

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

Automatisierte Detektion von Erklärungsbedarf in Nutzerfeedback zu Software

Masterarbeit

im Studiengang Informatik

von

Dominik Kupczyk

**Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Dr. Jil Ann-Christin Klünder
Betreuer: Martin Obaidi**

Hannover, 23. Oktober 2023

Erklärung der Selbstständigkeit

Erklärung der Selbstständigkeit. Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig 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 23. Oktober 2023

Dominik Kupczyk

ZUSAMMENFASSUNG

Die Erklärbarkeit von Systemen – ihre Fähigkeit, dem Nutzer das Systemverhalten verständlich zu machen – ist im Rahmen des Software- und Requirements-Engineering eine nicht-funktionale Anforderung, die maßgeblich die Softwarequalität beeinflusst. Um diese Anforderung zu erfüllen, wird angestrebt, den Erklärungsbedarf zu reduzieren. In dieser Arbeit wurde untersucht, wie der Erklärungsbedarf automatisch erkannt werden kann. Dafür wurde ein Datensatz von 2179 App-Reviews erstellt, bei denen Erklärungsbedarf festgestellt wurde. Diese Daten wurden typologisiert und in verschiedene Kategorien eingeteilt. Auf Grundlage dieses Datensatzes wurden sowohl Modelle des Machine Learning als auch des Deep Learning trainiert und bewertet. Zusätzlich wurde eine regelbasierte Filtermethode implementiert, die es ebenfalls ermöglicht, Reviews auf potenziellen Erklärungsbedarf zu untersuchen. Dabei zeigte sich, dass diese Methode auf einem unausgeglichenen Datensatz teilweise höhere Präzision als die Deep Learning Modelle erreicht, jedoch beim Recall Einbußen verzeichnet. Die trainierten Deep Learning Modelle zeigten insgesamt stabilere Ergebnisse, wiesen jedoch bei der Bewertung auf einem unausgewogenen Datensatz Einbußen hinsichtlich der Präzision auf. Beide Modelle wurden in eine GUI integriert, durch die eine Analyse von Reviews aus dem App- und Playstore möglich ist.

ABSTRACT

The explainability of systems – their ability to make the system behavior understandable to the user – is, within the context of software and requirements engineering, a non-functional requirement that significantly influences software quality. To meet this requirement, the aim is to reduce the need for explanations. In this study, we investigated how the need for explanations can be automatically detected. For this purpose, a dataset of 2,179 app reviews was created, in which a need for explanation was identified. These data were typologized and categorized. Based on this dataset, both machine learning and deep learning models were trained and evaluated. Additionally, a rule-based filtering method was implemented, which also allows for the examination of reviews for potential explanation needs. It was found that this method, when applied to an imbalanced dataset, sometimes achieved higher precision than the deep learning models but recorded losses in recall. The trained deep learning models showed overall more consistent results but had losses in precision when evaluated on an imbalanced dataset. Both models were integrated into a GUI, allowing for the analysis of reviews from the App Store and Play Store.

Inhaltsverzeichnis

1. Einleitung.....	1
1.1 Problemstellung	1
1.2 Lösungsansatz	2
1.2. Struktur der Arbeit	3
2 Grundlagen	4
2.1 Nutzerfeedback zu mobilen Apps	4
2.2 Erklärungsbedarf.....	5
2.2.1 Impliziter und expliziter Erklärungsbedarf.....	6
2.2.2 Taxonomie	7
2.6. Datensatzerstellung.....	11
2.3. Natural Language Processing.....	14
2.3.1 Textvorverarbeitung.....	14
2.3.2 Klassifikationstechniken.....	15
2.5. Evaluations Metriken.....	19
3 Verwandte Arbeiten.....	22
3.1 Arbeiten im Zusammenhang zum Erklärungsbedarf und Erklärbarkeit	22
3.2 Klassifikation von Userfeedback	23
3.3 Filterbasierte Methoden.....	24
3.4 Abgrenzung von den Verwandten Arbeiten	25
4. Datensatzerstellung	26
4.1. Filterungskriterien	27
4.2. Datensatzgrundlage	28
4.3 Filterung.....	30
4.4 Datenanalyse und Validierung.....	31
4.4.1 Vorgehen: Identifizierung Erklärungsbedarf.....	31
4.4.2 Diskussionsgrundlage während des Labelns.....	33
4.4.3 Auswertung der Labelung.....	34
5. Datensatzanalyse	35
5.1 Eingesetztes Datenanalyse Tool.....	36
5.2 Auffälligkeiten und Probleme während der Unterteilung in die Kategorien.....	37
5.1. Datensatzvorstellung.....	38
6. Implementierung	43
6.1 Trainingsdatensatz	43
6.2 Wörterbuchmethode	44

6.2.1 Individuelle Phraseneinordnung	45
6.2.2 Phrasenzusammenstellung	46
6.3 Machine- und Deep-Learning Modelle.....	51
6.3.1 Genutzte Frameworks	51
6.3.2 Training der Deep-Learning Modelle.....	52
7. Evaluation der Ergebnisse	53
7.1 Validierung auf Basis eines ausgeglichenen Datensatzes.....	54
7.1.1 Detektion von Reviews mit expliziten Erklärungsbedarf	54
7.1.2 Detektion von Reviews mit impliziten Erklärungsbedarf.....	57
7.1.3 Detektion von Reviews mit allgemeinem Erklärungsbedarf.....	59
7.1.3 Multiklassen-Klassifikation.....	60
7.1.4 Einordnung.....	63
7.2 Validierung auf Basis eines unausgeglichenen Datensatzes.....	64
7.2.1 Erkennung von Reviews mit expliziten Erklärungsbedarf.....	65
7.2.2 Erkennung von Reviews mit impliziten Erklärungsbedarf	66
7.2.3 Erkennung von Reviews mit allgemeinen Erklärungsbedarf.....	67
7.2.4 Wörterbuchmethode.....	68
7.2.5 Multiklassen-Klassifizierung.....	69
7.2.6 Einordnung.....	71
7.3 Fazit der Evaluation	76
8. GUI Einbindung	77
9 Threads of Validity	79
9.1 Interne Bedrohungen der Validität	79
9.1.1 Datensatzfilterung	79
9.1.2 Labelung und Agreement:.....	79
9.1.3 Kategorie Verteilung.....	80
9.2 Externe Bedrohungen der Validität.....	81
9.2.1 Grundlage des Datensatzes:.....	81
9.2.2 Anwendbarkeit auf weitere Forschungsarbeiten.....	81
8.2.3 Einordnung der Evaluationsergebnisse	81
10. Fazit	82
10.1 Diskussion	82
10.2 Zusammenfassung.....	85
10.3 Ausblick.....	86
Literaturverzeichnis:	87

Anhang	93
A1 Übersicht der Apps aus dem Appstore, die als Datengrundlage genutzt wurden dienten.....	94
A2: Übersicht Apps aus dem Play Store, die als Datengrundlage genutzt wurden dienten	95
A3 Zusätzliche Evaluationen.....	96
A4 Eingordnete Beispielergebnisse.....	99
A5 Zuordnung der in der Arbeit zitierten Reviews	103

Abkürzungsverzeichnis

AB.....	Ada Boost
CV.....	Count Vectorizer
EN.....	Explanation Need
Exp	Explizit
Imp	Implizit
KNN.....	K Nearest Neighbor
LR.....	Logistische Regression
NB.....	Naive Bayes
Prec.....	Precision
Rec	Recall
RF.....	Random Forest
TF	TF-IDF Transformer

1. Einleitung

Im Bereich Software- und Requirements Engineering wird Erklärbarkeit, das heißt die Fähigkeit eines Systems die Funktionsweise verständlich zu vermitteln, zunehmend als Qualitätsmerkmal betrachtet [40]. Eine Methode, um Erklärungsbedarf in mobilen Apps zu identifizieren kann die Auswertung von App Bewertungen darstellen [1]. Diese bieten bereits wertvolle Informationen über Kundenzufriedenheit, Fehlerberichte oder Verbesserungsvorschläge [3]. Durch eine detaillierte Untersuchung dieser Bewertungen kann Erklärungsbedarf identifiziert und Software hinsichtlich der Erklärbarkeit optimiert werden [1].

1.1 Problemstellung

Aufgrund der hohen Downloadzahlen populärer Apps [41] und dem geringen Anteil von nur 5%, an Bewertungen, in denen explizit eine Erklärung gefordert wird [1], gestaltet sich die präzise und ausführliche Zusammenstellung relevanter Bewertungen während der Datenanalyse als zeitaufwendig da. Daher wird untersucht, inwieweit eine automatisierte Erkennung von Reviews mit Erklärungsbedarf möglich ist. Da in der Vergangenheit lediglich Datensätze zusammengestellt wurden, in denen Bewertungen mit einem expliziten Erklärungsbedarf betrachtet wurden, stellt sich die Detektion von Bewertungen mit einem impliziten Erklärungsbedarf durch den Einsatz von Machine-Learning Modellen als zusätzliche Herausforderung dabei dar.

1.2 Lösungsansatz

Ziel dieser Arbeit ist es, Modelle zu entwickeln, die zwischen implizitem, explizitem und keinem Erklärungsbedarf unterscheiden können. Hierfür wird ein Datensatz erstellt, wobei die Herausforderungen und Besonderheiten der Labelung ausführlich dokumentiert werden. Dies gewährleistet, dass der Datensatz auch in zukünftigen Forschungsprojekten genutzt werden kann. Um einen tieferen Einblick in den impliziten und expliziten Erklärungsbedarf zu gewinnen und die Modelle effektiv trainieren zu können, wird angestrebt, dass der Datensatz eine umfassende Sammlung von Reviews mit impliziten und expliziten Erklärungsbedarf enthält. Daher wird eine Vorfilterung durchgeführt, um gezielt Reviews mit potentiell Erklärungsbedarf zu identifizieren.

Auf Basis des erstellten Datensatzes entwickeln und evaluieren wir sowohl Machine-Learning- als auch Deep-Learning-Modelle. Zusätzlich wird eine Wörterbuchmethode implementiert, bei der Texte auf das Vorkommen spezifischer Phrasen hin untersucht und ebenfalls eine Klassifikation ermöglicht. Abschließend wird die Leistungsfähigkeit der verschiedenen Modelle bewertet, um zu bestimmen, inwieweit und in welchen Anwendungsfällen die implementierten Ansätze in der Lage sind, Bewertungen mit Erklärungsbedarf zu identifizieren.

Im Rahmen dieses Prozesses wird angestrebt folgende Forschungsfragen zu evaluieren:

- F1:** Wie präzise lässt sich ein Datensatz aus Reviews mit impliziten und expliziten Erklärungsbedarf durch eine gezielte Nutzung von regelbasierten Filtern zusammenstellen?
- F2:** Kann durch eine Filterbasierte Datensatzerstellung gewährleistet werden, dass alle möglichen Bereiche abdeckt, in denen Erklärungsbedarf vorkommen kann?
- F3:** Wie gut können trainierte Modelle Reviews mit Erklärungsbedarf hinsichtlich Precision und Recall erkennen?
- F4:** Wie vergleicht sich eine filterbasierte Wörterbuchmethode gegenüber traditionelle Machine-Learning und Deep-Learning Methoden bei der Identifizierung von Reviews mit Erklärungsbedarf?

1.2. Struktur der Arbeit

Zu Beginn werden in Kapitel 2 die Grundlagen bezüglich Erklärungsbedarf in Appreviews und die Methoden zur automatisierten Erkennung durch Methoden des *Natural Language Processings* vermittelt. Kapitel 3 beschäftigt sich mit verwandten Forschungsarbeiten im Bereich der Klassifikation von Reviews einer bestimmten Art. Daraufhin beschreibt Kapitel 4 die Erstellung des Datensatzes und dessen Auswertung. In Kapitel 5 wird der erstellte Datensatz tiefer in Hinblick auf Auffälligkeiten und Kategorie Unterscheidungen untersucht. Anschließend wird in Kapitel 6 beschrieben, wie Modelle zur automatisierten Erkennung implementiert und trainiert werden. Kapitel 7 beschäftigt sich mit der Auswertung der Modelle, deren Einbindung in eine GUI in Kapitel 8 beschrieben wird., was auch einen praktischen Einsatz der erstellten Modelle ermöglicht. In Kapitel 9 wird die Validität der Ergebnisse diskutiert. Eine Zusammenfassung der Ergebnisse samt Ausblick findet sich in Kapitel 10.

2. Grundlagen

2.1 Nutzerfeedback zu mobilen Apps

Schätzungen zufolge wird im Bereich mobiler Apps bis 2027 ein Marktvolumen von 721,40 Milliarden Euro erreicht [12], was das wirtschaftliche Interesse an diesem Markt hervorhebt. Der Erfolg einer App hängt unter anderem von der Sichtbarkeit oder auch dem Ranking im Store von den Bewertungen ab [3]. User haben dort die Möglichkeit textbasiertes Feedback für andere Nutzer und Anbieter einsehbar zu hinterlassen [10]. Diese App-Rezensionen stellen ein wertvolles Instrument da, um Entwicklern Informationen über Kundenzufriedenheit, Fehlerberichte oder Verbesserungsvorschläge zu kommunizieren [3]. Die Analyse dieser Bewertungen ist schon länger Gegenstand der Forschung [1,3,6]. Dadurch ist es möglich, Nutzergruppen und Stakeholdern zu identifizieren und Probleme schnell zu erkennen [3]. Eine Umfrage unter Release-Ingenieuren bestätigte die Bedeutung des Kundenfeedbacks: Die Mehrheit ist überzeugt, dass Bewertungen ausschlaggebend für den Erfolg von mobilen Apps darstellen. Durch Feedback lässt sich die Lücke zwischen ihren angestrebten Funktionalitätsanforderungen mit den tatsächlichen Bedürfnissen und Erwartungen der Nutzer überbrücken [6]. So kann Nutzerfeedback den Entwicklern signalisieren, falls es Unklarheiten hinsichtlich des Systemverhaltens gibt.

2.2 Erklärungsbedarf

Im Rahmen des Requirements Engineerings wird Explainability („Erklärbarkeit“) eine nicht-funktionale Anforderung da, die mit anderen Software Qualitätsaspekten wie Transparenz [42,43], Verständlichkeit [44,45] und end-user Privacy [46] zusammenhängt. Dadurch wird ausgesagt, wie gut das System dazu in der Lage ist, das Verhalten und die Absichten verständlich zu kommunizieren [44].

Erklärungsbedarf besteht sofern ein Adressat A ein unvollständiges Wissen über einen Aspekt X des Systems S im Kontext C hat und einen Informationskorpus I anfordert, der von einer Entität E bereitgestellt wird und es A ermöglicht, X von S in C zu verstehen. [47].

Erklärbarkeit wird in Bereichen wie der Medizinischen Informatik immer relevanter, wo Systeme Gesundheitsentscheidende Entscheidungen treffen [48], beim Thema Datenschutz, wo nicht immer klar verständlich ist, was mit den Daten passiert [49] und auch Im Rahmen des Requirements Engineerings. Hier kann die Reduzierung von potentiell Erklärungsbedarf vor der Entwicklung besonders relevant sein, da Architekturentscheidungen hiervon beeinflusst werden können [44]. Eine gute Erklärbarkeit führt beim Endnutzer zu erhöhter Kundenzufriedenheit, Transparenz und Verständlichkeit. So kann die Identifizierung von Erklärungsbedarf die Softwareentwicklung und Optimierung dieser: Kundenzufriedenheit, der Marktanalyse und Produktmanagement optimieren und eine Erklärbarkeit, wird als Qualitätsattribut verstanden [51,50] und stellt im Feld des Software Engineerings (SE) und Requirement Engineerings (RE) ein relevantes Feld dar [42].

Beliebte Apps erhalten täglich eine Vielzahl von Bewertungen, die manuell nur schwer zu analysieren sind [9]. Eine genaue Typologisierung und Unterscheidung von Taxonomien kann dabei helfen, komplexe Konzepte besser zu kommunizieren und die Ergebnisse verschiedener Forschungen besser in Bezug zueinander zu setzen [42]. In dieser Arbeit werden zu Forschungszwecken freigegebene Definitionen des Software Engineering

Instituts der Leibniz Universität Hannover als Grundlage für die Einteilung in Typen und Kategorien von Erklärungsbedarf genutzt [36]. Da nach den Untersuchungen von Unterbusch et. Al [1] lediglich 5% der Reviews expliziten Erklärungsbedarf enthalten, es jedoch keine konkreten Untersuchungen in Bezug auf Reviews mit impliziten Erklärungsbedarf gibt, wird für ein konsistentes Vorgehen angenommen, dass der Anteil der Reviews mit impliziten Erklärungsbedarf sich ebenfalls auf 5% beschränkt.

2.2.1 Impliziter und expliziter Erklärungsbedarf

Erklärungsbedarf kann dabei in zwei Typen unterteilt werden: expliziten und impliziten Erklärungsbedarf. **Expliziter Erklärungsbedarf** liegt vor sobald von einem Nutzer deutlich gemacht wird, dass eine Klarstellung oder Erklärung wünscht. Dies kann unter anderem durch Fragewörter wie „why“, „how“ oder „where“ eingeleitet oder durch andere Formulierungen wie „Please tell me how to“ klargemacht werden.

Impliziter Erklärungsbedarf bezieht sich hingegen auf Situationen in denen der Nutzer nicht direkt um eine Erklärung bittet, aber auf andere Art und Weise seine Unklarheiten formuliert, die auf eine gewünschte Klarstellung hindeuten. Dazu gehören neben dem Ausdruck von Unverständnis auch die Beschreibung eines überraschenden oder nicht nachvollziehbaren Verhaltens oder auch die Beschreibung, dass die Software ohne Hilfe nicht bedienbar ist (z.B. „I don't know how I can find my Inventory“) [36].

2.2.2 Taxonomie

Neben der Unterscheidung zwischen impliziten und expliziten Reviews lässt sich Erklärungsbedarf auch Kategorien zuordnen, auf die sich der Erklärungsbedarf bezieht [36]. Wie oben beschrieben kann Erklärungsbedarf in unterschiedlichen Bereichen auftreten, bei denen Unklarheit über Systemaspekte auftreten kann. Dabei wird unterschieden, ob es sich um direkte oder indirekte Systemaspekte handelt.

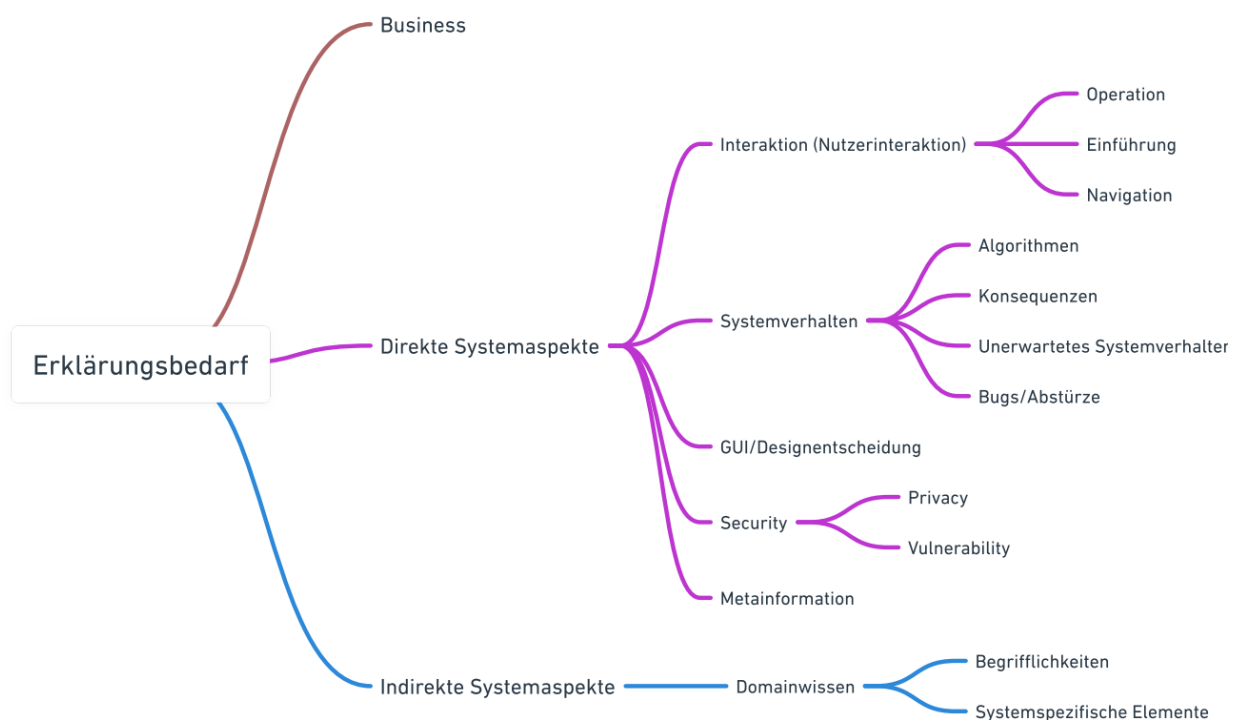


Abbildung 1: Taxonomie bezüglich Erklärungsbedarf

Direkte Systemaspekte

Betrifft Unklarheiten in Bezug auf die genutzte Software.

Interaktion:

Bei Themen, die sich auf die Interaktion mit der Software beziehen, wird dies der Oberkategorie „Interaktion“ zugeordnet. Dieser Bereich lässt sich in folgende Unterbereiche gliedern:

- **Operation:** Hier geht es um die Nutzung bestimmter Funktionalitäten des Systems (z.B. „Wie kann ich nach Artikeln suchen?“).
- **Navigation:** Bezieht sich auf Unklarheiten, die während der Navigation auftreten können (z.B. „Wo finde ich meinen Einkaufswagen?“).
- **Einführung:** Wenn ein Tutorial angefragt wird, welche Schritte erforderlich sind, um etwas zu erreichen.

