Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Evaluation von Informationsquellen zur Formulierung von Erklärungen am Beispiel einer Navigations-App

Evaluation of Information Sources for the Formulation of Explanations Using a Navigation App as an Example

Bachelorarbeit

im Studiengang Informatik

von

Nicolas Voß

Prüfer: **Prof. Dr. Kurt Schneider**
Zweitprüferin: **Dr. rer. nat. Jil Ann-Christin Klünder**
Betreuer: **Martin Obaidi / Florian Herzog**

Hannover, 30.08.2024

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 30.08.2024

Nicolas Voß

Zusammenfassung

Für Erklärungsbedarfe in Reviews ist es für Unternehmen meist ein manueller Vorgang, um diese detailliert beantworten zu können. Die Erklärungen sind des Öfteren die gleichen wie bei anderen Reviews derselben App, müssen allerdings selbstständig zugeordnet und gegebenenfalls leicht angepasst werden. Zur leichteren Einordnung können die Reviews mit einer Kategorie aus einer Taxonomie gelabelt werden, um eine Zugehörigkeit zu einer Person oder einem Team zu erteilen.

In dieser Bachelorarbeit wurde die automatisierte Zuordnung von Taxonomiekategorien an Reviews mit Erklärungsbedarf und die daraus folgernde Zuordnung an ein internes Team in einer Unternehmensstruktur und einer Erklärung evaluiert. Für die Datengrundlage der Reviews wurde eine Software zum Scrapen für den Google Play Store und den Apple App Store entwickelt. Die Reviews werden mithilfe einer weiteren Software einer Taxonomiekategorie durch eine Wörter- und Redewendungenfiltermethode zugeordnet und anschließend manuell überprüft. Anhand der Zuordnung der Taxonomiekategorie weist die Software den Reviews einen Bezugspunkt (Person oder Team) und, mithilfe einer API der Support-Webseite von Graphmasters GmbH und Antworten aus dem Google Play Store, eine Quelle zu. Durch Interviews und einer Umfrage mit dem Unternehmen Graphmasters GmbH wurden die Ergebnisse evaluiert und in der Praxis getestet. Die Auswertung zeigt, dass eine Zuordnung an einen Bezugspunkt keine klaren Ergebnisse aufweist, da Erklärungsbedarfe in der Praxis intuitiv beantwortet werden und selten festen Mustern folgen. Daher wird eine Zuordnung in absteigender Reihenfolge, bezogen auf die am wahrscheinlichsten zutreffenden Bezugspunkte, vorgenommen und kein einzelner Bezugspunkt zugeordnet.

Abstract

It is usually a manual process for companies to provide detailed answers to requests for explanations in reviews. The explanations are often the same as for other reviews of the same app, but must be assigned independently and slightly adjusted if necessary. For easier classification, the reviews can be labeled with a category from a taxonomy in order to assign them to a person or group.

In this bachelor's thesis, the automated assignment of taxonomy categories to reviews with a need for explanation and the resulting assignment to an internal group in a company structure and an explanation were evaluated. Software for scraping was developed for the Google Play Store and Apple App Store to provide the data basis for the reviews. The reviews are assigned to a taxonomy category using a word and phrase filtering method with the help of additional software and then checked manually. Based on the assignment of the taxonomy category, the software assigns a reference point (person or team) to the reviews and, using an API from the Graphmasters GmbH support website and answers from the Google Play Store, a source. Through interviews and a survey with the company Graphmasters GmbH, the results could be evaluated and tested in practice. The evaluation shows that an assignment to one reference point proves to be difficult, as explanatory needs are answered intuitively in practice and rarely follow fixed patterns. Therefore, an assignment is made in descending order, based on the most likely applicable reference points, and no individual reference point is assigned.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation 1
1.2 Problemstellung 1
1.3 Lösungsansatz 2
1.4 Struktur der Arbeit 2

2 Grundlagen **3**
2.1 Erklärungsbedarf 3
2.1.1 Abgrenzung: Erklärbarkeit in der KI 4
2.1.2 Expliziter und impliziter Erklärungsbedarf 4
2.2 Taxonomie zur Kategorisierung von Erklärungsbedarf 4
2.3 Evaluationsmetriken 7
2.3.1 Software 7
2.3.2 Interrater-Reliabilität 9
2.3.3 Validität 10
2.4 Begriffe 10

3 Anforderungen **13**
3.1 Stakeholder 13
3.2 Funktionale Anforderungen 14
3.2.1 Review Scraper 14
3.2.2 Taxonomie Zuweiser 15
3.3 Nichtfunktionale Anforderungen 16
3.3.1 Review Scraper 16
3.3.2 Taxonomie Zuweiser 17
3.4 Priorisierung der Anforderungen 17

4 Konzept **19**
4.1 Studienkonzept 19
4.2 Wörterfilterkonzept 21
4.3 Interviewkonzept 22

5 Studiendesign **23**
5.1 Forschungsfragen 23
5.2 Datensatzerstellung 24
5.2.1 Play Store 25
5.2.2 App Store 25
5.2.3 Kombinierung der Appreviews 26
5.2.4 Zusammensetzung des Datensatzes 26
5.2.5 Bedienung der Software zum Scrapen von Appreviews . 26
5.3 Datensatzanalyse 28
5.3.1 Expliziter und impliziter Erklärungsbedarf 28
5.3.2 Taxonomiekategorienzuordnung 30
5.3.3 Bezugspunkt 32
5.3.4 Quelle 32
5.3.5 Bedienung der Software zur Zuordnung des Bezugspunktes und der Quelle 32
5.4 Evaluation durch Interviews und Umfragen 35

6 Ergebnisse **37**
6.1 Auswertung der Filtermethode 37
6.1.1 Erklärungsbedarf erkennen 37
6.1.2 Taxonomiekategorie anhand von einem Wörterfilter erkennen 38
6.2 Ergebnisse der Interviews und Umfrage 39
6.2.1 Taxonomiekategorienzuordnung 39
6.2.2 Teamzuordnungen 41
6.3 Zuordnung des Bezugspunktes 41
6.4 Zuordnung der Quelle 44

7 Verwandte Arbeiten **45**
7.1 Erklärungsbedarfe erkennen 45
7.2 Erklärungsbedarfe beantworten 46
7.3 Erklärungen und Erklärungsbedarf in eine Taxonomie einordnen 47
7.4 Abgrenzung zu verwandten Arbeiten 47

8 Diskussion **49**
8.1 Beantwortung der Forschungsfragen 49
8.2 Interpretation der Ergebnisse 52
8.2.1 Taxonomieerweiterung 52
8.2.2 Ergebnisse der Studie 53

9 Validität **55**
9.1 Threads of Validity 55
9.1.1 Construct Validity 55
9.1.2 Internal Validity 55
9.1.3 Conclusion Validity 56
9.1.4 External Validity 56

10 Zusammenfassung und Ausblick **57**
10.1 Zusammenfassung 57
10.2 Ausblick 58

A Software Interaktion **59**
A.1 Screenshots Review Scraper 59
A.1.1 Review Scraper - ohne Konfigurationsdatei 59
A.1.2 Review Scraper - mit Konfigurationsdatei 60
A.1.3 Review Scraper - alle Reviews scrapen 61
A.2 Screenshots Taxonomie Zuordner 62
A.2.1 Taxnomie Zuordner - Eingabefehler 62
A.2.2 Taxnomie Zuordner - Normaler Durchlauf 62
A.2.3 Taxnomie Zuordner - Parameterbeispiel anhand von - assign 63
A.3 Konfigurationsdatei Beispiel 64

B Beispiele Reviews und Filter **65**
B.1 Redewendungen Filter 65
B.1.1 Englische Redewendungen 66
B.1.2 Deutsche Redewendungen 67
B.2 Wörtergruppierung 68
B.2.1 Akzeptierte Gruppierungen 68
B.2.2 Verworfene Gruppierungen 68
B.2.3 Gefilterte Wörter 68
B.3 Wörter-Taxonomiekategorie Zuordnung 69

C Interview und Evaluation **71**
C.1 Interview Guidelines 71
C.1.1 Onlineinterview Beispiel 72

Kapitel 1

Einleitung

1.1 Motivation

Im Software- und Requirements Engineering ist eine nicht-funktionale Systemanforderung einer Software die Erklärbarkeit [1][2], die einen kritischen Aspekt der Softwarequalität darstellt [3]. Die Erklärbarkeit ist ein noch wenig erforschtes und aktuelles Thema im Bereich des Software- und Requirements Engineerings [4]. In vergangenen Arbeiten [5][6] wurden bereits automatisierte Verfahren entwickelt, um Erklärungen für Erklärbarkeitsanforderungen zu formulieren. Zur besseren Aufschlüsselung von Erklärungsbedarfen kann eine Taxonomie [7][8][9] verwendet werden. Eine Taxonomie lässt Reviews in einzelne Kategorien unterteilen, die den Eigenschaften des Reviews entsprechen. Diese betreffen beispielsweise systemspezifische oder systemunspezifische Elemente [8]. Im Folgenden wird zur Auswertung der Erklärungsbedarfe die Taxnomie von Droste et al. [8] verwendet. Ein Konzept zur Bestimmung der Informationsquellen wird im Rahmen dieser Arbeit entwickelt und auf das Unternehmen Graphmasters GmbH[1] angewendet und evaluiert. Zur differenzierteren Betrachtung der Informationsquelle wurde diese in Bezugspunkt und Quelle unterteilt und soll in der Arbeit durch einen semiautomatischen Prozess ermittelt werden.

1.2 Problemstellung

In der Softwareentwicklung kommt es nach der Veröffentlichung eines Produktes zu Erklärungsbedarfen von Nutzern bei der Verwendung der Software. Nutzer formulieren in Form von Reviews in App-Stores oder auf anderen Wegen (E-Mail, Support-Hub, Telefon, etc.) ihre Fragen oder

[1] https://www.graphmasters.net/

auftretenden Probleme mit der Software. Die App-Stores von *Google* und *Apple* stellen den größten Marktanteil dar [10]. Die Beantwortung der Erklärungsbedarfe erfordert Zeit und eine individuelle Behandlung der Fragen. Der dafür verwendete Zeitaufwand wirkt sich auf Kosten für das Unternehmen aus [11]. Eine vollautomatisierte Behandlung von Reviews in App-Stores ist aufgrund der Komplexität und der geringen Datenmenge schwer umsetzbar [12].

1.3 Lösungsansatz

Das Ziel der Arbeit ist es, den Prozess der Behebung von Erklärungsbedarfen zu vereinfachen und zu automatisieren. Um das Ziel zu erreichen, wurden mithilfe einer Taxonomie Appreviews in eine Taxonomiekategorie eingeordnet. Mithilfe der Kategorie wurde dem Review ein Bezugspunkt zugeordnet. Die Datenbasis für die Zuordnung der Taxonomiekategorie an einen Bezugspunkt wird durch die Interviews und einer Umfrage mit dem Unternehmen *Graphmasters* erstellt. Durch eine API der Support-Webseite von *Graphmasters* und Antworten aus dem Google Play Store wurde eine Quelle für das Review ermittelt. Das Verfahren soll semiautomatisch funktionieren. Da Appreviews vielfältig und individuell sind und ein Datensatz für eine spezielle App meist nicht groß genug ist, ist nach jedem Schritt eine weitere Überprüfung, durch einen Anforderungsengineer, vorgesehen.

1.4 Struktur der Arbeit

Die Arbeit ist unterteilt in zehn Kapitel. Das Kapitel 1 beinhaltet die Einleitung in das Thema der Arbeit und gibt einen Überblick über die verwendeten Lösungsansätze und Forschungsfragen. Im Kapitel 2 werden die Grundlagen beschrieben, die aus anderen Arbeiten hervorgehen und in der Arbeit verwendet werden. Im folgenden Kapitel 3 werden die Anforderungen an die Software zur automatischen Zuordnung der Antwort des Erklärungsbedarfs an Bezugspunkt und Quelle erklärt. Das Kapitel 4 beschreibt die entwickelten Konzepte zur Findung von Bezugspunkt und Quelle eines Erklärungsbedarfs. Im Kapitel 5 folgt das Design der Studie, die in Datensatzerstellung, Datensatzanalyse und Evaluation durch Interviews unterteilt ist. Das Kapitel 6 behandelt die Ergebnisse aus der Studie. Kapitel 7 stellt die verwandten Arbeiten zu dieser Bachelorarbeit dar. Das Kapitel 8 gibt die abschließende Diskussion wieder. Im Kapitel 9 wird die Validität der Studie belegt. Das Kapitel 10 schließt mit einer Zusammenfassung und einem Ausblick ab.

Kapitel 2

Grundlagen

In diesem Kapitel werden Begriffe wie Erklärungsbedarf und Taxonomie erläutert. Des Weiteren werden Methoden wie die Datensatzerstellung, die Interviewdurchführung und die Wörterbuchmethode erklärt.

2.1 Erklärungsbedarf

Der Begriff „Erklärungsbedarf" stammt aus dem Requirements Engineering und ist darin ein Teilgebiet der Erklärbarkeit. Die Erklärbarkeit ist eine nicht funktionale Systemanforderung, da sie nur der Nutzerzufriedenheit dient und nicht systemrelevant ist. Aus Chazette et al. [13] geht eine Definition hervor, die auf den Inhalt dieser Arbeit angepasst wurde:

> *Eine Navigations-App ist in Bezug auf die Routenführung der Navigations-App relativ zu einem Nutzer im Kontext der Handhabung im Straßenverkehr nur dann erklärbar, wenn es einen Supportteammitarbeiter gibt, der es dem Nutzer durch Angabe eines Supportartikels oder einer Antwort in einem App-Store ermöglicht, die Routenführung von der Navigations-App bei der Handhabung im Straßenverkehr zu verstehen.*

Erklärbarkeit in moderner Software nimmt starken Einfluss auf die Kundenzufriedenheit. Je besser einem Nutzer eine Software erklärt wird, desto höher ist seine Zufriedenheit bei der Nutzung [14]. In Appreviews wird die Zufriedenheit durch eine Sternemetrik von 1-5 Sternen angegeben. Ein Stern entspricht hierbei einer schlechten Bewertung und 5 Sterne einer guten. Daher ist der Erfolg einer Software, die auf einer Plattform mit öffentlicher Bewertungsfunktion zur Verfügung gestellt wird, abhängig von guter Erklärbarkeit der Software.

2.1.1 Abgrenzung: Erklärbarkeit in der KI

Erklärbarkeit im Softwareengineering wird oft im Zusammenhang mit Künstlicher Intelligenz (KI) verwendet. Der Begriff der Erklärbarkeit wird in dieser Arbeit als allgemeine Software-Erklärbarkeit verwendet und nicht als KI-Erklärbarkeit. KI-Erklärbarkeit ist ein Teil der Software-Erklärbarkeit und beschränkt sich auf die Aspekte der Verständlichkeit, Transparenz, Effizienz und Vertrauenswürdigkeit [15]. Da es sich in der Arbeit um Erklärungsbedarfe von individuellen Reviews einer App handelt, kann jedes der Kriterien der Erklärbarkeit von Deters [16] auftreten.

2.1.2 Expliziter und impliziter Erklärungsbedarf

Erklärungsbedarf wird in dieser Arbeit in zwei Kategorien, implizit und explizit, unterteilt. Droste et al. [8] verwenden noch eine weitere Kategorie „unspezifische Erklärung", die in dieser Arbeit eliminiert werden konnte, da „unspezifische Erklärungen" anhand von Triggerwörtern eindeutig zugeordnet werden konnten oder durch Abgleich der Meinung von mehreren Anforderungsengineers bestimmt werden konnten.

Expliziter Erklärungsbedarf lässt sich durch einen klar formulierten Wunsch zur Klärung einer Sachlage erkennen [8]. Dies kann durch Triggerwörter wie „warum", „wie" oder auch „was" erkannt werden oder durch Formulierungen wie „Bitte erkläre ..." oder „Helfen Sie mir bitte ..." (siehe B.1.2).

Impliziter Erklärungsbedarf hingegen ist schwieriger zu identifizieren, da der Nutzer nicht direkt nach einer Klärung der Sachlage fragt [8]. Bei implizitem Erklärungsbedarf gibt es wenige eindeutige Triggerwörter, wie *bizarr*, *ahnungslos* oder *unklar*. Impliziter Erklärungsbedarf kann hauptsächlich aus dem Kontext heraus erkannt werden oder über Formulierungen wie „Ich verstehen nicht warum ...[sic]" oder auch „Es funktioniert nicht ..." (siehe B.1.2).

2.2 Taxonomie zur Kategorisierung von Erklärungsbedarf

Erklärungsbedarf lässt sich zur Differenzierung voneinander in verschiedene Kategorien einordnen [8]. Es gibt für einzelne Kategorien zum Teil auch noch Oberkategorien, worunter sich diese sammeln und vereinfachen lassen. In dieser Arbeit wurde die Taxonomie von Droste et al. [8] noch um drei weitere Kategorien, „Business", „Metainformation" und „Feature Fragen",

erweitertet, um die Reviews spezifischer zu unterteilen. Der Grund der Erweiterung lässt sich im Kapitel 8.2.1 nachlesen.

Die Taxonomie lässt sich in folgende Kategorien unterteilen:

Interaktion:

Themen, die sich auf die Interaktion des Benutzers mit der Software beziehen.

Operation: Verwendung von bestimmten Funktionen des Systems

Beispiel: „Wie kann ich die PKW-Einstellungen ändern?"

Navigation: Unklarheiten, die während der Benutzung auftreten können

Beispiel: „Wo finde ich die Einstellung der Route?"

Einführung: Wenn angefragt wird, welche Schritte erforderlich sind, um zum Ziel zu gelangen

Beispiel: „Kann mir jemand eine Anleitung geben, wie ich die Navigation starte?"

Systemverhalten:

Unklarheiten bezüglich des Verhaltens des Systems

Unerwartetes Systemverhalten: Unerklärliches Verhalten der Software für den Benutzer

Beispiel: „Warum vibriert die App manchmal?"

Bugs/Abstürze: Fehlermeldungen oder Aufhängen/Abstürzen der App

Beispiel: „Warum hängt sich die App immer bei der Zieleingabe auf?"

Algorithmus: Fragen wie das System das Ergebnis ermittelt
Beispiel: „Warum wird diese Route gewählt?"

Konsequenzen: Unklarheiten bei Auswirkung verschiedener Aktionen
Beispiel: „Was passiert, wenn ich falsch abbiege?"

Domainwissen:

Fragen zu indirekten Systemaspekten

Begrifflichkeiten: Unklare Abkürzungen oder Fachbegriffe
Beispiel: „Was bedeutet StVO?"

Systemspezifische Elemente: Verschiedene Versionen, Apps oder Modelle
Beispiel: „Wie unterscheidet sich die Premium-Version von der kostenlosen?"

Geheimhaltung und Sicherheit:

Fragen zu Datensicherheit und Vertraulichkeit

Geheimhaltung: Was mit den gesammelten Daten gemacht wird
Beispiel: „Werden meine Daten an Dritte weitergeleitet?"

Sicherheit: Sicherheitslücken in der App
Beispiel: „Kann das GPS-Signal von anderen gehackt werden?"

Kategorien ohne Oberkategorie:

Designentscheidungen: Fragen zu der Benutzeroberfläche
Beispiel: „Warum werden rote und nicht blaue Pfeile verwendet?"

Business: Fragen, die nicht das System direkt betreffen, sondern in Bezug auf den Anbieter fern vom Softwarebereich sind

Beispiel: „Wie kommt der Preis der Pro-Version zustande?"

Metainformationen: Wenn keine oder mehrere der anderen Kategorien betroffen sind

Beispiel: „Warum sind andere Autofahrer immer am Hupen bei Stau?"

