gespeichert wird und nicht durch den Wechsel von Mitarbeitern verloren geht;

---

<sup>5</sup> Graphische Oberflächen, die in Dialogen Daten erfassen, sind mit dem als „Taylorismus“ bezeichneten Prinzip konfrontiert. Durch eingeschränkte Handlungs- und Entscheidungsspielräume kommt es zu negativen Auswirkungen (In der Arbeitspsychologie auch als „der lange Arm der Arbeit“ bekannt). (vgl. [NuRö88], S. 50ff+150ff) Normen wie die ISO 9241 oder Prinzipien von Nielsen und Shneiderman für die Gestaltung von Oberflächen entschärfen zwar das Problem. Dieses könnte jedoch durch Dialogsysteme, die eine natürliche Interaktion erlauben, gänzlich eliminiert werden.

<sup>6</sup> Eine offene Frage, welche über den Rahmen dieser Masterarbeit hinausgeht, ist, ob die freien Arbeitsressourcen genutzt werden könnten, um Tätigkeitsfelder, die bislang Personalmangel aufwiesen, zu beleben, oder ob die Qualifizierung dafür nicht adäquat ist. Ähnliches gilt für Berufe der Zukunft, die in der Regel eine höhere Qualifikation erfordern. Ein potenzieller Lösungsansatz ist das bedingungslose Grundeinkommen.

- **sparen Kosten**, denn Personalkosten bilden in den meisten Branchen den größten Kostenblock im Unternehmen. Dialogsysteme helfen, diese zu optimieren. In den Anfängen des Telebankings musste jeder Anrufer, gegebenenfalls mit Wartezeit, zu einem Mitarbeiter der Bank durchgestellt werden. Dialogsysteme übernehmen diese Aufgabe und Mitarbeiter werden nur zugeschaltet, wenn der Kunde dies wünscht. Die Bank kann, bei gleichem Service für den Kunden, Personalkapazitäten einsparen;
- **ermöglichen Gestaltungsfreiheiten**, da ohne notwendige Bindung an eine Oberfläche die Aufmachung von digitalen und physischen Systemen umgestaltet werden kann. Probleme wie ein zu kleiner Bildschirm oder das Layout von Oberflächen (inklusive der Auswahl der funktionalen Bedienelementen) wären daher minder relevant;
- **entfernen sich vom Kunden**, da ein zuvor via Telefon bestandener Kundenkontakt wegfällt. Emotionale Bindung, Vertrauen und Zuverlässigkeit sind für Kunden relevant und müssen im Bedarfsfall sichergestellt werden. Eine vollständige Automatisierung ohne Kontaktmöglichkeit ist daher nicht ratsam. (vgl. [NuRö88], S. 20f)

Zusammenfassend übertreffen die positiven Aspekte die negativen Gesichtspunkte, welche zudem durch dargestellte Ansatzpunkte zum Gegensteuern handhabbar werden. Für das konkrete Beispiel des Facility-Managements lassen sich Vorfälle durch ein Dialogsystem, bei gegebener Marktreife und Überwindung der in 1.1 skizzierten Probleme, schneller erfassen und auswerten, sodass eine Behebung des Vorfalls effizienter und zügiger abläuft. So wird die Meldungsannahme durch den Chat-Bot sichergestellt und Mitarbeiter können sich in Stresssituationen auf die durch den Chat-Bot klassifizierten vorrangigen Vorfälle konzentrieren. Der Chat-Bot nimmt eine Problemeinschränkung vor und zeigt dem Mitarbeiter eine oder mehrere Interpretationen des Vorfalls auf. Letztere ermöglicht Anregungen des Chat-Bots aufzugreifen und für die Bearbeitung mit einzubeziehen. Folglich lässt sich der Servicebereich mit geringerem Personalaufwand betreiben.

## 2 Problembereich

Das folgende Kapitel erläutert die Grundlagen des Problembereiches. Dazu wird das Problemfeld der Chat-Bots und des Facility-Managements beschrieben. In dem Abschnitt Handhabung bei Facility-Management Unternehmen folgt ein Erfahrungsbericht aus dem Praxisalltag von Unternehmen.

### 2.1 Chat-Bots

Chat-Bots sind Dialogsysteme, die in einem Diskurs mit menschlichen Dialogpartnern diverse Leistungsmerkmale annehmen; etwa um Fragen zu beantworten, Befehle auszuführen oder um belanglose Konversationen zu führen.

Das Thema umfasst ein interdisziplinäres Feld der Sprachwissenschaft und Informatik, in dem es gilt, verschiedene Herausforderungen zu unterschiedlichen Unterproblemen und Themengebieten zu meistern. Diese werden im Folgenden erläutert. Ebenfalls werden Definitionen und Grundannahmen, der Aufbau und die Komponenten eines Chat-Bots, sowie die Herausforderungen bei der Konzeption von Chat-Bots herausgestellt.

#### 2.1.1 Allgemeines

Ein Dialog bezeichnet im Rahmen dieser Arbeit eine adressierte Kommunikation zwischen zwei oder mehr Parteien; zum einen das Dialogsystem und zum anderen der menschliche Akteur. Der Akteur wird in diesem Zusammenhang bewusst im Singular verwendet, um sich von der Komplexität der Multilogie (mit mehr als einem menschlichen Akteur) abzugrenzen.

Der Begriff Dialogsysteme wird von Rieser und Kruijff-Korbayová (vgl. [CrIm06], S. 78) wie folgt definiert:

„Das Anwendungsgebiet Dialogsysteme beschäftigt sich mit Entwurf, Umsetzung und Evaluation natürlichsprachlicher Benutzerschnittstellen. Dazu gehört die Erforschung und vereinfachte Abbildung menschlicher Dialogfähigkeit, wie zum Beispiel die Fähigkeit, Dialogstrukturen zu erkennen, kontextabhängige Interpretationen und Entscheidungen zu treffen und Dialogbeiträge zu planen.“

Für das weitere Verständnis von „natürlichsprachlich“ folgt zunächst ein Exkurs in die Unterscheidung von formalen und natürlichen Sprachen.

Eine **formale Sprache** verfügt über ein Alphabet (oder auch Zeichenvorrat) und Regeln zur Herstellung von Ausdrücken (Formationsregeln). Wird die formale Sprache durch eine kontextfreie Grammatik beschrieben, erlaubt dies einen definierten Leseprozess (Interpretation) von Ausdrücken. Die Prüfung einer Zeichenkette im Sinne der Grammatik-

regeln kann beispielsweise durch ein Programm (sogenannte Parser) automatisiert werden. Die Ausdrücke können im Kontext eines Chat-Bots für Befehle stehen, die eine korrekte Ausführung von Anweisungen eines Benutzers ermöglichen. (vgl. [Lew85], S. 512+691)

```

<Befehlsfolge> ::= <Befehl> | <Befehl> <Befehlsfolge>
<Befehl>       ::= <Objekt> <Tätigkeit>
<Objekt>       ::= Videomaterial | Audiostück
<Tätigkeit>    ::= <Starten> | <Stoppen>
<Starten>      ::= abspielen | starten
<Stoppen>      ::= stoppen | pausieren | anhalten

```

*Abbildung 2.1: Beispiel einer kontextfreien Grammatik in Backus-Naur-Form*

Die Backus-Naur-Form ist ein Beispiel für eine kompakte formale Metasprache zur Darstellung kontextfreier Grammatiken. Ableitungsregeln stehen auf der linken Seite der Zeichenfolge „::=“ und definieren die Nichtterminalsymbole aus Terminal- und Nichtterminalsymbolen, wobei der vertikale Strich „|“ als Alternative dient. Um eine Regel abzuleiten, werden die Nichtterminalsymbole (in spitzen Klammern) der rechten Seite durch die Definition dieser ersetzt. Eine Regel ist vollständig abgeleitet, wenn auf der rechten Seite ausschließlich Terminalsymbole stehen. (vgl. [HMU02], S. 180ff) Weitere Erläuterungen zu formalen Sprachen finden sich bei Hopcroft und eine Hierarchie von Klassen formaler Grammatiken bei Chomsky.

Abbildung 2.1 zeigt exemplarisch, wie Befehlsfolgen in dieser formalen Sprache definiert werden können. Auf diese Weise kann der Befehl „Audiostück abspielen“ gebildet werden. Jedoch sind nur Ausdrücke zulässig, die exakt der definierten Sprache entsprechen.

Die **natürliche Sprache** ist eine historisch gewachsene Alltagssprache, die Dialekte und Soziolekte (Sprachvarianten) aufweist. Zudem wird die Sprachstruktur unmittelbar von der Gesellschaftsstruktur bestimmt, welche eine Matrix kommunikativer Handlungen bildet. Im Gegensatz zu der formalen Sprache weist natürliche Sprache nicht nur strukturelle und lexikalische Mehrdeutigkeiten, sondern auch Vagheiten auf. (vgl. [Lew85], S. 740)

Beispiele für lexikalische Mehrdeutigkeit sind Homonyme wie die Bank (das Geldinstitut und die Sitzgelegenheit) oder „der Teich schlug leise flache Wellen“ für strukturelle Mehrdeutigkeit (schlug der Teich leise oder sind die Wellen leise und flach) und lexikalische Vagheit (von leise und flach).

**Natürlichsprachliche** (auch natürlichsprachige) Dialogsysteme sind dadurch definiert, dass sie Eingaben (Wörter und/oder Sätze) einer real existierenden Sprache verarbeiten können, um daraus Informationen zu extrahieren, diese zu verarbeiten und schließlich eine ebenfalls in natürlicher Sprache formulierte Antwort zu erzeugen. Ein System, welches lediglich vordefinierte Kommandos einer natürlichen Sprache entgegennimmt und verarbeiten kann, ist somit ebenso ein natürlichsprachliches Dialogsystem, wie ein Chat-Bot, der vollständige Sätze interpretiert. Auch wenn ein Kommando-Bot eher wie ein

System basierend auf einer formalen Sprache mit einer limitierten Anzahl an Regeln der Grammatik wirkt, unterscheiden sich beide Systeme vielmehr in Formulierung, Verständnis und Verhalten. Je menschenähnlicher diese drei Komponenten sind, desto natürlicher ist das Dialogsystem. (vgl. [Ber13], S. 372)

Der Leistungsumfang natürlichsprachlicher Dialogsysteme, die besonders menschenähnlich sind, umfasst nach Berg folgende Eigenschaften die auch untereinander in Verbindung stehen (vgl. [Ber13], S. 374ff):

- **Gemischte Initiative**

Interaktive beschreibt den leitenden Gesprächsteilnehmer. In der Dialogführung wird zwischen vom Nutzer geführte, vom System geführte und gemischt-initiative Dialoge unterschieden. Gemischte Initiative geben dem Nutzer die Möglichkeit, den Dialogablauf zu beeinflussen und helfen dennoch dabei, ein bestimmtes Ziel zu erreichen. Die Dialogführung wird darüber hinaus durch die Art oder den Zweck eines Chat-Bots bestimmt und eingeschränkt. Ein Dialogsystem, welches dazu konzipiert ist, eine vordefinierte Anzahl an Befehlen in einer Domäne entgegenzunehmen, um daraufhin Aktionen auszuführen, ist weniger für eine gemischte Dialogführung geeignet. Dahingegen ist die vom System geführte Unterstützung zum Erreichen bestimmter Ziele tendenziell sinnvoll. Der vom System geführte Dialog ermöglicht eine zuverlässigere Texterkennung, da die Antwortmöglichkeiten des Benutzers bereits eingeschränkt werden (vgl. [ABD+01], S. 27).

- **Adaptivität**

Konfigurierbare Systeme lassen sich auf die Vorlieben des Benutzers anpassen, beispielsweise bei der Höflichkeitsform oder Wortwahl. Bei einem adaptierbaren System wird die Einstellung stets vom Nutzer vorgenommen, wohingegen das adaptive System das Nutzerverhalten analysiert und sich diesem automatisch anpasst.

- **Implizite Bestätigung**

Eine Rückmeldung, welche das Verständnis der vorherigen Eingabe bestätigt, ist monoton; sie wird durch eine implizite Bestätigung attraktiver. Beispielsweise indem die durch den Nutzer mitgeteilte Information in einer folgenden Aussage des Systems aufgegriffen wird.

- **Nachfragen und Ambiguitätenresolution**

Sprachliche Mehrdeutigkeiten sollten, wenn diese nicht im Kontext aufgelöst werden können, durch eine Nachfrage an den Nutzer geklärt werden.

- **Korrektur**

Eingaben jederzeit revidieren oder verbessern zu können ist essenziell für einen natürlichen Dialog. Aber auch Teilkorrekturen, die bestimmte Details verändern, sind zu beachten.

- **Überbeantwortung**

Eine Überbeantwortung bedeutet, dass dem System mehr Informationen mitgeteilt werden, als erfragt wurden. Im folgenden Dialogverlauf sind dann entsprechende Detailfragen durch das System zu vermeiden.

- **Verneinungen**

Verneinungen, die Möglichkeiten ausschließen oder einschränken, sind als diese vom System zu erfassen. Dies ist durch eine alleinige Schlüsselwortanalyse der Eingabe nicht möglich.

- **Diskurs und Rückbezüge**

Kontextwissen über den Nutzer und aus dem bisherigen Diskurs sollte gespeichert und bei der Verarbeitung bedacht werden.

- **Interpretation von Umgangssprache**

Umgangssprache (und Kurzsprache) sowie Formulierungen stellen derzeit das größte Problem beim Verstehen natürlicher Sprache dar. Hierbei wird aktueller Wortschatz und Kontextwissen benötigt.

- **Formulierung/Sprachgenerierung**

Das System sollte leicht verständlich, eindeutig, informativ und der Situation angepasst antworten. Formulierungen sollten den griceschen Maximen (siehe unten) entsprechen. Zur Prävention von Monotonie können Antworten abwechslungsreich, ohne jedoch die inhaltliche Aussage zu verändern, gestaltet werden.

- **Sozialverhalten**

Den Benutzer zu begrüßen, Meinungen zu äußern oder einen Dank zu erwidern, verleiht dem System ein menschlicher wirkendes Verhalten.

- **Spracherkennung und Sprachsynthese**

Neben textbasierten Chat-Bots gibt es die Möglichkeit, die Eingabe über auditive Spracherkennung (*Automated Speech Recognition*), beziehungsweise die Ausgabe mittels Sprachsynthese (*Text-To-Speech*) audiobasiert abzuwickeln. Diese vor- und nachgelagerten Module eines Chat-Bots werden im Kontext dieser Arbeit und für textbasierte Ein- und Ausgaben nicht weiter betrachtet. Folglich wird die Verarbeitung von Spontansprache, die unter anderem Wiederholungen und Räuspern beinhalten kann, vernachlässigt.

Darüber hinaus fassen Ginzburg und Fernández (vgl. [CFL13], S. 431f) weitere Problemstellungen im Dialog zusammen; selbst beantwortende Aussagen, Multiloge, Meinungsverschiedenheiten, eingeschränktes Verständnis, unvollständige Äußerungen und der Umgang mit einzelnen Satzfragmenten, die aber in der Äußerung vollständig sind. Im

Folgendes wird erklärt, wie durch Klassifizierung in die Art der Äußerung die Interpretation dieser Aussagen verbessert wird:

Die Sprechakttheorie von Searle, genauer gesagt der illokutionäre Akt, besagt, dass mit jeder Aussage auch die Wirklichkeit verändert wird. Ein Beispiel stellt der Richter bei der Urteilsverkündung dar, indem er den Angeklagten für schuldig erklärt. Daraus folgt die These, dass mit dem Gesagten auch ein bestimmtes Ziel verfolgt wird, was im Umkehrschluss beeinflusst, was gesagt wird. Daher wird versucht, Aussagen zu kategorisieren, um die maschinelle Interpretation zu erleichtern. Unter anderem haben Carletta et al. sowie Core und Allen Taxonomien erstellt, die Aussagen gruppieren. Beispielsweise werden „einen Dialog beginnende“ von „auf einen Dialog antwortende“ Intentionen eines Sprechers differenziert. (vgl. [CFL13], S. 432f; [CoAl97], S. 3f; [CID+97], S. 14f)

Berg relativiert jedoch die Wichtigkeit von menschlichem Verhalten im Vergleich zu den Verarbeitungsfähigkeiten, insbesondere für die Sprachproduktion. Zuverlässigkeit ist daher laut Berg wichtiger als Natürlichkeit. Denn sonst wird dem System mitunter mehr Intelligenz beigemessen, als das System in Wirklichkeit hat. Dies könnte somit zu unrealistischen Erwartungshaltungen führen. (vgl. [Ber13], S. 373)

Die Sprachproduktion sollte klare und eindeutige Aussagen produzieren, was den Konversationsmaximen entspricht, welche Grice aufgestellt hat ([Gri75], S. 47):

- Maxime der Quantität: Sei so informativ wie nötig!
- Maxime der Qualität: Sage die Wahrheit und stelle keine Vermutungen als erstere dar!
- Maxime der Relevanz: Sage nichts, was nicht zum Thema gehört und vollziehe keine plötzlichen Themenwechsel!
- Maxime der Modalität: Sei konkret und eindeutig!

Chat-Bots werden daraufhin unterschieden, ob sie domänenspezifisch oder universell konzipiert und eingesetzt werden können. Innerhalb einer Domäne können Fachtermini definiert und sprachliche Mehrdeutigkeiten aufgelöst oder eingeschränkt werden. Es gibt ein bestimmtes Ziel, eine Reihe von Aufgaben oder Problemen, welche ein domänenspezifischer Chat-Bot beherrschen muss, anstatt die ganze Bandbreite unseres täglichen Alltags beherrschen zu müssen.

Im Gegensatz zu audiovisueller und zwischenmenschlicher Kommunikation erhält ein textbasierter Chat-Bot nur ein Textfragment als Eingabe. Es gibt keine auditive Wahrnehmung, aus der sich Eigenschaften der Prosodie wie Akzent, Betonung, Tempo, Rhythmus und Pausen gewinnen lassen. Keine visuelle Wahrnehmung, die Gesten und Mimik erfasst und interpretieren kann. Diese und weitere sensorische Fähigkeiten eines Menschen fehlen einem Chat-Bot, wodurch beispielsweise die Erkennung des Gemütszustandes unmöglich ist, ohne Hinweise in der textuellen Eingabe zu geben.

## 2.1.2 Aufbau und Komponenten

Ein textbasierter Chat-Bot besteht aus einer Komponente für das Verständnis natürlicher Sprache (*Natural language understanding*, NLU), welche die Wortsequenz aus Nutzereingaben analysiert und interpretiert, um daraus eine semantische Repräsentation zu erzeugen; dem Dialog Manager, der diese Daten erhält und den Dialogfluss steuert, indem vorherige Beiträge, integriertes Wissen und Informationen (aus dem Backend) ausgewertet werden, um daraus die Antwort und nächsten Schritte im Dialog zu bestimmen; und einem Modul für die Sprachproduktion (*Natural language generation*, NLG), das im Anschluss die Antwort, die der Nutzer erhält, generiert. Abbildung 2.2 visualisiert die Architektur des Dialogsystems. (vgl. [CFL13], S. 443; [SmHi94], S. 47)

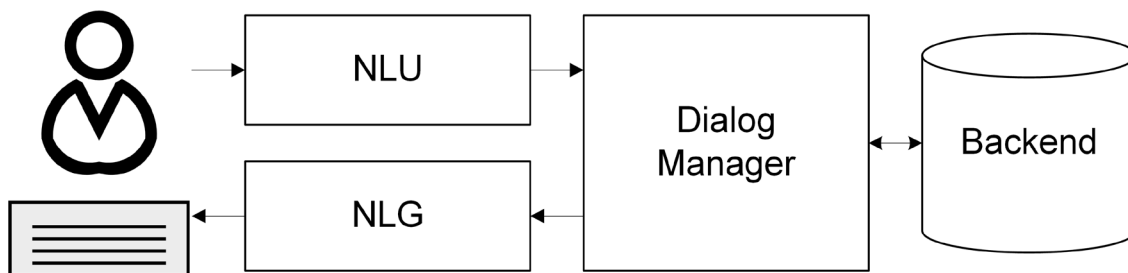


Abbildung 2.2: Komponenten eines textbasierten Dialogsystems (vgl. [CFL13], S. 444)

Der Dialog Manager wird oft als die zentrale Schnittstelle im Dialogsystem verstanden. Hier werden Schnittstellen zu externen Datenquellen, Wissen über die Domäne, den User und Dialogstrukturen mit der aktuellen Eingabe und dem Zustand im Dialogverlauf verknüpft. Aus externen Datenquellen werden Informationen abgerufen, wenn der Dialog Manager erkennt, dass diese im Gesprächsverlauf benötigt werden. Dies sind beispielsweise unternehmensinterne Produktdaten oder öffentlich zugängliche Daten wie Wetterinformationen.

Zudem werden Dialog Manager basierend auf der Architektur für die Verwaltung der Dialoge unterschieden. Es gibt drei Hauptgruppen:

- Beim Dialogmanagement mit einer endlichen Anzahl an vordefinierten Zuständen hat das System die volle Dialogkontrolle. Jede Frage muss durch Benutzer exakt beantwortet werden, bevor das System den Zustand wechselt.
- Das rahmenbasierte Dialogmanagement verwendet Platzhalter, um darin Werte zu speichern, die im Rahmen einer Unterhaltung vom System oder durch Initiative des Benutzers erfragt beziehungsweise überbeantwortet wurden. Somit sind Abweichungen im Dialogverlauf möglich, wodurch erfahrenen Benutzern eine zügige Verarbeitung ermöglicht wird. Außerdem können fehlende Informationen durch gezielte Rückfragen eingeholt werden.
- Als inferenzbasiert wird ein Dialogmanagement bezeichnet, wenn ein rational handelnder Akteur durch Planungsaufgaben vorangetrieben und dessen Verhalten von Schlussfolgerungen bestimmt wird. Durch systematisches Ausprobieren von



Zustandsänderungen wird versucht, eine geeignete Folge an Änderungen zu finden, welche einen definierten Systemzustand erzeugt.

Der Akteur wird mit festen Grundsätzen, Plänen, Zielen und einem Theorembe-  
weiser für maschinengestütztes Beweisen ausgestattet. Zudem kann der Akteur  
auch mit Überzeugungen, Wünschen und Absichten ausgestaltet werden.

(vgl. [CFL13], S. 444ff; [ABD+01], S. 27f; [RiLe11], S. 34)

Die NLU-Komponente umfasst Algorithmen und Methoden, die meist basierend auf einem Lexikon mit semantischen und lexikalischen Beziehungen (Ontologien) die Aussage parsen und mit grammatikalischen Regeln in eine semantische Zuordnung überführen. Im Detail muss die NLU-Komponente immer auf die Sprache und das verwendete Gesamtsystem abgestimmt und angepasst werden. Insbesondere müsste die NLU-Komponente für das inferenzbasierte Dialogmanagement logische Schlussfolgerungen durch Behauptungen in Prädikatenlogik bereitstellen. (vgl. [LaDr98], S. 209)

Für das NLG-Modul werden zwei Arten der Sprachproduktion unterschieden:

- **Abrufende Systeme**, die basierend auf einer durch Schlüsselwörter geordneten Sammlung an Textvorlagen geeignete Textbausteine über ein Regelwerk identifizieren und für die Textausgabe verwenden. Diese vordefinierten Aussagen können gegebenenfalls noch parametrisiert werden, um im Programmablauf vordefinierte Informationen zu vermitteln.
- **Generierende Systeme** ermöglichen eine Produktion von flexiblen Aussagen oder gänzlich neuen Texten. Ähnlich wie bei der NLU-Komponente ist die konkrete Vorgehensweise der Textherstellung abhängig von der Sprache und Domäne. Bestandteile und Verarbeitungsschritte umfassen die Festlegung der auszudrückenden Information, Inhaltsorganisation, Textplanung und Linearisierung, Satzplanung sowie grammatische und lexikalische Entscheidungen.

(vgl. [CEE+10], S. 633ff)

## 2.2 Facility-Management

Die deutsche Literatur befasst sich vielfach mit einer genauen Deutung des englischen Begriffes (vgl. [BOH+01], S. 1ff; [MoRo01], S. 27). Die im Weiteren als geltend angenommene Definition von Facility Management ist der GEFMA, dem Deutschen Verband für Facility Management e.V., zuzuordnen und lautet wie folgt:

„Facility Management (FM) ist eine Managementdisziplin, die durch ergebnisorientierte Handhabung von Facilities<sup>7</sup> und Services<sup>8</sup> im Rahmen geplanter, gesteuerter und beherrschter Facility Prozesse eine Befriedigung der Grundbedürfnisse von Menschen am Arbeitsplatz, Unterstützung der Unternehmens-Kernprozesse und Erhöhung der Kapitalrentabilität bewirkt.

Hierzu dient die permanente Analyse und Optimierung der kostenrelevanten Vorgänge rund um bauliche und technische Anlagen, Einrichtungen und im Unternehmen erbrachte (Dienst-) Leistungen, die nicht zum Kerngeschäft gehören.“

Im Folgenden werden die Strukturen und Aufgabenbereiche zu FM beschrieben, die für diese Arbeit relevant sind. Auf vertiefende Details wie wissenschaftlichen Fragestellungen, Lebenszyklusphasen und IT-Transformation wird verzichtet.

### 2.2.1 Strukturen

Eine erste Übersicht zu den Akteuren und Aktivitäten im FM schafft Abbildung 2.3 mit einem Prozessmodell. FM-Kunden sind die Eigentümer von Facilities. Sie definieren durch Anforderungen Erfordernisse oder Erwartungen an das FM. Diese Anforderungen umfassen beispielsweise die Erhaltung baulicher und anlagentechnischer Werte und sind durch Service Levels (Messgrößen) spezifiziert und vertraglich in einem Service-Level-Agreement (SLA) geregelt. Service Levels beschreiben zum Beispiel die Reinigungszyklen einer Unterhaltsreinigung oder Reaktionszeiten bei technischen Störungen. FM-Dienstleister setzen die einzelnen Tätigkeiten in Form geplanter, gesteuerter und beherrschter Facility-Prozesse um. Daraus resultieren Facility-Produkte, deren Merkmale den Kundenanforderungen entsprechen sollen. Dieser Nutzen kann durch die Kundenzufriedenheit gemessen werden. (vgl. [GEFMA-100-1], S. 2)

---

<sup>7</sup> Beispiel: Anlagen, Geräte aber auch Arbeitsmittel und Energie sowie Hard- und Software

<sup>8</sup> Beispiel: Ver- und Entsorgungsdienstleistungen oder Planungs- und Beratungsleistungen

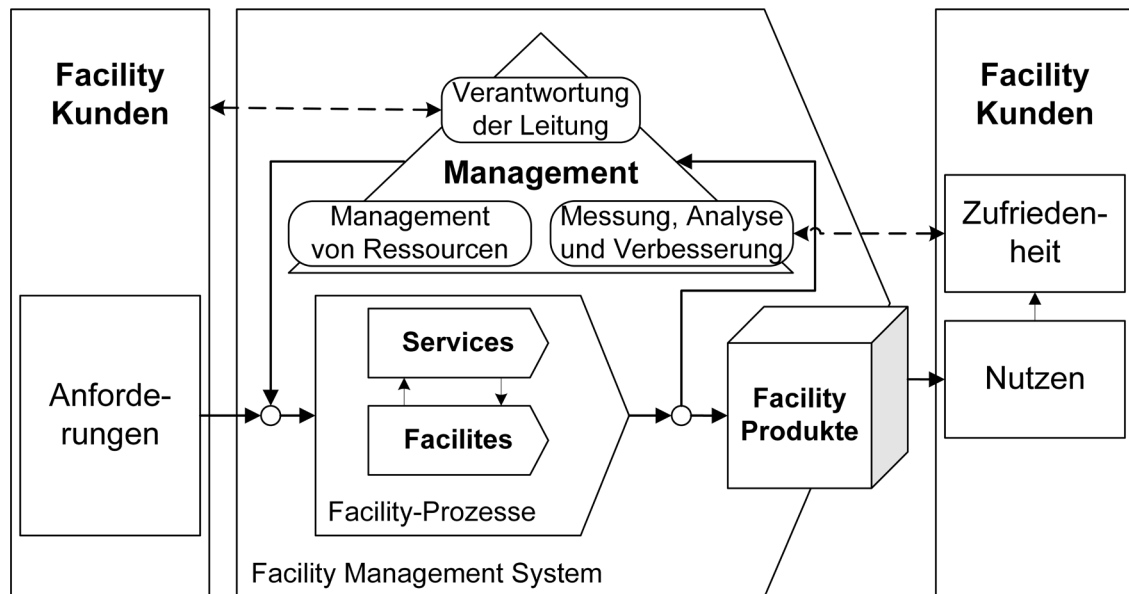


Abbildung 2.3: Prozessmodell für FM (vgl. [GEFMA-100-1], S. 2)

Der Ablauf des operativen Geschäfts (siehe Abbildung 2.4) wird durch fünf Gruppen wesentlich gekennzeichnet: Nutzer von Facilities, First-Level-Support, Dispatcher, CAFM und Dienstleister. Das CAFM<sup>9</sup> (*Computer Aided Facility Management*) beschreibt ein abstraktes Datenverarbeitungssystem mit Informationen zu Facilities, Kunden und Dienstleistern, Vorfällen sowie weiterem domänenspezifischem Wissen.

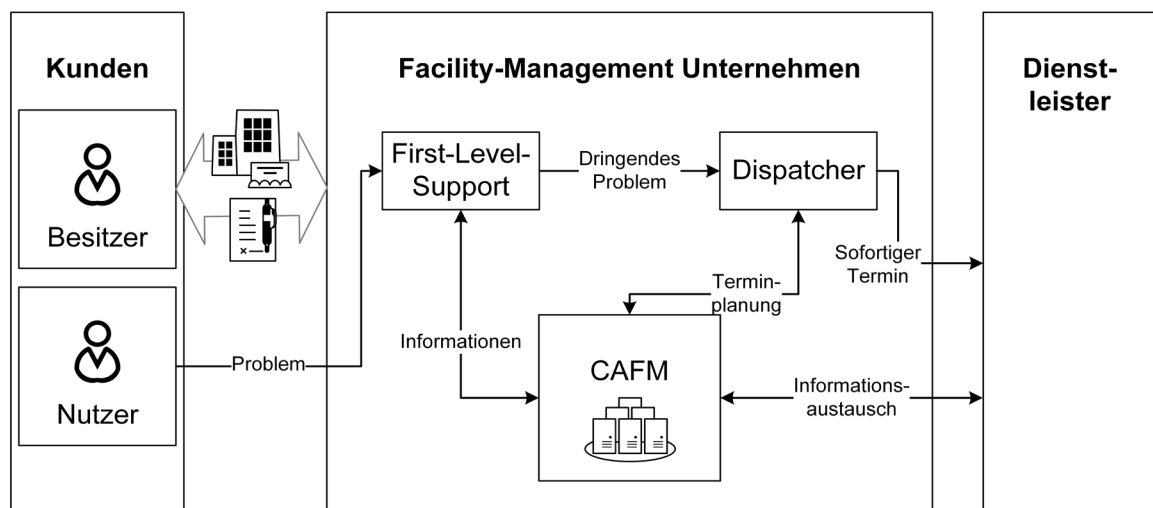


Abbildung 2.4: Operatives Geschäft

Nutzer sind alle Personen, die in Verbindung zu der Facility stehen, beispielsweise der Facility-Besitzer, das Wachpersonal oder Geschäftsleute. Tritt ein Vorfall (siehe Kapitel 2.2.2) auf, etwa durch einen Defekt in der Beleuchtung, wenden sich die Nutzer an das

<sup>9</sup> Die GEFMA zertifiziert CAFM-Softwareprodukte

FM-Unternehmen und schildern dem First-Level-Support das Problem. Der First-Level-Support nimmt die Schilderung auf und erfasst diese im CAFM. Weiterhin entscheidet dieser, ob es sich dabei um ein dringendes Problem handelt. Bei dringenden Problemen wird der Dispatcher kontaktiert, der im Regelfall alle vertraglich fixierten Aufgaben terminlich organisiert und plant. Der Dispatcher kontaktiert dazu einen oder mehrere für diesen Vorfall relevante Dienstleister, die im Rahmen der vertraglich geregelten Anforderungen das Problem lösen können.

Der Problembereich dieser Arbeit fokussiert sich auf die Schnittstelle zwischen dem Nutzer und dem FM-Unternehmen, genauer gesagt dem First-Level-Support. Die Mitarbeiter im First-Level-Support können durch die Kommunikation mit dem Nutzer den Vorfall einordnen und entsprechende Maßnahmen treffen. Durch die teilweise über die Jahre gesammelten Erfahrungen der Mitarbeiter sind präzise Angaben an den Dienstleister nicht selten. Damit können eine zusätzliche Prüfung der zu erledigenden Aufgaben durch den Dienstleister, fehlendes Werkzeug oder Material am Ort, fehlende Qualifikation oder gar die Bestellung des falschen Dienstleisters vermieden werden. Beispielsweise muss der First-Level-Support bei der Information, dass eine Wand im Erdgeschoss bei einem mehrstöckigen Haus feucht ist, von verschiedenen Fehlerquellen ausgehen. Dies kann durch einen Wasserrohrbruch passiert sein, aber ein nicht geschlossener Wasserhahn oder ein undichtes Dach sind nicht immer ausgeschlossen. Ergänzende Informationen zum Gebäude über Baujahr, verbaute Materialien und bereits getätigte Instandsetzungsmaßnahmen können bei der Einschränkung der Ursache helfen.

### **2.2.2 Aufgabenbereiche und Prioritäten**

Der Aufgabenbereich des FM-Unternehmens umfasst alle mit dem Facility-Besitzer vertraglich vereinbarten Aufgaben. Das Standardleistungsverzeichnis<sup>10</sup> ist ein erster Anhaltspunkt für die vertragliche Ausarbeitung, denn Verträge werden dem individuellen Bedarf und Wünschen angepasst.

Die Aufgaben werden in planbare Arbeiten sowie Not- und Störfälle unterteilt. Zu den planbaren Aufgaben gehören beispielsweise die Wartung der Grünflächen oder die Reinigung der Außenfenster, die dann mit dem FM-Dienstleister abgestimmt werden. Die Not- und Störfälle treten unvorhergesehen auf und müssen entsprechend den vertraglichen Rahmenbedingungen behoben werden. Tabelle 2.1 zeigt die Kategorien und Zeitfenster der Not- und Störfälle aus dem Standardleistungsverzeichnis (StLV) von GEFMA und RealFM<sup>11</sup>. (vgl. [GEFMA-520], S. 23f)

---

<sup>10</sup> Erarbeitet von GEFMA und RealFM.

<sup>11</sup> RealFM e.V. Berufsverband für Facility und Real Estate Manager.

Tabelle 2.1: Not- und Störfälle im FM aus dem StLV (vgl. [GEFMA-520], S. 23f)

Kategorie	Beschreibung	Entstörzeit <sup>12</sup> zu Geschäftszeiten	Entstörzeit <sup>12</sup> außerhalb Geschäftszeiten
A	„Gefahr im Verzug“, Gefahr für Leib und Leben, Feuer, Einbruch, Überfall	unverzüglich	unverzüglich
B	<ul style="list-style-type: none"> <li>• Drohende elementare Störungen/Unterbrechungen des Kerngeschäfts des AG/Nutzers,</li> <li>• Störungen an kritischen Anlagen (Hauptstromversorgung, Kühlung von Serveranlagen etc.)</li> </ul>	1 Stunde	1 Stunde
C	<ul style="list-style-type: none"> <li>• Drohende merkliche Störungen/Unterbrechungen des Kerngeschäfts des AG/Nutzers,</li> <li>• Störungen an zentralen Anlagen (Heizung, Klima, Sanitär)</li> </ul>	4 Stunden	1 Tag
D	Drohende dauerhafte Schädigung der Bausubstanz	nach Rücksprache mit dem Auftraggeber	
E	Keine weiteren unmittelbar drohenden Auswirkungen	nach Rücksprache mit dem Auftraggeber	
F	Befreiung eingeschlossener Personen aus Aufzügen	30 Minuten	30 Minuten

In der Praxis werden die Kategorien in den FM-Unternehmen mitunter anders gehandhabt. Die synexs GmbH, ein FM-Unternehmen der HOCHTIEF Aktiengesellschaft, zeigt Defizite bei der Einteilung der Kategorien im StLV auf.<sup>13</sup>

Erstens lässt sich die drohende dauerhafte Schädigung der Bausubstanz (Kategorie D) mit den Kategorien A, B, C und E ausdrücken. Ein beschädigtes Dach mit herabstürzenden

<sup>12</sup> Das Zeitlimit der Entstörzeit gilt nach Meldung der Störung.

<sup>13</sup> Folgende Informationen wurden in einem Telefonat mit der synexs GmbH erarbeitet.

Elementen birgt die Gefahr für Leib und Leben, wohingegen eine beschädigte und nicht tragende Wand im Hinterhof keine unmittelbar drohenden Auswirkungen hat.

Zweitens sind die zeitlichen Unterschiede der Kategorien A, B, C und F durch die Grundlast an Aktivitäten im Entstörungsvorgang, wie der Anfahrt, so gering, dass eine Harmonisierung dieser Kategorien sinnvoll erscheint. Dagegen werden Störfälle minderer Priorität im StLV undifferenziert in Kategorie E abgebildet. Obwohl beide keine unmittelbar drohenden Auswirkungen haben und somit nach dem StLV der gleichen Kategorie angehören, ist eine schwergängige Tür innerhalb der unternehmensrelevanten Bereiche höher zu priorisieren, als in Sekundärbereichen.