Systemverhalten:

Betrifft Unklarheiten in Bezug auf das Verhalten des Systems.

- Algorithmus:** Bezieht sich darauf, wie das System zu einem Ergebnis gelangt (z.B. „warum wird mir dieses Hotel als erstes angezeigt?“)
- Konsequenzen:** Betrifft die Unklarheiten welche Auswirkungen ein Verhalten auf das System hat (z.B. „was passiert, wenn ich einen Screenshot von einer Story mache?“)
- Unerwartetes Systemverhalten:** Bezieht sich auf unklare oder nicht nachvollziehbare Aktionen der Software (z.B. „Warum hat sich der Hintergrund verändert?“)
- Bugs/Abstürze:** Falls der Nutzer Erklärungsbedarf bezüglich einer Fehlermeldung oder eines Absturzes ausdrückt (z.B. „Warum ist die App abgestürzt?“).

Security

- Hierrunter fallen Unklarheiten zu Themen wie **Privacy** oder **Vulnerabilitys** in Bezug auf das System (z.B. „Was passiert mit meinen Daten“ oder „Liegt hier eine Sicherheitslücke vor“)

Designentscheidungen:

- Bezieht sich auf Unklarheiten in Bezug auf die Gestaltung der Benutzeroberfläche und einzelner Designelementen. Dabei geht es nicht um die Interaktion mit dieser oder die Funktionalität (z.B. „Warum ist die Suchleiste so klein?“)

Metainformationen:

- Sofern der Erklärungsbedarf keiner der anderen Kategorien zugehörig ist oder mehrere Kategorien betroffen sind, wird die Art Erklärungsbedarf der Kategorie „Metainformationen“ zugeordnet.

Indirekte Systemaspekte

Domainwissen

Unklarheiten die sich auf **Begrifflichkeiten** oder **Systemspezifische Elemente** beziehen (z.B. „Was bedeutet USD“ oder „Wie unterscheidet sich die Premium- von der Gratisversion?“)

Business:

Unklarheiten die sich nicht direkt auf das System beziehen (z.B. „Warum ist das Abonnement teurer geworden?“)

2.6. Datensatzerstellung

Für die Erstellung eines allgemeingültigen Goldstandard-Datensatzes sind die Sicherstellung der Validität der Daten, eine gute Dokumentation sowie auch Unvoreingenommenheit und Objektivität von Bedeutung [34].

Intercoder Agreement

Um subjektive Verzerrungen ausschließen zu können, ist üblich, mehrere Personen für die Labeln und Codierung heranzuziehen. Durch ein „Intercoder Agreement“ kann überprüft werden, ob die Daten die gewünschte Übereinstimmung aufweisen.

Kappa Statistik

Bei Kappa handelt es sich um ein statistisches Maß, welches die Überschneidung der gesetzten Label zwischen Codierern untersucht. Dieser Wert kann zwischen 0 und 1 liegen, wobei 1 würde eine perfekte Übereinstimmung und ein Wert von 0 bedeutet, dass die Ergebnisse nicht besser als eine zufällige Übereinstimmung sind (Tabelle 1) [35].

Kappa	Zuverlässigkeit
>0.8	fast perfekt
>0.6	substanziell
>0.4	moderat
>0.2	mäßig
0-0,2	gering
<0	mangelhaft

Tabelle 1: Interpretation der Kappa Werte [5]

Fleiss Kappa:

Fleiss Kappa ist eine Erweiterung von Cohen's Kappa und ermöglicht die Berechnung eines Kappa-Werts für mehr als zwei Codierer. Eine hohe Übereinstimmung deutet auf eine hohe Inter-Rater-Reliabilität hin [5]. Dieses Maß wird mit dem Fleiß Kappa bestimmt (Formel 1). Dabei wird der Zusammenhang zwischen der beobachteten Übereinstimmung (p_o) und der erwarteten Übereinstimmung, bei einer zufälligen Beurteilung (p_e) untersucht

$$k = \frac{p_o - p_e}{1 - p_e}$$

Formel 1: Fleiß Kappa

Hier beschreibt p_o den durchschnittlichen Prozentsatz der Übereinstimmung zwischen den Ratern (Formel 2). Die erwartende Übereinstimmung p_e gibt dabei an, wie oft man erwarten würde, dass die Rater zufällig übereinstimmen. (Formel 3).

$$p_o = \frac{1}{N * n * (n - 1)} \left(\sum_{i=1}^N \sum_{j=1}^K n_{ij}^2 - N * n \right)$$

Formel 2: der beobachteten Übereinstimmung (p_o). Anzahl der Reviews (N) und Anzahl der Rater (n)

$$p_e = \sum p_j^2$$

Formel 3: Erwartete Übereinstimmung bei Zufälliger Beurteilung. Dabei steht p_j für den Anteil der Bewertungen, die der Kategorie j zugeordnet sind

2.3. Natural Language Processing

Unter dem **Natural Language Processing (NLP)** versteht man einen spezifischen Bereich der Künstlichen Intelligenz, der sich darauf spezialisiert menschliche Sprache zu analysieren. Zu den Hauptanwendungen von NLP zählen unter anderem die Stimmungserkennung, Textzusammenfassung, Textgenerierung und auch die für diese Arbeit relevante Textklassifizierung [14]. Um Systeme mit menschlicher Sprache interagieren zu lassen, ist es notwendig den Text in eine für die Maschine verständliche Form zu übersetzen. Dazu werden Vektorisierungstechniken wie TF-IDF oder Count Vectorizer genutzt, wobei die Auswahl der Vektorisierung Einfluß auf die Ergebnisse haben kann [13].

Der Prozess der Vorbereitung wird als **Preprocessing** bezeichnet. Dieser Schritt hat das Ziel Störungen aus den Daten zu entfernen und Wörter zu konsolidieren, um die Datenanalyse zu vereinfachen. Nach einer sorgfältigen Vorverarbeitung kann der Datensatz zum Training und zu Auswertung eines Klassifikators genutzt werden [15].

2.3.1 Textvorverarbeitung

Damit die Modelle optimal mit den Daten arbeiten können, sollten eine Reihe von Schritten verwendet werden, die den Text vorverarbeiten [5]. Diese Vorverarbeitung umfasst mehrere Schritte:

1. **Tokenisierung:** Der Text wird in kleinere Einheiten, sogenannte Tokens, zerlegt. Dies können Wörter, Sätze oder Absätze sein [15].
2. **Entfernung von Stoppwörtern (Stopword removal):** Hierbei werden Wörter geringer semantischer Bedeutung entfernt. Dazu gehören Gruppen wie Artikel oder Konjunktionen [15]
3. **Stemming und Lemmatization:** In diesem Schritt werden Wörter auf ihre Grundform reduziert. Während beim Stemming die Wortendungen entfernt werden, wird beim Lemmatizing versucht die Grundform des Wortes zu erhalten (oft Zeitintensiver) [16]

4. **Named Entity Recognition (NER)**: Hier werden Entitäten erkannt und klassifiziert (z.B. Orte oder Personen) [16]
5. **Part-of-Speech Tagging**: Jedem Wort wird einer Wortart zugeordnet [16]

2.3.2 Klassifikationstechniken

Es gibt eine Vielzahl von NLP Techniken und Algorithmen die für die Textklassifizierung eingesetzt werden können. Dazu gehören regelbasierte Ansätze, traditionelle Techniken des Machine-Learnings und Deep-Learning-Methoden.

2.3.2.1 Regelbasierte Methoden

Regelbasierte Methoden setzen auf manuell vom Menschen definierte linguistische und heuristische Regeln, die als Entscheidungsgrundlage dienen Beispiele hierfür ist die Schlüsselwörterkennung oder die Wörterbuchmethode, die in dieser Arbeit genutzt wird. Dabei wird der Text auf das Vorkommen bestimmte Muster hin untersucht. [17]. Der Vorteil dieser Methoden ist die Transparenz und Interpretierbarkeit der Filterung. Aufgrund der Vielfalt und Ambiguität natürlicher Sprache, können diese Modelle bei komplexen Aufgaben schlechter Anwendbar sein, als Modelle, die Kontextbezogene Entscheidungen treffen.

2.3.2.2 Ansätze des Maschinellen Lernens

Maschinelle Ansätze haben gegenüber regelbasierten Ansätzen den Vorteil, dass Muster auf abstrakter Ebene automatisch erkannt werden können und diese Modelle selbst Regeln zur Entscheidungsfindung definieren [25]. So können komplexe und große Aufgaben ohne Expertenwissen in der betrachteten Textdomain bewältigt werden [25].

Beim **Naive Bayes** handelt es sich um einen Algorithmus, der einfach und effizient Daten hoher Dimension untersuchen kann. Der Algorithmus beruht auf dem Bayes Theorem und ist „naiv“, da hier die Features nicht in Bezug zueinander gesetzt, sondern unabhängig betrachtet werden.

Auch wenn dies intuitiv nicht zur komplexen menschlichen Kommunikation zugreift, ist es in diesem Fall „häufig ein guter erster Ansatz, wenn die Trainingsdaten begrenzt sind“ [18] und zur Textklassifikation eingesetzt werden.

Bei **Support Vektor Machines** wird versucht, eine Hyperebene zu finden, die die Daten in einem höherdimensionalen Raum abgrenzt. Auch wenn die Trainingsdauer und Nutzung der SVMs rechenintensiv ist, zeigt sich eine gute Effektivität mit Daten höherer Dimension [19].

Bei der **Linearen Regression** handelt es sich um eine statistische Methode, die schnell und gut mit kleineren Datensätzen arbeiten kann, sich jedoch auf lineare Beziehungen beschränkt. Bei großem Vokabular kann dieses Verfahren rechenintensiv sein [20].

Im Fall der **Entscheidungsbäume**, wird der Inputraum in Bereiche eingeteilt, wo für jeden Bereich der Zweig mit der höchsten Wahrscheinlichkeit gewählt wird. Ein Training kann jedoch leicht zum *overfitting* führen (das Training passt sich zu stark an die Trainingsdaten an) [21].

Random Forest nutzt mehrere Entscheidungsbäume, die jeweils auf einen anderen Teil des Datensatzes trainiert sind. Dies kann eine Überanpassung an die Trainingsdaten (overfitting) während des Trainings entgegenwirken, hat jedoch eine höhere Trainingsdauer zur Folge [22].

2.3.2.2 Deep Learning Ansätze:

Deep Learning Ansätze eignen sich zum Teil noch besser um komplexe Muster in Daten zu erfassen, Merkmale automatisch zu extrahieren, arbeiten aber am besten mit hoher Datenmenge [26]. Aufgrund der effektiven Verarbeitung sequenzieller Daten eignen sich Modelle wie Convolutional Neural Networks, Recurrent Neural Networks oder auch Long Short-Term Memory gut zur Textklassifizierung [26]. Da angestrebt wird, die Ergebnisse von Unterbusch et. Al [1] zu optimieren und sich in den letzten Jahren Modelle der Transformer Architektur bewährt haben wird im Bereich der Transformer Architektur hiermit gearbeitet. Diese Modelle haben den Vorteil lokale und globale Kontexte zu verstehen, können jedoch auch groß und rechenintensiv sein [23].

Zu den state-of-the-art Methoden gehört unter anderem **BERT** [24]

BERT

Bei **BERT** („Bidirectional Encoder Representations Transformers) handelt es sich um ein von Forschern von Google entwickeltes Modell für die maschinelle Sprachverarbeitung. Es kann für verschiedene NLP Probleme eingesetzt werden, unter anderem auch zur Textklassifikation. Es handelt sich um ein vortrainiertes Modell, welches in der Forschung eingesetzt wird, indem es auf relevante Bereiche angepasst wird. Es hat den Vorteil, da das vortrainierte Modell bereits stark darin ist Kontextinformationen automatisch erfassen zu können und es sinnvoll für eigene Sachverhalte angepasst werden kann.

SetFit

SetFit, abgekürzt für „Sentence Transformer Fine-Tuning“ ist ein Deep-Deep-Learning Framework. Analysen zeigten mit kleineren Datensätzen ähnliche und teils bessere Ergebnisse als andere Modelle wie BERT [29]. Bei SetFit wird zuerst ein Sentence Transformer Modell mit einem geringen Teil der Trainingsdaten und anschließend der Klassifizierer trainiert [29]. Durch Tupelbildung aus positiven und negativen Paaren, kann das Modell auch mit einer geringen Anzahl von Trainingsdaten gute Anpassungen erzielen [29].

2.5. Evaluations Metriken

Für die Bewertung der Modelle kann es wichtig sein, verschiedene Metriken zu berücksichtigen. Dabei das Verhältnis zwischen den korrekt und inkorrekt vorhergesagten Daten analysiert. Dabei werden die Vorhersagen wie folgt eingeordnet [30].

TP (True Positive): Daten die „positive“ gelabelt wurden und auch als positiv vom Modell klassifiziert wurden

FP (False Positive): Daten die „negativ“ gelabelt wurden und vom Modell als positive klassifiziert wurden

FN (False Negative): Daten die „positive“ gelabelt wurden und vom Modell als negativ klassifiziert wurden

TN (True Negative): Daten die „negativ“ gelabelt wurden und auch als negativ vom Modell klassifiziert wurden

Dies lässt sich in einer Confusion Matrix abbilden (Tabelle 2).

		Modellvorhersage	
		Erklärungsbedarf erkannt	Kein Erklärungsbedarf erkannt
Label	Erklärungsbedarf	130	70
	Kein Erklärungsbedarf	50	500

Tabelle 2: Beispiel einer Confusion Matrix

Durch die **Accuracy** wird das Verhältnis zwischen richtig und falsch vorhergesagten Ergebnissen bestimmt. Diese Metrik kann jedoch, insbesondere bei unausgeglichenen Datensätzen irreführend sein. Vor allem bei einem unausgewogenem Datensatz, kann ein hoher Accuracy Wert suggerieren, dass das Modell generell gut performt, auch wenn schwache Metriken für die unterrepräsentierte Klasse vorliegt [30].

Die **Precision** gibt Verhältnis zwischen den positiv klassifizierten Daten und den positiv gelabelten Daten an. Sie ist besonders relevant, wenn die Kosten für False Positives hoch sind. Eine hohe Precision bedeutet beispielsweise bei einem Spamfilter, dass nur wenige legitime E-Mails fälschlicherweise als Spam erkannt werden [30].

Der **Recall** Value misst hingegen, wie viele der tatsächlich positiven Daten korrekt als solche. Diese Metrik ist vor allem von Bedeutung, falls hohe Kosten für False Negatives anfallen. Ein Beispiel wäre hier die Früherkennung von Krankheiten, bei der das Übersehen einer Erkrankung gravierendere Folgen haben kann, als eine Falschdiagnose [30].

Der **F1-Score** stellt ein harmonisches Mittel zwischen Precision und Recall dar. Im Falle des Früherkennung einer Krankheit könnte ein hoher F1-Score Auskunft darüber geben, dass die Krankheit bei allen betroffenen Patienten gefunden wird ohne dabei gesunde Personen als krank zu diagnostizieren [30].

Der **F1-Mikro Score** berücksichtigt alle Vorhersagen eines Modells unabhängig von einer Klassengewichtung sind. Hierbei fließen alle Berechnungen in die Gesamtkalkulation des F1-Scores mit ein. Bei einer Foto-App, in der die Bilder in Kategorien wie „Menschen“, „Tieren“ und „Gebäuden“ unterteilt, bewertet der F1-Mikro Score die Gesamtpformance unabhängig von dem Vorkommen der Daten einer Klasse [27].

Im Gegensatz dazu berechnet der **F1-Makro Score** die Leistung für jede Klasse einzeln und bestimmt anschließend den Durchschnitt [27]. Bei einem Modell welches gut die dominanten Krankheiten *A*, *B* erkennt, jedoch schlecht die selten vorkommende Klasse *C* als solche erkennt, würde der F1-Makro score geringer als der F1-Mikro Score ausfallen [30].

Durch den **AUC-Score** (Area Under the Curve) die Fähigkeit eines Modells untersuchen. Es ist besonders Aussagekräftig bei binären Klassifikatoren, wobei ein Wert von 1 eine perfekte Klassifikation aussagt und ein Wert von 0,5 aussagt, dass die Erkennungsgenauigkeit nicht besser als zufälliges Raten aussagt [30].

.

3. Verwandte Arbeiten

3.1 Arbeiten im Zusammenhang zum Erklärungsbedarf und Erklärbarkeit

In ihrer Untersuchung zur automatisierten Erkennung von Erklärungsbedarf implementierten und evaluierten Unterbusch et al. [1] verschiedene Methoden aus dem Bereich des Natural Language Processing (NLP). Dazu gehören regelbasierte Ansätze, Maschinelles Lernen und Deep Learning. Dabei erzielten regelbasierte Methoden und das BERT-Modell die besten Ergebnisse bei der Erkennung von explizitem Erklärungsbedarf in einem Cross-Validation-Datensatz mit 5078 Bewertungen. Beide Ansätze erreichten dort einen Macro-F1-Score von 93% [1]. Insgesamt zeigte das BERT-Modell eine bessere Generalisierungsfähigkeit auf ungesehenen Daten. Auffallend war dabei eine starke Verschlechterung des Recall und der Precision bei der Evaluation an einem unausgewogenen Datensatz, bei dem der Anteil der Reviews mit explizitem Erklärungsbedarf lediglich 5% betrug, was den Anteil der Verteilung in der Praxis widerspiegelt [1]. Insgesamt wurden 268 Reviews mit explizitem Erklärungsbedarf zur Verfügung gestellt. Trotz der vielversprechenden Ergebnisse wird für den praktischen Einsatz empfohlen, die einzelnen Ansätze in Bezug auf den Recall-Wert stärker zu optimieren. Um für den praktischen Einsatz qualifiziert zu sein, müssten die Modelle nach Unterbusch et al. [1] auf über 80% verbessert werden. Auch wurde ein Vorkommen von explizitem Erklärungsbedarf in lediglich 5% der Reviews festgestellt. Zur Optimierung der Ergebnisse empfehlen die Autoren unter anderem eine Erweiterung des Trainingsdatensatzes. Die in dieser Studie verwendeten Trainingsdaten bilden die Grundlage für die vorliegende Arbeit. Chazette et. Al [47] fanden heraus, dass der Bedarf an Erklärungen stark vom Nutzer und Kontext abhängt, was die Identifizierung dessen erschwert.

Unter dem Projekt SoftXPlain [36] am Software Engineering Institut der Leibniz Universität Hannover wird untersucht, inwiefern Erklärungsbedarf in Software vorkommt, welche Herausforderungen sich daraus ergeben und wie Erklärungsbedarf reduziert werden kann [38,39,45,43,33]. Im Rahmen des SoftXPlain-Projekts [36] wurden Befragungen durchgeführt, die untersuchten, inwiefern impliziter und expliziter Erklärungsbedarf von Probanden wahrgenommen wird. Dieser Datensatz wurde zusammen mit einer Taxonomie und Definition (siehe Kapitel 2.2), für diese Arbeit zur Verfügung gestellt [36].

Im Rahmen seiner Bachelorarbeit hat Timo Kurz ein Tool entwickelt [32], das darauf ausgelegt ist, Reviews aus dem App- und Playstore herunterzuladen und diese nach verschiedenen Kriterien, wie deren Sentiment oder Noise, zu filtern und übersichtlich anzuzeigen. In diesem Zusammenhang wurde auch ein grober Filter für Reviews mit Erklärungsbedarf eingebaut. Dieser Filter prüft grob nach dem Vorkommen einiger Schlüsselwörter, die auf Erklärungsbedarf hindeuten.

3.2 Klassifikation von Userfeedback

Es gibt eine Vielzahl wissenschaftlicher Arbeiten, die untersuchen, inwiefern Feedback durch verschiedene Techniken klassifiziert werden kann und wie sich unterschiedliche Ansätze kombinieren lassen um die Ergebnisse zu optimieren. In der Studie von Stanik et al. [2] wurde die Wirksamkeit von Deep Learning im Vergleich zu traditionellen Methoden des maschinellen Lernens bei der Klassifizierung von Benutzerfeedback in Twitter-Posts untersucht. Stanik et al. [2] klassifizierten das Feedback in Problembereiche, Anfragen und irrelevante Beiträge. Bei der Klassifizierung von App-Bewertungen erwiesen sich traditionelle Methoden des maschinellen Lernens, einschließlich Naive Bayes und Support Vector Machines, als überlegen gegenüber Deep Learning-Methoden, die unter anderem ein fein abgestimmtes und weiter trainiertes Convolutional Neural Network einsetzen. Die Autoren schlussfolgerten, dass eine

Erweiterung des Trainingsdatensatzes oder eine Kombination von Methoden das Ergebnis des Deep Learnings verbessern könnte. Es ist jedoch zu beachten, dass aktuelle state-of-the-art Deep Learning-Ansätze wie BERT aus unbekanntem Gründen ausgelassen wurden. Untersuchungen von Maalej et al. [4] und Edna et al. [5] verglichen Textmerkmalsextraktionstechniken wie BoW, TF-IDF und CHI2 sowie Machine-Learning-Algorithmen wie SVM, kNN und MNB. Dabei stellte sich heraus, dass die Kombination von TF-IDF mit einer logistischen Regression oder einer Support Vector Machine die besten Ergebnisse erzielte [4]. Zudem konnten die Ergebnisse durch den Einsatz von Bigrammen und Lemmatisierung verbessert werden. Maalej et al. [4] und Edna et al. [5] zeigten, dass probabilistische Techniken eine bessere Performance als String-Matching-Methoden erzielen. Lu et al. [8] stellten ebenfalls fest, dass die Kombination verschiedener Methoden Vorteile bietet. Die Genauigkeit von Deep Learning-Methoden wie BERT bei der Klassifizierung von Texten und der Vorteil von Feintuning und Transferlernen wurden auch von Restrepo et al. [6] bestätigt. Scalabrino et al. [9] betonten zudem die Bedeutung der Vorverarbeitung. Sie untersuchten, inwiefern Nutzerbewertungen effizient kategorisiert und geordnet werden können. Neben typischen Methoden wie dem Entfernen von Stoppwörtern oder Stemming sollten auch Aspekte wie das Arbeiten mit Verneinungen oder Smileys berücksichtigt werden. In Bezug auf das Entfernen von Verneinungen wurden Methoden entwickelt, die effizienter und genauer sind als der Stanford-Parser.