Feature Fragen: Fragen zu kommenden, ausstehenden oder wünschenswerten Features

Beispiel: „Warum gibt es Feature XY nicht?"

2.3 Evaluationsmetriken

2.3.1 Software

Zur Evaluation der Softwareergebnisse werden Evaluationsmetriken zur Auswertung verwendet. Die Metriken geben Auskunft darüber, wie zutreffend die Zuordnung des Ortes und der Quelle, die von der Software ermittelt wurde, ist. Die verwendeten Metriken sind Precision, Recall, Accuracy und der F-score [17] [18].

Definitionen:

tp = true positive: Gibt die Werte an, die positiv gelabelt wurden und auch positiv sind.

fp = false positive: Gibt die Werte an, die positiv gelabelt wurden, allerdings negativ sind.

tn = true negative: Gibt die Werte an, die negativ gelabelt wurden und auch negativ sind.

fn = false negative: Gibt die Werte an, die negativ gelabelt wurden,

allerdings positiv sind.

Precision

Gibt das Verhältnis zwischen den positiv gelabelten Daten, die tatsächlich positiv sind, und den gesamten positiv gelabelten Daten an. Dabei wird ermittelt, welcher Prozentsatz von den gesamten positiv gelabelten Daten, korrekt zugeordnet wurde.

$$\text{Precision: } P = \frac{tp}{tp + fp} \tag{2.1}$$

Recall

Zeigt das Verhältnis von Daten, die positiv gelabelt wurden, zu Daten, die tatsächlich positiv sind. Hierbei steht im Fokus wie viele tatsächlich positive Daten, durch Labeln im Vorfeld, gefunden werden konnten.

$$\text{Recall: } R = \frac{tp}{tp + fn} \tag{2.2}$$

Accuracy

Gibt die allgemeine Anzahl an richtigen Voraussagen über alle vorhergesagten Ereignisse an.

$$\text{Accuracy: } A = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.3}$$

F-score

Dieser Wert bildet einen ausgewogenen Mittelweg von Precision und Recall. Der F-score kann hierbei beliebig mit dem β Wert angepasst werden, wodurch Precision und Recall eine differenzierte Gewichtung erhalten (2.4). Bei dem F1-score liegt Precision und Recall eine gleiche Gewichtung zugrunde (2.5).

$$\text{F-score} : F_\beta = (1 + \beta) \cdot \frac{P \cdot R}{\beta \cdot P + R} \tag{2.4}$$

$$\text{F1-score} : F_1 = F = 2 \cdot \frac{P \cdot R}{P + R} \tag{2.5}$$

Kappa Statistic	Stärke der Übereinstimmung
<0.00	Schlecht
0.00-0.20	Gering
0.21-0.40	Angemessen
0.41-0.60	Moderat
0.61-0.80	Substanziell
0.81-1.00	Fast Perfekt

Tabelle 2.1: Interpretation nach Landis und Koch [20]

2.3.2 Interrater-Reliabilität

Kappa Statistik

Die Kappa Statistik ist ein Maß für Interrater-Reliabilität, um die Übereinstimmung von Probanden zu untersuchen [19]. Der Maßwert liegt zwischen 0 und 1, wobei Werte <0 erreicht werden können. Wenn dies der Fall ist, dann stellt dies eine schlechtere Übereinstimmung als der erwarteten zufälligen Übereinstimmung dar. 0 stellt eine Übereinstimmung der erwarteten zufälligen Übereinstimmung dar und 1 eine vollständige Übereinstimmung der Daten (2.6). Kappa beurteilt hierbei nicht nur die reine Übereinstimmung von Probanden, sondern wertet auch die Wahrscheinlichkeit einer zufälligen Übereinstimmung in den Maßwert mit ein. Cohens Kappa wird für zwei Probanden angewendet, wohingegen Fleiss' Kappa für >2 Probanden verwendet werden kann. Landis und Koch [20] entwickelten eine Tabelle, um die Ergebnisse von Kappa einzuordnen (siehe 2.1).

$$\mathrm{K} = \frac{p_0 - p_e}{1 - p_e} \tag{2.6}$$

Cohens Kappa

Zur Analyse von zwei Probanden kann der Cohens Kappa [19] als Metrik verwendet werden. Der p_0 Wert wird aus den summierten übereinstimmenden Werten, geteilt durch die gesamten eingeschätzten Beurteilungen, errechnet (2.7). Für die erwartete Übereinstimmung p_e wird die Summe der Randhäufigkeiten verwendet (2.8).

$$p_0 = \frac{\sum_{i=1}^{z} h_{ii}}{N} \tag{2.7}$$

$$p_e = \sum_{i=1}^{z} h_{i.} h_{.i} \tag{2.8}$$

Fleiss' Kappa

Zur Analyse ab zwei Probanden kann der von Fleiss [21] entwickelte Kappa Wert als Metrik verwendet werden, da Cohens Kappa nur für zwei Probanden verwendbar ist. Hierbei beschreibt, wie beim Cohens Kappa, der p_0 Wert die relative Übereinstimmung und p_e beschreibt die Wahrscheinlichkeit einer zufälligen Übereinstimmung.

$$p_0 = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{n(n-1)} \sum_{j=1}^{k} (n_{ij}^2 - n_{ij}) \right) \tag{2.9}$$

$$p_e = \sum_{j=1}^{k} p_j^2 \tag{2.10}$$

2.3.3 Validität

Die Validität, definiert nach Lienert und Raatz [22], bezeichnet die inhaltliche Übereinstimmung einer empirischen Messung mit einem logischen Messkonzept. Die Validität gibt hierbei einen Grad an Genauigkeit zwischen 0-100% an mit dem die empirische Messung mit der tatsächlichen Messung verglichen wird. In dieser Arbeit werden die Einordnungen der Taxonomiekategorien von den Probanden der Studie und die Zuordnung der Anforderungsengineers verglichen.

2.4 Begriffe

Quelle

Wenn in dieser Arbeit von der Quelle der Antwort des Erklärungsbedarfs gesprochen wird, ist damit die bereits bestehende Antwort zum Erklärungsbedarf gemeint und wo diese lokalisiert ist. Dies kann entweder eine Antwort vom Entwickler zu dem Review sein, ein Artikel auf der Supportwebseite vom Entwickler oder eine andere Form einer schriftlich

vorliegenden Antwort.

Bezugspunkt

Wenn in dieser Arbeit von einem Bezugspunkt der Antwort des Erklärungsbedarfs gesprochen wird, ist damit ein Team oder eine Einzelperson im Unternehmen von *Graphmasters* gemeint, die über das Wissen verfügt, den Erklärungsbedarf zu beantworten. Die internen Teams im Unternehmen von *Graphmasters* teilen sich wie folgt auf: Mobile, Courier Backend, UI/UX, Traffic Management, Routing, Business, Support, Traffic Strategies und Meta. Meta beschreibt hierbei eine Einordnung, die nicht eindeutig vorgenommen werden kann oder mehrere Teams im Unternehmen betrifft.

Anforderungsengineers

Anforderungsengineers sind Experten im Gebiet des Softwareengineerings, die sich mit der Einordnung in expliziten und impliziten Erklärungsbedarf von Appreviews und des Weiteren mit der Taxonomie zur Einordnung von Appreviews mit Erklärungsbedarf auskennen.

Scraping von Reviews

Das Scraping von Reviews beschreibt in dieser Arbeit das Sammeln und Abspeichern von Reviews einer bestimmten App aus den App-Stores von *Google* und *Apple*. Der Begriff des „Scrapings" beschreibt generell alle Verfahren des Auslesens von Texten und Bildinhalten, wobei hier speziell die Reviews in den App-Stores gemeint sind.

Ground Truth

Als Ground Truth werden bei einem Datenvergleich die Daten genannt, die als korrekt angenommen werden. Mit der Basis des Ground Truth können Vergleichswerte für eine Validität von erhobenen Daten geschaffen werden.

Kapitel 3

Anforderungen

In folgendem Kapitel werden die Anforderungen an die Anwendung zur Einordnung des Erklärungsbedarfs von Appreviews in expliziten und impliziten Erklärungsbedarf sowie die Einordnung in die zugehörige Taxonomiekategorie gestellt. Dies wird durch die Umsetzung zweier Softwares gewährleistet, die separat voneinander agieren. Nachdem die Erklärungsbedarfe erkannt und in explizite, implizite und möglicher Erklärungsbedarfe eingestuft wurden, wird eine manuelle Überprüfung von Anforderungsengineers vorgenommen. Nachdem die Überprüfung abgeschlossen ist, wird die zweite Software verwendet, um den Reviews mit Erklärungsbedarf eine Taxonomiekategorie zuzuordnen und darauf basierend einen Bezugspunkt für die Antwort des Erklärungsbedarfs zu ermitteln. Des Weiteren weist die zweite Software, mithilfe einer API der Supportwebseite des Unternehmens und Reviewantworten aus dem Google Play Store, dem Review mit Erklärungsbedarf eine Quelle zu. Hierbei steht nicht die Antwort selbst im Vordergrund, sondern wo diese zu finden ist und wer diese weiß.

3.1 Stakeholder

Den vorrangigen Stakeholder stellt in dieser Arbeit die Firma Graphmasters GmbH dar, die die Evaluierung der Software durch Bestätigung der zugeteilten Bereiche (Bezugspunkt und Quelle) vornimmt. Die Software soll aber auch für jeden beliebigen anderen Stakeholder funktionieren, der das Tool zur internen Organisation der Reviews und dessen Erklärungen verwendet. Das Konzept soll weitläufig anwendbar sein und auf andere Unternehmen, die eine vergleichbare Größe wie *Graphmasters* aufweisen, mit einem Aufbau von internen Teams und einem Supportteam, das sich um die Beantwortung des Kundenfeedbacks kümmert, übertragbar sein. Dies soll durch allgemein

einstellbare Parameter ermöglicht werden, um für verschiedene Anwender konfigurierbar zu sein.

3.2 Funktionale Anforderungen

Die gestellten Anforderungen sind hierbei aufgeteilt auf zwei separate Softwares und werden folglich gesondert aufgelistet. Die Anforderungen wurden mit Absprache des Betreuers von *Graphmasters* erarbeitet.

3.2.1 Review Scraper

[R01]: Erstellung einer CSV-Datei mit Metadaten

Die Anwendung kann nach Angabe des Namens einer App und der gewünschten Anzahl an Reviews eine CSV-Datei erstellen, die die Reviews und weitere Metadaten auflistet.

[R01.1]: Alle Geräte im *Play Store* können gescraped werden

Die Anwendung kann alle im Google Play Store[1] vorhandenen Apps mit der Angabe des korrekten Namens finden und auswerten. Das verwendete Gerät der Review soll dabei keine Rolle spielen und somit von allen Geräten die Reviews heraussuchen. Die möglichen Geräte sind: Telefon, Tablet, Smartwatch, Chromebook und TV.

[R01.2]: Alle Regionen im *App Store* können gescraped werden

Die Anwendung kann alle im Apple App Store[2] vorhandenen Apps mit der Angabe des korrekten Namens und der korrekten ID finden und auswerten. Das Herkunftsland der Review soll keine Rolle bei der Auswertung haben und somit soll auch jedes Land berücksichtigt werden.

[R02]: Zusammenfügen von App-Stores

Die erstellte CSV-Datei soll sowohl alle Google Play Store als auch Apple App Store Reviews kombiniert darstellen und mit einer einzigartigen ID versehen. Die vorhandene ID des *Play Stores* soll beibehalten werden und für den *App Store* soll eine ID aus Region, iterierter Nummer und Datum

[1] https://play.google.com/store/games?hl=de
[2] https://www.apple.com/de/app-store/

erstellt werden.

[R02.1]: Auswahl der App-Stores

Wenn nur vom Google Play Store oder nur vom Apple App Store die Reviews gewünscht sind, kann der jeweilige App-Store einzeln gescraped werden.

[R03]: Wörterbuchsuche nach Erklärungsbedarf

Die Anwendung soll mithilfe eines Wörterbuchs Erklärungsbedarf erkennen und markieren. Der Erklärungsbedarf soll gegliedert werden in expliziten, impliziten und möglichen Erklärungsbedarf. Der mögliche Erklärungsbedarf wird hier von allgemeinen Triggerwörtern erkannt, aber weist nicht genug spezifische Merkmale auf, um zu explizitem oder implizitem Erklärungsbedarf zugeordnet zu werden. Dies soll außerdem in der erstellten CSV-Datei eingetragen werden. Diese neu erstellte Spalte wird folglich von Anforderungsengineers manuell überprüft und der mögliche Erklärungsbedarf wird durch expliziten, impliziten oder keinen Erklärungsbedarf eliminiert.

[R04]: Sammlung von Metadaten

Die Anwendung soll neben den Reviews auch weitere Metadaten sammeln. Dazu gehören Daten, IDs, Namen der Ersteller, Titel und Ratings. Die Titel werden hierbei nur bei *App Store* Reviews gesammelt, da der *Play Store* keine Review-Titel hat. IDs werden nur aus dem *Play Store* gesammelt und für den *App Store* künstlich erstellt.

3.2.2 Taxonomie Zuweiser

[R05]: Taxonomiekategorienzuordnung durch Triggerwörer

Nachdem die Reviews aus den App-Stores gescraped wurden, sollen diese anhand einer Wörterliste automatisch Taxonomiekategorien zugeordnet werden. Die Wörterliste mit den Taxonomiekategoriezuordnungen soll hierbei für einzelne Apps vorab angepasst werden. Die zugeordneten Taxonomiekategorien sollen anschließend von Anforderungsengineers noch einmal überprüft werden.

[R06]: Bezugspunkt des Erklärungsbedarfs bestimmen

Die zugeordnete Taxonomiekategorie soll anschließend verwendet werden, um anhand dieser den Bezugspunkt für die Antwort des Erklärungsbedarfs automatisch zu bestimmen. Die Zuteilung von Taxonomiekategorien zu einer Person oder einem Team im Unternehmen wird in einer eigenen Datei festgelegt und der Software als Input gegeben. Der Bezugspunkt wird nach Zuordnung der Software zur Überprüfung noch einmal von einem Developer der jeweiligen Firma bestätigt.

[R07]: Quelle des Erklärungsbedarfs bestimmen

Die Quelle des Erklärungsbedarfs soll automatisch mithilfe einer Support-API und der Antwort von einem Unternehmen auf die Appreviews in den App-Stores gefunden werden. Die Quelle wird zur Überprüfung noch einmal von einem Developer der jeweiligen Firma bestätigt oder, sollte noch keine vorhanden sein, wird eine Antwort auf den Erklärungsbedarf neu verfasst.

3.3 Nichtfunktionale Anforderungen

3.3.1 Review Scraper

[NR01]: Optimierung der Suche nach Regionen im App Store

Zur Optimierung der Geschwindigkeit soll die Suche nach Regionen im Apple App Store über mehrere Threads realisiert werden. Da alle 154 Regionen[3] einzeln auf Verfügbarkeit der App und darauffolgender Verfügbarkeit von Reviews überprüft werden müssen, nimmt dies eine lange Vorbearbeitungszeit in Anspruch, bevor die Reviews gescraped werden können.

[NR02]: Anzahl an Reviews einschränken

Die Anzahl an gescrapten Reviews soll einstellbar sein. Dabei soll angegeben werden können, ob alle Reviews gescraped werden sollen. Wird dies verneint, kann die Anzahl an gewünschten Scrapes angegeben werden. Die Reviews können hierbei in Zehnerschritten gescraped werden.

[3] https://gist.github.com/daFish/5990634

[NR03]: Meldungen für den User

Die Anwendung gibt über die Konsole Angaben über den Fortschritt des Bearbeitungsdurchlaufs. Dabei soll der derzeitige Schritt im Terminal angezeigt werden und, wo es möglich ist, auch einen „Ladebalken".

[NR04]: Konfiguration zum Scrapen der Appreviews über eine config.ini Datei

Als Erleichterung der Bedienung soll eine Konfigurationsdatei erstellt werden können, in der der Name der App im *Play Store* wie auch im *App Store* abgespeichert werden soll und auch die ID der App im *App Store*. Zusätzlich soll konfigurierbar sein, ob alle Reviews der jeweiligen App-Stores gescraped werden sollen oder nur eine vordefinierte Anzahl an Reviews pro App-Store.

3.3.2 Taxonomie Zuweiser

[NR05]: Speicherung der Wörterliste zur Taxonomiekategorieneinordnung

Da jede App eigene Triggerwörter für die jeweiligen Taxonomiekategorien vorweist, soll die Zuordnung der Triggerwörter gespeichert werden können, um bei erneutem Start der Software eine Neuzuordnung zu verhindern. Hierbei soll vorab geprüft werden, ob bereits Listen bestehen. Sollten noch keine Zuweisungen von Triggerwörtern bestehen, wird die Frage zur Verwendung von bereits vorhandenen Listen nicht eingeblendet.

[NR06]: Normierung von Wörtern

Damit bei der Wörterzuweisung ähnliche Wörter unter einen Begriff zusammengefasst werden, um die Zuteilung zu einer Taxonomiekategorie zu verbessern, soll eine Normierung und Gruppierung der Wörter vorgenommen werden.

3.4 Priorisierung der Anforderungen

Die nachfolgenden Tabellen 3.1 und 3.2 stellen die Anforderungspriorisierungen der zuvor definierten funktionalen und nichtfunktionalen Anforderungen für den Review Scraper und den Taxonomie Zuweiser dar.

Anforderung	Priorisierung
[R01]	hoch
[R02]	hoch
[R03]	hoch
[R04]	hoch
[R01.1]	mittel
[R01.2]	mittel
[NR01]	mittel
[NR03]	mittel
[R02.1]	gering
[NR02]	gering
[NR04]	gering

Tabelle 3.1: Priorisierung der Anforderungen Review Scraper, sortiert

Anforderung	Priorisierung
[R05]	hoch
[R06]	hoch
[R07]	hoch
[NR05]	hoch
[NR06]	hoch

Tabelle 3.2: Priorisierung der Anforderungen Taxonomie Zuweiser, sortiert

Kapitel 4

Konzept

Im Rahmen der Bachelorarbeit wurde ein Konzept von dem Sammeln der Reviews bis hin zur Findung des Bezugspunktes und der Quelle für den Erklärungsbedarf der Reviews entwickelt. Das Konzept spiegelt den Arbeitsablauf der Durchführung des Vorhabens wider. Hierbei werden einmal das gesamte Studienkonzept in einem Überblick dargestellt und zwei Teilbereiche des Gesamtkonzeptes, der Wörterfilter und das Interview zur Evaluation, genauer beleuchtet. Die Konzepte werden mithilfe der Flow-Methode von Stapel und Schneider [23] erklärt.

4.1 Studienkonzept

Das Gesamtkonzept der Arbeit, das in Abbildung 4.1 zu sehen ist, beginnt mit einer Datengrundlage in Form von Appreviews aus dem Google Play Store oder Apple App Store. Die Reviews werden im nächsten Schritt auf Erklärungsbedarf untersucht. Wenn Erklärungsbedarfe mithilfe eines Filters oder durch nachfolgende Überprüfung eines Anforderungsengineers gefunden werden, werden diese anschließend in impliziten oder expliziten Erklärungsbedarf eingeordnet. Die Liste aus Reviews mit Erklärungsbedarf wird darauffolgend in die Taxonomiekategorien zugeordnet. Dies geschieht automatisch über einen Wörter- und Redewendungsfilter, der detaillierter im Kapitel 4.1 beschrieben wird, und darauffolgend zur Überprüfung durch Anforderungsengineers. Im letzten Schritt wird eine Liste aus Reviews mit zugeordneten Taxonomiekategorien durch einen teilautomatisierten Prozess mithilfe von anschließenden Evaluationsinterviews, Bezugspunkten und Quellen für die Beantwortung der Erklärungsbedarfe ausgegeben. Die Evaluationsinterviews werden vertieft noch einmal im Unterkapitel 4.3 erläutert.

