

# Künstliche Intelligenz im Requirements Engineering

## Eine Systematische Literaturrecherche

Judith Breuninger<sup>1</sup>, Franziska Kücher<sup>2</sup> und Natali Mistic<sup>3</sup>

**Abstract:** Dem Einsatz von Künstlicher Intelligenz (KI) im Requirements Engineering (RE) wird ein hohes Potenzial zugeschrieben. Der Stand der Forschung gestaltet sich jedoch unübersichtlich. Im Rahmen einer Systematischen Literaturrecherche werden 27 wissenschaftliche Publikationen aus drei Datenbanken identifiziert und analysiert. Anschließend werden diese in die RE-Phasen der Anforderungserhebung, -analyse, -spezifikation und -validierung eingeordnet und zusammengefasst. Die Ergebnisse zeigen, dass KI in den vier Phasen eingesetzt wird, allerdings ist die Anwendung unterschiedlich stark ausgeprägt. Weitere tiefgehende Forschungsarbeit, insbesondere zum Einsatz von KI in der Anforderungvalidierung, ist notwendig. Die vorliegende Arbeit stellt dafür einen wesentlichen Ausgangspunkt dar, indem sie einen strukturierten Überblick der verschiedenen KI-Methoden zum Einsatz im RE aufzeigt und diskutiert.

**Keywords:** Künstliche Intelligenz, Requirements Engineering, Systematische Literaturrecherche

## 1 Einleitung

Obwohl die Wurzeln der Künstlichen Intelligenz (KI) bis in die Mitte des zwanzigsten Jahrhunderts zurückreichen, besitzt das Thema insbesondere heute hohe Relevanz in Wissenschaft und Praxis [Er08]. Die Zahl der qualitätsgeprüften Publikationen stieg bspw. im Jahr 2018 um knapp die Hälfte im Vergleich zum Vorjahr [Ac20].<sup>4</sup> Im Requirements Engineering (RE) leistet KI Unterstützung aufgrund der steigenden Aufgabenkomplexität. Zukünftig könnte der Problemlösungsfähigkeit durch KI eine relevante Rolle bzgl. der Automatisierung von Prozessen im RE zukommen [Ba20], [Er08], [De15].

Das RE beschäftigt sich mit Anforderungen während des gesamten Lebenszyklus des Softwareprodukts. Ziel ist es, ein grundlegendes Verständnis der Anforderungen bei allen Projektbeteiligten zu schaffen. Bereits in den frühen Phasen der Entwicklung können Fehler aufgezeigt, behoben und damit Kosten reduziert werden, um eine erfolgreiche Softwareentwicklung zu erreichen [BF14], [Eb19], [IE90], [JIA18], [Ng12], [Pa10].

In der wissenschaftlichen Literatur wird KI zwar intensiv erforscht, gilt jedoch zugleich als schwer definierbar. Grundsätzlich wird ein System als künstlich intelligent verstanden,

---

<sup>1</sup> Universität Stuttgart, Keplerstr. 17, 70174 Stuttgart, judith.breuninger7@gmail.com

<sup>2</sup> Universität Stuttgart, Keplerstr. 17, 70174 Stuttgart, franzi.kuecher@gmail.com

<sup>3</sup> Universität Stuttgart, Keplerstr. 17, 70174 Stuttgart, natali.mistic@web.de

<sup>4</sup> Im Jahr 2018 wurden knapp 90.000 Beiträge in Europa, China und den USA veröffentlicht, was einen Anstieg von ungefähr 45 Prozent im Vergleich zum Vorjahr bedeutet [Ac20].

wenn es Aufgaben oder Konzepte selbstständig bzw. effizient erlernen und Herausforderungen in der realen Welt lösen kann [Ba20], [Er08], [Ma16]. Trotz des hohen Potenzials von KI im RE weist die Fachliteratur darauf hin, dass sich Forschung noch am Anfang befindet. Darüber hinaus existiert kein übersichtlicher Forschungsstand zum Einsatz von KI im RE [Ba20], [De15]. Zur Lösung dieses Problems soll die vorliegende Arbeit einen Beitrag leisten. Im Rahmen einer Systematischen Literaturrecherche (SLR) wird der Status quo hinsichtlich des Einsatzes von KI im RE erarbeitet und für weitere Forschung dargelegt. Aufgrund dessen lautet die Forschungsfrage: “Wie gestaltet sich der Einsatz von KI-Methoden in den einzelnen Phasen des RE nach aktuellem Stand der wissenschaftlichen Literatur?”.

## 2 Methodik

Zur strukturierten Erhebung und Analyse der wissenschaftlichen Literatur orientiert sich diese Forschungsarbeit am Ansatz zur Durchführung einer SLR nach Kitchenham [Ki09].

Der Suchterm der SLR lautet: „artificial intelligence“ AND „requirements engineering“. Dieser wird in allen durchsuchbaren Feldern, insbesondere im Titel, Abstract und Volltext, für den Zeitraum von 2010 bis 2020 innerhalb der folgenden Datenbanken angewandt: Association for Computing Machinery Digital Library (ACM DL), ScienceDirect sowie EBSCOhost. Die Datenbanken bieten Zugang zu umfassenden Volltext-Artikeln, Journalen und Fachliteratur [As15], [EB20], [EI20].