3.3 Filterbasierte Methoden

Zibran et al. [31] haben ein Tool entwickelt, das darauf abzielt, kurze Texte hinsichtlich ihres Sentiments zu analysieren. Dabei werden Begriffe hierarchisch geordnet, wobei jedem Wort im Wörterbuch ein spezifischer Wert zugeordnet wird. In der praktischen Anwendung werden Texte auf das Vorkommen dieser Wörter hin untersucht, sodass der analysierte Text eine endgültige Bewertung in Bezug auf das erkannte Sentiment erhält.

3.4 Abgrenzung von den Verwandten Arbeiten

In der bisherigen wissenschaftlichen Literatur wird deutlich, dass es keinen universellen Ansatz zur Klassifikation von Nutzerbewertungen gibt. Der Kontext und der spezielle Schwerpunkt des Finetunings beeinflussen maßgeblich die Genauigkeit. Die Arbeiten zeigten die Möglichkeit auf, dass die Ergebnisse durch Kombinationen von Methoden, eine Erweiterung des Trainingsdatensatzes und präzise Vorverarbeitung verbessert werden können. In dieser Arbeit wird auf den bisherigen Forschungsstand aufgebaut. Auch wird der implizite Erklärungsbedarf in dieser Arbeit näher betrachtet, welcher in bisherigen Forschungen zum Erklärungsbedarf nicht gezielt berücksichtigt wurde. Zur Erkennung von Reviews mit implizitem und explizitem Erklärungsbedarf werden verschiedene Ansätze untersucht und verglichen, und es wird versucht, den Klassifizierungsprozess weiter zu optimieren. Eine Analyse des Datensatzes von Unterbusch et al. [1] zeigte, dass der zur Verfügung gestellte Datensatz auch Beispiele enthält, in denen Reviews als explizit gelabelt wurden, die jedoch nach der aktuellen Definition des Software Engineering Instituts als implizit eingestuft würden. So wurden nach Unterbusch Definition Reviews mit Phrasen wie „I don't know what this is“ als explizit gelabelt. Da sich bei der Unterscheidung zwischen implizitem und explizitem Erklärungsbedarf in dieser Arbeit an den Guidelines des Software Engineering Instituts orientiert wird, findet hier eine Abgrenzung zur Arbeit von Unterbusch statt. Im Rahmen dieser Arbeit würde solch ein Beispiel, aufgrund einer fehlenden Erklärungsaufforderung als implizit gelten [36].

4. Datensatzerstellung

Angesichts der geringen Anzahl von Daten, mit denen Modelle zum Training der Erkennung von Erklärungsbedarf, wird ein spezifischer Datensatz erstellt. Mit diesem Datensatz können Modelle trainiert und eine Wörterbuchmethode implementiert werden, um Reviews mit Erklärungsbedarf automatisch zu identifizieren. Um der Herausforderung, dass nur 5% der Reviews einen (expliziten) Erklärungsbedarf aufweisen, werden Filtermethoden angewendet, die Reviews mit potenziellem implizitem und explizitem Erklärungsbedarf aus einem größeren Datensatz extrahieren. Dieser vorgefilterte Datensatz wurde dann basierend auf den in Kapitel 2.2.1 beschriebenen Prinzipien von mehreren Bewertern analysiert. Das Ergebnis ist ein Datensatz, der Reviews mit implizitem, explizitem und keinem Erklärungsbedarf differenziert. Dieses Konzept ist in Abbildung 2 dargestellt.

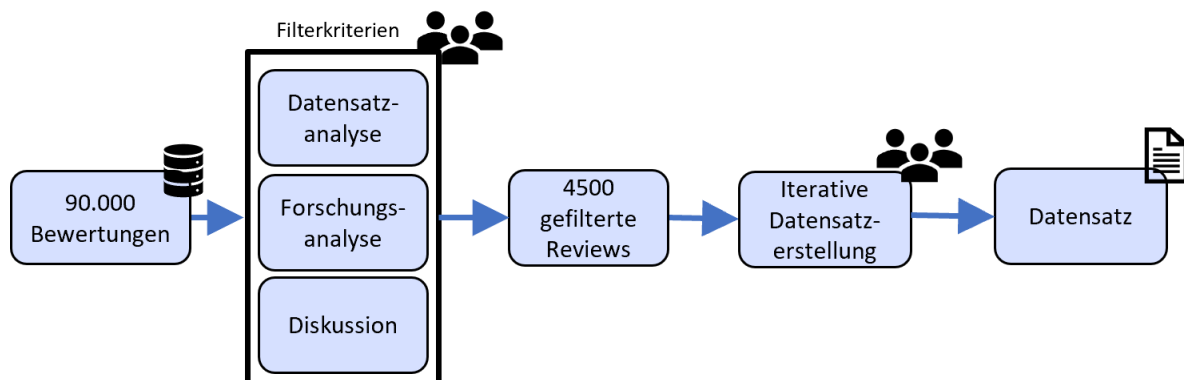


Abbildung 2: Vorgehen bei der Datensatzerstellung

4.1. Filterungskriterien

Wie bereits in Kapitel 3 beschrieben, haben Unterbusch et. Al [1] einen Datensatz zusammengestellt, der 286 Reviews mit expliziten Erklärungsbedarf enthält. Auch wurden im Rahmen von Forschungsarbeiten am Software Engineering Institut der Leibniz Universität Hannover untersucht, inwiefern Probanden expliziten und impliziten Erklärungsbedarf erfuhren. Insgesamt werden hier Aussagen zu wahrgenommen expliziten und impliziten Erklärungsbedarf zusammengetragen. Außerdem legte das Software Engineering Institut Definitionen zu impliziten und expliziten Erklärungsbedarf, als auch Unterscheidungskriterien in Bezug auf die Kategorie fest, auf die sich die Unklarheit bezieht [36].

Gemeinsam mit zwei Masterstudenten der Informatik wurden diese Daten Muster untersucht, die eindeutig auf expliziten oder impliziten Erklärungsbedarf verweisen.

Zur Erstellung eines Filters, der Reviews mit expliziten Erklärungsbedarf extrahiert, wurde mit dem Datensatz von Unterbusch et. Al [1], dem SoftXPlain [36] Datensatz und den Definitionen zu expliziten Erklärungsbedarfs gearbeitet. Die gesammelten Phrasen, die auf einen expliziten Erklärungsbedarf verweisen, wurden analysiert, um zu sehen, ob ebenfalls Reviews ohne expliziten Erklärungsbedarf erkannt werden. Jeder Phrase wurde eine Metrik zugeordnet und Phrasen mit einer Precision von 100% für den Filter genutzt. Bei einer Precision von über 90% oder einem Recall von über 5% wurde diskutiert, ob eine Nutzung für den Filter sinnvoll sein kann.

Für die Zusammenstellung der Phrasen bezüglich der Erkennung von implizitem Erklärungsbedarf, wurde neben den Guidelines mit dem SoftXPlain [36] Datensatz gearbeitet. Hier konnte festgestellt werden, dass impliziter Erklärungsbedarf häufig erst durch den Kontext erkennbar ist und eher auf Schlüsselwörter wie „irritated“ oder „don't know why“ zu erkennen ist. Deshalb baut der Filter für implizite Reviews überwiegend auf Schlüsselwörter auf, die Unklarheit andeuten. Mit einem Synonymlexikon wurden die im Datensatz identifizierten Phrasen und Begriffe ergänzt.

4.2. Datensatzgrundlage

Um sicherzustellen, dass nach der Filterung noch genügend Reviews bestehen, wurde ein umfangreicher Datensatz als Basis zusammengestellt. Da für weitere Forschung und eine ausführlichere Trainingsgrundlage angestrebt wird, 2000 Beispiele von Reviews mit impliziten und expliziten Erklärungsbedarfs zusammenzustellen, wird sich gezielt für eine große Anzahl an Reviews entschieden, die den Stammdatensatz für die Filterung bilden.

Dieser besteht aus Bewertungen der 15 beliebtesten Kategorien (Abbildung 3). Für jede dieser Kategorien wurden jeweils 3000 Apps sowohl aus dem App- als auch dem Playstore betrachtet, was zu einer Gesamtzahl von 6000 Apps pro Kategorien führt. Eine genaue Auflistung der betrachteten Apps findet sich im Anhang (A1, A2). Insgesamt ergibt das 90000 Reviews die als Filtergrundlage dienen. Für das Laden der Reviews wurde ein Tool verwendet, welches im Rahmen der Abschlussarbeit von Timo Kurz entwickelt wurde [32]. Jeder Review wird dabei einer global einzigartige ID (GUID) zugeordnet und zusammen mit der Sternebewertung und der App-ID in einem Datensatz gespeichert.

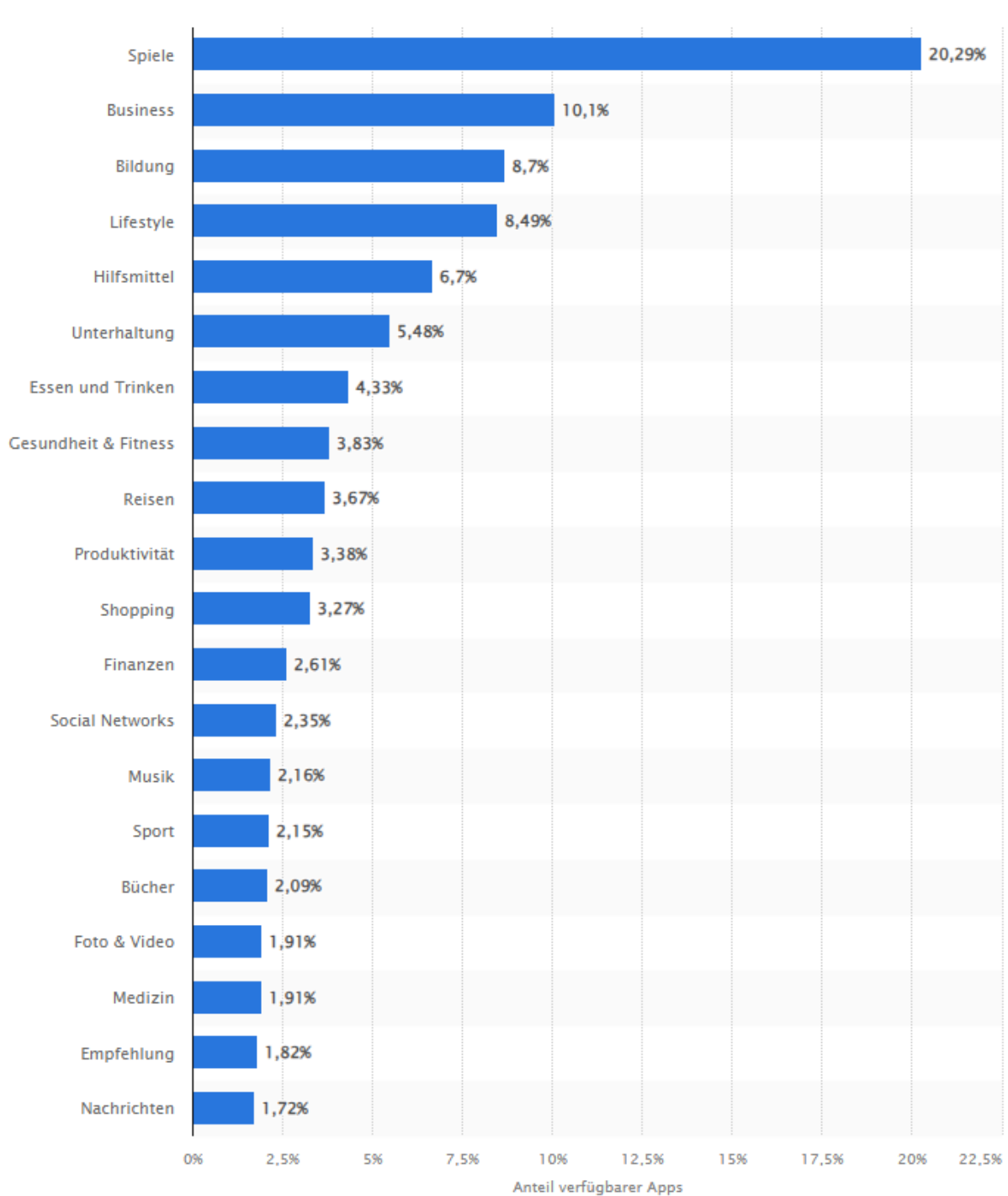


Abbildung 3: Anteil der Apps im App Store nach den Top-20-Kategorien (Juni 2023). Im Rahmen dieser Arbeit wurden die 15 beliebtesten Apps untersucht [40].

4.3 Filterung

Um den Datensatz auf expliziten und impliziten Erklärungsbedarf zu filtern, wurden die vorgestellten Filter eingesetzt. Von den ursprünglich 90000 Reviews konnten 1500 Reviews herausgefiltert, die potentiellen impliziten- und 1500 Reviews mit potentiellen expliziten Erklärungsbedarfs enthalten. Zudem wurden 1500 Reviews, bei denen keiner der Filter anschlug, zunächst als Review ohne Erklärungsbedarf betrachtet.

4.4 Datenanalyse und Validierung

Da nicht ausgeschlossen werden kann, dass unter den gefilterten Reviews auch Reviews enthalten sind, die nicht dem gewünschten Label entsprechen, ist es notwendig die Reviews manuell zu untersuchen und die Filterung zu überprüfen.

Um Verzerrungen bei der Überprüfung zu vermeiden, wird der Filterdatensatz von 3 Master-Informatik Studenten analysiert. Die Definitionen des Softwareinstituts sowie die Einteilung der Kategorien dienen hier als Grundlage für die Identifizierung von impliziten, expliziten oder keinem Erklärungsbedarf.

4.4.1 Vorgehen: Identifizierung Erklärungsbedarf

Die 4500 vorgefilterten Reviews wurden in 4 Iterationen zu dritt unabhängig von einander gelabelt. Zu diesem Zweck wurde ein Tool entwickelt, welches den Labelnden durch die Reviews führt (siehe Abbildung 4). Dabei kann der Labelnde der Review einer der fünf Label zuordnen:

0. Kein Erklärungsbedarf
1. Explizier Erklärungsbedarf
2. Impliziter Erklärungsbedarf
3. Erklärungsbedarf ohne Kontext
4. Diskussionsrunde

Review Checker

Last Review Next Review

Why do you keep making unnecessary changes?:I understand updating the app and trying to improve it to keep up with other forms of social media such as TikTok, Instagram, etc., but some changes they've made recently just do not make any sense. To be specific, I do not understand why they felt it was necessary to get rid of the following tab. Following people is almost obsolete now and we have completely lost control over the pins we see. Having pins suggested on the "for you" page is nice, but I still would like to see the pins created/shared by the people and boards I went out of my way to follow. Both can happen. Also, I am incredibly disappointed that they got rid of photo comments. I used Pinterest mainly to find baking recipes and posting photo comments was my way of keeping track of what I made. I also got inspired by seeing other people's takes on things. If I can't add photo comments, at least make a way for me to add things to my "tried" section.

You chose: Explicit Explanation Need

Index: 16

No Explanation Need (0)	Explicit Explanation Need (1)	
Implicit Explanation Need (2)	Discussion (4)	Explanation Need without clear Context (3)

Abbildung 4: Tool zur Typologisierung von Erklärungsbedarf

Da nicht ausgeschlossen werden kann, dass unter den gefilterten Reviews auch Reviews enthalten sind, die nicht dem gewünschten Label entsprechen, ist es notwendig die Reviews manuell zu untersuchen und die Filterung zu überprüfen.

Um Verzerrungen bei der Überprüfung zu vermeiden, wird der Filterdatensatz von 3 Master-Informatik Studenten analysiert. Die Definitionen des Softwareinstituts sowie die Einteilung der Kategoriegruppen dienen hier als Grundlage für die Identifizierung von impliziten, expliziten oder keinem Erklärungsbedarf.

4.4.2 Diskussionsgrundlage und Auffälligkeiten während des Labelns

Je nach Interpretation, können Beschreibungen von Problemen auch als impliziten Erklärungsbedarf verstanden werden. Um sich jedoch von der Identifizierung von Systemproblemen abzugrenzen, werden Reviews in denen nicht klar auf eine Verwirrtheit oder Unklarheit aufmerksam gemacht werden als Reviews ohne Erklärungsbedarf gehandhabt. Auch Fragen die nicht direkt mit dem System oder dem Unternehmen dahinter zu tun haben, werden nicht als Erklärungsbedarf klassifiziert. Auch rhetorische Fragen, die selbst vom Schreiber beantwortet werden, werden nicht als Erklärungsbedarf gelabelt.

Fälle in denen zu in den ersten Iterationen Uneinigkeit bestand und als Reviews ohne Erklärungsbedarf festgehalten wurden (IDs finden sich mit weiteren Beispielen im Anhang unter A4 und A5):

- *„Can't sign in. Useless. server error“*
- *“I donâ€™t know why this app has mostly 5 stars?”*
- *“[.] I even remembered a few albums I had downloaded but guess what? Since they aren't in any of the lists I have to search for the artist to find them, which doesn't work without internet!”*
- *“I like this app but your Amazon fresh is not working. It won't let me checkout. Please fix this“*

In Kapitel 5 werden Grenzfälle diskutiert bei denen sich auf Erklärungsbedarf geeinigt wurde. Diskussionsgrundlage waren ebenfalls Reviews in denen der Erklärungsbedarf nicht direkt mit der Software zusammenhängt, wie wenn es um die Zahlungsabwicklung,

Abos oder Prozesse ging. In Abstimmung mit den Kategoriegruppen für Erklärungsbedarf wurde nochmals definiert, dass Themen, die einer Taxonomie zugeordnet werden können auch als Reviews mit Erklärungsbedarf gelabelt werden sollen. Reviews in denen Erklärungsbedarf explizit festgestellt wird, es aber keine Triggerwörter gab, wurden Kontextabhängig entschieden.

4.4.3 Auswertung der Labelung

Insgesamt wurde eine Kappa Übereinstimmung von 0.63% erreicht, was auf eine substanzielle Zuverlässigkeit schließt. Die Auswertung der Vorfilterung zeigt eine hohe Precision bei der Filterung von Reviews mit expliziten Erklärungsbedarf und eher schwächere Ergebnisse bei der Identifizierung von Reviews mit impliziten Erklärungsbedarf. Während der Auswertung ist auch die Schwächen dieser Methode auffallen, wie das falschgeschriebene Ausdrücke wie „I d nt know“ von der Methode nicht als Hinweis auf potentiellen impliziten Erklärungsbedarfs gesehen werden oder dass der Filter „clueless“ hauptsächlich Reviews filterte, in der sich auf die gleichnamige Serie bezogen wird, jedoch keinen Erklärungsbedarf andeuten.

Durch die Betrachtung der Ergebnisse lässt sich die **F1**, wie präzise sich ein Datensatz aus Reviews mit impliziten und expliziten Erklärungsbedarf, durch eine gezielte Nutzung von sprachlichen Filtern zusammenstellen lässt beantworten:

Durch den Einsatz von Filtern lassen sich mit einer Precision von 79,8% Reviews mit expliziten Erklärungsbedarfs rausfiltern (Tabelle 3). Bei Reviews mit impliziten Erklärungsbedarf erreicht die Filtermethode eine Precision von 40,8%. Falls nicht zwischen implizitem und explizitem Erklärungsbedarf unterschieden werden soll, erreicht der Filter eine Precision von 68,2%.

Vorfilterung	Als Explizit gelabelt	Als implizit gelabelt	Unklarere Kontext	Kein Erklärungsbedarf festgestellt	Precision
Explizit	1197	64	24	215	79,8%
Implizit	173	612	24	22	40,8%
Kein EN	82	51	11	1356	90,4%

Tabelle 3: Auswertung der Vorfilterung

5. Datensatzanalyse

Als Vorbereitung für die Wörterbuchmethode und zur Kategorisierung des Erklärungsbedarfs werden die Reviews des erstellten Datensatzes auf Phrasen untersucht, die einen Erklärungsbedarf andeuten und die Kategorie in Hinsicht auf die Taxonomie (Kapitel 2.2.2) festgehalten, was den Datensatz näher beleuchtet und für zukünftige Forschungsfragen relevant sein kann. Um die Objektivität dieser Labelung zu gewährleisten wird die Analyse von 3 Personen größtenteils unabhängig voneinander durchgeführt. Der Prozess ist in Abbildung 5 beschrieben.

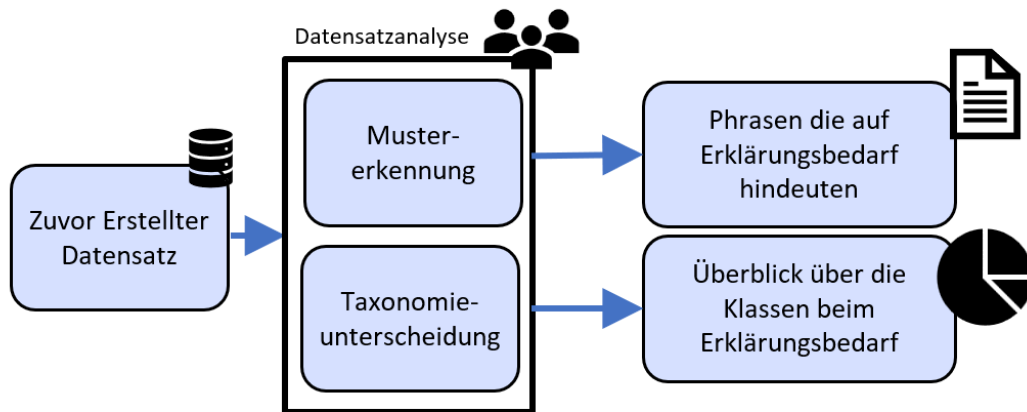


Abbildung 5: Vorgehen während der Datensatzanalyse

5.1 Eingesetztes Datenanalyse Tool

MaxQDA

Bei MaxQDA handelt es sich um eine Datenanalyse Tool, welches es ermöglicht unstrukturierte Daten effizient analysieren zu können. Dabei können die zu analysierenden Texte importiert und relevante Stellen codiert werden. Durch visuelle und strukturierte Anordnungen lassen sich die codierten Daten effizient durchsuchen [28]. Es lassen sich dabei Projekte zusammenführen und Codierungen verschiedener Codierer effizient zusammenführen.