Drittens verwendet nur ein Teil der FM-Kunden die Vorlage aus dem StLV. Kategorien werden durch den Kunden erweitert, ergänzt oder entfernt.

Darüber hinaus werden Vorfälle bei der synexs GmbH durch die Art und den Bereich, in dem der Vorfall auftritt beziehungsweise welche Bereiche der Vorfall betrifft, klassifiziert und in einem vierstufigen Prioritätenraster einsortiert. Zum Beispiel ist eine defekte Beleuchtung in einer repräsentativen Empfangshalle oder einem Schaufenster als problematischer einzustufen, als in einem Geräteraum.

Tabelle 2.2 fasst die Kategorien aus dem StLV mit den Erfahrungen der synexs GmbH zusammen.

*Tabelle 2.2: Störungs- und Prioritätenkategorien der Not- und Störfälle im FM*

Priorität	Beschreibung
1	<ul style="list-style-type: none"> <li>• Gefahr für Leib und Leben,</li> <li>• Drohende Störungen/unmittelbares und ernsthaftes Sicherheitsrisiko (Fehlfunktion der Sicherheitstechnik, Rohrbruch) und</li> <li>• Störungen an kritischen und zentralen Anlagen/Unterbrechungen des Kerngeschäfts des AG/Nutzers (Hauptstromversorgung, Datennetz, Kühlung von Serveranlagen etc.)</li> </ul>
2	<ul style="list-style-type: none"> <li>• Beeinträchtigung der Funktionalität und Optik in unternehmensrelevanten und repräsentativen Bereichen (zu kalt oder kein Strom am Arbeitsplatz, zerstörte Gegenstände, überfüllte Mülleimer, verschmutzte Sanitäranlagen oder Eingangshalle, Vandalismus) und</li> <li>• Beeinträchtigung der Sicherheit, der Gesundheit oder des Arbeitsschutzes, die jedoch ein geringes Risikopotenzial aufweisen (Beleuchtung defekt, Feuerlöscher abgelaufen, Verbandkasten unvollständig)</li> </ul>

3	Geringfügige Beeinträchtigung der Funktionalität und Optik in unternehmensrelevanten Bereichen (Kaffeeplecken auf dem Teppich, klemmende Tür in einem Arbeitsraum)
4	Unerhebliche Störungen und Schäden in unbedeutenden Bereichen wie Lager- und Technikräumen (Licht defekt, Fenster klemmt)

### 2.2.3 Aktivitäten im Vorfalldmanagement

In der IT Infrastructure Library (ITIL), welche eine Sammlung aus vordefinierten Prozessen, Funktionen und Rollen im Unternehmen zusammenstellt, findet sich auch eine Empfehlung für das Vorfalldmanagement (engl. Incident Management), das folgende Aktivitäten umfasst:

- die **Identifizierung** des Vorfalls durch Erkennung und Meldung;
- die **Registrierung**, durch die Erfassung im System (im Fall vom FM im CAFM);
- die **Kategorisierung** nach Gefährlichkeit, Priorität oder vertraglicher Regelung;
- die **Priorisierung**, um personelle und materielle Ressource besser auszulasten;
- die **Diagnose** für die Erfassung aller Problemerscheinungen;
- die **Eskalation**, falls der Support weitere Unterstützung von anderen Bereichen benötigt;
- die **Untersuchung**, wenn kein Lösungsweg aus der Vergangenheit bekannt ist und eine genaue Erforschung der Problemursache nötig wird;
- die **Lösung**, durch die vollständige Behebung des Vorfalls; und
- die **Schließung des Vorfalls** im System. (vgl. [vBo12], S. 140ff)

Relevant für die weitere Betrachtung sind die Aktivitäten von der Identifizierung bis zur Kategorisierung.





### 3 Stand der Technik

Dieses Kapitel beschreibt den in der Literatur und Praxis aktuellen Stand der Techniken im Bereich der Chat-Bots und deren verwendete Algorithmen und Methoden. Hierbei wird zwischen einzelnen Algorithmen und Methoden die für Teile eines Chat-Bots verwendet werden, und Frameworks, die komplette Abläufe mit den erwähnten Algorithmen und Methoden sowie Schnittstellen zu Kommunikationsdiensten wie Facebook Messenger oder Skype integrieren, unterschieden.

#### 3.1 Algorithmen und Methoden

Im Folgenden werden Algorithmen und Methoden für die Textvorverarbeitung, Textanalyse und Textproduktion, welche zum wesentlichen Funktionsumfang eines Chat-Bots gehören, vorgestellt und erläutert. Das Kapitel Klassifikation befasst sich mit Verfahren zur systematischen Einteilung beziehungsweise Einordnung von Objekten, in vorab festgelegte Klassen, durch welche diese auf einer abstrakten Ebene beschrieben werden.

Da maschinelles Lernen, künstliche neuronale Netzwerke und Entscheidungsbäume Bestandteil einiger nachfolgend vorgestellter Algorithmen sind, folgt an dieser Stelle eine kurze Einleitung dazu. Der Begriff **maschinelles Lernen** meint das Erkennen von Mustern aus Daten. Dazu gibt es verschiedene Varianten; das überwachte Lernen prüft die ermittelten Resultate mit einem Zielwert und bessert gegebenenfalls nach, wohingegen das unüberwachte Lernen einem algorithmisch bestimmten Ziel folgt. (vgl. [Bis09], S. 2f; [RiLe11], S. 21ff) Ein **künstliches neuronales Netzwerk** besteht aus mehreren Neuronen. Bei jedem Neuron gehen Informationen ein, die nach im Voraus definierten Regeln zueinander gewichtet werden. Eine Übertragungsfunktion berechnet aus den gewichteten Daten ein Ergebnis. Sollte dies über einem definierten Schwellenwert liegen, wird das Neuron aktiviert. Durch Lernalgorithmen, welche die Gewichte und den Schwellenwert anpassen, wird das Aktivieren des Neurons angepasst, bis die Mustererkennung den gewünschten Vorstellungen entspricht. Neuronale Netzwerke können die Anzahl der Neuronen und Verbindungen zueinander anpassen, um komplexere Bedingungen und Informationen zu verarbeiten. ([MST94], S. 3) **Entscheidungsbäume** können, neben anderen Einsatzgebieten, zur Klassifikation von Elementen, durch Entscheidungsregeln, dienen. Die Regeln werden manuell definiert oder mittels maschinellem Lernen erfasst. (vgl. [Qui87], S.83ff)

### 3.1.1 Textvorverarbeitung

Mit Textvorverarbeitung sind alle Schritte, die vor der Textanalyse stattfinden, gemeint. Durch lexikalische und syntaktische Analysen werden die Eingaben des Benutzers geprüft und gegebenenfalls verändert, um die Textanalyse im nachfolgenden Schritt zu erleichtern oder gar erst zu ermöglichen, da beispielsweise Rechtschreibfehler die Analyse erschweren.

Für das weitere Verständnis werden die Begrifflichkeiten und Grundlagen aus dem Bereich der **Morphologie** erklärt. Die Morphologie ist die wissenschaftliche Auseinandersetzung mit Wörtern; deren Struktur und Beziehungen zueinander. Dabei ist ein Wort die größte Einheit, die aus vielen kleinen Bausteinen, genannt Morpheme, zusammengesetzt ist. Ein Morphem hat eine grammatische Funktion durch inhaltliche und formale Merkmale. Silben dagegen besitzen dies nicht; sie haben phonologische Eigenschaften. Zur Verdeutlichung wird das Verb „abspielen“ durch Silbentrennung in „ab•spie•len“ und bei morphologischer Trennung in „ab-spielen“ getrennt. In diesem Fall enthält der Wortstamm ein Präfix. Das Wort „Autos“ hat den Suffix „s“ und „Autobahn“ besteht aus zwei vollständigen Wortstämmen („Auto-bahn“). Es gibt noch weitere Unterteilungen in der Gruppe der Affixe, die aber an dieser Stelle in der Erklärung zu umfänglich sind. Die Trennung in Morpheme erlaubt eine bessere Weiterverarbeitung im Hinblick auf die Interpretation, auch für die Bedeutung des Textes. (vgl. [CrIm06], S. 13; [Mug86], S. 29ff; [CFL13], S. 364ff; [SmHi94], S. 125)

Zudem lassen sich Texte durch **N-Gramme**, wobei N für eine Zahl größer Null steht, in Textfragmente einteilen und dadurch besser analysieren. Ein Trigramm beispielsweise besteht aus drei Zeichen, sodass die Begrüßung „Guten Morgen“ in „Gut-enM-org-en“ aufgeteilt wird. Alternativ können statt Zeichen auch Wörter als Einheit benutzt werden, sodass „Guten Morgen“ als einzelnes Bigramm gesehen werden kann. N-Gramme werden genutzt, um Sprachmodelle darzustellen, wobei den Einträgen Wahrscheinlichkeiten zugeordnet werden, die dann beispielsweise für die Wortprognose oder Korrektur verwendet werden können. (vgl. [CFL13], S. 79ff; [MaSc05], S. 192ff)

Die **Bestimmung von Wortgrenzen** (word breaking) ist ein Teilbereich der Textsegmentierung und beschäftigt sich mit der Auftrennung von Wörtern. Ein Beispiel dafür ist die Eingabe „Bitteallemalächeln!“, die in einzelne Wörter unterteilt wird. Dies wird auch als Tokenisierung bezeichnet. Während „Bitte“ erkennbar ist, muss zwischen „allemal“ und „alle mal“ eine Entscheidung im Kontext getroffen werden. An dieser Stelle sollte auch entschieden werden, ob zusammenhängende Wörter wie „E-Mail“ oder „gibt’s“ getrennt werden. (vgl. [CrIm06], S. 51; [RAN07], S. 501) Dazu gibt es zwei Herangehensweisen. Bei der Ersten werden Greedy-Algorithmen verwendet, um die Wörter im Text durch einen Abgleich mit dem Wörterbuch zu erfassen und an diesen Stellen eine Segmentierung vorzunehmen. Beispiele für Algorithmen sind MK10 und Sequitur. Die Zweite benutzt probabilistische Modelle, welche den Wörtern im Wörterbuch Wahrscheinlichkeiten über das Vorkommen zuordnen. Anhand der zuletzt identifizierten Wörter und den

darauffolgenden Zeichen wird dann die Wahrscheinlichkeit für das nächste anvisierte Wort berechnet. Beispiele hierfür sind die minimale Beschreibungslänge und das hierarchische Bayes. Beide Ansätze lassen sich auch kombinieren. (vgl. [CFL13], S. 373ff)

Im Zusammenspiel der Worttrennung ist eine **Rechtschreibprüfung**, die sich auf die orthografische Analyse beschränkt, sinnvoll, um Tippfehler bei der Eingabe zu korrigieren und, um die Wortgrenzen besser zu bestimmen. Hierzu werden die generierten Tokens mit Einträgen aus dem Wörterbuch verglichen. Um eine gewisse Unschärfe beim Vergleich zu erlangen, gibt es Algorithmen für die **Textähnlichkeit**. Der Soundex-Algorithmus, entwickelt für die englische Sprache, klassifiziert Wörter aufgrund ihres Klangs, sodass ähnlich klingende Wörter als identisch angenommen werden. Der Jaccard-Ähnlichkeits-Algorithmus vergleicht Texte oder Wörter, indem N-Gramme gebildet und der Koeffizient aus der Schnittmenge durch die Vereinigungsmenge berechnet wird. Ein weiterer Algorithmus ist die Levenshtein-Distanz (auch Editierdistanz). Hierbei wird die minimale Anzahl der Operationen aus Hinzufügen, Bearbeiten und Entfernen von Buchstaben für die Transformation der beiden verglichenen Worte berechnet. (vgl. [SmHi94], S. 132ff; [Kuk92], S.377ff)

Mit **part-of-speech Tagging** (POST) ist die Zuordnung der Satzzeichen und Wörter eines Textes in Kategorien (die den Wortarten ähnlich sind), genannt Tags<sup>14</sup>, gemeint. Dazu werden für jedes Wort, durch Suche im Wörterbuch, die Zuordnungen zu dem Wort erfasst. Dies können mehrere Zuordnungen sein, da die Wortart durch den Kontext definiert wird. Beispielsweise ist die Unterscheidung zwischen Adjektiv und Adverb für das Wort „laut“ im Satz „Die Nachbarn sind laut“ erst durch den Bezug zum Substantiv „Die Nachbarn“ möglich. Es gibt verschiedene Ansätze. Bei einem vorhandenen und entsprechend großen Trainingsdatensatz gibt es den Markov Model Tagger mit dem Viterbi-Algorithmus. Alternativ dazu existiert der Hidden Markov Model Tagger, welcher ohne Trainingsdaten aufgrund von Wahrscheinlichkeiten in der Wortfolge eine Kategorisierung vornimmt. Ein anderer Ansatz bestimmt mithilfe von Entscheidungsbäumen Attribute im Kontext und kategorisiert die Wortart entsprechend. Die Entscheidungsbäume werden durch Techniken des maschinellen Lernens, die anhand von Datensätzen wie der Penn Treebank trainiert werden, bestimmt. (vgl. [MaSc05], S. 341ff; [CFL13], S. 192f)

Die **Entfernung von Stoppwörtern**, Wörter, die keine inhaltliche Bedeutung haben, geschieht durch den Ausschluss von entsprechenden Wörtern auf Basis von Listen. Damit reduziert sich ein Satz auf aussagekräftige Kernbegriffe, welche die weitere Verarbeitung vereinfachen können und daher in Betracht gezogen werden sollten (vgl. [SiRi03], S. 1665; [RAN07], S. 501). Davon abzugrenzen ist die Methode zum Indexieren von Dokumenten, bei der alle Wörter, die eine definierte Häufigkeit überschreiten, auch solche mit inhaltlichem Wert, entfernt werden (vgl. [Fox89], S. 19f).

---

<sup>14</sup> Ein Auszug an POS Tags: JJ = Adjektiv, NN = Nomen, RB = Adverb, VB = Verb

**Normalisierung** bedeutet das Zurückführen von morphologischen oder orthografischen Varianten eines Wortes auf eine kanonische Form. Dabei wird in Lemmatisierung und Stemming unterschieden. Bei der **Lemmatisierung** wird die Grundform eines Wortes mittels Wörterbuch gebildet. Bei Nomen wird das Wort beispielsweise auf den Singular zurückgeführt, bei Verben auf den Infinitiv. Das Wörterbuch als Grundlage ist jedoch problematisch, da die Suche zeitaufwendig ist, nicht alle Wörter im Wörterbuch hinterlegt sind und Mehrdeutigkeiten entstehen können. (vgl. [GlRö05], S. 376) **Stemming** meint das regelbasierte Abschneiden von Suffixen eines Wortes. Einer der bekanntesten Algorithmen ist der Porter-Stemmer-Algorithmus. Hierbei werden iterativ, aufgrund einer Menge an Verkürzungsregeln, Suffixe entfernt, bis eine Minimalanzahl an Silben erreicht wird. Auch beim Stemming können Mehrdeutigkeiten entstehen, als auch Wortstämme, die nicht existieren. (vgl. [Hul96], S.70f)

Die Erweiterung um **Thesauren** meint Wörter durch Verbindungen zu Synonymen und Fachtermini zu erweitern, um sprachliche Varianten bei der Textanalyse abzudecken. Hierbei gilt es die semantische Ähnlichkeit zu berechnen, um unbekannte Wörter, die in Relation zu bekannten stehen, zu verstehen. (vgl. [JiCr94], S. 146f; [MaSc05], S. 294f)

### 3.1.2 Textanalyse

Die Textanalyse umfasst die semantische Analyse und Interpretation der Eingaben. Erste Ansätze beschränken sich auf die Erkennung von definierten Stichworten. Durch Platzhalter oder mit regulären Ausdrücken können entsprechende Ansätze aufgewertet werden.

Die **Eigennamenerkennung** (named entity recognition, NER) meint die Erfassung von Entitäten wie Namen, Organisationen, Orten oder Zahlen aus einem Text und die Klassifizierung als diese (vgl. [CrIm06], S. 62). Zudem ist auch das vergleichende Prüfen zu bereits erkannten Entitäten im Text oder Dialogverlauf zur Erkennungsverbesserung sinnvoll (vgl. [FGM05], S. 363). Für NER gibt es verschiedene Ansätze:

- Rein listenbasierte Systeme führen einen Abgleich mit einer definierten Liste aus Begriffen mit den Wörtern im Text durch. Diese Listen, auch Gazetteers genannt, werden nach den zu klassifizierenden Kategorien benannt. Beispielsweise sind in einer Ortsliste alle Ortsnamen abgebildet. Jedoch kann man nicht davon ausgehen, dass diese Listen vollständig sind, und eine regelmäßige Datenpflege ist nötig, um neue Begriffe aufzunehmen. Zudem kann derselbe Begriff in verschiedenen Listen mit unterschiedlicher Bedeutung auftauchen, wodurch eine Disambiguierung ohne weitere Informationen problematisch ist.
- Regelbasierte Systeme nutzen bekanntes Wissen von internen und externen Evidenzen der Wörter. Internen Evidenzen sind solche, die unmittelbar mit dem Wort verbunden sind. Die Groß- und Kleinschreibung, Wortbestandteile oder Endungen, aber auch Einträge in Wörterbüchern können die Zugehörigkeit zu einer bestimmten Gruppe symbolisieren. Externe Evidenzen sind Informationen, die aus

dem Kontext des Wortes hervorgehen. Beispielsweise lässt sich eine Uhrzeit in der Phrase „ab ... Uhr“ durch die signalisierenden Wörter „ab“ und „Uhr“ extrahieren.

- Durch maschinelles Lernen lässt sich ebenfalls ein System für die Erkennung von Eigennamen entwickeln. Hierzu sind entsprechende Daten für das Training notwendig. Diese Daten bestimmen auch die Sprache, die im Gegensatz zu den vorherigen Ansätzen nicht explizit integriert wird. Jedoch sind diese Systeme für Details, die Erkennung von bestimmten Mustern, nur schwer optimierbar. Weiterhin gestaltet sich die Verwendung des Systems in einer anderen Domäne schwierig.

(vgl. [NaSe07], S. 3ff; [BSA+98], S. 152ff)

Die **Klassifizierung von Sprechakten** versucht die Intention der Aussage zu erfassen und einzuordnen. Dazu entwickelten Stolcke et al. einen umfassenden Ansatz, der neben lexikalischen und kollokativen<sup>15</sup> Hinweisen auch die kontextuellen Informationen aus dem Dialogverlauf betrachtet (vgl. [SRC+00], S. 339ff). Der Dialogverlauf wird als Hidden Markov Model interpretiert und die daraus resultierenden Sprechakte als Zustand dargestellt. Diese Zustände werden unter anderem durch Entscheidungsbäume und neuronale Netzwerke eingeschränkt. Andere Ansätze wie von Grau et al. benutzen nur das naive Bayes und N-Gramme oder Wilks et al. N-Gramme und Stichwörter, wobei letztere annähernd gute Ergebnisse erzielten wie der umfängliche Ansatz von Stolcke et al. (vgl. [GSC+04], S. 495ff; [WHW05], S. 451ff). Die Convolutional Neural Networks und Recurrent Neural Networks, Weiterentwicklungen der neuronalen Netzwerke, ermöglichten zuletzt bessere Erkennungsraten gegenüber Ansätzen mit naive Bayes (vgl. [KGN16], S. 2013ff)

Mit der **Erkennung des zentralen Wortes** ist die Suche nach dem ausschlaggebenden Nomen in der Aussage gemeint (vgl. [ZhLi07], S. 631). Das Extrahieren des Hauptverbs zur Analyse der Handlung ist analog dazu. Das zentrale Wort soll im Idealfall das Thema der Aussage bestimmen oder eingrenzen. Lafferty et al. verglichen Conditional Random Fields (CRF) mit dem Hidden Markov Model und dem Maximum-entropy Markov Model und stellten fest, dass ihr Ansatz der CRF bessere Erkennungsraten als auch ein robusteres Verhalten bei einem fehlerhaften Model aufzeigt. CRF werden ebenfalls durch einen Datensatz, der mit POST annotiert wurde, trainiert. (vgl. [LMP01], S. 282ff)

---

<sup>15</sup> Kollokation meint das gehäufte benachbarte Auftreten von Wörtern (vgl. [Fir51]; S.194ff)

### 3.1.3 Textproduktion

Textproduktion meint das Erstellen einer Antwort in Bezug auf den Dialogverlauf. Bei abrufenden Systemen werden vordefinierte Textbausteine durch ein Regelwerk bereitgestellt und an den Benutzer übermittelt. Diese Textbausteine, als auch sprachliche Varianten davon, können regelbasiert an eine Aussage oder Intention gebunden werden. Die Artificial Intelligence Markup Language<sup>16</sup> (AIML) von Wallace beispielsweise verbindet in sogenannten Wissensseinheiten, Antworten mit Aussagen. Alternativ können Textbausteine in einem Datensatz gespeichert und entsprechend indexiert werden, sodass diese bei bestimmten Stichwörtern in einer Aussage abgerufen werden. Dies ähnelt Frage-Antwort Systemen, die Antworten auch durch Abruf von Informationen aus dem Internet bereitstellen (vgl. [HiGa01], S. 275ff).

Textgenerierende Systeme, welche die Sätze, wie von Reiter und Dale beschrieben, gänzlich produzieren, werden im Rahmen dieser Arbeit nicht weiter betrachtet (vgl. [ReDa00], S. 41ff).

### 3.1.4 Klassifikation

Die Klassifikation gehört zu den überwachten Verfahren des maschinellen Lernens. Hierbei werden Daten, die sich durch ein oder mehrere Merkmale auszeichnen, voneinander unabhängigen Klassen zugeordnet. Das überwachte Lernen ermöglicht die Ausprägung von Klassifizierern aufgrund der Merkmale in den Daten. Dazu werden bestehende Datensätze, die bereits einer Klasse zugeordnet sind, benötigt. (vgl. [Kos07], S. 3)

Die zu klassifizierenden Daten werden darin unterschieden, ob diese quantitativ oder qualitativ sind. Quantitative Daten sind diskret oder kontinuierlich. Qualitative Daten sind binär, nominal oder ordinal. Ferner ist relevant, ob die Klassifikation anhand eines oder mehrerer Merkmale ausgemacht wird und wie viele Einteilungsklassen es gibt. Der einfachste Fall besteht aus einer binären Entscheidung, also zwei Klassen.

Durch verschiedene Algorithmen und Kombinationen der Algorithmen sowie Abwandlungen dieser wird die Klassifizierungsmethode dem Datensatz angepasst. Ebenfalls ist die Art und Anzahl der Daten für die Wahl der Klassifizierungsmethode entscheidend.

Um sich in der Vielzahl an Auswahlmöglichkeiten für einen Algorithmus zurechtzufinden, wurden sogenannte Machine Learning Cheat Sheets entwickelt. Diese Spickzettel

---

<sup>16</sup> AIML 1.0 wurde im September 2001 vorgestellt, der letzte Stand der Spezifikation ist Version 1.0.1 von 2011 und kann hier eingesehen werden: <http://www.alicebot.org/TR/2011/> Daraufhin wurde Version 2 entwickelt, der letzte Stand ist Revision 1.0.2.22 vom März 2014: <https://docs.google.com/document/d/1wNT25hJRyupcG51aO89UcQEiG-HkXRXusukADpFnDs4/pub>.

geben eine Schritt für Schritt Anleitung vor, die anhand von Meta-Informationen des Datensatzes jene Algorithmen empfehlen, welche sich am ehesten für die Klassifizierung eignen. Beispielsweise bieten dies Andreas Müller<sup>17</sup> und Microsoft<sup>18</sup> an.

Auf der internationalen Konferenz zum Data Mining wurden im Jahr 2006 durch verschiedene Wahlverfahren die zehn für die Forschung relevantesten Algorithmen gewählt. (Unter diesen waren nicht nur Klassifikationsverfahren.) Als Gewinner ging der C4.5 Algorithmus für Entscheidungsbäume hervor. (vgl. [WKQ+07], S. 2ff)

---

<sup>17</sup> Der Spickzettel von Andreas Müller angepasst für scikit-learn ist zu finden auf: [http://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/](http://scikit-learn.org/stable/tutorial/machine_learning_map/).

<sup>18</sup> Der Spickzettel von Microsoft ist für das hauseigene Machine Learning Studio entwickelt, aber die gewählten Algorithmen und deren Vorteile gelten auch ohne die Verwendung dessen: <https://docs.microsoft.com/en-in/azure/machine-learning/machine-learning-algorithm-cheat-sheet>.

## 3.2 Frameworks

Chat-Bot Frameworks sind Programmiergerüste, die Basisfunktionalitäten für die Entwicklung von Chat-Bots bündeln und bereitstellen. Sie existieren in Form freier Projekte, die meist in einem wissenschaftlichen Rahmen angelegt sind, wie auch in kommerziellen Ausprägungen. Der folgende Abschnitt stellt eine Auswahl vor.

### 3.2.1 Facebook

Facebook bietet mit Wit.ai<sup>19</sup> ein Framework, das auch für den kommerziellen Gebrauch kostenlos ist. Facebook gibt über 50 unterstützte Sprachen an, darunter Chinesisch, Deutsch und Englisch. (vgl. [WIT-1-ol])

Der Messenger beziehungsweise die Kommunikationsfunktion von Facebook wurde im April 2016 für die Verwendung von Chat-Bots freigegeben. Seitdem besteht die Möglichkeit, auch über externe Dienste die Kommunikationsoberfläche<sup>20</sup> von Facebook zu benutzen. Die Oberfläche ermöglicht neben klassischen Textzeilen die Verwendung von strukturierten Vorlagen, die auch Bilder und vordefinierte Antworten beinhalten können. (vgl. [FB-ol])

Das Wit.ai Framework bündelt die Integration mit dem Facebook Messenger und den Komponenten eines Chat-Bots. Bei den Komponenten wird zwischen dem Verstehen, bei dem durch eine Textanalyse strukturierte Daten gewonnen werden, und der Konversation, bei der die nächste Aktion durch den Bot definiert wird, unterschieden. Zudem kann der aktuelle Zustand im Gesprächsverlauf zusammen mit gesammelten Informationen in einem Kontext gespeichert werden. (vgl. [WIT-2-ol])

Das Verstehen funktioniert durch Zuordnung zu zuvor angelegten Absichten. Beispielsweise sind Absichten die Suche nach einem Restaurant oder der Frage nach dem Wetter. Mittels Beispielsätzen für Absichten, die manuell zugeordnet werden müssen, wird der Bot, durch überwacht maschinelles Lernen, trainiert. (vgl. [WIT-2-ol])

Außerdem können, beispielsweise durch Eigennamenerkennung, verschiedene Entitäten wie Orte, Zeiten und Schlüsselwörter, aber auch ganze Satzteile extrahiert werden. Falls eine Entität nicht automatisch erkannt wird, kann diese manuell durch die Auswahl des oder der Wörter im Text ergänzt werden. Alle Eingaben, auch diese, die durch die Nutzer getätigt werden, werden in Wit.ai gespeichert und mit einer Absicht und Entitäten anno-

---

<sup>19</sup> Wit.ai wurde im Januar 2015 von Facebook übernommen.

<sup>20</sup> Auf der Webseite <https://messengerplatform.fb.com/> werden Neuigkeiten über Chat-Bots, die die Oberfläche von Messenger nutzen, kommuniziert.



tiert. Diese automatische Klassifizierung kann und sollte vom Entwickler überwacht werden, um Fehler bei der Klassifizierung zu korrigieren sowie Sprachvariationen und bislang unbekannte Entitäten zu erfassen. (vgl. [WIT-2-ol])

Bei der Frage nach dem Wetter und der gegebenen Entität für den Ort schließe sich als nächster Schritt an, eine Datenquelle nach dem Wetter an dem spezifizierten Ort zu fragen. Wit.ai bietet dazu Aktionen an, durch die ein Aufruf von benutzerspezifischem Code gestartet wird und der Kontext darüber mit externen Informationen erweitert wird. Daraufhin kann die Antwort mithilfe von Daten aus dem Kontext definiert werden. Antwortvarianten sind nicht möglich. Programmierkenntnisse werden nur für die Behandlung von Aktionen benötigt. Außerdem verfügt Wit.ai über eine Stimmungsanalyse. (vgl. [WIT-2-ol])

Für die Gestaltung der Antwort und dem Unterhaltungsverlauf bietet Wit.ai einen Dialogmanager an, welcher basierend auf einem Dialogfluss und der Beantwortung von Fragen in einem definierten Rahmen agiert und bei Bedarf Rückfragen stellt. (vgl. [WIT-2-ol])

### 3.2.2 Microsoft

Microsoft bietet ein ähnliches Funktionsspektrum, wie das von Wit.ai, gliedert es jedoch in verschiedene Dienste. Mit dem Bot Framework<sup>21</sup> und den Cognitive Services (frei übersetzt mit Kognitive Dienstleistungen, ehemals Project Oxford) unterhält Microsoft eine Vielzahl an Produkten, die sich für die Entwicklung von Chat-Bots verknüpfen lassen. Ab einer bestimmten Benutzungsfrequenz erhebt Microsoft Gebühren für die Verwendung der Cognitive Services. Die Sprachvielfalt variiert je nach Service von ausschließlich US-amerikanischem Englisch im Bing Spell Check (für die Korrektur der Texteingabe) bis hin zu neun Sprachen für die Language Understanding Intelligent Services (LUIS) (frei übers. mit intelligenten Dienstleistungen für das Sprachverständnis), darunter Chinesisch, Deutsch und Englisch. (vgl. [MS-1-ol]; [MS-2-ol]; [MS-3-ol]; [MS-4-ol])

Mit Skype hat Microsoft eine Kommunikationsplattform, die seit März 2016 auch für Chat-Bots freigegeben wurde (vgl. [SKYPE-1-ol]). Über das Anniversary Update für Windows wurde im Herbst 2016 Skype automatisch für alle Windows 10 Benutzer aktualisiert, welches die Bot-Funktionalität einschließt (vgl. [SKYPE-2-ol]). Ähnlich wie im Facebook Messenger unterstützen Skype-Bots strukturierte Vorlagen mit Bildern und vordefinierten Antwortmöglichkeiten. Darüber hinaus sind mit dem Skype-Bot Konversationen mit mehr als einem Teilnehmer möglich. (vgl. [BFW-1-ol])

---

<sup>21</sup> FUSE Labs (eine Forschungsabteilung von Microsoft) veröffentlichte auf <https://dev.botframework.com/> im März 2016 eine erste Version des Bot Frameworks.

Das Bot Framework umfasst drei wesentliche Funktionen, welche sich, durch die zwingende Umsetzung in Programmcode, mit hohen Freiheitsgraden auszeichnen. Erstens werden über den Bot Connector Anbindungen zu verschiedenen Kommunikationsdiensten, wie Skype, SMS, Facebook, bereitgestellt und über ein zentrales Register, das Bot Directory, veröffentlicht. Zweitens agiert es als Dialogmanager für zustands- oder rahmenbasiertes Dialogmanagement sowie zur Gestaltung des Dialogflusses. Ebenfalls kann das Bot Builder SDK formatierte Nachrichten mit Bildern generieren. Drittens können darüber weitere Komponenten für die linguistische Verarbeitung durch Schnittstellen zu den Cognitive Services eingebunden werden. Gleichfalls sind eigene Algorithmen und Methoden anwendbar. Die Antworten werden im Programmcode definiert, wodurch abwechselnde Antworten nicht ausgeschlossen sind. Mittels Microsoft Azure können Bots in der Cloud entwickelt, gewartet und betrieben werden. Ferner erlaubt die Oberfläche des Microsoft Bot Framework Channel Emulators den testweisen Betrieb, ohne den Chat-Bot in öffentlichen Verzeichnissen zu listen. (vgl. [BFW-2-ol])

Die Cognitive Services umfassen die eingangs erwähnte Textkorrektur und LUIS, sowie Linguistic Analysis (linguistische Analyse), Text Analytics (Textanalyse), Translator (Übersetzer) und Web Language Model (Web-Sprachmodell) im Bereich der Sprache. Darüber hinaus gibt es weitere Bereiche der Cognitive Services für auditive und visuelle Verarbeitung, als auch die für Knowledge (Wissen), welche für diese Arbeit aber nicht weiter relevant sind.

Gegenüber der Textkorrektur und dem Übersetzer lässt sich der Funktionsumfang aus der Begrifflichkeit für LUIS, der linguistischen und textuellen Analyse, nicht ohne Weiteres schließen.

LUIS<sup>22</sup> ähnelt der Funktion des Verstehens von Wit.ai. Auch hier werden Texteingaben in zuvor definierte Absichten kategorisiert, aus denen auch erkannte Entitäten extrahiert werden können. Kann eine Eingabe nicht zugeordnet werden, so greift eine vordefinierte Standardabsicht. Ebenfalls gibt es eine Reihe von vordefinierten Entitäten, wie Alter oder Datum. Orte, Produkte und andere Eigennamen werden über Einträge aus einer Enzyklopädie bestimmt und in über 100 Unterkategorien eingeteilt. Diese Funktion beschränkt sich derzeit jedoch auf den US-amerikanischen Raum. Zusätzlich können Entitäten über reguläre Ausdrücke und Listen aus Phrasen und Synonymen erfasst werden. (vgl. [MS-4-ol]; [MS-5-ol]; [MS-6-ol]; [MS-7-ol]; [MS-8-ol])

Die linguistische Analyse umfasst die Erkennung von Sätzen in einem gegebenen Text und die darauffolgende Aufteilung von Sätzen in Wörter (Tokenisierung). Ferner werden POST und eine Phrasenanalyse angeboten. (vgl. [MS-9-ol])

---

<sup>22</sup> Die Oberfläche auf Luis.ai bietet interaktives maschinelles Lernen durch Zuordnen von Absichten. Das Sprachverständnis wird durch existierende Sprachmodelle aus Bing und Cortana ermöglicht, wobei spezielle Modelle mit dem Web Language Model auch möglich sind.

Mit der Textanalyse können Textsprache, Schlüsselwörter und Stimmung aus einem gegebenen Satz identifiziert werden. (vgl. [MS-10-ol])

### 3.2.3 Google

Google bietet mit Api.ai<sup>23</sup> ein Framework, das die Integration mit den Plattformen und die Komponenten des Chat-Bots (inklusive der linguistischen Verarbeitung) auf einer Oberfläche vereint. Kenntnisse einer Programmiersprache sind nicht notwendig<sup>24</sup>, da Aktionen und Antworten in der Webanwendung definiert werden. Es werden 15 Sprachen unterstützt, darunter Chinesisch, Deutsch und Englisch. In derzeit 37 domänenspezifischen Paketen, die zu einem Bereich wie Sport oder Wetter bereits schlüsselfertig für die Verwendung sind, reduziert sich die Unterstützung für verschiedene Sprachen jedoch auf maximal neun Sprachen, wobei in vielen Fällen nur Englisch unterstützt wird. Jedoch sind die domänenspezifischen Pakete (bis auf das für Small Talk) nicht kostenlos. Weitere kostenpflichtige Elemente gibt es nicht. (vgl. [AA-1-ol]; [AA-2-ol]; [AA-3-ol])

Mit Allo, Duo, Google Messenger und Hangouts hat Google zwar verschiedene Dienste zur Kommunikation, jedoch ist keiner in Api.ai integriert. Prinzipiell ist dies aber möglich (siehe dazu Fußnote 24). Dafür sind Slack wie auch Skype und der Facebook Messenger ohne zusätzliche Implementierungsmaßnahmen einsatzbereit. (vgl. [AA-4-ol])

In Api.ai werden die Chat-Bots als Agents bezeichnet. Auch hier findet sich die Technologie des maschinellen Lernens für die Klassifizierung in zuvor definierte Absichten wieder. Die Klassifizierung der Absichten ist durch Annotation der Entitäten als auch durch explizite Vorlagen möglich. Das maschinelle Lernen lässt sich für letztere und bei umfangreicher Verwendung von zusammenhängenden Entitäten auf einen Hybrid Modus umschalten, der bessere Ergebnisse insbesondere bei weniger Beispielen und einer Vielzahl an Anpassungen der Entitäten verspricht. Absichten können durch Angabe einer Priorität in ein Verhältnis der Vorrangigkeit eingeordnet werden; wenn Eingaben in keine der Absichten klassifiziert werden können, gibt es dafür eigens Alternativen. Über eine globale Einstellung lässt sich definieren, ab wann die Unsicherheit der Klassifizierung für eine Einordnung zu hoch ist und folglich eine Alternative greift. (vgl. [AA-5-ol]; [AA-6-ol])

Die Antworten werden über Parameter mit den erkannten Entitäten angereichert; auch Formulierungsvarianten sind möglich. Durch einen Webhook, also einen Aufruf einer Webanwendung durch Api.ai, können externe Dienste und Daten für die Generierung der

---

<sup>23</sup> Api.ai wurde im September 2016 von Google übernommen.

<sup>24</sup> Für die Verknüpfung von nicht integrierten Kommunikationsdiensten sind Programmierkenntnisse nötig, ebenso für ergänzende Algorithmen beispielsweise aus externen Services.

