1 Einführung

Erklärbarkeit wird im Software Engineering (SE) zunehmend als Fähigkeit verstanden, Systemverhalten so zu kommunizieren, dass es für Nutzer nachvollziehbar und verständlich ist. Im Requirements Engineering (RE) hat sich Erklärbarkeit damit als Non-Functional Requirement (nicht-funktionale Anforderung) (NFR) etabliert [1, 2] und wird als relevanter Aspekt von Softwarequalität diskutiert [3]. Der Bedarf an Erklärungen ist dabei kontextabhängig: Er variiert mit Nutzervoraussetzungen und Nutzungssituation [3, 4], mit dem Typ der Software [5] sowie mit externen Faktoren, welche die Wahrnehmung und Erwartungen an ein System prägen [3]. Entsprechend reicht es nicht aus, Erklärbarkeit lediglich als Eigenschaft von Modellen der Künstliche Intelligenz (KI) zu betrachten. Im RE umfasst sie auch Interaktion, Systemverhalten und Nutzungskontext [6, 7, 8].

Für ein systematisches Erklärbarkeitsmanagement ist die vorgelagerte Identifikation von Erklärungsbedarf zentral. Erklärungsbedarf beschreibt eine Wissenslücke, die geschlossen werden muss, um einen relevanten Systemaspekt im gegebenen Kontext zu verstehen [9, 7]. In der Praxis äußert sich Erklärungsbedarf häufig in nutzernahen Rückmeldekanälen, insbesondere in App-Reviews [9, 10]. Diese Reviews aus dem Apple App Store und Google Play Store enthalten Hinweise auf Zufriedenheit, Fehlerbeobachtungen und Verbesserungsvorschläge und eignen sich damit als Datenbasis für nutzerzentriertes RE (CrowdRE) [11, 12]. Zudem transportieren Reviews häufig eine erkennbare Stimmung (z. B. Frustration oder Begeisterung), die sich mittels Sentimentanalyse auch featurebezogen extrahieren lässt und damit Kontextsignale für Erklärungsbedarf liefern kann [13]. Im Vergleich zu klassischen Erhebungsmethoden wie Interviews oder Fragebögen bieten App-Reviews zwei zentrale Vorteile: Sie entstehen direkt im Nutzungskontext und können dadurch Verzerrungen durch Annahmen oder hypothetische Szenarien reduzieren [14]. Gleichzeitig erfordern sie keine aktive Datenerhebung und lassen sich kontinuierlich sowie in großer Menge automatisiert sammeln, was die Analyse deutlich besser

skalierbar macht als interview- oder umfragebasierte Ansätze. Damit verschiebt sich die Herausforderung jedoch von der Datenerhebung zur Selektion relevanter Inhalte: Nur ein kleiner Anteil der Reviews enthält explizite Erklärungsanfragen (z. B. 5–10%) [9], während App-Stores große Mengen an Feedback erzeugen [12]. Es entsteht somit ein Vorfilterungsproblem, sodass eine manuelle Sichtung und Identifikation erklärungsrelevanter Fälle häufig unpraktikabel ist.

Bestehende Review-Mining-Ansätze zeigen, dass sich Reviews automatisiert strukturieren lassen, etwa durch Klassifikation, Clustering und Priorisierung [15, 16]. Diese Verfahren fokussieren jedoch überwiegend allgemeine Kategorien wie Bug Reports, Feature Requests oder Nutzererfahrungen und adressieren Erklärungsbedarf typischerweise nicht als eigenständiges Ziel [17, 13]. Parallel dazu existieren Erklärbarkeits-Frameworks und Modelle, die Qualitätskriterien und Kontextsensitivität von Erklärungen konzeptionell fassen [18, 19, 20]. Empirische Studien zeigen, dass Erklärungen zu *Übervertrauen* führen können: Wenn Nutzer Ergebnisse kaum überprüfen können, erhöhen plausibel wirkende Erklärungen das Vertrauen und die Bereitschaft, der Systementscheidung zu folgen – auch bei unzuverlässigen Vorhersagen [21]. Damit entsteht eine Forschungslücke: Es fehlt ein durchgängiger, operationalisierter Prozess, der (i) Erklärungsbedarf aus nutzungsnahen Daten zuverlässig erkennt und kategorisiert, (ii) diesen Bedarf konsistent in Erklärbarkeitsanforderungen überführt und (iii) daraus geeignete, hilfreiche Erklärungen ableitet und formuliert.