Gemeinsam mit einem Kommilitonen wurden die Reviews auf die Kategorien iterationsweise markiert. Durch regelmäßigen Austausch und Diskussionsrunden wurden Unklarheiten reduziert. In der Abbildung 6 ist eine Beispielcodierung abgebildet. Der Teil

„Don't know why it had to change“ deutet hier auf Erklärungsbedarf in Bezug eine Änderung der GUI hin. Die markierte Stelle wurde sowohl mit Designentscheidung, als auch mit „Änderung“ markiert. Bei der Bestimmung der Kategorien gab es teilweise Unstimmigkeiten, die iterativ besprochen wurden. Insgesamt wurde eine Kappa Übereinstimmung von 55,9%, während der ersten Iteration und eine Übereinstimmung 60,3% während der zweiten Iteration erreicht erreicht.

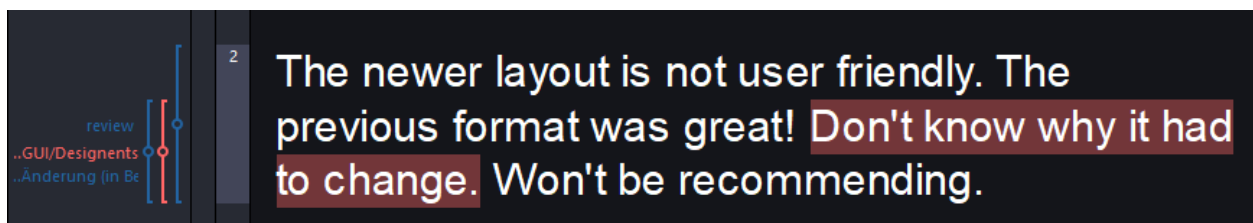


Abbildung 6: MaxQDA Interface am Beispiel einer Codierung

5.2 Auffälligkeiten und Probleme während der Unterteilung in die Kategorien

Bei der Labelung gab es unter anderem Unstimmigkeiten was die Kategorien „Operation“ und „Unerwartetes Verhalten“ betrifft. Je nach Aussage kann ein Problem bei der Interaktion teils auf den User und Teils auf das System zurückzuführen sein. Die Review: *“Hi, I've recently upgraded to a Galaxy S21 and toggle WiFi no longer works. How do I fix this?”* beschreibt gezielt ein Problem. Der Erklärungsbedarf bezieht sich jedoch nicht auf eine Unklarheit in Bezug auf das Systemverhalten, sondern auf eine Fragestellung bezüglich der Problemlösung. Deshalb wird solch ein Fall als Operation gewertet.

Ähnlich bei der Unterscheidung zwischen der Labelung „Bug“ und „Unerwartetem Verhalten“. Deshalb wurde sich nach der ersten Iteration (nach 1000 Reviews) darauf geeinigt die Reviews danach zu labeln, als dass User die Situation wahrnehmen. Deshalb

wird eine Problemstellung die vom User als unerwartetes Systemverhalten wahrgenommen, vom Labelnden jedoch als Error verstanden wird, trotzdem als „Unerwartetes Systemverhalten“ gemeldet, insofern kein Systemabsturz gemeldet oder explizit ein Systemfehler beschrieben wird. Im Zweifelsfall wurde die Review als Metainformation gelabelt, das heißt, dass die Passage nicht einem eindeutigen Typen zugeordnet werden kann.

Des Weiteren war die Abgrenzung zwischen „Metainformationen“ und „Business“ in einige Fällen die das Business betreffen, aber auch teilweise mit der Software zu tun haben nicht eindeutig. Es wurde sich in solchen Fällen darauf geeinigt Erklärungsbedarf der eine Schnittstelle zur Software aufweist als „Meta“ gelabelt wird („How can I contact support about a refund“) oder sowohl „Meta“ als auch Business gelabelt werden soll („I dont know why I was charged this fee“). Beispiele für die jeweiligen Kategorien finden sich im Anhang (A4, A5).

5.1. Datensatzvorstellung

Insgesamt ergeben das 1456 Reviews mit expliziten, 727 mit impliziten und 184 mit impliziten und expliziten Erklärungsbedarf, welche als explizit betrachtet werden. Die Verteilung ist in Tabelle 4 abgebildet und in Abbildung 7 und 8 Visualisiert.

Daraus ergibt sich, dass die Methodik einen großen Anteil an Reviews mit Erklärungsbedarf in Bezug auf Systemverhalten und Interaktion zusammengetragen hat. Durch die Betrachtung der Ergebnisse lässt sich die Forschungsfrage **F2** beantworten, wonach untersucht werden sollte ob eine filterbasierte Datensatzerstellung gewährleistet kann, dass alle möglichen Informationstypen von Erklärungsbedarf abgedeckt werden:

Eine Gleichverteilung auf alle Kategorien kann durch die genutzte Methodik nicht garantiert werden. Die Verteilung zeigt eine Überrepräsentation der Aspekte

„Unerwartetes Systemverhalten“, „Operation“ und „Metainformationen“.
„Sicherheitsrelevante Themen“, sowie die Kategorien „Konsequenzen“, „Einführung“
und „Algorithmen“ traten eher selten auf.

1. Direkte Systemaspekte
 - a. Interaktion
 - i. Operation (452 exp, 68 imp)
 - ii. Einführung (12 exp, 40, imp)
 - iii. Navigation (43 exp, 35 imp)
 - b. Systemverhalten
 - i. Algorithmus (20 exp, 35 imp)
 - ii. Konsequenzen (8 exp, 4 imp)
 - iii. Unerwartetes Systemverhalten (396 imp, 369 exp)
 - iv. Bugs/Abstürze (64 imp, 62 exp)
 - c. Designentscheidungen (84 imp, 49 exp)
 - d. Security
 - i. Privacy (28 imp, 6 exp)
 - ii. Vulnerability (3 exp, 0 imp)
 - e. MetaInformationen (363 imp, 192 exp)
2. Indirekte Systemaspekte
 - a. Domainwissen
 - i. Begrifflichkeiten (3 exp, 2 imp)
 - ii. Systemspezifische Aspekte (85 exp, 48 imp)
3. Timing/Context
 - a. Änderung/Vergangenheit (178 exp, 100 imp)
 - b. Zukunftsplan (107 exp, 4 imp)
4. Business (322 exp, 141 imp)

Tabelle 4: Taxonomieüberblick

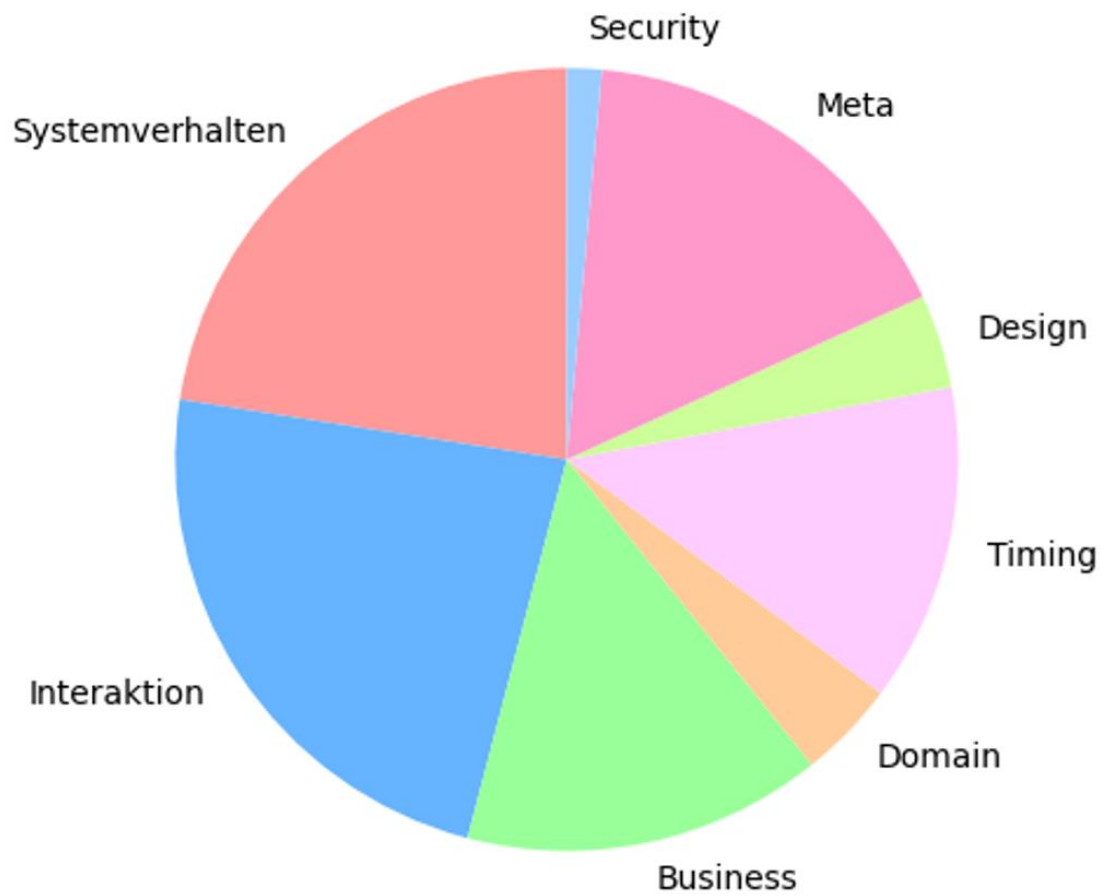


Abbildung 8: Verteilung der Kategorien in Bezug auf expliziten Erklärungsbedarf

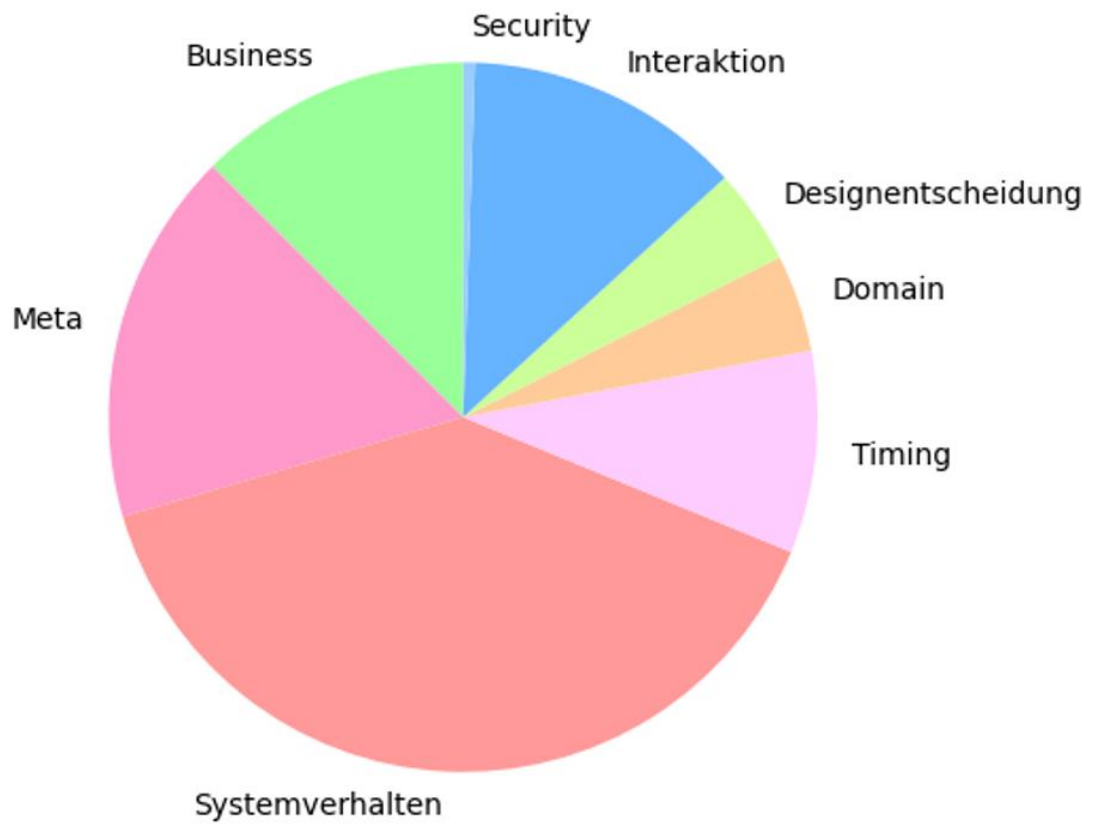


Abbildung 9: Verteilung der Kategorien in Bezug auf impliziten Erklärungsbedarf

6. Implementierung

6.1 Trainingsdatensatz

Mit dem Ziel, automatisiert Reviews mit Erklärungsbedarf möglichst effizient zu erkennen, werden verschiedene Ansätze implementiert und evaluiert. Neben der Wörterbuchmethode, die das gezielte Einsetzen von Filtermethoden verwendet, werden auch Modelle des maschinellen Lernens und des Deep Learnings trainiert. In dieser Arbeit wird zwischen explizitem und implizitem Erklärungsbedarf unterschieden. Daher wird auch analysiert, wie gut sich Modelle in Abhängigkeit vom Typ des Erklärungsbedarfs verhalten.

Grundlage für die Trainingsdaten ist zuvor erstellte Datensatz, der sich aus 727 Reviews mit implizitem, 1452 Reviews mit explizitem Erklärungsbedarf und 2262 Reviews ohne Erklärungsbedarf zusammensetzt. Zudem enthält der Datensatz 60 Reviews, die Erklärungsbedarf aufweisen, bei denen jedoch kein klarer Kontext erkennbar ist. Im Rahmen der Analyse werden diese Reviews als solche ohne Erklärungsbedarf behandelt, da sie nicht der Definition von Erklärungsbedarf entsprechen. Vor dem Training wird dieser Datensatz in Trainings- und Validierungsdaten unterteilt (siehe Tabelle 5), dabei werden 90% der Daten den Trainings und 10% für die Validierung der Modelle eingesetzt. Die implementierten Modelle werden anhand des Trainingsdatensatzes optimiert und anschließend am Validierungsdatensatz im nächsten Kapitel ausgewertet.

	Explizit EN	Implizit EN	Kein EN
Trainingsdatensatz	1307	654	2089
Validierungsdatensatz	145	73	233

Tabelle 5: Datensatzaufteilung zwischen Trainings- und Validierungsdaten

6.2 Wörterbuchmethode

Die erkannten Muster, die während der Kategorieunterteilung in [Kapitel 5](#) festgehalten wurden, werden anschließend daraufhin untersucht, inwiefern von diesen auf expliziten, impliziten und allgemeinen (explizit und implizit zusammen betrachtet) Erklärungsbedarf geschlossen werden kann. Nach einer individuellen Analyse der Phrasen, werden diese gezielt durch eine Parameteroptimierung kombiniert, wodurch für jeden der 3 Typen von Erklärungsbedarf Modelle erstellt werden, die auf einen hohen Recall, eine hohe Precision und F1 Wert abzielen. Da jeder Phrase Metriken zugeordnet werden, wird im Rahmen dieser Arbeit der Prozess als Wörterbuchmethode verstanden. Der Prozess ist in [Abbildung 10](#) abgebildet.

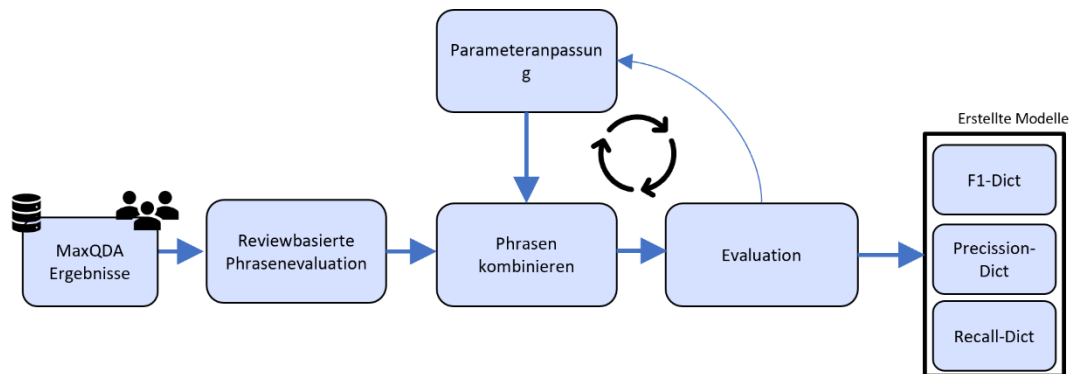


Abbildung 10: Vorgehen für die Wörterbuchmethode

6.2.1 Individuelle Phraseneinordnung

Für die identifizierten Phrasen wird individuell überprüft, wie gut diese zur Identifizierung von Reviews mit implizitem und explizitem Erklärungsbedarf geeignet sind. Orientiert an dem Modell von Zibran et. Al [65] werden den einzelnen Phrasen Scores zugeordnet und hierarchisch geordnet. So wird für jede Phrase untersucht, welche Metriken sich ergeben, falls Reviews mit impliziten, expliziten oder allgemeingültigen Erklärungsbedarf auf diese Phrasen hin untersucht werden. In Tabelle 6 sind beispielhaft die Metriken für die Erkennung von Reviews mit explizitem Erklärungsbedarf dargestellt.

Ausdruck	Precision	Recall	F1
why (cant (can not) can('' o)t) (i we you)	0,96	0,11	0,1902
How can I	0,88	0,069	0,13
Doesn't do	0,5	0,0045	0,009
Worried about	0,10	0,004	0,007

Tabelle 6: Ausschnitt der Metriken bezüglich der Erkennung von explizitem Erklärungsbedarf

6.2.2 Phrasenzusammenstellung

Um sicherzustellen, dass möglichst viele Reviews mit Erklärungsbedarf korrekt als solche identifiziert werden, wird bei der Wörterbuchmethode neben einer hohen Precision auch ein hoher Recall angestrebt. Da Phrasen mit hoher Precision nicht zwangsläufig einen hohen Recall aufweisen, werden zur umfassenderen Erkennung verschiedene Phrasen kombiniert. Insgesamt werden zur Erkennung je Typ von Erklärungsbedarfs 3 Modelle trainiert, die auf unterschiedliche Metriken (Precision, Recall, F1) hin optimiert wurden. Dadurch kann ebenfalls untersucht werden, inwiefern die Methode in Bezug auf andere Ansätze einzuordnen ist.

Zu diesem Zweck wurden die zuvor individuell evaluierten 153 Phrasen nach eingehender Analyse kombiniert. Um den Suchraum der möglichen Kombinationen zu begrenzen, wurden mittels Gridsearch Schwellenwerte (Threshold Values) ermittelt, die Mindestanforderungen an die Precision, Recall und F1-Werte der einzelnen Phrasen setzen (siehe entsprechende Tabelle). Für jede dieser Kombinationen wurden die

Metriken zur Evaluation festgelegt. Insgesamt wurden jeweils drei Modelle für die Erkennung von explizitem, implizitem und allgemeinem Erklärungsbedarf erstellt, die auf hohe Werte in Recall, Precision und F1 optimiert wurden. Die optimal bestimmten Schwellenwerte sind in Tabelle 7 abgebildet.

	Precision-Grenze	Recall-Grenze	F1-Grenze
Filter_en_f1	0,72	-	-
Filter_exp_f1	0,76	-	0,06
Filter_imp_f1	0,38	0,005	0,005
Filter_en_Recall	0,1	-	-
Filter_exp_Recall	0,1	-	-
Filter_imp_Recall	0,2472	-	-
Filter_en_Precision	0,96	-	-
Filter_exp_Precision	0,96	0,22	0,36
Filter_imp_Precision	0,62	-	0,005

Tabelle 7: Optimale Konfigurationen der Filtermethoden

Das Modell, das auf eine hohe Precision optimiert wurde, erreichte im Trainingsdatensatz eine Precision von 96,05%, wies jedoch nur einen Recall von 22,41% auf (Tabelle 8). Das auf einen hohen Recall ausgerichtete Modell erzielte in dieser Metrik 99,1%, jedoch lag die Precision bei lediglich 47%. Das Modell, welches auf einen optimalen F1-Wert trainiert wurde, erreichte einen F1-Wert von 84,1%. Dabei betrug die Precision 76,02% und der Recall ebenfalls 95,1%. Das F1-Modell erreicht den höchsten AUC-Score, welcher Aussagekraft darüber gibt, wie gut das Modell zwischen zwei Klassen unterscheiden kann.

	Explicit EN			No explicit EN			AUC	F1-Makro
	Prec	Recall	F1	Prec	Recall	F1		
exp_f1	0,76	0,95	0,84	0,97	0,85	0,91	0,90	0,87
exp_rec	0,47	0,99	0,63	0,99	0,45	0,62	0,72	0,63
exp_prec	0,96	0,22	0,36	0,72	1,0	0,72	0,61	0,60

Tabelle 8: Expliziter Erklärungsbedarf. Trainingsdurchlauf, mit Fokus auf optimierten F1, Precision und Recall Wert

Die Metriken der Modelle zur Identifikation von implizitem Erklärungsbedarf (siehe Tabelle 9) sind generell niedriger als die zur Erkennung von explizitem Erklärungsbedarf. Die Evaluationsergebnisse für die Erkennung von Reviews mit implizitem Erklärungsbedarf ähneln den Trainingsergebnissen. Das auf hohe Precision trainierte Modell erreichte eine Precision von 63%, erreicht dabei jedoch schwache Recall-Werte. Das auf hohen Recall optimierte Modell erreicht in dieser Kategorie einen Wert 94%, hat aber eine niedrige Precision von 24% im Training und 25% in der Evaluation.