Antwort mit einbezogen werden. Neben Textantworten werden, sofern es die Kommunikationsdienste erlauben, Schnellantworten und strukturierte Vorlagen mit Bildern unterstützt. (vgl. [AA-7-ol])

Vordefinierte Entitäten erlauben unspezifische Angaben wie eine Zahl oder ein Datum, als auch spezifische, wie eine nicht negative Zahl, einen Zeitraum oder eine Stadt. Die erfassten Entitäten werden, wenn möglich, genormt oder standardisiert. Darüber hinaus können Listen aus Synonymen und kombinierte Entitäten aus zwei oder mehr Entitäten angelegt werden. (vgl. [AA-8-ol])

Der aktuelle Zustand im Gesprächsverlauf und die gesammelten Informationen werden durch den Ein- und Ausgabekontext gespeichert. Eine Absicht kann einen Ausgabekontext erzeugen, sodass dieser im Verlauf der Sitzung gilt. Der Eingabekontext erwartet, dass ein bestimmter Kontext für eine Absicht gilt, damit diese durch Aussagen klassifiziert wird. Beispielsweise wird gesagt, dass das Licht im Obergeschoss ausgeschaltet werden soll. Als Ausgabe Kontext wird nun „Obergeschoss“ definiert. Bei einem weiteren Befehl zum Herunterfahren der Rollläden unter dem Eingabekontext „Obergeschoss“ kann gezielt die Absicht für das Obergeschoss ausgeführt werden. (vgl. [AA-9-ol])

Über Ereignisse wie „Willkommen“, die durch die Kommunikationsdienste ausgelöst werden, können Antworten auch ohne Eingaben durch den Benutzer generiert werden. Absichten erlauben zudem die Spezifizierung von Aktionen, wodurch verschiedene Absichten gleiche Aktionen definieren können, etwa, wenn verschiedene Kontexte gesetzt sind. (vgl. [AA-10-ol])

Beim Dialogmanagement steht die Auswahl zwischen rahmenbasierten und einem variablen Dialogmanager. Letzterer wird über die Kontexte gesteuert. Dies erlaubt zudem, den Dialogfluss zu steuern. (vgl. [AA-11-ol])

Zusätzlich bietet Google mit der Cloud Natural Language API Werkzeuge für die Syntaxanalyse mit Tokenisierung, POST, als auch die Erkennung von Entitäten und einer Stimmungsanalyse. Hierfür fallen ab 5.000 Serveranfragen monatliche Kosten an. (vgl. [GOOGLE-1-ol])

### 3.2.4 IBM

IBM bietet mit Watson Conversation<sup>25</sup> ein natürlichsprachliches Chat-Bot Framework, das auf der Cloud-Plattform IBM Bluemix als ein Service bereitgestellt wird. Es werden zehn Sprachen, darunter Chinesisch, Deutsch und Englisch unterstützt. Pro Serveranfrage wird eine Gebühr erhoben. (vgl. [IBM-1-ol]; [IBM-2-ol])

---

<sup>25</sup> Die erste offizielle Version von Watson Conversation wurde im Juli 2016 vorgestellt und ist auf [ibmwatsonconversation.com/](http://ibmwatsonconversation.com/) verfügbar. Dazu kaufte IBM mehrere Start-ups, unter anderem Cognea und Teile von Fluid.

Das Einbetten von Watson Conversation in eine Anwendung ist zwingend erforderlich, da keine Schnittstellen zu Kommunikationsdiensten existieren und auch externe Dienste, und damit Daten, ohne dies nicht in den Kommunikationsfluss integriert werden können. (vgl. [IBM-1-ol])

In Watson Conversation ist ein rahmenbasiertes Dialogmanagement ebenso wie ein variabler Dialogfluss durch Sprünge möglich. Durch Bedingungen können fehlende Informationen nachgefragt und zusätzliche Daten sowie extrahierte Informationen im Kontext gespeichert werden. (vgl. [IBM-4-ol])

Anders als die bislang vorgestellten Frameworks sind die vordefinierten Entitäten auf Zahlen und Uhrzeiten beschränkt. Orte oder andere Eigennamen müssen manuell hinzugefügt werden. Aufzählungen und Synonyme werden unterstützt. (vgl. [IBM-3-ol])

Antworten auf erkannte Absichten erlauben Varianten in der Formulierung (vgl. [IBM-4-ol]), jedoch werden Schnellantworten und strukturierte Vorlagen mit Bildern, nicht zuletzt durch die fehlende Integration zu Kommunikationsdiensten, nicht unterstützt.

Neben Watson Conversation unterhält IBM den AlchemyLanguage<sup>26</sup> Service. Damit soll unter anderem die Extraktion von Schlüsselwörtern und Entitäten möglich sein. (vgl. [IBM-5-ol])

### 3.2.5 Amazon

Mit Lex<sup>27</sup> hat Amazon, nach dem Erscheinen des persönlichen virtuellen Assistenten Alexa und des Amazon Echo, ein Chat-Bot Framework innerhalb der Amazon Web Services (AWS). Auch bei Lex werden Eingaben in zuvor definierte Absichten klassifiziert und die Entitäten daraus analysiert, um mit diesen Informationen gezielte Aktivitäten auszuführen. Derzeit unterstützt Lex nur US-amerikanisches Englisch, obwohl die Entitäten, für die auf die Datenbasis des Alexa Skills Kit zurückgegriffen wird, teils britisches Englisch und Deutsch unterstützen. Bei den Entitäten wird zwischen der Gruppe aus Zahlen, Datum und Uhrzeiten sowie der Gruppe von Kategorien aus Listen unterschieden. Die Listen umfassen Sprachen, Orte, Geschäfte aber auch Ontologien zu Geschäftstypen (Autoverleih, Bekleidung) oder Raum (Badezimmer, Küche). Die Absichten aus dem Alexa Skills Kit, die mit einer systematisch definierten Signatur durch eine Aktion (Hinzufügen, Suche) und einem oder mehreren Entitätsgruppen (Wettervorhersage, Wiedergabeliste) eine Absicht beschreiben, werden hingegen nicht verwendet. (vgl. [LEX-1-ol], S. 5ff; [LEX-2-ol]; [LEX-3-ol])

---

<sup>26</sup> Ehemals ein Teil der AlchemyAPI.

<sup>27</sup> Die erste Version von Amazon Lex wurde im November 2016 vorgestellt. Auf der Webseite <https://aws.amazon.com/de/lex/> wird Lex noch als Vorschauversion bezeichnet.

Der Dialogmanager ist rahmenbasiert und kann bei fehlenden Informationen nachfragen. Die Antworten werden auf der Lex Oberfläche definiert. Strukturierte Vorlagen mit Bildern und vordefinierten Antwortmöglichkeiten sind nur innerhalb einer Absicht für das gezielte Abfragen einer fehlenden Information möglich. Externe Dienste werden über den Datenverarbeitungsservice AWS Lambda unterstützt. (vgl. [LEX-1-ol], S. 5ff)

Für die Integration mit anderen Kommunikationsdiensten wird auf das AWS Mobile SDK verwiesen. Hierfür muss dann eine mobile Anwendung entwickelt werden. Alternativ wird der Facebook Messenger unterstützt. (vgl. [LEX-1-ol], S. 61ff)

Amazon Lex erhebt pro Serveranfrage eine Gebühr. (vgl. [LEX-4-ol])

### 3.2.6 Recast.ai

Recast.ai<sup>28</sup> nutzt ebenfalls maschinelles Lernen für die Klassifizierung von Eingaben in Absichten und kann Entitäten erkennen. Der Service ist für öffentliche Chat-Bots mit bis zu 25.000 Serveranfragen kostenlos. Neben Englisch wird Französisch in der Spracheingabe akzeptiert. In den Einstellungen wird zwar eine Standardsprache definiert, jedoch kann der Chat-Bot vollkommen zweisprachig eingerichtet werden.<sup>29</sup> Ebenso lässt sich in Analogie zu API.ai ein globales Kriterium für die Genauigkeit festlegen. Recast.ai bietet Programmcode für die Integration zu Slack, Facebook Messenger sowie dem Bot Framework, welches seinerseits weitere Kommunikationsdienste unterstützt. (vgl. [RA-1-ol]; [RA-2-ol])

Die Absichten von öffentlichen Chat-Bots werden mit der Community geteilt, sodass Absichten, wiederverwendet werden können, die auch die gelernten Formulierungen beinhalten. Die Liste der vordefinierten Entitäten ist umfangreich und enthält unter anderem Adressen, Berufe, Farben, Organisationen, Persönlichkeiten und Zeiten. Die Entitäten werden, wenn möglich, genormt oder standardisiert. Als goldene Entitäten werden jene bezeichnet, die bereits trainiert sind. Andere vor- und benutzerdefinierte lassen sich in einer Gazette durch eine Liste an Synonymen (inklusive Normierung) beschreiben. Jede Gazette bestimmt zudem mittels Kriterium über die Genauigkeit im Textvergleich. Die Analyse von Recast.ai umfasst zusätzlich die Klassifikation von Sprechakt, Satztyp und Stimmung. Außerdem werden in den Entitäten Pronomen erkannt.<sup>30</sup> (vgl. [RA-2-ol])

---

<sup>28</sup> Recast.ai wurde im September 2015 gegründet.

<sup>29</sup> Hinter Recast.ai steckt ein französisches Start-up-Unternehmen, weshalb diese Zweisprachigkeit naheliegend ist. Zudem ähnelt Recast.ai in der Gestaltung und Aufbau der Oberfläche von Wit.ai, welches von drei französischen Studenten gegründet wurde.

<sup>30</sup> Recast.ai ist nicht Teil einer NLP-Infrastruktur, die beispielsweise POST in einer separaten API bereitstellt. Aus diesem Grund ist davon auszugehen, dass Recast.ai diese Information mit in die Auswertung aufgenommen hat.

Das Dialogmanagement wird in Recast.ai durch einen Dialogfluss bestimmt. In einer Verkettung von Absichten werden Informationen erhoben, bis ein definiertes Ziel erreicht ist. Formulierungsvarianten und Rückfragen zu fehlenden Informationen, entsprechend dem rahmenbasierten Dialogmanagement, werden unterstützt. (vgl. [RA-2-ol])

### 3.2.7 ChatScript

ChatScript (maßgeblich von Bruce Wilcox entwickelt) ist im Gegensatz zu den vorgestellten Web-Services eine Open Source Skriptsprache und Desktopanwendung, mit der Wilcox viermal den Loebner Preis<sup>31</sup> gewann, zuletzt 2015. Aus Sicht von Wilcox, der aus der Videospielbranche stammt, braucht ChatScript weniger Regeln (welche das Sprachverständnis beschreiben), und damit Rechenzeit, gegenüber Systemen von Apple oder Google (vgl. [CS-7-ol]). ChatScript unterstützt Englisch und kann prinzipiell um jede Sprache erweitert werden. Für Deutsch, Französisch und Spanisch ist der part-of-speech Tagger TreeTagger<sup>32</sup> von Helmut Schmid integriert. Für diese und weitere Sprachen fehlen jedoch Sprachdefinitionen, beispielsweise für Synonyme und die Erkennung von Emotionen. (vgl. [CS-1-ol]; [CS-2-ol]; [CS-3-ol])

Basierend auf Regelsätzen, die in Textdateien angelegt werden, wird das Sprachverständnis beschrieben. Die Syntax besteht aus einem Typ, einem optionalen Label, dem Auslöser der Regel, der Antwort durch den Chat-Bot und einem optionalen Kommentar.

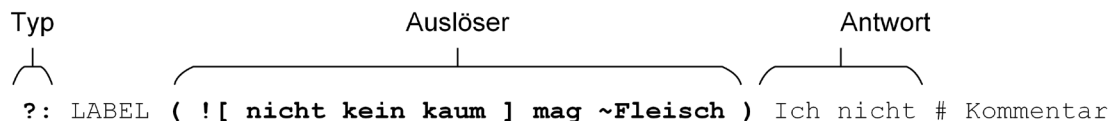


Abbildung 3.1: Beispiel einer ChatScript Regel

Als Typ kann zwischen Aussagen und Fragen unterschieden werden, als auch eine beginnende Aussage durch den Chat-Bot. Ein Ausdruck innerhalb der geschwungenen Klammern erzeugt, wenn dieser im Kontext der Eingabe wahr ist, eine Ausgabe. Abbildung 3.1 zeigt ein Beispiel, wie auf eine Frage des Benutzers, ob dieser Fleisch mag, geantwortet wird. Innerhalb von eckigen Klammern können alternative oder ähnliche Wörter gesammelt werden. Dies kann auch separat gespeichert und wiederverwendet werden, wie am Beispiel von „Fleisch“ zu sehen ist. Ein mit dem Ausrufezeichen vorangestellter Ausdruck schließt Wörter aus. Der Funktionsumfang der ChatScript Syntax erlaubt noch deutlich mehr als im Beispiel gezeigt; auch die Einbeziehung der part-of-speech Informationen. (vgl. [CS-4-ol]; [CS-5-ol])

<sup>31</sup> Der Preis wird jährlich vergeben und ehrt Programmierer, die einen besonders menschenähnlichen Chat-Bot entwickelt haben. Das Prüfformat ist ein Turing-Test.

<sup>32</sup> Mehr dazu auf <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>.

Disambiguierung findet durch die Definition von Themengebieten und Gliederung der Regeln in diese statt. Neben textbasierten Antworten ist es möglich, Funktionen und Makros zu schreiben, die auch externen Code ausführen können. Ebenfalls sind Antwortvarianten und das Speichern von Zuständen für die Definition eines Kontextes möglich. Weitere Funktionen umfassen die Lemmatisierung und eine Rechtschreibkorrektur. Ebenso werden Variablen benutzt, um das rahmenbasierte Dialogmanagement zu ermöglichen. (vgl. [CS-5-ol]; [CS-6-ol])

Eine Integration zu Kommunikationsdiensten ist nicht Bestandteil von ChatScript.

### 3.2.8 Pandorabots

Pandorabots<sup>33</sup> ist eine Oberfläche für die Erstellung von Chat-Bots basierend auf der Artificial Intelligence Markup Language (AIML). Es werden 136 Sprachen unterstützt und für Entwickler gibt es eine kostenlose Plattform; darüber hinaus werden Gebühren für die Bereitstellung als Web-Service erhoben. Pandorabots bietet keine Integration zu Kommunikationsdiensten, jedoch ist dies über die API und SDK möglich.

Der Aufbau einer Wissensseinheit von AIML besteht aus dem Kategorie-Tag, welcher einen Ausdruck-Tag und einen Vorlagen-Tag für die Antwort beinhaltet. Stimmt die Eingabe mit dem des Ausdruck-Tags überein, wird der Inhalt des Vorlagen-Tags ausgegeben. Wissensseinheiten können miteinander verbunden und Ausdrücke durch Bedingungen und Platzhalter aufgeweicht und eingeschränkt werden. (vgl. [AIML-ol])

```
<topic name = "Nahrung">
  <category>
    <pattern>Magst du Fleisch?</pattern>
    <that>Lass uns übers Essen sprechen</that>
    <template>Ich mag kein Fleisch.</template>
  </category>
</topic>
```

*Abbildung 3.2: Beispiel einer AIML Regel*

Abbildung 3.2 definiert eine AIML Regel, die bei der Frage „Magst du Fleisch?“ die Antwort „Ich mag kein Fleisch“ erzeugt. Dabei definiert der Kontext-Tag „that“, dass im Gesprächsverlauf schon einmal „Lass uns übers Essen sprechen“ vorkam. Weiterhin kann das Thema, wie im Beispiel angedeutet, eingeschränkt werden. Variablen und Sprachvarianten sind für die Ausgabe möglich.

---

<sup>33</sup> Seit Mai 2011 gibt es die Community Plattform auf <https://playground.pandorabots.com/>.



AIML erlaubt die Ausführung von JavaScript und wird für viele Chat-Bots benutzt, darunter ELIZA<sup>34</sup>, A.L.I.C.E.<sup>35</sup> und Mitsuku<sup>36</sup>.

### 3.2.9 RiveScript

Bei RiveScript<sup>37</sup> ist genau wie ChatScript eine Open Source Skriptsprache mit entsprechenden Interpretern für verschiedene Programmiersprachen. Der Entwickler von RiveScript beschreibt den Unterschied, gegenüber ChatScript und AIML, durch eine einfachere Syntax. Der Chat-Bot wird in einer Textdatei definiert und ist auf keine Sprache beschränkt, wobei die Beispiele allesamt für Englisch sind. Schnittstellen zu Kommunikationsdiensten sind nicht integriert. (vgl. [RS-1-ol]; [RS-2-ol]; [RS-3-ol])

Abbildung 3.3 zeigt das Beispiel „Ich mag Fleisch“ in der RiveScript Syntax. Der Auslöser der Regel wird mit einem Pluszeichen vorangestellt. Hier sind Platzhalter, optionale Ausdrücke und Alternativen in der Wortwahl, also auch eine Gewichtung des gesamten Auslösers, möglich. Nicht möglich ist das Ausschließen von Wörtern.

```
> topic global
    + Ich mag @Fleisch
    - Ich nicht! {weight=20}
    - Nicht so gerne! {weight=80}
< topic
```

*Abbildung 3.3: Beispiel einer RiveScript Regel*

Gefolgt von dem Auslöser folgt die Antwort darauf. Hierbei werden ebenfalls Varianten, die auch gewichtet werden können, unterstützt. Regeln können in Themengebiete gruppiert werden und unterstützen Variablen, wodurch kontextuelle Informationen und Aktionen möglich sind. Zudem ist die Ausführung von externem Code innerhalb von RiveScript möglich. (vgl. [RS-1-ol])

### 3.2.10 Motion AI und andere

Bei Motion AI wird durch das Verknüpfen von Modulen ein definierter Dialogfluss erzeugt, der zudem visuell dargestellt wird. Die Module können Telefonnummern oder Uhrzeiten abfragen, Ja-Nein-Entscheidungen treffen oder externen Code ausführen. Basierend auf den Antworten wird der Dialogfluss gelenkt. Die Textanalyse basiert aber

---

<sup>34</sup> ELIZA ist einer der ersten Chat-Bots, die von Joseph Weizenbaum 1964 entwickelt wurden.

<sup>35</sup> A.L.I.C.E. gewann den Loebner Preis dreimal und wurde anfangs von dem AIML Autor Richard Wallace entwickelt.

<sup>36</sup> Mitsuku, entwickelt von Steve Worswick, gewann zwei Mal den Loebner Preis, zuletzt 2016.

<sup>37</sup> Die Version 2.0, entwickelt von Noah Petherbridge, wurde im Dezember 2015 veröffentlicht.

lediglich auf der Erkennung von vordefinierten Ausdrücken. Eine Integration mit Kommunikationsdiensten wie dem Facebook Messenger ist möglich. Motion AI ist für 2 Bots und 1.000 Nachrichten kostenlos, darüber hinaus kostet der Dienst je nach Bedarf. (vgl. [MA-1-ol]; [MA-2-ol]; [MA-3-ol])

Bot Libre, Botsify, Chatfuel, Flow XO, oder auch Rebot.me sind weitere Services zum Erstellen von Chat-Bots, die jedoch nur Stichwörter oder Phrasen abgleichen, um darauf zu antworten oder Aktionen auszuführen. Maschinelles Lernen und die Erkennung von Entitäten wird bestenfalls rudimentär unterstützt.

### 3.2.11 Botkit

Botkit<sup>38</sup> ist ein Open Source Toolkit für die Node.js Plattform, welches Schnittstellen zu Kommunikationsdiensten, Datenbanken und Frameworks für die natürlichsprachliche Analyse und Verarbeitung anbietet. Darüber hinaus ist ein Dialogmanager integriert. Außerdem wird mit dem Botkit Studio eine Entwicklungsplattform in der Cloud angeboten. (vgl. [BK-1-ol]; [BK-2-ol])

### 3.2.12 NLP-Toolkits

Neben den Chat-Bot Frameworks sind eine Reihe von Toolkits mit Algorithmen und Methoden für die natürlichsprachliche Analyse und Verarbeitung verfügbar. Insbesondere der Dialogmanager und die Integration zu den Kommunikationsdiensten müsste, neben Auswahl und Verknüpfung der Chat-Bot Komponenten, jedoch zusätzlich entwickelt werden.

Das Natural Language Toolkit (NLTK) ist eine für die Programmiersprache Python entwickelte Open Source Sammlung an Algorithmen und Methoden. Die Sammlung umfasst unter anderem die Tokenisierung, Stemming, POST, NER, als auch entsprechende Wörterbücher wie WordNet. Neben Englisch werden je nach Algorithmus und Implementierung auch andere Sprachen unterstützt. (vgl. [NLTK-1-ol]; [NLTK-2-ol])

Das General Architecture for Text Engineering (GATE) enthält eine Open Source Sammlung an Algorithmen und Methoden in Java. Dazu gehören Tokenisierung, POST und NER, aber auch Wörterbücher wie WordNet, mit dem Ontologien abgebildet werden können. GATE ist für zwölf Sprachen verfügbar, darunter Chinesisch, Deutsch und Englisch, wobei die Qualität der Umsetzung jedoch schwankt. (vgl. [GATE-1-ol]; [GATE-2-ol])

---

<sup>38</sup> Auf der Quellcodeverwaltung GitHub wurde die erste Version Ende März 2016 veröffentlicht.

## 4 Lösungskonzept

Im Folgenden wird ein Lösungskonzept für die Erstellung eines Chat-Bots, der zu einer Problemeinschränkung von durch Nutzern gemeldeten Vorfällen im Rahmen des Facility-Managements dient, vorgestellt.

### 4.1 Anforderungen

Die Anforderungen des Chat-Bots umfassen die Bereitstellung von einem Kommunikationsdienst für die Informationsübermittlung zwischen dem Nutzer und Chat-Bot, der Auswahl und Implementierung der benötigten Komponenten eines Chat-Bots (beispielsweise durch ein Framework), sowie den domänenspezifischen Informationen für die Analyse und Kategorisierung der Meldungen zu Vorfällen, und um fehlende oder ungenaue Informationen durch Rückfragen einzuholen, um daraus ein Datenpaket mit strukturierten Angaben für die weitere Verarbeitung bereitzustellen.

Die Auswahl des Kommunikationsdienstes richtet sich nach den bereitgestellten Schnittstellen des Chat-Bot Frameworks. Ob der Facebook Messenger oder Skype für die Kommunikation verwendet wird, ist für die Funktionalität der Texterkennung unwesentlich, obwohl einige Kommunikationsdienste mehr Möglichkeiten bei der Darstellung von Nachrichten beispielsweise durch vordefinierbare Antworten bieten. Zudem ist eine spätere Implementierung einer Schnittstelle zu Kommunikationsdiensten, etwa bei geringer Kundenakzeptanz, in den meisten Fällen möglich. Ebenfalls ist die damit verbundene Unterstützung der Endgeräte sekundär, da Kommunikationsdienste in der Regel breit aufgestellt sind beziehungsweise eine Webanwendung anbieten.

Wichtiger ist die Auswahl des erforderlichen Frameworks für die Umsetzung. Aufgrund der Anzahl an verfügbaren Frameworks, den unterschiedlichen Ansätzen und Funktionen ist eine pauschale Einstufung in ein qualitativ gutes oder schlechtes Framework kaum möglich. Auch wenn einige Eckdaten, wie die unterstützten Sprachen, erfassbar sind, ist ein qualitativer Vergleich an dieser Stelle erst durch eine partielle Realisierung erreichbar. Für die weitere Betrachtung sollte daher die Implementierung eines Chat-Bots in verschiedenen Frameworks in Betracht gezogen werden.

Da einige Frameworks auf maschinellem Lernen basieren, sind Aussagen, die Nutzer potenziell treffen könnten, erforderlich, um den Chat-Bot zu trainieren. Darüber hinaus ist ein solcher Katalog an Vorfällen auch für Frameworks relevant, die nicht auf maschinellem Lernen basieren. In diesem Fall werden aus den Daten bestimmte Muster extrahiert, um daraus die Auslöser zu definieren, welche die Aussagen des Nutzers kategorisieren und dann weiterverarbeiten.

Neben den potenziellen Aussagen sind Datenmodelle erforderlich, welche die für die weitere Verarbeitung notwendigen Bestandteile einer Meldung, wie den Ort des Geschehens,

erfassen. Insbesondere ist eine Struktur zu entwickeln, welche den Vorfall derart beschreibt, dass dieser für eine maschinelle Auswertung lesbar ist. Im Falle von nicht vollständigen oder auch missverständlich ausgedrückten Daten muss der Chat-Bot dem Nutzer eine Rückfrage stellen, um die Information korrekt zu erfassen. Dazu sind entsprechend verständliche und nach den griceschen Maximen formulierte Textvorlagen für Rückfragen notwendig, aber auch ein Dialogfluss, welcher eine Struktur zur Datenabfrage anbietet. Der Dialogfluss bietet somit eine Richtschnur für den Dialogverlauf an.

Ferner sind Kategorien erforderlich, in welche die Vorfälle einsortiert werden. Diese Kategorien sollen dabei helfen, die Meldung und den damit verbundenen Vorfall zu beschreiben, die daraus resultierenden Tätigkeiten zu erfassen sowie die Dringlichkeit zu erkennen. Ebenfalls ist ein geeignetes Klassifikationsverfahren zu wählen, welches dies umsetzt.

Schließlich sind in einem vereinten Datenmodell die erfassten Informationen strukturiert zu speichern, damit diese dann über eine Schnittstelle für externe Systeme zugänglich gemacht werden können.

## **4.2 Lösungssegmente**

Die Lösungssegmente bieten ein Lösungskonzept für die aufgezeigten Problemstellungen mit den zuvor definierten Anforderungen.

### **4.2.1 Auswahl und Bewertung der Frameworks**

Wie in den Anforderungen beschrieben, ist die Auswahl eines geeigneten Frameworks für die Realisierung eines Chat-Bots ohne eine vorherige, zumindest teilweise Implementierung nicht möglich. Dennoch lassen sich die verschiedenen Frameworks anhand des Funktionsumfangs einordnen, mit dem eine gewisse Bewertung möglich wird. In Tabelle 4.1 werden die Frameworks anhand der Betriebskosten, Sprachenvielfalt und Methodik der Textanalyse verglichen.

Frameworks, die auf maschinellem Lernen basieren, unterscheiden sich wesentlich in Betriebskosten, den unterstützten Sprachen, Anzahl und Granularität der vordefinierten Entitäten und Gliederung der Oberfläche für die Erstellung eines Chat-Bots. Letztere umfasst die Zuordnung zu Absichten und Möglichkeiten zur Lenkung des Dialogablaufs, beispielsweise durch kontextuelle Einschränkungen. Ebenso verwaltet die Oberfläche die Schnittstellen zu Kommunikationsdiensten. Die Integration zu Letzteren ist bei Watson Conversation nicht vorhanden. Bei LUIS wird dies über das Bot Framework abgewickelt.

Das Bot Framework von Microsoft kann ohne LUIS, welches die Funktionalität für das maschinelle Lernen bereitstellt, betrieben werden. Doch gerade das maschinelle Lernen samt der Eigennamenerkennung und Gestaltung der Weboberfläche sorgt dafür, dass ein

lernfähiges und leicht zu bedienendes System entsteht. Nachdem die Absichten und Entitäten durch Experten definiert und erläutert wurden, können Formulierungen ohne zusätzliche Expertenhilfe im System eingefügt und klassifiziert werden. Außerdem werden alle Formulierungen die dem System neu sind gespeichert, sodass eine Validierung oder Korrektur erfolgen kann. Infolgedessen gewinnt das System an Kenntnis in der Erkennung der vom Benutzer mitgeteilten Informationen. Daher und da ChatScript und andere vorgestellte Frameworks ebenfalls Funktionen für die regelbasierte Kategorisierung bereitstellen, kann die Variante ohne LUIS vernachlässigt werden.

Obwohl Watson Conversation nur eine geringe Anzahl an vordefinierten Entitäten unterstützt, welche zudem lediglich grob klassifiziert werden, ist Watson Conversation den anderen Frameworks gegenüber nicht schlechter einzuschätzen, da durch individuelle Entitäten selbiges erreicht werden kann. Jedoch können Entitäten, im Gegensatz zu den anderen Frameworks, nicht in Formulierungen der Aussagen klassifiziert werden, sodass davon auszugehen ist, dass der Algorithmus nur bekannte Entitäten erkennen kann und daher im Bereich dieser nicht lernfähig ist. Ob die angelegten Entitäten die Informationen genauso gut erfassen, lässt sich jedoch erst durch eine partielle Implementierung überprüfen.

Einzig Lex von Amazon muss für die weitere Betrachtung ausgeschlossen werden, da Lex bislang ausschließlich auf Anfrage für einen gewählten Personenkreis freigegeben wird, der im Rahmen dieser Masterarbeit nicht gegeben war.

*Tabelle 4.1: Vergleich der Chat-Bot Frameworks*

Anbieter	Kostenlos	Sprachen	Textanalyse
Facebook (Wit.ai)	Ja	50 inkl. Engl.	Maschinelles Lernen mit Absichten und Eigennamenerkennung
Microsoft (LUIS)	10.000 Nachrichten <sup>39</sup>	8 inkl. Engl.	Maschinelles Lernen mit Absichten und Eigennamenerkennung
Google (Api.ai)	Ja	15 inkl. Engl.	Maschinelles Lernen mit Absichten und Eigennamenerkennung
IBM (Watson Conversation)	1.000 Nachrichten <sup>39</sup>	10 inkl. Engl.	Maschinelles Lernen mit Absichten und Eigennamenerkennung

---

<sup>39</sup> Pro Monat.

Amazon (Lex)	Nein	Nur Engl.	Maschinelles Lernen mit Absichten und Eigennamenerkennung
Recast.ai	25.000 Nachrichten <sup>39</sup>	2 inkl. Engl.	Maschinelles Lernen mit Absichten und Eigennamenerkennung
ChatScript	Ja	Nur Engl. <sup>40</sup>	Regelbasierte Kategorisierung mit Eigennamenerkennung
Pandorabots (AIML)	Ja	136 inkl. Engl. <sup>40</sup>	Regelbasierte Kategorisierung
RiveScript	Ja	Nur Engl. <sup>40</sup>	Regelbasierte Kategorisierung
Motion AI	1.000 Nachrichten <sup>39</sup>	104 inkl. Engl. <sup>40</sup>	Regelbasierte Kategorisierung

Die Frameworks, welche die Aussagen durch ein zu definierendes Regelwerk kategorisieren, unterscheiden sich, abgesehen von Motion AI, durch die Syntax, in welcher die Regeln beschrieben werden, und dem Umfang des Regelwerks. Pandorabots bietet mit AIML eine sehr verbreitete Syntax; RiveScript zeichnet sich mit einer besonders leicht zu verstehenden Syntax aus und ChatScript umfasst, neben der Erkennung von Entitäten, umfangreiche Möglichkeiten bei der Spezifizierung von Regeln. Im Gegensatz zu AIML und RiveScript können Regeln in ChatScript gezielt bestimmte Wörter ausschließen.

Motion AI vereinfacht auf einer grafischen Oberfläche die Definition von Regeln, verwendet aber das gleiche Prinzip und gestattet zudem die Integration zu Kommunikationsdiensten.

Bei der regelbasierten Kategorisierung enthält ChatScript den größten Funktionsumfang und aufgrund der Abhängigkeit zum Regelwerk lässt sich ausschließen, dass andere Frameworks geeigneter sind. Der Nachteil der regelbasierten Frameworks besteht jedoch darin, dass Regeln darauf abzielen, die Merkmale in den Aussagen durch die Erkennung von Stichwörtern und Phrasen auszumachen. Die Anzahl und Komplexität der Regeln wird erstens durch die möglichen Varianten in Satzbau und Wortwahl bestimmt. Zweitens müssen Regeln erstellt werden, welche die Aussagen herausfiltern, die nicht für den Vorfall relevant sind. Beispielsweise müsste die Angabe, dass jemandem der neue Ge-

---

<sup>40</sup> Bei der regelbasierten Kategorisierung wird die Sprache durch die erstellten Regeln bestimmt, da hierbei die sprachlichen Muster explizit angegeben werden.

bäudeanstrich gefällt, durch eine Regel, die alle gefällt mir Angaben filtert, erstellt werden. Folglich würde sich ein umfangreicher Regelsatz ergeben, der im Gegensatz zu den Frameworks mit maschinellen Lernen nicht lernfähig ist und bei Änderungen durch Experten anzupassen ist. Daher werden die regelbasierten Systeme für den weiteren Verlauf nicht weiter in Betracht gezogen.

Zusammenfassend lassen sich bei den Frameworks Wit.ai, LUIS, Api.ai, Watson Conversation und Recast.ai als potenziell geeignet herausstellen. Diese Frameworks bieten auf einer leicht zu bedienenden Weboberfläche die lernfähige Erkennung von Absichten und Entitäten an, welche mit den regelbasierten Systemen nicht realisierbar ist. Ferner scheint es zweckmäßig für die nachfolgende Erstellung eines Vorfallkatalogs, Englisch als Sprache auszuwählen, da diese von allen Chat-Bot Frameworks unterstützt wird und sich der Einfachheit<sup>41</sup> halber anbietet.

#### 4.2.2 Erstellung eines Katalogs an Vorfällen

Die Erstellung eines Vorfallkatalogs, welcher die Aussagen der Vorfälle meldenden Nutzer zusammenträgt, ist für die Frameworks mit Techniken des maschinellen Lernens notwendig. Hierfür werden die Aussagen wörtlich verwendet, um damit den Chat-Bot zu trainieren. Im Anschluss an die Implementierung werden Aussagen verwendet, um die Qualität und Zuverlässigkeit der Klassifizierung zu prüfen.

Im Rahmen dieser Arbeit werden insgesamt 120 Vorfälle erstellt; zum einen soll die Ausarbeitung der Vorfälle nicht zu aufwendig werden, da nicht Hauptbestandteil dieser Arbeit, zum anderen jedoch ausreichen, um genügend Variabilität für das maschinelle Lernen zu erzielen. Zudem lässt sich die Zahl nach der 70/30-Verteilung ohne Rest teilen.<sup>42</sup> Somit enthält das Trainingsset 80 Vorfälle und das Testset 40 Vorfälle. Das Testset wird für die Evaluation der Qualität und Zuverlässigkeit, sowie für den Vergleich der Chat-Bot Frameworks, benutzt. Die 120 Vorfälle sind an den Erfahrungsschatz der synexs GmbH angelehnt.

Aufgrund der Diversität der Vorfälle erscheint es sinnvoll, besonders die Vorfälle vermehrt in den Vorfallkatalog aufzunehmen, die gegenüber weniger kritischen eine höhere Priorität aufweisen, zum Beispiel bei Gefährdung von Personen. Dadurch kann eine genauere Erfassung von gefährlichen Vorfällen sichergestellt werden.

Anhand der Störungs- und Prioritätenkategorien wird die Verteilung über die Art der Vorfälle hergeleitet. Dazu listet die Tabelle 4.2 die Anzahl an Aussagen für Trainings- und

---

<sup>41</sup> Die englische Sprache zeichnet sich unter anderem durch ihre einfache Grammatik aus.

<sup>42</sup> Das 70/30-Verhältnis wird von der Industrie (Amazon, Microsoft) als Standard angesehen. In der Literatur (Mingers, Beleites et al.) wird das Verhältnis aus Größe des Datensatzes und Anzahl an Klassen berechnet. Zur Vereinfachung wird das 70/30-Verhältnis verwendet.

Testset nach Kategorie. Beispielsweise müssen Vorfälle der Kategorie A unverzüglich entstört werden. Die erhöhte Anzahl an Datensätzen erlaubt verschiedenste Formulierungsvarianten als auch Ausprägungen innerhalb dieser Kategorie, sodass diverse Schilderungen erfasst werden können und eine Abgrenzung von anderen Kategorien leichter fällt.

Die Verteilung und Kategorisierung der Vorfälle ist dagegen nicht dazu gedacht, das maschinelle Lernen oder die Definition der Merkmale derart zu spezifizieren, um damit den Vorfall bereits in eine Kategorie einzuordnen, sondern lediglich die Begrifflichkeiten zu erkennen.

*Tabelle 4.2: Verteilung der Vorfälle im Katalog*

<b>Priorität</b>	<b>Prozentual</b>	<b>Trainingsset (80 Vorfälle)</b>	<b>Testset (40 Vorfälle)</b>
1	0,4	32	16
2	0,3	24	12
3	0,2	16	8
4	0,1	8	4

Die prozentuale Verteilung auf die Art des Vorfalls sowie die Inhalte der Vorfälle sind nicht repräsentativ, sondern wie bereits beschreiben, aus den Erfahrungen eines FM-Unternehmens abgeleitet. Deshalb ist nicht auszuschließen, dass bei einem Vergleich mit anderen FM-Unternehmen eine Verschiebung oben erwähnter Werte denkbar ist. Ebenso umfassen die 120 Vorfälle nur einen Teil, wenn auch, nach den Erfahrungen des FM-Unternehmens, wichtigen.