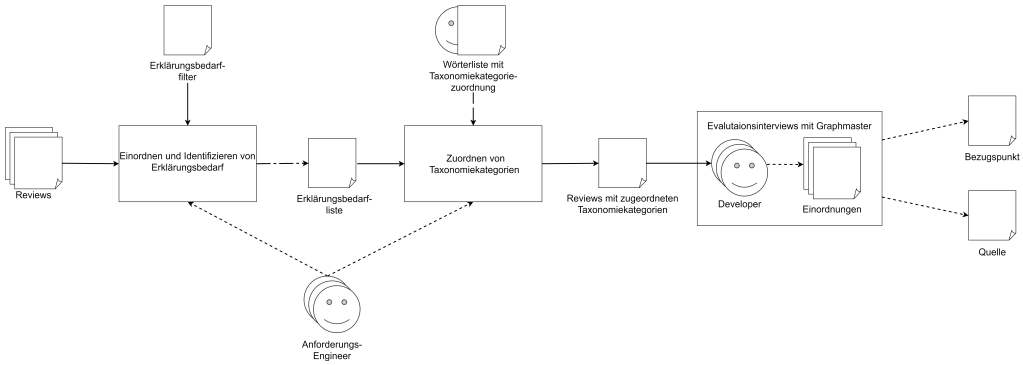

Abbildung 4.1: Gesamtes Studienkonzept in Flow-Methode [23]

4.2 Wörterfilterkonzept

Der Wörterfilter wird mit jedem neu hinzukommenden Review neu evaluiert. Es werden die einzelnen Wörter aus den gesammelten Reviews separiert und gezählt. Diese daraus entstehende Wörterliste wird mithilfe eines bereits bestehenden Wörterfilters dezimiert. Der Wörterfilter enthält Pronomen, Artikel und weitere Wörter, die von Anforderungsengineers identifiziert wurden und nicht einer Taxonomiekategorie zuzuordnen sind (siehe B.2). Die daraus entstehende Wörterliste wird dahingehend überprüft, ob nicht zuzuordnende Wörter weiterhin vorhanden sind. Sollten weitere Wörter gefunden werden, werden diese dem Wörterfilter hinzugefügt und der Vorgang wird wiederholt. In der daraus resultierenden Wörterliste werden den Wörtern einzelne Taxonomiekategorien zugeordnet, die auf die jeweilige Kategorie hinweisen. Diese Wörterliste wird zur automatischen Erkennung von Taxonomiekategorien der Reviews verwendet und soll den Arbeitsprozess verschnellern und erleichtern.

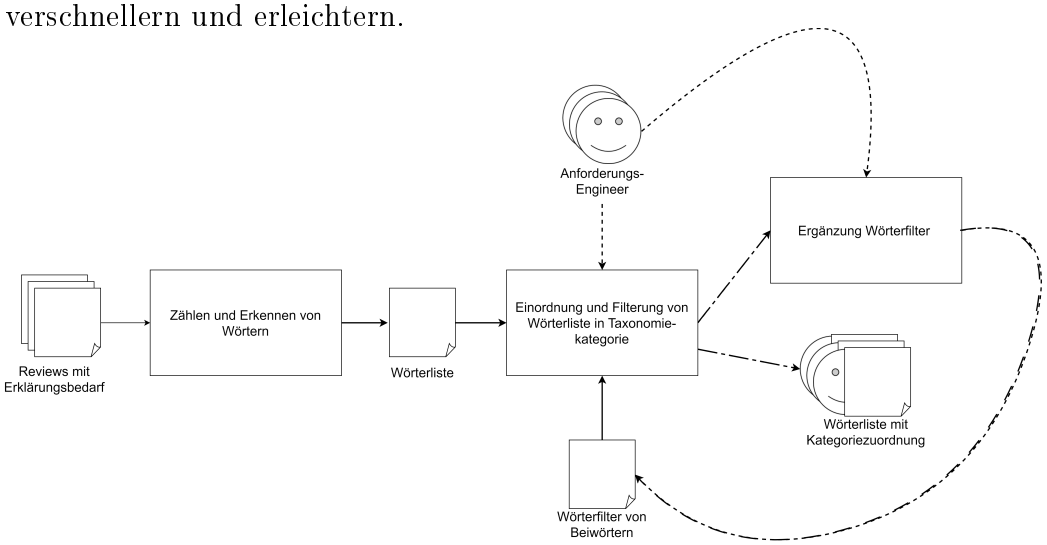

Abbildung 4.2: Wörterfilterkonzept in Flow-Methode [23]

4.3 Interviewkonzept

Das Interviewkonzept dient der Evaluation der Ergebnisse. Nachdem jedes Review in eine Taxonomiekategorie eingeteilt wurde, werden durch jeweils einzelne Verfahren automatisch ein Bezugspunkt und eine Quelle für die Beantwortung des Erklärungsbedarfs zugeordnet. Die Zuordnung des Bezugspunktes geschieht anhand der Taxonomiekategorienzuordnung der Reviews. Eine Taxonomiekategorie wird hierbei einem jeweiligen Team im Unternehmen zugeordnet und diese werden anschließend automatisch den Reviews zugeordnet. Die Ermittlung der Quelle geschieht hierbei über zwei verschiedene Methoden. Zuerst wird auf der Support-Webseite mithilfe einer API ein Artikel gesucht, der die Thematik des Reviews beantwortet. Sollte kein Supportartikel vorhanden sein oder gefunden werden können, wird die verfasste Antwort des Unternehmens im jeweiligen App-Store verwendet. Die automatisch ermittelten Bezugspunkte und Quellen wurden zur Überprüfung durch Interviews mit dem Unternehmen evaluiert. Nach dem Interview mit den Developern des Unternehmens sind, solange sie gefunden werden konnten, ein Bezugspunkt und eine Quelle für die jeweiligen Reviews mit Erklärungsbedarf ermittelt und die Zutrefflichkeit der Software evaluiert worden.

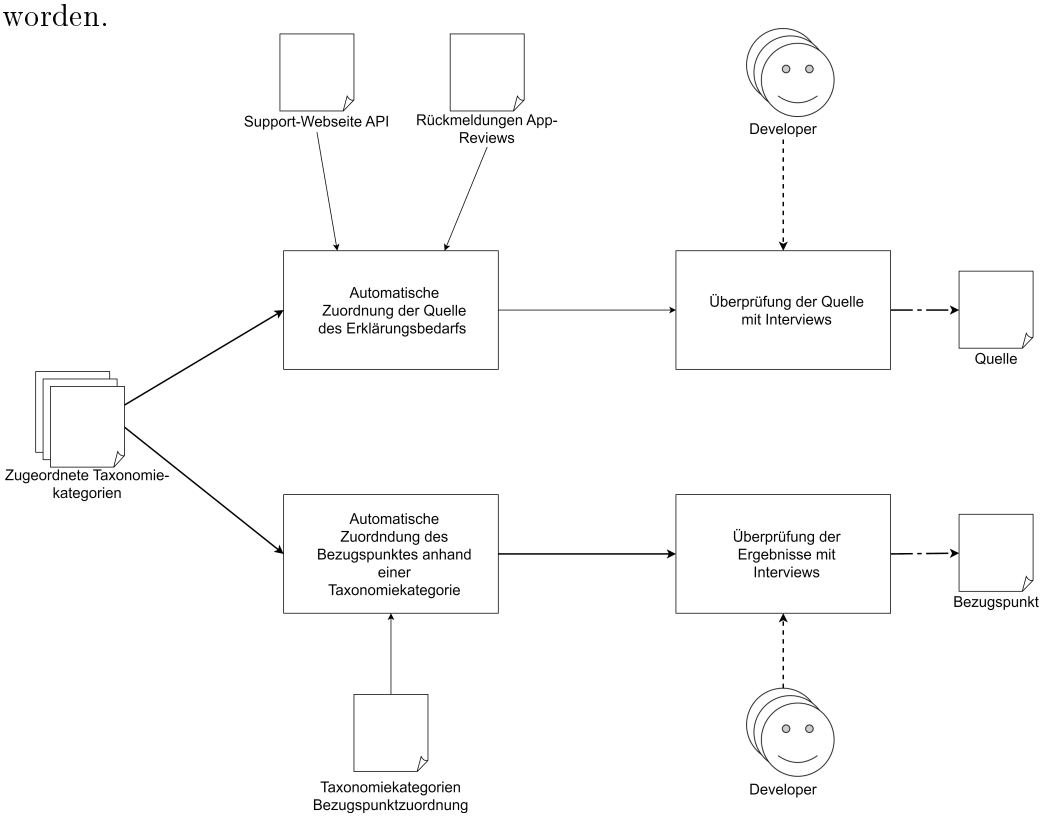

Abbildung 4.3: Interviewkonzept in Flow-Methode [23]

Kapitel 5

Studiendesign

Zu Beginn des Kapitels werden die Forschungsfragen der Studie erläutert (siehe 5.1). Im ersten Schritt der Studie wurde ein Datensatz von Appreviews aus dem Google Play Store und Apple App Store der Navigations-Apps von *Graphmasters* erstellt (siehe 5.2). Das Kapitel 5.3 beschreibt die anschließende Analyse des Datensatzes. Zur Evaluierung der Datenanalyse wurde ein Interview mit dem Unternehmen Graphmasters durchgeführt und ausgewertet (siehe 5.4). Das Konzept der Studie ist unter dem Kapitel 4.1 zu finden.

5.1 Forschungsfragen

RQ1.1: Mit welcher Genauigkeit können Nutzerreviews mit Erklärungsbedarf einem Bezugspunkt für dessen Behebung von Erklärungsbedarf auf Basis einer Taxonomiekategorie zugeordnet werden?

RQ1.2: Mit welcher Genauigkeit können Nutzerreviews mit Erklärungsbedarf einer Quelle für deren Behebung von Erklärungsbedarf auf Basis einer Taxonomiekategorie zugeordnet werden?

RQ1.1 und **RQ1.2** stellen die übergeordneten Ziele der Studie dar. Durch Interviews und eine Umfrage mit *Graphmasters* wird die Zuordnung zu einem Bezugspunkt und einer Quelle evaluiert. Zur Kategorisierung der Reviews werden mithilfe einer Taxonomie die Erklärungsbedarfe in den Reviews differenziert. Mithilfe dieser Einordnung wird geprüft, ob Rückschlüsse auf den Bezugspunkt und die Quelle geschlossen werden können.

RQ2.1: Kann jedem Nutzerreview mit Erklärungsbedarf ein eindeutiger Bezugspunkt zugeordnet werden?

RQ2.2: Kann jedem Nutzerreview mit Erklärungsbedarf eine eindeutige Quelle zugeordnet werden?

In den Forschungsfragen **RQ2.1** und **RQ2.2** wird die allgemeine Frage gestellt, ob ein eindeutiger Bezugspunkt und eine eindeutige Quelle zu den jeweiligen Erklärungsbedarfen vorhanden sind. Diese Forschungsfragen zielen darauf hinaus, Randfälle zu betrachten.

RQ3.1: Unter welchen Voraussetzungen, die ein Unternehmen bietet, kann ein Bezugspunkt für ein Nutzerreview, bestehend aus einer Person oder einem Team, angegeben werden?

RQ3.2: Unter welchen Voraussetzungen, die ein Unternehmen bietet, kann eine Quelle für ein Nutzerreview, bestehend aus Information für die Erfüllung der Erklärbarkeitsanforderung, angegeben werden?

Mit den Forschungsfragen **RQ3.1** und **RQ3.2** wird untersucht, welche Vorraussetzungen ein Unternehmen bieten muss, um Erklärungsbedarfe von Nutzerreviews zu den Apps des Unternehmens in einen Bezugspunkt und eine Quelle einzuordnen.

RQ4: Wie viel echten Erklärungsbedarf kann eine Firma adressieren?

Mit der abschließenden Forschungsfrage **RQ4** soll in der Praxis festgestellt werden, wie viel von echtem Erklärungsbedarf eine Firma tatsächlich beantworten kann.

5.2 Datensatzerstellung

Der Datensatz der Studie wurde aus Nutzerreviews der *Nunav Navigation App*[1], *Nunav Courier App*[2], *Nunav Truck App*[3] und der *Nunav Bus App*[4]

[1] https://play.google.com/store/apps/details?id=com.nunav.play&hl=de
[2] https://play.google.com/store/apps/details?id=com.nunav.logistics&hl=de
[3] https://play.google.com/store/apps/details?id=com.nunav.truck&hl=de
[4] https://play.google.com/store/apps/details?id=com.nunav.bus&hl=de

aus dem Google Play Store und Apple App Store erstellt. Es wurden hierbei neben der vorrangigen *Nunav Naviagion App*, drei weitere Apps von *Graphmasters* zur Datensatzerstellung verwendet, da diese eine hohe Ähnlichkeit aufweisen. Die betrachteten Apps werden zur Navigation im Straßenverkehr verwendet, die als Unterschiede marginale visuelle Anpassungen, Einstelloptionen und andere Routenoptimierungen aufweisen. Durch das Einbringen der *Nunav Courier App*, *Nunav Truck App* und der *Nunav Bus App* wurde der Datensatz um 231 Reviews erweitert.

Zur Beschaffung der Daten wurde eine neue Software entworfen, da öffentlich verfügbare Review-Scraper keine Vollständigkeit der Reviews gewährleisten konnten und kein Zugang zu den internen Unternehmensaccounts der App-Stores, mit Datenbasis der Apps, zur Verfügung stand. Verfügbare online Review-Scraper, die eine funktionierende API zur Verfügung stellen, sind bei mehrfacher Nutzung oder in ihrer Funktion generell eingeschränkt und nur gegen Entgelt in Vollversion erhältlich. Als Beispiel gibt es die Webseiten *SerpApi*[5], *Octoparse*[6] und *Outscraper*[7].

5.2.1 Play Store

Die neu erstellte Software basiert auf einem Selenium Webdriver[8] von Chrome, der die Google Play Store Reviews scraped. Der Scraper geht hierbei explizit jedes einzelne Gerät im *Play Store* durch, das für die App verwendet werden kann, und fügt anschließend alle Reviews in einem Pandas[9] Dataframe zusammen.

5.2.2 App Store

Für die Sammlung der Appreviews aus dem *App Store* wird eine API[10] verwendet. Hierbei wird zuerst überprüft, in welchen Regionen[11] Reviews für die App verfasst wurden. Anschließend wird jede Region, die Reviews der App enthält, einzeln über die API gescraped und anschließend in einem Dataframe gesammelt. Zur Optimierung werden die Suche der Regionen und

[5] https://serpapi.com/
[6] https://www.octoparse.de/
[7] https://outscraper.com/de/
[8] https://www.selenium.dev/documentation/webdriver/
[9] https://pandas.pydata.org/
[10] https://pypi.org/project/app-store-scraper/
[11] https://gist.github.com/daFish/5990634

das anschließende Scrapen der Regionen auf mehrere Threads ausgelagert.

5.2.3 Kombinierung der Appreviews

Die *Play Store* und *App Store* Reviews werden anschließend in einem Dataframe zusammengefasst und in einer CSV abgespeichert. Damit der Dataframe unifiziert werden kann, müssen bei den *Play Store* Reviews Titel erzeugt werden. Da die Titel in dem Datensatz nicht zur Analyse verwendet werden, werden für die jeweiligen Reviews Leerstellen bei den Titeln eingesetzt. Bei den Reviews des *App Store's* werden zur Unifizierung IDs erstellt. Die IDs setzen sich aus Region, iterierter Nummer und Datum zusammen. Wenn mehrere Datensätze von verschiedenen Apps verwendet werden, können diese in einer Microsoft Excel[12]-Tabelle für die weitere Datenanalyse hintereinander eingefügt werden.

5.2.4 Zusammensetzung des Datensatzes

Der Datensatz setzt sich zusammen aus:

	nunav navigation	nunav truck	nunav logistics	nunav bus	Gesamt
Play Store	1955	7	187	28	2177
App Store	181	0	0	8	189
Gesamt	2136	7	187	36	2366

Tabelle 5.1: Aufteilung der Appreviews auf App-Store und App

5.2.5 Bedienung der Software zum Scrapen von Appreviews

Die Software ist in Python programmiert und wird über das Terminal bedient. Nach Aufrufen der Software mit dem Kommando *python review_scraper.py* wird der Nutzer durch die Konfiguration der Software geleitet und aufgefordert, die erforderlichen Parameter zur Softwarenutzung auszuwählen (siehe A.1.1). Hierzu wird anfangs gefragt, ob eine Konfigurationsdatei geladen werden soll (siehe A.1.2). Durch das Voreinstellen einer

[12]https://www.microsoft.com/de-de/microsoft-365/excel?market=de

Konfigurationsdatei kann bei mehrfacher Reviewssuche der gleichen App oder wenigen Parameterveränderung nach wiederholtem Starten der Software ein schnellerer Vorgang gewährleistet werden. Die Abbildung 5.1 stellt ein Ablaufdiagramm für die Nutzerinteraktionen mit der Software dar.

Sollte die Konfigurationsdatei fehlerhaft sein, wird der Benutzer aufgefordert, die Daten manuell einzugeben. Die manuelle Abfrage erfolgt hierbei über das Terminal. Wenn bei der manuellen Angabe der Daten eine unrealistische oder unpassende Antwort gegeben wird, zum Beispiel wenn bei der Anzahl der zu scrapenden Reviews keine Zahl eingegeben wird, wird der Nutzer darauf hingewiesen und aufgefordert, erneut die Zahl einzugeben. Sollte der Nutzer auswählen, alle vorhandenen Reviews der eingegebenen App zu scrapen, so wird die Abfrage nach der Anzahl der zu scrapenden Reviews nicht gestellt (siehe A.1.3).