	Implicit EN			No implicit EN				
Kind	Prec	Recall	F1	Prec	Recall	F1	AUC	F1-Makro
imp_f1	0,52	0,82	0,55	0,96	0,78	0,86	0,80	0,71
imp_rec	0,25	0,94	0,39	0,97	0,55	0,61	0,69	0,50
imp_prec	0,63	0,07	0,12	0,84	0,99	0,91	0,53	0,52

Tabelle 9: Impliziter Erklärungsbedarf: Trainingsdurchlauf, mit Fokus auf optimierten F1, Precision und Recall Wert

Modelle, die allgemein darauf ausgerichtet sind, Erklärungsbedarf zu identifizieren (unabhängig davon, ob implizit oder explizit), weisen sowohl auf den Trainings-, als auch beim Evaluationsdatensatz, hohe Metriken in Ihrer Disziplin auf (Tabelle 10). So erreicht das auf einen F1-Score optimierte Modell sowohl beim Training, als auch bei der Evaluation einen F1-Score von 83%. Die anderen Modelle erzielten in ihren optimierten Disziplinen Ergebnisse von über 90%. In Kapitel 7 wird untersucht, wie gut die Modelle auf Trainingsfremde Daten performen und sie sich von anderen Ansätzen unterscheiden.

	Explanation Need			No Explanation Need				
Kind	Prec	Recall	F1	Prec	Recall	F1	AUC	F1-Makro
en_f1	0,78	0,93	0,83	0,91	0,70	0,79	0,81	0,81
en_rec	0,69	0,98	0,81	0,96	0,57	0,72	0,77	0,76
en_prec	0,96	0,17	0,28	0,66	0,99	0,71	0,59	0,50

Tabelle 10: Allgemeiner Erklärungsbedarf: Trainingsdurchlauf, mit Fokus auf optimierten F1, Precision und Recall Wert

6.3 Machine- und Deep-Learning Modelle

6.3.1 Genutzte Frameworks

Die in Kapitel 2 beschriebenen Methoden des maschinellen Lernens und Deep Learning wurden mittels Python umgesetzt. Zur Datenanalyse kam die pandas-Bibliothek zum Einsatz, welche die Analyse und Manipulation tabellarischer Daten erleichtert. Mit dieser wurden die erstellten Datensätze eingelesen. Das Natural Language Toolkit (NLTK) wurde für die Verarbeitung natürlicher Sprache verwendet. Im Zuge der Datenvorverarbeitung wurden Texte in Kleinbuchstaben umgewandelt, tokenisiert und nicht-numerische Wörter entfernt. Anschließend wurde eine binäre Variable erstellt. Zur Sicherstellung eines ausgewogenen Klassenverhältnisses fand ein Undersampling statt. Die Daten wurden dann mittels Feature-Extraktion für die maschinellen Lernmodelle vorbereitet. Mit der sklearn-Bibliothek wurden verschiedene maschinelle Lernmethoden wie Naive Bayes, Support Vector Machine, logistische Regression, KNN und Entscheidungsbäume trainiert. Diese Bibliothek unterstützt auch die Parameteroptimierung durch Gridsearch und die Evaluierung mittels Cross-Validation.

BERT und SetFit

Für die Datenverwaltung und Ergebnisevaluierung wurden die Bibliotheken numpy und sklearn genutzt. Das vortrainierte BERT-Modell wurde über die Open-Source-Bibliothek transformers geladen, die eine Schnittstelle zu Hugging Face bietet. Bei SetFit kommt ein sentence-transformer von Hugging Face zum Einsatz. Die torch-Bibliothek unterstützt nicht nur die Modellverwaltung hinsichtlich der Hardware, sondern auch das GPU-Training. Zur Sicherung der Reproduzierbarkeit der Ergebnisse wurde ein vordefinierter

Seed verwendet. Nach dem Datenimport erfolgte die Tokenisierung, um BERT und SetFit die Datenverarbeitung zu ermöglichen. In einem weiteren Schritt wurde eine benutzerdefinierte Klasse entwickelt, die die tokenisierten Daten in PyTorch-Datensätze umwandelt. Das vortrainierte Modell diente als Basis für ein angepasstes Modell, welches an die Datenstruktur angepasst wurde. Nach dem Training dieses Modells erfolgte die abschließende Evaluierung der Ergebnisse.

6.3.2 Training der Deep-Learning Modelle

Um in Bezug auf die Beantwortung der Forschungsfrage 4 zu untersuchen, inwiefern Modelle auf die Erkennung von unterschiedlichen Typen von Erklärungsbedarf optimiert werden können. Zu dem Zweck werden sowohl Multilabelklassifikatoren, als auch Binärklassifikatoren trainiert und evaluiert. Die Binären Klassifikatoren unterscheiden die Reviews dabei zwischen:

- explizitem und keinem explizitem Erklärungsbedarf (impliziten und keinem Erklärungsbedarf)
- zwischen implizitem und keinem impliziten Erklärungsbedarf (explizitem und keinem Erklärungsbedarf)
- zwischen allgemeinem (impliziten und expliziten) und keinem Erklärungsbedarf

Um eine Klassenbalance zu gewährleisten werden vor dem Training die Datensätze reduziert (Undersampling). Die Aufteilung ist in Tabelle 11 dargestellt. Bei den Binären-Klassifikatoren, werden dabei Reviews, die nicht dem Fokus des Trainings entsprechen, als Reviews ohne Erklärungsbedarf gesehen.

Datensatzart	Exp Reviews	Imp Reviews	No EN Reviews
Multilabel	654	654	654
Binär-EN	1307	654	653
Binär-Exp	1307	654	653
Binär-Imp	327	654	327

Tabelle 11: Trainingsdatensätze für die Machine- und Deep-Learning Modelle

7. Evaluation der Ergebnisse

In diesem Teil der Arbeit werden die zuvor Trainierten Modelle auf unterschiedliche Datensätzen evaluiert. Zunächst werden die Modelle daraufhin untersucht, wie sie auf einen Ausgeglichenem Datensatz performen. Dadurch kann der Trainingserfolg mit Trainingsfremddaten untersucht werden.

Anschließend wird untersucht, wie gut die Daten auf einem unausgeglichenem Datensatz performen. Dafür werden die Evaluationssätze so angepasst, dass der Anteil an Erklärungsbedarf den realen Verhältnissen angepasst wird. Der Anteil an Reviews mit expliziten Erklärungsbedarf beschränkt sich laut Unterbusch et. Al [1] auf 5%. Da es keine Daten zu dem Anteil der Reviews mit impliziten Erklärungsbedarf gibt, wird zur Evaluation angenommen, dass sich dieser ebenfalls auf 5% beschränkt. Darauf aufbauend, werden die Datensätze reduziert und die Modelle darauf ausgewertet.

7.1 Validierung auf Basis eines ausgeglichenen Datensatzes

Wie bei der Zusammenstellung der Trainingsdatenmodelle in Kapitel 6.3.2, wird der Validierungsdatensatz so angepasst, dass eine Klassenbalance gegeben ist. Der Aufbau der Validierungsdatensätze ist in Tabelle 8 dargestellt.

Datensatzart	Exp Reviews	Imp Reviews	No EN Reviews
Dict/Binär-Exp	145	72	73
Dict/Binär-Imp	36	73	37
Dict/Binär-EN	145	73	218
Multilabel	73	73	73

Tabelle 12: Evaluationsdatensätze (ausgeglichen)

7.1.1 Detektion von Reviews mit expliziten Erklärungsbedarf

Zunächst wird untersucht, wie gut binäre Modelle dabei abschneiden, Reviews mit explizitem Erklärungsbedarf zu erkennen. Die Ergebnisse sind in Tabelle 13 dargestellt.

Die höchste Präzision bei der Erkennung von explizitem Erklärungsbedarf liefert dabei die Wörterbuchmethode, die auf eine hohe Präzision optimiert wurde (97%). Darauf folgen die Deep Learning-Ansätze SetFit und BERT, beide mit einem Ergebnis von 91%. Die auf eine hohe F1-Performance optimierte Wörterbuchmethode erreicht 87,34%, gefolgt von den Machine Learning-Ansätzen, wobei AdaBoost mit 83,59% am besten abschneidet.

Unter dem Recall-Wert zeigt sich, wie groß der Anteil der korrekt als Erklärungsbedarf klassifizierten Ergebnisse ist. Hier erreicht die Wörterbuchmethode die besten Ergebnisse. Mit Recall-Werten von über 91% folgen die Deep Learning-Ansätze. Die Machine Learning-Methoden liegen zwischen 52-83%, wobei RandomForest, mit Count Vector trainiert, am stärksten abschneidet.

Der F1-Wert zeigt ein Gleichgewicht zwischen Präzision und Recall. Hier schneiden die Machine Learning-Methoden generell schlechter ab als die Deep Learning-Methoden. Die besten Ergebnisse erzielen SetFit, die Wörterbuchmethode und BERT.

Die Trennungsfähigkeit der Modelle wird durch den AUC-Wert bestimmt. Die Deep Learning-Ansätze BERT (91,03%) und SetFit (91,37%) sowie die Wörterbuchmethode (90,58%) überzeugen hier. Bei den ML-Modellen schneidet RandomForest mit TFIDF (79%) am besten ab.

Insgesamt überzeugen die Deep Learning-Modelle mehr als die ML-Modelle. Je nach betrachteter Metrik sind SETFIT oder BERT die Spitzenreiter. Bei den Machine Learning-Methoden schneiden KNN und Naive Bayes weniger gut ab, während AdaBoost und RandomForest bessere Ergebnisse liefern. Bezüglich der Transformer zeigt sich, dass TFIDF genauere Ergebnisse liefert.

Es ist auch zu beachten, dass die Metriken bei der Erkennung von nicht explizitem Erklärungsbedarf konstant über denen der Erkennung von Reviews mit Erklärungsbedarf liegen. Die Rangfolge der Klassifikatoren bleibt hierbei ähnlich.

Model	Exp EN			No exp EN			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
NB (TF)	0,64	0,81	0,72	0,75	0,55	0,63	0,68	0,68
NB(CV)	0,71	0,77	0,74	0,75	0,68	0,71	0,73	0,73
SVM (TF)	0,75	0,74	0,74	0,74	0,75	0,75	0,74	0,74
SVM (CV)	0,82	0,71	0,76	0,74	0,84	0,79	0,78	0,77
RF (TF)	0,80	0,81	0,80	0,80	0,79	0,80	0,80	0,80
RF (CV)	0,77	0,83	0,80	0,82	0,74	0,78	0,79	0,79
LR (TF)	0,76	0,70	0,73	0,72	0,77	0,75	0,74	0,74
LR (CV)	0,80	0,71	0,75	0,74	0,83	0,78	0,77	0,77
AB (TF)	0,84	0,74	0,78	0,77	0,86	0,81	0,80	0,80
AB (CV)	0,81	0,73	0,77	0,75	0,83	0,79	0,78	0,78
KNN (TF)	0,65	0,54	0,59	0,61	0,71	0,66	0,63	0,63
KNN (CV)	0,68	0,52	0,59	0,61	0,76	0,68	0,64	0,64
SetFit	0,92	0,91	0,91	0,91	0,92	0,91	0,95	0,91
BERT	0,91	0,91	0,91	0,91	0,91	0,91	0,96	0,91
Dict_f1	0,87	0,95	0,91	0,95	0,86	0,90	0,91	0,91
Dict_rec	0,68	0,99	0,80	0,99	0,52	0,68	0,76	0,74
Dict_prec	0,97	0,21	0,35	0,56	0,99	0,71	0,60	0,53

Tabelle 13: Evaluationsergebnisse der binären Klassifizierer, die expliziten Erklärungsbedarf erkennen (an einem ausgeglichenem Datensatz)

7.1.2 Detektion von Reviews mit impliziten Erklärungsbedarf

Die Evaluationsergebnisse zur Erkennung von Reviews mit implizitem Erklärungsbedarf sind in Tabelle 14 dargestellt. Insgesamt zeigen die Deep-Learning-Modelle die besten Ergebnisse in allen Kategorien. Hinsichtlich der Precision Rate erzielen SetFit und BERT den höchsten Wert von 91,78%. Die Wörterbuchmethode kommt auf eine Präzision von 75%, während die traditionellen Machine-Learning-Methoden Werte zwischen 49,35% und 66,67% aufweisen.

Sowohl SETFIT als auch BERT haben einen Recall-Wert von 91,78%. Im Vergleich dazu liegen die Recall-Werte der Machine-Learning-Methoden zwischen 40% und 80,82%. Die Wörterbuchmethode erreicht hierbei 82,19%. Es ist bemerkenswert, dass ein Wörterbuchansatz, der auf hohe Precision optimiert wurde, eine Präzision von 100% bei einem Recall von nur 10% erzielt.

In Bezug auf die F1, AUC, F1-Macro und F1-Micro Werte übersteigen SETFIT und BERT mit Werten von über 91% die Performance. Die Wörterbuchmethode folgt mit Werten zwischen 75% und 77%.

Zusammenfassend lässt sich feststellen, dass SETFIT und BERT insgesamt besser abschneiden als die Wörterbuchmethode. Dennoch übertrifft die Wörterbuchmethode in allen Metriken die traditionellen Machine-Learning-Methoden.

Model	Imp EN			No IMP En			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
NB (TF)	0,52	0,70	0,59	0,53	0,34	0,42	0,52	0,50
NB(CV)	0,55	0,75	0,64	0,61	0,38	0,47	0,57	0,55
SVM (TF)	0,57	0,64	0,60	0,59	0,51	0,54	0,58	0,57
SVM (CV)	0,65	0,64	0,65	0,65	0,66	0,65	0,65	0,65
RF (TF)	0,58	0,75	0,65	0,65	0,45	0,53	0,60	0,59
RF (CV)	0,60	0,81	0,69	0,71	0,47	0,56	0,64	0,63
LR (TF)	0,62	0,67	0,64	0,64	0,59	0,61	0,63	0,63
LR (CV)	0,63	0,62	0,63	0,63	0,64	0,64	0,63	0,63
AB (TF)	0,58	0,70	0,63	0,62	0,49	0,55	0,60	0,59
AB (CV)	0,67	0,68	0,68	0,68	0,66	0,67	0,67	0,67
KNN (TF)	0,52	0,45	0,48	0,51	0,58	0,54	0,51	0,51
KNN (CV)	0,49	0,52	0,51	0,49	0,47	0,48	0,49	0,49
SetFit	0,92	0,92	0,92	0,92	0,92	0,92	0,96	0,92
BERT	0,92	0,92	0,92	0,92	0,92	0,92	0,94	0,92
Dict_f1	0,75	0,82	0,78	0,80	0,73	0,76	0,77	0,77
Dict_rec	0,51	0,92	0,65	0,57	0,11	0,18	0,51	0,42
Dict_prec	1,00	0,10	0,18	0,53	1,00	0,69	0,55	0,43

Tabelle 14: Evaluationsergebnisse der binären Klassifizierer, die impliziten Erklärungsbedarf erkennen (an einem ausgeglichenem Datensatz)

7.1.3 Detektion von Reviews mit allgemeinem Erklärungsbedarf

Anschließend wird untersucht, wie gut binäre Modelle in der Lage sind, Reviews mit explizitem Erklärungsbedarf zu identifizieren. Die Evaluationsergebnisse bezüglich der Detektion von allgemeinem Erklärungsbedarf reihen sich zwischen denen der anderen binären Klassifikatoren ein (Tabelle 15). Auch in diesem Bereich zeigen die Deep-Learning-Modelle die überzeugendsten Ergebnisse in den aussagekräftigen Metriken. Die Wörterbuchmethoden zeichnen sich ebenfalls durch hohe Recall-Werte bei den F1- und Recall-optimierten Modellen aus (über 90%). Das auf die Precision trainierte Wörterbuchmodell weist zwar nur einen Recall von 16% auf, erreicht jedoch eine Precision von 97%, was 10% über den zweitbesten Ergebnissen im Vergleich liegt.

Model	Explanation Need			No Explanation Need			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
NB (TF)	0,68	0,83	0,75	0,79	0,61	0,68	0,72	0,72
NB(CV)	0,68	0,84	0,75	0,79	0,61	0,69	0,72	0,72
SVM (TF)	0,78	0,84	0,81	0,83	0,76	0,79	0,80	0,80
SVM (CV)	0,85	0,78	0,81	0,79	0,86	0,82	0,82	0,82
RF (TF)	0,73	0,86	0,79	0,83	0,69	0,75	0,78	0,77
RF (CV)	0,72	0,83	0,77	0,80	0,67	0,73	0,75	0,75
LR (TF)	0,78	0,85	0,82	0,84	0,76	0,80	0,81	0,81
LR (CV)	0,82	0,78	0,80	0,79	0,83	0,81	0,81	0,80
AB (TF)	0,81	0,77	0,79	0,78	0,83	0,80	0,80	0,80
AB (CV)	0,85	0,72	0,78	0,75	0,87	0,81	0,79	0,79
KNN (TF)	0,72	0,39	0,51	0,58	0,84	0,69	0,62	0,60
KNN (CV)	0,73	0,63	0,68	0,68	0,77	0,72	0,70	0,70
SetFit	0,87	0,94	0,90	0,94	0,86	0,89	0,90	0,92
BERT	0,85	0,90	0,88	0,89	0,84	0,87	0,87	0,93
Dict_f1	0,74	0,91	0,82	0,89	0,68	0,77	0,80	0,80
Dict_rec	0,70	0,97	0,81	0,95	0,58	0,72	0,78	0,77
Dict_prec	0,97	0,16	0,27	0,54	1,00	0,70	0,58	0,48

Tabelle 15: Evaluationsergebnisse der binären Klassifizierer, die Erklärungsbedarf erkennen (an einem ausgeglichenem Datensatz)

7.1.3 Multiklassen-Klassifikation

Bei der Auswertung der Multiklassen-Klassifikatoren zeigen sich insgesamt bessere Metriken für die Erkennung von explizitem im Vergleich zu implizitem Erklärungsbedarf (vgl. Tabelle 12).

Für die Erkennung von Reviews mit explizitem Erklärungsbedarf erzielten SetFit und BERT die höchsten Precision-Werte von über 85%. Im Gegensatz dazu bewegen sich

die Präzisionswerte der Machine-Learning-Methoden im Bereich von 46% bis 69%. Die Deep-Learning-Methoden, insbesondere SetFit, zeigen ihre Stärken auch bei den Recall- und F1-Werten, mit Werten über 89% bzw. 88%. Adaboost ist das leistungsstärkste ML-Modell in dieser Kategorie und erreicht einen Recall von 66% und einen F1-Wert von 67%.

Die Ergebnisse für die Erkennung von implizitem Erklärungsbedarf und ohne Erklärungsbedarf ähneln denen für expliziten Erklärungsbedarf, allerdings sind die Metriken insgesamt etwas schlechter. Hierbei ist SETFIT das Modell mit den besten Ergebnissen: Eine Precision von 81%, ein Recall von 89% und ein F1-Wert von 85%. Auch bei der Erkennung von Reviews ohne Erklärungsbedarf führt SetFit.

	No en			Exp en			Imp EN			F1- macro	F1- micro	AUC
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1			
NB (TF)	0,70	0,38	0,50	0,46	0,56	0,51	0,47	0,58	0,52	0,51	0,51	0,71
NB(CV)	0,76	0,36	0,49	0,50	0,60	0,55	0,52	0,68	0,59	0,54	0,55	0,72
SVM (TF)	0,75	0,55	0,63	0,55	0,62	0,58	0,57	0,66	0,61	0,61	0,61	0,79
SVM (CV)	0,65	0,68	0,67	0,60	0,58	0,59	0,63	0,62	0,62	0,62	0,63	0,79
RF (TF)	0,72	0,56	0,63	0,58	0,62	0,60	0,53	0,62	0,57	0,60	0,60	0,81
RF (CV)	0,73	0,55	0,63	0,63	0,62	0,62	0,54	0,68	0,61	0,62	0,62	0,82
LR (TF)	0,72	0,56	0,63	0,58	0,66	0,62	0,58	0,63	0,61	0,62	0,62	0,81
LR (CV)	0,70	0,68	0,69	0,66	0,63	0,64	0,58	0,62	0,60	0,64	0,64	0,81
AB (TF)	0,77	0,74	0,76	0,65	0,62	0,63	0,55	0,60	0,58	0,65	0,65	0,82
AB (CV)	0,71	0,75	0,73	0,69	0,66	0,67	0,61	0,60	0,61	0,67	0,67	0,81
KNN (TF)	0,48	0,73	0,58	0,49	0,32	0,38	0,44	0,37	0,40	0,45	0,47	0,68
KNN (CV)	0,50	0,68	0,57	0,60	0,29	0,39	0,47	0,53	0,50	0,49	0,50	0,65
SetFit	0,89	0,75	0,81	0,86	0,90	0,88	0,81	0,89	0,85	0,85	0,85	0,94
BERT	0,83	0,68	0,75	0,87	0,89	0,88	0,75	0,86	0,80	0,81	0,81	0,93

Tabelle 16: Evaluation des Multilabel-Klassifikators an einem ausgeglichenem Evaluationsdatensatz

7.1.4 Einordnung

Durch einen Vergleich der Evaluationsergebnisse zeigt sich der Trend, dass die Deep-Learning Ansätze SetFit und BERT die traditionellen Machine Learning Modelle in allen relevanten Metriken Übertreffen. Lediglich die Wörterbuchansätze fallen durch hohe F1 und Recall Ergebnisse bei den Modellen auf, die auf einen hohen F1 und Recall optimiert wurden. Das Wörterbuchmodell, welches auf eine hohe Precision hin trainiert wurde erreicht dabei die höchsten Precision im Vergleich zu den anderen Modellen (über 95%). Jedoch wird auch nur ein Recall von unter 20% erreicht.