Der in Abbildung 1 dargestellte Exklusionsprozess der relevanten Literatur erfolgt in fünf Schritten im Zeitraum von Mai bis Juni 2020. Die anfänglichen Suchtreffer belaufen sich auf 746 Publikationen. In drei Schritten werden die Suchergebnisse eingegrenzt. Schließlich bleiben nach dem vierten Schritt 30 Publikationen bestehen. Durch eine identische Publikation in zwei Datenbanken kann die Anzahl auf 29 Publikationen reduziert werden.<sup>5</sup> Des Weiteren werden zwei Publikationen ausgeschlossen, die ebenfalls eine SLR durchführen und damit für diese Arbeit Sekundärquellen darstellen. Zudem verwenden die ausgeschlossenen SLR eine abweichende Phaseneinteilung des RE und beziehen sich lediglich auf jeweils eine KI-Methode. Im Rahmen der Datensynthese werden 27 Publikationen analysiert und den Phasen des RE inhaltlich zugeordnet.

Die Phaseneinteilung des RE ist essentiell für das methodische Vorgehen, da diese die Grundlage der Strukturierung und Auswertung der Ergebnisse bildet. Die vorliegende Arbeit orientiert sich an der Definition des IEEE, welche das RE in die folgenden vier Phasen unterteilt [IE90]: Die Anforderungserhebung als erste Phase im RE hat den Zweck, Anforderungen an ein Softwaresystem zu ermitteln oder aus Systemanforderungen abzuleiten. Dabei wird ein umfassendes Verständnis der Problemstellung erarbeitet, welche das

---

<sup>5</sup> Die Publikation von [IAA18] entstammt den Datenbanken ScienceDirect und EBSCOhost.

zu entwickelnde Softwaresystem lösen soll. In der zweiten Phase, der Anforderungsanalyse, wird die Vollständigkeit, Korrektheit, sowie die technische und finanzielle Durchführbarkeit auf Basis der erhobenen Anforderungen untersucht und bewertet. Außerdem werden Grenzen des Softwaresystems identifiziert, indem dieses von seinem Kontext abgegrenzt wird. In der Anforderungsspezifikation werden im dritten Schritt die erhobenen und analysierten Anforderungen im Spezifikationsdokument präzisiert festgehalten. Das Dokument wird dabei systematisch überprüft, bewertet und genehmigt. In der vierten Phase, der Anforderungsvalidierung, wird das Spezifikationsdokument auf Vollständigkeit und Korrektheit überprüft [BF14], [DD10].

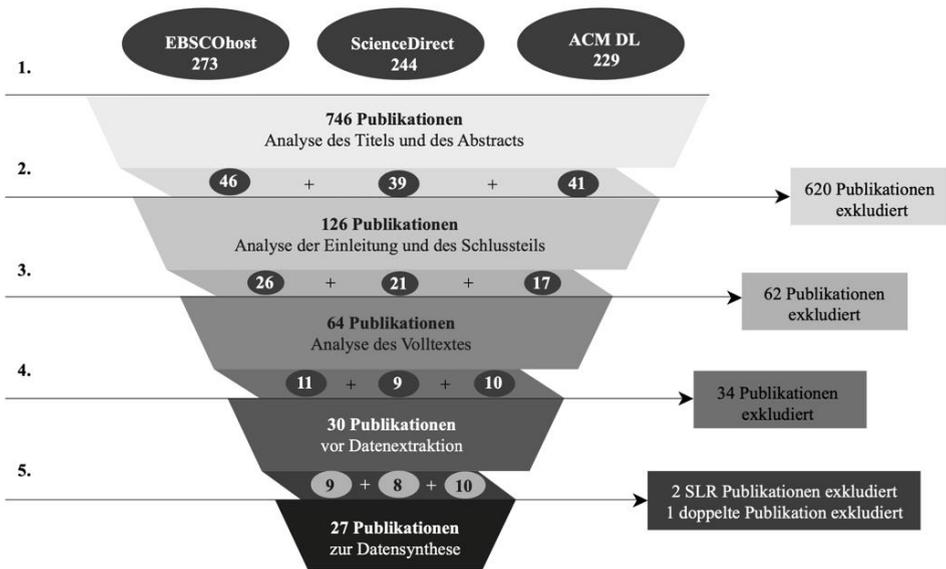


Abb. 1: Exklusionsprozess im Rahmen der SLR

### 3 Ergebnisse

Aus der SLR können zum Einsatz von KI im RE sieben Publikationen der Anforderungserhebung zugeordnet werden. Für die Anforderungsanalyse und Anforderungsspezifikation werden jeweils neun Publikationen identifiziert. Zwei Publikationen werden in die Anforderungsvalidierung eingeordnet. In wenigen Publikationen werden KI-Methoden phasenübergreifend thematisiert, was eine konkrete Zuordnung erschwert. In diesen Fällen wird die Publikation in jene Phase eingeordnet, die vordergründig behandelt wird. Eine zusammenfassende Übersicht aller relevanten Publikationen und den darin thematisierten KI-Methoden enthält Tabelle 1.