Die Beschreibung eines Vorfalls umfasst ebenfalls die Angabe einer Anschrift. Hierzu werden für die 120 Vorfälle zufällige Adressen aus den 79 Großstädten Deutschlands von Hand ausgewählt. Diese Städte bieten sich wegen der hohen Bevölkerungsdichte, die tendenziell auch eine höhere Gebäudedichte bedeutet, an und sind somit relevanter in Bezug auf das Facility-Management. Bei der Auswahl sollte darauf geachtet werden, dass nicht nur syntaktisch ähnliche Adressen, die zudem noch auf Allee, Straße oder Weg enden, gewählt werden, sondern auch Straßen, die, wie in der Quadratstadt Mannheim, den Straßennamen A3 oder F4 tragen und damit keine interne Evidenz enthalten beziehungsweise keinem definierten Schema entsprechen. So kann vermieden werden, dass sich der Algorithmus für die Erkennung der Straßennamen einem bestimmten Muster beziehungsweise dem Trainingsset zu stark anpasst.

Auf die Vorfälle wird an diese Stelle, angesichts des Umfangs, auf den Anhang verwiesen.



### 4.2.3 Deutungsmöglichkeiten der Aussagen

Ausgehend von dem Vorfallkatalog und dessen Aussagen ist ein Schema, das die benötigten Informationen für die weitere Verarbeitung erfasst und deutet, notwendig.

Aus den Vorfällen gehen Informationstypen über vier Merkmale, die sich in W-Fragen beschreiben lassen, hervor. Zusätzlich kann der Meldende den Vorfall als zeitkritisch beziehungsweise als wichtig einstufen. Die Tabelle 4.3 stellt die Informationstypen mit den verschiedenen Deutungsmöglichkeiten dar.

*Tabelle 4.3: Deutungsmöglichkeiten der Informationstypen*

Informationstyp	Deutungsmöglichkeiten
Wer	<ul style="list-style-type: none"> <li>• die den Vorfall meldende Person</li> <li>• eine dem Meldenden in Beziehung stehende Person, welche ebenfalls gleichen Vorfall beobachtet</li> <li>• die unmittelbar betreffende(n) Person(en) in der Opferrolle</li> <li>• die unmittelbar betreffende(n) Person(en) in der Rolle des Verursachers</li> </ul>
Was	<ul style="list-style-type: none"> <li>• der Vorfall</li> <li>• das/die den Vorfall betreffende(n) Objekt(e)</li> <li>• das/die verursachende(n) Objekt(e)</li> </ul>
Wann	<ul style="list-style-type: none"> <li>• der Zeitpunkt der Meldung</li> <li>• ein Zeitpunkt des Auftretens des Vorfalls</li> </ul>
Wo	<ul style="list-style-type: none"> <li>• die Gebäudeadresse</li> <li>• die Position des Vorfalls auf dem Gelände oder im Gebäude</li> <li>• die aktuelle Position der meldenden Person auf dem Gelände oder im Gebäude</li> </ul>
Dringlichkeit/ bis wann	<ul style="list-style-type: none"> <li>• eine Aussage über die Dinglichkeit</li> <li>• der Zeitpunkt, bis wann der Vorfall behoben werden soll</li> </ul>

Bei der „Wer“-Information werden zum Beispiel bei einem Stromausfall im zweiten Stockwerk der Arbeitskollege, der über selbiges Problem klagt; die im feststeckenden Fahrstuhl eingesperrten Personen; oder der Verursacher, welcher versehentlich eine Fensterscheibe beschädigt hat, erwähnt.

Die „Was“-Information kann den Vorfall beschreiben, beispielsweise einen Stromausfall, die vom Stromausfall betroffenen Geräte oder den defekten Stromverteiler.

Die „Wann“-Information meint entweder den Zeitpunkt, an dem die Meldung des Vorfalles registriert wird; oder den Zeitpunkt, an dem sich der Vorfall, zum Beispiel der Stromausfall, ereignete.

Bei der Wo-Information kann es sich um die Adresse mit Straße und Hausnummer, gegebenenfalls auch Postleitzahl und Ort handeln, einer Position auf dem Gelände oder im Gebäude mit Informationen zu Gebäudeblock, Stockwerk und Raum. Letztere kann die Position des Vorfalles oder des Meldenden beschreiben.

Außerdem kann der Meldende die Dringlichkeit mitteilen. Beim Beispiel des Stromausfalls sind möglicherweise Personen im Fahrstuhl eingeschlossen. Der Meldende könnte zudem eine „bis wann“-Information übermitteln, die von der „Wann“-Information zu unterscheiden ist. Die Information gibt beispielsweise eine Uhrzeit oder Zeitdauer bis zum Eintreffen des Dienstleisters oder der Entstörung an.

Aufgrund der verschiedenen Deutungsmöglichkeiten ist eine verständliche und nach den griceschen Maximen formulierte Fragestellung entscheidend, um Rückfragen und Schwierigkeiten bei der Disambiguierung zu vermeiden. Gleichmaßen kann dem Nutzer durch einen Dialogfluss eine Orientierung angeboten werden. Durch diese Ordnung der Fragen ist es wahrscheinlicher, dass der Nutzer auch eine inhaltliche Ordnung in seinen Aussagen formuliert.

#### **4.2.4 Beschreibung des Dialogflusses**

Der Dialogfluss gibt der Erhebung der Daten, die für eine Meldung eines Vorfalles relevant sind, eine Struktur. Die in 4.2.1 ausgewählten Chat-Bot Frameworks unterstützen alle das rahmenbasierte Dialogmanagement. Die Intention der Aussage bestimmt, welche Daten zu erfassen sind. Sind Daten unverständlich, unvollständig oder fehlen schlichtweg, können diese Daten durch gezielte Rückfragen erhoben werden.

Der Dialogfluss lässt sich, vor den ersten Eingaben, über eine Willkommensnachricht an den Nutzer steuern. Der Chat-Bot übernimmt damit die Initiative für die erste Fragestellung. Alternativ kann auch die Eingabe des Benutzers abgewartet werden. Möglicherweise birgt dies jedoch folgende Nachteile:

- der Meldende bekommt kein Feedback, ob der Chat-Bot funktioniert;
- der Meldende zunächst überlegt, welche Information er erstens übermittelt; oder
- der Meldende nicht relevante Informationen übermittelt.

Einen Vorteil hat die Ersteingabe des Benutzers hingegen nicht. Auch bei der Initiative des Chat-Bots können, wenn der Benutzer nicht wie erwartet auf die Frage des Chat-Bots antwortet, trotzdem Intention und Informationen der Aussage ermittelt und ausgewertet werden.

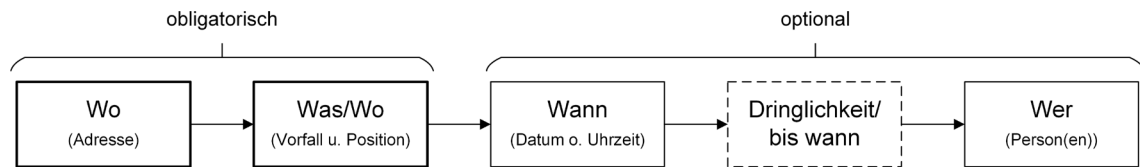


Abbildung 4.1: Dialogfluss

Abbildung 4.1 stellt den Dialogfluss, sortiert nach den wichtigsten Informationen, dar. Der Dialogfluss ist aber kein starres Konstrukt, sondern kann durch die Aussagen des Benutzers verändert werden. Die optionalen Daten und damit Fragen an den Nutzer können durch vorherige Überbeantwortung und Daten aus dem Kontext erfasst werden. Daher kann die Meldung eines Vorfalls bereits nach der Beantwortung von zwei Fragen abgeschlossen sein. Ferner ist es auch möglich, die weitere Befragung zu unterbrechen oder vorzeitig abzuschließen, nachdem die obligatorischen Fragen geklärt sind.

Die bedeutendste zu klärende Information ist die Gebäudeadresse. Gleichmaßen wie beim Notruf ist Ort, Straße und Hausnummer zu nennen. Diese Information kann zudem in einem Kartendienst, wie Google Maps normiert<sup>43</sup>, sowie mit zusätzlichen Informationen zur Facility aus dem CAFM aufgewertet werden. Durch die Informationen aus dem CAFM können verwandte Meldungen zu Vorfällen erfasst und potenzielle Ursachen ein- und ausgeschlossen werden. Überdies kann bei einigen Kommunikationsdiensten wie Facebook Messenger oder Skype der Standort durch den Sensor im Endgerät mitgeteilt werden, sodass die fehleranfällige Texteingabe überflüssig wäre. Eine Texteingabe bleibt dennoch nötig, wenn die automatische Erfassung der geografischen Daten nicht möglich oder zu ungenau ist.

Danach könnte nach der genauen Position auf dem Gelände oder Gebäude, nachfolgend als lokale Ortsangabe bezeichnet, gefragt werden. Jedoch wäre hierbei unklar, ob es sich um die Position des Geschehens oder den Aufenthaltsort der Person handelt. Durch die Verknüpfung von dem Vorfall und der lokalen Ortsangabe in einer Frage wird dem Nutzer vermittelt, dass nach der Position des Geschehens gefragt wird. Beide zusammen sind die zweitwichtigste Informationen und bilden semantisch eine Einheit, daher ist es nicht sinnvoll, diese zu trennen. Ferner erlaubt diese Art der Fragestellung den Dialog möglichst kurz zu gestalten und trotzdem alle relevanten Informationen zu erfassen, ohne den Nutzer zu überfordern.

Beispielsweise ist es sinnvoll, beim Stromausfall zu erfahren, ob dieser das ganze Gebäude, eine Etage oder nur einen Raum betrifft. Daher sind beide Informationen seman-

<sup>43</sup> Darüber hinaus kann mit Diensten, wie SmartyStreets, eine Validierung erfolgen, sodass nur Adressen akzeptiert werden, die existieren. Google Maps und andere Kartendienste zeigen lediglich an, wo sich die Adresse befinden würde. (vgl. [SMST-ol])

tisch stärker verbunden, als andere; folglich sollte die Fragestellung diese gemeinsam erwähnen und erfassen. Weiterhin sind diese Informationen Hauptbestandteil der Problemeinschränkung.

Mit den „Was“ und „Wo“-Informationen sind bereits die relevantesten Daten erfasst. Daraufhin kann die „Wann“-Information erfragt werden. Im Gesprächsverlauf kann unlängst eine „Wann“-Information, durch Formulierungen mit „jetzt“ oder „vor zehn Minuten“, gegeben sein. Insbesondere, wenn der Zeitpunkt oder die Zeitangabe des Vorfalls nicht mit dem Zeitpunkt der Meldung übereinstimmt, kann dies eine wertvolle Information darstellen, beispielsweise bei einem zurückliegenden Einbruch. Außerdem kann diese im Falle von mehreren Meldungen innerhalb einer Facility für die genauere zeitliche Einschränkung genutzt werden. Zudem wird mit dem Kontakt zum Chat-Bot ebenfalls eine zeitliche Information aufgezeichnet, welche als Referenz gilt, ab wann die Entstörungsfrist beginnt.

Durch die Analyse der erhaltenen Informationen kann die Problemeinschränkung eine Priorisierung oder zeitliche Bewertung bereitstellen. Darüber hinaus kann durch die Aussage des Meldenden eine Dringlichkeit beziehungsweise „bis wann“-Information erfasst werden. Beispielsweise beim Stromausfall mit eingesperrten Personen im Fahrstuhl wird der Meldende voraussichtlich auch die Eiligkeit in dem Gesprächsverlauf erwähnen. Bei vorherrschender Unklarheit über die Dringlichkeit empfiehlt es sich, eine entsprechende Rückfrage zu stellen.

Schließlich kann die „Wer“-Information erhoben werden. Über die Kommunikationsdienste können möglicherweise bereits Daten aus dem Profil des Nutzers erfasst werden, wie Name, Telefonnummer oder E-Mail-Adresse. Andernfalls können in einer Fragestellung diese Information erhoben werden. Um bei der „Wer“-Information zwischen dem Meldenden und anderen benannten Personen zu unterscheiden, können Ja-Nein-Rückfragen eingesetzt werden.

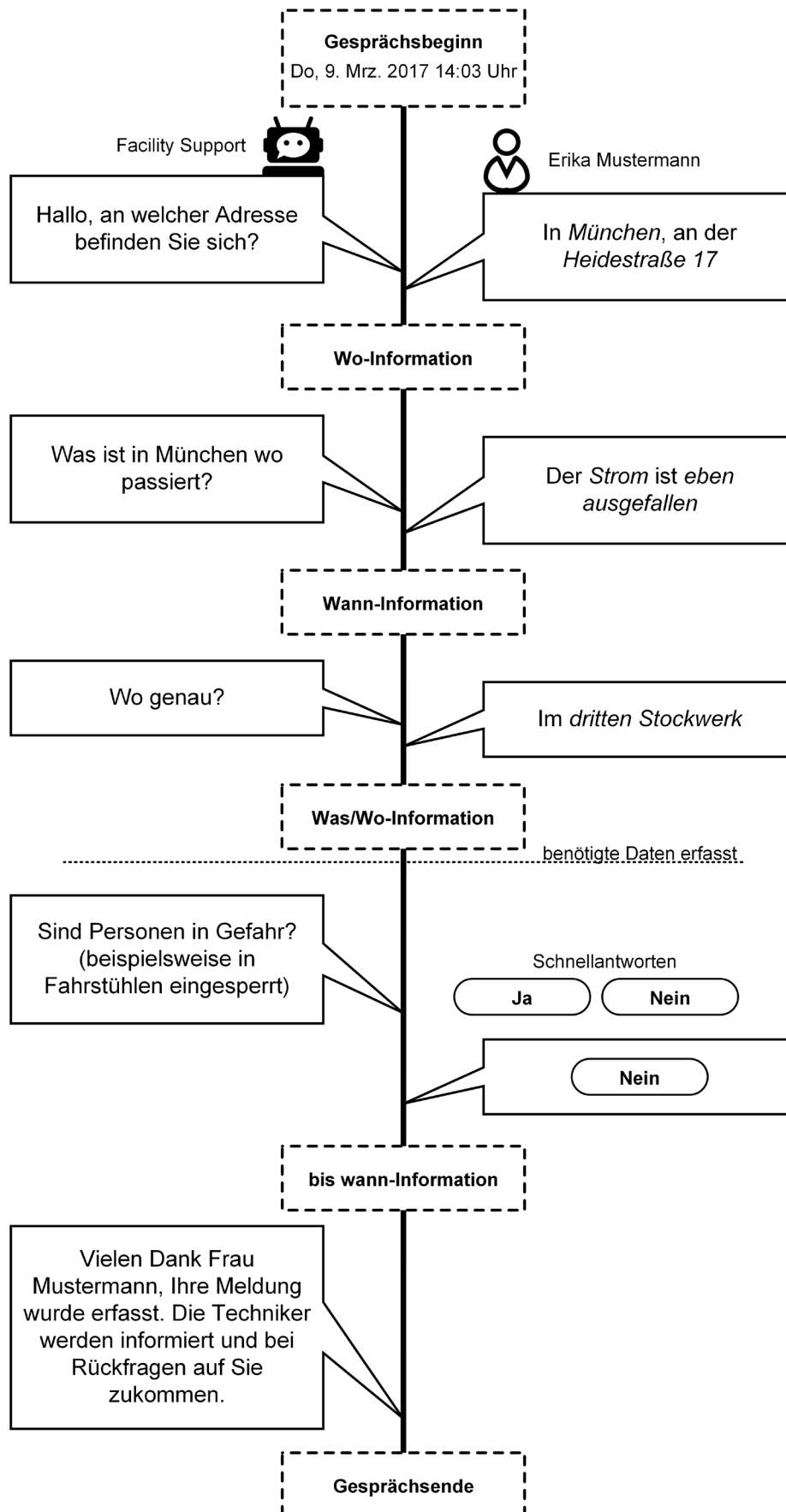


Abbildung 4.2: Beispiel eines Dialogverlaufes

Abbildung 4.2 zeigt ein Dialogbeispiel, in dem der Chat-Bot das Gespräch mit der Frage nach der Gebäudeadresse einleitet. Die Meldende nennt die Adresse, welche im Zeitstrahl dann als erfasst markiert wird. Folgend setzt der Chat-Bot seine Frage entsprechend der Priorisierung fort und benutzt die implizite Bestätigung, um dem Gegenüber zu signalisieren, dass diese Information erfasst wurde. Im Beispiel, auf die Frage nach der „Was“ und „Wo“-Informationen, wird zusätzlich eine zeitliche Information, jedoch nicht die „Wo“-Information übermittelt. Daher wird an dieser Stelle nur die „Wann“-Information als erfasst markiert und die fehlende „Wo“-Information in einer Rückfrage gestellt. Da es ein Problem mit dem Strom gibt, welches als Gefährdungspotenzial gilt, wird eine Frage gestellt, ob sich Personen in Gefahr befinden. Hier bietet es sich an, eine Schnellantwortmöglichkeit bereitzustellen. Schließlich wird der Meldenden mitgeteilt, dass der Vorgang abgeschlossen ist. In diesem Beispiel sind die relevanten „Wer“-Daten im Profil des Kommunikationsdienstes hinterlegt, sodass sich die Frage nach dem „Wer“ erübrigt.

Alle vorgefertigten Formulierungen des Chat-Bots, dazu gehören unter anderem die Begrüßung und die Fragen nach den zu erfassenden Informationen inklusive möglichen Rückfragen, listet Tabelle 4.4. Die Fragen sind verständlich und nach den griceschen Maximen gestaltet. Beispielsweise wird in der Begrüßung mittelbar nach der Adresse gefragt. Hier wird absichtlich das Wort Adresse benutzt, nicht Wörter wie Ort, Position oder Standort, da diese weniger konkret die zu erfassende Information widerspiegeln.

*Tabelle 4.4: Begrüßung, Fragen und Rückfragen des Chat-Bots*

Kategorie	Frage
Begrüßung und initiiierende Frage nach dem Ort	Hello, what is the address of your current location? (Hallo, an welcher Adresse befinden Sie sich?)
Nachfrage nach der Stadt	What is the name of the city you are located at? (Wie heißt die Stadt in der Sie sich befinden?)
Nachfrage nach der Straße	What is the name and number of the street you are located at? (Wie heißt die Straße und Hausnummer in der Sie sich befinden?)
Frage nach dem Geschehen und aktueller Position	What happened in <city> and where exactly? (Was ist in <Stadt> wo passiert?)

Nachfrage nach dem Geschehen	What exactly happened (in <position  floor>)? (Was genau ist (in <Position  Stockwerk>) passiert?)
Nachfrage nach dem Ort des Geschehens	Where exactly (is <Reference>)? (Wo genau (ist <Referenz>)?)
Nachfrage nach der Orientierung oder Stockwerk des Geschehens	What is the orientation or floor of <position>? (Welche Orientierung oder welches Stockwerk gehört zu <position>?)
Nachfrage nach dem Stockwerk des Geschehens	What is the floor of <position>? (Welches Stockwerk hat <position>?)
Nachfrage nach der Raumnummer des Geschehens	What is <position> number? (Welche Nummer hat <position>?)
Frage nach dem Zeitpunkt des Geschehens	When did it happen? (Wann ist es passiert?)
Frage nach der Priorität (bis wann)	Are there people in danger? (Eg locked in elevators) (Sind Personen in Gefahr? (beispielsweise in Fahrstühlen eingesperrt))
Frage nach dem/der Meldenden	Can you please tell me your name and contact? (Nennen Sie mir bitte Ihren Namen und eine Kontaktmöglichkeit)

#### 4.2.5 Datenmodell eines Vorfalles und dessen Meldungen

Für die eingehenden Meldungen zu einem Vorfall ist ein Datenmodell nötig, welches die Informationen strukturiert abspeichert, sodass diese für die weitere Bearbeitung durch andere Systeme erfasst werden können.

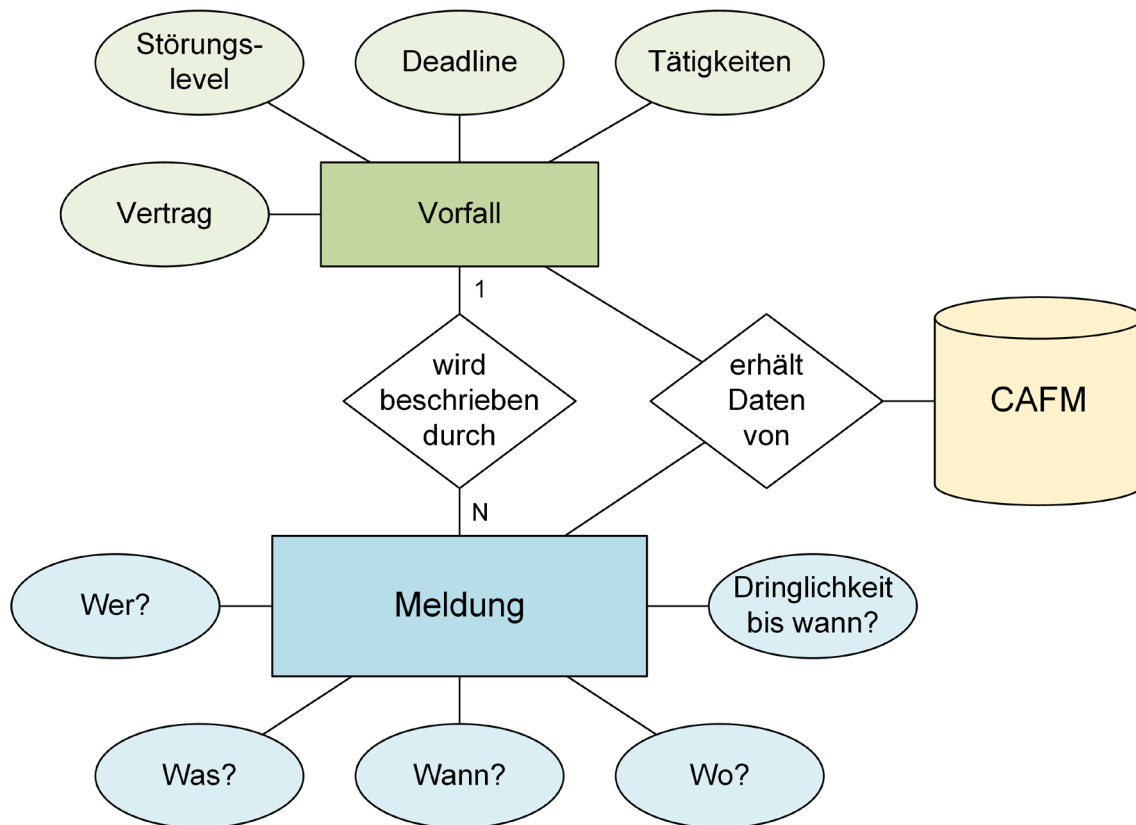


Abbildung 4.3: Datenmodell eines Vorfalls und Meldungen

Die Abbildung 4.3 stellt ein solches Datenschema vor. Die blau markierte Entität der Meldung zeigt die in dieser Arbeit zu realisierende Informationserfassung. Darüber hinaus werden die Beziehungen zum Vorfall und dem zentralen Datenspeicher, dem CAFM skizziert. Der Vorfall wird durch beliebig viele Meldungen beschrieben, die insgesamt den Informationsgehalt über das Geschehen erhöhen können. An einem Vorfall hängen Attribute, welche die zeitlichen Restriktionen und damit verbundenen Tätigkeitsfelder für die Entstörung beschreiben. Überdies regelt der Vertrag die Details für die Behebung der Störung.

Um den Vorfall vertragskonform zu beheben, sind aus den Meldungen die Priorität und die Tätigkeiten für die zu beauftragenden FM-Dienstleister auszulesen. Erstere leiten sich aus dem Geschehen und der Schwere beziehungsweise dem Ort des Geschehens ab. Letztere ergeben sich aus der Art des Vorfalls.

Im Folgenden werden zuerst die zu erfassenden Informationen in die Bestandteile zerlegt, sodass anhand deren Ausprägung die Einteilung in Prioritäten und Tätigkeitsfelder erfolgen kann.



#### 4.2.6 Bestandteile der zu erfassenden Daten

Die zu erfassenden Daten wurden durch die W-Informationen, welche der Nutzer auf die W-Fragen antwortet, umschrieben. Die W-Informationen setzen sich jedoch aus einzelnen Bestandteilen zusammen, wie teils in den Beispielen angedeutet. Diese Bestandteile einer W-Information müssen, um die vollständige Information zu erlangen, erkannt werden.

Die Gebäudeadresse ist per Definition eine wohldefinierte Angabe aus Straßenbezeichnung, Hausnummer, Postleitzahl und Ortschaft. Optional ist die Länderangabe. (vgl. [WIKI-1-ol]) Da sich der Vorfallkatalog auf deutsche Großstädte beschränkt, ist die Länderangabe nicht relevant und wird im Rahmen dieser Arbeit vernachlässigt. Die Ortschaft meint in Deutschland den organisatorischen oder rechtlichen Status einer Siedlung, also beispielsweise Gemeinden und kreisfreie Städte (vgl. [WIKI-2-ol]). Die Postleitzahlen geben Zustellorte an und orientieren sich nicht zwangsläufig an Gemeindegrenzen (vgl. [WIKI-3-ol]). Daher repräsentieren die Postleitzahl und die Ortschaft nicht die gleiche Information. Jedoch sollten die Straßennamen innerhalb einer Ortschaft einmalig und eindeutig sein (vgl. [WIKI-4-ol]). Deshalb ist es für diese Arbeit weniger relevant, ob beide Angaben vorhanden sind. Zudem wird zur Vereinfachung angenommen, dass beide die gleiche Information beschreiben. Natürlich sollten beide Angaben, wenn diese angegeben werden, erfasst werden, sodass die Übereinkunft beider Werte geprüft werden kann, um potenzielle Tippfehler auszuschließen.

Verglichen mit der Adresse ist die „Was“ und „Wo“-Information, also die Erfassung des Vorfalls und dem Ort des Geschehens, deutlich komplexer. Durch geeignete Informationstypen ist zum einen der Vorfall und zum anderen der Ort innerhalb eines Gebäudes oder auf einem Grundstück zu beschreiben. Im Unterkapitel der Deutungsmöglichkeiten wurde bereits in die Beschreibung des Vorfalls, die den Vorfall betreffenden Objekte und die den Vorfall verursachenden Objekte unterschieden. Ebenfalls wurde im Dialogfluss die starke Verbindung des Vorfalls mit der lokalen Ortsangabe herausgestellt.

Aus der Analyse der Vorfälle wurden zwei Informationstypen zu einem Vorfall herausgearbeitet und als maßgeblich erachtet. Beide Typen geben samt der lokalen Ortsangabe die wesentliche Information zu einem Vorfall wieder.

Einerseits kann ein Vorfall ein **Symptom** beschreiben, wie, dass etwas passiert ist, etwas nicht richtig ist, etwas nicht funktioniert, etwas nicht möglich ist oder etwas fehlt. Ähnlich wie in der Medizin die Krankheit, beschreibt das Symptom die Anzeichen und Merkmale eines Vorfalls. Andererseits kann die Schilderung eine **Referenz** auf ein Ereignis oder Geschehen („Feuer“), eine oder mehrere Personen („Einbrecher“) oder ein oder mehrere Objekte („Tresen“) beinhalten. Stellt eine Referenz jedoch eine ortsbezeichnende Information dar, so wird in diesem Fall die Angabe vorrangig der lokalen Ortsangabe zugeordnet, nicht der Referenz. Ein weiterer Sonderfall meint Ereignisse, die mittelbar auch

ein Symptom beschreiben, dieses aber nicht die Ursache des Vorfalls darstellt, beispielsweise „Feuer“. Das Symptom und die Referenz entsprechen den Deutungsmöglichkeiten der „Was“-Information.

Die Informationstypen wurden so gewählt, dass jeder Typ einen sinngebenden Zusammenhang mit dem Vorfall besitzt, dieser aber nicht zu speziell ist. Beispielsweise könnte das Symptom „total zerstört“ in dem Satz: „der Tresen in meinem Geschäft wurde total zerstört“ weiter in ein Eigenschaftswort und die eigentliche Symptomatik aufgeteilt werden. Jedoch würde dies die Informationsgewinnung aus den Aussagen nicht verbessern. Erstens wird die Eigenschaft bereits erfasst und im Nachgang kann diese immer noch detaillierter ausgewertet werden. Zweitens erhöht sich mit steigender Anzahl an Entitäten die Komplexität des maschinellen Lernens, wohingegen sich die Zuverlässigkeit verringert. Zum Beispiel könnte dann das Wort „teure“ aus der Aussage „der teure Tresen ...“ fälschlicherweise als Eigenschaftswort des Symptoms erkannt werden.

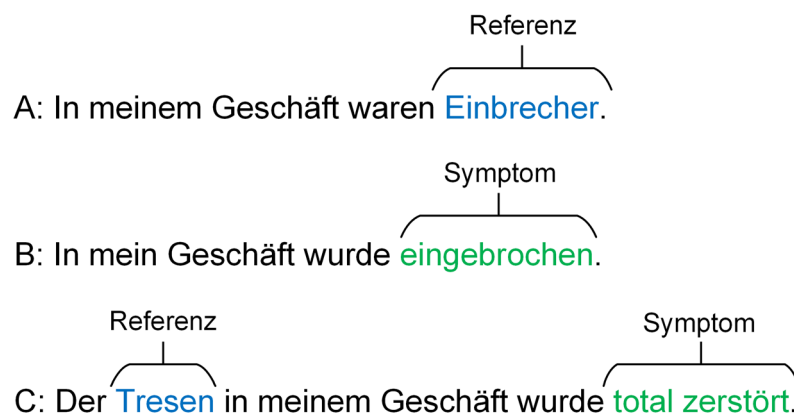


Abbildung 4.4: Aussagenklassen eines Vorfalls

Durch die beiden Informationstypen lassen sich die Vorfälle in drei Aussagenklassen unterscheiden. Die Abbildung 4.4 stellt diese Klassen A, B und C dar. Die Klasse A nennt die Referenz, ohne das Symptom zu nennen. Klasse B beschreibt das Symptom eines Vorfalls ohne eine Referenz und Klasse C nennt sowohl die Referenz als auch das Symptom.

Außerdem wird mit dem Vorfall der Ort des Geschehens abgefragt. In Abbildung 4.5 werden die identifizierten Ortsbezeichnungen aus dem Vorfallkatalog anhand von Beispielen erläutert. Die Ortsbezeichnung unterscheidet in folgende fünf Arten:

- der **feststehenden Ortsbezeichnung**, welche allgemein geläufig und bekannt ist,
- der **allgemeinen Ortsbezeichnung**, die in der Aussage vage ist und viele Bereiche meinen kann,
- der **exakten Ortsbezeichnung**, die durch eine Identifikationsnummer den genauen Ort ausdrückt,
- der **einschränkenden Ortsbezeichnung**, welche bestimmte Bereiche in einem Gebäude meint, diese aber offenlässt, welche genau, und

- der **präzisierenden Ortsbezeichnung**, die bestimmte Bereiche eingrenzt und somit den Ort soweit einschränkt, dass dieser rasch auffindbar ist.

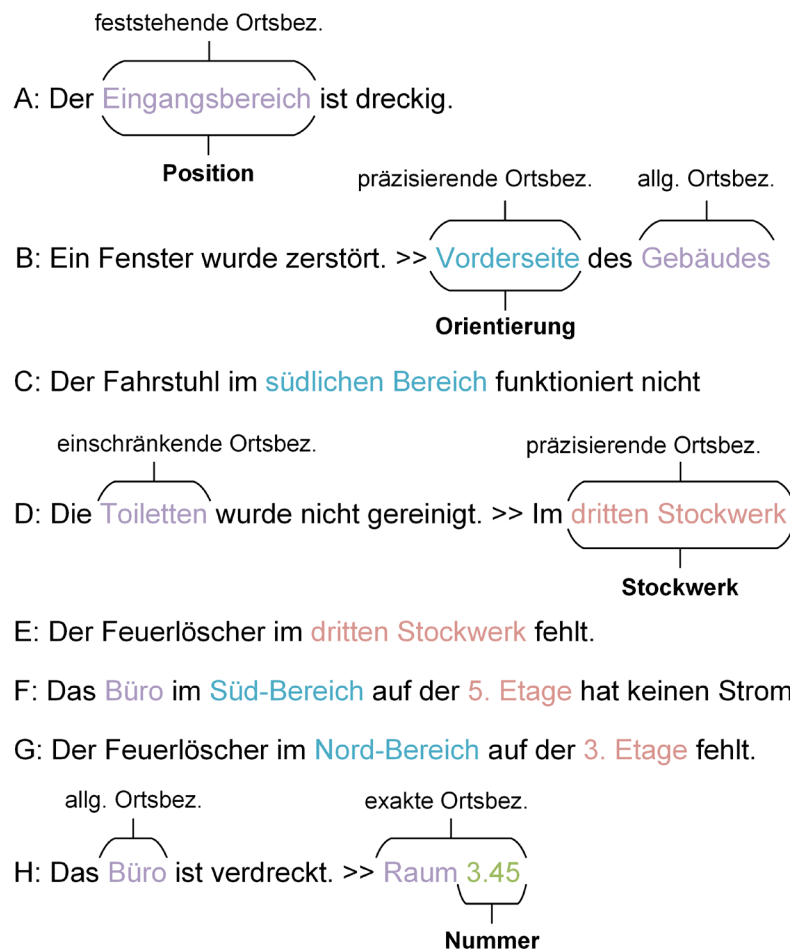


Abbildung 4.5: Lokale Ortsbezeichnungen und Entitäten

Aus den fünf lokalen Ortsbezeichnungen, welche die semantische Bedeutung repräsentieren, wurden zur systematischen Darstellung vier Informationstypen (Position, Orientierung, Stockwerk und Nummer) identifiziert.

Die **Position** meint Bereiche, Räume oder Gebäude(teile). Dabei kann die Position eine allgemeine, einschränkende oder feststehende Ortsbezeichnung annehmen. Die allgemeine Ortsbezeichnung kann sich auf Räume oder Gebäude(teile) beziehen. Bei Räumen ist eine exakte Ortsbezeichnung notwendig, wohingegen bei Gebäude(teile)n bestenfalls eine Orientierung angegeben kann. Die einschränkende Ortsbezeichnung benötigt die Angabe einer Orientierung oder eines Stockwerks, die feststehende Ortsbezeichnung hingegen nicht.

Durch eine Auswertung, ob die Position den beschriebenen Anforderungen genügt, lässt sich feststellen, ob weitere Informationen eingeholt werden müssen. Dementsprechend

sind die Aussagen im zweiten, vierten und letzten Beispiel nicht ausreichend präzise, so dass eine Rückfrage gestellt wird. Aus Gründen der Übersichtlichkeit wurde auf die Darstellung der Rückfrage an dieser Stelle verzichtet.

Zudem wird erneut der semantische Zusammenhang des Vorfalls und des lokalen Ortes deutlich. Im Beispiel stellen der Eingangsbereich, das Büro und die Toiletten jeweils eine Referenz dar, diese werden aber vorrangig der lokalen Ortsbezeichnung zugeordnet. Anzumerken ist, dass Beschädigungen an Fenstern, Türen oder Wänden als Referenz anzusehen sind, wohingegen die Eingangstür, das Rolltor oder die Fassade eine lokale Ortsbezeichnung widerspiegeln.

Die **Orientierung** stellt eine (geografische) Richtung oder einen relativen Verweis auf etwas dar und ist im Gegensatz zu der Position nicht physisch. Das **Stockwerk** sowie die **Nummer** geben die Etage beziehungsweise die Raumnummer an.

Aufgrund dieser vier Informationstypen können die Angaben zum Ort des Geschehens in acht Aussagenklassen unterschieden werden. Der Angabe:

- A. der Position (als feststehende Ortsbez.);
- B. der Position (als einschränkende Ortsbez.) und der Orientierung;
- C. der Orientierung;
- D. der Position (als einschränkende Ortsbez.) und des Stockwerks;
- E. des Stockwerks
- F. der Position (als einschränkende Ortsbez.), der Orientierung und des Stockwerks;
- G. der Orientierung und des Stockwerks, und
- H. der Position (als allgemeine Ortsbez.) und Nummer.

Ferner können zusätzliche Ortsinformationstypen erkannt werden, jedoch ist die systematische Verarbeitung der Ortsangaben stets auf dieses Schema zurückzuführen. Hierbei sind präzisere Informationen den weniger präzisen vorzuziehen. Somit lässt sich bei vorhandener Information zu Position, Orientierung und Nummer die Aussage in die Aussagenklasse H einordnen, da die Angabe der Orientierung in Bezug auf die Präzision unbedeutend wird.

Die Erfassung des Zeitpunktes oder von zeitlichen Informationen begrenzt sich auf einen Informationstyp. Der Zeitpunkt wird durch eine textuelle oder in einem Zeitformat definierte Angabe übergeben.

Die Dringlichkeit kann, wie im Beispiel des Dialogverlaufes in Abbildung 4.2, durch eine Ja-Nein-Frage mit vordefinierten Antworten abgefragt werden, wenn dies aus dem Dialogverlauf heraus als nötig erachtet wird. Daher wird dafür keine Entität benötigt.

Die Erfassung der „Wer“-Daten meint die persönlichen Informationen des Meldenden. Diese werden durch den Vor- und Nachnamen, die E-Mail-Adresse und die Telefonnummer bestimmt.