Diese Dissertation adressiert diese Lücke, indem sie ein integriertes Erklärbarkeitsmanagement untersucht, das den Weg *vom Nutzerfeedback zur Erklärung* systematisch unterstützt. Der Schwerpunkt liegt auf App-Reviews als nutzungsbasierter Datenquelle [9, 11] und auf der Entwicklung wiederverwendbarer Artefakte, die zentrale Aktivitäten im Erklärbarkeitskontext operationalisieren und (semi-)automatisieren [7, 18]. Die Forschungsfragen (Research Questions) (RQs) und das Forschungskonzept werden in Kapitel 4 eingeführt.

1.1 Wissenschaftlicher Ansatz

Die Dissertation folgt einem artefakt- und evidenzorientierten Vorgehen im Sinne der Design Science Research (DSR) von Hevner und Chatterjee [22]. DSR zielt darauf ab, relevante Probleme durch die Konstruktion innovativer Artefakte zu adressieren und zugleich wissenschaftliches Wissen durch deren fundierte Evaluation zu erweitern [22]. Zur Strukturierung wird die Drei-Zyklen-Sicht auf DSR genutzt, bestehend aus *Relevance Cycle*, *Rigor Cycle* und dem zentralen *Design Cycle* [22].

Relevance Cycle (Problem- und Anwendungsbezug) Ausgangspunkt ist die praktische Herausforderung, Erklärungsbedarf in großskaligem Nutzerfeedback effizient zu identifizieren und in umsetzbare Maßnahmen im RE zu überführen [12, 9]. App-Reviews dienen dabei als primäre Datenbasis, da sie reale Nutzungserfahrungen widerspiegeln und zugleich eine hohe Skalierung ermöglichen [11, 15]. Akzeptanzkriterien ergeben sich aus der Nützlichkeit der Artefakte in typischen Analyse- und

Entscheidungsprozessen, beispielsweise durch Reduktion manueller Aufwände und konsistentere Spezifikation erklärungsbezogener Anforderungen.

Rigor Cycle (Wissensbasis und methodische Fundierung) Die Artefaktgestaltung wird durch bestehende Erkenntnisse zu Erklärbarkeit als NFR, Qualitätsmodellen und Operationalisierung sowie kontextsensitive Perspektiven auf Erklärungen fundiert [18, 23, 24, 19, 20]. Zusätzlich werden Befunde zur Wirkung und zu Risiken von Erklärungen berücksichtigt, um Anforderungen an Qualität, Transparenz und Validierung zu begründen [21]. Für automatisierte Analysen werden etablierte Konzepte aus Review-Mining und domänenspezifischer Textanalyse herangezogen [16, 17, 25].

Design Cycle (Build–Evaluate) Im Design Cycle werden Artefakte iterativ konstruiert und evaluiert [22]. Entlang der End-to-End-Kette entstehen insbesondere: (i) Taxonomien, Annotationsrichtlinien und Datensätze zur Operationalisierung von Erklärungsbedarf [9, 7], (ii) Klassifikations- und Analyseverfahren zur automatisierten Erkennung und Kategorisierung von Erklärungsbedarf in Reviews in Anlehnung an Review-Mining-Pipelines [15, 16], sowie (iii) Leitlinien, Templates und Formulierungshilfen zur konsistenten Erstellung von Erklärbarkeitsanforderungen und zur Formulierung hilfreicher Erklärungen [18, 24]. Die Evaluation erfolgt artefaktadäquat und kombiniert quantitative und qualitative Methoden. Für Daten- und Labelqualität werden Reliabilitätsanalysen eingesetzt, für Klassifikationsverfahren werden etablierte Leistungsmetriken verwendet [26]. Für Leitlinien und Tools werden nutzerzentrierte Evaluationsformen wie Interviews, Workshops, Umfragen und heuristische Analysen herangezogen. Das methodische Vorgehen orientiert sich dabei an etablierten Prinzipien empirischer SE-Forschung [27, 28].

1.2 Struktur der Dissertation

Abbildung 1.1 zeigt eine Übersicht über den Aufbau der Dissertation und visualisiert die durchgängige Linie vom Problemkontext über Artefaktentwicklung bis zur Evaluation.