<b>Nr.</b>	<b>Literatur</b>	<b>KI-Methode</b>
<b>Anforderungserhebung</b>		
1	[DR20]	Algorithmus basierend auf NLP
2	[GAS17]	ML
3	[Ma19]	KNN, Word Embedding, ML, Inference Machine
4	[MX19]	NLP, NLU
5	[Ng12]	Ontologies
6	[Ro19]	Data Mining
7	[We12]	Tool basierend auf NLP
<b>Anforderungsanalyse</b>		
8	[BMS20]	Neural Word Embedding Model
9	[Ca14]	NLP, Data Mining
10	[DS11]	Unified Modeling Language Model Generator with NLP
11	[GG18]	Particle Swarm Optimization basierend auf ML
12	[Ka19]	KNN, Deep-Learning Algorithm
13	[Lu17]	Tool basierend auf NLP
14	[MSP16]	NLP, ML
15	[Pa15]	Rule Induction Techniques basierend auf ML
16	[Vi10]	Multiagent Systems
<b>Anforderungsspezifikation</b>		
17	[BLG17]	Ontologies
18	[ENT18]	Tool basierend auf NLP
19	[HK16]	ML, Association Rule Mining
20	[IAA18]	KNN, NLP
21	[JIA18]	NLP, Semantic Role Labeling
22	[JN14]	NLP, Inference Machine, Natural Language Generation
23	[Na20]	Stanford CoreNLP
24	[Sa12]	NLP, Semantics
25	[Zh10]	Algorithm basierend auf ML
<b>Anforderungvalidierung</b>		
26	[Al15]	Logic-based Learning
27	[Mo16]	Tool basierend auf NLP

Tab. 1: Ergebnisse der SLR

### 3.1 Anforderungserhebung

In [GAS17] werden zwei Ansätze Maschinellem Lernen (ML) vorgestellt. Durch ML kann eine Software auf Basis bestehender Datensätze trainiert werden, um darauf aufbauend Vorhersagen zu treffen. Mit Hilfe von Algorithmen können Anforderungen an eine

Software aus Twitter Tweets erhoben werden. Diese können relevante Informationen für Requirements Engineers enthalten, wie bspw. Softwarefehler oder Wünsche für neue Funktionen. Auf Basis manuell ausgewerteter Tweets werden fünf Algorithmen trainiert, um den Text der Tweets zu analysieren. Als Ergebnis des ersten Ansatzes zeigt sich, dass Tweets nach ihrer Relevanz für technische, nicht-technische und allgemeine Stakeholder klassifiziert werden können. Somit können Requirements Engineers die für sie relevanten technischen Tweets für die Erhebung von Anforderungen nutzen. Im zweiten Ansatz lassen sich durch die Verwendung der Algorithmen von Bots generierte Tweets filtern. Dadurch können sich Stakeholder und Requirements Engineers auf relevante, manuell verfasste Tweets konzentrieren [BS19]. [Ma19] beschreibt zudem die Anwendung von ML in Form von Künstlichen Neuronalen Netzen (KNN). Dabei handelt es sich um ein Netzwerk, welches das biologische Nervensystem zur Informationsverarbeitung nachahmt. KNN tragen dazu bei, unerwünschte Verhaltensweisen eines späteren Produkts frühzeitig zu erkennen und zu reduzieren. Bspw. werden diese eingesetzt, um bei der Generierung von Modellen Elemente aus Use Cases zu filtern [IAA18].

[Ng12] thematisiert ein wissensgestütztes Framework, das auf Ontologien basiert. Letztere beschreiben Relationen zwischen Objekten. Das Domänenwissen und die Semantik von Anforderungen werden in einer Ontologie kodiert, sodass implizite Konsequenzen von expliziten Merkmalen in Anforderungen enthüllt werden. Des Weiteren wird eine Beschreibungsllogik als Regel in der Ontologie verwendet, die zur Erfassung von Anforderungen sowie deren Einschränkungen und Beziehungen notwendig ist. Diese eignet sich, um Domänenkonzepte sinnvoll zu repräsentieren. Folglich lassen sich Inkonsistenzen und fehlende Elemente in Anforderungen identifizieren und beheben. Darüber hinaus werden zur Realisierbarkeit einer Anforderung automatisch Analysen durchgeführt [Er08], [Ng12].