Die Internet Engineering Task Force beschreibt mit dem RFC 2822<sup>44</sup> Standard das Format von E-Mail-Adressen. Darüber hinaus haben verschiedene Organisationen wie die Internationale Fernmeldeunion mit der Kennung E.123 und das Deutsche Institut für Normung mit der Norm 5008 Gestaltungsregeln für E-Mail-Adresse und Telefonnummer erstellt.

Die benötigten Absichten und dazugehörigen Entitäten für die Aufnahme einer Meldung werden in Tabelle 4.5 zusammengefasst. Die optionalen Absichten und Entitäten sind kursiv dargestellt.

*Tabelle 4.5: Absichten und deren Entitäten*

<b>Absicht</b>	<b>Entitäten</b>
None/Default (Keine/Standard)	
ProvideAddress (Adresse angeben)	<ul style="list-style-type: none"><li>• City name (Stadt)</li><li>• <i>Zipcode</i> (Postleitzahl)</li><li>• Street and Number (Straße und Hausnummer)</li><li>• <i>Time</i> (Zeitpunkt)</li></ul>
ProvideIncidentAndPosition (Vorfall und Position angeben)	<ul style="list-style-type: none"><li>• Reference (Referenz)</li><li>• Symptom</li><li>• Position</li><li>• <i>Orientation</i> (Orientierung)</li><li>• <i>Floor</i> (Stockwerk)</li><li>• <i>Room number</i> (Raumnummer)</li></ul>

---

<sup>44</sup> Der Request For Comments (RFC) 2882 wurde im April 2001 verfasst und ist auf folgender Webseite einsehbar: <https://www.ietf.org/rfc/rfc2822>.

<i>ProvideIncidentTime</i> (Zeitpunkt des Vorfalls angeben)	<ul style="list-style-type: none"> <li>• Time (Zeitpunkt)</li> </ul>
<i>ProvideIncidentUrgency</i> (Dringlichkeit des Vorfalls angeben)	
<i>ProvidePersonalInformation</i> (Persönliche Informationen angeben)	<ul style="list-style-type: none"> <li>• Name</li> <li>• E-Mail-Address (E-Mail-Adresse)</li> <li>• Phone number (Telefonnummer)</li> </ul>

Sollte der Nutzer Eingaben formulieren, die die Fragen des Chat-Bots nicht beantworten oder irrelevante Informationen bereitstellen, so sind solche Absichten von anderen Absichten zu identifizieren und zu unterscheiden. Daher ist eine Standardabsicht nötig, denn ohne eine solche Absicht oder eine zu geringe Anzahl an trainierten Lernbeispielen kann es zu falsch klassifizierten Absichten im maschinellen Lernen kommen. Diese Aussagen liegen dem Anhang bei.

#### 4.2.7 Kategorisierung des Vorfalls

Aus den Informationen der Meldung sollen Rückschlüsse auf die Art des damit verbundenen Vorfalls, dessen Dringlichkeit und den für die Entstörung benötigten Dienstleister gewonnen werden.

Abbildung 4.6 veranschaulicht die bereits gewonnenen Erkenntnisse und gibt einen Ausblick auf dieses Kapitel. Aus den zuvor erarbeiteten Informationstypen Referenz (beziehungsweise Position, wenn keine Referenz angegeben) und Symptom wird der Vorfallstyp entwickelt. Dieser soll dazu dienen, die Art des Vorfalls mittels der in 3.1.4 vorgestellten Klassifikation einzuschränken. Unter der Annahme, dass die Gesamtheit der vorab definierten Klassen den vollständigen Katalog an Problemen darstellt, schränkt die jeweilige Zuordnung zu einer Klasse das Problem ein. Daher entspricht dies einem Klassifizierungsproblem.

Bei der Referenz, dem Symptom und der Position handelt es sich um qualitative Daten. Für die Umsetzung wird ein Verfahren benötigt, welches aufgrund beider Merkmale eine Klassifikation durchführt. Ebenfalls ist zu beachten, dass der Vorfallstyp, wie nachfolgend beschrieben, durch mehrere Klassen repräsentiert wird. Das Verfahren muss daher eine Klassifizierung in verschiedene Einteilungsklassen erlauben. Der C4.5 Algorithmus erfüllt diese Anforderungen. Auch in der medizinischen Diagnose, die Ähnlichkeiten mit diesem Anwendungsfall aufweist, werden unter anderem Entscheidungsbäume genutzt (vgl. [Qui86], S. 83ff; [Kon01], S. 105).

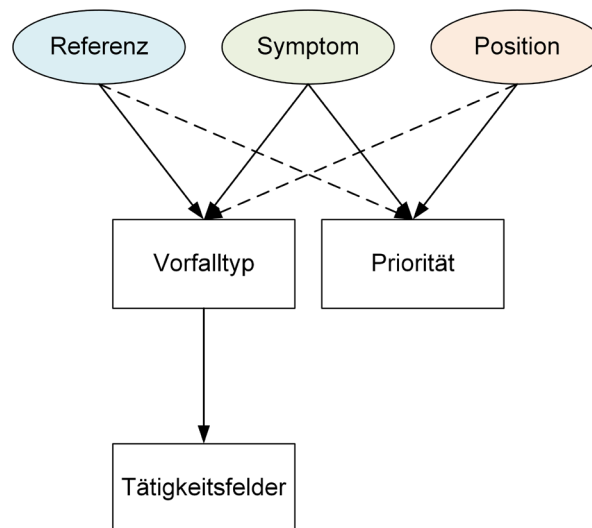


Abbildung 4.6: Informationsabhängigkeiten

Ebenso ist die Klassifizierung für die Priorität umzusetzen, welche sich aus den Merkmalen des Symptoms und der Position (beziehungsweise der Referenz) zusammensetzt.

Der Vorfalltyp wurde so gewählt, dass sich, wie im Folgenden erläutert, daraus die Tätigkeitsfelder ableiten. Daher bedarf es für die Tätigkeitsfelder keines Klassifikationsverfahrens.

Für diese drei Informationen (Vorfalltyp, Tätigkeitsfelder, Priorität) werden Kategorien für die Klassifizierung benötigt. Die Prioritätenkategorien wurden bereits in Tabelle 2.2 zusammengestellt.

Für die Klassifikation des Vorfalls und der dafür benötigten Dienstleister wurden aus dem Vorfallkatalog 13 Vorfalltypen herausgearbeitet. Diese wurden obendrein so gewählt, dass die Priorität bereits ohne die Positionsangabe eingeschränkt werden kann. Der Typ „Gefahr“ beschreibt die aus Priorität 1 bekannten Fälle von der Gefahr für Leib und Leben und ernststen Sicherheitsrisiken. Die beiden Typen „Defekt“ und „Sicherheit“ bilden ähnliche Bereiche ab, die aber eine geringere Priorität aufweisen. Der Typ „Defekt“ meint alle anfallenden kleineren Störungen oder die, die nicht zugeordnet werden können. Der Typ „Sicherheit“ vereinigt alle Sicherheitsrisiken, die kein akutes Sicherheitsrisiko darstellen. Die Sicherstellung der Energieversorgung wird durch den Typ „Elektrizität“ gewährt. Probleme mit den technischen Anlagen, wie Heizung, Klima und Lüftung, werden durch ebensolchen Typ abgebildet. Der „Temperatur“ Typ ist letzterem ähnlich, begrenzt sich jedoch auf Probleme mit der Raumtemperatur, Luftqualität und Luftfeuchtigkeit, die nicht zwangsläufig aus einer defekten Anlage resultieren. Ebenso ist der Typ „Geruch“ eine Sammlung für die Luftqualität und unangenehme Gerüche, die beispielsweise durch andere Probleme, wie Verschmutzung oder Reinigungsmängel, hervorgerufen werden. Die Vorfälle in den sanitären Anlagen, wie Defekte, Störungen und Reinigungsmängel werden allesamt in diesem Typ abgebildet. Der Typ „Wasser“ meint alle Probleme mit Wasser, die außerhalb der sanitären Anlagen auftreten. Das äußere Erscheinungsbild,

dazu gehören der Garten und die Fassade, werden in entsprechendem Typ zusammengefasst. Die Tabelle 4.6 fasst diese Typen zusammen und nennt die für die Entstörung relevanten Berufe.

*Tabelle 4.6: Vorfalltypen und relevante Berufe*

Typ	Priorität	Berufe
Gefahr	1	Fachkraft für Schutz und Sicherheit
Defekt	2-4	
Elektrizität	1-2	Elektroniker
Datennetzwerk	1-2	Elektroniker für Geräte und Systeme/Systemelektroniker/Systeminformatiker
Wasser	1-3	<ul style="list-style-type: none"> <li>• Klempner</li> <li>• Gebäudereiniger</li> </ul>
Temperatur	1-3	Anlagenmechaniker für Sanitär-, Heizungs- und Klimatechnik
Geruch	1-3	<ul style="list-style-type: none"> <li>• Anlagenmechaniker für Sanitär-, Heizungs- und Klimatechnik</li> <li>• Gebäudereiniger</li> </ul>
Zutritt	1	Fachkraft für Schutz und Sicherheit
Sicherheit	2	<ul style="list-style-type: none"> <li>• Fachkraft für Schutz und Sicherheit</li> <li>• Elektroniker für Energie- und Gebäudetechnik</li> </ul>
Sauberkeit	1-4	Gebäudereiniger
Erscheinungsbild	1-2	<ul style="list-style-type: none"> <li>• Gärtner</li> <li>• Maler und Lackierer</li> </ul>
technische Anlagen	1-2	<ul style="list-style-type: none"> <li>• Anlagenmechaniker für Sanitär-, Heizungs- und Klimatechnik</li> <li>• Elektroniker für Gebäude- und Infrastruktursysteme</li> </ul>



sanitäre Anlagen	1-3	<ul style="list-style-type: none"> <li>• Anlagenmechaniker für Sanitär-, Heizungs- und Klimatechnik</li> <li>• Klempner</li> <li>• Gebäudereiniger</li> </ul>
------------------	-----	---

In Tabelle 4.6 wurde bewusst der Beruf des Hausmeisters oder Hauswarts ausgelassen. Dieser kann, sofern in der Nähe, für eine genauere Einschränkung und Einschätzung des Vorfalls sorgen und ist somit für jeden Typ relevant. Insbesondere für die Defekte, die in erster Linie durch diesen behoben werden.

Abbildung 4.7 skizziert die Vorfalltypen und die für die Entstörung relevanten Berufsgruppen. Jede Berufsgruppe, außer dem Gärtner und dem Maler (welche sich eine Farbe teilen), ist in einer Farbe dargestellt. Durch Nähe versucht das Diagramm miteinander verwandte Tätigkeiten abzubilden. Die sechs Typen, die durch größere Ellipsen hervorgehoben werden, zeichnen sich durch drei Eigenschaften aus. Erstens wird in der Problemschilderung von Problemen gesprochen, die nicht zwangsläufig an Geräten oder Objekten festgemacht werden können, beispielsweise der Strom, die Luft oder das Internet. Zweitens werden diese als Hyperonym, also einer Oberkategorie an Problemen, verstanden. Drittens zeichnen sich diese Vorfälle durch eine tendenziell höhere Priorität aus.

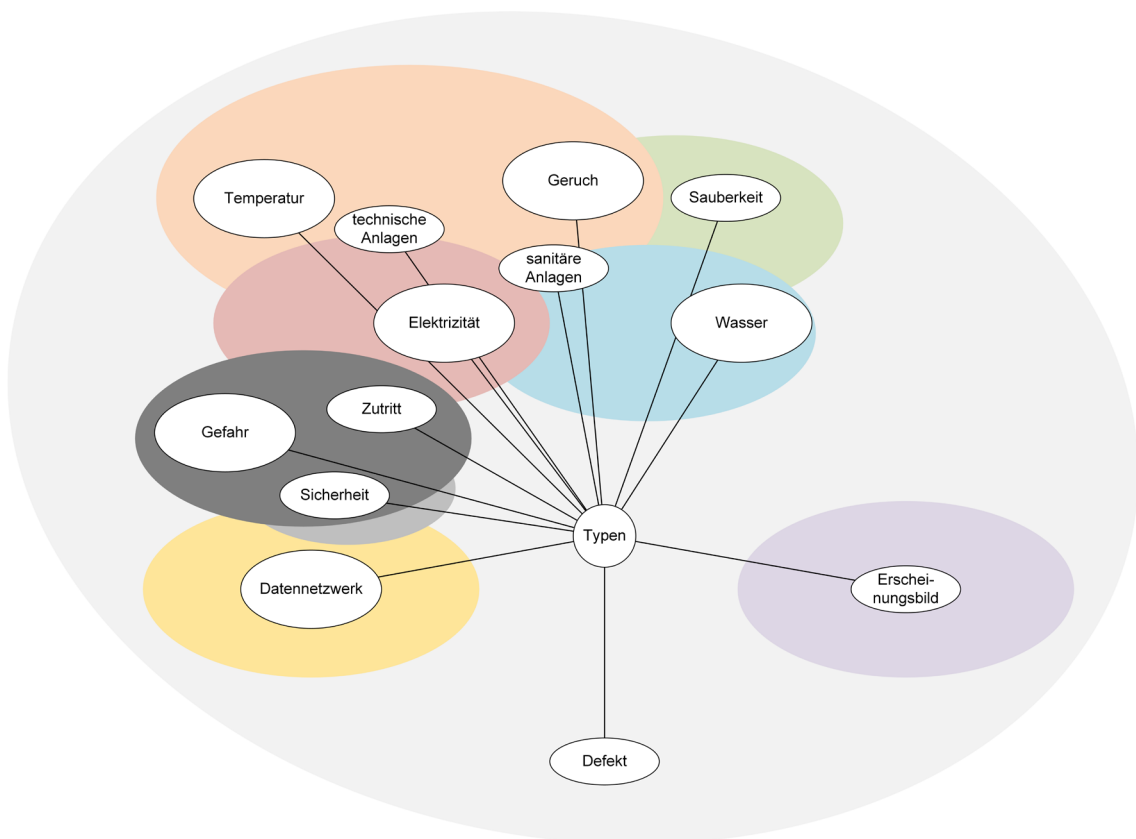


Abbildung 4.7: Typklassifikation der Vorfälle und relevanten Berufe

Die für die Klassifizierung benötigten Komponenten stehen folglich bereit. Die Vorfälle aus dem Vorfallkatalog, die Vorfalltypen und die Merkmale in den Vorfällen können im Folgenden dazu benutzt werden, den Klassifizierungsalgorithmus zu trainieren.

Im Falle von durch den Algorithmus nicht klassifizierbarer Vorfälle ist die pauschale Zuordnung in den allgemeinen Typ „Defekt“ denkbar. Alternativ ist eine Auswahl durch den Meldenden oder die Verbindung mit einem Supportmitarbeiter möglich.

Werden mehrere Vorfalltypen erkannt, kann ebenfalls eine Auswahl dieser durch den Meldenden erfolgen.

Bei Störfällen, die nicht in der Typklassifikation erfasst werden, ist eine Anpassung des Schemas notwendig.

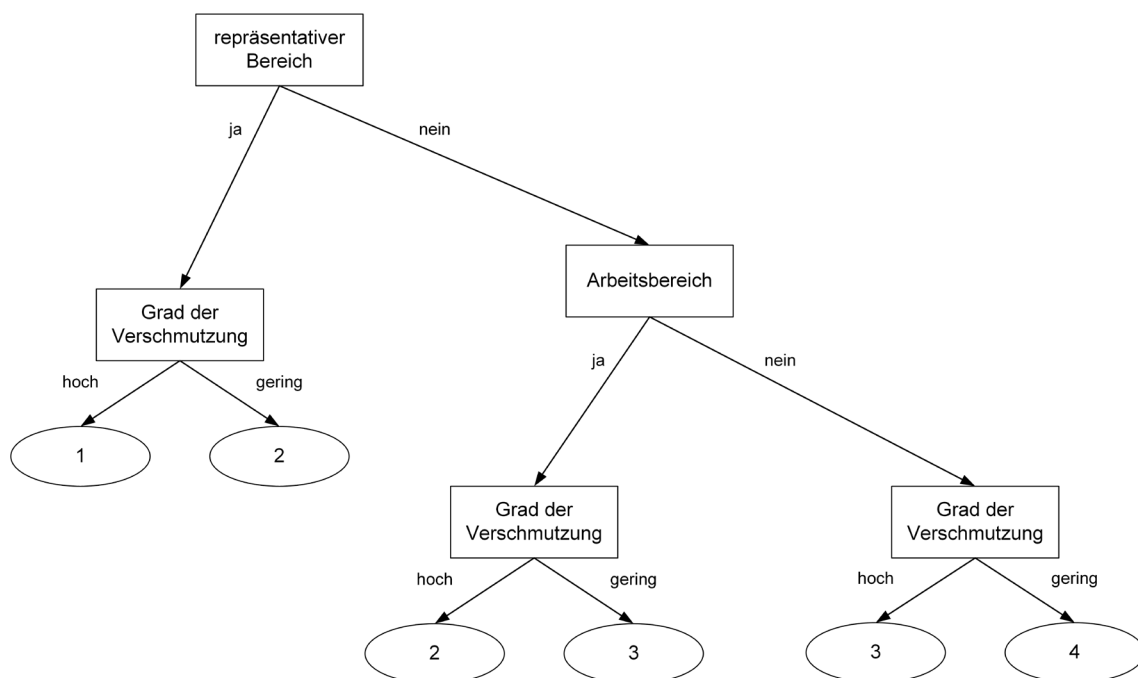


Abbildung 4.8: Beispiel eines Entscheidungsbaums für Priorität der Sauberkeit

Die Abbildung 4.8 skizziert ein Beispiel für einen Entscheidungsbaum, der das Ergebnis des Klassifizierungsalgorithmus ist. Hierbei wird die Priorität bei dem Typ Sauberkeit anhand von drei Entscheidungsregeln festgestellt.

## 5 Implementierung

Das Kapitel der Implementierung beschreibt die Realisierung des im vorherigen Kapitel beschriebenen Konzepts. Dabei liegt der Schwerpunkt auf der Erfassung des Geschehens und des Ortes, weshalb diese Aspekte detaillierter beschrieben werden. Die Implementierung sowie eine Kurzanleitung für die Inbetriebnahme des Chat-Bots ist unter [Kla-1-ol] zu finden. Die vollständige Anleitung ist im Anhang.

Im Folgenden wird an Hand der Architektur ein Überblick der Implementierung und den damit verbundenen Komponenten gegeben, das Verständnis natürlicher Sprache durch Absichten und Entitäten in Bezug auf die Chat-Bot Frameworks beschrieben, die Abläufe und Zustände im Dialogmanagement erläutert, die zu erfassenden Daten und Modelle dargestellt sowie die Benutzeroberfläche skizziert.

### 5.1 Architektur

Der Abschnitt erläutert die gewählte Chat-Bot-Architektur, die verwendeten Komponenten und Pakete sowie die Aufteilung der Programme.

Aufgrund der Tatsache (siehe 4.2.1), dass eine Bewertung und damit Auswahl der bereits gewählten Chat-Bot Frameworks, ohne zumindest eine teilweise Implementierung zu vollziehen, nicht möglich ist, ist eine Realisierung des Chat-Bots in allen Frameworks notwendig.

Zwar ähneln sich die Frameworks in ihrer Grundfunktion (dem Erkennen von Absichten und Entitäten), jedoch bieten sie zudem erweiternde und unterstützende Funktionalitäten, wie die Sprachkorrektur, an. Folglich kann dadurch die Bewertung der zu prüfenden NLU-Funktionalität erschwert werden. Darüber hinaus ist der Aufwand für die separate Implementierung in den Chat-Bot Frameworks mit den jeweiligen Eigenheiten nicht unerheblich.

Botkit (siehe 3.2.11) ermöglicht durch Schnittstellen die Konnektivität zu Komponenten der natürlichsprachlichen Analyse und Verarbeitung. Genauer gesagt kann mit Botkit die NLU-Funktionalität aus den vorgestellten Chat-Bot Frameworks angesprochen und durch standardisierte Schnittstellen nutzbar gemacht werden. Dadurch reduziert sich die Implementierung für die jeweiligen NLU-Dienste auf die Vervollständigung der Schnittstelle zu Botkit.

Das Dialogmanagement, die Schnittstellen zu externen Diensten und Ressourcen sowie die Erstellung von natürlichsprachlichen Antworten und Fragen können in Botkit abgebildet werden.

Aus diesen Gründen eignet sich Botkit für die Implementierung eines Chat-Bots innerhalb dieser Arbeit. Botkit ist ein Node.js-Modul, wodurch sich Node.js als zugrunde liegende Plattform ergibt. Letztere ist eine serverseitige Laufzeitumgebung, die JavaScript,

in einer ressourcensparenden und auf viele gleichzeitig bestehende Netzwerkverbindungen optimierte Architektur, interpretiert. Außerdem unterhält Node.js mit dem Paketmanager npm die weltweit größte Sammlung an Open-Source Bibliotheken (vgl. [NODE-ol]).

### 5.1.1 Komponenten

Der Chat-Bot besteht, wie in Abbildung 5.1 dargestellt, im Wesentlichen aus zwei Komponenten. Der Benutzeroberfläche (*User Interface*) und der Logik (*Logic*). Komponenten, die mit einer gestrichelten Linie skizziert sind, werden entweder nicht genutzt oder werden im weiteren Verlauf nicht implementiert (siehe 5.2.1). Diese zeigen jedoch exemplarisch Nutzungsalternativen und Funktionalitätserweiterungen auf.

Die **Benutzeroberfläche** zeigt die auf den Endgeräten vorhandenen Kommunikationsdienste und deren Anwendung. Die Ausnahme stellt der Microsoft Bot Framework Channel Emulator (*Emulator*) dar, welcher nur zu Test- und Entwicklungszwecken genutzt wird. Der Microsoft Bot Framework Connector (*Bot Connector*) verbindet sich mit den jeweiligen Kommunikationsdiensten. Zudem stellt der Bot Connector eine Schnittstelle bereit, mit der sich beispielsweise das Bot Builder SDK (ein Teil des Microsoft Bot Frameworks), aus der Logik-Komponente verbinden kann. Hierdurch werden Nachrichten seitens der Benutzer mit den vom Chat-Bot generierten Nachrichten ausgetauscht.

Die **Logik** enthält den Dialogmanager, bestehend aus dem *Botkit*, Schnittstellen zu den NLU-Diensten (*NLU (Connector)*) sowie dem *Backend*, welches externe und zusätzliche Funktionalität (siehe unten) bereitstellt. Das Botkit als zentrale Schnittstelle besteht aus Komponenten der Kommunikationsdienste, wie dem *Bot Framework* und dem *FB Messenger*, für den Empfang, die Erstellung und das Senden von Nachrichten sowie der Speicherung von Entitäten, Zuständen und Dialogverläufen (*Storage*). Außerdem ist darin die domänenspezifische Logik (*Business Logic*) des Vorfalldmanagements implementiert.

Über die „*hears()*“-Schnittstelle des Botkits (im Diagramm als *Message* dargestellt) werden Nachrichten des Gesprächspartners an die jeweiligen NLU-Komponente und damit den NLU-Dienst gesendet. Der NLU-Dienst erkennt die Absichten und Entitäten, welche er dem Botkit wiederum über die NLU-Komponente samt weiteren Informationen zur Verfügung stellt. Die Geschäftslogik liest, verarbeitet, speichert und reagiert dann jeweils nutzerbezogen auf diese Daten. Zur Verarbeitung wird eine Normierung der Adresse mit der Google Maps API durchgeführt. Ebenfalls ist ein Datenaustausch mit dem CAFM denkbar, um weitere Informationen der Facilities, bereits eingegangene Meldungen sowie vertragliche Regelungen zu ermitteln. Die Klassifizierung der Vorfälle (siehe 4.2.7) wird durch das Limdu.js-Paket unterstützt. Hieraus wurde der ID3-Algorithmus verwendet, da die in Node.js verfügbaren C4.5-Implementierungen die Klassifizierung bei nicht trainierten Merkmalen abbrechen. Die Implementierungen im Limdu.js-Paket hingegen schränken bei Ungenauigkeit die Lösung auf mehrere Klassen ein, statt abzuberechnen.

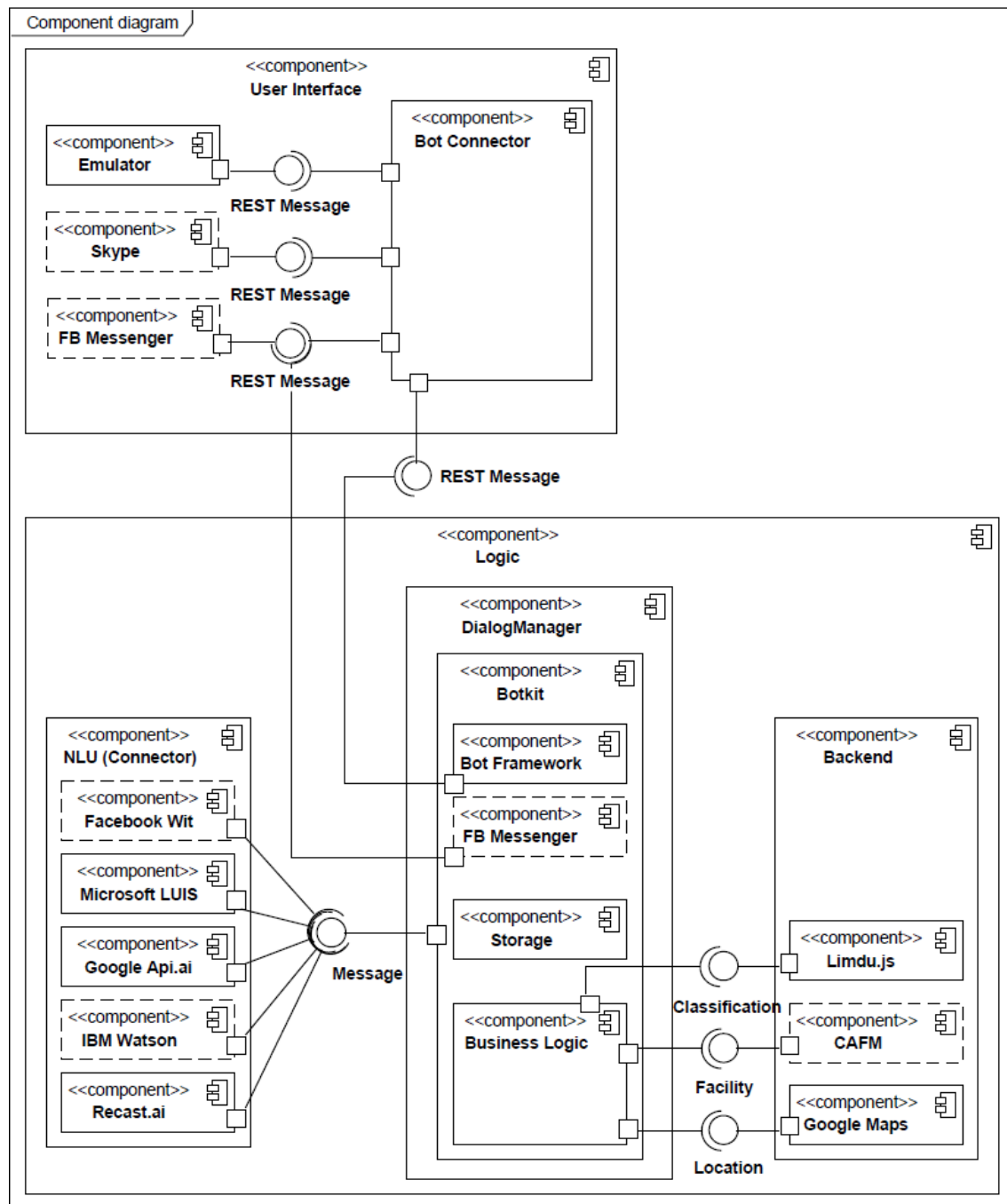


Abbildung 5.1: Komponentendiagramm

Das Komponentendiagramm zeigt zudem die vielfältigen Auswahlmöglichkeiten bei der Realisierung der Kommunikations- und NLU-Dienste. Die Implementierung ist nicht notwendigerweise an ein bestimmtes Unternehmen oder eine Organisation gebunden. Im Diagramm ist als Beispiel der Facebook Messenger genannt, womit das Bot Framework vollständig umgangen werden kann. Auf die Nennung anderer Optionen wurde lediglich aufgrund der Übersichtlichkeit verzichtet. Der Grund für die Auswahl des Bot Frame-

works ist die Möglichkeit, den Chat-Bot über den Emulator lokal und somit nicht öffentlich zu testen. Darüber hinaus bietet das Bot Framework umfangreiche Unterstützung von zusätzlichen Kommunikationsdiensten.

### 5.1.2 Pakete

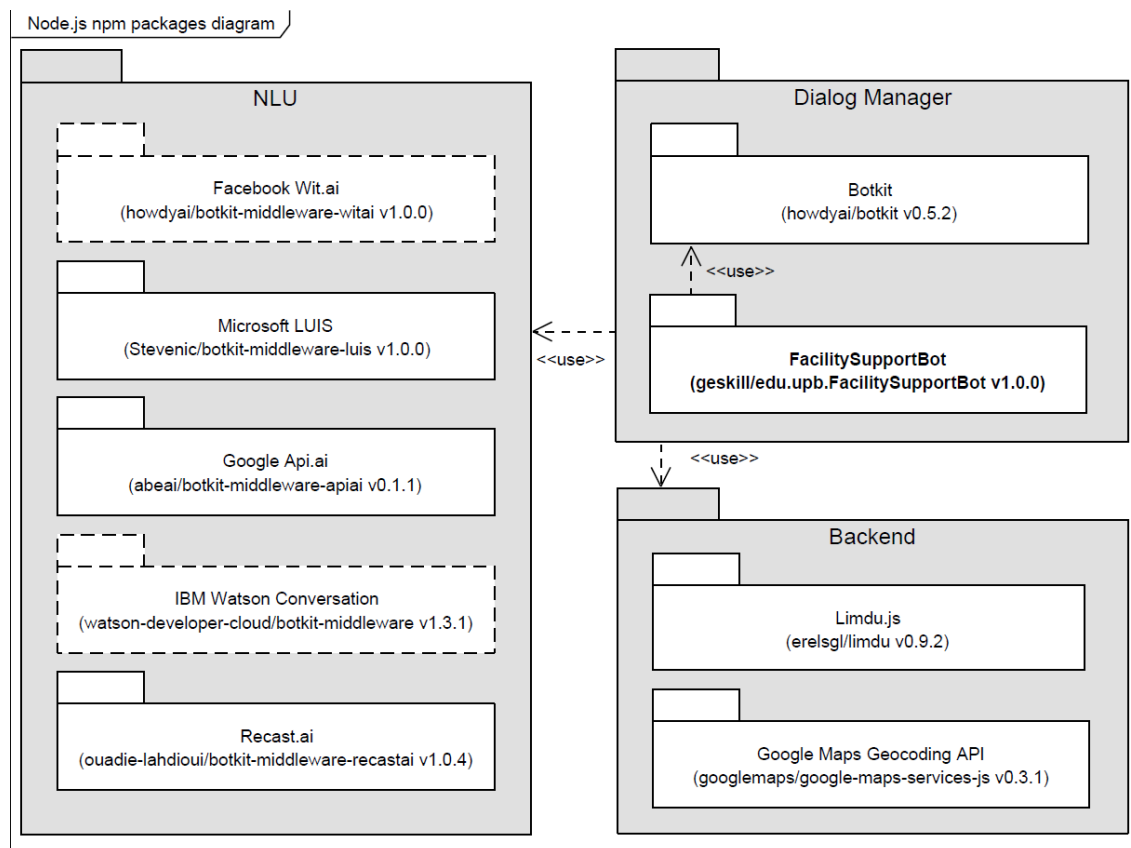


Abbildung 5.2: Paketdiagramm der Node.js Pakete

Die Node.js Architektur organisiert Funktionalität in Paketen. Die bereits vorgestellten Komponenten stammen aus diesen Paketen. Das in Abbildung 5.2 hervorgehobene Paket *FacilitySupportBot* realisiert die für diese Arbeit benötigte Funktionalität. Im Diagramm werden zudem die GitHub Verzeichnisse der Pakete und die verwendeten Versionsnummern aufgeführt.

### 5.1.3 Programme

Für die Nutzung des Chat-Bots wurden separate Programme erstellt, die jeweils ausschließlich die NLU-Schnittstellen sowie das Mapping zu einem Basismodul *BasicBot* implementieren. Das Basismodul beinhaltet jegliche Funktionalität, beispielsweise das Botkit. Abbildung 5.3 stellt diese Vorgehensweise grafisch dar.

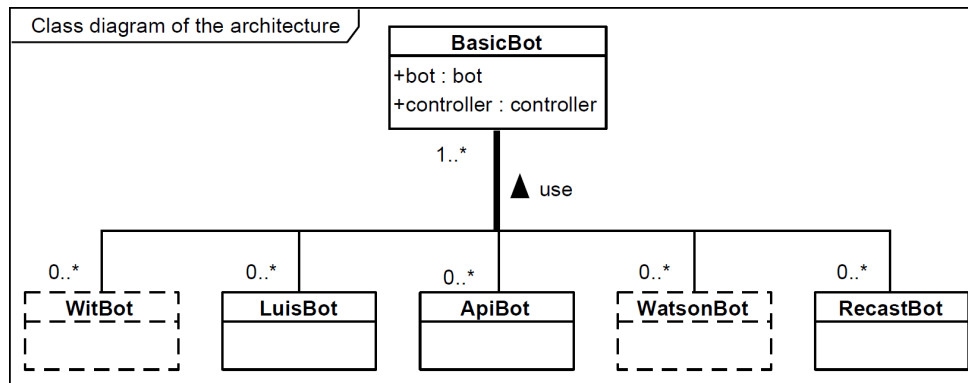


Abbildung 5.3: Klassendiagramm der Programme

Eine Alternative dazu stellt ein Programm dar, welches nach dem Start nach dem zu verwendenden NLU-Dienst fragt beziehungsweise diesen durch den parametrisierten Start festlegt.

## 5.2 Absichten und Entitäten

Die Aussagen des Nutzers enthalten zum einen Intentionen, die durch die Absichten und zum anderen Informationen, die mittels Entitäten erkannt werden. Dazu wurden im Lösungskonzept verschiedene NLU-Dienste und zu erfassende Informationen vorgestellt.

Die Realisierung der Absichten und Entitäten, insbesondere der Art der Entitäten, ist in den jeweiligen NLU-Diensten unterschiedlich. Ferner werden verschiedene Lösungswege innerhalb eines NLU-Dienstes zur Erkennung der Entitäten angeboten. Um die benötigten Entitäten bestmöglich zu erkennen, werden vor der vollständigen Implementierung verschiedene Lösungsmöglichkeiten bei den jeweiligen NLU-Diensten experimentell, durch je 20 aus dem Trainingsset, getestet. Sollte der NLU-Dienst keine vordefinierte Entität bereitstellen, so sind benutzerdefinierte Entitäten anzulegen. Um die Zuverlässigkeit letzterer zu prüfen, werden 20 weitere Vorfälle aus dem Testset genutzt. Dieses Vorgehen ist nötig, um zu unterscheiden, ob der NLU-Dienst nur die Entitäten aus dem Trainingsset erkennt oder ob durch die Lernfähigkeit bislang unbekannte Entitäten erkannt werden.

Dazu wird im Folgenden die Realisierung der Absichten mit den dazugehörigen Entitäten in den jeweiligen NLU-Diensten besprochen. Als Kennzahl für die Zuverlässigkeit wird eine Erkennungsrate von 95 Prozent benutzt.

### 5.2.1 Erfassung der Adresse

Für die Erkennung der Adresse wird die Stadt oder die Postleitzahl sowie die Straße und Hausnummer benötigt. Eine optionale zeitliche Angabe ist, wenn vorhanden, ebenfalls zu erfassen.

In Vorbereitung wurden Datensätze mit 2.060 deutschen Städtenamen<sup>45</sup> (auf Englisch) und 8185 deutschen Postleitzahlen<sup>46</sup> erstellt. Die vierstelligen Postleitzahlen wurden mit und ohne vorangestellte Null aufgenommen.