Kapitel 2 legt die theoretischen Grundlagen zu Erklärbarkeit, Erklärungsbedarf und relevanten Analyseverfahren. Kapitel 3 ordnet die Arbeit in den Forschungsstand ein und leitet die Forschungslücke her. Kapitel 4 beschreibt Forschungsziel, Forschungskonzept und die Einordnung in einen RE-Prozess für Erklärbarkeitsmanagement.

Die folgenden Kapitel sind entlang der End-to-End-Kette des Erklärbarkeitsmanagements strukturiert: Kapitel 5 behandelt die systematische Erkennung und Kategorisierung von Erklärungsbedarf in Nutzerfeedback. Darauf aufbauend fokussiert Kapitel 6 die konsistente Formulierung von Erklärbarkeitsanforderungen, bevor Kapitel 7 die Erstellung hilfreicher Erklärungen adressiert. Kapitel 8 integriert die zuvor entwickelten Artefakte und überprüft deren industrielle Anschlussfähigkeit im Anwendungskontext anhand zweier Fallstudien. Das abschließende Kapitel 9 fasst

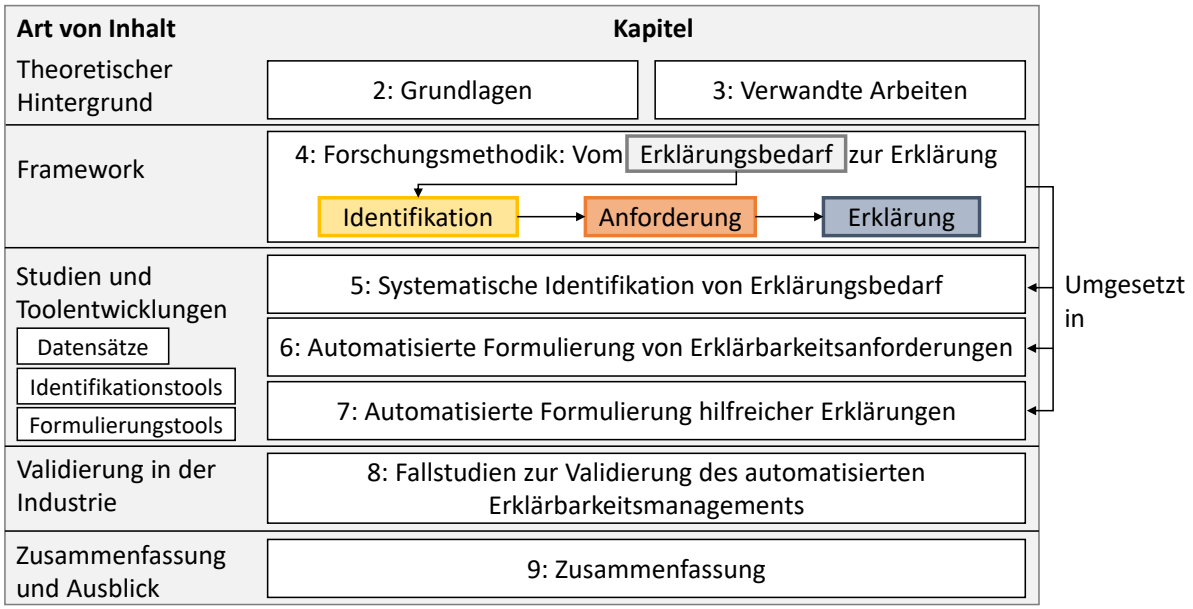

Abbildung 1.1: Struktur der Dissertation

die zentralen Erkenntnisse zusammen, ordnet die Ergebnisse kritisch ein und skizziert zukünftige Forschungsperspektiven.

2
Grundlagen

2.1 Erklärbarkeit

Mit der stetig zunehmenden Komplexität moderner Softwaresysteme und dem verstärkten Einsatz datengetriebener oder KI-basierter Komponenten wächst das Bedürfnis der Nutzer nach einem besseren Verständnis des Systemverhaltens [1]. Während klassische Qualitätsmerkmale wie *Usability* und *Transparenz* den Zugang zu einem System erleichtern, beantworten sie nicht hinreichend die Frage, *warum* ein System eine bestimmte Entscheidung trifft oder sich auf eine bestimmte Weise verhält. Dieses Bedürfnis wird im RE durch das Qualitätsmerkmal der Erklärbarkeit (*Explainability*) aufgegriffen, das sich in den letzten Jahren als eigenständige NFR etabliert hat [23, 7, 2].