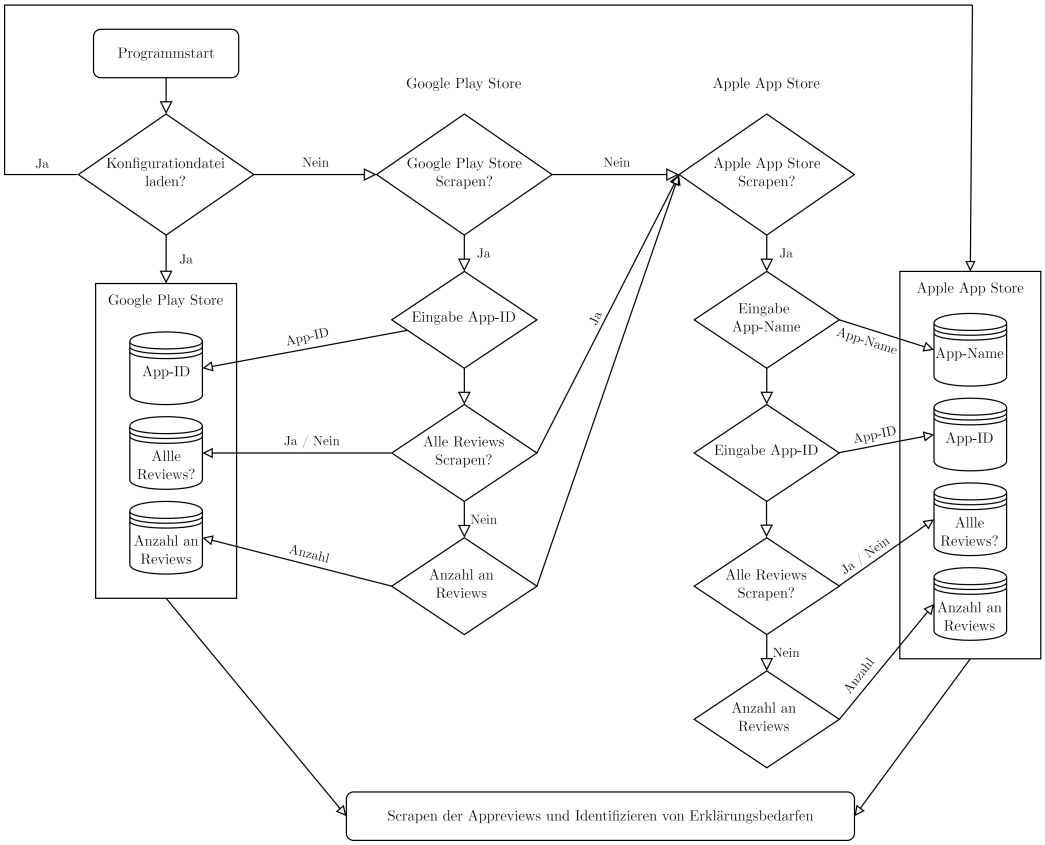

Abbildung 5.1: Software Ablaufdiagramm des *Review Scrapers*

5.3 Datensatzanalyse

Nachdem ein vollständiger Datensatz, der aus einer oder auch mehreren Apps bestehen kann, erstellt wurde, kann dieser als Input in einer weiteren Software zur Analyse verwendet werden. Der nächste Analyseschritt ist die Zuordnung des expliziten oder impliziten Erklärungsbedarfs.

5.3.1 Expliziter und impliziter Erklärungsbedarf

Der explizite und implizite Erklärungsbedarf wird mithilfe einer Wörterbuch- und Redewendungenmethode automatisch an die jeweiligen Reviews zugeordnet. Die Redewendungen und Schlagwörter, um expliziten und impliziten Erklärungsbedarf zu erkennen, entstammen aus verschiedenen Datensätzen. Für diese Arbeit wurden mehrere Datensätze kombiniert, um eine höhere Treffsicherheit für Erklärungsbedarf in Reviews zu erreichen. Hierbei gibt es für die erste Analyse zusätzlich die Einteilung „möglicher Erklärungsbedarf", die später durch zwei Anforderungsengineers geprüft werden soll. Der größte Datensatz für den Erklärungsbedarfsfilter entstammt aus der Arbeit von Kupczyk [24] mit einer Anzahl von 245 Redewendungen (siehe B.1.1). Dieser Datensatz wurde anschließend verdoppelt, indem mithilfe des Übersetzungstools DeepL[13] die Redewendungen ins Deutsche übersetzt wurden, da der Großteil der verwendeten Reviews auf Deutsch verfasst ist (siehe B.1.2). Der Datensatz von Kupczyk besteht aus Redewendungen, die eindeutig explizitem oder implizitem Erklärungsbedarf zugeordnet werden. Aus der Bachelorarbeit von Kurtz [25] wurden weitere Schlagwörter hinzugenommen, die den allgemeinen Erklärungsbedarf erkennen sollen. Diese Schlagwörter wurden anschließend als möglicher Erklärungsbedarf eingeordnet und auch mithilfe von DeepL ins Deutsche übersetzt. Aus dem Datensatz des Papers von Droste et al. [26] entstammen noch weitere Triggerwörter, die entnommen wurden und den Datensatz für möglichen Erklärungsbedarf erweitert haben. Diese wurden anschließend mithilfe von DeepL ins Englische übersetzt. Der Datensatz mit Wörtern und Redewendungen wurde anschließend nicht reduziert bei Wörtern und Redewendungen, die keine zugehörigen Reviews gefunden haben. Da dieser Datensatz aus anderen Arbeiten hervorgeht und bereits evaluiert wurde, ist davon auszugehen, dass die nicht zutreffenden Triggerwörter und Redewendungen bei anderen Apps Treffer finden könnten. Der Datensatz zum Labeln für explizite und implizite Erklärungsbedarfe zielt hierbei auf eine hohe Precision ab (siehe 6.2). Die Labelung mit möglichem Erklärungsbedarf zielt auf einen hohen Recall ab, um alle

[13] https://www.deepl.com/de/translator

Reviews zu labeln, die auch mit einer niedrigen Wahrscheinlichkeit einen Erklärungsbedarf haben (siehe 6.2). Die Tabelle 5.2 zeigt eine detaillierte Aufteilung der Navigations-Apps und der dazu gelabelten Erklärungsbedarfe an. Die Software gibt hierbei für alle 2366 Reviews eine Labelung von 26 expliziten, 30 impliziten, 484 möglichen und 1825 keinen Erklärungsbedarf an.

	expliziter Erklärungsb.	impliziter Erklärungsb.	möglicher Erklärungsb.	kein Erklärungsb.
nunav navigation	24	28	425	1659
nunav truck	0	0	0	7
nunav logistics	2	1	49	135
nunav bus	0	1	10	25
Gesamt	26	30	484	1825

Tabelle 5.2: Identifizierter Erklärungsbedarf nach Wörter- und Redewendungenfilter der einzelnen Apps

Die aus der automatischen Zuordnung resultierende Tabelle wurde anschließend von zwei Anforderungsengineers mithilfe von MaxQDA[14] analysiert. Dabei wurden auch die Reviews in Betracht gezogen, die nicht von der Software erkannt wurden. Der Bereich, der den Erklärungsbedarf enthält, wurde zusätzlich markiert und mit explizitem oder implizitem Erklärungsbedarf gekennzeichnet. Der bereits automatisch markierte mögliche Erklärungsbedarf wurde durch expliziten, impliziten oder kein Erklärungsbedarf ausgetauscht. Bei Ungleichheiten der Markierung von zwei Anforderungsengineers wurden die Aspekte des impliziten und expliziten Erklärungsbedarfs abgewogen und zur ausführlichen Klarstellung zwei weitere Anforderungsengineers befragt. Die daraus resultierenden Ergebnisse werden für die zukünftige Auswertung als Ground Truth angesehen und mit der von der Software verglichen 5.3. Bei der Summe der durch die Anforderungsengineers gelabelten Erklärungsbedarfe kommt man auf insgesamt 2376 Reviews. Der Unterschied von 10 weiteren Reviews im Vergleich zu den von dem Scraper gefundenen Reviews lässt sich darin begründen, dass einzelne Reviews mehr als einen Erklärungsbedarf aufweisen, die differenziert betrachtet und gezählt wurden. Hierbei wurden jeweils 5 Reviews, die von der Software als expliziter und

[14]https://www.maxqda.com/de/

möglicher Erklärungsbedarf gelabelt wurden, gefunden, die mehr als einen Erklärungsbedarf haben und somit öfter gezählt wurden. Des Weiteren werden die Reviews der Apps für die folgenden Analysen zusammengefasst und nicht mehr einzeln betrachtet, da dies für die Analyse aufgrund der Ähnlichkeit der Erklärungsbedarfe der Reviews nicht relevant ist. Die Übergangsmatrix 5.4 stellt den Zusammenhang zwischen den von der Software markierten Erklärungsbedarfen und den tatsächlichen Erklärungsbedarfen dar. In der Matrix sind die Überschrift der Spalten die von der Software gelabelten Erklärungsbedarfe und die Überschrift der Zeilen die von den Anforderungsengineers gelabelten Erklärungsbedarfe.

expliziter Erklärungsbedarf	impliziter Erklärungsbedarf	kein Erklärungsbedarf
136	22	2218

Tabelle 5.3: Labelung von Anforderungsengineers von 2376 Reviews

	expliziter Erklärungsb.	impliziter Erklärungsb.	möglicher Erklärungsb.	kein Erklärungsb.
expliziter Erklärungsb.	24	1	97	14
impliziter Erklärungsb.	0	4	6	12
möglicher Erklärungsb.	0	0	0	0
kein Erklärungsb.	5	25	386	1802

Tabelle 5.4: Übergangsmatrix Identifizieren von Erklärungsbedarf: Wörter- und Redewendungenfilter zu Anforderungsengineers; von 2376 Reviews

5.3.2 Taxonomiekategorienzuordnung

Nachdem die Reviews von den Anforderungsengineers eindeutig als Erklärungsbedarfe identifiziert wurden und in einen expliziten oder impliziten Erklärungsbedarf gelabelt wurden, werden die Reviews in eine Taxonomie (siehe 2.2) eingeordnet. Zur Einordnung in die Taxonomie und die daraus resultierende Zuordnung an einen Bezugspunkt wurde eine weitere Software entwickelt, die anhand von in den Reviews gesammelten und gruppierten

Wörtern eine Zuordnung zu einer Taxonomiekategorie vornimmt (siehe 4.2). Dafür wurden die Wörter selektiert und in Minuskeln umgewandelt. Zusätzlich wurden alle Sonderzeichen entfernt und durch Leerzeichen ersetzt. Zur Gruppierung der Wörter wurde eine Normierung durchgeführt, die ähnliche Wörter zusammenfasst. Die Normierung wurde hierbei mit dem Python Modul difflib[15] durchgeführt. Das in dem Modul vorhandene SequenceMatcher ermöglicht ein Abgleichen von Wörtern und gibt einen Float zurück, der die Ähnlichkeit der Wörter vergleicht. Der für diese Arbeit verwendete Vergleichswert wurde auf 0,86 festgelegt, da dies der Schwellenwert ist, in dem dreistellige Wörter nicht mit vierstelligen Wörtern zusammengefasst werden. Somit wurden Wörter wie „nich" und „ich" nicht gruppiert, allerdings Wörter wie „über" und „übers". Je länger die Wörter sind, desto höher ist die Toleranz der Wortunterschiede (siehe B.2). Wörter, die von Anforderungsengineers als ungeeignet für eine Zuweisung an eine Taxonomiekategorie gesehen wurden, wurden durch einen Wörterfilter entfernt (siehe B.2.3). Diese Wort-Taxonomie Zuweisung wird für verschiedene Reviewsammlungen einzeln angelegt, da für verschiedene Apps spezifische Begriffe verwendet werden, woraus eine Taxonomiekategorie hergeleitet werden kann. Die Zuteilung kann anschließend gespeichert werden und muss beim wiederholten Starten der Software nicht neu erarbeitet werden.

Business	Operation	Einführung	Navigation	Algorithmus
29	38	9	5	17

Konsequenzen	Unerwartetes Systemverhalten	Bugs/Abstürze
2	22	6

Designentscheidungen	Geheimhaltung	Sicherheit	Metainformation
6	3	0	18

Begrifflichkeit	Systemspezifische Elemente
0	3

Tabelle 5.5: Taxonomiekategorien Einordnung von 158 Reviews mit Erklärungsbedarf

Nach der Zuordnung der Taxonomiekategorien an die einzelnen Erklärungsbedarfe werden die Ergebnisse noch einmal von Anforderungsengineers ergänzt, korrigiert und überprüft. Daraus resultierte eine Aufteilung der Taxonomiekategorien, die der Tabelle 5.5 zu entnehmen ist.

[15] https://docs.python.org/3/library/difflib.html

5.3.3 Bezugspunkt

Zur automatischen Zuweisung des Bezugspunktes für die Antwort des Erklärungsbedarfs aus den einzelnen Reviews werden die Taxonomiekategorien verwendet. Hierfür wird eine Zuordnung einer Taxonomiekategorie zu einem oder mehreren Bezugspunkten des Unternehmens verwendet. Die Daten für die Zuordnung entstammen aus den Interviews mit dem Supportteam von *Graphmasters*. Sollte die Einordnung eines Teams zu einer Taxonomiekategorie nicht eindeutig sein, so wird eine Gliederung gegeben, welche Teams in absteigender Reihenfolge am wahrscheinlichsten eine Antwort auf den vorliegenden Erklärungsbedarf liefern kann. Sollte ein Team mindestens 25% zu einer Taxonomiekategorie zugehörig sein, wird diese mit in die Gliederung aufgenommen (siehe 6.7 und 6.8).

5.3.4 Quelle

Die Quelle, in der die Antwort für den Erklärungsbedarf formuliert ist, wird mithilfe einer API von der *Graphmasters* Support-Webseite versucht zu ermitteln. Dazu vergleicht die Software mithilfe des SequenceMatcher von difflib die Ähnlichkeit der Supportartikel auf der Support-Webseite von *Graphmasters* mit der gestellten Frage und versucht so zu ermitteln, ob die Frage zutreffend zu dem Supportartikel ist oder nicht. Sollte kein zutreffender Supportartikel auf der Webseite gefunden werden, so wird die Antwort zu dem Erklärungsbedarf aus der Antwort der Appreview in dem jeweiligen App-Store entnommen. Sollte auch wiederum dort keine Antwort auf den Erklärungsbedarf gefunden werden, so wird von *Graphmasters* eine neue Antwort auf den Erklärungsbedarf formuliert.

5.3.5 Bedienung der Software zur Zuordnung des Bezugspunktes und der Quelle

Die Software ist in Python programmiert und wird über das Terminal ausgeführt. Nachdem alle Dateien fehlerfrei (siehe A.2.1) gefunden wurden, wird abgefragt, ob bereits eine gespeicherte Wörter-Taxonomiekategorien Zuordnung verwendet werden soll. Wenn eine bereits vorhandene Wörter-Taxonomiekategorien Zuordnung verwendet werden soll, sucht die Software in dem dafür vorgesehenen *saved_taxonomy* Ordner nach bereits vorhandenen Zuordnungen (siehe A.2.2). Nach Auswahl der bereits gespeicherten Zuordnung fängt die Software an, die einzelnen Reviews zu labeln. Sollte keine gespeicherte Wörter-Taxonomiekategorien Zuordnung verwendet werden oder ist keine vorhanden, so wird eine neue Liste aus

Wörtern erstellt, die von einem oder mehreren Anforderungsengineers in Taxonomiekategorien eingeordnet werden. Die in eine Taxonomiekategorie gelabelten Reviews werden anschließend zur Überprüfung in einer weiteren Excel-Tabelle *./output/Fertige_Einordnung.xlsx* abgespeichert und können durch Anforderungsengineers überprüft und ergänzt werden. Dieser Schritt kann übersprungen werden. Dies hat zur Folge, dass Reviews, die nicht durch die Wörter-Taxonomiekategorien Zuordnung gelabelt werden konnten, keinen Bezugspunkt für die Antwort des Erklärungsbedarfs zugeordnet bekommen. Nachdem die Taxonomiekategorien überprüft und gegebenenfalls ergänzt wurden, werden den Reviews ein Bezugspunkt und eine Quelle zugeordnet. Die Datei wird anschließend im *output* Ordner unter dem Namen *./output/Fertige_Einordnung.xlsx* abgespeichert.

Die Software kann als zusätzlichen Parameter beim Ausführen im Terminal einen Analysemodus hervorrufen, der die Wort-Taxonomie Zuordnung evaluiert. Zum Starten des Analysemodus wird beim Softwarestart der Parameter „-analyse" beigefügt. Die Nutzung dieses Modus ist nur sinnvoll, wenn bereits eine vollständige Einordnung in eine Taxonomie der Reviews stattgefunden hat. Durch den Analyse-Modus kann anschließend die Wort-Taxonomie Zuordnung verbessert werden, um zukünftige Reviews dieser Apps mit einer höheren Präzision in eine Taxonomiekategorie einzuordnen.

Die Einordnung in einen Bezugspunkt, eine Quelle oder beides kann die Software, anhand einer fertigen Taxonomiekategorien Zuordnung der Reviews separat vornehmen. Hierzu gibt es die zusätzlichen Parameter „-location" für die Zuordnung des Bezugspunktes, „-source" für die Zuordnung der Quelle und „-assign" um Bezugspunkt und Quelle zuzuordnen (siehe A.2.3).

Ein- und Ausgabelisten der Software zur Bestimmung von Bezugspunkt und Quelle

Alle Ein- und Ausgabelisten sind in Excel verfasst.

Nur Bezugspunkte und Quellen Suchen
Input: • Fertige Review-Taxonomiekategorie Zuteilung • Taxonomiekategorie-Team Zuteilung
Output: • Alle Reviews mit Metadaten, Bezugspunkten und Quellen

Nur Bezugspunkte Suchen
Input: • Taxonomiekategorie-Team Zuteilung • Fertige Review-Taxonomiekategorie Zuteilung
Output: • Alle Reviews mit Metadaten und Bezugspunkten

Nur Quellen Suchen
Input: • Fertige Review-Taxonomiekategorie Zuteilung
Output: • Alle Reviews mit Metadaten und Quellen

Normaler Durchlauf
Input: • Erklärungsbedarf aus MaxQDA • Alle Reviews mit Metadaten • *(optional)* Wörter-Taxonomiekategorie Zuteilung • *(optional)* Taxonomiekategorie-Team Zuteilung
Output: • Alle Reviews mit Metadaten, Bezugspunkten und Quellen

Analyse-Modus
Input: • Erklärungsbedarf aus MaxQDA • Alle Reviews mit Metadaten • Wörter-Taxonomiekategorie Zuteilung • Fertige Review-Taxonomiekategorie Zuteilung
Output: • Evaluation von Wörter-Taxonomiekategorie Zuteilung

5.4 Evaluation durch Interviews und Umfragen

In der Studie wurden zur Evaluation der Daten Interviews und Umfragen mit dem Unternehmen *Graphmasters* durchgeführt. Die Probanden sollen in der Studie Appreviews einer Taxonomiekategorie (2.2) zuordnen und das Team im Unternehmen nennen (siehe 5.3.3), das über das Wissen verfügt, eine Antwort zu dem Erklärungsbedarf zu formulieren.

Hierbei wurden vier Probanden aus dem Support-Team von *Graphmasters* akquiriert. Das Support-Team von *Graphmasters* ist für die Beantwortung der Reviews zuständig und stellt somit eine Expertengruppe dar. Nachdem die Interviews in Präsenz durchgeführt wurden, in der die Appreviews mit Erklärungsbedarf einer Taxonomiekategorie und einem Bezugspunkt zugeordnet wurden, haben die Probanden an einer anschließenden Online-Umfrage teilgenommen. In der Online-Umfrage wurden weitere Appreviews mit Erklärungsbedarf in dem gleichen Schema der Fragen und Antwortmöglichkeiten wie das Präsenz Interview abfragt. Die Interviews dienen hierbei zur besseren Einarbeitung in das Thema und um anfängliche Unverständlichkeiten aufzuklären. Zur Sicherstellung der gleichen Durchführung der Interviews bei allen Probanden wurden Interview-Guidelines erstellt (siehe C.1). Die Onlineumfrage gibt für Anmerkungen die Möglichkeit, Kommentare zu der Einordnung zu vermerken, wenn es gewünscht ist (siehe C.1.1).

Bei dem Interview wurden insgesamt 75 Reviews abgedeckt. Die Reviews wurden in drei Sätze mit jeweils 25 Reviews unterteilt. Zur Durchführung wurden die vier Probanden in zwei Gruppen eingeteilt. Der erste Satz aus 25 Reviews wurde sowohl von Gruppe eins als auch von Gruppe zwei beantwortet. Der zweite Satz aus 25 Reviews wurde nur von Gruppe eins beantwortet und der dritte Satz folglich nur von Gruppe zwei. So konnte das Interview in einem Rahmen von 30 Minuten bleiben und trotzdem insgesamt 75 Reviews abdecken.

Die restlichen 83 Reviews von den 158 [5.3] Reviews mit Erklärungsbedarf wurden durch die Online-Umfrage evaluiert.