Natural Language Processing (NLP) ist ein Prozess der Verarbeitung natürlicher Sprache zu digitalisiertem Text. Eine bedeutende Herausforderung stellt nach [MX19] das Verständnis und die Interpretation der natürlichen Sprache dar. Dabei handelt es sich um Natural Language Understanding (NLU), welches ein Teilbereich des NLP ist. Der Einsatz betrifft Sprachaktivierung, Textkategorisierung, Inhaltsanalyse und Beantwortung von Kundenfragen. Mit Hilfe von Techniken des NLU wird maschinelles Lesen und die Interpretation natürlicher Sprache ermöglicht. Ziel ist es, die Formulierung von Anforderungen qualitativ zu verbessern, indem Redundanzen, Mehrdeutigkeiten und Unvollständigkeiten identifiziert werden. In [DR20] wird eine interaktive Chatbot-Anwendung für Android eingesetzt, die mit Hilfe eines Algorithmus basierend auf NLP eine Nutzeranfrage aus einem Chat aufbereitet. Aus einer auf KI basierenden Wissensdatenbank generiert der Chatbot anschließend die Antwort auf eine Nutzeranfrage. Dadurch gelingt die Schließung der Kommunikationslücke zwischen Entwicklern und Nutzern eines Softwaresystems. Durch die Interaktion mit den Nutzern können neue Anforderungen aufgezeigt und Feedback berücksichtigt werden.

In [Ro19] wird ein auf Data Mining gestützter Ansatz vorgestellt. Dieser soll den Kontext spezifizieren, in dem ein zu entwickelndes Softwaresystem eingesetzt wird. Data Mining

umfasst einen Prozess, welcher Wissen aus Daten zieht und dieses anschließend darstellt und anwendet. Der Ansatz ist in der Lage, Kontextvariablen und Kombinationen solcher Variablen zu identifizieren und die Unsicherheiten im operativen Umfeld zu reduzieren. Mit Hilfe des Ansatzes wird somit die kontextbezogene Anforderungserhebung optimiert und detailliert dargestellt [Er08].

### 3.2 Anforderungsanalyse

[Lu17] und [Ca14] zeigen, dass NLP über die Anforderungserhebung hinausgeht. Die extrahierten Anforderungen werden in der Anforderungsanalyse mit NLP-Tools analysiert und auf ihre Qualität geprüft. Nach [Lu17] können empirische Heuristiken bei der Generierung konzeptueller Modelle unterstützen, um relevante Objekte oder Sachverhalte und deren Beziehungen zu erheben. Ein weiteres Tool generiert spezielle Ansichten des konzeptuellen Modells durch Clustering oder Filtern. [Ca14] beschreibt eine Methode, die NLP und Data Mining einsetzt, um Requirements Engineers bei der Extraktion und Modellierung von Zielen aus Textdokumenten zu unterstützen. Bei der Entwicklung komplexer Systeme besteht häufig die Notwendigkeit, eine große Anzahl von veralteten Anforderungsspezifikationen, Forschungspublikationen oder Standards zu analysieren. Data-Mining-Techniken können helfen, den manuellen Aufwand für die Analyse dieser Ressourcen zu reduzieren, indem die Abstracts der veralteten Dokumente automatisch gesammelt werden. Durch NLP werden anschließend Sätze aus dem Abstract extrahiert, die zielbezogene Schlüsselwörter enthalten. Der Ansatz ist eine effektive Möglichkeit, um Ziele für komplexe und forschungsintensive Systeme zu modellieren. In [DS11] wird eine Technik mit NLP vorgestellt, die Entwickler bei der Generierung von Unified Modeling Language-Modellen aus natürlichsprachlichen Anforderungen unterstützt. Es lassen sich Use Case Diagramme und verschiedene Analysen generieren. Aus [Ca14], [DS11] und [Lu17] kann entnommen werden, dass NLP in dieser Phase, ebenso wie in der Anforderungserhebung, zur Verbesserung von Anforderungsformulierungen und zur Reduktion des manuellen Aufwands eingesetzt wird.

Durch die Kombination von NLP und ML werden laut [MSP16] manuell interpretierbare Cluster aus semantisch verwandten Anforderungen in Textform generiert. Domänenrelevante Begriffe lassen sich extrahieren und die semantische Nähe zwischen Begriffen und Anforderungen analysieren. Für ein besseres Verständnis werden die entstandenen Cluster mit Erklärungen versehen. Laut [Pa15] werden Regelinduktionstechniken, welche auf ML basieren, zur Nachbildung von Bewertungen eingesetzt, die ein Qualitätsexperte eines Projekts vornehmen würde. Als Input erhalten diese einen Datensatz, woraus Lernbeispiele gezogen werden mit dem Ziel, einen Entscheidungsbaum oder eine Zusammenstellung von Regeln zu erstellen. Dadurch können neue Stichproben aus dem Datensatz bestimmt und klassifiziert werden. Durch die frühzeitige Qualitätsüberprüfung der Daten lassen sich fehlerhafte Anforderungen sowie Kosten oder Lieferungsverzögerungen bei der Erreichung der vorgegeben Ziele reduzieren. ML in Form von KNN wird nach [Ka19] genutzt, um Anforderungen aus unvollständigen Produktbeschreibungen auf bspw. Internetseiten zu erhalten und zu modellieren. Die KNN gleichen Anforderungsbeschreibungen

verwandter Softwaresysteme miteinander ab, extrahieren mit Hilfe eines Deep-Learning-Algorithmus relevante Merkmale und entwickeln diese weiter. Hierdurch wird die Qualität der Anforderungen an ein Softwaresystem verbessert.