2.1.1 Definition und konzeptionelle Grundlagen

Erklärbarkeit beschreibt die Eigenschaft eines Softwaresystems, Informationen so bereitzustellen, dass relevante Systemaspekte für einen bestimmten Adressaten nachvollziehbar und verständlich werden.

Eine häufig zitierte Definition nach Chazette et al. [29] lautet in deutscher Übersetzung:

> Definition: *Ein System S ist erklärbar in Bezug auf einen Aspekt X von S relativ zu einem Adressaten A im Kontext C genau dann, wenn es eine Entität E (den Erklärer) gibt, die A durch Bereitstellung eines Informationskorpus I (der Erklärung von X) ermöglicht, X von S in C zu verstehen.*

Diese Definition verdeutlicht, dass Erklärbarkeit keine isolierte Eigenschaft des Systems ist, sondern sich aus mehreren kontextabhängigen Komponenten zusammensetzt:

- Aspekt (X): Der zu erklärende Teil des Systems, z. B. ein Algorithmus, ein beobachtetes Verhalten oder die Konsequenzen einer Nutzeraktion.
- Adressat (A): Die Zielgruppe der Erklärung (z. B. Endnutzer, Entwickler, Domänenexperte), deren Vorwissen und Perspektive die Form der Erklärung bestimmen.
- Kontext (C): Die Nutzungssituation, in der die Erklärung erfolgt. Sie beeinflusst Relevanz und Detaillierungsgrad der Informationen.
- Erklärer (E): Die Instanz, die die Erklärung bereitstellt – das kann das System selbst, die Dokumentation oder eine externe Person sein.

Erklärbarkeit fungiert somit als vermittelnde Qualität zwischen System und Nutzer: Sie befähigt Adressaten, das Systemverhalten zu *verstehen*, ohne notwendigerweise technisches Detailwissen zu besitzen. In der Literatur wird sie als *Mittel zum Zweck* beschrieben, das andere Qualitätsziele wie Verständlichkeit, Transparenz oder Vertrauenswürdigkeit unterstützt [24].

> Definition: *Ein System gilt als erklärbar, wenn es Informationen bereitstellt, die es dem jeweiligen Nutzer ermöglichen, einen relevanten Aspekt seines Verhaltens oder seiner Funktionsweise in einem gegebenen Nutzungskontext nachzuvollziehen und zu verstehen.*

Diese von mir aufgestellte vereinfachte Definition verzichtet bewusst auf formale Parameter, um den semantischen Kern hervorzuheben: das Bereitstellen verständlicher Informationen über Systemaspekte. Während die Definition von Chazette et al. [29] Bedingungen für ein erklärbares System beschreibt, charakterisiert die Definition nach Droste et al. [7] den Begriff *Erklärbarkeit* selbst:

> Definition: *Erklärbarkeit bezeichnet die Fähigkeit einer Software, Erklärungen bereitzustellen, die Informationen über beliebige Systemaspekte für ihre Stakeholder verständlicher machen.*

Erklärbarkeit bezeichnet das übergeordnete Qualitätsziel, Software so zu gestalten, dass sie Erklärungen über ihr Verhalten, ihre Abläufe oder Entscheidungen bereitstellen kann. In der Literatur wird *Explainability* teilweise als *Erklärungsfähigkeit* übersetzt. Letztere bezieht sich jedoch auf die tatsächlich realisierte Fähigkeit eines Systems, Erklärungen bereitzustellen, während *Erklärbarkeit* das übergeordnete Ziel beschreibt, diese Fähigkeit bereits im Systemdesign zu ermöglichen und sicherzustellen [29, 23, 24].