Kapitel 6

Ergebnisse

6.1 Auswertung der Filtermethode

6.1.1 Erklärungsbedarf erkennen

Der Wörter- und Redewendungenfilter hat 548 Reviews mit einer Art von Erklärungsbedarf gelabelt [5.4]. Dieser Filter sollte möglichst viel Erklärungsbedarf erkennen. Bei der Summierung aller Arten von Labelungen, des expliziten, impliziten und des möglichen Erklärungsbedarfs, wird ein *Recall* von 0,8291 erreicht [6.2]. Die *Precision* der summierten Labelungen ergibt 0,2391 und ist somit nicht akkurat. Aufgrund der schlechten *Precision* fällt der *F1-Score* mit 0,3533 gering aus. Die schlechte *Precision* wurde hierbei hingenommen, um eine möglichst hohen *Recall* zu erreichen. Die Vorfilterung von 2376 Reviews auf 548 wurde mit einer *Accuracy* von 0,3149 erreicht. Dieser geringe Wert ist auf die Precision von 0,2106 des möglichen Erklärungsbedarfs zurückzuführen. Die *Precision* der Erkennung von explizitem Erklärungsbedarf erreicht einen Wert von 0,8276. Die *Precision* lässt sich auf den spezifischen expliziten Redewendungenfilter zurückführen, der besonders fein auf expliziten Erklärungsbedarf eingestellt ist. Durch die Verwendung des präzisen Filters wird allerdings nur ein *Recall* von 0,1765 erreicht. Die Erkennung des impliziten Erklärungsbedarfs zeigt die niedrigsten Werte auf, weil dieser implizite Erklärungsbedarf nur indirekt eine Frage darstellt. Die *Precision* dafür lag bei 0,1333 und der *Recall* Wert bei 0,1818. Daraus resultiert ein geringer *F1-Score* von 0,1538.

Metrik	expliziter Erklärungsb.	impliziter Erklärungsb.	möglicher Erklärungsb.	kein Erklärungsb.
Precision	0,8276	0,1333	0,2106	0,9858
Recall	0,1765	0,1818	0,6519	0,8124
F1-Score	0,2909	0,1538	0,3184	0,8901

Tabelle 6.1: Von der Software gelabelter Erklärungsbedarf im Bezug auf den Ground Truth der von Anforderungsengineers gelabelten Erklärungsbedarfe.

Precision	Recall	F1-Score	Accuracy
0,2391	0,8291	0,3533	0,2282

Tabelle 6.2: Insgesamt von der Software gelabelter Erklärungsbedarf im Bezug auf den Ground Truth der von Anforderungsengineers gelabelten Erklärungsbedarfe.

6.1.2 Taxonomiekategorie anhand von einem Wörterfilter erkennen

Die Verwendung eines feinen Wörterfilters [B.1], um eine Taxonomiekategorie zu erkennen, hat nach Anpassung des Filters eine *Precision* von 0,6727 und einen *Recall* Wert von 0,2341 ergeben (siehe Tabelle 6.3). Im Vergleich dazu steht ein grober Wörterfilter [B.2], der mehr Wörter beinhaltet, aber die Wörter nicht so präzise zu einer Taxonomiekategorie zuordnungsbar sind wie beim feinen Wörterfilter. Der grobe Wörterfilter erreicht eine *Precision* von 0,5079 und einen *Recall* von 0,4051. Beim Vergleich des *F1-Scores* ist der grobe Wörterfilter mit einem Wert von 0,4507 dem feinen Wörterfilter mit einem Wert von 0,3473 überlegen. Da hierbei die *Precision* für das Endergebnis des Bezugspunktes eine höhere Relevanz hat und durch weniger falsche Ergebnisse die Korrektur erleichtern soll, kann beim *F-Score* ein β-Wert von 0,2 verwendet werden, um die Auswertung anzupassen. Bei einem β-Wert von 0,2 erreicht der feine Filter einen *F-Score* von 0,5126 und der grobe Filter 0,4873. Die Zuordnung von Anforderungsengineers, nur anhand von Wörtern eine Taxonomiekategorie zuzuordnen, hat nur eine geringe Aussagekraft. Ein Wort einzeln in einen Kontext zu setzen und dies einer Taxonomie einzuordnen, hat einen *Precision* Wert von 0,2211 und einen *Recall* Wert von 0,2857. Aufgrund der schlechten Maßwerte ist dies in der Praxis nicht anwendbar. Der grobe und feine Wörterfilter ist mithilfe der Auswertung der bereits eingeordneten Taxonomiekategorien zu den Reviews mit Erklärungsbedarf entstanden. Dieser Filter ist nicht für neue Apps anwendbar, sondern kann erst nach einer Einordnung der

Taxonomiekategorien für weitere Reviews einer bereits evaluierten App verwendet werden.

Metriken	Einstufung Anforderungsengineers	grober Filter	feiner Filter
Precision	0,2211	0,5079	0,6727
Recall	0,2857	0,4051	0,2341
F1-Score	0,249	0,4507	0,3473
Accuracy	0,2857	0,4051	0,2341

Tabelle 6.3: Evaluation der Wörterfilter zur Zuordnung an Taxonomiekategorien

6.2 Ergebnisse der Interviews und Umfrage

Die Interviews und die Umfrage geben einen Einblick in die Umsetzbarkeit in einem praktischen Umfeld. Hierbei wird die Interrater-Reliabilität mithilfe des Kappa-Wertes geprüft, der angibt, in welchem Maß die Probanden bei ihren Angaben übereinstimmen. Da die Reviews mit Erklärungsbedarf von unterschiedlichen Probandenanzahlen geprüft wurden, entstanden für die Review-Abschnitte 1-25, 26-50, 51-75 und 76-158 einzelne Kappa-Werte und bei der Einordnung der Taxonomiekategorie einzelne Validitäten der korrekten Antwort (siehe 6.4, 6.5, 6.6). Bei den Interviews wurden die Probanden in zwei verschiedene Gruppen (1 und 2) aufgeteilt (siehe 5.4). Für die Reviews 1-25 wurde Fleiss' Kappa-Wert berechnet und für die Reviews 26-50 und 51-75 der Cohens Kappa-Wert. Die Reviews 76-158 haben keinen Kappa-Wert zugeordnet bekommen, da sie ausschließlich von einem Probanden (Gruppe 3) ausgewertet wurden und es somit keine Interrater-Reliabilität gibt. Des Weiteren wurden bei der Auswertung in einer weiteren Tabelle die Einordnungen in die Oberkategorien evaluiert (siehe 6.5). Diese wurden nicht gesondert in den Interviews und der Umfrage abgefragt sondern wurden im Nachgang aus den einzelnen Taxonomiekategorien zusammengesetzt.

6.2.1 Taxonomiekategorienzuordnung

Bei der Auswertung der Kappa-Werte für die Einordnung in die Taxonomiekategorien fällt eine geringe bis angemessene Übereinstimmung aus. Hierbei gibt es signifikante Unterschiede in der Übereinstimmung der Gruppen. Bei der Auswertung der Reviews 1-25, wo Gruppe 1 und 2

bei der Zuordnung beteiligt waren, gab es einen Kappa-Wert von 0,202, der auf eine gerade noch angemessene Übereinstimmung hindeutet. Die Gruppe 1 zeigt bei den Reviews 26-50 einen Kappa-Wert von 0,393, was als angemessene Übereinstimmung gilt. Der Kappa-Wert grenzt aber bereits an die 0,4-0,6 Einstufung von Landis und Koch [20] an, die als moderate Übereinstimmung gesehen wird. Im Gegensatz zu den Reviews 1-25 und 26-50 ist die Übereinstimmung der Reviews 51-75 signifikant geringer mit einem Kappa-Wert von 0,004. Im Vergleich zu den Einordnungen in die Oberkategorien erlangen alle Kappa-Werte einen Anstieg. Der signifikanteste Anstieg ist hierbei bei den Reviews 26-50 zu vermerken, von 0,202 auf 0,442. Die Reviews 1-25 und 26-50 erreichen durch die Zusammenfassung der einzelnen Taxonomiekategorien in Oberkategorien eine moderate Übereinstimmung, anstatt einer angemessenen. Die Reviews 51-75 verbleiben in einer schlechten Übereinstimmung. Bei der von den Probanden eingeordneten Taxonomiekategorie im Vergleich zu den tatsächlichen Taxonomiekategorien der Reviews mit Erklärungsbedarf wird eine Validität von $<50\%$ erreicht. Wenn man die Taxonomieoberkategorien betrachtet, wird die Validität erhöht auf Werte zwischen 54% und 72%.

Review IDs	Probanden Anzahl	Gruppe	Validität	Cohens Kappa	Fleiss Kappa
1-25	4	1,2	37%	/	0,202
26-50	2	1	42%	0,393	/
51-75	2	2	38%	0,004	/
76-158	1	3	26,51%	/	/

Tabelle 6.4: Taxonomiezuordnung von Interviews und Umfrage an Unterkategorien

Review IDs	Probanden Anzahl	Gruppe	Validität	Cohens Kappa	Fleiss Kappa
1-25	4	1,2	62%	/	0,442
26-50	2	1	72%	0,518	/
51-75	2	2	54%	0,085	/
76-158	1	3	43.37%	/	/

Tabelle 6.5: Taxonomiezuordnung von Interviews und Umfrage an Oberkategorien

6.2.2 Teamzuordnungen

Die Übereinstimmungen der Teamzuordnung zu den Reviews (6.6) behalten das gleiche Muster bei wie bei den Taxonomiekategoriezuordnungen. Den höchsten Kappa-Wert von 0,558 erreichen wieder die Reviews 26-50, was als eine moderate Übereinstimmung eingestuft wird. Die Reviews 1-25 erreichten einen Kappa-Wert von 0,307, der als angemessen gilt, und die Reviews 51-75 weisen erneut eine schlechte Übereinstimmung mit einem Kappa-Wert von 0,146 auf. Die Reviews 76-158 haben erneut keinen Kappa-Wert, da diese nur von einem Probanden angegeben wurden.

Review IDs	Probanden Anzahl	Gruppe	Cohens Kappa	Fleiss Kappa
1-25	4	1,2	/	0,307
26-50	2	1	0,558	/
51-75	2	2	0,146	/
76-158	1	3	/	/

Tabelle 6.6: Erklärungsbedarfe an Team Zuordnung von Interviews und Umfrage

6.3 Zuordnung des Bezugspunktes

Der Bezugspunkt wurde mithilfe der Auswertung der Interviews und Umfrage mit dem Unternehmen *Graphmasters* (siehe 6.8), das den Erklärungsbedarf einem internen Team im Unternehmen zugeordnet hat, und der Taxonomiekategorie, die von den Anforderungsengineers zugeordnet wurde, ermittelt. Wenn ein Team zu einer Taxonomiekategorie mehr als >25% der Zuordnungen der insgesamt zu der Taxonomiekategorie zugeordneten Teams erhält, wird dieses Team verwendet, um automatisch der Taxonomiekategorie durch die Software zuzuordnen (siehe Tabellen 6.7 und 6.8).

Durch die 25% Regelung, die Einbeziehung eines Teams zur Zuordnung einer Taxonomiekategorie, kann eine durchschnittliche Validität von >=0,75 erreicht werden. Die tatsächliche Validität liegt bei 0,792. In den errechneten Wert wird einbezogen, dass auch wenn ein Team als zweite oder dritte Option zutrifft, diese als richtiger Treffer gewertet wird. Da die Interrater-Reliabilität nur eine schlechte bis moderate Übereinstimmung ergeben hat und somit die Zuteilung nicht eindeutig ist, wurde sich für eine Abstufung der Teamzuteilung entschieden und keine direkte Zuteilung zu nur einem zugehörigen Team. Auffällig häufig wird das Team *Mobile* und *Support*

genannt. *Mobile* wird acht Mal als erstes zugehöriges Team genannt und *Support* sechs Mal. Im Durchschnitt liefert zu 52,5% die erste Option ein richtiges Team, das den Erklärungsbedarf beantworten kann. Mit einer Wahrscheinlichkeit von 34,9% liefert die zweite Option, wenn eine zugeordnet wurde, das richtige Team, und bei dem einzigen Fall, wo eine dritte Option zugeordnet wurde, liefert diese zu 28,3% das richtige Team.

Das „Team" *Meta* erhält 25 der 283 ausgewerteten Einordnungen. *Meta* ist als Option zu verstehen, wenn keine eindeutige Teamzuteilung möglich ist.

Taxonomiekategorie	Team(s)	Zutrefflichkeit
Business	1. Business	28%
	2. Support	28%
	3. Mobile	28%
Operation	1. Mobile	43%
	2. Support	41%
Einführung	1. Support	54%
	2. Mobile	30%
Navigation	1. Support	35%
	2. Mobile	35%
Algorithmen	1. Routing	42%
	2. Support	28%
Konsequenzen	Mobile	88%
Unerwartetes Systemverhalten	1. Mobile	42%
	2. Routing	33%
Bugs/Abstürze	Mobile	57%
Designentscheidungen	1. UI/UX	55%
	2. Mobile	36%
Geheimhaltung	1. Mobile	75%
	2. Meta	25%
Sicherheit	/	/
Metainformationen	1. Mobile	36%
	2. Support	36%
Begrifflichkeiten	/	/
Systemspezifische Elemente	1. Support	75%
	2. Mobile	25%

Tabelle 6.7: Taxonomiekategorie Teamzuordnung

	Operation	Navigation	Einführung	Unerwartetes Systemverhalten	Bugs/Abstürze	Algorithmus	Konsequenzen	Begrifflichkeit	Systemspezifische Elemente	Geheimhaltung	Sicherheit	Designentscheidungen	Business	Metainformationen
Mobile	29	5	4	14	8	2	7	0	1	3	0	4	15	9
	.43	.35	.30	.42	.57	.06	.88	.00	.25	.75	.00	.36	.28	.36
Courier Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
UI/UX	0	1	1	1	0	1	0	0	0	0	0	6	0	0
	.00	.08	.08	.03	.00	.03	.00	.00	.00	.00	.00	.55	.00	.00
Traffic Management	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	.00	.00	.00	.03	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
Routing	5	1	0	11	3	15	0	0	0	0	0	0	2	2
	.07	.08	.00	.33	.21	.42	.00	.00	.00	.00	.00	.00	.04	.08
Business	2	2	1	1	0	2	0	0	0	0	0	0	15	2
	.03	.14	.08	.03	.00	.06	.00	.00	.00	.00	.00	.00	.28	.08
Support	28	5	7	5	2	10	1	0	3	0	0	1	15	9
	.41	.35	.54	.15	.14	.28	.13	.00	.75	.00	.00	.09	.28	.36
Traffic Strategies	0	0	0	0	0	3	0	0	0	0	0	0	0	0
	.00	.00	.00	.00	.00	.08	.00	.00	.00	.00	.00	.00	.00	.00
Meta	4	0	0	0	1	3	0	0	0	1	0	0	6	3
	.06	.00	.00	.00	.07	.08	.00	.00	.00	.25	.00	.00	.11	.12
Geamt	68	14	13	33	14	36	8	0	4	4	0	11	53	25

Tabelle 6.8: Ground truth Taxonomiekategorien verglichen mit der Teamzuordnung der Erklärungsbedarfe durch Interviews und einer Umfrage.

Die obere Zahl pro Team ist die Anzahl an zugeteilten Teams zu der jeweiligen Taxonomiekategorie. Die untere Zahl ist der prozentuale Anteil des Teams, bezogen auf die gesamte Taxonomiekategorie

6.4 Zuordnung der Quelle

Die Quelle wurde über drei verschiedene Bezugspunkte ermittelt. Einerseits von der Supportwebseite von *Graphmasters* mithilfe einer API, von Google Play Store Reviewantworten, formuliert von *Graphmasters*, und falls keine der anderen Bezugspunkte eine Antwort auf die Reviews stellen konnte, wurden diese manuell neu verfasst. Dabei wurde die Präferenz der Bezugpunkte auf die Supportwebseite gelegt, danach folgend auf die Antworten im Google Play Store und zuletzt die manuelle Neuverfassung.

Die API konnte 27 Supportartikel für Erklärungsbedarfe ermitteln. Von den 27 verfassten Artikeln waren 10 Supportartikel nicht zutreffend hinsichtlich des Erklärungsbedarfs. Die API erlangte eine *Precision* von 0,630 und einen *Recall* von 0,109. Die verfassten kombinierten Supportartikel der ausgewerteten Apps belaufen sich auf 69 (Stand 11.08.2024). Bei den 69 separat angezeigten Supportartikeln treten allerdings Dopplungen auf, wie zum Beispiel, dass jede einzelne App eine formulierte Frage zu den bereitgestellten Sprachen beinhaltet: „welche Sprachen unterstützt die NUNAV (Courier/Cargobike/Navigation/Trucks) App?". Diese separat dargestellten Artikel führen alle auf den gleichen Beitrag. Des Weiteren stellt *Graphmasters* zu jeder App und jeder Version der App alle Release-Notes bereit, die als Antwort auf einen Erklärungsbedarf dienen können. Dies kann in Fällen geschehen, wenn Fragen zu Features implementiert, Bugs behoben oder auch neue Erklärungen im System eingeführt wurden. Durch den Google Play Store konnten weitere 126 Antworten auf die Erklärungsbedarfe hinzugefügt werden. Manuell wurde der Datensatz durch 15 weitere durch *Graphmasters* neu verfasste Antworten zu den Erklärungsbedarfen ergänzt. Von diesen 15 Ergänzungen stammen 13 aus dem Apple App Store, die insgesamt nur 18 der 158 Erklärungsbedarfe ausmachen.

Von den 158 Erklärungsbedarfen wurden 116 in die Apps integriert. Als mögliche Gründe für das Nichtintegrieren der Erklärungsbedarfe in die Software wurde angegeben, dass eine Frage zu selten gestellt wurde, es in Deutschland ein rechtlich nicht erlaubtes Feature darstellt, das Problem zu individuell ist oder gar kein Erklärungsbedarf besteht. Auf jeden Erklärungsbedarf wurde von *Graphmasters* eine Antwort formuliert, mit Ausnahme von denen, die keine Kontaktiermöglichkeit geboten haben, um die Belange zu beantworten. Von den 156 gestellten Erklärungsbedarfen konnten tatsächlich 136 gelöst werden. Für die restlichen konnte das Unternehmen keine Antwort liefern. Beispielhafte Gründe für einen Erklärungsbedarf, der nicht durch das Unternehmen gelöst werden konnte, sind Erklärungsbedarfe, in denen kein Bedarf vom Unternehmen besteht, diese zu lösen, eine weitere Nachfrage benötigt wird oder kein Bezug zu einer App von *Graphmasters* herrscht.

Kapitel 7

Verwandte Arbeiten

Dieses Kapitel behandelt die bereits vorausgegangenen Arbeiten zur Erklärbarkeit, die automatische Beantwortung eines Erklärungsbedarfs und die Verwendung einer Taxonomie zur Kategorisierung von Erklärungsbedarfen.

7.1 Erklärungsbedarfe erkennen

In der Arbeit von Kupczyk [24] wird die automatische Detektion von Erklärungsbedarf in Appreviews untersucht. Hierbei werden zwei verschiedene Methoden, eine regelbasierte Filtermethode und ein Deep Learning Modell, ausgewertet, die den Erklärungsbedarf in einer Appreview erkennen sollen. Die verwendete regelbasierte Filtermethode konnte bei unausgeglichen Datensätzen teilweise eine höhere *Präzision* erwirken als das Deep Learning Modell, verzeichnete allerdings im *Recall* schlechtere Werte. Insgesamt schnitt die Deep Learning Methode besser ab, außer bei unausgewogenen Datensätzen, wo diese gleichstark wie die regelbasierte Filtermethode war.