Die Generierung semantischer Ähnlichkeitsmaße in der Anforderungsanalyse wird durch die Nutzung eines neuronalen Modells zur Wordembedding unterstützt. So lassen sich Glossar-begriffe aus umfangreichen Anforderungsdokumenten gruppieren [BMS20].

### 3.3 Anforderungsspezifikation

Für die Anforderungsspezifikation ist die Extraktion von Aktionen und Akteuren aus natürlichsprachlichen Anforderungen entscheidend. Dieser manuelle Vorgang gilt als anspruchsvoll und zeitaufwendig. [IAA18] stellen die Kombination aus NLP und KNN als eine selbst entwickelte intelligente Technik des RE vor, welche zwei Arten der Verarbeitung, Syntax und Semantik, umfasst. Während der Generierung von Aktionen und Akteuren erfordert der halbautomatisierte Prozess Reaktionen des Benutzers. Jede Interaktion des Benutzers mit dem System wird als Use Case formuliert, aus welchem die Akteure und Aktionen mittels NLP erfasst werden. KNN werden zur semantischen Verarbeitung eingesetzt, um die aus der Anwendung von NLP hervorgegangenen Wörter auf die Semantik des Use Cases zu beziehen. [JIA18] zeigt die Kombination von NLP mit Semantic Role Labeling auf, die zur Extraktion von Aktionen und Akteuren sowie zur Erstellung von Use Case Diagrammen eingesetzt wird. Zum Verständnis der natürlichen Sprache wird durch Semantic Role Labeling Wörtern oder Phrasen eine Beschriftung zugewiesen, welche die semantische Rolle im Satz angibt. Laut [Sa12] kann das Spezifikationsdokument mit einem System, das NLP mit Semantik kombiniert, qualitativ verbessert werden. So lassen sich Mehrdeutigkeiten und Inkonsistenzen automatisch erkennen.

Im Kontext agiler Methoden kann NLP das Schreiben funktionaler Anforderungen unterstützen. [ENT18] beschreibt die automatisierte Umwandlung von User Stories in Use Case Diagramme, die in Unified Modeling Language abgebildet werden. Zunächst wird eine Textdatei mit User Stories durch einen Algorithmus um alle verzichtbaren Wörter reduziert. Anschließend wird mit Hilfe von NLP die neue Datei zerlegt und Syntaxbäume für jede User Story erstellt. Durch die Einbindung von NLP wird der Bezug von Anforderungen zum Design und die Generierung der Ergebnisse in Unified Modeling Language erleichtert. Die Anwendung von NLP im agilen RE lässt sich zudem in [Na20] finden. Ebenso werden dort die Spezifikationen von User Stories in ein Unified Modeling Language Diagramm umgewandelt.

In [Zh10] wird ein Algorithmus basierend auf ML in der Anforderungsspezifikation verwendet. Dieser Algorithmus kann Handlungsmodelle mit Quantoren und logischen Implikationen erlernen. Ausgewählte Formeln werden dabei in gelernte Handlungsmodelle umgewandelt, welche sich durch Experten optimieren lassen. Der Nutzen besteht darin, den manuellen Aufwand zu verringern.

Um die Eigenschaften und Bedeutung eines Softwaresystems zu bewerten und festzuhalten, lassen sich Ontologien zur Dokumentation eines Qualitätsschemas einsetzen. Dadurch sollen Entwickler Qualitätsaspekte leichter verstehen und messen können. Ontologie und Semantik tragen dazu bei, den Inhalt eines Softwaresystems darzustellen und die Qualitätsattribute mit den Metriken zu verknüpfen [BGL17].

### 3.4 Anforderungvalidierung

Der kombinierte Einsatz von Modellprüfung und logikbasiertem Lernen soll nach [A115] die Validierung, Diagnose und Reparatur von Anforderungsspezifikationen automatisiert unterstützen. Dadurch wird der manuelle Aufwand reduziert, sodass eine robustere Software entwickelt werden kann. Mittels Modellprüfung sollen Fehler automatisch in der formalen Beschreibung von Anforderungen erkannt und ein Gegenbeispiel erzeugt werden, welches den Fehler veranschaulicht. Da dieses Gegenbeispiel nicht erschöpfend ist, können weitere Beispiele von den Softwareentwicklern manuell ermittelt oder durch eine weitere automatisierte Analyse generiert werden. Nachdem die Gegenbeispiele erfasst sind, führt ein Algorithmus mittels logikbasierten Lernens automatisch Diagnose- und Reparaturaufgaben durch, woraus korrekt überarbeitete Beschreibungen resultieren.