In der Forschung zur *Explainable Artificial Intelligence (XAI)* wird der Begriff *Explainability* häufig enger gefasst und auf die Erklärung interner Modellmechanismen beschränkt [30]. Gilpin et al. [31] beschreiben Explainability als Mittel zur Erreichung

von *Interpretierbarkeit* und *Vollständigkeit*, wobei ersteres das Verständnis der internen Abläufe eines Modells und letzteres deren korrekte Darstellung umfasst. In der Regel dient Explainability in der XAI somit der Bereitstellung menschlich verständlicher Begründungen für Modellentscheidungen, um Vertrauen und Transparenz zu fördern [32, 33, 34]. Im RE hingegen wird Erklärbarkeit deutlich umfassender verstanden: Sie beschreibt das Qualitätsmerkmal erklärbarer Software im weiteren Sinne – einschließlich Interaktion, Systemverhalten und Nutzungskontext [6].

2.1.2 Erklärbarkeit als NFR

Im RE wird Erklärbarkeit als NFR betrachtet, die festlegt, *wie nachvollziehbar* und *verständlich* ein System gestaltet sein muss. Sie steht in enger Beziehung zu anderen Qualitätseigenschaften und kann deren Ausprägung sowohl fördern als auch einschränken:

- Transparenz: Eine Software gilt als transparent, wenn Informationen über ihre internen Abläufe verfügbar und zugänglich sind. Erklärbarkeit erweitert diesen Aspekt, indem sie die bereitgestellten Informationen für die jeweilige Zielgruppe verständlich aufbereitet.
- Vertrauen und Akzeptanz: Verständliche Erklärungen erhöhen die wahrgenommene Vertrauenswürdigkeit eines Systems, insbesondere in sicherheits- oder datensensitiven Domänen [23].
- Nutzbarkeit und mentale Belastung: Übermäßige oder unpassende Erklärungen können die Nutzbarkeit beeinträchtigen und zu kognitiver Überforderung führen. Daher müssen Umfang, Zeitpunkt und Format der Erklärungen sorgfältig an die Nutzungssituation angepasst werden [7].

Erklärbarkeit ist somit Teil eines multidimensionalen Qualitätsgefüges, in dem sie mit anderen NFRs interagiert und teilweise konkurrierende Ziele adressiert.

2.1.3 Erklärungsbedarf und Erklärbarkeitsanforderungen

Voraussetzung für die Spezifikation erklärbarer Systeme ist die Identifikation des Erklärungsbedarfs (*Explanation Need*). Ein Erklärungsbedarf beschreibt eine Wissenslücke, die ein Adressat schließen möchte, um einen bestimmten Aspekt eines Systems besser zu verstehen [9]. Das Konzept wurde in den Kontext des RE übertragen, um daraus konkrete *Erklärbarkeitsanforderungen* abzuleiten [7].

Die folgende vereinfachte Formulierung basiert auf Unterbusch et al. [9] und lehnt sich an die formale Definition von Chazette et al. [29] an, verzichtet jedoch bewusst auf mathematische Parameter, um den semantischen Kern für den RE-Kontext zu betonen:

Definition: *Ein Erklärungsbedarf liegt vor, wenn ein Nutzer ein unvollständiges Verständnis über einen bestimmten Aspekt eines Softwaresystems hat und daher zusätzliche Informationen benötigt, um dessen Verhalten oder Funktionsweise in der jeweiligen Nutzungssituation nachvollziehen zu können.*

Der Erklärungsbedarf bildet somit den Ausgangspunkt für die Ableitung konkreter *Erklärbarkeitsanforderungen*, also Anforderungen, die festlegen, welche Erklärungen ein System bereitstellen soll, um diese Wissenslücken der Stakeholder gezielt zu schließen.

Ein solcher Bedarf kann in Nutzerfeedback oder App-Reviews *explizit* auftreten (etwa durch direkte Fragen, Kritik oder Forderungen nach Erklärungen) oder *implizit*, etwa durch Formulierungen, die Unverständnis, Verwirrung oder unerwartetes Verhalten ausdrücken [35, 36, 8, 37]. Die systematische Identifikation und Klassifikation solcher Bedarfe bildet die Grundlage für das Ableiten von Erklärbarkeitsanforderungen und die Entwicklung geeigneter Erklärungskonzepte innerhalb des Softwaredesigns.

