

Bestimmung der semantischen Funktion von Sätzen in Anforderungsbeschreibungen

Dokumentenart: Exposé für eine Bachelorarbeit
Autor: Dana Tomova
Matrikel-Nr.: 1960083
Studiengang: Informatik Bachelor
Betreuer: Tobias Hey
Datum: 28. August 2019

1 Motivation

Die Rückverfolgbarkeit von Anforderungen ist eine große Herausforderung im Bereich der Softwaretechnik, die Aufgaben wie Wirkungsanalyse und Wiederverwendung von Teilen eines Systems unterstützt. Die Generierung von Informationen zur Rückverfolgbarkeit ist mit einem kostspieligen manuellen Aufwand verbunden. Diese Generierung zu automatisieren und insbesondere eine Verknüpfung von Anforderungen und Quelltext zu erzeugen würde Zeit und Kosten sparen. Um diese Abbildungen vollständig herzustellen, benötigt es ein tiefgehendes Verständnis der Semantik der Anforderungen.

Funktionale Anforderungen können verschiedene Informationen tragen. Abhängig von der Bedeutung der Sätze können sie unterschiedlich gehandhabt werden. Daher ist es sinnvoll, Anforderungen nach ihrer Semantik zu klassifizieren. Um einen Satz semantisch zu erfassen, ist es wichtig, ein Verständnis der beschriebenen Tätigkeiten und Zustände zu erlangen. In diesem Zusammenhang identifizierte Glinz [Gli05] bereits eine operative [1] und deklarative [2] Einordnung der Anforderungen:

„The system shall compute the sum of all applicable deductions.“ [1]

„The user must provide accurate data for all input fields of the form.“ [2]

Die operativen Anforderungen beziehen sich auf das Verhalten des Systems, während die deklarativen bestimmte Eigenschaften beschreiben. Darüber hinaus können Anforderungen unter anderem auszuführende Aktionen, benötigte und bereitzustellende Daten, Zustände oder irrelevante Informationen umfassen. Es gibt Anforderungen, in denen mehrere dieser Funktionen vertreten sein können. Sie lassen sich in sinnvolle Teilanforderungen zerlegen.

Das Bestimmen der semantischen Funktion der (Teil-)Sätze würde bei der Identifikation möglicher Methoden, Parameter und Attribute im Quelltext helfen. Die Erkennung solcher Unterschiede im Kontext der semantischen Darstellung der Anforderungsbeschreibungen erleichtert die Gewinnung von Informationen zur Rückverfolgbarkeit.

2 Projekt INDIRECT

Das Projekt INDIRECT [Hey19] wird entwickelt, um eine automatische Generierung von Informationen zur Rückverfolgbarkeit von Quelltext und Anforderungen zu ermöglichen. Zu diesem Zweck werden gleichzeitig zwei Absichtsmodelle erstellt - eins für die Anforderungen und eins für den Quelltext. Sie modellieren die Absichten der Interessengruppen und Entwickler unabhängig voneinander. Beide Modelle enthalten Wissen über zugrunde liegende Konzepte und die Beziehungen zwischen ihnen. Nach ihrer Generierung wird eine Zuordnung zwischen beiden Absichtsmodellen gelernt, anstatt direkt zwischen den ursprünglichen Artefakten abzubilden. Diese Zuordnung wird genutzt, um die Rückverfolgbarkeitsverknüpfungen zu erstellen.

Die Verarbeitung der Texteingaben basiert auf der Rahmenarchitektur PARSE [WHT17]. Mithilfe von Agenten wird eine semantische Repräsentation der Anforderungen erstellt. Dadurch kann INDIRECT bereits die einzelnen Entitäten, ihre semantischen Rollen und die Konzepte, zu denen sie gehören, bestimmen.

3 Zielsetzung

Das Ziel dieser Bachelorarbeit besteht darin, die Teile der Anforderungen automatisiert nach ihrer Semantik zu klassifizieren. Zuerst sollen die möglichen Klassen identifiziert werden. Basierend darauf soll ein passender Ansatz gewählt werden, um das Verfahren zur Klassifikation zu entwickeln.

Wie in der Motivation schon erwähnt, liegt der Fokus auf funktionalen Anforderungen. Manche Aussagen beschreiben Aktivitäten und bestimmen mögliche Methoden im Quelltext. Andere liefern Angaben über Systemeigenschaften, die als Zustände kategorisiert werden können. Allerdings gibt es auch Anforderungen, die mehr als eine Verbphrase beinhalten und mehreren Klassen zugeordnet werden können. Ein Beispiel dafür wird in Abbildung 1 dargestellt. Hier gibt es eine Zustandsbeschreibung [1], einen Zustandswechsel [2] und eine Aktionsbeschreibung [3]. Daher ist es ebenso Teil der Aufgabe die Bereiche eines Satzes zu bestimmen, die einer gewissen Klasse zugeordnet werden können.

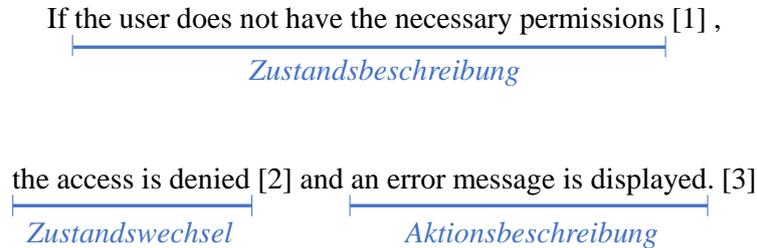


Abbildung 1: Beispiel für komplexe Anforderung

4 Vorgehensweise

Um die aufgestellten Ziele zu erreichen, soll zuerst untersucht werden, welche möglichen Klassen es für Arten von Anforderungssätzen gibt. Zu diesem Zweck sollen verschiedene Datensätze untersucht werden. Hierbei wird ihre Struktur betrachtet und es werden typische Arten von funktionalen Anforderungen abgegrenzt.