Unterbusch et al. [27] untersuchten mithilfe von verschiedenen Modellen aus dem Bereich des Natural Language Processing die automatische Erkennung von Erklärungsbedarf in Appreviews. Das optimalste gefundene Modell war das BERT-Modell, das auf einen F_β-*score* von 0,93 kam, bei einer Gewichtung von $\beta = 19{,}52$. Das BERT-Modell ist ein Deep Learning Modell, das verglichen wurde mit regelbasierten Modellen. Die regelbasierten Modelle konnten im *Recall* bei keiner ausgewerteten App mithalten, mit Ausnahme einer App, die gleich mit dem BERT-Modell abschnitt. Den signifikantesten Performanzverlust erleiden beide Modelle bei der *Precision*, dabei generierten sie eine hohe Anzahl an f_p.

Das Identifizieren von Erklärungsbedarf ist ein Teilaspekt in der Arbeit von Kurtz [25]. Kurtz entwickelte eine Software, um Reviews aus App-Stores

zu scrapen und diese anschließend zu analysieren. Hierbei verwendete er einen Wörterfilter, um Merkmale in den Reviews zu erkennen. Mit diesem Filter wurden Erklärungsbedarfe in den Reviews erkannt und markiert.

7.2 Erklärungsbedarfe beantworten

Im Rahmen der Masterarbeit von Brandt [28] wurde generative KI zur Generierung von Erklärungen für Software Systeme verwendet. Hierbei wurde aus einer Vorstudie von Droste et al. [8] der Datensatz mit Erklärungsbedarf entnommen und mithilfe von *ChatGPT*[1] beantwortet. Mithilfe einer Onlinestudie und Interviews wurde herausgearbeitet, dass die Nutzer die Erklärungen von ChatGPT als sehr zielführend und zufriedenstellend erachteten. Die Integration in eine Software und die Effektivität dieser waren mit gemischter Meinung von den Teilnehmern der Studie betrachtet worden.

Horstmann et al. [29] haben in ihrer Arbeit die Selbsterklärung von KI-Systemen analysiert. Die formulierten Erklärungen haben sie aus psychologischer Sicht im Hinblick auf die Bedürfnisse der betroffenen Person betrachtet. Des Weiteren wurden rechtliche und ethische Erfordernisse geprüft, die von einem KI-System eingehalten werden sollen. In den Erfordernissen stehen die informationelle Selbstbestimmung und die damit eingehenden Rechte und das Datenschutzrecht im Vordergrund.

Liu et al. [30] evaluieren die Nutzung von *ChatGPT* im Vergleich zu traditionellen Empfehlungsmethoden, die meist aufgabenspezifische Probleme lösen. *ChatGPT* bekommt bei der Suche nach Erklärungen keine Feinabstimmungen, wie sie traditionelle Empfehlungsmethoden erhalten. In den Ergebnissen wird festgestellt, dass *ChatGPT*, trotz seiner Einschränkungen der spezifischen Informationen in Bezug auf die Bewertung von erklärungsbedürftigen Aufgaben, die Ergebnisse der traditionellen Empfehlungsmethoden übertrifft.

Fechner [31] hat in seiner Bachelorarbeit mithilfe einer Studie die Nachfrage nach Erklärungsbedarf in einer Software ermittelt und lieferte auf Anfrage der Studienteilnehmer eine Erklärung dieser. Die Studienteilnehmer sollten während ihrer Bedienung der Software auftretende Fragen stellen, womit Erklärungsbedarf bei einzelnen Komponenten ermittelt werden konnte. Komponenten, die in der Software immer an der gleichen Stelle auftraten und auf Nachfrage erklärt wurden, wurden als hilfreich wahrgenommen.

[1] https://openai.com/chatgpt/

7.3 Erklärungen und Erklärungsbedarf in eine Taxonomie einordnen

Tsakalakis et al. [9] verwenden eine Taxonomie, um Erklärungen einzuordnen. Die Taxonomie lässt sich in neun Dimensionen aufteilen und soll dabei helfen, Regularien und geschäftliche Anforderungen bei der Erklärung zu erfüllen. Zusätzlich soll die Transparenz im Prozess der automatischen Erklärungsformulierung verbessert werden.

Droste et al. [8] haben eine Taxonomie entwickelt, die genutzt werden kann, um Erklärungsbedarf in Kategorien einzuordnen. Des Weiteren wurden mithilfe von Online-Umfragen Erklärungsbedarfe für Softwaresysteme erkannt und in explizite und implizite Erklärungsbedarfe unterteilt. Durch die identifizierten Erklärungsbedarfe wurde ein Datensatz erstellt, der mit den Taxonomiekategorien gelabelt wurde. Es wurden hierbei zwölf Kategorien erarbeitet, die teilweise in Oberkategorien zusammengefasst wurden.

Panichella et al. [7] entwarfen anhand der 17 herausgearbeiteten allgemeinen gemeinsamen Themen in App-Bewertungen von D. Pagano und W. Maalej [32] und einer von 300 Emails entworfenen Taxonomie mit sechs Kategorien eine für die Wartung und Weiterentwicklung von Software relevante Taxonomie aus vier Kategorien. Die vier Kategorien lauten: *Information Giving*,*Information Seeking*,*Feature Request* und *Problem Discovery*.

7.4 Abgrenzung zu verwandten Arbeiten

In der vorliegenden Arbeit wird ein neues Konzept entwickelt, das eine Erklärung für Erklärungsbedarf nicht direkt formuliert, wie in der Arbeit von Brandt [28] in der generative KI zur Generierung von Erklärungsbedarf evaluiert wurde, sondern auf bereits vorhandene Quellen verweist und den Bezugspunkt für die zuständige Person oder das Team angibt. In der Arbeit steht nicht die Beantwortung der Reviews selbst im Mittelpunkt, da diese bereits zum großen Teil vorhanden sind und andernfalls manuell nachgeliefert werden, sondern die Analyse, wie man an diese Antwort kommt.

Die Grundlage für die Datenbasis wird aus Appreviews gewonnen. Die verwendeten Filter zur Erkennung von Erklärungsbedarf in Appreviews werden hierbei aus den Arbeiten von Kupczyk [24] und Kurtz [25] als Grundlage verwendet und in der Arbeit auf deutsche Rezensionen angepasst und erweitert. Für die Sammlung der Reviews wurde eine eigene Software entworfen, die nicht auf Basis einer anderen Arbeit hervorgeht. Der Grund für einen eigenen Scraper liegt an den spezifischen Eigenschaften der bereits vorhandenen Scraper, die nicht alle Reviews aus den App-Stores abgedeckt

haben.

Zur Einordnung der Reviews in eine Taxonomie, um daran den Bezugspunkt der Antwort zu mappen, wurde als Grundlage die Taxonomie von Droste et al. [8] verwendet. Diese Taxonomie wurde mithilfe der Einordnung der Reviews in die Taxonomiekategorien und der Studie um weitere drei Kategorien erweitert, um die Zugehörigkeit der einzelnen Reviews zu verbessern und zu spezifizieren. Des Weiteren wurde die Taxonomie ins Deutsche übersetzt und mit geeigneten Beispielen auf Deutsch ergänzt.

Kapitel 8

Diskussion

In diesem Kapitel werden die Forschungsfragen, die im Unterabschnitt 5.1 formuliert wurden, mithilfe der Interviews und einer Umfrage beantwortet. Anschließend werden die Ergebnisse im Kontext dieser Arbeit interpretiert.

8.1 Beantwortung der Forschungsfragen

RQ1.1 *Mit welcher Genauigkeit können Nutzerreviews mit Erklärungsbedarf einem Bezugspunkt für dessen Behebung von Erklärungsbedarf auf Basis einer Taxonomiekategorie zugeordnet werden?*

Wenn einbezogen wird, dass nicht ein Bezugspunkt, sondern eine Abstufung der Bezugspunkte von einer Wahrscheinlichkeit des Zutreffens eines Bezugspunktes, übergeben wird, kann mit 79,2% das richtige Team angeben werden, das den Erklärungsbedarf beantworten kann (siehe 6.3). Sollte darauf verzichtet werden, eine abgestufte Teamzuteilung zu verwenden, kann die Zuteilung zu einem einzelnen Team mit 52,5% gegeben werden. Die Taxonomiekategorien weisen maßgeblich auf das zugehörige Team hin (siehe 6.7). Auffällig ist die Verteilung auf die Teams *Mobile* und *Support*. Dabei ist das Team *Mobile* acht Mal als erste zugehöriges Team zu den Taxonomiekategorien genannt und *Support* sechs Mal. Durch die Einschränkung der zugehörigen Teams erhöht sich die Genauigkeit der Zuordnung, da die Auswahlmöglichkeiten geringer sind.

RQ1.2 *Mit welcher Genauigkeit können Nutzerreviews mit Erklärungsbedarf einer Quelle für deren Behebung von Erklärungsbedarf auf Basis einer Taxonomiekategorie zugeordnet werden?*

Im Vergleich zum Bezugspunkt des Erklärungsbedarfs ist eine Quelle nicht einer Taxonomiekategorie zuzuordnen, da kein direkter Zusammenhang

besteht. Die Taxonomiekategorie gibt einen Bereich an, wo Erklärungsbedarf besteht, allerdings kann dadurch kein direkter Rückschluss auf die individuelle Frage gezogen werden. Durch die Ermittlung des Bezugspunktes mithilfe einer Taxonomiekategorie wird jedoch der Ursprung für eine Quelle gefunden, um den Erklärungsbedarf zu beantworten. Die Quelle selbst wird durch die Support-API und die bereits vorhandenen Reviewantworten im Google Play Store ermittelt (siehe 6.4. Die Support-API hat hierbei 17 Supportartikel für die Beantwortung der Erklärungsbedarfe geliefert, mit einer *Precision* von 0,630 und einem *Recall* von 0,109. Der geringe *Recall* und die mäßige *Precision* lassen sich auf die geringe Anzahl an verfassten Supportartikeln zurückführen. Die Reviewantworten aus dem Google Play Store lieferten 126 Antworten als Quelle. Des Weiteren wurden für die 15 nicht ermittelten Quellen neue Antworten von *Graphmasters* verfasst.

RQ2.1 *Kann jedem Nutzerreview mit Erklärungsbedarf ein eindeutiger Bezugspunkt zugeordnet werden?*

In den Interviews und der Umfrage wurden 258 von 283 Bezugspunkten eindeutig einem Erklärungsbedarf zugeordnet (siehe 6.3). Wie bei den Taxonomiekategorien, gibt es auch hier Sonderfälle, die mit dem Tag *Meta* gelabelt wurden. *Meta* beschreibt einen nicht eindeutig zuzuordnendes Team oder ein Erklärungsbedarf, der in das Feld von mehr als nur einem Team fällt und übergreifend gelöst werden muss. Als Beispiel für eine teamübergreifende Frage wäre „Ist eine Import/Export Funktion oder Account Erstellung in Planung?". Diese Frage fällt in die Taxonomiekategorie *Business*, ist aber bei Beantwortung und Auswertung nicht alleine vom *Business*-, *Mobile*- oder *Supportteam* beantwortbar, sondern erfordert eine teamübergreifende Abstimmung zur Beantwortung und auch internen Klärung der Frage. In dem Unternehmen von *Graphmasters* hingegen treten in der Praxis keine uneindeutig zuordnungsbaren Teams auf, da für die genannte Einordnung *Meta* immer das Supportteam verantwortlich ist, das sich schlussendlich mit der Frage auseinandersetzt.

RQ2.2 *Kann jedem Nutzerreview mit Erklärungsbedarf eine eindeutige Quelle zugeordnet werden?*

Wie aus dem Unterkapitel 6.4 hervorgeht, wurde von *Graphmasters* jedem Erklärungsbedarf eine Antwort geliefert. Diese Quellen waren vorab aber nicht alle erhältlich. *Graphmasters* formulierte auf 15 der 158 Erklärungsbedarfe eine neue Antwort, da weder auf der Supportwebseite von *Graphmasters* ein passender Artikel verfasst war noch eine Antwort aus dem Google Play Store bestand. Bei den Reviews aus dem Apple App Store

gab es ausschließlich die Möglichkeit, eine Quelle aus der Supportwebseite zu beziehen. Dies schränkte die Optionen der Quellenfindung für den Apple App Store erheblich ein. 13 der 15 neu verfassten Antworten durch *Graphmasters* wurden für die Apple App Store Reviews verfasst, die insgesamt 18 der 158 Reviews mit Erklärungsbedarf ausmachen.

RQ3.1 *Unter welchen Voraussetzungen, die ein Unternehmen bietet, kann ein Bezugspunkt für ein Nutzerreview, bestehend aus einer Person oder einem Team, angegeben werden?*

Für die Zuordnung eines Bezugspunktes zu einem Erklärungsbedarf muss das Unternehmen die interne Struktur angeben, welche Teams es für bestimmte Bereiche im Unternehmen gibt (siehe 2.4). Des Weiteren muss ein Datensatz bestehen, bei dem Zuordnungen von Teams an Erklärungsbedarfe bereits bestehen, um daraus Schlussfolgerungen für weitere Erklärungsbedarfe zu ziehen. Der Datensatz kann mithilfe von Interviews, Umfragen oder anderen Evaluationen entstehen. Erst wenn diese Grundlagen geschaffen wurden, kann ein Bezugspunkt automatisch an einen Erklärungsbedarf adressiert werden (siehe 4.1).

RQ3.2 *Unter welchen Voraussetzungen, die ein Unternehmen bietet, kann eine Quelle für ein Nutzerreview, bestehend aus Information für die Erfüllung der Erklärbarkeitsanforderung, angegeben werden?*

Für die Beantwortung eines Erklärungsbedarfs in einem Nutzerreview gibt es die Möglichkeit, allgemeine Artikel auf einer Supportwebseite zu erstellen (siehe 5.3.3). Dies hat den Vorteil, dass bei wiederauftretenden Erklärungsbedarfen bereits eine Antwort zur Klärung formuliert wurde. Des Weiteren kann eine Quelle ermittelt werden, wenn ein Unternehmen, manuell oder auch automatisch, Antworten auf Nutzerreviews direkt in den App-Stores öffentlich beantwortet. Sollte keine Quelle vorhanden sein, kann das Unternehmen diese manuell nachreichen. Für Erklärungsbedarfe, die nicht das Unternehmen selbst betreffen oder bei denen nicht genügend Informationen vom Nutzer geliefert werden, kann auch das Unternehmen keine Voraussetzungen bieten, eine Quelle bereitzustellen. Bei Fragen zu fachfremden Erklärungsbedarfen besteht die Möglichkeit einer Eigenrecherche des Unternehmens, um den Erklärungsbedarf zu beantworten. Dies wird aus Aufwandsgründen im Unternehmen *Graphmasters* nur mit Verweisen beantwortet und nicht mit ausführlicher Antwort.

RQ4 *Wie viel echten Erklärungsbedarf kann eine Firma adressieren?*

Graphmasters konnte 136 von den 158 Erklärungsbedarfen nicht adressieren,

da diese nicht das Unternehmen betrafen oder durch zu geringe Information keine Lösung der Sachlage möglich war (siehe 6.4). Ein Beispiel für zu wenig Information wäre „Manchmal werden komische Linien auf der Karte angezeigt." und ein weiteres Beispiel für unternehmensfremde Fragen wäre „Wie bekomme ich ein Update auv mein Navi im Auto VW Passat.[sic]".

8.2 Interpretation der Ergebnisse

8.2.1 Taxonomieerweiterung

Bei der Zuordnung der Taxonomiekategorien durch die zwei Anforderungsengineers (siehe 5.3.2) wurde bereits festgestellt, dass in die Taxonomie von Droste et al. [8] nicht jeder Erklärungsbedarf zuzuordnen war, wie es bei Droste et al. selbst der Fall war.

Um Elemente spezifischer in die Taxonomie einzuordnen, wurde die Taxonomie um die Kategorien *Business* und *Metainformation*, wie es auch Kupcyzk [24] in seiner Arbeit aufgenommen hat, erweitert. *Metainformationen* wurden auch in dem Datensatz von Droste et al. [26] aufgeführt, konnten allerdings durch Zuordnungen an andere Kategorien eliminiert werden und wurden folglich später im Paper nicht aufgeführt. Die Kategorien *Business* und *Metainformation* wurden hierbei keiner Oberkategorie zugeordnet und stehen somit für sich. *Business* beschreibt alle Erklärungsbedarfe, die sich nicht konkret auf das System beziehen, sondern Fragen zum Anbieter sind. Durch die Kategorie *Metainformationen* wurden folglich alle Erklärungsbedarfe gesammelt, die nicht zuzuordnen waren oder mehrere Kategorien betrafen. Die Kategorie *Metainformationen* umfasste damit schlussendlich weiterhin 18 von 158 Erklärungsbedarfen.

Von diesen gelabelten Erklärungsbedarfen könnten 10 in eine weitere neue Kategorie *Feature Fragen* eingeordnet werden. Die Kategorie *Feature Fragen* ist hierbei eine neue Kategorie, die nicht aus anderen Arbeiten hervorgegangen ist. Die Kategorie *Feature Fragen* deckt Fälle ab, die oftmals mehrere Taxonomiekategorien beeinflussen und somit nicht eindeutig zuordnungsbar wären. Als Beispiel kann hier ein Review nach einer Einführung für eine polnische Sprache genommen werden: „hmm gibts auch Polnische Sprache fur navi und auch andere Sprachen[sic]". Dabei wird die Taxonomiekategorie *Operation* angesprochen für das Verwenden einer bestimmten Funktion, der Sprachänderung. Andererseits betrifft dies auch die *Einführung* in die Software, da eine solche Frage bei Beginn der Nutzung auftreten kann. Schließlich wird die Kategorie *Business* angeschnitten, da die Frage nach zukünftiger Spracherweiterung gestellt wird und ob diese in

Zukunft vorgesehen ist. Die Kategorie *Feature Frage* würde hier eindeutig zutreffen und den Erklärungsbedarf klar kategorisieren können.

8.2.2 Ergebnisse der Studie

Einordnung in eine Taxonomie

Die Validität der Taxonomiekategoriezuordnungen (siehe 6.4), von den Probanden der Studie, lässt Rückschließen, dass Schwierigkeiten in der Einordnung der Taxonomiekategorien bestanden und diese entweder nicht vollumfänglich verstanden wurden oder eine Einordnung zu subjektiv vorgenommen wurde, basierend auf Erfahrungen aus der Praxis, und somit nicht der Kern des Erklärungsbedarfs erkannt wurde. Aus den Interviews ging hervor, dass die Anzahl der Taxonomiekategorien zu groß sei und der Überblick darüber schwer zu verschaffen sei. Eine Lösung wäre die Reduzierung der Taxonomiekategorien auf die Oberkategorien und alleinstehenden Kategorien. Dies erhöht die Validität der Zuordnungen im Schnitt um 23,6% (siehe 6.5). Der Nachteil läge allerdings in der schlechteren Zuordnung des Bezugspunktes an die Erklärungsbedarfe, der auf einer Einordnung der Taxonomiekategorie basiert.