2.2 Künstliche Intelligenz

Die Verarbeitung natürlicher Sprache (Natural Language Processing (NLP)) hat sich von frühen statistischen Verfahren zu einem zentralen Bestandteil moderner Künstlicher Intelligenz entwickelt [38]. Frühe Ansätze, etwa N-Gramm-Modelle, ermöglichten lediglich eine begrenzte Kontextbetrachtung und lieferten damit nur eingeschränkt sprachliches Verständnis [39]. Mit der Einführung neuronaler Netze, insbesondere Rekurrenter Neuronaler Netze (RNNs), konnten erstmals längere Wortfolgen modelliert und komplexere sprachliche Strukturen erfasst werden [40].

Einen grundlegenden Durchbruch stellte das Transformer-Modell von Vaswani et al. [41] dar. Durch den Mechanismus der *Self-Attention* kann es große Textkontexte parallel verarbeiten und langfristige Abhängigkeiten deutlich zuverlässiger abbilden als frühere Architekturen. Transformer-Modelle bilden daher die Basis nahezu aller modernen Sprachmodelle und dominieren seit ihrer Einführung die Weiterentwicklung von KI-Systemen [42, 43].

Ein aktueller Forschungsschwerpunkt liegt auf der Verbesserung des *Reasoning*, also der Fähigkeit eines Modells, nachvollziehbare Zwischenschritte und logische Argumentationen zu erzeugen. Methoden wie das *Chain-of-Thought Reasoning* [44] erweitern die Transparenz und Kohärenz der Antworten und gelten als wichtiger Schritt hin zu allgemeineren kognitiven Fähigkeiten in KI-Systemen [45].

2.2.1 Transformer

Transformer-Modelle haben sich durch großskaliges Vortraining auf umfangreichen Textkorpora sowie durch Feintuning-Verfahren wie überwachtes Feintuning (Supervised Fine-Tuning) und *Reinforcement Learning from Human Feedback* (RLHF) als

Standardarchitektur moderner Sprachmodelle etabliert [46]. Die daraus resultierenden *Large Language Models (große Sprachmodelle) (LLMs)* verfügen über ausgeprägte Fähigkeiten im Sprachverständnis, in der Textgenerierung und bei allgemeinen kognitiven Aufgaben.

Neben proprietären Modellen haben offene, frei verfügbare Modelle in den vergangenen Jahren erhebliche Fortschritte erzielt. Open-Source-Modelle wie die *Llama-* [47], *DeepSeek-* [48], *Mistral-* [49] oder *Qwen*-Reihe [50] veröffentlichen ihre Modellgewichte, wodurch sie eigenständig gehostet, angepasst und auf domänenspezifische Aufgaben spezialisiert werden können. Diese Offenheit fördert reproduzierbare Forschung, erleichtert die Integration in bestehende Systeme und beschleunigt den Transfer von KI-Technologien in praktische Anwendungen. Transformer-basierte LLMs bilden damit den technologischen Kern von KI und stellen die Grundlage für die im weiteren Verlauf dieser Dissertation eingesetzten Modelle dar.

2.2.2 Large Language Models

Im Folgenden werden die LLMs vorgestellt, die in dieser Dissertation mithilfe des erhobenen Goldstandard-Datensatzes feinabgestimmt und evaluiert werden. Im Mittelpunkt stehen dabei Transformer-Modelle, die entweder als offene Foundation-Modelle mit veröffentlichten Gewichten oder als proprietäre Referenzmodelle eingesetzt werden.

ModernBERT Im Gegensatz zu Decoder-basierten, primär generativen Sprachmodellen stellt *ModernBERT* ein Encoder-basiertes Modell dar, das speziell für effiziente Einbettungs- und Klassifikationsaufgaben konzipiert wurde [51]. Es knüpft konzeptionell an *BERT* [52] und *RoBERTa* [53] an, integriert jedoch aktuelle Architekturverbesserungen und unterstützt deutlich längere Kontextlängen.

Encoder-Modelle wie *ModernBERT* sind insbesondere für Aufgaben geeignet, bei denen Texte in Vektorrepräsentationen überführt und anschließend klassifiziert oder in Retrieval-Szenarien verwendet werden. In dieser Dissertation wird *ModernBERT* als ressourceneffiziente Alternative zu großen Decoder-basierten LLMs eingesetzt, um zu untersuchen, inwieweit kompaktere Modelle bei der Emotions- und Sentimentanalyse mit großen Sprachmodellen konkurrieren können.