Des Weiteren soll noch entschieden werden, ob eine multi-label (any-of) oder eine multinomiale (one-of) Klassifikation [JM18] verwendet wird. Manche Anforderungen bestehen aus mehreren verknüpften Aussagen. Die Sätze lassen sich entweder mehreren Klassen zuordnen, oder können noch vor der Klassifikation in Teilanforderungen zerlegt werden. Im ersten Fall muss eine Multi-label Klassifikation erstellt werden, da die Teilsätze unterschiedliche semantische Merkmale haben können und eine Zuweisung von mehreren Bezeichnungen erlaubt ist. Dagegen kann im zweiten Fall auch eine multinomiale Klassifikation umgesetzt werden, bei der sich die Klassen gegenseitig ausschließen und jeder Teilsatz zu genau einer Klasse gehört. Diese Entscheidung hängt von der Wahl der Klassen ab und hat großen Einfluss auf das weitere Vorgehen.

Als nächstes soll ein Verfahren für die Klassifikation der Anforderungsbeschreibungen ausgewählt und entworfen werden. Ein möglicher Ansatz für die Umsetzung der Klassifikation ist der Einsatz von maschinellem Lernen. Hierbei werden die Modelle darauf trainiert, Aussagen den jeweiligen Klassen zuzuordnen. Maschinelles Lernen kann im Kontext der Aufgabenstellung verwendet werden, da es die Erkennung von unterscheidenden semantischen Merkmalen der Anforderungen ermöglicht. Durch Bewertung ihrer lexikalischen und syntaktischen Eigenschaften werden die entsprechenden Klassen vorhergesagt. Dieses Verfahren bietet viele Vorteile, wie Lernfähigkeit und Möglichkeiten zur Optimierung und Anpassung an veränderte Randbedingungen. Die Verfügbarkeit von genügend Trainingsdaten ist dennoch nötig,

da der Mangel an großen Datenmengen zu ungenauen Ergebnissen führt.

Ein weiterer Ansatz, der in Betracht kommt, sind regelbasierte Systeme. Sie umfassen handgeschriebene Regeln und sind daher sehr präzise und nachvollziehbar. Ein regelbasiertes Verfahren wird zum Beispiel von Vlas und Robinson [VR11] zur Ermittlung und Klassifikation von Anforderungen in Open-Source-Projekten entwickelt. Der Nachteil bei diesem Ansatz ist, dass die Erstellung von Regeln sehr aufwändig sein kann.

Einen Einfluss auf die Entscheidung zwischen diesen beiden Vorgehensweisen haben einerseits die gefundenen Klassen und andererseits die Verfügbarkeit von Trainingssätzen von Anforderungen. Durch Einsatz von maschinellem Lernen werden manchmal Muster identifiziert, die für den Menschen möglicherweise nicht erkennbar sind. Bei wenig Trainingsdaten wäre aber ein regelbasiertes Verfahren effizienter.

5 Evaluation

Für die Evaluation des Werkzeugs werden Musterlösungen für mehrere Datensätze von Anforderungen erstellt. Es werden frei zugängliche Datensätze, wie [JMLP07] und [ZNZ18], verwendet, die für akademische Zwecke erzeugt wurden. Sie umfassen einige hundert Anforderungen, von denen nur die funktionalen berücksichtigt werden. Dann wird die Wahrscheinlichkeit beobachtet, mit der der Agent die korrekten Klassen der Teilanforderungen vorhersagt. Hierzu werden Präzision, Ausbeute und F-Maß [JM18] verwendet.

Literatur

- [Gli05] GLINZ, Martin: Rethinking the Notion of Non-Functional Requirements. In: *in Proceedings of the Third World Congress for Software Quality (3WCSQ'05, 2005*
- [Hey19] HEY, Tobias: Intent-driven Requirements-to-Code Traceability. In: *ICSE'19 DS: Doctoral Symposium of the 2019 IEEE/ACM 41th International Conference on Software Engineering, 2019, S. 4*
- [JM18] JURAFSKY, Daniel ; MARTIN, James H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2018 <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>
- [JMLP07] JANE CLELAND-HUANG ; MAZROUEE, Sepideh ; LIGUO, Huang ; PORT, Dan: *Nfr*
- [VR11] VLAS, R. ; ROBINSON, W. N.: A Rule-Based Natural Language Technique for Requirements Discovery and Classification in

Open-Source Software Development Projects. In: *2011 44th Hawaii International Conference on System Sciences*, 2011, S. 1–10

[WHT17] WEIGELT, Sebastian ; HEY, Tobias ; TICHY, Walter F.: Context Model Acquisition from Spoken Utterances. In: *The 29th International Conference on Software Engineering & Knowledge Engineering*,. Pittsburgh, PA, Juli 2017, S. 201–206

[ZNZ18] ZAIN SHAUKAT ; NASEEM, Rashid ; ZUBAIR, Muhammad: *Software Requirement Risk Prediction Dataset*