Die binären Klassifikatoren übertreffen dabei die Multiklassen-Klassifikatoren und performen alle ähnlich gut.

7.2 Validierung auf Basis eines unausgeglichenen Datensatzes

Um die Leistungsfähigkeit der Modelle in einem realistischen Szenario zu beurteilen, werden die Modelle an einem unausgeglichenen Datensatz getestet. Dafür wird davon ausgegangen, dass 5% der Reviews auf einen expliziten und 5% der Reviews auf einen impliziten Erklärungsbedarf hinweisen. Der in Abschnitt 7.1 vorgestellte Evaluationsdatensatz wird um jene Daten erweitert, die während des Undersamplings aus dem Trainingsdatensatz entfernt wurden. Dadurch enthält der Evaluationsdatensatz 5% Reviews mit dem entsprechenden Erklärungsbedarf. Da die Wörterbuchmethode den gesamten Trainingsdatensatz berücksichtigt, erfolgt ihre Evaluation an einem kleineren Datensatz

Datensatzart	Exp Reviews	Imp Reviews	No EN Reviews
Multiklassen	73	73	1440
Binär-EN	10	9	360
Binär-Exp/	99	72	1815
Binär-Imp	73	73	1387
Dict	13	12	227

Tabelle 17: Evaluationsdatensätze (unausgeglichen)

7.2.1 Erkennung von Reviews mit expliziten Erklärungsbedarf

Die Modelle, die darauf trainiert wurden, Reviews mit explizitem Erklärungsbedarf zu erkennen, weisen auf dem unausgeglichenen Datensatz eine geringere Performance hinsichtlich des F1-Scores auf (Tabelle 13). Dies ist auf die niedrigere Precision zurückzuführen, wobei SetFit den Spitzenwert von 34% erzielt. Obwohl der AUC-Score auf eine gute Trennungsfähigkeit der Modelle hinweist, sind sowohl der F1-Makro-Score als auch der Precision Score niedrig. Ansonsten gibt es keine besonderen Auffälligkeiten im Vergleich zur Evaluation an einem ausgeglichenem Datensatz

Model	Exp EN			No exp EN			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
NB (TF)	0,09	0,80	0,16	0,98	0,54	0,70	0,67	0,43
NB(CV)	0,10	0,77	0,18	0,98	0,62	0,76	0,69	0,47
SVM (TF)	0,14	0,73	0,24	0,98	0,75	0,85	0,74	0,54
SVM (CV)	0,18	0,76	0,29	0,98	0,81	0,89	0,78	0,59
RF (TF)	0,16	0,80	0,26	0,99	0,76	0,86	0,78	0,56
RF (CV)	0,15	0,84	0,25	0,99	0,73	0,84	0,78	0,55
LR (TF)	0,14	0,70	0,23	0,98	0,76	0,85	0,73	0,54
LR (CV)	0,18	0,73	0,29	0,98	0,81	0,89	0,77	0,59
AB (TF)	0,18	0,75	0,29	0,98	0,80	0,88	0,78	0,59
AB (CV)	0,18	0,76	0,29	0,98	0,81	0,89	0,78	0,59
KNN (TF)	0,10	0,57	0,17	0,97	0,71	0,82	0,64	0,49
KNN (CV)	0,11	0,54	0,19	0,97	0,77	0,85	0,65	0,52
SetFit	0,34	0,92	0,50	0,99	0,90	0,95	0,92	0,72
BERT	0,30	0,90	0,46	0,99	0,88	0,94	0,94	0,70

Tabelle 18: Evaluationsergebnisse der binären Klassifizierer, die expliziten Erklärungsbedarf erkennen (an einem unausgeglichenem Datensatz)

7.2.2 Erkennung von Reviews mit impliziten Erklärungsbedarf

Die Beobachtungen die bei der Untersuchung des binären Klassifizierers, der auf expliziten Erklärungsbedarf trainiert wurde, lassen sich auch auf die Auswertung des impliziten Klassifizier übertragen. Die höchste Precision erreicht dabei SetFit mit 28% (Tabelle 19).

Model	Implicit EN			No implicit EN			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
NB (TF)	0,06	0,70	0,11	0,97	0,43	0,60	0,57	0,35
NB(CV)	0,06	0,75	0,11	0,97	0,43	0,60	0,59	0,36
SVM (TF)	0,07	0,64	0,13	0,97	0,59	0,73	0,62	0,43
SVM (CV)	0,09	0,64	0,16	0,97	0,68	0,80	0,66	0,48
RF (TF)	0,08	0,75	0,14	0,98	0,55	0,70	0,65	0,42
RF (CV)	0,08	0,81	0,14	0,98	0,53	0,69	0,67	0,41
LR (TF)	0,08	0,67	0,14	0,97	0,59	0,73	0,63	0,44
LR (CV)	0,09	0,62	0,15	0,97	0,67	0,79	0,64	0,47
AB (TF)	0,09	0,70	0,16	0,98	0,63	0,77	0,67	0,46
AB (CV)	0,10	0,68	0,17	0,98	0,69	0,81	0,69	0,49
KNN (TF)	0,06	0,45	0,11	0,96	0,65	0,77	0,55	0,44
KNN (CV)	0,06	0,52	0,11	0,96	0,59	0,73	0,55	0,42
SetFit	0,28	0,92	0,42	1,00	0,88	0,93	0,94	0,68
BERT	0,25	0,92	0,40	1,00	0,86	0,92	0,93	0,66

Tabelle 19: Evaluationsergebnisse der binären Klassifizierer, die impliziten Erklärungsbedarf erkennen (an einem unausgeglichenem Datensatz)

7.2.3 Erkennung von Reviews mit allgemeinen Erklärungsbedarf

Bei dem binären Klassifizierer, der den allgemeinen Erklärungsbedarf identifiziert, ist ein ebenfalls ein Rückgang der Precision zu beobachten (Tabelle 20). Die beste Precision wird von SetFit mit 41% erzielt. Trotz hoher Recall-Werte (88% bei BERT und 93% bei SetFit) beeinträchtigt der niedrige Precision-Wert von 41% die Gesamtleistung der Klassifikation für den Erklärungsbedarf. Die besten AUC-Werte werden von SetFit und BERT erreicht.

Model	Explanation Need			No Explanation Need			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
NB (TF)	0,20	0,85	0,32	0,97	0,61	0,75	0,73	0,54
NB (CV)	0,18	0,75	0,29	0,96	0,63	0,76	0,69	0,53
SVM (TF)	0,27	0,85	0,41	0,98	0,75	0,85	0,80	0,63
SVM (CV)	0,30	0,70	0,42	0,96	0,82	0,88	0,76	0,65
RF (TF)	0,23	0,88	0,37	0,98	0,68	0,80	0,78	0,58
RF (CV)	0,20	0,78	0,31	0,96	0,65	0,77	0,71	0,54
LR (TF)	0,28	0,85	0,42	0,98	0,76	0,85	0,80	0,64
LR (CV)	0,29	0,72	0,41	0,96	0,80	0,88	0,76	0,65
AB (TF)	0,31	0,72	0,43	0,96	0,82	0,89	0,77	0,66
AB (CV)	0,32	0,65	0,43	0,96	0,84	0,90	0,75	0,66
KNN (TF)	0,16	0,23	0,19	0,91	0,87	0,89	0,55	0,54
KNN (CV)	0,23	0,62	0,34	0,95	0,77	0,85	0,70	0,59
SetFit	0,41	0,93	0,57	0,99	0,85	0,92	0,92	0,74
BERT	0,41	0,88	0,56	0,98	0,86	0,92	0,93	0,74

Tabelle 20: Evaluationsergebnisse der binären Klassifizierer, die Erklärungsbedarf erkennen (an einem unausgeglichenem Datensatz)

7.2.4 Wörterbuchmethode

Das Modell, das speziell darauf optimiert wurde, expliziten Erklärungsbedarf mit hoher Präzision zu identifizieren, erreicht auch im unausgeglichenen Evaluationsdatensatz eine Präzision von 80% (Tabelle 21). Im Gegensatz dazu erzielt das Modell, das auf hohe Präzision bei der Erkennung von implizitem Erklärungsbedarf trainiert wurde, keine Treffer. Die auf F1 optimierten Modelle erzielen dabei einen höheren Precisionscore und generell bessere Ergebnisse, sowohl auf das Modell bezogen, welche allgemeinen, als auch impliziten Erklärungsbedarf erkennt.

Model	(EN/IMP/EXP) Explanation Need			No (EN/IMP/EXP) Explanation need			AUC	F1- Makro
	Prec	Rec	F1	Prec	Rec	F1		
F1_EN	0,42	0,88	0,57	0,99	0,87	0,92	0,94	0,75
Rec_EN	0,24	0,88	0,38	0,98	0,69	0,81	0,79	0,59
Prec_EN	0,21	0,96	0,34	0,99	0,59	0,74	0,78	0,54
F1_imp	0,20	0,92	0,32	0,99	0,81	0,89	0,86	0,61
Rec_imp)	0,10	0,92	0,18	0,99	0,60	0,74	0,76	0,46
Prec_imp	0,00	0,00	0,00	0,95	1,00	0,97	0,50	0,49
F1_exp	0,27	1,00	0,42	1,00	0,85	0,92	0,92	0,67
Rec_exp	0,11	1,00	0,20	1,00	0,57	0,73	0,78	0,46
Prec_exp	0,80	0,31	0,44	0,96	1,00	0,98	0,65	0,71

Tabelle 21: Evaluierungsergebnisse anhand eines unbanzierten Datensatzes. Explanation Need bezieht sich dabei auf den Fokus des Modells

7.2.5 Multiklassen-Klassifizierung

Die Auswertung des Multiklassen-Klassifikator zeigt ebenfalls Einbußen in Hinblick auf die Precision bei der Erkennung von impliziten und expliziten Erklärungsbedarf. Setfit erzielt eine Precision von 32% für expliziten und 18% für impliziten Erklärungsbedarf, bei nur leicht schlechteren Recall-Werten. Die Ergebnisse sind in Tabelle 22 abgebildet.

	No en			Exp en			Imp EN			F1- macro	F1- micro	AUC
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1			
NB (TF)	0,99	0,46	0,63	0,05	0,52	0,10	0,05	0,57	0,09	0,27	0,47	0,69
NB(CV)	0,99	0,43	0,60	0,05	0,55	0,10	0,05	0,73	0,10	0,26	0,44	0,70
SVM (TF)	0,99	0,60	0,75	0,07	0,52	0,12	0,07	0,61	0,12	0,33	0,60	0,76
SVM (CV)	0,99	0,69	0,81	0,09	0,52	0,16	0,07	0,59	0,13	0,37	0,68	0,77
RF (TF)	0,99	0,60	0,74	0,09	0,59	0,15	0,05	0,52	0,09	0,33	0,59	0,79
RF (CV)	0,99	0,57	0,72	0,09	0,57	0,15	0,05	0,61	0,10	0,32	0,57	0,80
LR (TF)	0,99	0,63	0,77	0,08	0,57	0,14	0,07	0,59	0,12	0,34	0,62	0,79
LR (CV)	0,99	0,68	0,81	0,10	0,57	0,16	0,08	0,59	0,13	0,37	0,68	0,76
AB (TF)	0,99	0,66	0,79	0,10	0,52	0,16	0,07	0,64	0,13	0,36	0,66	0,80
AB (CV)	0,99	0,67	0,80	0,10	0,55	0,18	0,06	0,55	0,11	0,36	0,67	0,78
KNN (TF)	0,97	0,69	0,81	0,06	0,34	0,11	0,05	0,34	0,08	0,33	0,68	0,68
KNN (CV)	0,97	0,68	0,80	0,08	0,27	0,12	0,05	0,52	0,10	0,34	0,66	0,66
SetFit	0,99	0,73	0,84	0,32	0,90	0,48	0,20	0,89	0,32	0,55	0,74	0,91
BERT	0,99	0,71	0,83	0,32	0,89	0,47	0,18	0,86	0,30	0,53	0,73	0,93

Tabelle 22: Evaluation des Multilabel-Klassifikators an einem unausgeglichenem Datensatz

7.2.6 Einordnung

7.2.6.1 Bewertung der Machine- und Deep-Learning Ansätze

In der Analyse zeigte sich, dass binäre Klassifikatoren im Vergleich zu Multiklassen-Klassifikatoren überlegen sind (Tabelle 23). Besonders hervorzuheben sind jene Klassifikatoren, die auf die Erkennung von allgemeinem Erklärungsbedarf trainiert wurden. Im Gegensatz dazu weist der Multiklassen-Klassifikator insbesondere bei der Erkennung von Reviews mit implizitem Erklärungsbedarf Schwächen auf.

Zur Beantwortung der Forschungsfrage **F3** wurden Recall und Precision der Modelle detailliert analysiert. Es hat sich herausgestellt, dass Modelle, die spezifisch für die binäre Erkennung von explizitem, implizitem oder allgemeinem Erklärungsbedarf trainiert wurden, die bessere Leistung in Hinblick auf die relevanten Kategorien erzielt. Dies lässt sich sowohl durch die Auswertung des ausgeglichenem als beim unausgeglichenem Datensatz erkennen.

Kind	(EN/IMP/EXP) Explanation Need			No (exp/imp) EN			AUC	F1- Makro	F1- Mikro
	Pre	Rec	F1	Prec	Rec	F1			
SetFit (en)	0,41	0,93	0,57	0,99	0,85	0,92	0,92	0,74	0,86
SetFit (exp)	0,34	0,92	0,50	0,99	0,90	0,95	0,92	0,72	0,90
SetFit (imp)	0,28	0,92	0,42	1,00	0,88	0,93	0,94	0,68	0,88
SetFit(Multi-exp)	0,32	0,90	0,48	0,99	0,73	0,84	0,91	0,55	0,74
SetFit(Multi-imp)	0,20	0,89	0,32	0,99	0,73	0,84	0,91	0,55	0,74

Tabelle 23: Die Stärksten Modelle jeder Disziplin. In Hinblick auf den unausgeglichenem Datensatz

7.2.6.2 Wörterbuchmethode gegenüber Deep-Learning Ansätze

Da die Wörterbuchmethode mit der Gesamtanzahl der 4050 Trainingsdaten trainiert wurde, werden die erzielten Ergebnisse der Wörterbuchmethode mit den SetFit Ergebnissen verglichen, da es sich dabei um das Modell mit der besten Gesamtperformance handelt. Dazu wird untersucht wie gut SetFit auf dem unausgeglichenen Datensatz performt der zur Evaluation der Wörterbuchmethode genutzt wurde.

Erkennung von allgemeinem Erklärungsbedarf

Bei der allgemeinen Erkennung von Erklärungsbedarf zeigt sich, dass SetFit insgesamt (gemessen an der F1-Metrik) insgesamt besser abschneidet (Tabelle 24). Lediglich das Wörterbuchmodell, welches auf eine hohe Precision optimiert wurde, kann in der Disziplin 80% erreichen, wohingegen SetFit nur 42% der Reviews mit Erklärungsbedarf präzise erkennt. Abhängig vom Anwendungsfall kann jedoch das auf Precision trainierte Modell Vorteile bieten. Wenn ein hoher Precision-Wert gewünscht ist, könnte auch der Einsatz der Wörterbuchmethode (Dict) vorteilhaft sein.

Model	Explanation Need			No Explanation Need			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
SetFit (en)	0,42	0,88	0,57	0,99	0,87	0,92	0,94	0,75
Dict_EN F1	0,24	0,88	0,38	0,98	0,69	0,81	0,79	0,59
Dict_EN Rec	0,21	0,96	0,34	0,99	0,59	0,74	0,78	0,54
Dict_EN_Prec	0,80	0,16	0,27	0,91	1,00	0,95	0,58	0,61

Tabelle 24: Wörterbuchmethode im Vergleich zu SetFit (Erkennung von allgemeinem Erklärungsbedarf)

Erkennung von explizitem Erklärungsbedarf

Dasselbe lässt sich auf die auch auf die Methode zur Identifizierung von expliziten Erklärungsbedarf übertragbar. Die Wörterbuchmethode kann eine hohe Präzision erreichen, was jedoch einen Recall Wert ermöglicht.

Model	Exp En			No Exp EN			AUC	F1-Makro
	Prec	Rec	F1	Prec	Rec	F1		
SetFit (exp)	0,39	1,00	0,56	1,00	0,92	0,96	0,97	0,76
Dict_Exp_F1	0,27	1,00	0,42	1,00	0,85	0,92	0,92	0,67
Dict_Exp_Rec	0,11	1,00	0,20	1,00	0,57	0,73	0,78	0,46
Dict_Exp_Prec	0,80	0,31	0,44	0,96	1,00	0,98	0,65	0,71

Tabelle 25: Wörterbuchmethode im Vergleich zu SetFit (Erkennung von expliziten Erklärungsbedarf)

Erkennung von impliziten Erklärungsbedarf

Beim Vergleich in Bezug auf die Erkennung von Reviews mit impliziten Erklärungsbedarf werden die Schwächen der Wörterbuchmethode deutlich (Tabelle 26). Das Wörterbuchmodell welches auf eine hohe Precision hin optimiert wurde, erzielt hier gar keine Treffer. Da es sich um einen kleinen Datensatz handelt, kann hieraus keine allgemeingültige Aussage getroffen werden, macht aber die Problematik klar.

Model	Implicit EN			No Implicit EN				
	Prec	Rec	F1	Model	Prec	Rec	F1	Model
SetFit (imp)	0,20	0,91	0,33	0,99	0,82	0,90	0,93	0,62
Dict_Imp_F1	0,20	0,92	0,32	0,99	0,81	0,89	0,86	0,61
Dict_Imp_Rec	0,10	0,92	0,18	0,99	0,60	0,74	0,76	0,46
Dict_Imp_Prec	0,00	0,00	0,00	0,95	1,00	0,97	0,50	0,49

Tabelle 26: Wörterbuchmethode im Vergleich zu SetFit (Erkennung von impliziten Erklärungsbedarf)

Einordnung

In Bezug auf die Beantwortung der Forschungsfrage **F4**, wie gut sich eine regelbasierte Wörterbuchmethode gegenüber traditionelle Machine-Learning und Deep-Learning Methoden bei der Identifizierung von Reviews mit Erklärungsbedarf bewährt, lässt sich festhalten, dass der Ansatz der Wörterbuchmethode in Hinblick auf die F1-Performance generell schlechter abschneidet als die Deep-Learning Ansätze. Auswertungen konnten jedoch auch zeigen, dass die Methodik dazu geeignet sein kann Erklärungsbedarf mit hoher Precision zu filtern. Aufgrund der stärkeren Filterkriterien, können die Ergebnisse dieser Filterung je nach Datensatz nur einen geringen Recall aufweisen, wodurch nicht sichergestellt werden kann, dass eine Breite an Reviews mit Erklärungsbedarf gefunden werden kann.

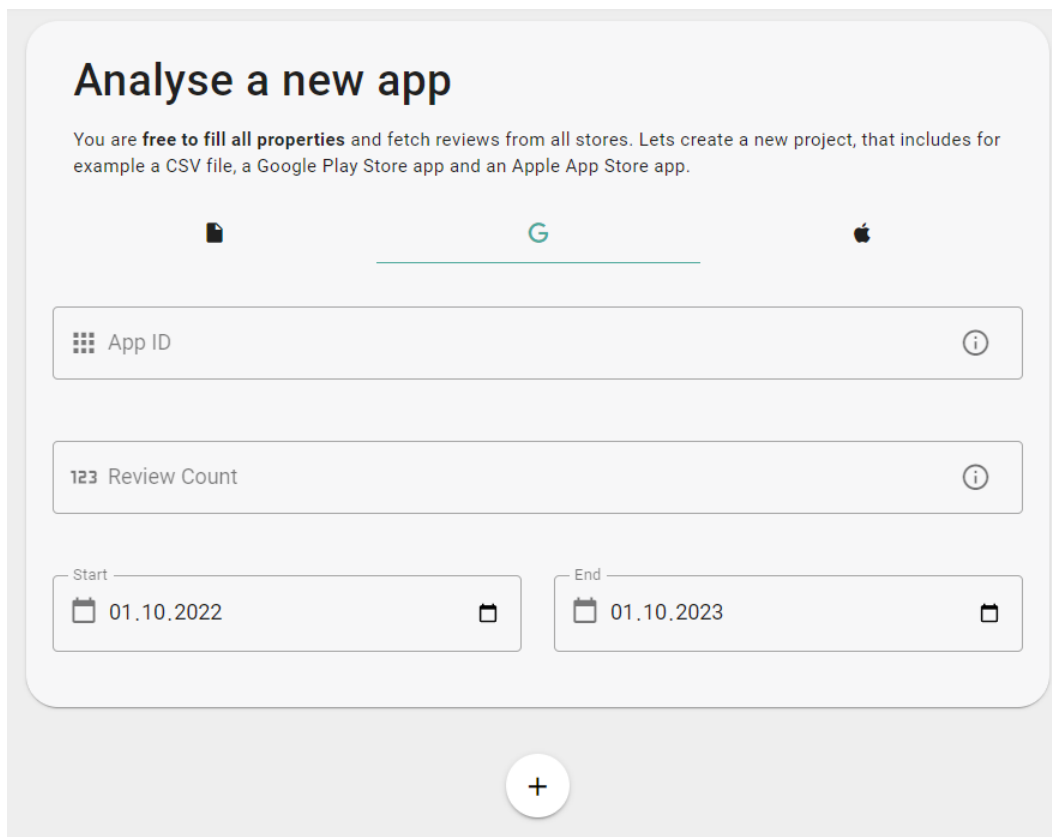
7.3 Fazit der Evaluation

Bei allen untersuchten Modellen zeigt sich ein signifikanter Rückgang des Precision-Werts, wenn man von einem ausgeglichenen zu einem praxisnah verteilten Datensatz wechselt. Obwohl der Recall-Wert bei den Top-Modellen auch bei der Evaluation mit einem unausgeglichenen Datensatz weitgehend stabil bleibt, fällt der Precision-Wert auf Werte zwischen 20% und 40%. Unsere Analyse hat zudem ergeben, dass binäre Klassifikatoren gegenüber Multiklassen-Klassifikatoren Vorteile aufweisen.