Einordnung in einen Bezugspunkt

Die Einordnung der Bezugspunkte war unterhalb der Probanden nicht eindeutig. Mit Kappa Werten zwischen 0,146 und 0,558 konnte keine eindeutige Zuordnung eines Erklärungsbedarfs an ein Team vorgenommen werden (siehe 6.3). Daher wurden für die Bestimmung des Bezugspunktes Abstufungen mit Wahrscheinlichkeitszuordnung erstellt. Diese unterschiedlichen Einschätzungen können sich auf die subjektive Denkweise der Problembewältigung von Erklärungsbedarfen zurückführen. Oftmals wurde zur Beantwortung des Erklärungsbedarfs das Team des Supports genannt, das den Erklärungsbedarf in der Praxis beantwortet. Andere Probanden wählten für das gleiche Review jedoch ein anderes Team, das den Erklärungsbedarf außerhalb des Supportteams beantworten kann. Die Verwechslung zwischen Praxis und Theorie spielte nach Erkenntnis der Aussagen der Interviews die größte Rolle für die unterschiedlichen Einordnungen. Hierbei war es trotz klarer Vorgabe, das ursprüngliche Team für die Beantwortung des Erklärungsbedarfs zu nennen, oftmals zur Orientierung an der Praxis gekommen, die die Einordnungen verfälscht. In der Praxis ist es folglich schwer, eine genaue Einordnung eines Teams vorzunehmen, das in der Theorie die Antwort auf den Erklärungsbedarf kennt.

Kapitel 9

Validität

9.1 Threads of Validity

Wohlin et al. [33] haben vier Kategorien beschrieben, die die Validität der genutzten Methoden und Daten der Arbeit beeinträchtigen können. Die Kategorien von Wohlin et al. [33] werden auf diese Arbeit angewendet und geprüft.

9.1.1 Construct Validity

Wenn eine Einordnung einer unbekannten Taxonomie vorgenommen werden soll, kann dies bei einer erhöhten Komplexität der Taxonomie oft zu Schwierigkeiten bei der Anwendung führen. Hierbei kann eine Taxonomie falsch verstanden und anschließend dann auch falsch verwendet werden.

Um dem entgegenzuwirken, wurden für die Taxonomiekategorien ein Paper mit Einlesungszeit und Fragezeit am Anfang des Interviews vergeben. Die Kategorien lagen zur Klarheit dem Probanden während des Interviews vor, um erneut Klärung zu schaffen. Hierzu wurden zu jeder Taxonomiekategorie Beispiele genannt, die die Kategorie in der Anwendung leichter erklären sollen [2.2].

9.1.2 Internal Validity

Bei der Durchführung der Studie kann es zu zeitlichen Umständen kommen, die die Teilnehmer negativ beeinflussen. Dies kann zum Beispiel der Fall sein, wenn die Interviews in der Freizeit geführt werden und somit Zeit, die zur freien Verfügung steht, verloren geht. Dadurch kann der Proband persönlich beeinflusst sein, das Interview möglichst schnell durchzuführen.

Um diesem Threat entgegenzuwirken, wurden alle Teilnehmenden zur Arbeitszeit am Arbeitsplatz interviewt, um die Zeit der Teilnahme an dem Interview für alle gleich zu gestalten.

9.1.3 Conclusion Validity

In der Studie der Arbeit wurden nur vier Teilnehmer interviewt. Durch die geringe Anzahl an Teilnehmern sind die Daten in statistischen Analysen nur gering aussagekräftig und können zu falschen Schlussfolgerungen führen.

Um diesem Threat entgegenzuwirken, wurden als Teilnehmer Experten gewählt, die mindestens drei Jahre in der Firma auf dem Gebiet gearbeitet haben, um die Interviews aussagekräftiger zu machen.

Bei der Durchführung der Studie kann es bei anderer Herleitung zum Thema und Ablauf des Interviews zu unterschiedlichen Ergebnissen führen, wenn zum Beispiel die Taxonomiekategorien nach Austeilung an den Probanden nicht durch den Interviewer auf Nachfragen erklärt werden.

Um dies zu umgehen und einen gleichen Ablauf für jeden Studienteilnehmer zu gewährleisten, wurde eine Interview-Guideline entwickelt, die bei jedem Teilnehmer gleich angewandt wurde (siehe C.1). Die Interview-Guideline gibt einen zeitlichen Verlaufsplan an und beschreibt die Punkte, an denen Erklärungen gegeben werden, und in welchen Momenten die Herleitung zur Lösung ohne Hilfe geschehen soll.

9.1.4 External Validity

Um die externe Validität zu gewährleisten, wurden die Rahmenbedingungen für alle Studienteilnehmer gleichgesetzt. Hierbei wurden Ort, Zeit und die Fähigkeiten des Studienteilnehmers beachtet. Der Studienteilnehmer muss bereits länger als drei Jahre für das Unternehmen gearbeitet haben und Teil des Supportteams oder die Fähigkeiten eines Supportteammitglieds im Unternehmen besitzen.

Durch die geringe Anzahl der Studienteilnehmer kann die Generalisierbarkeit nicht gewährleistet werden. Die gewählte Stichprobe ist möglicherweise nicht repräsentativ für eine größere Zielgruppe. Da diese Arbeit an einem Unternehmen durchgeführt wurde, das auf eine Software spezialisiert ist, sind die Anforderungen auch darauf spezialisiert (siehe 3). Daher können die Ergebnisse der Studie nicht unbedingt auf andere Kontexte angewandt werden.

Kapitel 10

Zusammenfassung und Ausblick

In diesem Kapitel werden die wichtigsten Ergebnisse der Arbeit zusammengefasst und ein Ausblick gegeben, welche neuen Arbeiten aus dieser Arbeit heraus entstehen könnten.

10.1 Zusammenfassung

Die Arbeit beschäftigt sich mit den Informationsquellen von Erklärungsbedarf. Hierzu wurden die Informationsquellen in Bezugspunkt und Quelle aufgeteilt und am Beispiel einer Navigations-App vom Unternehmen *Graphmasters* evaluiert. Mit der Bestimmung des Bezugspunktes und der Quelle soll der Prozess der Behebung von Erklärungsbedarf vereinfacht werden und eine klare Zuteilung der Verantwortung zur Beantwortung des Erklärungsbedarfs entstehen.

Für eine automatische Zuordnung des Bezugspunktes und der Quelle wurden zwei Softwares auf Grundlage eines Konzeptes entwickelt. Eine Software scraped Reviews einer beliebigen App aus dem Google Play Store und Apple App Store. Die zweite Software beschäftigt sich mit der Zuteilung des Bezugspunktes und der Quelle. Dazu teilt die Software die Erklärungsbedarfe von Reviews in eine Taxonomie ein und bestimmt mithilfe dieser den Bezugspunkt des Erklärungsbedarfs. Mit einer API von der Support Webseite von *Graphmasters*, Antworten aus dem Google Play Store und manuell neu verfassten Antworten wird eine Quelle für den Erklärungsbedarf ermittelt.

Die Evaluation durch Interviews und eine Umfrage mit *Graphmasters* ergibt, dass eine Zuordnung zu einem Bezugspunkt nicht eindeutig erfolgen kann, da selbst die Probanden des Supportteams von *Graphmasters* eine geringe Übereinstimmung in der Teamzuordnung vorweisen. Daher wird

eine Abstufung von einem wahrscheinlichen Bezugspunkt angegeben, die in aufsteigender Reihenfolge die wahrscheinliche Zuordnung des Teams angibt.

Die Verwendung einer Einordnung in eine Taxonomiekategorie zur Bestimmung des Bezugspunktes, erweist sich in der Praxis als schwer umsetzbar, da diese als zu detailliert und spezifisch angesehen wird und somit der Überblick schnell verloren geht.

Die Zuordnung der Quelle kann durch weitere Supportartikel auf der Support-Webseite exakter und in einer größeren Abdeckung geschehen. Durch Supportartikel können zukünftig Verweise darauf geliefert werden, damit nicht manuell neue Erklärungen geschrieben werden müssen.

Abschließend hat die Arbeit ein Konzept aufzuweisen, wie ein Bezugspunkt und eine Quelle ermittelt werden können.

10.2 Ausblick

Für zukünftige Arbeiten können auf dieser Arbeit aufbauend folgende Probleme angegangen werden:

Die Zuordnung eines Teams ist nur die grobe Einordnung eines Bezugspunktes zur Beantwortung eines Erklärungsbedarfs. Es könnten folglich, anhand von geschriebenem Code oder eingeteilten Aufgabenbereichen, einzelne Personen angegeben werden, die Experten in der Beantwortung eines Erklärungsbedarfs sind.

Mithilfe von größeren Datensätzen könnte eine künstliche Intelligenz trainiert werden, die automatisch einen Bezugspunkt und eine Quelle für einen Erklärungsbedarf liefert.

Die Taxonomie zur Einordnung von Erklärungsbedarf könnte weiter evaluiert werden, um zukünftig die Zuordnung in die Kategorie *Metainformationen* weiter zu verkleinern oder sogar ganz wieder zu entfernen. In diesem Kontext kann auch die Verwendung einer anderen Taxonomie evaluiert werden.

Die Teamzuteilungen gingen in dieser Arbeit vom Supportteam von *Graphmasters* aus, die den Erklärungsbedarf einem internen Team zugeordnet haben. Die anschließende Überprüfung des Teams, das die Antwort für einen Erklärungsbedarf zugeteilt bekommen hat, ob diese wirklich den Erklärungsbedarf lösen können, ist in dieser Arbeit nicht evaluiert worden und kann Bestandteil von zukünftigen Arbeiten sein.

Eine Ausweitung der Datenbasis auf eine größere Firma mit vielfältigerer interner Struktur stellt eine weitere Möglichkeit für zukünftige Arbeiten dar.

Anhang A

Software Interaktion

A.1 Screenshots Review Scraper

A.1.1 Review Scraper - ohne Konfigurationsdatei

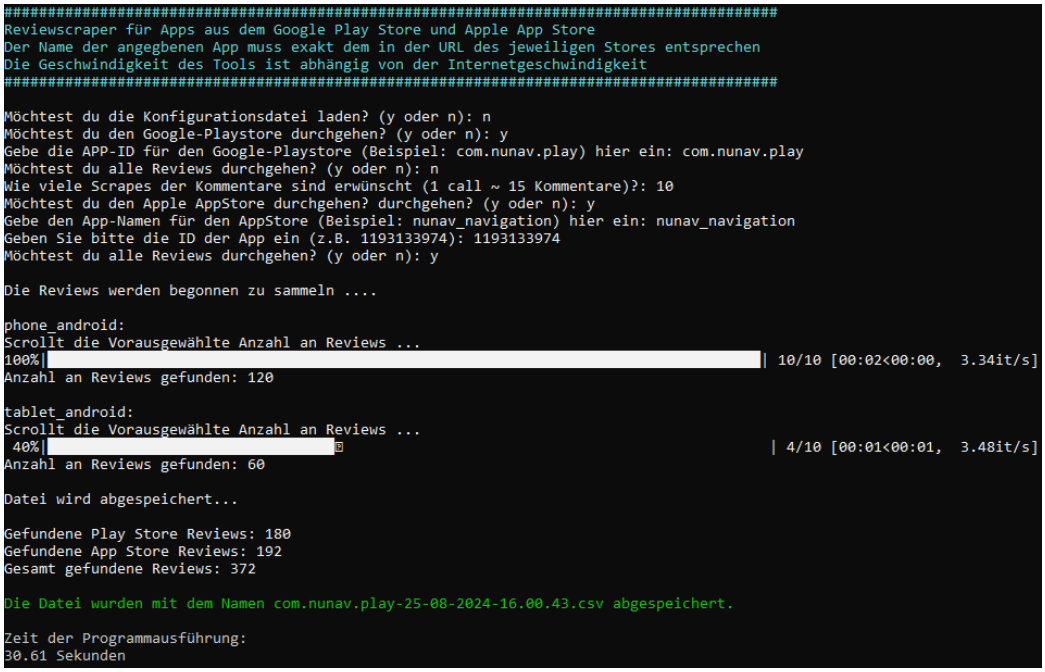

Abbildung A.1: Review Scraper - ohne Konfigurationsdatei

A.1.2 Review Scraper - mit Konfigurationsdatei

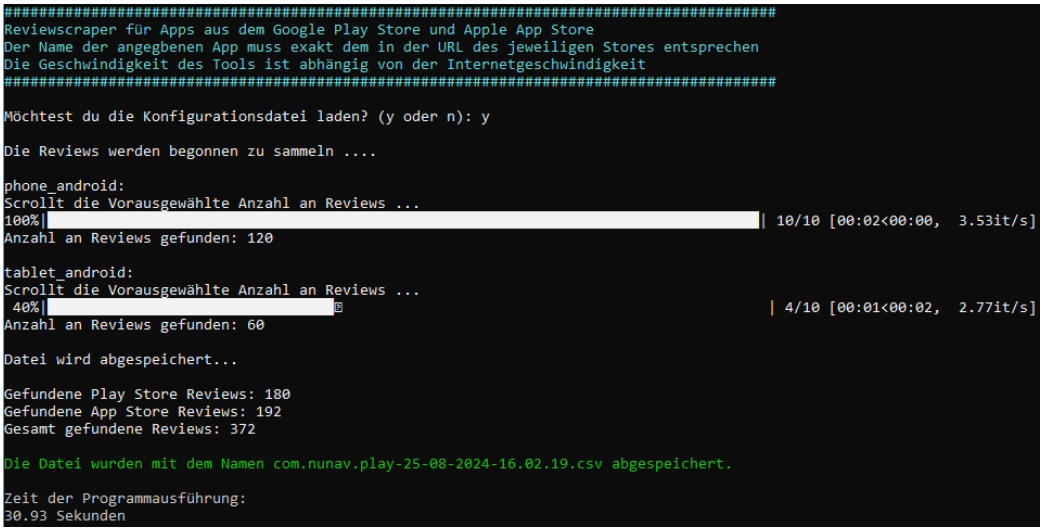

Abbildung A.2: Review Scraper - mit Konfigurationsdatei

A.1.3 Review Scraper - alle Reviews scrapen

```
##############################################################################################
Reviewscraper für Apps aus dem Google Play Store und Apple App Store
Der Name der angegbenen App muss exakt dem in der URL des jeweiligen Stores entsprechen
Die Geschwindigkeit des Tools ist abhängig von der Internetgeschwindigkeit
##############################################################################################

Möchtest du die Konfigurationsdatei laden? (y oder n): n
Möchtest du den Google-Playstore durchgehen? (y oder n): y
Gebe die APP-ID für den Google-Playstore (Beispiel: com.nunav.play) hier ein: com.nunav.play
Möchtest du alle Reviews durchgehen? (y oder n): y
Möchtest du den Apple AppStore durchgehen? durchgehen? (y oder n): y
Gebe den App-Namen für den AppStore (Beispiel: nunav_navigation) hier ein: nunav_navigation
Geben Sie bitte die ID der App ein (z.B. 1193133974): 1193133974
Möchtest du alle Reviews durchgehen? (y oder n): y

Die Reviews werden begonnen zu sammeln ....

phone_android:
Öffnet alle Reviews ...
Die Zeit des Prozesses ist abhängig von der Anzahl der Reviews ...
Anzahl an Reviews gefunden: 1919

tablet_android:
Öffnet alle Reviews ...
Die Zeit des Prozesses ist abhängig von der Anzahl der Reviews ...
Anzahl an Reviews gefunden: 80

Datei wird abgespeichert...

Gefundene Play Store Reviews: 1999
Gefundene App Store Reviews: 192
Gesamt gefundene Reviews: 2191

Die Datei wurden mit dem Namen com.nunav.play-25-08-2024-16.10.44.csv abgespeichert.

Zeit der Programmausführung:
428.6 Sekunden
```

Abbildung A.3: Review Scraper - alle Reviews scrapen

A.2 Screenshots Taxonomie Zuordner

A.2.1 Taxnomie Zuordner - Eingabefehler

```
Öffnet die Excel...
Willst du eine gespeicherte Wörter-Taxonomiekategorien Zuordnung verwenden? (j oder n): ja
Die Eingabe ist ungültig!
Willst du eine gespeicherte Wörter-Taxonomiekategorien Zuordnung verwenden? (j oder n): j
Es werden gespeicherte Wörter-Taxonomiekategorien Zuordnungen gesucht...
Bitte wähle aus den vorhandenen Dateien (1 - 3) eine zur Analyse der Reviews aus:
1: taxonomy_anforderungsengineers.xlsx
2: taxonomy_evaluated_fein.xlsx
3: taxonomy_evaluated_grob.xlsx
4
Die Eingabe ist eine ungültige Zahl. Möchtest du eine Datei wählen? (j oder n): j
Bitte wähle aus den vorhandenen Dateien (1 - 3) eine zur Analyse der Reviews aus:
1: taxonomy_anforderungsengineers.xlsx
2: taxonomy_evaluated_fein.xlsx
3: taxonomy_evaluated_grob.xlsx
eins
Die Eingabe ist keine Zahl. Möchtest du eine Datei wählen? (j oder n): nein
Willst du eine gespeicherte Wörter-Taxonomiekategorien Zuordnung verwenden? (j oder n): j
Es werden gespeicherte Wörter-Taxonomiekategorien Zuordnungen gesucht...
Bitte wähle aus den vorhandenen Dateien (1 - 3) eine zur Analyse der Reviews aus:
1: taxonomy_anforderungsengineers.xlsx
2: taxonomy_evaluated_fein.xlsx
3: taxonomy_evaluated_grob.xlsx
1
Reviews mit Erklärungsbedarf wird mithilfe einer Wörtersuche eine Taxonomiekategorie zugeordnet
Erstellen einer neuen Liste mit allen Metadaten...
```

Abbildung A.4: Review Scraper - alle Reviews scrapen

A.2.2 Taxnomie Zuordner - Normaler Durchlauf

```
Öffnet die Excel...
Willst du eine gespeicherte Wörter-Taxonomiekategorien Zuordnung verwenden? (j oder n): j
Es werden gespeicherte Wörter-Taxonomiekategorien Zuordnungen gesucht...
Bitte wähle aus den vorhandenen Dateien (1 - 3) eine zur Analyse der Reviews aus:
1: taxonomy_anforderungsengineers.xlsx
2: taxonomy_evaluated_fein.xlsx
3: taxonomy_evaluated_grob.xlsx
1
Reviews mit Erklärungsbedarf wird mithilfe einer Wörtersuche eine Taxonomiekategorie zugeordnet
Erstellen einer neuen Liste mit allen Metadaten...
100%|                                                                                | 2428/2428 [00:11<00:00, 205.54it/s]
100%|                                                                                | 158/158 [00:03<00:00, 48.34it/s]

Eine Datei Namens ,,./output/Fertige_Einordnung.xlsx'' wurde in dem output Verzeichnis gespeichert.
Die Datei enthält die, anhand von Triggerwörtern, automatisch zugeordneten Taxonomiekategorien zu den jeweiligen Reviews

Bitte überprüfe und ordnete die nicht vorhandenen Taxonomiekategorien den einzelnen Reviews zu.
Dieser Schritt kann übersprungen werden, dadruch werden jedoch nicht allen Reviews ein Bezugspunkt für die Antwort des E
rklärungsbedarfs zugeordnet.

Drücke eine beliebe Taste um fortzufahren...

Ordnet den, in einer Taxonomiekateogie Zugeordneten, Reviews einen Bezugspunkt zu...

Ordnet den Reviews eine Quelle zu...
100%|                                                                                | 158/158 [00:40<00:00,  3.91it/s]
Die Datei wurde unter ,,/output/Fertige_Einordnung.xlsx'' gespeichert
```

Abbildung A.5: Review Scraper - alle Reviews scrapen

A.2.3 Taxnomie Zuordner - Parameterbeispiel anhand von -assign

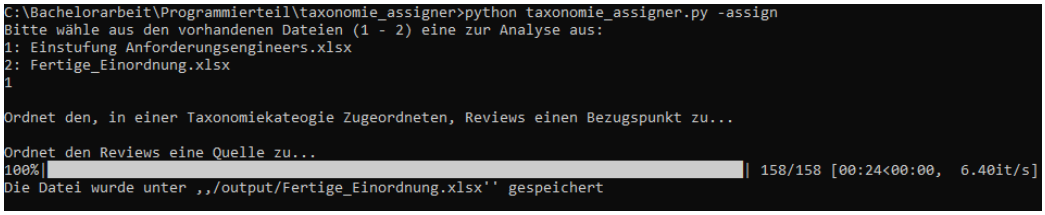

Abbildung A.6: Review Scraper - alle Reviews scrapen

A.3 Konfigurationsdatei Beispiel

```
[PlayStore]

; Gebe an ob der Play Store gescraped werden soll,
   falls nein sind alle anderen Angaben nicht nötig
scrape = y

; Gebe die APP-ID für den Google Play Store (
   Beispiel: com.nunav.play) hier ein:
name = com.nunav.play

; Möchtest du alle Reviews durchgehen? (y oder n):
scrape_all = n

; Wie viele Scrapes der Kommentare sind erwünscht (1
    call ~ 15 Kommentare)?
; Sollte ,,scrape_all'' auf ,,y'' gestellt sein,
   kann die kommende Zeile mit einer ,,0'' gefüllt
   werden
calls = 10

[AppStore]

; Gebe an ob der App Store gescpraed werden soll,
   falls nein sind alle anderen Angaben nicht nötig
scrape = y

; Gebe den App-Namen für den App Store (Beispiel:
   nunav_navigation) hier ein
name = nunav_navigation

; Geben Sie bitte die ID der App ein (z.B.
   1193133974):
id = 1193133974

; Möchtest du alle Reviews durchgehen? (y oder n):
scrape_all= y

; Wie viele Scrapes der Kommentare sind erwünscht (1
    call ~ 20 Kommentare)?
; Sollte ,,scrape_all'' auf ,,y'' gestellt sein,
   kann die kommende Zeile mit einer ,,0'' gefüllt
   werden
calls = 0
```

Anhang B

Beispiele Reviews und Filter

B.1 Redewendungen Filter

In diesem Anhang werden Beispiele des Filters erläutert, die Erklärungsbedarf erkennen sollen.