In LUIS gibt es keine vordefinierte Entität für Postleitzahlen. Ein erster Ansatz ist, die vordefinierte Entität für Zahlen zu nutzen, jedoch erfasst diese auch die Hausnummern, wodurch in einigen Fällen zwar die Hausnummer, aber nicht die Postleitzahl erkannt wird. Daher wird die benutzerdefinierte Entität („ZipCode“) für die Postleitzahl angelegt. Ohne weitere Spezifizierung werden unbekannte Postleitzahlen aus dem Testset bereits ausgemacht. Die benutzerdefinierten Entitäten können ferner durch die Verwendung einer Phrasenliste, einem Gazetteer, mit Informationen angereichert werden. Eine Phrasenliste, welche die Postleitzahlen repräsentiert, ist trotzdem nicht zielführend, da sie bei LUIS in der Kapazität beschränkt ist. Ebenfalls können reguläre Ausdrücke eine Entität näher spezifizieren. Doch scheint die nur in LUIS vorhandene Funktionalität keine erkennbare Verbesserung bei der Erkennung der Postleitzahlen zu zeigen. Daher wird auf die Spezifizierung der Entität mittels regulären Ausdrucks verzichtet, da ohnehin Gruppierungen in regulären Ausdrücken und damit die gezielte Auswahl von Textelementen nicht unterstützt werden. Diese Gruppierung wird aber benötigt, um Textgrenzen zu bestimmen, da ansonsten eine neunstellige Zahl eine vier- und fünfstellige Postleitzahl ausdrücken würde. Die Städte können durch eine vordefinierte Entität der Geografie („builtin.geography.city“) erkannt werden, jedoch ist diese nicht zuverlässig. Daher wird die benutzerdefinierte Entität Stadt („City“) erstellt. Für eine verlässliche Bestimmung der Stadt wird zudem die Phrasenliste mit den deutschen Städtenamen benötigt. In diesem Fall ist die Kapazität ausreichend. Für die Straße und Hausnummer kann eine zusammengesetzte Entität („StreetAndNumber“) aus zwei benutzerdefinierten Entitäten Straße und Nummer verwendet werden. Der Vorteil gegenüber einer Entität, welche letztere zusammenfasst, besteht nur darin, dass für die weitere Verwendung die Straße und Hausnummer bereits durch LUIS aufgeteilt wurden. Dagegen spricht, dass die Markierung der Entitäten in LUIS für die zusammengesetzte als auch jeweils für die einzelnen Entitäten durchgeführt werden muss und somit zeitaufwendiger ist. Funktionell führen beide Herangehensweisen zum gleichen Ergebnis im Kurztest. Anzumerken ist, dass die vordefinierte Entität Zahlen nicht für die Hausnummern verwendet werden kann, da Hausnummern mitunter Buchstaben, Binde- und Schrägstriche enthalten.

---

<sup>45</sup> Die deutschen Städtenamen wurden von der englischen Wikipedia Seite ausgelesen: [https://en.wikipedia.org/wiki/List\\_of\\_cities\\_and\\_towns\\_in\\_Germany](https://en.wikipedia.org/wiki/List_of_cities_and_towns_in_Germany).

<sup>46</sup> Die Postleitzahlen wurden von Andreas Manfrin 2006 aus der OpenGeoDB zusammengestellt und beinhalten keine Postfächer und Großkunden. Für diese Arbeit sind letztere aber nicht relevant: <http://www.manfrin-it.com/postleitzahlen/plz.html>. Auch der alte Datenbestand ist im Rahmen der prototypischen Implementierung zu vernachlässigen.



Bei Api.ai wurden die Städte, Postleitzahlen und Straßen im Test mit den vordefinierten Entitäten („@sys.geo-city“, „@sys.zip-code“ und „@sys.street-address“) verlässlich identifiziert.

Recast.ai bietet die vordefinierte zudem goldene Entität für Orte („LOCATION“) an, welche jedoch im Testlauf keine guten Ergebnisse erzielte. Deshalb wurden die vordefinierten Entitäten Stadtname („CITY\_NAME“) und Postleitzahl („ZIPCODE“) verwendet. Im folgenden Test wurden dennoch Städte und Postleitzahlen nicht zuverlässig erkannt, sodass auch hier der Datensatz hinterlegt wurde. Für die Erkennung von Straßen und Hausnummern im Test eignet sich die vordefinierte Entität („STREET“).

In Watson Conversation gibt es keine vordefinierte Entität für Städte und Postleitzahlen. Daher wurden entsprechende Entitäten angelegt. Aufgrund der fehlenden Funktionalität Entitäten in der Aussage zu markieren, wurden Städtenamen und Postleitzahlen erst erkannt, als die entsprechenden Datensätze hinterlegt wurden. Da Watson Conversation für die Straßen und Hausnummern ebenfalls keine vordefinierten Entitäten bereithält, ist die Erkennung problematisch. Der hinterlegte Algorithmus besinnt sich lediglich auf die hinterlegten Datensätze. Was bei den Städten und Postleitzahlen noch akzeptabel war, wird im Fall der Straße und Hausnummer schwierig. Auch der Test bestätigt, dass nur Straßen und Hausnummern erkannt wurden, bei denen die Informationen aus den Datensätzen exakt übereinstimmten. Daher ist die derzeitige Erkennung für die Straße und Hausnummer durch Watson Conversation für das weitere Vorgehen unbefriedigend, da nicht alle Straßennamen samt Kombinationen an Hausnummern in den Entitäten hinterlegt werden können.

Wit.ai gestattet die Erkennung von Orten über eine vordefinierte Entität („wit/location“). Allerdings wird angegeben, dass dies vornehmlich für US-amerikanische Adressen konzipiert wurde. Im Test wurden Städtenamen, Postleitzahlen, Straßennamen und Hausnummer jedoch nur teilweise erkannt und auch nur dann, wenn dies in einem bestimmten, vollständig zusammenhängenden Format angegeben wurden. Folglich wurden benutzerdefinierte Entitäten für den Stadtnamen, die Postleitzahl sowie die Straße und Hausnummer angelegt. Verglichen mit der Zuverlässigkeit von LUIS, welches die Postleitzahlen in einem ähnlichen Prinzip behandelt, schneidet Wit.ai schlechter ab. Zwar erlaubt Wit.ai die Definition von Gazetteers, jedoch können die Werte dazu nicht wie bei LUIS in einer strukturierten Datei hochgeladen werden. Das manuelle Eintippen ist sehr zeitaufwendig, weswegen sich die Erkennung der Städte und Postleitzahlen bei Wit.ai problematisch gestaltet. Wit.ai arbeitet jedoch an der Verbesserung (vgl. [WIT-3-ol]). Straßennamen samt Hausnummern wurden zuverlässig identifiziert.

Zusammenfassend liefern LUIS, Api.ai und Recast.ai zuverlässige Ergebnisse bei der Erkennung der Städtenamen, Postleitzahlen, Straßen und Hausnummer. Zwar bieten Api.ai und Recast.ai, im Gegensatz zu LUIS, eine vordefinierte Entität für die Straße und Hausnummer an, dafür erkennt LUIS diese aber auch mit benutzerdefinierten Entitäten verlässlich. Dagegen ist die Erkennung der Straßen bei Watson Conversation und die der

Städtenamen und Postleitzahlen bei Wit.ai unzureichend, weshalb beide für die weitere Betrachtung ausgeschlossen werden.

### 5.2.2 Erfassung des Geschehens und des lokalen Ortes

Die Erkennung des Geschehens mit der Referenz und dem Symptom, sowie dem lokalen Ort mit der Position, der Orientierung, dem Stockwerk und Raumnummer wurden in den NLU-Diensten, anders als die Angabe der Adresse, ähnlich realisiert.

In LUIS kann die bereits existierende Entität der Hausnummer („Number“) für die Raumnummer wiederverwendet werden. Die restlichen fünf der benötigten Entitäten werden durch benutzerdefinierte Entitäten angelegt („Reference“, „Symptom“, „Position“, „Orientation“ und „Floor“). Die Zuverlässigkeit nach der Klassifizierung der 20 Vorfälle aus dem Trainingsset war im Testset jedoch nicht befriedigend. Daher wurden Phrasenlisten für die Referenz, das Symptom, die Position sowie die Orientierungs- und Stockwerkanzeige hinzugefügt. Daraufhin verbesserte sich die Erkennung, auch bei bislang unbekannten Formulierungsvarianten aus dem Testset. Die Abbildung 5.4 stellt das Vorgehen exemplarisch an einem Beispiel dar. Da LUIS entweder die Werte oder die Variablennamen der Entitäten angibt, wurden in eine Fotomontage beide Darstellungen kombiniert.

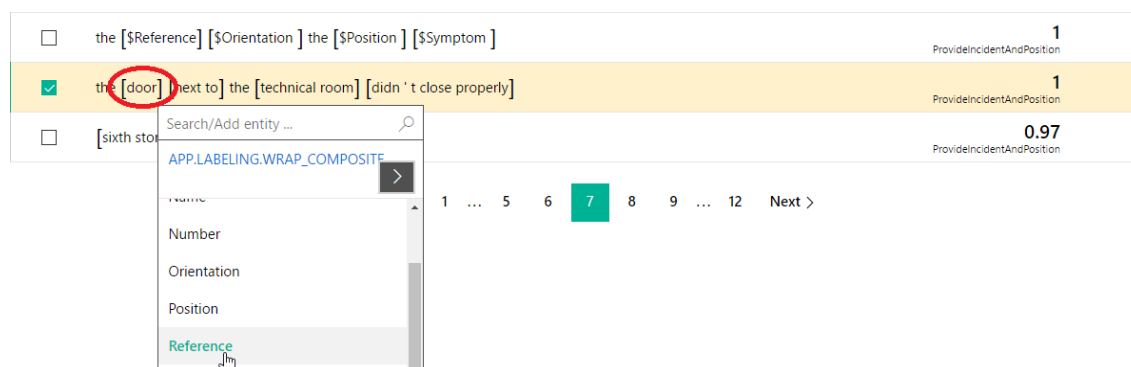


Abbildung 5.4: Zuweisung der Entitäten in LUIS

Bei Api.ai mussten sechs Entitäten angelegt werden („@Reference“, „@Symptom“, „@Position“, „@Orientation“, „@Floor“ und „@RoomNumber“). Abbildung 5.5 veranschaulicht die Zuweisung der Entitäten bei Api.ai.

PARAMETER NAME	ENTITY	RESOLVED VALUE	
Reference	@Reference	door	×
Orientation	@Orientation	next to	×
Position	@Position	technical room	×
Symptom	@Symptom	didn't close properly	×

Abbildung 5.5: Zuweisung der Entitäten in Api.ai

Die Entitäten in Recast.ai unterscheiden sich mit denen von Api.ai nur im Namen („REFERENCE“, „SYMPTOM“, „POSITION“, „ORIENTATION“, „FLOOR“ und „ROOM-NUMBER“). Abbildung 5.6 skizziert das Vorgehen bei Recast.ai.



Abbildung 5.6: Zuweisung der Entitäten in Recast.ai

### 5.2.3 Erfassung von Zeit, Dringlichkeit und persönlichen Daten

Die Erkennung von Zeit, Dringlichkeit und persönlichen Informationen ist optional für die Klassifizierung und Erstellung der Meldung zu einem Vorfall. Dennoch werden diese Informationen vom Chat-Bot erfasst, sollte der Meldende diese mitteilen.

Der Zeitpunkt oder die Zeitangabe des Vorfalls kann in LUIS („builtin.datetime“), Api.ai („@sys.date“ und „@sys.time“ beziehungsweise „@sys.time-period“ für Zeiträume) und Recast.ai („DATETIME“) jeweils mit vordefinierten Entitäten identifiziert werden.

Da die Dringlichkeit durch eine Ja-Nein-Frage mit vordefinierten Antwortmöglichkeiten erfasst wird, ist eine Identifizierung von Entitäten durch die NLU-Dienste nicht nötig, da die Antwort durch das Botkit interpretiert wird.

Die Erkennung der persönlichen Daten in LUIS wurde im ersten Schritt mit vordefinierten Entitäten getestet. Für den Namen wurde auf die vordefinierte Entität („Enzyklopädie“) zurückgegriffen. Bis auf die E-Mail-Adresse, in welcher es in den 20 Vorfällen nur zu 2 Fehlern kam, wurden die Informationen unzureichend erkannt. Daher wurde für den Namen eine neue Entität „Name“ angelegt und für die Telefonnummer die bereits existierende Entität „Number“ genutzt.

Bei Api.ai wurden die persönlichen Daten mit den vordefinierten Entitäten nach Markierung der Entitäten identifiziert („@sys.given-name“, „@sys.last-name“, „@sys.phone-number“ und „@sys.email“).

In Recast.ai wurde die E-Mail-Adresse mit einer vordefinierten goldenen Entität („EMAIL“) bestimmt. Name und Telefonnummer wurden hingegen nur teils mit den vordefinierten goldenen Entitäten („PERSON“ und „PHONE“) erkannt. In den Fällen, in denen diese nicht identifiziert wurden, wurde auf die vordefinierten Entitäten („NAME“ und „PHONENUMBER“) zurückgegriffen.

### 5.3 Abläufe und Zustände

In diesem Kapitel werden die Abläufe und Zustände vom Chat-Bot erläutert. Hierbei liegt der Fokus auf dem Ablauf zur Erkennung der Adresse und der Klassifizierung des Vorfalls.

Die Zustände im Dialogverlauf entsprechen denen, die bereits mit Abbildung 4.1 dargestellt wurden. Diese W-Informationen werden nacheinander durch den Chat-Bot vom Nutzer abgefragt. Der Nutzer seinerseits hat jedoch immer die Möglichkeit, mit der Texteingabe eine andere Information bereitzustellen. Somit können jederzeit Informationen ergänzt oder korrigiert werden.

Die NLU-Dienste ordnen die Eingabe des Meldenden durch die Erkennung der Absicht einer dieser Informationsgruppen zu. Wenn die Absicht nicht erkannt wird, die Wahrscheinlichkeit bei der Erkennung der Absicht zu gering ist oder der Meldende vorsätzlich nicht relevante Informationen übermittelt, greift die Standardabsicht.

Sind alle benötigten Daten einer Informationsgruppe erfasst, so wird eine Standardantwort ausgeführt. Diese wählt basierend auf dem aktuellen Informationsstand und dem definierten Dialogfluss die jeweils nächste relevante Frage. Die Standardantwort wird auch durch die Standardabsicht ausgelöst, jedoch mit einem Parameter, der die fehlerhafte oder problematische Erkennung markiert. Ein interner Zählerstand zeichnet dies auf. Bei der zweiten falschen Erkennung folgt ein Hinweis. Sobald mehr als zwei Falsche vorliegen, wird dem Meldenden angeboten, sich direkt mit einem Servicemitarbeiter zu verbinden.

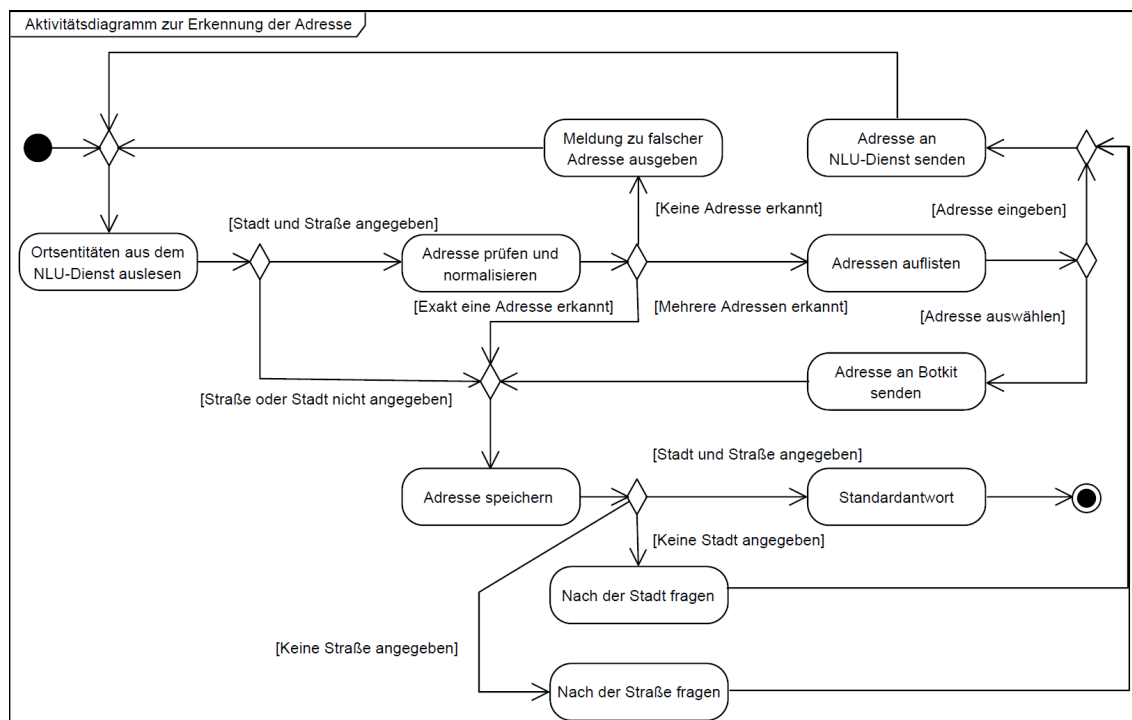


Abbildung 5.7: Aktivitätsdiagramm zur Erkennung der Adresse

Abbildung 5.7 stellt die Aktivitäten im Rahmen der **Erkennung der Adresse** dar. Beginnend mit der Eingabe des Meldenden, die durch den NLU-Dienst als Mitteilung der Adresse klassifiziert wird, folgt die Prüfung, ob die Entitäten Stadtname beziehungsweise Postleitzahl und Straßename inklusive Hausnummer vorliegen. Diese ergeben sich entweder aus den erkannten Entitäten vom NLU-Dienst oder sind im Speicher auf Grund einer vorherigen Verarbeitung vorhanden.

Sollte ausschließlich der Stadt- oder Straßename zur Verfügung stehen, wird in einer Nachricht an den Nutzer um eine Bereitstellung der fehlenden Daten gebeten. Sind beide Informationen vorhanden, wird über die Google Maps API die Adresse auf ihre Existenz geprüft und eine Normierung durchgeführt.

Falls der Kartendienst die Adresse nicht erkennt, erfolgt eine Rückfrage mit Bitte um Korrektur. Für den Fall, dass die Adresse Mehrdeutigkeiten aufweist, werden diese im Dialogfenster als Schnellauswahlmöglichkeiten bereitgestellt.

Geht aus der Anfrage des Kartendienstes nur eine Adresse hervor beziehungsweise wählt der Benutzer diese aus, wird nach deren Speicherung schlussendlich die Aktivität der Standardantwort durchgeführt.

Die Vorgehensweise zur **Erkennung und Klassifikation des Vorfalls** wird in Abbildung 5.8 skizziert. Den Ausgangspunkt stellt das Lesen der durch den NLU-Dienst identifizierte Entitäten dar. Nach Speicherung dieser wird geprüft, ob erforderliche Informationen vorliegen.

Sollten die Referenz und das Symptom fehlen, wird eine Rückfrage nach dem Geschehen gestellt. Fehlt die lokale Ortsangabe, wird entsprechend nach dieser gefragt. Sollte die lokale Ortsangabe eine allgemeine Ortsangabe darstellen, wie einen bestimmten Raum oder ein bestimmtes Büro, so wird eine Rückfrage nach einer exakten Positionsbezeichnung, der Raumnummer gestellt. Liegt eine einschränkende Ortsbezeichnung vor, wird nach der Orientierung oder dem Stockwerk gefragt.

Sind alle Daten vorhanden, wird der Vorfall in keine, eine oder mehrere Klassen des Vorfalltyps klassifiziert. Wenn kein Vorfalltyp erkannt wird, wird dem Meldenden eine Schnellauswahl angeboten. Diese ermöglicht, den Vorfall pauschal als „Defekt“ zu deklarieren, sich mit einem Supportmitarbeiter zu verbinden. Ebenfalls können hinterlegte Angaben zum Vorfall aus dem System entfernt werden. Diese Option soll dem Benutzer helfen, den Vorfall erneut durch eine veränderte Wortwahl zu schildern. Der Löschvorgang stellt dabei sicher, dass die Referenz und das Symptom explizit durch das System verlangt werden.

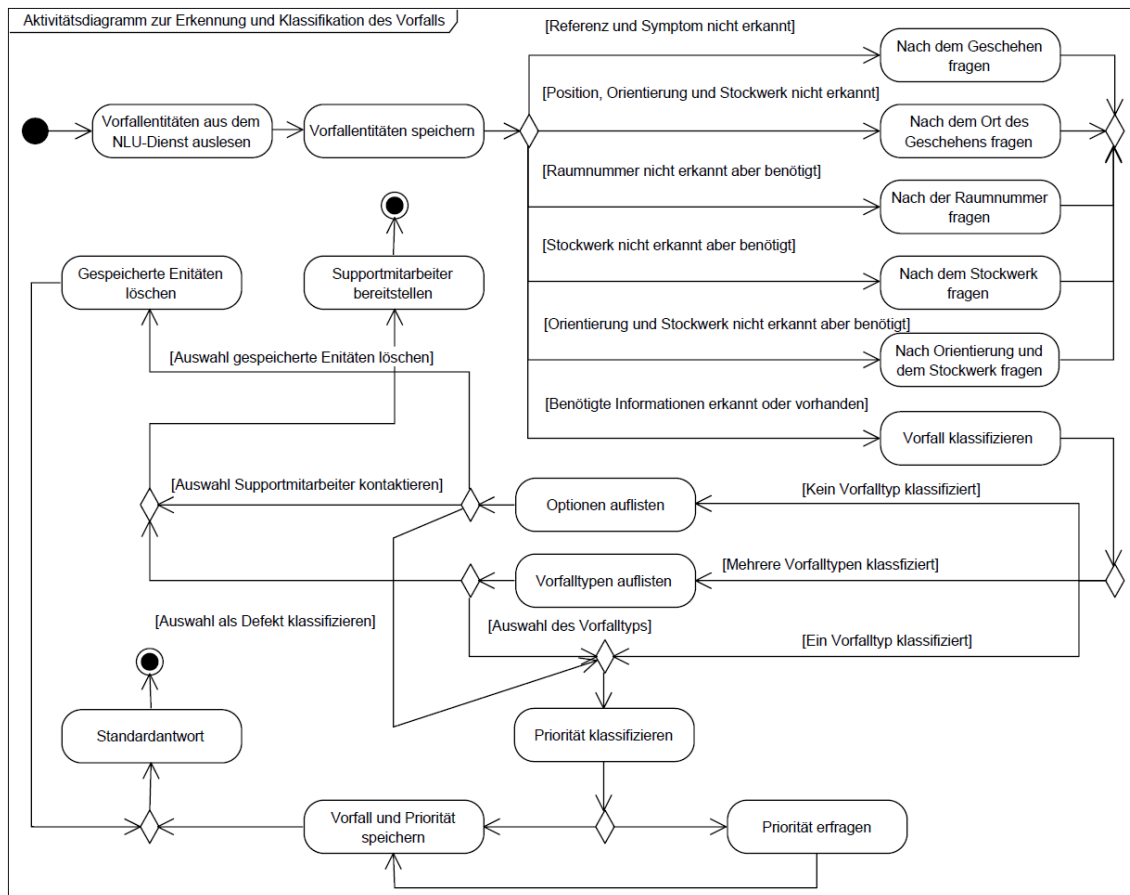


Abbildung 5.8: Aktivitätsdiagramm zur Erkennung und Klassifikation des Vorfalls

Resultieren aus der Klassifizierung mehrere Vorfalltypen, so werden diese dem Nutzer als Auswahlmöglichkeiten vorgelegt. Zusätzlich wird die Option des Telefonsupports angeboten, da ein Wegfallen der korrekten Option nicht auszuschließen ist.

Nachdem ein Vorfalltyp beispielsweise durch die Nutzerauswahl bestimmt wurde, folgt die Klassifikation der Priorität. Sollte das Ergebnis dieser nicht eindeutig sein, wird dem Meldenden eine Ja-Nein-Rückfrage gestellt, ob es sich um ein zeitkritisches Problem handelt.

Abschließend folgt der Aufruf der Standardantwort.

## 5.4 Datenmodell und Datenverarbeitung

Das folgende Kapitel listet ein Klassendiagramm, welches im Unterschied zu Abbildung 4.3 die Spezifizierung der verwendeten Variablen, dessen Typen und Kapselung enthält. Im Weiteren wird die Datenverarbeitung erörtert.

Eine neue Konversation startet, indem ein Nutzer sich mit dem Chat-Bot verbindet. Dabei wird eine neue Instanz der Klasse „User“ erstellt. Darin wird mit einem Zeitstempel der Beginn sowie die Anzahl der Fehlversuche bei der Absichtenerkennung gespeichert. Die Speicherung erfolgt innerhalb der von Botkit bereitgestellten „Storage“-Funktionalität<sup>47</sup> nutzergebunden.

Folglich werden im Gesprächsverlauf benötigte Informationen erfasst und in die jeweiligen Klassen, skizziert in Abbildung 5.9 gespeichert. Der Vorfall („Incident“), die Adresse („Address“), die Dinglichkeit sowie der Kontakt („Contact“) bilden eine Gliederungsebene. Die Kontakt-Klasse wird zudem durch die von Botkit weitergeleitete Information des jeweiligen implementierenden Frameworks, in diesem Fall das Bot Framework, ergänzt.<sup>48</sup>

Die Datenstrukturen der eingehenden Nachrichten, die der NLU-Dienste für die Bereitstellung der erkannten Entitäten sowie die für die Generierung der Nachrichten hängen vom Dienstleister ab. Entsprechend mussten diese, beispielsweise bei den NLU-Diensten, durch ein Mapping auf ein einheitliches Schema angepasst werden.

---

<sup>47</sup> Die gespeicherten Informationen werden nur zur Laufzeit gespeichert. Eine permanente Speicherung ist möglich, im Rahmen dieses Projekts aber nicht nötig.

<sup>48</sup> Beispielsweise ist damit der Zugriff auf den öffentlichen Namen in Skype möglich. Personenbezogene Daten sind aus Datenschutzgründen nicht zugänglich. Es ist aber denkbar, dass zukünftige Versionen des Bot Frameworks eine Anfrage ermöglichen, welche, nach Bestätigung dieser, die personenbezogenen Daten zugänglich macht.

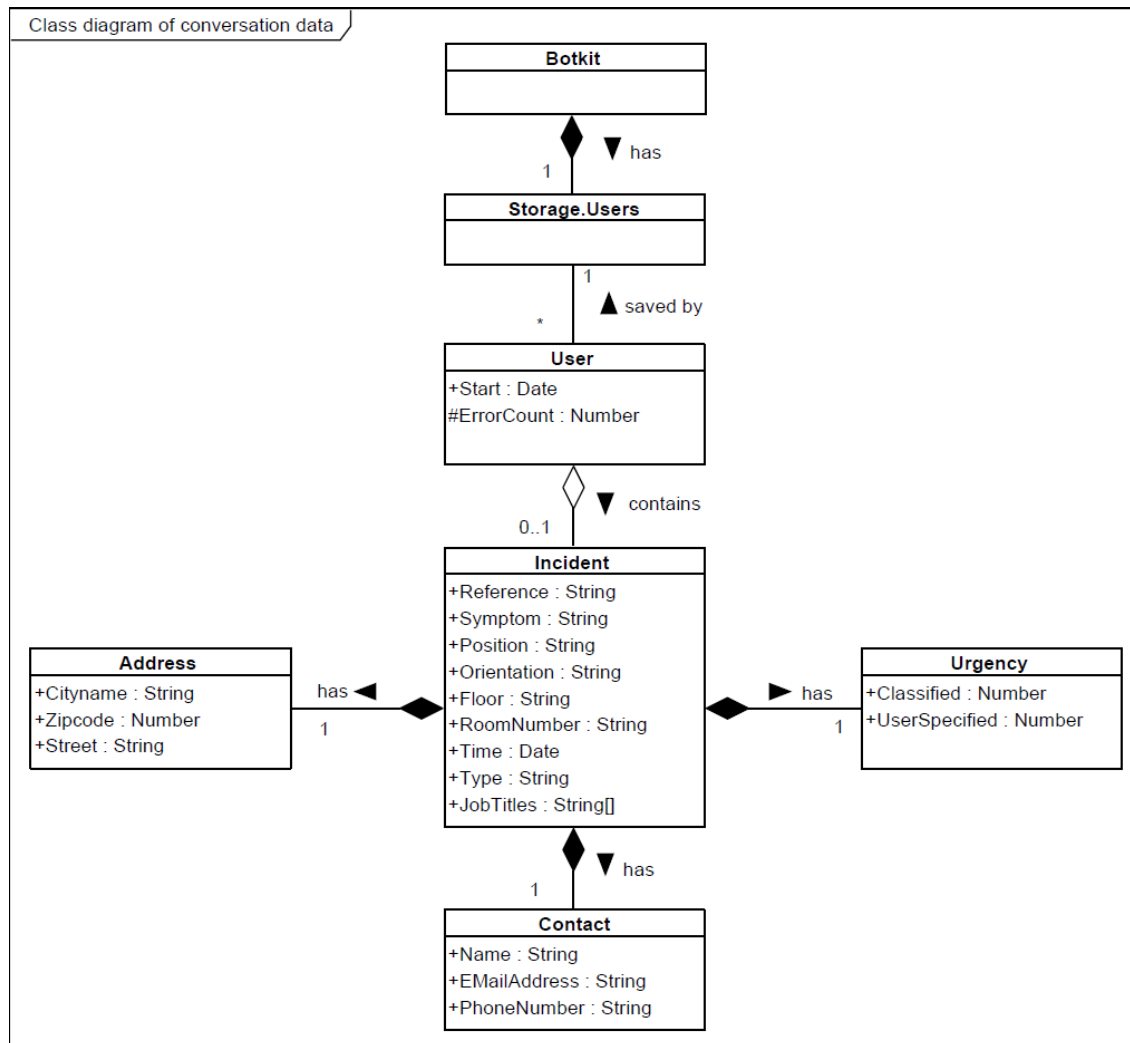


Abbildung 5.9: Klassendiagramm der Konversationsdaten



## 5.5 Benutzeroberfläche

Das Kapitel stellt die Benutzeroberfläche des Chat-Bots dar. Wie bereits erläutert wird dazu der Emulator genutzt. Die Oberfläche unterscheidet sich von Skype oder einem anderen Kommunikationsdienst nur insoweit, dass in der oberen Zeile und rechts Details über die internen Kommunikationsprotokolle dargestellt werden.

Die Abbildung 5.10 skizziert, die Erfassung der Geschäftsadresse. Hierbei stellt das System fest, dass die angegebene Stadt und Straße keine gültige Kombination ist. Daraufhin werden Alternativen aufgezeigt. In diesem Fall wählt der Benutzer keine Alternative, sondern gibt die Straße erneut ein, welche das System daraufhin akzeptiert.

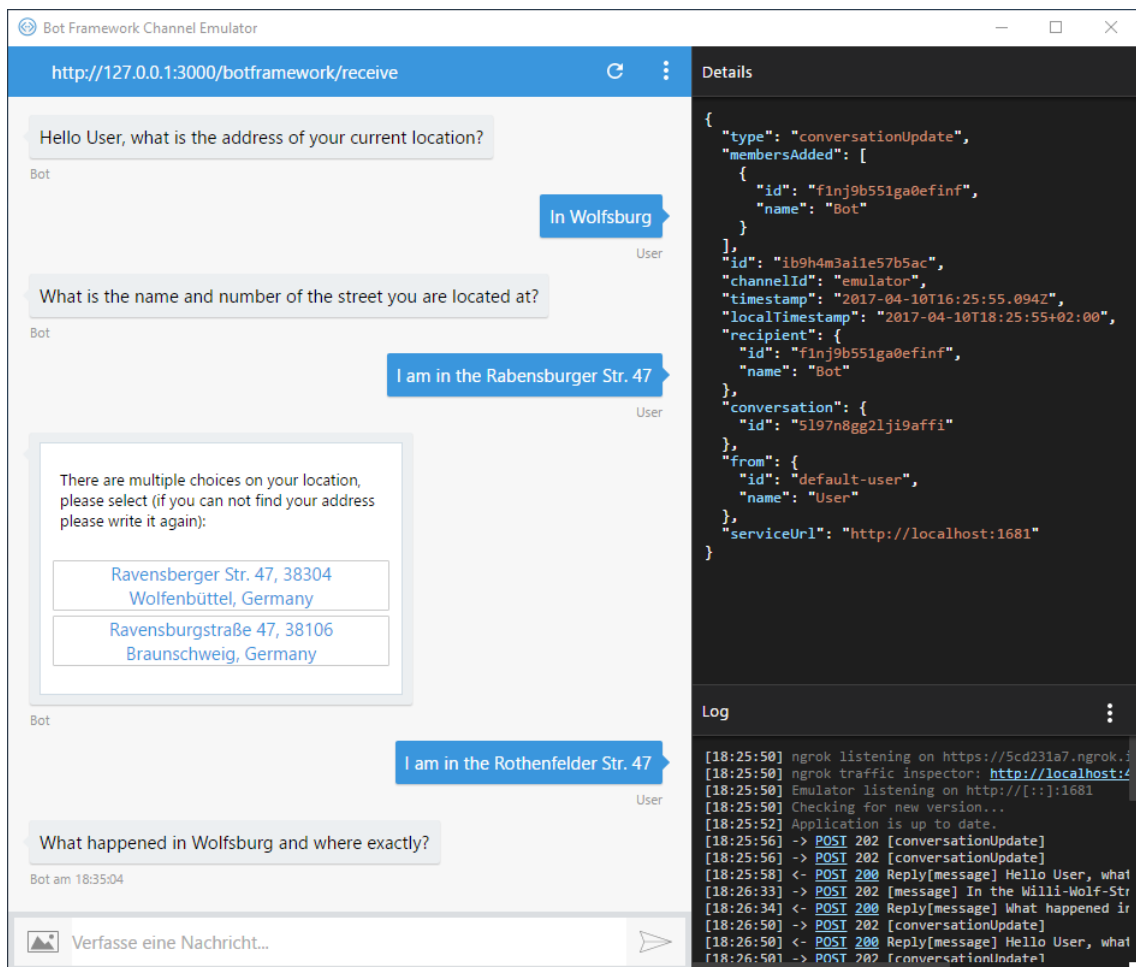


Abbildung 5.10: Beispiel der Meldung eines Vorfalls im Emulator (1)

Darüber hinaus zeigt Abbildung 5.11 den Ablauf, wenn alle Daten erfasst werden, sodass das System eine Klassifizierung des Vorfalltyps und der Priorität durchführt. Entsprechend bekommt der Meldende die Information, dass das Sicherheitspersonal informiert wurde und sich umgehend um die Problemlösung kümmert. Das Verhalten, sollte der Nutzer nicht auf die Fragen des Chat-Bots antworten sowie Mehrdeutigkeiten bei der Klassifizierung des Vorfalltyps, werden in Abbildung 5.12 visualisiert.

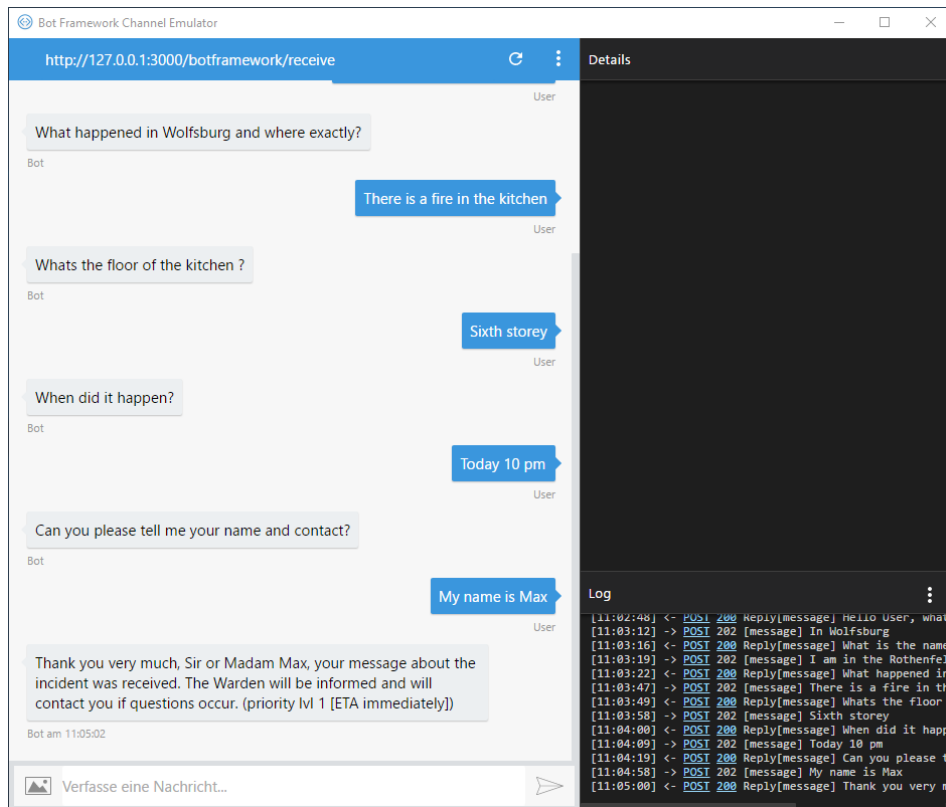


Abbildung 5.11: Beispiel der Meldung eines Vorfalls im Emulator (2)

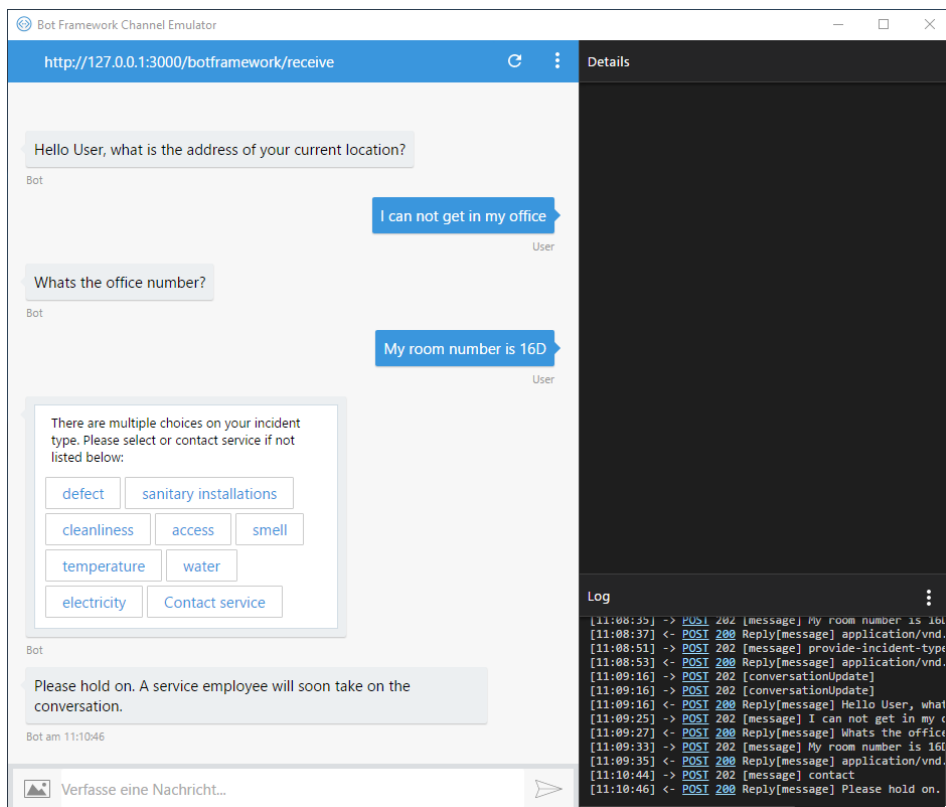


Abbildung 5.12: Beispiel der Meldung eines Vorfalls im Emulator (3)

## 6 Evaluation

Im Folgenden werden die Natürlichsprachlichkeit, die verwendeten NLU-Dienste sowie die Klassifikation des Vorfalltyps und der Priorität evaluiert.