Die Ergebnisse ermöglichen es, die vierte Forschungsfrage zu beantworten, welche die Effektivität der Wörterbuchmethode im Vergleich zu Deep Learning-Ansätzen untersucht. Insbesondere bei der Erkennung von Reviews mit explizitem Erklärungsbedarf zeichnet sich die Wörterbuchmethode, die auf hohe Precision optimiert wurde, durch gute Ergebnisse aus. Im Vergleich dazu weisen Deep Learning-Modelle jedoch bessere Werte beim Recall, AUC und F1-Makro-Score auf. Dies deutet darauf hin, dass die Deep Learning-Modelle effektiver darin sind, zwischen den Klassen zu unterscheiden und mehr Fälle von Erklärungsbedarf zu identifizieren.

8. GUI Einbindung

Für die praktische Anwendung werden die Modelle, die in Bezug auf die F1-Metrik die besten Leistungen zeigen, in das Tool von Timo Kurz [32] integriert. In diesem Tool ist es bereits möglich, Reviews aus dem Play Store und App Store herunterzuladen und sie anhand von Kriterien, wie beispielsweise einer Sentiment-Analyse, zu filtern. Der Nutzer kann dabei neben der App-ID (aus dem Play Store oder App Store) auch die gewünschte Anzahl an Reviews sowie einen bestimmten Zeitraum angeben (Abbildung 10).



The image shows a search mask titled "Analyse a new app". Below the title is a sub-header: "You are **free to fill all properties** and fetch reviews from all stores. Lets create a new project, that includes for example a CSV file, a Google Play Store app and an Apple App Store app." There are three platform icons: a file icon, a "G" for Google, and an Apple logo. The "G" icon is selected and underlined. Below the icons are three input fields: "App ID" with a grid icon and an information icon; "123 Review Count" with an information icon; and two date pickers labeled "Start" and "End" with calendar icons, showing dates "01.10.2022" and "01.10.2023" respectively. At the bottom center is a large white circular button with a plus sign.

Abbildung 10: Suchmaske der GUI

Die Modelle, die in das Tool integriert werden, sind SetFit und die Wörterbuchmethode. Für die Einbindung von SetFit wurde das zuvor trainierte Modell auf Hugging Face hochgeladen, was eine effiziente Integration in das Tool ermöglicht. Die Wörterbuchmethode ist direkt in das Tool integriert. Sobald die Analyse einer App gestartet wird, erfolgt neben der Klassifikation der Sentimente auch eine Untersuchung der Reviews hinsichtlich des Erklärungsbedarfs.

Dabei werden zum einen durch die Wörterbuchmethode, die auf eine hohe Precision hin optimiert wurde, untersucht, welche Reviews wahrscheinlich Erklärungsbedarf aufweisen. Außerdem kann der Nutzer die Reviews nach Bewertungen filtern, die vom SetFit Modell als Review mit Erklärungsbedarf erkannt wurden.

Dadurch steht es dem Nutzer frei, ob für ihn bei der Analyse eine hohe Precision oder ein hoher Recall mit eingeschränkter Precision relevant ist.

Die ausgewählten Beispielreviews werden in einer Übersicht dargestellt (siehe Abbildung 12).

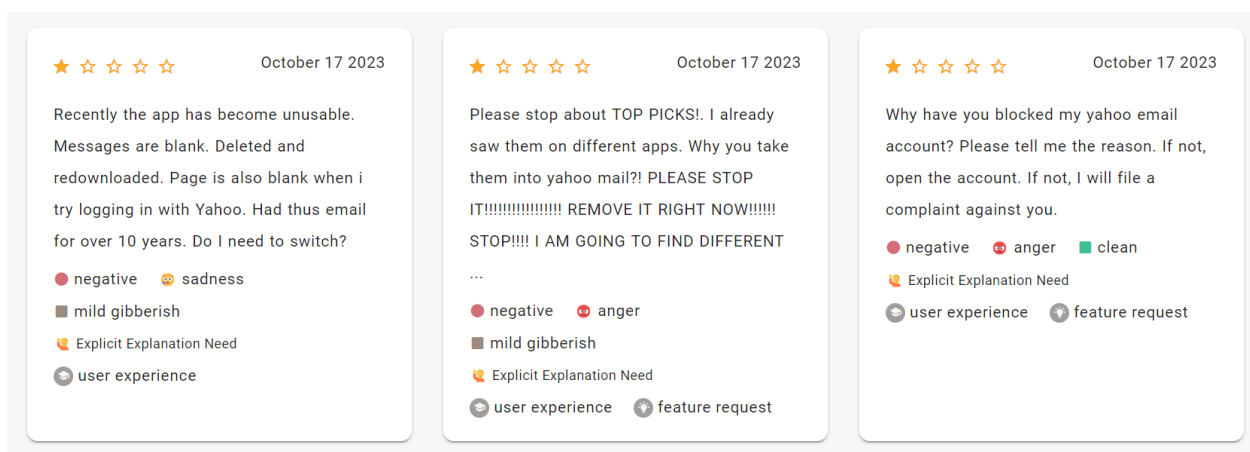


Abbildung 11: GUI Interface bei der Betrachtung der Ergebnisse

9. Threads of Validity

9.1 Interne Bedrohungen der Validität

9.1.1 Datensatzfilterung:

Die Vorfilterung der Bewertungen erfolgte anhand bestimmter Wörter und Phrasen. Daher reflektiert der erstellte Datensatz nicht notwendigerweise die tatsächliche Verteilung von Bewertungen mit Erklärungsbedarf. Obwohl der Filter auf Basis typischer Phrasen und Schlüsselwörter für expliziten und impliziten Erklärungsbedarf erstellt wurde, ist es möglich, dass es weitere Phrasen und Schlüsselwörter gibt, die auf Erklärungsbedarf hinweisen. Es wurde versucht, Phrasen mit hoher Präzisionsrate und hohem Recall-Wert zu kombinieren, um eine breite Palette von Mustern erkennen zu können. Eine genaue Filtermethode für Bewertungen mit implizitem Erklärungsbedarf konnte jedoch nicht angewendet werden.

9.1.2 Labelung und Agreement:

Trotz Diskussionen über die Richtlinien, Kategorieunterscheidungen und Phrasenmarkierungen vor, während und nach der Labelung/Codierung der Daten können Fehler und Missverständnisse nicht ausgeschlossen werden. Um Missverständnisse bezüglich der Taxonomiegruppen und der Arten von Erklärungsbedarf zu minimieren, wurden die Definitionen ausführlich mit Beispielen erläutert. Für die Erstellung eines Goldstandard-Datensatzes sollte die Labelung jedoch vollständig unabhängig voneinander erfolgen. Aufgrund von Unklarheiten gab es dennoch Diskussionen während des Prozesses

9.1.3 Kategorie Verteilung

Die Vorfilterung beeinflusste auch die Anzahl der Bewertungen in den einzelnen Kategorien. So lassen sich beispielsweise Bedienungsprobleme leichter bestimmten Phrasen zuordnen als Fragen zum Algorithmus. Dies hat zur Folge, dass einige Kategorien, wie „Operation“ oder „Unerwartetes Systemverhalten“ präsenter vertreten sind als andere. So kann der Datensatz nur wenig Einblick in den Bereich „Security“ in Bezug auf Erklärungsbedarf geben.

9.2 Externe Bedrohungen der Validität

9.2.1 Grundlage des Datensatzes:

Der Datensatz basiert auf 90.000 Bewertungen aus dem App Store und Play Store, die die Top-Apps der beliebtesten Kategorien repräsentieren. Es kann jedoch nicht garantiert werden, dass diese 60 Apps alle möglichen Arten von Erklärungsbedarf abdecken oder ob die Ergebnisse ausschließlich auf den Google Play Store und App Store beschränkt sind.

9.2.2 Anwendbarkeit auf weitere Forschungsarbeiten

Die Einordnung in impliziten und expliziten Erklärungsbedarf baut auf die Definition des Software Engineering Instituts der Leibniz Universität Hannover auf. Da es unterschiedliche Definitionen von Erklärungsbedarf geben kann, ist es für die Nutzung der Modelle des Datensatzes von Bedeutung, das Verständnis von Erklärungsbedarf abzugleichen.

8.2.3 Einordnung der Evaluationsergebnisse

Die Evaluation wurde ebenfalls auf einem für die Arbeit zusammengestellten Datensatz durchgeführt. Da der Datensatz auf der Grundlage von Vorfilterungen aufgebaut wurde, kann nicht sichergestellt werden, dass sowohl die Verteilung der Kategorien im Trainingsdatensatz, als auch die Evaluationsergebnisse auf die Praxis übertragbar sind.

10. Fazit

10.1 Diskussion

Auf Basis der beantworteten Forschungsfragen, wird das Vorgehen nochmal evaluiert und eingeordnet.

F1: Wie präzise lässt sich ein Datensatz aus Reviews mit impliziten und expliziten Erklärungsbedarf, durch eine gezielte Nutzung von regelbasiertem Filter zusammenstellen?

In Kapitel 4.4.3 wurde gezeigt, dass durch den Einsatz von Filtern können Reviews mit explizitem Erklärungsbedarf mit einer Precision von 79,8% und Reviews mit implizitem Erklärungsbedarf mit einer Precision von 40,8% identifiziert werden. Wenn nicht zwischen implizitem und explizitem Erklärungsbedarf unterschieden wird, erreicht der Filter eine Precision von 68,2%. Der Hauptgrund für diese Unterschiede ist die Kontextabhängigkeit von implizitem Erklärungsbedarf. Durch die Fokussierung auf Schlüsselwörter kann nicht immer garantiert werden, dass es sich tatsächlich um impliziten Erklärungsbedarf handelt. Reviews mit explizitem Erklärungsbedarf können aufgrund ihrer typischen Phrasenstruktur präziser gefiltert werden. Dennoch zeigt der Wert von 79,8%, dass es Fälle gibt, in denen eine Review fälschlicherweise als explizit gekennzeichnet wurde.

F2: Kann durch eine Filterbasierte Datensatzerstellung gewährleistet werden, dass alle möglichen Bereiche abdeckt werden, in denen Erklärungsbedarf vorkommen kann?

Eine gleichmäßige Verteilung über alle Kategorien kann durch die angewandte Methodik nicht sichergestellt werden. Die in Kapitel 5.1 vorgestellten Daten zeigen eine Überrepräsentation der Aspekte "Unerwartetes Systemverhalten", "Operation" und "Metainformationen". Sicherheitsrelevante Themen sowie die Kategorien "Konsequenzen", "Einführung" und "Algorithmen" sind hingegen unterrepräsentiert. Dies hängt mit der Herausforderung zusammen, gezielte Filter zu erstellen, die zu diesen weniger vertretenen Kategorien führen. Mit der aktuellen Methode wurden hauptsächlich Themen herausgefiltert, die das Systemverhalten, die Interaktion mit dem System oder Kombinationen mehrerer Kategorien betreffen.

F3: Wie können trainierte Modelle Reviews, Erklärungsbedarf, in Hinblick auf Precision und Recall erkennen?

Die Analysen aus Kapitel 7 zeigen, dass Modelle, die binär auf die Erkennung von explizitem, implizitem oder allgemeinem Erklärungsbedarf trainiert wurden, besser performen als Multiklassen-Klassifikatoren. Am besten performen die Modelle auf einem ausgeglichenem Datensatz, wo die Deep Learning Ansätze für die Erkennung von impliziten und expliziten Erklärungsbedarf Precision und Recall Werte von über 90% erreichen.

Auf einen unausgeglichenem Datensatz, lässt sich jedoch für alle Modelle ein wesentlich schlechter werdender Recall beobachten. Das Deep-Learning-Modell "SetFit" erzielte bei der Klassifikation von Erklärungsbedarf hierbei die besten Ergebnisse mit einem Recall-Wert von 93%, und einer Precision von 41%. Bei einer gezielten Klassifikation von Reviews mit impliziten oder expliziten Erklärungsbedarf sank die Precision auf 28% für die Erkennung von impliziten und 34% für die Erkennung von impliziten Erklärungsbedarf.

F4: Wie vergleicht sich eine filterbasierte Wörterbuchmethode gegenüber traditionelle Machine-Learning und Deep-Learning Methoden bei der Identifizierung von Reviews mit Erklärungsbedarf?

Insbesondere bei der Erkennung von Reviews mit explizitem Erklärungsbedarf zeichnet sich die Wörterbuchmethode, die auf hohe Precision optimiert wurde, durch ein hohes Precision-Ergebnis aus. Im Vergleich zur Deep-Learning-Methode weist dieser Ansatz jedoch geringere Recall-Werte auf. Dies bedeutet, dass je nach Anwendungsfall entweder das "SetFit"-Modell oder die Wörterbuchmethode bevorzugt werden sollte. Wenn ein hoher Recall-Wert gewünscht ist, wird das "SetFit"-Modell besser abschneiden. Wenn jedoch eine hohe Precision erforderlich ist, kann die Wörterbuchmethode präzise Beispiele aus einem großen Datensatz herausfiltern, die Erklärungsbedarf aufweisen. Die Deep-Learning Ansätze weisen generell eine stabilere Performance auf unterschiedliche Datensätze auf.

10.2 Zusammenfassung

Im Rahmen dieser Arbeit wurde durch die gezielte Wahl von Filterkriterien ein unangetasteter Datensatz, der sich aus Bewertungen von 60 unterschiedlichen Apps zusammensetzt, auf potenziellen impliziten und expliziten Erklärungsbedarf hin gefiltert. Die Ergebnisse dieser Filterung wurden auf ihre Richtigkeit hin von drei Codierern überprüft. Dabei wurde eine Precision von 79,8 % für die Erkennung von Reviews mit explizitem Erklärungsbedarf sowie eine Precision von 40,8 % für die Erkennung von Reviews mit implizitem Erklärungsbedarf festgestellt.

Reviews des Datensatzes wurden anschließend auf Kategorien hin überprüft, auf die sich der jeweilige Erklärungsbedarf bezieht, und auf Muster untersucht.

Während des Trainings der Modelle zeigte sich, dass binäre Klassifikatoren den -- Detektoren überlegen sind. Die höchsten Recall- und Precision-Werte erreichten dabei Modelle, die impliziten und expliziten Erklärungsbedarf gemeinsam betrachteten, gefolgt von Modellen, die speziell auf die Erkennung von explizitem Erklärungsbedarf trainiert wurden. Die Ergebnisse für die Erkennung von implizitem Erklärungsbedarf allein schneiden in einem praxisorientierten Datensatz, dessen Anteil an Erklärungsbedarf gering ist, eher schlecht ab.

Die entwickelte Wörterbuch-Methode schnitt generell schlechter als die Deep-Learning-Ansätze ab. Die Ausnahme bilden Wörterbuch-Modelle, die auf eine hohe Precision hin optimiert wurden. Die Metrik betreffend erreichten diese Ansätze stärkere Ergebnisse, erreichen aber dabei auch nur einen geringen Recall. Die hohe Precision kann für Anforderungsanalysten vorteilhaft sein, insbesondere wenn gezielt Reviews mit Erklärungsbedarf betrachtet werden sollen.

Beide Ansätze wurden in das Tool von Timo Kurtz [64] integriert, was es ermöglicht, auf die Stärken beider Ansätze zurückzugreifen.

10.3 Ausblick

Die Analyseergebnisse bieten wertvolle Einblicke in den impliziten und expliziten Erklärungsbedarf und verdeutlichen, inwiefern eine gezielte Phrasenfilterung für die Klassifikation nützlich sein kann. Zukünftige Studien könnten die tatsächliche Effizienz der Modelle anhand eines umfangreichen, ungefilterten Datensatzes prüfen, der ebenfalls zwischen implizitem und explizitem Erklärungsbedarf unterscheidet. Um den idealen praktischen Nutzen zu erzielen und das Verständnis von Erklärungsbedarf zu vertiefen, sollte das Ziel sein, kontinuierlich die Precision und den Recall zu verbessern und Modelle zu entwickeln, die nicht nur zwischen implizit und explizit unterscheiden, sondern auch die spezifische Kategorie des Erklärungsbedarfs bestimmen können. Neben der Identifikation von Erklärungsbedarf ist die Klärung von Unklarheiten von Bedeutung. In zukünftigen Forschungsarbeiten könnte untersucht werden, wie automatisierte Reaktionen auf geäußerten Erklärungsbedarf zur Klärung beitragen können.

Literaturverzeichnis:

- [1] Unterbusch, Max & Sadeghi, Mersedeh & Fischbach, Jannik & Obaidi, Martin & Vogelsang, Andreas. (2023). Explanation Needs in App Reviews: Taxonomy and Automated Detection.
- [2] Stanik, Christoph; Haering, Marlo; Maalej, Walid. "Classifying Multilingual User Feedback using Traditional Machine Learning and Deep Learning" (2019).
- [3] Maalej, Walid; Nayebi, Marouane; Johann, Timo; Ruhe, Guenther. "Toward Data-Driven Requirements Engineering" in IEEE Software, Vol. 33, No. 1, Jan.-Feb. 2016, pp. 48-54. doi: 10.1109/MS.2015.153.
- [4] Maalej, Walid; Kurtanovic, Zlatko; Nabil, Hoda; Stanik, Christoph. "On the Automatic Classification of App Reviews" in Software Engineering 2017, Bonn: Gesellschaft für Informatik e.V., 2017, pp. 61.
- [5] Dias Canedo, Edna; Cordeiro Mendes, Bruno. "Software Requirements Classification Using Machine Learning Algorithms" in Entropy, Vol. 22, No. 9, 2020, pp. 1057. doi: 10.3390/e22091057.
- [6] Restrepo, Pablo; Fischbach, Jannik; Spies, Dominik; Frattini, Julian; Vogelsang, Andreas. "Transfer Learning for Mining Feature Requests and Bug Reports from Tweets and App Store Reviews" (2021).
- [7] Maalej, Walid; Nayebi, Marouane; Ruhe, Guenther. "Data-Driven Requirements Engineering - An Update" in 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 2019, pp. 289-290. doi: 10.1109/ICSE-SEIP.2019.00041.

-
- [8] Lu, Mengmeng; Liang, Peng. "Automatic Classification of Non-Functional Requirements from Augmented App User Reviews" (2017). doi: 10.1145/3084226.3084241.
- [9] Scalabrino, Stefano; Bavota, Gabriele; Russo, Barbara; Penta, Massimiliano Di; Oliveto, Rocco. "Listening to the Crowd for the Release Planning of Mobile Apps" in *IEEE Transactions on Software Engineering*, Vol. 45, No. 1, Jan. 2019, pp. 68-86. doi: 10.1109/TSE.2017.2759112.
- [10] Pagano, Daniele; Maalej, Walid. "User feedback in the appstore: An empirical study" in *2013 21st IEEE International Requirements Engineering Conference (RE)*, Rio de Janeiro, Brazil, 2013, pp. 125-134. doi: 10.1109/RE.2013.6636712.
- [11] Maalej, Walid; Nabil, Hoda. "Bug report, feature request, or simply praise? On automatically classifying app reviews" in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, Ottawa, ON, Canada, 2015, pp. 116-125. doi: 10.1109/RE.2015.7320414.
- [12] Statista. (n.d.). App-Nutzung weltweit - Statistik & Prognose. Abgerufen am 12.05.2023 von <https://de.statista.com/outlook/dmo/app/weltweit>
- [13] Jurafsky, D., & Martin, J. H. (2019). *Speech and language processing* (3rd ed.).
- [14] [13] Indurkha, N., & Damerau, F. J. (2010). *Handbook of natural language processing* (2nd ed.). CRC Press.
- [15] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.
- [16] Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), 3-26.
- [17] Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, 28(9), 921-932.
- [18] Jurafsky, D. & Martin, J.H. (2021). *Speech and Language Processing* (S. 269f)

-
- [19] Machines: Learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning (ECML 1998) (S. 137-142).
- [20] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). Springer.
- [21] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.
- [22] Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- [23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
- [24] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding.
- [25] Mitchell, T. (1997). *Machine Learning*. McGraw Hill.
- [26] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org/>
- [27] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.
- [28] MaxQDA. (2023). Problem with intercoder agreement in qualitative research. <https://www.maxqda.com/help-mx20/teamwork/problem-intercoder-agreement-qualitative-research>. Zugriff am 01.10.2023.
- [29] Tunstall, Lewis & Reimers, Nils & Jo, Unso & Bates, Luke & Korat, Daniel & Wasserblat, Moshe & Pereg, Oren. (2022). Efficient Few-Shot Learning Without Prompts. 10.48550/arXiv.2209.11055.
- [30] D. Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. *Mach. Learn. Technol.*, 2, 01 2008.