Die Anforderungvalidierung wird ebenso durch NLP unterstützt. In [Mo16] wird ein automatisiertes kollaboratives Tool basierend auf NLP vorgestellt, welches die Zusammenarbeit zwischen Stakeholder und Requirements Engineers unterstützt. Durch das Tool, das im webbasierten Texteditor Etherpad integriert ist, lassen sich Anforderungen virtuell in Echtzeit validieren und kollaborativ bearbeiten. Darüber hinaus werden Requirements Engineers durch den Einsatz von Vorlagen darin unterstützt, vollständige und korrekte Testanforderungen zu verfassen. Diese werden mit Test Cases aus einer Bibliothek mit Testmustern überprüft. Die Verarbeitung natürlicher Sprache sichert die Korrektheit und Genauigkeit der Testanforderungen und Test Cases.

## 4 Implikation

Auf Basis der Ergebnisse dieser SLR wird gezeigt, wie sich der Forschungsstand zum Einsatz von KI-Methoden in den einzelnen Phasen des RE gestaltet. Die Phasen der Anforderungserhebung, -analyse und -spezifikation werden durch unterschiedliche KI-Methoden unterstützt. In der Anforderungsanalyse treten NLP und ML dominant auf. Die Phase der Anforderungsspezifikation ist insbesondere durch den Einsatz von NLP geprägt. Darüber hinaus kommen in geringem Maße ML und Ontologien zur Anwendung. Grundsätzlich können im Rahmen der SLR lediglich zwei Publikationen der Anforderungvalidierung zugeordnet werden [A115], [Mo16]. Demnach ist die Phase der Anforderungvalidierung vergleichbar geringfügig in der Literatur behandelt. Um weitere Aussagen diesbezüglich treffen zu können, sollte diese Phase zukünftig stärker in den Fokus von Forschungsarbeiten, insbesondere auf normativer Ebene, gerückt werden.

Im Rahmen der SLR zeichnen sich Techniken des NLP als häufig verwendete Methode ab, die in allen RE-Phasen Anwendung findet. Daraus lässt sich schließen, dass NLP in den speziellen Aufgaben im Anforderungsprozess der Softwareentwicklung vielfältig einsetzbar ist. Folglich ergibt sich ein hohes Potenzial zur automatisierten Sprachverarbeitung von Anforderungen in Textform. Allerdings zeigen sich hier Schwierigkeiten, die es bei der Integration von KI im RE zu bewältigen gibt. Die fehlerhafte und mehrdeutige Interpretationsweise der natürlichen Sprache ist Gegenstand der Forschungen. Mit Hilfe von NLU wird insbesondere die Verringerung von Redundanz, Mehrdeutigkeit, Inkonsistenz und Unvollständigkeit in Anforderungen erreicht. Die Errungenschaften in der Sprachverarbeitung können zur korrekten Extraktion der Anforderungen von Stakeholdern in verschiedenen Kontexten beitragen. Allerdings ist die automatische und fehlerfreie Extraktion von Anforderungen aus natürlichsprachlichen Beschreibungen bisher nicht vollständig gelungen. Es ist zu erwarten, dass NLP weiterentwickelt und nahezu alle Sprachen weltweit abdecken wird, wodurch ein großes Potential in der vollautomatischen Verarbeitung von natürlicher Sprache besteht. Diesbezüglich bedarf es Verbesserungen der Interpretationsfähigkeit von Worten und Phrasen, die dem natürlichsprachlichen Ursprung entstammen, um eine zuverlässige Verarbeitung der Anforderung von Stakeholdern zu gewährleisten [DS11], [Ma19], [Mo16], [MX19], [Ng12], [Sa12].

Nach aktuellem Stand der wissenschaftlichen Literatur herrscht kein Konsens darüber, welche KI-Methode für welche Aufgaben im Anforderungsprozess genutzt wird [Ba20], [SP13]. So dienen in der Anforderungsanalyse NLP oder ML mehrfach zur Überprüfung der Qualität der erhobenen Anforderungen [Ca14], [Ka19], [Lu17], [Pa15]. Die SLR deutet jedoch ebenso darauf hin, dass für ähnliche Aufgabenstellungen im RE dieselbe KI-Methode angewendet wird. In der Anforderungsspezifikation trägt die Kombination von NLP mit Semantik zur Extraktion von Aktionen und Akteuren und zur Erstellung von Use Case Diagrammen bei [IAA18], [JIA18]. Darüber hinaus kann NLP im Rahmen von agilen Methoden User Stories automatisiert in Use Case Diagramme umwandeln [ENT18], [Na20]. Wie die Beispiele zeigen, könnte ein gesamtheitliches Framework zum Einsatz von KI-Methoden bei speziellen Aufgaben im Anforderungsprozess förderlich sein, um Stakeholder und Requirements Engineers adäquat zu unterstützen. Diesbezüglich ist weitere Forschung notwendig, um fundierte Aussagen darüber zu treffen, welche KI-Methoden in den vorgestellten RE-Phasen bereits möglich sind. So können weitere SLR das breite Spektrum der Forschung näher eingrenzen und zukünftigen Forschungsbedarf noch präziser ableiten. Außerdem sollten Gründe für die unterschiedlich ausgeprägte Nutzung von KI in den RE-Phasen weiter ausgearbeitet werden. Ferner ist zu berücksichtigen, dass der SLR ein breiter KI-Begriff zugrunde liegt. Dies hat einerseits den Effekt, dass das Thema weitläufig erfasst wird. Andererseits bietet es zukünftig die Möglichkeit, einzelne KI-Methoden oder RE-Phasen spezifischer zu betrachten.