Diese drei Merkmale stellen die wesentlichen Qualitätseigenschaften des Chat-Bots dar. Im Rahmen dieser Arbeit war es nicht möglich, den Chat-Bot als komplettes System zu evaluieren, da kein vergleichbares Programm in der Literatur oder Praxis existiert.

Alternativ kann mit der Aktivität von einem oder mehreren Mitarbeitern verglichen werden. Jedoch ist dies nicht nur aufwendig, sondern auch unnötig, da anhand der genannten Merkmale die Qualität bereits ausreichend abzusehen ist, um eine erste Einschätzung zu treffen. Je nach Resultat ist ein Vergleich mit Mitarbeitern immer noch möglich.

### 6.1 Natürlichsprachlichkeit

Anhand der von Berg erarbeiteten Eigenschaften zum Leistungsumfang natürlichsprachlicher Dialogsysteme soll die Natürlichsprachlichkeit des Chat-Bots bewertet werden.

Die **gemischte Initiative** findet Anwendung, obwohl eine interne Reihenfolge der Informationsbeschaffung existiert, da der Nutzer dem Chat-bot die Informationen in beliebiger Reihenfolge mitteilen kann. Dieser passt sich an und sammelt dazugehörige, für den Vorfall relevante Informationen.

**Implizit bestätigt** wird im Fall der Stadt, der Position und des Namens. Bei anderen Entitäten erfolgt dies nicht, da sie nicht immer vorhanden sind oder benötigt werden, um den Vorfall aufzunehmen.

Eine Komponente für die **Sprachgenerierung** wurde nicht implementiert. Daher war die implizite Bestätigung aufgrund der fehlenden syntaktischen Regeln im Fall des Symptoms, welches Wörter beinhalten kann, die je nach Satzstruktur angepasst werden müssen, natürlichsprachlich nicht realisierbar.

Zwar werden **Nachfragen**, die anhand der griceschen Maximen formuliert wurden, gestellt, jedoch um fehlende Informationen einzuholen oder ungenaue zu spezifizieren, nicht um sprachliche Mehrdeutigkeiten zu resolutionieren. Letztere Funktionalität wurde aufgrund der Domänenspezifität nicht implementiert.

Eine **Korrektur** von Informationen ist jederzeit möglich. Die **Überbeantwortung** von Aussagen nur dann, wenn dies nicht mit einer Absichtenkollision und wesentlichen Änderung der Satzstruktur einhergeht. Das System ist dazu konzipiert, die genau abgegrenzten Informationstypen, die W-Fragen, zu ermitteln. Folglich existieren in den NLU-Diensten jeweils Absichten für diese Informationstypen. Nennt der Meldende beispielsweise sowohl die Geschäftsadresse als auch den Vorfall, so wird der NLU-Dienst sich für

eine der beiden Absichten entscheiden. Demnach wird auch nur eine Information aufgenommen. Wird bei der Nennung der Geschäftsadresse jedoch ausschließlich eine zeitliche Information übermittelt, dann wird diese miterfasst, da diese die Satzstruktur, welche für die Erkennung der Absicht analysiert wird, nicht wesentlich verändert.

**Verneinungen** und die **Interpretation von Umgangssprache** standen nicht im Fokus dieser Arbeit. Zum einen sind Verneinungen durch die präzise formulierten Fragenstellungen und den domänenspezifischen Bereich von untergeordneter Wichtigkeit und zum anderen sind diese über die NER und eine zusätzliche POST-Analyse erkennbar. Die Interpretation von Umgangssprache ist, aufgrund der Lernfähigkeit durch Hinzufügen von Formulierungen, nicht ausgeschlossen.

Auch wurde die **Adaptivität** nicht umgesetzt, da die Anzahl möglicher Konfigurationsparameter und die Nutzungsfrequenz pro Meldenden gering ausfällt.

Das **Kontextwissen** wird nur verwendet, um die nächste Fragestellung zu wählen oder um Auswertungen zum Vorfall durchzuführen, nicht aber für die Absichtenanalyse des Gesagten. Dies ist in der fehlenden Funktionalität von Botkit begründet (siehe unten).

Die Eigenschaften des **Sozialverhaltens** sowie die der **Spracherkennung und Sprachsynthese** waren nicht Bestandteil dieser Arbeit.

## 6.2 NLU-Dienste

Für den Test wurden 80 Vorfälle aus dem Vorfallkatalog entnommen, um die drei NLU-Dienste zu trainieren. Dazu wurden Absichten und Entitäten, welche das Geschehen und weitere vorfallrelevante Daten erfassen, wie in Kapitel 5.2 beschrieben, angelegt. Die 40 verbliebenen Vorfälle wurden genutzt, um den jeweiligen NLU-Dienst zu testen. Getestet wurde ausschließlich die Erfassung des Vorfalls inklusive der lokalen Ortsangabe, da diese bei allen NLU-Dienst unter den gleichen Bedingungen (mit Ausnahme der Raumnummer) implementiert wurde. Die Erkennung der Adresse, des Zeitpunkts und der persönlichen Informationen wird in den NLU-Diensten unterschiedlich implementiert, weshalb die Testergebnisse nicht vergleichbar wären. Für die Zusammenstellung und automatisierte Berechnung der Ergebnisse wurde ein Programm erstellt und liegt dem Projektverzeichnis bei.

Die Testergebnisse (siehe Tabelle 6.1) stellen die Auswertung der erkannten Entitäten aus den drei NLU-Diensten dar. Die linke Spalte der Tabelle beziffert die Genauigkeit bei der Erkennung. Die Eins markiert die korrekte Erfassung der Entität, die Zwei eine maximale Abweichung von zwei basierend auf der Levenshtein-Distanz. Die Spalte Drei bis Fünf meint, dass maximal eine Abweichung von fünf existiert. Darüber hinaus gibt die Zeile mit dem Sternchen alle Abweichungen oder eine nicht erkannte Entität an.

In LUIS wird die Raumnummer mit der gleichen Entität wie die der Hausnummer erfasst. Folglich hat diese mehr Trainingsdaten, wodurch sich die besseren Ergebnisse im Vergleich zu Api.ai und Recast.ai erklären lassen. Jedoch sind die Ergebnisse bei den Entitäten der Referenz, dem Symptom, der Position und des Stockwerks bei LUIS schlechter als bei Api.ai und Recast.ai. Api.ai ist bei der Erfassung des Symptoms um zwei erkannte Entitäten besser. Hingegen ist Recast.ai bei der Referenz und Position jeweils um eine erkannte Entität besser. Demnach sind die beiden in diesem Test als vergleichbar gut anzusehen, wohingegen LUIS schlechter abschneidet.

*Tabelle 6.1: Vergleich der erkannten Entitäten*

	Referenz	Symptom	Position	Orientierung	Stockwerk	Raumnummer
<b>LUIS</b>						
1	7	7	12	0	5	1
2	0	0	0	0	0	2
3-5	1	0	2	0	0	1
*	15	32	19	3	6	1
<b>Api.ai</b>						
1	9	15	15	0	9	0
2	0	0	0	0	0	0
3-5	2	0	1	0	0	0
*	12	24	17	3	2	5
<b>Recast.ai</b>						
1	10	13	16	0	8	0
2	0	0	0	0	0	0
3-5	0	0	0	0	0	0

*	13	26	17	3	3	5
---	----	----	----	---	---	---

Das gewählte Trainings- und Testset ist mit 120 Vorfällen umfangreich genug, um maschinelles Lernen durchzuführen, jedoch würde ein größerer Datensatz mit vielzähligen Formulierungsvarianten die Zuverlässigkeit erhöhen.

Die Fachtermini der Orte, Bereiche und Räume sowie die Symptome in dem Vorfallkatalog decken nur einen Teil der Vorfälle im Facility-Management ab. Durch die Erstellung spezifischer Listen mit Fachtermini und Redewendungen wäre es möglich, die Erkennung des Geschehens zu verbessern.

Der NLU-Dienst bekam keine Informationen über die Fragestellung, auf welche der Benutzer antwortete. Die Aussage des Nutzers wurde allein durch die Algorithmen des maschinellen Lernens klassifiziert. Api.ai und Recast.ai bieten innerhalb des Frameworks durch Kontextvariablen die Möglichkeit, die in Frage kommenden Absichten einzuschränken oder zu priorisieren. Diese Möglichkeit wird durch Botkit nicht unterstützt, könnte aber ergänzt werden. Allerdings wären die Testergebnisse nicht vergleichbar, da LUIS diese Option nicht anbietet.

### 6.3 Klassifikation

Die Evaluation der Klassifikation wurde ebenfalls mit einem separaten Programm durchgeführt. Hierbei wurde ebenso mit den 80 Vorfällen trainiert und mit den verbleibenden 40 Vorfällen getestet. Zudem wurde der im Chat-Bot verwendete Algorithmus für Entscheidungsbäume ID3 mit dem Winnow und C4.5-Algorithmus verglichen.

Für die Klassifikation des Vorfalltyps und der Priorität wurde das in 4.2.7 beschriebene Standardverhalten (*S*) für die Auswahl der Merkmale geprüft. Sollte der Wert der Referenz nicht vorliegen, so wird der Positionswert als Merkmal genutzt. Analog dazu wird für die Priorität bei nicht vorhandener Positionsangabe der Wert der Referenz übernommen. Die Varianten *V1* und *V2* implementieren ein alternatives Verhalten. Erstere verwendet immer alle drei Angaben (Referenz, Symptom und Position). Letztere nutzt beim Vorfalltyp nur die Referenz und das Symptom. Analog werden bei der Priorität nur das Symptom und die Positionsangabe genutzt.

Da die Algorithmen keine binäre Klassifikation durchführen, sind keine oder mehrere Ergebnisse möglich. Daher wurde in der Auswertung die Güte der Klassifikation beachtet und folgend gesondert gelistet.

Die Ergebnisse zum Vorfalltyp (siehe Tabelle 6.2) zeigen keine signifikanten Unterschiede in der Klassifikation des korrekten Wertes zwischen dem ID3 und Winnow-Algorithmus auf. Die Standardabweichung beträgt 0,5 zwischen den beiden Verfahren. Der ID3-Algorithmus unterscheidet sich dennoch von dem Winnow-Algorithmus, da die Vorfalltypen der Wahrscheinlichkeit nach auch als zweites beziehungsweise dritt bis fünftes

klassifiziert wurden. Beim Winnow-Algorithmus wurden diese hauptsächlich nicht klassifiziert (6-\*). Da von der Eigenschaft, bei der dem Meldenden bei Mehrdeutigkeiten eine Liste an Auswahlmöglichkeiten angeboten wird, im Chat-Bot Gebrauch gemacht wird, ist der ID3-Algorithmus dem Winnow-Algorithmus vorzuziehen. Der C4.5-Algorithmus stellt zwar eine Verbesserung zum ID3-Algorithmus dar, stoppt jedoch in der Node.js-Implementierung die Klassifizierung bei nicht trainierten Merkmalen, statt den Vorfall mit vorhandenen Merkmalen einzuschränken.

*Tabelle 6.2: Vergleich der Vorfalltyp-Klassifikation*

	<b>ID3</b>			<b>Winnow</b>			<b>C4.5</b>
	<b>S</b>	<b>V1</b>	<b>V2</b>	<b>S</b>	<b>V1</b>	<b>V2</b>	<b>S</b>
1	9	9	9	10	8	8	7
2	3	3	3	0	1	0	0
3-5	9	13	10	0	0	0	0
6-*	19	15	18	30	31	32	33

Das Resultat der Klassifikation der Priorität (siehe Tabelle 6.3) macht ebenfalls keine signifikanten Unterschiede in der Klassifikation des korrekten Wertes zwischen dem verwendeten ID3 und Winnow-Algorithmus deutlich. Auch bei den getesteten Varianten lassen sich keine signifikanten Unterschiede herausstellen. Ebenso listet der ID3-Algorithmus bei der Priorität der Wahrscheinlichkeit nach den als zweiten klassifizierten Wert. Aufgrund der vier Klassifikationsklassen ist eine weitere Aufteilung jedoch nicht sinnvoll. Die schlechteren Ergebnisse des C4.5-Algorithmus sind im Vergleich zu den anderen Algorithmen auch auf die Node.js Implementierung zurückzuführen.

*Tabelle 6.3: Vergleich der Priorität-Klassifikation*

	<b>ID3</b>			<b>Winnow</b>			<b>C4.5</b>
	<b>S</b>	<b>V1</b>	<b>V2</b>	<b>S</b>	<b>V1</b>	<b>V2</b>	<b>S</b>
1	10	10	10	10	11	9	5
2	7	8	8	0	1	0	0
3-*	23	22	22	30	28	31	35

Die Folgerung aus dem vorherigen Kapitel, ein umfangreicheres Trainingsset verbessere die Ergebnisse, lässt sich ebenfalls auf die Klassifikation übertragen. Insbesondere unbekannte, im Trainingsset nicht gelernte Begrifflichkeiten waren für die Algorithmen kaum oder nicht korrekt klassifizierbar.



## 7 Fazit und Ausblick

Das Ziel dieser Arbeit war es, einen Chat-Bot zu entwickeln, der die Not- und Störfälle im Facility-Management annimmt, verarbeitet und interaktiv einschränkt. Anhand diesem sollte geprüft werden, ob ein Dialogsystem eine praktikable Ergänzung zum Telefonsupport darstellt. Dies könnte den Arbeitstag entlasten und die Schulung der Mitarbeiter fördern und folglich zur systematischen Verbesserung der Qualität beitragen.

Für die Umsetzung wurden verschiedene Frameworks, die auf maschinellem Lernen basieren, für die Annahme der Vorfälle und Verarbeitung natürlicher Sprache sowie Algorithmen für die Einschränkung untersucht. Zugleich wurde ein Vorkatalog erstellt, um basierend auf diesem ein Datenmodell für die strukturierte Erfassung der Vorfälle zu entwickeln. Ebenfalls wurden daraus Arten an Vorfällen erarbeitet, um das Geschehen über die Klassifikation einzuschränken.

Die Natürlichsprachlichkeit wurde anhand der von Berg erarbeiteten Eigenschaften zum Leistungsumfang natürlichsprachlicher Dialogsysteme qualitativ bewertet. Zusammenfassend wurden für den domänenspezifischen Bereich relevante Merkmale, wie die gemischte Initiative oder die Korrektur von Aussagen, implementiert und auf andere, beispielsweise ein Sozialverhalten, verzichtet. Kritisch anzumerken ist, dass die Unterstützung von Kontexten, die in der fehlenden Funktionalität von Botkit begründet ist, fehlt. Eine Komponente für die Sprachgenerierung sowie ein auditives Dialogsystem zu entwickeln beziehungsweise zu nutzen war nicht Bestandteil dieser Arbeit, weshalb diese Leistungsmerkmale keine Relevanz aufzeigen.

Die NLU-Dienste ebenso wie die Klassifikation der Art der Vorfälle und Priorität wurden mithilfe des Vorkatalogs evaluiert. Dazu wurden 80 Vorfälle entnommen, um die NLU-Dienste beziehungsweise die Klassifikation zu trainieren. Anhand der restlichen 40 Vorfälle wurden die NLU-Dienste beziehungsweise die Verfahren getestet.

Bei den NLU-Diensten wurde ausschließlich die Erfassung des Vorfalls verglichen. Die dazugehörigen Entitäten Referenz, Symptom und Position wurden von Google (Api.ai) und Recast.ai im Mittel zu 41,4 Prozent erkannt. Zwischen diesen beiden Diensten gab es kaum erkennbare Unterschiede. Dagegen erkannte Microsoft (LUIS) bei der Referenz 2,5, bei dem Symptom 7 und bei der Position 3,5 Entitäten weniger als im Mittel bei Api.ai und Recast.ai, und kommt im Mittel auf eine Erkennung von 28,3 Prozent.

Bei der Klassifikation des Vorfalltyps wurde die korrekte Art des Vorfalls im Standardverhalten zu 22,5 Prozent erkannt. Die richtige Priorität wurde zu 25 Prozent klassifiziert. Die Abweichungen zu zwei anderen Varianten, die eine andere Kombination der Merkmale für die Klassifikation verwenden, sowie ein alternatives Klassifikationsverfahren stellten keine signifikanten Änderungen in der Erkennungsrate dar.

Im Folgenden werden die Ergebnisse, getroffene Annahmen, mögliche Alternativen und ein Ausblick anhand des Konzepts inklusive den Frameworks und dem Vorfalkatalog sowie der Umsetzung diskutiert.

Das **Konzept** bestand darin, Intentionen aus den Aussagen des Meldenden zu erfassen und in Klassen einzuteilen. Dies basiert auf der Annahme, dass einerseits Informationen eine semantische Einheit bilden und andererseits, dass diese durch Vorgabe der Reihenfolge und Formulierung der Fragestellungen systematisch erhoben werden können. Diese Absichten zur Informationsübermittlung lassen sich durch W-Fragen (Wo, Was/Wo genau, Wann, bis wann und Wer) abstrahiert beschreiben. Folglich reduziert dies zum einen die mögliche Informationsmenge, zum anderen werden die strukturierte Erfassung und Interpretation der Informationen erleichtert.

Infolgedessen schließt dies aber die Informationsextraktion verschiedener Informationstypen aus. Denn für die Zuordnung zu einer Klasse ist es nötig, dass die Aussagen zu einer Absicht möglichst homogen sind, um diese im Klassifikationsverfahren zu erkennen. Trotzdem stellt die Beschränkung der Informationsextraktion in einer Nachricht im Rahmen eines Chat-Bots kein Problem dar, da in der Regel die Nachrichten kurz sind.

Eine Alternative zu den Frameworks mit maschinellem Lernen stellen regelbasierte Frameworks dar. Jedoch wurden diese nicht untersucht, da die Erstellung von Regeln, welche die Informationen aus den Aussagen des Nutzers extrahieren, den Umfang dieser Arbeit übertreffen würden.

Die Auswahl der für dieses Vorgehen in 3.2 beschriebenen **Frameworks** bezieht sich auf keine Quelle. Die Frameworks wurden durch Recherche in Chat-Bot Foren und der Literatur ausgemacht. Darüber hinaus existieren weitere Frameworks, die im Rahmen dieser Arbeit nicht untersucht wurden, beispielsweise NADIA von Berg. Die darauffolgende Auswahl der Frameworks für die Implementierung wurde jedoch in Kapitel 5 begründet. Hervorzuheben ist, dass dies lediglich dem aktuellen Stand der Technik entspricht.

Aufgrund der intensiven Weiterentwicklungen der Frameworks können sich beschriebene Eigenschaften und damit die Ausgangsbedingungen rasch ändern, sodass Amazon Lex, Watson Conversation oder Wit.ai zu einem späteren Zeitpunkt ebenfalls als geeignet erscheinen könnten. Insbesondere die Systematik, mit welcher das Sprachverständnis in Amazon Alexa beschrieben wird, zeigt das Potenzial für Amazon Lex.

Weiterhin beschränkt sich diese Arbeit auf textbasierte Chat-Bots. Allerdings bildet die textbasierte Verarbeitung die Grundlage und stellt somit keine Einschränkung für eine folgende Erweiterung um eine auditive Schnittstelle dar. Diese ist beispielsweise in Api.ai integriert und ohne weiteren Implementierungsaufwand umsetzbar.

Die Daten im **Vorfallkatalog**, welche das Geschehen beschreiben, wurden an den Erfahrungsschatz der synexs GmbH angelehnt. Aus diesen 120 Vorfällen wurden das Trainings- und Testset für die Evaluation erzeugt, sowie eine Kategorisierung über die Art der Vorfälle erarbeitet. Trotzdem repräsentiert der Datensatz nur einen Teil der Vorfälle der synexs GmbH beziehungsweise aus der FM-Branche. Folglich ist die Zusammenstellung von umfangreichem Datenmaterial für die Verbesserung der Vorfallerkennung und Klassifikation unumgänglich. Andererseits ist auch festzuhalten, dass der Zugang zum Datenmaterial nicht öffentlich und nur über die FM-Unternehmen möglich ist.

Das für die **Implementierung** verwendete Meta-Framework Botkit verbindet und implementiert die NLU-Dienste, die Datenspeicherung und das Dialogmanagement. Botkit und die verwendeten NLU-Frameworks sind erst ein paar Monate alt. An dieser Stelle traten Fehler und Probleme auf, beispielsweise war die Klassifizierung von Entitäten in LUIS durch einen Softwarefehler immer dann nicht möglich, wenn sich Sonderzeichen in den Aussagen befanden. Die Probleme konnten jedoch im Kontakt mit dem NLU-Dienstleister behoben werden. Im Übrigen bieten Api.ai, Recast.ai und andere einen direkten Entwicklerkontakt über einen textbasierten Chat auf der Weboberfläche an.

Im Lösungskonzept wurde bereits an einigen Stellen ein Ausblick gegeben. Dazu gehören die Aussagen des Meldenden, welche die Absicht oder das Bedürfnis ausdrücken, einen Supportmitarbeiter zu kontaktieren. In der Implementierung endet die Unterhaltung im System an dieser Stelle.

Ebenso ist ein Datenaustausch mit dem CAFM über die Informationen zu Adresse des Gebäudes, installierten Geräten und Systemen, eingegangenen Meldungen und historisch bekannten Störfällen nicht implementiert worden. Dieser kann jedoch dazu beitragen, die Korrektheit der Informationen (beispielsweise der Adresse) sicherzustellen oder die Daten zur Verbesserung der Klassifizierung des Vorfalls und dessen Priorität zu nutzen.

Weiterhin wurde die Gebäudeadresse nicht über die Funktionalität von Google Maps hinaus geprüft. Eingaben abseits der Straße und Hausnummer wurden nicht akzeptiert. Eine Erweiterung dazu stellt die Erkennung von bekannten Orten wie Kaufhäusern oder Schulen dar. Ebenfalls ist eine Verbesserung in Kombination mit einer Schnittstelle zum CAFM denkbar.

Abschließend ist darauf hinzuweisen, dass der verwendete Klassifikationsalgorithmus ID3 aus dem Jahre 1986 stammt und von Quinlan mit dem C4.5 Algorithmus sowie dem C5 Algorithmus verbessert wurde. Die Node.js-Implementierung von Ersterem stoppte jedoch die Klassifizierung bei nicht trainierten Merkmalen, statt bei Ungenauigkeit mehrere Klassen anzugeben, und Letzterer steht unter kommerzieller Lizenz, weswegen keine Node.js-Implementierung vorlag. Wichtiger als algorithmische Verbesserungen im Klassifikationsverfahren ist jedoch die zu Grunde liegende Datenbasis, da insbesondere die von Quinlan vorgenommenen Verbesserungen für große Datensätze wirksam werden.

Insgesamt lässt sich hieraus der Schluss ziehen, dass für die Verbesserung der Resultate ein umfassender Datenbestand an Vorfällen benötigt wird. Erkennungsraten von unter 50 Prozent bei den Entitäten und unter 25 Prozent in der Klassifikation des Vorfalls sind zu gering, um den Chat-Bot im Praxisalltag zu nutzen. Dennoch wurde ein Lösungskonzept vorgeschlagen und eine Realisierung vorgestellt, welche die potenzielle Machbarkeit belegt und sich überdies durch eine integrierte Lernfähigkeit auszeichnet.

## Literaturverzeichnis

- [AA-1-ol] API.AI: Documentation: Key Concepts: Introduction. 2017.  
<https://docs.api.ai/docs/key-concepts> (Zugriff am 9. März 2017)
- [AA-2-ol] API.AI: Documentation: API Reference: Languages. 2017.  
<https://docs.api.ai/docs/languages> (Zugriff am 9. März 2017)
- [AA-3-ol] API.AI: Documentation: Domains: Domains Overview. 2017.  
<https://docs.api.ai/docs/domains> (Zugriff am 9. März 2017)
- [AA-4-ol] API.AI: Documentation: Integrations: API.AI Integrations. 2017.  
<https://docs.api.ai/docs/integrations> (Zugriff am 9. März 2017)
- [AA-5-ol] API.AI: Documentation: Key Concepts: Agents. 2017.  
<https://docs.api.ai/docs/concept-agents> (Zugriff am 9. März 2017)
- [AA-6-ol] API.AI: Documentation: Key Concepts: Machine Learning. 2017.  
<https://docs.api.ai/docs/machine-learning> (Zugriff am 9. März 2017)
- [AA-7-ol] API.AI: Documentation: Fulfillment: Webhook. 2017.  
<https://docs.api.ai/docs/webhook> (Zugriff am 9. März 2017)
- [AA-8-ol] API.AI: Documentation: Key Concepts: Entities. 2017.  
<https://docs.api.ai/docs/concept-entities> (Zugriff am 9. März 2017)
- [AA-9-ol] API.AI: Documentation: Key Concepts: Intents. 2017.  
<https://docs.api.ai/docs/concept-intents> (Zugriff am 9. März 2017)
- [AA-10-ol] API.AI: Documentation: Key Concepts: Actions and Parameters. 2017.  
<https://docs.api.ai/docs/concept-actions> (Zugriff am 9. März 2017)
- [AA-11-ol] API.AI: Documentation: Key Concepts: Dialogs. 2017.  
<https://docs.api.ai/docs/dialogs> (Zugriff am 9. März 2017)
- [ABD+01] Allen, James F.; Byron, Donna K.; Dzikovska, Myroslava; Ferguson, George; Galescu, Lucian; Stent, Amanda: Toward conversational human-computer interaction. *AI Magazine*. 22, 4, 27-37. AAAI, 2011
- [AIML-ol] Wallace, Richard S.: AIML 2.0 Working Draft. 2014.  
<https://docs.google.com/document/d/1wNT25hJRyupcG51aO89UcQEiG-HkXRXusukADpFnDs4/pub> (Zugriff am 9. März 2017)
- [Ber13] Berg, Markus: Natürlichsprachlichkeit in Dialogsystemen. *Informatik Spektrum*. 36, 4, 371-381. Springer, 2013
- [BFW-1-ol] Microsoft: Microsoft Bot Framework: Skype bots: Getting Started. 2016.  
<https://docs.botframework.com/en-us/skype/getting-started/> (Zugriff am 9. März 2017)
- [BFW-2-ol] Microsoft: Microsoft Bot Framework: Core Concepts: Overview. 2016.  
<https://docs.botframework.com/en-us/core-concepts/overview/> (Zugriff am 9. März 2017)
- [Bis09] Bishop, Christopher M.: Pattern recognition and machine learning. Springer Spektrum, 2009
- [BOH+01] Braun, Hans-Peter; Oesterle, Eberhard; Haller, Peter; Bauer, Rupert: Facility Management: Erfolg in der Immobilienbewirtschaftung. Springer, 2001

- [BSA+98] Borthwick, Andrew; Sterling, John; Agichtein, Eugene; Grishman, Ralph: Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. *Sixth Workshop on Very Large Corpora*. 1998, 152-160. ACL, 1998
- [BK-1-ol] XOXCO: Botkit. 2017. <https://howdy.ai/botkit/> (Zugriff am 9. März 2017)
- [BK-2-ol] XOXCO: GitHub Quellcodeverwaltung: Botkit. 2017. <https://github.com/howdyai/botkit/blob/master/readme.md> (Zugriff am 9. März 2017)
- [CEE+10] Carstensen, Kai-Uwe; Ebert, Christian; Ebert, Cornelia; Jekat, Susanne J.; Klabunde, Ralf; Langer, Hagen: Computerlinguistik und Sprachtechnologie: Eine Einführung. Springer Spektrum, 2010
- [CFL13] Clark, Alexander; Fox, Chris; Lappin, Shalom: The handbook of computational linguistics and natural language processing. Wiley-Blackwell, 2013
- [CID+97] Carletta, Jean; Isard, Stephen; Doherty-Sneddon, Gwyneth; Isard, Amy; Kowtko, Jacqueline C.; Anderson, Anne H.: The reliability of a dialogue structure coding scheme. *Computational Linguistics*. 23, 1, 13-31. MIT Press, 1997
- [CoAI97] Core, Mark G.; Allen, James F.: Coding dialogs with the DAMSL annotation scheme. *Working notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*. Boston, 1997
- [Crlm06] Cramer, Irene; Im Schulte Walde, Sabine: Computerlinguistik und Sprachtechnologie. Groos, 2006
- [CS-1-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: ChatScript. 2017. <https://github.com/bwilcox-1234/ChatScript> (Zugriff am 9. März 2017)
- [CS-2-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: ChatScript Wiki. 2017. <https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/README.md> (Zugriff am 9. März 2017)
- [CS-3-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: ChatScript Foreign Language Support. 2017. <https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/ESOTERIC-CHATSCRIPT/ChatScript-Foreign-Languages.md> (Zugriff am 9. März 2017)
- [CS-4-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: ChatScript Basic User Manual. 2017. <https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/ChatScript-Basic-User-Manual.md> (Zugriff am 9. März 2017)
- [CS-5-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: ChatScript Tutorial. 2017. <https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/OVERVIEWS-AND-TUTORIALS/ChatScript-Tutorial.md> (Zugriff am 9. März 2017)
- [CS-6-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: ChatScript Advanced User's Manual. 2017. <https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/ChatScript-Advanced-User-Manual.md> (Zugriff am 9. März 2017)
- [CS-7-ol] Wilcox, Bruce: GitHub Quellcodeverwaltung: What is ChatScript: Overview. 2017. <https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/OVERVIEWS-AND-TUTORIALS/What-is-ChatScript.md> (Zugriff am 9. März 2017)
- [FB-ol] Facebook: Messenger Platform Changelog. 2017. <https://developers.facebook.com/docs/messenger-platform/changelog> (Zugriff am 9. März 2017)
- [FGM05] Finkel, Jenny Rose; Grenager, Trond; Manning, Christopher: Incorporating non-local information into information extraction systems by Gibbs sampling. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. 2003, 363-370. ACL, 2003

- [Fir51] Firth, John Rupert: Modes of Meaning. *Papers in Linguistics 1934-1951 by J. R. Firth*. 1951, 190-215. Oxford University Press, 1957
- [Fox89] Fox, Christopher: A stop list for general text. *ACM SIGIR Forum*. 24, 1-2, 19-21. ACM, 1989
- [GATE-1-ol] The University of Sheffield: Plugins included in the GATE distribution. 2017. <https://gate.ac.uk/gate/doc/plugins.html> (Zugriff am 9. März 2017)
- [GATE-2-ol] The University of Sheffield: Chapter 15: Non-English Language Support. 2017. <https://gate.ac.uk/sale/tao/splitch15.html> (Zugriff am 9. März 2017)
- [GEFMA-100-1] GEFMA: Facility Management: Grundlagen. Beuth Verlag, 2004
- [GEFMA-520] GEFMA; RealFM: Standardleistungsverzeichnis Facility Services. Beuth Verlag, 2014
- [GIRö05] Glück, Helmut; Rödel, Michael: Metzler Lexikon Sprache. J.B. Metzler, 2016
- [GOOGLE-1-ol] Google: CLOUD NATURAL LANGUAGE API. 2017. <https://cloud.google.com/natural-language/> (Zugriff am 9. März 2017)
- [Gri75] Grice, Herbert Paul: Logic and Conversation. *Syntax and Semantics*. 3, 41-58. Academic Press, 1975
- [GSC+04] Grau, Sergio; Sanchis, Emilio; Castro, María José; Vilar, David: Dialogue act classification using a Bayesian approach. *Proceedings of the Ninth International Conference Speech and Computer*. 2004, 495-499. SPECOM, 2004
- [HiGa01] Hirschman, L.; Gaizauskas, R.: Natural language question answering: The view from here. *Natural Language Engineering*. 7, 4, 275-300. Cambridge University Press, 2001
- [HMU02] Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D., Theodor: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. Pearson Studium, 2002
- [HUB-ol] HubSpot Research: Artificial Intelligence Is Here - People Just Don't Realize It. 2017. <https://research.hubspot.com/reports/artificial-intelligence-is-here> (Zugriff am 9. März 2017)
- [Hul96] Hull, David A.: Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*. 47, 1, 70-84. Wiley, 1996
- [IBM-1-ol] IBM: Overview of the IBM Watson Conversation service. 2017. <https://www.ibm.com/watson/developercloud/doc/conversation/index.html> (Zugriff am 9. März 2017)
- [IBM-2-ol] IBM: IBM Watson Conversation service: Getting started. 2017. <https://www.ibm.com/watson/developercloud/doc/conversation/getting-started.html> (Zugriff am 9. März 2017)
- [IBM-3-ol] IBM: IBM Watson Conversation service: Defining entities. 2017. <https://www.ibm.com/watson/developercloud/doc/conversation/entities.html> (Zugriff am 9. März 2017)
- [IBM-4-ol] IBM: IBM Watson Conversation service: Building a dialog. 2017. <https://www.ibm.com/watson/developercloud/doc/conversation/dialog-build.html> (Zugriff am 9. März 2017)
- [IBM-5-ol] IBM: IBM AlchemyLanguage service: API Reference. 2017. <https://www.ibm.com/watson/developercloud/alchemy-language/api/v1/> (Zugriff am 9. März 2017)

- [JiCr94] Jing, Yufeng; Croft, W. Bruce: An Association Thesaurus for Information Retrieval. *Intelligent Multimedia Information Retrieval Systems and Management*. 1994, 146-160. C.I.D, 1994
- [KGN16] Khanpour, Hamed; Guntakandla, Nishitha; Nielsen, Rodney: Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, 2012-2021. COLING, 2016
- [Kla-1-ol] Klatte, Sebastian: GitHub Quellcodeverwaltung: FacilitySupportBot. 2017. <https://github.com/geskill/edu.upb.FacilitySupportBot> (Zugriff am 12. April 2017)
- [Kon01] Kononenko, Igor: Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*. 23, 1, 89-109. Elsevier, 2001
- [Kos07] Maglogiannis, Ilias G.: Supervised Machine Learning: A Review of Classification Techniques. *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. 2007, 3-24. IOS Press, 2007
- [Kuk92] Kukich, Karen: Technique for automatically correcting words in text. *ACM Computing Surveys*. 24, 4, 377-439. ACM, 1992
- [LaDr98] Lawler, John; Dry, Helen Aristar: Using computers in linguistics: A practical guide. Routledge, 1998
- [Lew85] Lewandowski, Theodor: Linguistisches Wörterbuch. Quelle & Meyer, 1985
- [LEX-1-ol] Amazon Web Services: Amazon Lex Documentation: Developer Guide. 2017. <http://docs.aws.amazon.com/lex/latest/dg/lex-dg.pdf> (Zugriff am 9. März 2017)
- [LEX-2-ol] Amazon: Alexa Skills Kit: Built-in Intent Library. 2017. <https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/built-in-intent-ref/built-in-intent-library> (Zugriff am 9. März 2017)
- [LEX-3-ol] Amazon: Alexa Skills Kit: Slot Type Reference. 2017. <https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/built-in-intent-ref/slot-type-reference> (Zugriff am 9. März 2017)
- [LEX-4-ol] Amazon Web Services: Amazon Lex Pricing. 2017. <https://aws.amazon.com/lex/pricing/> (Zugriff am 9. März 2017)
- [LMP01] Lafferty, John; McCallum, Andrew; Pereira, Fernando: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*. 2001, 282-289. Elsevier, 2001
- [MA-1-ol] Motion AI: Homepage. 2017. <https://www.motion.ai/> (Zugriff am 9. März 2017)
- [MA-2-ol] Motion AI: Documentation: Modules. 2017. <http://docs.motion.ai/docs/what-are-modules> (Zugriff am 9. März 2017)
- [MA-3-ol] Motion AI: Documentation: Extracted Data. 2017. <http://docs.motion.ai/docs/extracted-data> (Zugriff am 9. März 2017)
- [MaSc05] Manning, Christopher D.; Schütze, Hinrich: Foundations of statistical natural language processing. MIT Press, 2005