Die jeweiligen Äquivalente in anderer Sprache (Deutsch/Englisch) sind unter der gleichen Kategorie und der gleichen Nummer zu finden.

Syntax

- (... | ...)

  Die Verwendung von geklammerten Wörtern mit einem „|" getrennt hat zur Folge, dass jedes der aufgezählten Wörter als Kombination gesucht wird.

- „[\w\s]*"

  Der Ausdruck definiert, dass jede Wörter- und Satzzeichenkombination dort eingefügt werden kann.

- „(?!\s*\?)"

  Dies definiert für das Ende des Ausdrucks, dass darauffolgend kein Leerraumzeichen (\t, \n, \r, \f oder \v) und auch kein Fragezeichen stehen darf. Somit wird einbezogen, dass kein expliziter Erklärungsbedarf bei einer direkten Frage vorliegt.

- „\s*\w*\s*"

  Der Ausdruck wird verwendet, um ein Zwischenwort einzufügen. Dieses Zwischenwort in einem Satzgebilde kann vor und nach dem Wort Leerraumzeichen enthalten.

B.1.1 Englische Redewendungen

Expliziter Erklärungsbedarf

Aus dem Datensatz von Kupczyk [24]

1. „tell me what to do"
2. „what is the differentiation"
3. „why do i have to"
4. „please (clarify|explain|break it down|shed some light|tell)"
5. „how is [\w\s]* implemented"

Impliziter Erklärungsbedarf

Aus dem Datensatz von Kupczyk [24]

1. „puzzling to(?!\s*\?)"
2. „(is|'s|was|were)\s*\w*\s*beyond comprehension(?!\s*\?)"
3. „unclear how(?!\s*\?)"
4. „i'm struggling with [\w\s]*"
5. „baffled to(?!\s*\?)"

Möglicher Erklärungsbedarf

Aus dem Datensatz von Droste et al. [26]

1. „example",

   Bei kurzen Wörtern wurden vor und nach dem Wort im Filter Leerzeichen eingefügt, damit diese nicht fälschlicherweise innerhalb eines Wortes erkannt werden.

2. „ who "
3. „ what "

   Bei Wörtern mit verschiedenen Endungen („ed", „ing", etc. werden diese im Filter abgeschnitten, um die Wortendungen alle mit einzubinden.

4. „strugg"
5. „confus"

B.1.2 Deutsche Redewendungen

Expliziter Erklärungsbedarf

1. „sag mir was ich tun soll"
2. „was ist die differenzierung"
3. „warum muss ich"
4. „bitte (klären|erläutern|aufschlüsseln|erklären|erzählen)"
5. „wie wird [\w\s]* umgesetzt?"

Impliziter Erklärungsbedarf

1. „(rätzelhaft für|verwirrend für)(?!\s*\?)"
2. „(ist|war|wurde|wird)\s*\w*\s*unbegreiflich(?!\s*\?)"
3. „unklar wie(?!\s*\?)"

   Zur Verbesserung des Filters wurden beim Übersetzen zum Teil mehr Redewendungen eingebracht, als in der Englischen Version vorhanden sind.
4. „ich (kämpfe mit|habe probleme mit|habe schwierigkeiten mit) [\w\s]*"
5. „verwirrend für|rätzelhaft für|verblüffend für)(?!\s*\?)"

Möglicher Erklärungsbedarf

1. „beispiel"

   Bei kurzen Wörtern wurden vor und nach dem Wort im Filter Leerzeichen eingefügt, damit diese nicht fälschlicherweise innerhalb eines Wortes erkannt werden.
2. „ wer "
3. „ was "

   Bei Wörtern mit verschiedenen Endungen („ieren", „tion", etc.) werden diese im Filter abgeschnitten, um die Wortendungen alle mit einzubinden
4. „inform"
5. „verwirr"

B.2 Wörtergruppierung

Der festgelegte akzeptierte Vergleichswert für diese Arbeit von der difflib.SequenceMatcher().ratio() Funktion wurde auf 0.86 gelegt.

Im Folgenden sind Beispiele aufgelistet, welche Wörter gruppiert wurden und welche aufgrund der Verschiedenheit einzeln aufgefasst wurden.

B.2.1 Akzeptierte Gruppierungen

- „fahr" - „fahre" : 0,889
- „routenoptimierung" - „rutenoptimerungen" : 0,882
- „navigation" - „navgiation" : 0,9

B.2.2 Verworfene Gruppierungen

- „strand" - „strant" : 0,833
- „schnell" - „schnee" : 0,769
- „hügel" - „hügelig" : 0,833

B.2.3 Gefilterte Wörter

Beispiele für gefilterte Wörter:

- Pronomen:
  Ich, Du, Er, Sie, usw.
- Artikel:
  Der, Die, Das, Dieser, usw.
- Modalverben:
  will, soll, muss, darf, usw.
- Partikel:
  ja, mal, sehr, etwas, bloss, usw.
- Begriffe welche zu unspezifisch sind:
  App, Grund, Kunde, Unbekannt, usw.

B.3 Wörter-Taxonomiekategorie Zuordnung

android	Business
carplay	Business
lkw	Business
version	Business
anzeigen	Designentscheidungen
farben	Designentscheidungen
welcher	Interaktion->Einführung
zwischenziele	Interaktion->Operation
eingeben	Interaktion->Operation
meter	Systemverhalten->Unerwartetes Systemverhalten
brücken	Systemverhalten->Algorithmen
daten	Systemverhalten->Algorithmen
geladen	Systemverhalten->Bugs/Abstürze
kroatien	Systemverhalten->Unerwartetes Systemverhalten

Tabelle B.1: Wörter-Taxonomiekategorien Zuordnung fein

brücken	Systemverhalten->Algorithmen
carplay	Business
einstellen	Interaktion->Operation
farben	Designentscheidungen
geladen	Systemverhalten->Bugs/Abstürze
kroatien	Systemverhalten->Unerwartetes Systemverhalten
lkw	Business
meter	Interaktion->Operation
welcher	Interaktion->Einführung
zwischenziele	Interaktion->Einführung
anzeigen	Designentscheidungen
maps	Systemverhalten->Unerwartetes Systemverhalten
stau	Systemverhalten->Algorithmen
version	Business
android	Business
daten	Systemverhalten->Algorithmen
offline	Interaktion->Einführung
eingeben	Interaktion->Operation
google	Systemverhalten->Unerwartetes Systemverhalten
nunav	Systemverhalten->Algorithmen
autobahn	Interaktion->Operation
karte	Interaktion->Operation
route	Interaktion->Operation

Tabelle B.2: Wörter-Taxonomiekategorien Zuordnung grob

Anhang C

Interview und Evaluation

C.1 Interview Guidelines

Interview Guidelines

Single-Interviews mit Graphmasters GmbH
Bachelorarbeit - Nicolas Voß

1. Begrüßung und Vorstellung (1/2 min.)
2. Unterschrift der Datenschutzerklärung (1/2 min.)
3. Kurze Einleitung in das Thema der Bachelorarbeit (1 min.)
4. Erklärung der Taxonomiekategorien (Paper) (3 min.)
5. Verständnisfragen zur Taxonomiekategorien (2 min.)
    - Notieren, um diese bei anderen Teilnehmern gleich zu beantworten
6. Durchführung der Reviews-Einordnung (22 min.)
    - Abwechslung von Taxonomiekategorien
    - Abwechslung von leichten und schweren Einordnungen
    - Zwischenfragen zu den Taxonomiekategorien beantworten
        - Notieren, um diese bei anderen Teilnehmern gleich zu beantworten
    - Keine eigene Meinung zur Einordnung einbringen
    - Grund für die Einordnung bei längerer Wartezeit und nicht eindeutigen Einordnungen erfragen
7. Fragen, wie die Taxonomie verbessert/ergänzt/gekürzt werden kann.(1/2 min.)
8. Danken und nach Verbesserungsvorschlägen fragen (1/2 min.)
9. Interview Beenden (- min.)

C.1.1 Onlineinterview Beispiel

Abbildung C.1: Onlineumfrage Einordnung der Teams

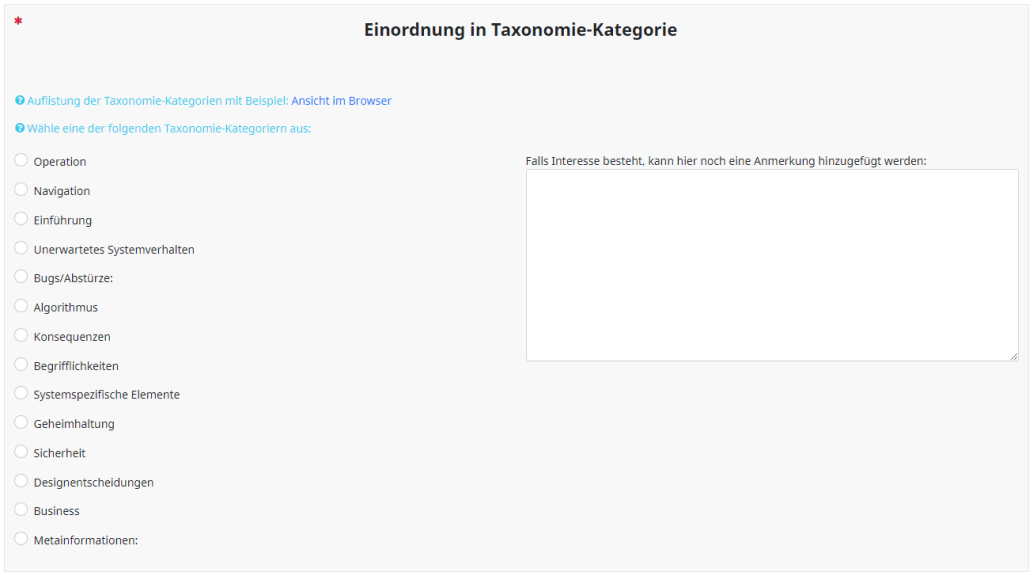

Abbildung C.2: Onlineumfrage Einordnung der Taxonomiekategorie

Literatur

[1] M. Köhl, K. Baum, M. Langer, D. Oster, T. Speith und D. Bohlender, „Explainability as a non-functional requirement," Englisch, *27th International Requirements Engineering Conference*, S. 363–368, 2019. DOI: 10.1109/RE.2019.00046.

[2] L. Chazette und K. Schneider, „Explainability as a non-functional requirement: challenges and recommendations," Englisch, *Requirements Engineering*, Jg. 25, S. 493–514, 2020. DOI: 10.1007/s00766-020-00333-1.

[3] L. Chazette, W. Brunotte und T. Speith, „Explainable software systems: from requirements analysis to system evaluation," Englisch, *Requirements Engineering*, Jg. 27, S. 457–487, 2022. DOI: 10.1007/s00766-022-00393-5.

[4] W. Brunotte, L. Chazette, V. Klös und T. Speith, „Quo Vadis, Explainability? – A Research Roadmap for Explainability Engineering," Englisch, *Requirements Engineering: Foundation for Software Quality 2022*, S. 26–32, 2022. DOI: 10.1007/978-3-030-98464-9_3.

[5] S. Krätzig, „Automatische Erstellung von Erklärbarkeitsanforderungen und Erklärungen," Masterarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Apr. 2024. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2024/MA_Kraetzig-2024_toBeSigned.pdf.

[6] E. Kim, H. Yoon, J. Lee und M. Kim, „Accurate and prompt answering framework based on customer reviews and question-answer pairs," Englisch, *Expert Systems with Applications*, Jg. 203, S. 12, 2022. DOI: 10.1016/j.eswa.2022.117405.

[7] S. Panichella, A. Sorbo, E. Guzman, C. Visaggio, G. Canfora und H. Gall, „How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution," Englisch, *21st International Conference on Software Maintenance and Evolution*, S. 281–290, 2015. DOI: 10.1109/ICSM.2015.7332474.

[8] J. Droste, H. Deters, M. Obaidi und K. Schneider, „Explanations in Everyday Software Systems: Towards a Taxonomy for Explainability Needs," Englisch, *32nd International Requirements Engineering conference*, Jg. 29, S. 197–208, 2024. DOI: 10.48550/arXiv.2404.16644.

[9] N. Tsakalakis, S. Stalla-Bourdillon, T. Huynh und L. Moreau, *A taxonomy of explanations to support Explainability-by-Design*, Englisch, Apr. 2022. DOI: 10.48550/arXiv.2206.04438.

[10] S. R. Department. „Anzahl der verfügbaren Apps in den Top App-Stores im Juli 2024." (2024), Adresse: https://de.statista.com/statistik/daten/studie/208599/umfrage/anzahl-der-apps-in-den-top-app-stores/ (besucht am 25. 08. 2024).

[11] B. Vermeulen, J. Kesselhut, A. Pyak und P. Saviotti, „The Impact of Automation on Employment: Just the Usual Structural Change?" Englisch, *Sustainability*, S. 27, 2018. DOI: 10.3390/su10051661.

[12] G. Marcus, „Deep Learning: A Critical Appraisal," Englisch, *Computing Research Repository*, S. 27, 2018. DOI: 10.48550/arXiv.1801.00631.

[13] L. Chazette, W. Brunotto und T. Speith, „Exploring explainability: a definition, a model, and a knowledge catalogue," Englisch, *29th international requirements engineering conference*, Jg. 29, S. 197–208, 2021. DOI: 10.1109/RE51729.2021.00025.

[14] J. Voges, „Ermittlung von Erklärbarkeitsanforderungen zur Erhöhung der Nutzerakzeptanz eines Stimmungsanalysetools," Masterarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Mai 2024. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2024/MA-Voges-2024.pdf.

[15] A. Adadi und M. Berrada, „Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," Englisch, *IEEE Access*, Jg. 6, S. 52 138–52 160, 2018. DOI: 10.1109/ACCESS.2018.2870052.

[16] H. Deters, „Criteria and Metrics for the Explainability of Software," Masterarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Sep. 2022. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2022/MA-Deters-2022.pdf.

[17] D. Powers, „Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," Englisch, *International Journal of Machine Learning Technology*, S. 37–63, 2011. DOI: 10.48550/arXiv.2010.16061.

[18] M. Hossin und M. Sulaiman, „A Review on Evaluation Metrics for Data Classification Evaluations," Englisch, *International Journal of Data Mining & Knowledge Management Process*, Jg. 5, S. 1–11, 2015. DOI: 10.5121/ijdkp.2015.5201.

[19] J. Cohen, „A Coefficient of Agreement for Nominal Scales," Englisch, *Educational and Psychological Measurement*, S. 37–46, 1960. DOI: 10.1177/001316446002000104.

[20] J. Landis und G. Koch, „A Coefficient of Agreement for Nominal Scales," Englisch, *Biometrics*, S. 159–174, 1977. DOI: 10.2307/2529310.

[21] J. Fleiss, B. Levin und M. Paik, *Statistical Methods for Rates and Proportions, Third Edition*. John Wiley & Sons, Inc, 2003, S. 598–626.

[22] G. Lienert und A. Gustav, *Testaufbau und Testanalyse*. BELTZ, 1998, ISBN: 978-3-621-27845-4.

[23] K. Stapel und K. Schneider, *FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW*, Accessed: 02.08.2024, 2012. arXiv: 1202.5919 [cs.SE]. Adresse: https://arxiv.org/abs/1202.5919.

[24] D. Kupczyk, „Automatisierte Detektion von Erklärungsbedarf in Nutzerfeedback zu Software," Masterarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Okt. 2023. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2024/MA-Voges-2024.pdf.

[25] T. Kurtz, „Entwicklung einer Software zur Extrahierung und Analyse von Reviews aus App Stores," Bachelorarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Aug. 2023. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2023/BA-Kurtz-2023.pdf.

[26] J. Droste, H. Deters, M. Obaidi und K. Schneider, *Artifact for Research Paper „Explanations in Everyday Software Systems: Towards a Taxonomy for Explainability Needs"*, Accessed: 21.08.2024. Adresse: https://zenodo.org/records/10871086.

[27] M. Unterbusch, M. Sadeghi, J. Fischbach, M. Obaidi und A. Vogelsang, „Explanation Needs in App Reviews: Taxonomy and Automated Detection," Englisch, *31st International Requirements Engineering Conference Workshops*, Jg. 5, S. 102–111, 2023. DOI: 10.48550/arXiv.2307.04367.

[28] P. Brandt, „Verwendung und Auswertung von generativer KI zur Generierung von Erklärungen für Software Systeme," Masterarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Apr. 2024. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2024/MA-Brandt-2024.pdf.

[29] A.Horstmann, N. Krämer, C. Geminn u. a., „Kann sich künstliche Intelligenz selbst erklären?" Deutsch, *IMPACT*, 2023. DOI: 10.17185/duepublico/77378.

[30] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou und Y. Zhang, „Is ChatGPT a Good Recommender? A Preliminary Study," Englisch, *32nd ACM International Conference on Information and Knowledge Management*, 2023. DOI: 10.48550/arXiv.2304.10149.

[31] M. Fechner, „Konzept und Implementierung einer Komponente zur Untersuchung des Erklärungsbedarfs von Software," Bachelorarbeit, Gottfired Wilhelm Leibniz Universität Hannover, Hannover, Mai 2023. Adresse: https://www.pi.uni-hannover.de/fileadmin/pi/se/Stud-Arbeiten/2023/BA-Fechner-23.pdf.

[32] D. Pagano und W. Maalej, „User feedback in the appstore: An empirical study," Englisch, *21st International Requirements Engineering Conference*, S. 125–134, 2013. DOI: 10.1109/RE.2013.6636712.

[33] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell und A. Wesslén, *Experimentation in Software Engineering*. Springer, 2012. DOI: 10.1007/978-3-642-29044-20.