SetFit *SetFit* (Sentence Transformer Fine-tuning) ist ein promptfreies Verfahren zur effizienten Feinabstimmung von Sentence-Transformer-Modellen für Textklassifikationsaufgaben, insbesondere bei wenigen gelabelten Beispielen [54]. Anstelle generativer Modelle nutzt *SetFit* dichte Texteinbettungen und folgt einem zweistufigen Vorgehen: Zunächst wird das Einbettungsmodell kontrastiv auf Basis gelabelter Textpaare angepasst, anschließend wird ein schlanker Klassifikator auf den erzeugten Einbettungen trainiert [54].

In dieser Dissertation wird *SetFit* als ressourceneffiziente Alternative zu großen Decoder-basierten LLMs betrachtet, um die Leistungsfähigkeit encoderbasierter Ansätze für die Emotions- und Sentimentanalyse einzuordnen.

Llama Die von *Meta AI* entwickelte *Llama*-Reihe stellt eine Familie offener Foundation-Modelle unterschiedlicher Größe dar, deren Gewichte der Forschungsgemeinschaft zur Verfügung gestellt werden [47, 55]. Die aktuellen *Llama-3*-Modelle wurden auf umfangreichen, mehrsprachigen Textkorpora vortrainiert und anschließend durch überwachte Feinabstimmung und Präferenzoptimierung an instruktionsbasierte Anwendungsfälle angepasst [55].

Durch die Offenheit der Gewichte eignen sich *Llama*-Modelle besonders für die domänenspezifische Feinabstimmung. In dieser Dissertation werden sie eingesetzt, um zu untersuchen, inwieweit offene LLMs auf einen speziell annotierten Datensatz im SE-Kontext angepasst werden können und welche Leistungsfähigkeit sie dabei in der Stimmungs- und Emotionsklassifikation erreichen.

Qwen Die *Qwen*-Modelle von *Alibaba* sind ebenfalls offene LLMs, die in mehreren Modellgrößen bereitgestellt werden und auf einem stark erweiterten, qualitativ kuratierten Trainingskorpus basieren [45]. Der Schwerpunkt der Entwicklung lag auf mehrsprachigen, insbesondere technischen und wissenschaftlichen Texten, um eine hohe Leistungsfähigkeit in komplexen, wissensintensiven Domänen zu erreichen.

Durch moderne Feinabstimmungs- und Verstärkungsverfahren, darunter überwachte Feinabstimmung und Präferenzoptimierung, zielen *Qwen*-Modelle auf präzise, strukturierte und sachliche Antworten ab. In dieser Dissertation dienen sie als weitere offene Modellfamilie, um die Übertragbarkeit der gewählten Feinabstimmungsstrategie auf unterschiedliche Architekturen und Trainingsdaten zu untersuchen.

Gemini Die *Gemini*-Modellfamilie von *Google DeepMind* umfasst leistungsstarke, proprietäre Foundation-Modelle auf Basis eines Transformer-Decoders [43]. Die Modelle sind multimodal, unterstützen lange Kontextlängen und wurden auf umfangreichen, mehrsprachigen Text- und Multimediadaten trainiert [56].

Ein besonderer Fokus liegt auf der Zuverlässigkeit der ausgegebenen Informationen und auf Sicherheitsaspekten, die durch gezielte Trainings- und Ausrichtungsverfahren adressiert werden [57, 43]. In dieser Dissertation wird *Gemini* als Closed-Source-Referenz genutzt, um die Ergebnisse der feinabgestimmten Open-Source-Modelle einzuordnen.

GPT *GPT-4* [42] ist ein proprietäres, auf der Transformer-Decoder-Architektur basierendes Sprachmodell [41], das durch umfangreiches Vortraining und einen mehrstufigen Ausrichtungsprozess (Alignment) optimiert wurde. Neben klassischen Vorhersageaufgaben wurde besonderer Wert auf Sicherheit, Robustheit und kontrollierbares Verhalten gelegt. Hierzu kommen unter anderem Reinforcement Learning aus menschlichem Feedback (RLHF) und weitere sicherheitsorientierte Verfahren zum Einsatz [42].

In dieser Dissertation dient *GPT-4* als leistungsstarkes Closed-Source-Modell, das insbesondere für Vergleichsexperimente und zur Einschätzung der Leistungsobergrenze bei der Emotions- und Sentimentanalyse herangezogen wird.