-
- [31] Md Rakibul Islam, Minhaz F. Zibran, SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text, *Journal of Systems and Software*, Volume 145,
- [32] Kurtz, T. (2023). Entwicklung einer Software zur Extrahierung und Analyse von Reviews aus App Stores (Bachelorarbeit, Fachgebiet Software Engineering). Leibniz Universität Hannover.
- [33] Deters HL, Droste JRC, Schneider K. A Means to what End? Evaluating the Explainability of Software Systems using Goal-Oriented Heuristics. In *EASE '23: Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*. 2023. S. 329-338.
- [34] Ashwin Ittoo, Le Minh Nguyen, Antal van den Bosch, Text analytics in industry: Challenges, desiderata and trends, *Computers in Industry*, Volume 78, 2016, Pages 96-107,
- [35] Landis, J. Richard, and Gary G. Koch. "The Measurement of Observer Agreement for Categorical Data." *Biometrics*, vol. 33, no. 1, 1977, pp. 159–74. *JSTOR*, <https://doi.org/10.2307/2529310>. Accessed 22 Oct. 2023.
- [36] Schneider, K. (2021). SoftXplain: Anforderungen für selbst-erklärende Software. Deutsche Forschungsgemeinschaft (DFG) - Projektnummer 470146331. Abgerufen am [Datum des Zugriffs] von <https://gepris.dfg.de/gepris/projekt/470146331>.
- [37] I. Nunes and D. Jannach, "A systematic review and taxonomy of explanations in decision support and recommender systems," *User Modeling and User-Adapted Interaction*, vol. 27, no. 3, pp. 393–444, 2017.
- [38] Deters HL, Droste JRC, Fechner M, Schneider K. Explanations on Demand - a Technique for Eliciting the Actual Need for Explanations. In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*. 2023. S. 345 – 351

-
- [39] Droste JRC, Deters HL, Puglisi J, Schneider K. Designing End-user Personas for Explainability Requirements using Mixed Methods Research. In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*. 2023. S. 129-135
- [40] Statista. (2023). Beliebteste Kategorien im App Store. Abgerufen am 01.10.2023 von <https://de.statista.com/statistik/daten/studie/166976/umfrage/beliebteste-kategorien-im-app-store/>.
- [41] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138– 52 160, 2018.
- [42] W. Brunotte, L. Chazette, V. Klös, and T. Speith, "Quo vadis, explainability? – A research roadmap for explainability engineering," in *Requirements Engineering: Foundation for Software Quality*, 2022, pp. 26–32.
- [43] Chazette, L., & Schneider, K. (2020). Explainability as a non-functional requirement: challenges and recommendations. *Requirements Engineering*, 25(4), 493-514.
- [44]: L. Chazette, W. Brunotte, and T. Speith, "Exploring explainability: A definition, a model, and a knowledge catalogue," in *2021 IEEE 29th International*
- [45] A. Rosenfeld and A. Richardson, "Explainability in human–agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 33, pp. 673– 705, 2019.
- [46] W. Brunotte, A. Specht, L. Chazette, and K. Schneider, "Privacy explanations – a means to end-user trust," *Journal of Systems and Software*, vol. 195, p. 111545, 2023.
- [47] L. Chazette, J. Klunder, M. Balci, and K. Schneider, "How can we " develop explainable systems? insights from a literature review and an interview study," in *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*, 2022, pp. 1–12.

-
- [48] C. Combi, B. Amico, R. Bellazzi, A. Holzinger, J. H. Moore, M. Zitnik, and J. H. Holmes, "A manifesto on explainability for artificial intelligence in medicine," *Artificial Intelligence in Medicine*, vol. 133, p. 102423, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0933365722001750>
- [49] W. Brunotte, L. Chazette, L. Kohler, J. Klunder, and K. Schneider, "What about my privacy? helping users understand online privacy policies," in *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*, ser. ICSSP'22. New York, NY, USA: Association for Computing Machinery, 2022, p. 56–65.
- [50] L. Chazette and K. Schneider, "Explainability as a non-functional requirement: challenges and recommendations," *Requirements Engineering*, vol. 25, no. 4, pp. 493–514, 2020.
- [51] M. A. Kohl, K. Baum, M. Langer, D. Oster, T. Speith, and D. Bohlender, "Explainability as a non-functional requirement," in *IEEE 27th International Requirements Engineering Conference*, 2019, pp. 363–368.

Anhang

A1 Übersicht der Apps aus dem Appstore, die als Datengrundlage genutzt wurden dienen

Kategorie	Gratis Apple	Kostenpflichtig Apple
Spiele	Monopoly Go (1000), Royal Match	Minecraft, Heads up
Business	LinkedIn (1000), Teams (500)	TurboScan (1000), Hot Schedules (500)
Bildung	PictureThis (1000), Duolingo (500)	Sky View (1000), Anki (500)
Lifestyle	Pinterest (1000), Ring (500)	Cloud Baby (1000), Styleboo (500)
Hilfsmittel	Google (1000), Widgetable (500)	Adblock (1000), Shadowrocket (500)
Unterhaltung	Max (1000), Tiktok (500)	Pocket God (1000), Akinator VIP (500)
Essen/Trinken	Too Good to go (1000), McDonalds (500)	Paprica Recipe Manager (1000), FitMenCook (500)
Reisen	Booking.com (1000), Uber (500)	Happy Cow (1000), OBD Fusion (500)
Gesundheit	Impluse (1000), Planet Fitness (500)	Wonderweek (1000), Autosleep (500)
Produktivität	Gmail (1000), Microsoft Authenticator (500)	Forest (1000), Screen Mirroring (500)
Shopping	Temu (1000), Amazon (500)	Boycott (1000), Christmas List (500)
Finanzen	Cash App (1000), Paypal (500)	iAllowance (1000), Accounts 2 Checkbook (500)
Soziale Netzwerke	BeReal (1000), Threads (500)	Badoo premium (1000), Friends Plus social Browser (500)
Musik	Spotify (1000), Shazam (500)	Tonal Energy (1000), iReal Pro (500)
Bücher	Audible (1000), Kindle (500)	Js (1000), Twenty-Four Hours a Day (500)

A2: Übersicht Apps aus dem Play Store, die als Datengrundlage genutzt wurden dienen

Kategorie	Gratis Play Store	Kostenpflichtig Play Store
Spiele	Clash of Clans (1000), Roblox (500)	Minecraft, Bloons TD6 (500)
Business	Microsoft Teams (1000), LinkedIn (500)	OfficeSuite Pro (1000), MDScan + OCR(500)
Bildung	Duolingo (1000), Samsung Global Goals (500)	Star walk 2 (1000), Toca Hair Salon (500)
Lifestyle	Samsung Wallet (1000), Pinterest (500)	Alarm Clock (1000), Alarm Plus Milenium (500)
Hilfsmittel	Google (1000), Clock (500)	Tasker (1000), 1DM+ (500)
Unterhaltung	Netflix (1000), Hulu (500)	Meme Generator Pro (1000), Scanner Radio Pro (500)
Essen/Trinken	DoorDash (1000), Starbuckt (500)	CookBook (500), On She Glows (482), My Cocktail Bar Pro (157), 101 Juice Recipes (361)
Reisen	Google Maps (1000), AirBnB (500)	OsmAnd+ (1000), Peakfinder (500)
Gesundheit	Komoot (1000), Samsung Health (500)	The Wonder Weeks (1000), White Noise (500)
Produktivität	Google Docs (1000), Google Calender (500)	HotSchedules (1000), RealCalc Plus(500)
Shopping	Amazon Shopping (1000), Walmart (500)	Shopping list voice input Pro (1000), Movie Collection & Inventory(500)
Finanzen	Paypal (1000), Cash App (500)	My Budget Book (1000), Monefy (500)
Soziale Netzwerke	Tiktok (1000), Threads (500)	Reddit Donations (1000), Tapatalk (500)

Musik	Spotify (1000), YouTube Music (500)	JetAudio HD (1000), FL Studio (500)
Bücher	Audible (1000), Kindle (500)	Moon+ Reader Pro (1000), ReadEra Premium(500)

A3 Zusätzliche Evaluationen

Datensatz	Reviews mit expliziten Erklärungsbedarf	Reviews mit impliziten Erklärungsbedarf	Reviews ohne Erklärungsbedarf	Anteil der reviews mit Erklärungsbedarf
Detect exp (val_large)	145 (Label: 1)	74 (Label: 0)	1617 (Label: 0)	7,8%
Detect imp (val_large)	1125 (Label: 0)	73 (Label: 1)	1944 (Label: 0)	2,32%
Detect Explanation Need (val_large)	145(Label: 1)	73(Label: 1)	312 (Label: 0)	41,32%

T1: Validierungsdatensätze, bestehend aus Daten die weder für das Training, noch bis die ausgeglichene Evaluation der Modelle genutzt wurden

T2: EN Ergebnisse bezüglich A3

	Precision		Recall		F1		AUC	F1-Makro	F1-Mikro
	Exp	No Exp	Exp	No Exp	Exp	No Exp			
NB (TF)	0,68	0,72	0,83	0,61	0,57	0,86	0,72	0,70	0,70
NB (CV)	0,68	0,73	0,84	0,63	0,58	0,87	0,73	0,71	0,71
SVM (TF)	0,75	0,81	0,84	0,75	0,67	0,89	0,80	0,78	0,79
SVM (CV)	0,75	0,84	0,78	0,82	0,72	0,86	0,80	0,79	0,80
RF (TF)	0,72	0,77	0,86	0,68	0,62	0,89	0,77	0,75	0,75
RF (CV)	0,69	0,74	0,83	0,65	0,59	0,87	0,74	0,72	0,72
LR (TF)	0,76	0,82	0,85	0,76	0,68	0,90	0,81	0,79	0,79
LR (CV)	0,74	0,83	0,78	0,80	0,71	0,86	0,79	0,79	0,79
AB (TF)	0,74	0,84	0,77	0,82	0,72	0,85	0,79	0,79	0,80
AB (CV)	0,73	0,84	0,72	0,84	0,74	0,83	0,78	0,78	0,80
KNN (TF)	0,49	0,78	0,39	0,87	0,64	0,70	0,63	0,63	0,69
KNN (CV)	0,63	0,77	0,63	0,77	0,62	0,78	0,70	0,70	0,72
SetFit	0,86	0,90	0,94	0,85	0,79	0,96	0,92	0,88	0,89
BERT	0,84	0,90	0,90	0,86	0,80	0,93	0,93	0,87	0,88

	Precision		Recall		F1		AUC	F1-Makro	F1-Mikro
	Exp	No Exp	Exp	No Exp	Exp	No Exp			
NB (TF)	0,03	0,99	0,70	0,48	0,06	0,65	0,59	0,35	0,48
NB (CV)	0,03	0,99	0,75	0,46	0,06	0,62	0,60	0,34	0,46
SVM (TF)	0,04	0,99	0,64	0,65	0,08	0,78	0,64	0,43	0,65
SVM (CV)	0,05	0,99	0,64	0,72	0,09	0,83	0,68	0,46	0,72
RF (TF)	0,04	0,99	0,75	0,62	0,08	0,76	0,68	0,42	0,62
RF (CV)	0,05	0,99	0,81	0,60	0,09	0,75	0,70	0,42	0,61
LR (TF)	0,04	0,99	0,67	0,66	0,08	0,79	0,66	0,44	0,66
LR (CV)	0,05	0,99	0,62	0,71	0,09	0,83	0,67	0,46	0,71
AB (TF)	0,05	0,99	0,70	0,68	0,09	0,81	0,69	0,45	0,68
AB (CV)	0,05	0,99	0,68	0,71	0,10	0,82	0,70	0,46	0,71
KNN (TF)	0,03	0,98	0,45	0,69	0,06	0,81	0,57	0,44	0,69
KNN (CV)	0,03	0,98	0,52	0,66	0,07	0,79	0,59	0,43	0,66
SetFit	0,11	1,00	0,92	0,83	0,20	0,90	0,91	0,55	0,83
BERT	0,10	1,00	0,92	0,80	0,18	0,89	0,91	0,53	0,81

T3: Imp Ergebnisse bezüglich A3

	Precision		Recall		F1		AUC	F1-Makro	F1-Mikro
	Exp	No Exp	Exp	No Exp	Exp	No Exp			
NB (TF)	0,13	0,97	0,81	0,54	0,22	0,70	0,68	0,46	0,56
NB (CV)	0,15	0,97	0,77	0,62	0,24	0,76	0,70	0,50	0,63
SVM (TF)	0,20	0,97	0,74	0,75	0,31	0,85	0,74	0,58	0,75
SVM (CV)	0,24	0,97	0,71	0,81	0,35	0,88	0,76	0,62	0,80
RF (TF)	0,22	0,98	0,81	0,76	0,34	0,86	0,78	0,60	0,76
RF (CV)	0,20	0,98	0,83	0,73	0,33	0,84	0,78	0,58	0,74
LR (TF)	0,20	0,97	0,70	0,76	0,31	0,85	0,73	0,58	0,75
LR (CV)	0,24	0,97	0,71	0,81	0,35	0,88	0,76	0,62	0,80
AB (TF)	0,24	0,97	0,74	0,80	0,36	0,88	0,77	0,62	0,80
AB (CV)	0,24	0,97	0,73	0,81	0,36	0,88	0,77	0,62	0,80
KNN (TF)	0,13	0,95	0,54	0,71	0,22	0,81	0,63	0,51	0,69
KNN (CV)	0,16	0,95	0,52	0,77	0,24	0,85	0,64	0,54	0,75
SetFit	0,41	0,99	0,91	0,89	0,57	0,94	0,92	0,75	0,89
BERT	0,39	0,99	0,91	0,88	0,55	0,93	0,94	0,74	0,89

T4: Exp Ergebnisse bezüglich A3

A4 Eingeordnete Beispielergebnisse

Beispiele für Reviews mit Erklärungsbedarf aus dem Bereich Interaktion:

Operation:

Es gibt Unklarheiten bezüglich einer Systemoperation. Dies kann sich auf die Ausführung einer Operation oder ein Problem mit einer Operation zusammenhängen:

- *“How do I subscribe and get an IP address“*
- *“[.]donâ€™t understand why we canâ€™t search[.]”*

Einführung:

Falls spezielle Schritten zur Zielführung angesprochen werden

- *“[.]Very hard to understand how to set up[.]“*
- *“But where is instruction for setup? “*

Navigation:

Es wird gezielt nach einer Bedienungshilfe während der Navigation gesprochen:

- *“where is it? How is it accessed?“*
- *“[.]It very unintuitive to find[.]”*

Bereich Systemverhalten:

Unerwartetes Verhalten:

Hierrein fallen Situationen in denen der User beschreibt, dass das System nicht auf die Art und weise reagiert, wie es erwartet wird. Es grenzt sich jedoch von dem Bereich der Interaktion ab und beschreibt Fälle, in denen der User beschreibt, dass das System Probleme macht.

- *“but I just randomly lost all of my information? That's super irritating“*

- "Why can't I sign in?"
- "[...]tell me why it does not sort the list of non-working and working countries"

Algorithmus:

Falls gezielt nach den Entscheidungsgrundlage der Systementscheidungen gefragt wird:

- *"I never ask for them to pick things for me I picked my own playlist and now they're choosing it for me?"*

Konsequenzen:

Es besteht Unklarheit darüber, inwiefern bestimmte Aspekte einfluß auf andere Bereiche haben.

- *"[...]I also find it irritating that if I change the speed on a book it changes that speed for all books"*
- *"When I enter the game, why is it consuming the network????"*

Bugs:

Falls gezielt ein Absturz oder ein Fehler benannt wird.

- *"I don't understand that. I tried it five times gives the same error."*
- *"Is this a bug?"*

Reviews sind mit **Meta** gelabelt, sofern sich der Erklärungsbedarf nicht einer eindeutigen Kategorie zugeordnet werden können (z.B. Interaktion und Systemverhalten hinterfragt werden). Auch Erklärungsbedarf bezüglich Aspekte die sich auf Aspekte um das System herum beziehen werden hier berücksichtigt (sofern es einen technischen Hintergrund hat).

Dazu gehören Reviews wie:

- “[..]why is it so hard to prevent people from hacking servers?[..]”
- “Why didnâ€™t they just update the old app”
- “doesnâ€™t block ads in apps:Why doesnâ€™t this block the ads in my apps?! So frustrating please fix this!!!! HOW do I block the ads in other apps?!?”

Systemspezifische Aspekte:

Domain:

Dem user ist unklar, was etwas bedeutet, was nicht speziell mit der App oder dem Anbieter zutun hat.

- “Don’t understand what SSN means“
- “Pricing is not in USD. What is COP?”

Beispiel indirekte Systemaspekte:

Es besteht unklarheit über einen Systemaspekt, welches eher das Konzept der App betrifft:

- “what’s the difference between pro and premium?”
- “[..] what does the shield do [..]”

Beispiel Designentscheidungen:

Designentscheidungen, die die Optik der App betreffen werden hinterfragt:

- “why are the profile pictures so massive? [..]”
- terrible design don’t understand why it’s randomly changing positions

Security

Unklarheiten bezüglich adressierter Sicherheitsrelevanter Themen oder Berechtigungen

- “I don’t understand how these scammers could get my info and do this “
- “Why do you need access to all data “

Business:

Fragen die nicht das System direkt betreffen und sich an den Anbieter fern von Softwarebereich adressieren.

-“I donâ€™t know why you keep emailing me “

-“I doesnâ€™t make sense to make something cost money “

-“£2.99 ??? Please explain.“

-“Why canâ€™t they just leave things shipped by UPS or FedEx? „

A5 Zuordnung der in der Arbeit zitierten Reviews

Review	ID	Art der Labelung
„Can't sign in. Useless. server error“	939866ed-3282-46fc-9e56-12a5ff759a9f38321	Kein Erklärungsbedarf
“I don't know why this app has mostly 5 stars?”	660ac3cc-e22c-415a-9810-833c9de37b5148042	Kein Erklärungsbedarf
“I even remembered a few albums I had downloaded but guess what? Since they aren't in any of the lists I have to search for the artist to find them, which doesn't work without internet”	1ad45638-1fb2-4edb-a82f-abff890a669752167	Kein Erklärungsbedarf
[..]How is that possible in this noisy city [..]”	d339d7a1-1088-42cc-91dd-57564f874c4964502	Kein Erklärungsbedarf
“[..]How is that possible in this noisy city [..]”	ce3fff32-3e4d-4eef-a42f-	Kein Erklärungsbedarf

	d67682b5c 05c8460,	
How do I subscribe and get an IP address	172c1e6f- 943b-43e3- 82ba- 0c93f974a3 5379006	Operation explizit
don't understand why we can't search	7f0e4d72- 65ba-441f- a45c- 870d39ee9 ec662523	Operation implizit
Very hard to understand how to set up	7deb3ace- 43e5-4b81- b275- f484f16047 6257510	Einführung, imp
But where is instruction for setup?	06047b fb-9faf- 4580- a07a- 35bf0ca f61537 8999	Einführung, exp
where is it? How is it accessed?	3bbba3 a9- e8e2- 420e- 826f- f7701c9 452d25 4497	Navigation, exp
it very unintuitive to find	3f95ff6 6-6c49- 4424- 99ed- 3b1b96 2cec74 81498	Navigation, imp
but I just randomly lost all of my information? That's super irritating	f9a50e d3- 3fb5- 4279- a429- def9bd	Unerwartets Verhalten, imp

	0c8f697 011	
Why can't I sign in?	f9571b87- be54-4ce1- b474- 1a81d894f3 9822003	Unerwartes Verhalten, exp
tell me why it does not sort the list of non- working and working countries -	313c9ff2- 5e2d-4af0- 9ba8- 13af9242e0 c079004	Unerwartetes Verhalten exp
I never ask for them to pick things for me I picked my own playlist and now they're choosing it for me?	834c95ea- c872-4e93- bd4d- fdc0cefedc 9d51653	Algorithmus exp
I also find it irritating that if I change the speed on a book it changes that speed for all books	0befd7f6- 3dbe-4346- a4b5- d74878393 d9f49551	Konsequenzen imp
When I enter the game, why is it consuming the network????	ef22d937- e3ef-410e- 85f2- 130d959a6 1811008	Konsequenzen exp
not sure what the difference is between my playlists and shared playlists	98251c4a- 8123-4723- 9c32- b8d4ac28d 09713013	Systemspezische Aspekte imp
Why is the most expensive texture pack in the store so simple?	ff4cfdd5- 9cb3-4dbb- ac9c- 0081e39b6 34d46000	Systemspezische Aspekte exp
I don't understand that. I tried it five times gives the same error.	6c873848- 366e-4da4- a262- 94286082a a6b29009	Bugs imp

Is this a bug?	94c5c384- d416-4e76- bc29- 63d2d6eab 26a70520	Bugs exp
"[..]why is it so hard to prevent people from hacking servers?[..]"	5e688878- a7c2-47c3- af97- ac0cca69a 3be46077	Meta, exp
doesn't block ads in apps:Why doesn't this block the ads in my apps?! So frustrating please fix this!!!! HOW do I block the ads in other apps!?	8535d3f3- 1ac2-4141- 896b- 1c6195cab 19353535"	Meta, exp
Don't understand what SSN means	164cab09- 20f8-452f- a23c- 007e0b918 88b36999	Domain, imp
Pricing is not in USD. What is COP?	e096eef1- af1d-43a6- 9986- bb7e37bb4 10c37002	Domain, exp
"what's the difference between pro and premium?"	a2e4f978- 9344-4628- beb0-	Indirekte Aspekte, exp

	cc65d24eb 47b18022	
“[...] what does the shield do [...]”	30bd 1f4d- f22a- 49da- b9d5- 5beadd 60887a 45028	Indirekte Aspekte
“why are the profile pictures so massive? [...]”	- 6c022b ec- 8390- 4520- 8cb2- f812cf6 8134b4 7017	GUI, exp
terrible design don't understand why it's randomly changing positions	e978fa 5d- 9b9c- 4af4- b25e- 453f53 4c5195 8042	GUI, imp
I don't understand how these scammers could get my info and do this	- d84f42 ac- 3b65- 409b- 8d6d-	Security, imp

	a3ce2c b7a96f 40500	
Why do you need access to all data	0e2f6869- e67e-4f31- 8870- 935aa76f0f bf15028 -	Security, exp
I don't know why you keep emailing me	bc8e7a79- d397-4c1f- aac2- 2edae4adb c0d80509	Business, imp
I doesn't make sense to make something cost money	b98be84b- 0629-415c- 88c0- f8858a0e9f 1c78503	Businesss, imp
"the instructions for opening a wish jar don't make sense because it says to tap on hatching a pet egg and there's literally nothing that says that in my entire app"	- a0fc0ad 3-7b51- 422c- bdf0- 628d98 a92168 78500	Einführung, imp
Great Wardrobe Tracker:Cher's closet from Clueless has come to life!	- 4d6b36 d8- c325- 4bec- 979e- 8951e9 5e9035 77999	Kein Erklärungsbedarf