Seit 1950 stellt KI ein Forschungsgebiet dar, welches sich stetig weiterentwickelt und neue Erkenntnisse hervorbringt. Die weitere Erforschung des Einsatzes von KI im RE kann auch zukünftig die komplexen Prozesse vereinfachen und die neuen Erkenntnisse im Bereich KI nutzen [Bu20]. Durch diese Forschungsarbeit wird ein Beitrag geleistet, um den

unübersichtlichen Forschungsstand zu strukturieren und diesbezüglich eine Ausgangsbasis für weitere Forschungsaktivitäten zu schaffen. Darüber hinaus werden Einsatzmöglichkeiten von verschiedenen KI-Methoden in den einzelnen Phasen des RE aufgezeigt. Dadurch trägt diese Forschungsarbeit zu einem KI-Methoden übergreifenden Überblick bei, was die Bildung einer Forschungsagenda unterstützt.

## 5 Literaturverzeichnis

- [Ac20] Acatech, <https://www.plattform-lernende-systeme.de/infografiken.html>, Stand: 19.06.2020.
- [Al15] Alrajeh, D. et.al.: Automated Support for Diagnosis and Repair. *Communications of the ACM* 02/15, S. 65-72, 2015.
- [As15] Association for Computing Machinery Digital Library, <https://dl.acm.org/search/advanced>, Stand: 05.06.2020.
- [Ba20] Barenkamp, M.: Künstliche Intelligenz in der Softwareentwicklung. *Wirtschaftsinformatik & Management* 12/20, S.120-129, 2020.
- [BF14] Bourque, P.; Fairley, R. E.: SWEBOK V3.0: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society Press 01/14, S.1-335, 2014.
- [BGL17] Blas, M. J.; Gonnet, S.; Leone, H.: An ontology to document a quality scheme specification of a software product. *Expert Systems* 05/17, 21 Seiten, 2017.
- [BMS20] Bhatia, K.; Mishra, S.; Sharma, A.: Clustering Glossary Terms Extracted from Large-Sized Software Requirements using FastText. In (ACM Hrsg.): Proc.13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference, S.1-11, 2020.
- [BS19] Buxmann, P.; Schmidt, H.: Künstliche Intelligenz: Mit Algorithmen zum wirtschaftlichen Erfolg. Springer, Berlin, 2019.
- [Bu20] Bundesregierung, <https://www.bundesregierung.de/breg-de/aktuelles/meilensteine-der-entwicklung-kuenstlicher-intelligenz-1708150>, Stand: 31.01.2020.
- [Ca14] Casagrande, E. et.al.: NLP-KAOS for Systems Goal Elicitation: Smart Metering System Case Study. *IEEE Transactions on Software Engineering* 10/14, S. 941-956, 2014.
- [DD10] Dube, R. R.; Dixit, S. K.: Process-oriented complete requirement engineering cycle for generic projects. In (ACM Hrsg.): Proc. 5th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, S. 194-197, 2010.
- [De15] Dermeval, D. et.al.: Applications of ontologies in requirements engineering: A systematic review of the literature. *Requirements Engineering* 04/16, S. 405-437, 2015.
- [DR20] Dwitam, F.; Rusli, A.: User stories collection via interactive chatbot to support requirements gathering. *TELKOMNIKA Telecommunication, Computing, Electronics and Control* 02/20, S. 890-898, 2020.