- [MoRo01] Moslener, Walther J. F.; Rondeau, Edmond P.: Facility-Management. Springer, 2001
- [MS-1-ol] Microsoft: Bing Spell Check API. 2016. <https://www.microsoft.com/cognitive-services/en-us/bing-spell-check-api> (Zugriff am 9. März 2017)
- [MS-2-ol] Microsoft: Documentation: Bing Spell Check API: Overview. 2016. <https://www.microsoft.com/cognitive-services/en-us/Bing-Spell-check-API/documentation> (Zugriff am 9. März 2017)
- [MS-3-ol] Microsoft: Language Understanding Intelligent Service. 2016. <https://www.microsoft.com/cognitive-services/en-us/language-understanding-intelligent-service-luis> (Zugriff am 9. März 2017)
- [MS-4-ol] Microsoft: Documentation: Language Understanding Intelligent Service: Overview. 2016. <https://www.microsoft.com/cognitive-services/en-us/luis-api/documentation/home> (Zugriff am 9. März 2017)
- [MS-5-ol] Microsoft: Documentation: Language Understanding Intelligent Service: Add Entities. 2016. <https://www.microsoft.com/cognitive-services/en-us/LUIS-api/documentation/Add-entities> (Zugriff am 9. März 2017)
- [MS-6-ol] Microsoft: Documentation: Language Understanding Intelligent Service: Pre-built Entities. 2016. <https://www.microsoft.com/cognitive-services/en-us/LUIS-api/documentation/Pre-builtEntities> (Zugriff am 9. März 2017)
- [MS-7-ol] Microsoft: Documentation: Language Understanding Intelligent Service: Add example utterances. 2016. <https://www.microsoft.com/cognitive-services/en-us/LUIS-api/documentation/Add-example-utterances> (Zugriff am 9. März 2017)
- [MS-8-ol] Microsoft: Documentation: Language Understanding Intelligent Service: Add Features. 2016. <https://www.microsoft.com/cognitive-services/en-us/LUIS-api/documentation/Add-Features> (Zugriff am 9. März 2017)
- [MS-9-ol] Microsoft: Documentation: Linguistic Analysis APIs: Overview. 2016. <https://www.microsoft.com/cognitive-services/en-us/linguistic-analysis-api/documentation/overview> (Zugriff am 9. März 2017)
- [MS-10-ol] Microsoft: Documentation: Text Analytics Documentation: Overview. 2016. <https://www.microsoft.com/cognitive-services/en-us/text-analytics/documentation> (Zugriff am 9. März 2017)
- [Mug86] Mugdan, Joachim: Was ist eigentlich ein Morphem? *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*. 39, 29-43. Akademie-Verlag, 1986
- [NaSe07] Nadeau, David; Sekine, Satoshi: A survey of named entity recognition and classification. *Lingvisticæ Investigationes: Named Entities: Recognition, classification and use*. 30, 1, 3-26. John Benjamins, 2007
- [NLTK-1-ol] NLTK Project: Homepage. 2017. <https://www.nltk.org/> (Zugriff am 9. März 2017)
- [NLTK-2-ol] NLTK Project: API / nltk Package. 2017. <https://www.nltk.org/api/nltk.html> (Zugriff am 9. März 2017)
- [NODE-ol] Node.js Foundation: Home. 2017. <https://nodejs.org/en/> (Zugriff am 9. März 2017)
- [NuRö88] Nullmeier, Erhard; Rödiger, Karl-Heinz: Dialogsysteme in der Arbeitswelt. BI-Wiss-Verl., 1988

- [Qui86] Quinlan, J. R.: Induction of Decision Trees. *Machine Learning*. 1, 1, 81-106. Elsevier, 1986
- [Qui87] Quinlan, J. R.: Simplifying decision trees. *International Journal of Man-Machine Studies*. 27, 3, 221-234. Elsevier, 1987
- [RA-1-ol] Recast.ai: Pricing. 2017. <https://recast.ai/pricing> (Zugriff am 9. März 2017)
- [RA-2-ol] Recast.ai: Documentation. 2017. <https://man.recast.ai/> (Zugriff am 9. März 2017)
- [RAN07] Runeson, Per; Alexandersson, Magnus; Nyholm, Oskar: Detection of Duplicate Defect Reports Using Natural Language Processing. *29th International Conference on Software Engineering*. 2007, 499-510. IEEE, 2007
- [ReDa00] Reiter, Ehud; Dale, Robert: Building natural language generation systems. Cambridge University Press, 2000
- [RiLe11] Rieser, Verena; Lemon, Oliver: Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation. Springer, 2011
- [RS-1-ol] Petherbridge, Noah: What is RiveScript? 2017. <https://www.rivescript.com/about> (Zugriff am 9. März 2017)
- [RS-2-ol] Petherbridge, Noah: History of RiveScript. 2017. <https://www.rivescript.com/history> (Zugriff am 9. März 2017)
- [RS-3-ol] Petherbridge, Noah: RiveScript Interpreters. 2017. <https://www.rivescript.com/interpreters> (Zugriff am 9. März 2017)
- [SiRi03] Silva, C.; Ribeiro, B.: The importance of stop word removal on recall values in text categorization. *Proceedings of the International Joint Conference on Neural Networks*. 2003, 1661-1666. IEEE, 2003
- [SKYPE-1-ol] Skype Team: Skype Bots preview comes to consumers and developers. 2016. <https://blogs.skype.com/news/2016/03/30/skype-bots-preview-comes-to-consumers-and-developers/> (Zugriff am 9. März 2017)
- [SKYPE-2-ol] Skype Team: Try the new Skype Preview with the Windows 10 Anniversary Update. 2016. <https://blogs.skype.com/news/2016/08/02/try-the-new-skype-preview-with-the-windows-10-anniversary-update/> (Zugriff am 9. März 2017)
- [SmHi94] Smith, Ronnie W.; Hipp, D. Richard: Spoken natural language dialog systems: A practical approach. Oxford University Press, 1994
- [SMST-ol] SmartyStreets: The Difference Between "Mappable" and "Valid". 2016. <https://smartystreets.com/articles/do-maps-verify-addresses> (Zugriff am 9. März 2017)
- [SRC+00] Stolcke, Andreas; Ries, Klaus; Coccaro, Noah; Shriberg, Elizabeth; Bates, Rebecca; Jurafsky, Daniel; Taylor, Paul; Martin, Rachel; van Ess-Dykema, Carol; Meteer, Marie: Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*. 26, 3, 339-373. MIT Press, 2000
- [TRA-ol] Tractica: Virtual Digital Assistants. 2016. <https://www.tractica.com/research/virtual-digital-assistants/> (Zugriff am 9. März 2017)
- [vBo12] van Bon, Jan: ITIL: A pocket guide. Van Harten Pub, 2012
- [WHW05] Webb, Nick; Hepple, Mark; Wilks, Yorick: Error Analysis of Dialogue Act Classification. *Proceedings of the 8th international conference on Text, Speech and Dialogue*. 2005, 451-458. Springer, 2005

- [WIKI-1-ol] Wikipedia: Gebäudeadresse. 2017. <https://de.wikipedia.org/wiki/Geb%C3%A4udeadresse> (Zugriff am 9. März 2017)
- [WIKI-2-ol] Wikipedia: Ortschaft. 2017. <https://de.wikipedia.org/wiki/Ortschaft> (Zugriff am 9. März 2017)
- [WIKI-3-ol] Wikipedia: Postleitzahl. 2017. [https://de.wikipedia.org/wiki/Postleitzahl\\_\(Deutschland\)](https://de.wikipedia.org/wiki/Postleitzahl_(Deutschland)) (Zugriff am 9. März 2017)
- [WIKI-4-ol] Wikipedia: Straßenname. 2017. <https://de.wikipedia.org/wiki/Stra%C3%9Fenname> (Zugriff am 9. März 2017)
- [WIT-1-ol] Wit.ai: Frequently Asked Questions. 2017. <https://wit.ai/faq> (Zugriff am 9. März 2017)
- [WIT-2-ol] Wit.ai: Documentation Recipes. 2017. <https://wit.ai/docs/recipes> (Zugriff am 9. März 2017)
- [WIT-3-ol] Wit.ai: Documentation Console Complete Guide. 2017. <https://wit.ai/docs/complete-guide> (Zugriff am 9. März 2017)
- [WKQ+07] Wu, Xindong; Kumar, Vipin; Ross Quinlan, J.; Ghosh, Joydeep; Yang, Qiang; Motoda, Hiroshi; McLachlan, Geoffrey J.; Ng, Angus; Liu, Bing; Yu, Philip S.; Zhou, Zhi-Hua; Steinbach, Michael; Hand, David J.; Steinberg, Dan: Top 10 algorithms in data mining. *Knowledge and Information Systems*. 14, 1, 1-37. Springer, 2007
- [ZhLi07] Zhang, Yi; Liu, Bing: Semantic Text Classification of Emergent Disease Reports. *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*. 2007, 629-637. Springer, 2007



## Anhang

Der Anhang umfasst den Vorfallkatalog, da dieser aufgrund des Umfangs nach nicht im laufenden Text untergebracht wurde. Die Tabelle A.1 enthält das Trainingsset mit 80 Vorfällen und Tabelle A.2 das Testset mit 40 Vorfällen. Zudem stellt Tabelle A.3 50 Aussagen mit irrelevanten Aussagen und Aussagen über den Zeitpunkt zusammen.

*Tabelle A.1: Vorfallkatalog - Trainingsset*

Kategorie	Priorität	Aussage
Gefahr	1	Fraunhoferstraße 25 in 10587 >> There is a fire in the kitchen >> Sixth storey >> My name is Peter Murchison, phone is 754-3010, mail is peter.murchison@example.com
Gefahr	1	I'm in Hamburg >> In Weckmannweg 10 >> Someone broke into the store >> In the jewelry store in the first floor >> I am Daryl Luckow
Gefahr	1	I am currently in Hamburg >> I'm at Rutschbahn 18 >> There was a hold-up >> On the ground floor >> Quiana Sepe
Gefahr	1	I am in Munich, in the Wertheimer Str. 23 >> There were burglars >> Main floor >> My name is Kip Russel phone 06 55 876 375
Gefahr	1	At the moment I am in Munich, in the Hitzestraße 1 >> The elevator does not work >> The elevator on the north side >> (822) 096-2687

Gefahr	1	Cologne, Huhns-gasse 12 >> The escalator on the southern area doesn't work >> My name is Giovanna Kimbrough my email is e-z@jjmjveuew.com
Gefahr	1	Wiesenau 27, Frankfurt am Main >> blocks of ice are falling from the roof >> I am Claretha Striegel
Gefahr	1	In the Senefelderstraße, 75 >> In Stuttgart >> The trees are endangering the building >> Alfredo Kinser phone (822) 744-9740 my email is jok@7-44gq.com
Gefahr	1	Currently i'm in the "Brennerstraße 3" >> Stuttgart >> There are terrorists >> Downstairs >> I am Mark Behrman phone (855) 441-9646
Gefahr	1	In Schadowstraße 76 >> I'm in Dusseldorf >> A window is shattered >> The front side of the building >> I am Tamika Britto my phone is (855) 335-1560
Gefahr	1	Right now i'm in the Lewitstraße 54 in Dusseldorf >> The front door is demolished >> I'm Anissa Windham, phone is 754-3010
Elektrizität	1	Hüttmannstraße 78 >> Essen >> There is no electricity in the whole building >> I'm Ona Vazquez phone (822) 744-9740
Elektrizität	1	At the Rolf-Dieter-Koch-Weg 16 in Essen >> The is no power on the third floor >> Zack Heather z9v@iwrinnx500i.com
Elektrizität	1	Lauschaer Weg no 1 >> In Leipzig >>

		<p>The electricity is failing &gt;&gt;  Throughout the building &gt;&gt;  I am Leeann Hale my phone is +49 202 5550155</p>
Daten-netzwerk	1	<p>Right now, I'm in the Gutenbergstraße number 36, Bremen &gt;&gt;  The web in room 534 is down &gt;&gt;  My name is Gerard Peckham my phone is +49 202 5550166</p>
Daten-netzwerk	1	<p>In the Wartburgstraße 38/1 &gt;&gt;  01309 Dresden &gt;&gt;  The internet in room D42 doesn't work &gt;&gt;  Alene Shuler my email is w3bgpb.6d@a7p96bkbee.com</p>
Wasser	1	<p>At the moment, in "Unkersdorfer Str. 11" in Dresden &gt;&gt;  A pipe has broken &gt;&gt;  Seems like the boiler room &gt;&gt;  Applin phone +49 202 5550142 mail  la.r7i@8b1vxqw4.com</p>
Wasser	1	<p>Currently in Hanover, Franzstraße 2 &gt;&gt;  A pipe burst &gt;&gt;  In the ground level &gt;&gt;  I'm Fatimah Height mail is okcn41xcw4ctw1@x-hsivn9.com</p>
Temperatur	1	<p>Nuremberg, Adamstraße number 8 &gt;&gt;  It's very cold in here &gt;&gt;  Room no 5.233 &gt;&gt;  I'm Kristy Bradburn phone +49 202 5550155</p>
Temperatur	1	<p>Address is Elsterstraße 12 in Duisburg &gt;&gt;  It's super hot in the office &gt;&gt;  In the fifth storey room number D243 &gt;&gt;  I'm Kirksey phone +1-202-555-0179</p>
Geruch	1	<p>This is in Bochum, Huestraße 14 &gt;&gt;  the dining hall stinks brutally &gt;&gt;  I'm Yajaira Salley, tb1d6subl-se@l0n0-5bfgn.com</p>

Zutritt	1	Kluser Höhe number 10, in Wuppertal >> I can't access my room >> This is 1225 >> I am Elbert Fuller, phone is +1-202-555-0150
Zutritt	1	Wuppertal in Otto-Hausmann-Ring 108 >> I can not get in my office >> My room number is 16D >> My name is Patsy Dipalma, +1-202-555-0142
Zutritt	1	In "Im Strohsiek 11", in Bielefeld city. >> The roller door is defective >> Outside >> I am Rocky Kovacich, 02686 89 69 90
Sauberkeit	1	In Bonn, Wasserturmstraße 30 >> The lounge is extremely dirty >> I am Dorris Wilken mail 2sf52byf5ibldu@9f-voim-qt.com
Sauberkeit	1	Right now, I'm in the street "Am Kleibach 14A" in Muenster >> the foyer looks devastated >> Adaline Maust phone 0341 35 87 18
Erscheinungsbild	1	I'm currently in Muenster in the Coerdestraße 3B. >> there's a racist graffiti on the facade >> My name is Delbert Arbaugh my email is ydgia8-kh@sdu6oi5j.com
Erscheinungsbild	1	Theodor-Heuss-Allee 15 >> the garden was vandalized >> I am Alma Marrow, 02804 24 06 74, mail is d6wkl17fcc_nrm@acius5wh.com
technische Anlagen	1	In Mannheim Q2 5 >> the air conditioning in room no. 325 isn't working >> Shakira, phone is 06207 73 12 18
technische Anlagen	1	In M1 6, Mannheim >> the ventilation has failed >>



		In my room 345 >> My name is Shelley Spector phone 1(992)941-75-96
Sanitäranlagen	1	The street name is Kanalstraße in Augsburg >> This is 14 >> toilet drain on the first floor is clogged up >> I'm Isaura Henninger
Sanitäranlagen	1	Right now I'm in Wiesbaden in Am Speiergarten 24D >> the restroom is under water >> Next to the entrance stairway >> Haywood Beckwith
Defekt	2	In Gelsenkirchen >> Schalker Str. 111 >> the window in the open-plan office cannot be opened >> On the fifth floor >> I'm Frank Mishoe my phone is 540(07)721-47-30
Defekt	2	In "Lüpertzender Str. 133" in Monchengladbach >> a light in the room 3.42 is flickering >> My name is Reiher my phone is 421(422)053-42-65
Elektrizität	2	In Braunschweig. The street name is Burgpassage 20 >> I have no power at my workplace >> Room number E.42 >> My name is Donnie Cruzan phone 8(6102)248-29-17
Elektrizität	2	The street is Sack 8 and the city Braunschweig >> my computer doesn't boot >> is 5th floor room number 42 >> my phone is 754-3010 my email is 7h8@ix4du59c9w.com
Daten- netzwerk	2	This is 09130 in Chemnitz. Street and number is Sebastian- Bach-Straße 18 >> I have no internet connection on my computer >> 134 >> I am Sofia Garrow phone 754-3010
Daten- netzwerk	2	My address is Kiel and the street is "Von-der-Tann-Straße 24" >> my laptop doesn't connect to the internet >>

		In my office, room 5.46 >> Meghann Amend
Wasser	2	The address is "Jungfernstieg 26" in Kiel >> a wall in the cafeteria is damp >> Jeannie Sheets, 754-3010, pvviozjv.ei@sjmavo.com
Temperatur	2	The city of Aachen. Neupforte 24 >> too humid at my workstation >> Top floor >> My name is Phyllis Kamrowski phone (855) 183-7220
Temperatur	2	Reilshof 6 >> In Halle >> the air is suffocating >> All over in the 4th level >> I am Jerrold Clemans, (833) 587-3811
Geruch	2	I'm in Magdeburg and the street name is Sudenburger Wuhne >> In Sudenburger Wuhne 65 >> the costumer area is smelly >> I'm Shay Mulder my phone is (811) 171-8580
Sicherheit	2	Right now, i am in Freiburg, ""Unterlinden 7"" >> Freiburg im Breisgau >> The fire extinguisher is expired >> On the upper floor >> I'm Mariella Spink
Sicherheit	2	This is in Freiburg im Breisgau. In the Roseggerstraße 7 >> The fire drencher is out-of-date >> On the second floor >> Celsa Schott my mail w4j1v9@secoepp.com
Sicherheit	2	I am in the city of Krefeld. It is the Viktoriapl. 15 >> The extinguisher on the 3rd lvl is missing >> I am Max Eifert my email is s.a.@i6g9p5m.com
Sicherheit	2	It's in the Am Konnertzfeld 19 street in Krefeld >> The lighting for escape routes is partially defect >>

		In the hallway of the ground floor >> I'm Verdie Seese, (855) 335-1560
Sicherheit		In Lübeck city. In the Dr.-Julius-Leber-Straße 23 >> The first aid kit next to my work place is incomplete >> Room number 434-3 >> Ema Henshaw
Sauberkeit	2	It's in Oberhausen, Hermannstadtstraße 48 >> The reception is filthy >> I am Donald Goettl my phone is +49 202 5550155
Sauberkeit	2	Am Hügel no 28 >> In Erfurt >> dirty toilet >> The toilet on the second lvl >> My name is Carissa Maye, phone is +49 202 5550166
Sauberkeit	2	In the Dijonstraße 20 >> In the city of Mainz >> the sink in the girls room on the ground floor is messy >> I'm Antonette Bonnett, phone is +49 202 5550150 my email is _zjh0@w5fjhcu6dm.com
Erscheinungs- bild	2	In the Willi-Wolf-Straße 11C, in Mainz >> the trash is overflowing >> Zulema Harlan my phone is +49 202 5550142, 3sh_gd@obgi0wp.com
Erscheinungs- bild	2	At the moment in Beim Waisenhaus 1. This is in Rostock >> the window lighting in the store isn't working >> I am Suzette Jayne, 766(27)744-03-57, 32x0- vt@nfcqa5qro.com
Erscheinungs- bild	2	My address is Goethestraße 39 34119 Kassel >> the garden is messy >> I am Caroyln Puglisi, 421(422)053-42-65 my email is g5t6de3vse@h4y9sj5.com
technische Anlagen	2	Currently i am in Kassel city. >> In Kastenalsgasse 11 >>

		the air conditing on the third floor doesn't work >> Jerry Schroeder, 3(64)430-87-11
Sanitäranlagen	2	Goldberghang no 18, Hagen >> the water faucet in the girl's room on the second lvl is squeaking >> I'm Wayne Bonnette my phone is 754-3010
Sanitäranlagen	2	In Landwehrpl. 1, Saarbrücken >> the toilet on the ground floor doesn't flush right >> Reita Astin my phone is +1-202-555-0115
Defekt	3	The address is Am Alten Markt 9, in Potsdam >> a light in the hallway is flickering >> Marcell Crossan my phone is +1-202-555-0150
Defekt	3	I am at the Von-der-Tann-Straße 22 in Ludwigshafen am Rhein >> the wall of the entrance room is damaged >> I'm Emerita Dike my phone is +1-202-555-0142
Defekt	3	Reinhold-Tiling-Weg number 14 Osnabrück >> there are forgotten dowels in the cafeteria >> My name is Dagmar Monette
Defekt	3	Right now, i' am in Solingen city. >> In Talblick 20 >> dented door in the 3rd level >> My name is Santa Luque, phone is 02686 89 69 90
Defekt	3	Im Gabelacker 7 in Heidelberg city. >> the doors worn out paint on the ground floor >> My name is Concetta Mchenry, phone is 0341 35 87 18
Wasser	3	In Neuss city in 41462. Nehringskamp 13A >> a wall in the basement is damp >> I'm Mariko Matthias
Temperatur	3	Currently in Darmstadt, Pulverhäuserweg no 17 >> the basement is too hot >> I'm Nedra Melius, 02804 24 06 74

Temperatur	3	Schubertstraße 19A >> the temperature in the technical room is wrong >> My name is Kasandra Liz my phone is 1(992)941-75-96
Temperatur	3	In 85049 Ingolstadt >> The street is "Westliche Ringstraße 18" >> the storage space is overheated >> Tuan Briggs phone 15(574)018-69-99
Geruch	3	I'm in Fürth city. In the Pillauer Str. 13 >> the air in the storeroom is stale >> Louann Drapeau
Sauberkeit	3	I am in the Rothenfelder Str. 47 >> In Wolfsburg >> There are some minor coffee stains on the carpet >> in the information area >> Haley Arnold
Sauberkeit	3	I'm at the Geschwister-Scholl-Straße 9 in Offenbach >> Offenbach am Main >> Someone spilled some tea on the rug >> On the ground floor in hall C4 >> My name is Ignacio Shomo
Sauberkeit	3	Mosergasse No. 3 >> Here are leftovers on the second floor >> My name is Daria Delapaz
Sauberkeit	3	This is in Pforzheim, the Güterbahnhof 11A >> There are hot chocolate spots on the wall >> The wall next to the mens room on the 2nd level >> My name is Dannie Chason my phone is 3(64)430-87-11
Sanitäranlagen	3	At the moment I'm in Göttingen, in the Brunnenbreite number 14 >> there's a squeaking water faucet >> In the mens room 3rd floor >> My name is Bonnie Cowell

Sanitäranlagen	3	Right now i am in Trier in the street Vor Plein 4 >> toilet of the girls room on the 5th lvl won't flush >> I'm Leland Woolery phone 06 55 876 375
Defekt	4	An der Rennbahn in 45659, the house number is 45 >> There's a window in the cellar that can not be opened >> My name is Fred Kump, g5t6de3vse@h4y9sj5.com
Defekt	4	This is in ""27574 Bremerhaven"" and the street is Am Klint 16 >> The door in the store room is squeaking >> I'm Rachelle Arch, 421(422)053-42-65
Defekt	4	The street is: Bienenstück street number 8 in Koblenz >> In the basement theres a flickering light >> Errol Square
Defekt	4	The city is Bergisch Gladbach and the street is Am Fürstenbrunnchen number 24 >> The lights in the storage room are defect >> Jessica Fray, phone is (855) 183-7220
Defekt	4	Currently, i am in Remscheid, Glassiepen 25 >> The door next to the technical room didn't close properly >> I'm Arnulfo Edgington
Sauberkeit	4	I am in Erlangen, in Hugentottenpl. house number 6 >> the stairway has been cleansed not very carefully >> mail is qjwksicv@37s55dag.com
Sauberkeit	4	Currently, i'm in the city of Siegen. >> Am Kopf number 16 >> toilet got cleaned insufficiently >> girl's room in the 2nd lvl >> I am Tabatha Boler, my email is pvviozjv.ei@sjmavo.com
Sauberkeit	4	At the moment i'm in 38226 >> "Am Brinke 30" >> The stairwell is full of spider webs >> I'm Loyce Kolb my phone is (855) 335-1560

Tabelle A.2: Vorfallkatalog - Testset

Kategorie	Priorität	Aussage
Gefahr	1	In Johanniterstraße 2, Berlin >> There's a fire on the second floor! >> Barret, Cami, 06 55 876 375 my email is ke8bj@fl8ygu69sq.com
Gefahr	1	Right now I am in Cologne >> In the Bleriotstraße 12 >> Lift is stuck >> Left lift in the lobby >> phone is (855) 183-7220
Gefahr	1	In Frankfurt, Hahnstraße 32 >> There is a raging storm and tiles are falling from the roof >> My name is Patricia phone (811) 171-8580
Gefahr	1	Kuntzestraße number 61 >> In Dortmund >> The building was struck by lightning >> 06 55 876 375
Gefahr	1	In the Breslaustraße 31 in Dortmund >> The sidewalk is slippery >> Abram Musick, phone is (811) 171-8580
Elektrizität	1	Currently in Hardenbergstraße 27, 4275 Leipzig >> There is an electricity failure >> In the entire building >> My name is Hendley
Datennetzwerk	1	Georg-Gröning-Straße no. 102 >> 28209 >> There is no internet connection >> Everywhere >> Russel Habib my phone is +49 202 5550150

Wasser	1	Max-Eyth-Straße 38 >> Hanover >> There is flooding >> It comes from lavatory >> my email is 4z44vg1pi.rnd1@lo115t02vy.com
Temperatur	1	In Nuremberg >> In the Kunigundenstraße 17 >> It is extremely hot >> In the facility >> I'm Merle Sasson, phone is +49 202 5550155
Temperatur	1	Right now at the Mainstraße 18 >> In Duisburg >> The entrance area is freezing cold >> I'm Milan Goley
Geruch	1	The address is Am Buernacker 3 >> In Bochum >> there's a horrible smell coming from the café >> I'm Merlyn Litt, +1-202-555-0115
Zutritt	1	I am in Bielefeld in and the street name is Welle >> The number is 6 >> I cannot exit the parking lot >> I am Lessley, Rashad
Sauberkeit	1	Am Nesselroderhof 4 in 53111 >> entrance hall is a mess >> I'm Lajuana Tingler, tvudctucwu@ssghyw5v9.com
Erscheinungs- bild	1	I am in Karlsruhe, in the Stabelstraße 4 >> here's a graffiti that needs to be gone immediately >> The wall facing the street side >> My name is Wilton Leopard, phone is 02626 29 02 60
technische An- lagen	1	I'm in Zeugpl. 7 in Augsburg >> the heating system is malfunctioning >> In room B22 >> I'm Larissa Lucke, phone is 15(574)018-69-99



Sanitäranlagen	1	Currently, in Wiesbaden city. In the Langgasse 23 >> no water from the taps >> I am Lavada Jahnke, 5(3518)753-81-63
Defekt	2	Right now i am in Gelsenkirchen >> In the Beskenstraße 4 >> a door in the customer area is squeaking >> I am Nickolas Pettengill my phone is 766(27)744-03-57
Defekt	2	This is the Breite Str. 131 in >> In Monchengladbach >> the roller blinds in the rooms 23-54 are not working >> I am Patricia Nader, 3(64)430-87-11
Datennetzwerk	2	My address is Bodelschwinghstraße 17, in Chemnitz >> The network is very slow >> 3-34 >> I'm Amalia Uribe, phone is 06 55 876 375
Wasser	2	It is Rennbahn 11, 52062 Aachen >> the kitchen is flooded >> On the ground-floor >> I'm Lennox, Goldie
Temperatur	2	It's in Halle, in the Volkmannstraße 39 >> my room is stuffy >> This is T2.94 >> My name is Eunice Bilyeu, (899) 493-9512
Geruch	2	At the Agnetenstraße 11 in Magdeburg >> the corridor is reeking >> The 3rd floor >> Edgar Stines, (822) 744-9740
Sicherheit	2	In 23552 Lübeck Hartengrube 46 >> The safety lights fell out >> The corridor on the second floor >> I'm Boris Fidler
Sicherheit	2	My address is Centroallee 263A in Oberhausen >> First-aid box is missing >>

		In the security area >> mail is tnzqts5t8fba@h15pxog.com
Sauberkeit	2	In Erfurt, this is the Schlösserstraße 32 >> Filthy restroom >> The mens room, 2nd floor >> my phone is +49 202 5550115
Erscheinungs- bild	2	My street address is Hundertmännerstraße 3A, Rostock >> the shop windows are dirty >> I'm Trudi Azar
technische An- lagen	2	Right now, i'm in Hagen. In the Fleyer Str. number 47 >> ventilation on the 2nd floor is working improperly >> Morton Preuss, 4cx9011s@re-vb5xm0t.com
Sanitäranlagen	2	I'm in the Nordenwall 29 in Hamm city. >> there's too little water coming from the tap >> mens room on 4th floor >> I'm Rayford Salvatore phone 06 55 876 375
Defekt	3	At the moment I'm in Mülheim an der Ruhr, in Dohne 6 >> some lights in the parking lot are not working >> I am Shelley Spindler my phone is +1-202-555-0166 mail i8shk@qag9wcn5.com
Defekt	3	At the moment: Unterm Berg no 84A in Oldenburg >> there's a hole in the wall in the reception area >> Shawna Avendano, phone is 02686 89 69 90, mail is pvviozjv.ei@sjmavo.com
Defekt	3	Am Märchen 12 Leverkusen >> fissures in the wall on the ground floor >> I am Nathaniel Malagon, 02686 89 69 90, mail is 7d4z@q4ccly2ncl9k.com
Defekt	3	An der Ziegelei 53 in 44625 Herne >> the floor in hall 11 got damaged >> I am Kenny Scherf, phone is 035605 35 52, ayy5kuymv25vcq@ffapb7bqn.com

Temperatur	3	In Paderborn "Am Bischofsteich 5" >> it's freezing in the cellar >> I am Elease Polzin phone 06207 73 12 18, mail is w4j1v9@secoepp.com
Geruch	3	The street is Walther-von-der-Vogelweide-Straße 48, in 97074 Würzburg >> the cellar smells musty >> I am Perry Hose my phone is 400(9025)667-16-35
Sauberkeit	3	The street is Am Hetzenbäumle 132 in Ulm >> Just now somebody spilled a cup of cocoa on the floor >> On the first floor >> Hazel Hulse phone 766(27)744-03-57
Sanitäranlagen	3	I am in Bottrop in Lehmschlenke >> 89 >> not enough water coming out of the tap >> It's men's room in the 6th floor >> I am Carey Bastian, 754-3010
Defekt	4	In Reutlingen, Spechtshartstraße >> Spechtshartstraße 2 >> The third-floor staircase window is jammed >> Nannie Bratt
Defekt	4	In Jena, Magdelstieg 138 >> Jammed door lock in the boxroom >> I'm Lili Outler, 7h8@ix4du59c9w.com
Sauberkeit	4	This is in the Dr.-Karl-Hirschberg-Straße 9 >> In 47447, Moers >> the stairwell was not cleaned >> ztfeumc@eoq945nboxsw.com
Sauberkeit	4	I'm in 31137 Hildesheim >> The street is Cheruskerring 52 >> the sinks haven't been cleaned for a long time >> In the washroom >> Clemmie Stewart phone (855) 335-1560, 7d4z@q4ccly2ncl9k.com

*Tabelle A.3: Irrelevante Aussagen und Aussagen zum Zeitpunkt*

<b>Irrelevante Aussage</b>	<b>Aussage zum Zeitpunkt</b>
I like fruit.	the incident happened at 4 o clock
I'm feeling blue.	the incident happened an hour ago
I have two kids.	the incident took place at 9
It's foul weather today.	the incident took place at 7 a.m.
The color of my car is green.	the incident occurred a few minutes ago
It kept raining almost all day long.	the incident occurred at eleven thirty
My car is a real clunker.	the incident was at 11:30
This is a test	it happened at 10 past 3
Test	it happened half an hour ago
Mrs Miller wants the entire house re-painted.	it was quarter past 5
I live in Paderborn	it was about midnight
I live near Berlin	happened just now
I am living in an apartment very close to Munich	happened midnight
I currently live near Hamburg	happened today morning
It was nice to meet you	happened ten minutes ago
You are very pretty.	just now
How are you?	24 minutes ago
How are you doing?	quarter past 9

What's your name?	quarter to
My cousin will visit me tomorrow	half past
I dyed my hair	3 hours ago
Today I had some ice cream	5 o'clock
Yesterday I bought ice cream	7 o'clock
Last week I had to buy a new fire extinguisher	20 minutes past 6
Luckily my basement is cold enough to store food	10 minutes to 7
I wrote a test today.	fifteen minutes past seven
Let's go to a party tonight!	fifteen minutes to eight
My favourite music genre is alternative rock.	midnight
Hello	midday
Ok	noon
My office is still unfurnished	a few hours ago
I'll be waiting in the hotel lobby	about one hour ago
I have to go to the bathroom	one hour ago
My sink is made out of stainless steel.	a few minutes ago
self-cleaning toilets are neat	some minutes ago
The restroom there was very relaxing	14:50
No	three fifteen a.m.

the assembly hall was huge	five fifteen p.m.
Maybe	01:04 p. m.
I guess	10:45 a. m.
My workroom is only sparsely decorated	10:15 p. m.
My cellar is quite large	7 pm
I only drink tap water	today morning
The coffee stain is bothering me a lot	this morning
I painted my wall 4 different colours	15 minutes ago
Today a friend spilled coffee on my jeans	in the morning
Where are you?	before 8 am
Do you know John?	ten minutes ago
Is this a test?	yesterday afternoon
Are you crazy?	at 12 pm

## Inbetriebnahme des Facility-Support Bots

Für den Facility-Support Bot wird die Node.js Plattform benötigt:

- Node.js v7: <https://nodejs.org/en/download/current/>

Ebenfalls werden Entwicklungswerkzeuge von Microsoft benötigt. Dazu gehört die Entwicklungsumgebung Visual Studio Code (VS Code) und der Bot Framework Channel Emulator (Emulator), welcher als Simulationswerkzeug für einen Kommunikationsdienst wie Skype dient. Beide Werkzeuge lassen sich über die folgenden Weblinks herunterladen:

- Visual Studio Code: <https://code.visualstudio.com/>
- Bot Framework Channel Emulator: <https://emulator.botframework.com/>

Darüber hinaus wird der Quellcode des Facility-Support Bots benötigt. Dieser lässt sich von der beiliegenden CD oder über die GitHub Quellcodeverwaltung herunterladen:

- Implementierung: <https://github.com/geskill/edu.upb.FacilitySupportBot>

Folglich ist der Quellcode in einem beschreibbaren Verzeichnis abzulegen. Dieser Ordner ist in VS Code zu öffnen. (Alternativ ist auch das Herunterladen des Quellcodes über VS Code möglich.)

Über das integrierte Terminal von VS Code (STRG + ö) können mit dem Befehl `npm install` die benötigten Node.js Pakete aktualisiert oder erstmalig heruntergeladen werden. (Dazu muss der Ordner im Terminal dem Projektordner entsprechen. Gegebenenfalls muss dieser mit dem Befehl `cd <Verzeichnis>` angepasst werden.)

Für die verwendeten Komponenten und Dienste im Bot werden Anmeldeinformationen und Zugangsschlüssel benötigt. Diese Daten sind ausschließlich im Projektverzeichnis der CD hinterlegt und sind vertraulich zu behandeln. Dazu ist die .env-Datei in das Projektverzeichnis auf dem Computer zu kopieren, sofern noch nicht geschehen.

Der Bot kann über das integrierte Terminal mit dem Befehl `node bot.js` gestartet werden. Zudem kann mittels dem in VS Code integrierten Debugger der Bot analysiert werden. Im Terminal wird dann die Adresse, an welcher der Bot die Funktionalität als Service bereitstellt (auch Endpoint genannt), ausgegeben. Auf Windows-Systemen lautet diese Adresse:

- <http://127.0.0.1:3000/botframework/receive>

Daraufhin ist der Emulator zu starten und innerhalb des Programms der Endpoint zu hinterlegen. Zudem können die Anmeldeinformationen für den Bot eingegeben werden. Diese befinden sich, wie bereits erwähnt, ausschließlich auf der beigelegten CD. Folglich sollte eine Nachricht des Bots im Emulator erscheinen und der Dialog beginnen.

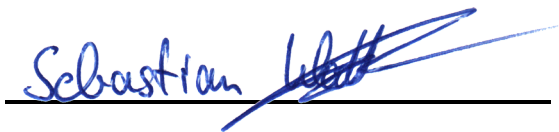
Hinweis: Wenn der Bot lokal betrieben wird, wird das ngrok-Werkzeug für Tunnelverbindungen nicht benötigt.





## Ehrenwörtliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte Hilfe Dritter angefertigt habe. Alle Stellen, die inhaltlich oder wörtlich aus Veröffentlichungen stammen, sind nach sorgfältiger Überprüfung und nach bestem Wissen kenntlich gemacht. Diese Arbeit lag in gleicher oder ähnlicher Weise noch keiner Prüfungsbehörde vor und wurde bisher noch nicht veröffentlicht.



Sebastian Klatte

Bad Salzuflen, den 15. April 2017