- [DS11] Deeptimahanti, D. K.; Sanyal, R.: Semi-automatic generation of UML models from natural language requirements. In (ACM Hrsg.): Proc. 4th India Software Engineering Conference, S. 165-174, 2011.
- [Eb19] Ebert, C.: Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten. dpunkt.verlag, 6.Auflage, Heidelberg, 2019.
- [EB20] EBSCO Industries, [https://connect.ebsco.com/s/article/What-Field-Codes-are-available-when-searching-EBSCO-Discovery-Service-EDS?language=en\\_US](https://connect.ebsco.com/s/article/What-Field-Codes-are-available-when-searching-EBSCO-Discovery-Service-EDS?language=en_US), Stand: 19.06.2020.
- [El20] Elsevier, <https://www.elsevier.com/solutions/sciencedirect>, Stand: 10.06.2020.
- [ENT18] Elallaoui, M.; Nafil, K.; Touahni, R.: Automatic transformation of user stories into UML use case diagrams using NLP techniques. *Procedia computer science* 04/18, S. 42-49, 2018.
- [Er08] Ertel, W.: *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*. Springer Vieweg, Wiesbaden, 2008.
- [GAS17] Guzman, E.; Alkadhi, R.; Seyff, N.: An exploratory study of Twitter messages about software applications. *Requirements Engineering* 03/17, S. 387-412, 2017.
- [GG18] Gupta, A.; Gupta, C.: CDBR: A semi-automated collaborative execute-before-after dependency-based requirement prioritization approach. *Journal of King Saud University - Computer and Information Sciences*, wird veröffentlicht, 12 Seiten, 2018.
- [HK16] Husain, M. S.; Khanum, M. A.: Word Sense Disambiguation in Software Requirement Specifications Using WordNet and Association Mining Rule. In (ACM Hrsg.): Proc. 2nd International Conference on Information and Communication, S. 1-4, 2016.
- [IAA18] Imam, A. T.; Al-Hroob, A.; Al-Heisa, R.: The Use of Artificial Neural Networks for Extracting Actions and Actors from Requirements Document. *Information and Software Technology* 09/18, S. 1-15, 2018.
- [IE90] IEEE Standards Coordinating Committee: IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). IEEE Computer Society 12/90, S. 1-84, 1990.
- [JIA18] Jebri, E. M.; Imam, A. T.; Al-Fayuomi, M.: An algorithmic approach to extract actions and actors (AAEAA). In (ACM Hrsg.): Proc. of the International Conference on Geoinformatics and Data Analysis, S. 13-17, 2018.
- [JN14] Jurkiewicz, J.; Nawrocki, J.: Automated events identification in use cases. *Information and Software Technology* 02/15, S. 110-122, 2018.
- [Ka19] Kang, Y. et.al.: A transfer learning algorithm for automatic requirement model generation. *Journal of Intelligent & Fuzzy Systems* 02/19, S. 1183-1191, 2019.
- [Ki09] Kitchenham, B. et.al.: Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 01/09, S. 7-15, 2009.
- [Lu17] Lucassen, G. et.al.: Extracting conceptual models from user stories with Visual Narrator. *Requirements Engineering* 03/17, S. 339-358, 2017.

- [Ma16] Mainzer, K.: *Künstliche Intelligenz - Wann übernehmen die Maschinen?*. Springer, Berlin und Heidelberg, 2016.
- [Ma19] Madala, K.: An artificial intelligence-based model-driven approach for exposing off-nominal behaviors. In (IEEE Press Hrsg.): *Proc. 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings*, S. 214-217, 2019.
- [Mo16] Mokerar, N. A. et.al.: An automated collaborative requirements engineering tool for better validation of requirements. In (ACM Hrsg.): *Proc. 31st IEEE/ACM International Conference on Automated Software Engineering*, S. 864-869, 2016.
- [MSP16] Misra, J.; Sengupta, S.; Podder, S.: Topic cohesion preserving requirements clustering. In (ACM Hrsg.): *2016 IEEE/ACM 5th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)*, S. 22-28, 2016.
- [MX19] Memon, K. A.; Xiaoling, X.: Deciphering and Analyzing Software Requirements employing the techniques of Natural Language Processing. In (ACM Hrsg.): *Proc. 4th International Conference on Mathematics and Artificial Intelligence*, S. 153-156. 2019.
- [Na20] Nasiri, A. et.al.: Towards a Generation of Class Diagram from User Stories in Agile Methods. *Procedia Computer Science* 04/20, S. 831-837, 2020.
- [Ng12] Nguyen, T. H. et.al.: REInDetector: a framework for knowledge-based requirements engineering. In (ACM Hrsg.): *Proc. 27th IEEE/ACM international conference on automated software engineering*, S. 386-389, 2012.
- [Pa10] Patsch, H.: *Requirements-Engineering systematisch*. Springer, 2. Auflage, Berlin und Heidelberg, 2010.
- [Pa15] Parra, E. et.al.: A methodology for the classification of quality of requirements using machine learning techniques. *Information and Software Technology* 11/15, S. 180-195, 2015.
- [Ro19] Rodrigues, A. et.al.: Enhancing context specifications for dependable adaptive systems: A data mining approach. *Information and Software Technology* 08/19, S. 115-131, 2019.
- [Sa12] Sateli, B. et. al.: ReqWiki: a semantic system for collaborative software requirements engineering. In (ACM Hrsg.): *Proc. 8th Annual International Symposium on Wikis and Open Collaboration*, S. 1-4, 2012.
- [SP13] Sharma, S.; Pandey, S. K.: Integrating AI techniques in requirements phase: a literature review. In (Institution of Engineering and Technology Hrsg.): *Proc. 4th International IT Summit Confluence 2013*, S.21-25, 2013.
- [Vi10] Villata, S.: A normative multiagent approach to requirements engineering. *Logic Journal of the IGPL* 01/10, S. 245-274, 2010.
- [We12] Wen, K. et.al.: A model based transformation paradigm for cross-language collaborations. *Advanced Engineering Informatics* 01/13, S. 27-37, 2012.
- [Zh10] Zhuo, H. H. et.al.: Learning complex action models with quantifiers and logical implications. *Artificial Intelligence* 12/10, S. 1540-1569, 2010.