Natural Language Processing und RE in der Praxis – Erfahrungen aus unterschiedlichen Anwendungskontexten

David Morais Ferreira, Patric van Zwamen, Max Schmitt, Martin Becker
Fraunhofer-Institut für Experimentelles Software Engineering IESE, Kaiserslautern
{david.morais, patric.zwamen, max.schmitt, martin.becker}@iese.fraunhofer.de

Motivation

In den frühen Phasen von Engineering Projekten müssen viele Informationen aus unterschiedlichsten Quellen erfasst, inspiziert und im aktuellen Projektkontext bewertet werden. Zumeist handelt es sich hierbei um Spezifikationen, Use-Cases, Benutzerdokumentation sowie sonstige Dokumente in natürlicher Sprache. Diese können entweder frei oder mit Hilfe von Satzschablonen formuliert sein. Die besondere Herausforderung besteht darin, relevante Informationen aus den frei formulierten Texten möglichst effizient zu extrahieren und zu interpretieren. Im Zuge der Digitalisierung gewinnt darüber hinaus der Aufbau von expliziten Domänenmodellen massiv an Bedeutung. Dies geht auch mit einer Zunahme an firmeninternen Artefakten und Verknüpfungen zwischen diesen einher [1]. Die erforderlichen Inspektions-Arbeiten an den Dokumenten wurden bisher von Experten durchgeführt, die sowohl Erfahrung im Projektmanagement, RE und in der Domäne haben und damit für die Firmen relativ teuer sind [2].

Da die Verarbeitung von natürlicher Sprache in den letzten Jahren enorme Fortschritte gemacht hat, bietet sich bei diesen Aktivitäten und Informationsextraktionen die Möglichkeit, den Automatisierungsgrad zu erhöhen und somit die Experten zu unterstützen und zu entlasten. Funktioniert diese automatisierte Extraktion, können sich die Experten auf die Bewertung und Revision der extrahierten Informationen konzentrieren und müssen nicht erst deren Erhebung durchführen. Somit muss nicht jeder Satz manuell einzeln erfasst und in Relation zu den bereits erhobenen Informationen gesetzt werden. Dieses Vorgehen bietet nicht nur in Software und Systems Engineering Projekten Vorteile, sondern auch in unterschiedlichsten Anwendungsdomänen. Naheliegenderweise ist die Verwendung von NLP-gestützten Verfahren zur Verarbeitung von textuellen Artefakten ein aktuelles und aufstrebendes Forschungsgebiet.

Projektkontexte und Anwendungsfälle

In unterschiedlichen Projektkontexten sind wir bislang auf folgende Fragestellungen und Anwendungsfälle rund um NLP in RE gestoßen:

AF1 Qualitätsprüfung von Anforderungen. Eine Firma will die Gesamtqualität des Satzes von Anforderungen verbessern. Dazu ist es nötig zu wissen, welche Defekte und mögliches Verbesserungspotential in den Anforderungen steckt sowie deren Verteilung falls diese von mehreren Autoren verfasst wurden. Beispiele hierfür sind die Erkennung von Passivkonstruktionen sowie die Atomarität von Anforderungen.

AF2 Aufbau von Domänenmodellen. Eine Firma will in Zukunft ein Modellbasiertes Systems Engineering (MBSE) betreiben. Hierfür muss das informelle Wissen über Funktionen, Schnittstellen, Systembestandteile etc. aus vorliegenden Engineering-Dokumenten extrahiert werden. Der werkzeuggestützte Aufbau entsprechender Domänenmodellen würde die Domänenexperten entlasten und die MBSE-Einführung deutlich vereinfachen.

AF3 Aufbau von Wiederverwendungsdatenbanken. Eine Firma entwickelt maßgeschneiderte Lösungen in Kundenprojekten in einer Anwendungsdomäne. Die Anforderungen ähneln sich dabei stark. Daher möchte man eine Datenbank mit wiederverwendbaren Anforderungen aufbauen und die Projekte darauf abbilden, um neue Anforderungen zu identifizieren.

AF4 Feature Extraktion. Analog zu AF3 möchte eine Firma ein Feature-Modell aufbauen, um die Wiederverwendung von Engineering-Artefakten, z.B. Anforderungen, Design, Code oder Testfällen, zentral steuern zu können. Neben einer Hierarchie von Merkmalen, sollen auch Abhängigkeiten zwischen diesen erfasst werden.

Herausforderungen bei der NLP-Nutzung

Bei der Nutzung von NLP-Ansätzen in obigen Anforderungsfällen sind wir auf folgende Herausforderungen gestoßen. Im Einzelnen sind einzelne NLP-Methoden meist einfach anzuwenden, jedoch stellt die *Verkettung bzw. Automatisierung einzelner Methoden* eine große Herausforderung dar. Aufgrund der vielen Konfigurationsmöglichkeiten der Bausteine und deren wechselseitigen Auswirkungen auf andere Verarbeitungsschritte verkörpert eine NLP-Verarbeitungskette in ihrer Gesamtheit ein *komplexes System*. Zusätzlich nimmt die Bedeutung der automatisierten Verarbeitung von natürlich sprachlichen Texten stark zu, da die Menge an digitalen Dokumenten stets steigt. Auch wenn diese Dokumente eine gewisse Struktur wie z.B. Inhaltsverzeichnis oder Schlagwortverzeichnis enthalten, handelt es sich um unstrukturierten Text. Für den Umgang und die Analyse von solchen Texten existiert noch *keine einheitliche Vorgehensweise*, da die Tools für die verschiedenen Sprachen und jeweiligen Aspekte unterschiedlich aufgebaut und angepasst werden müssen, da u.a. *unterschiedliche Grammatiken* vorliegen. Somit ist die vorhandene Tool-Unterstützung abhängig von der zu untersuchen-

den Sprache. So führen die *grammatikalischen Sonderfälle im Deutschen* zu komplexen Algorithmen zum Erkennen von Satzstrukturen und Informationsextraktion. Eine weitere Herausforderung bei der automatisierten Analyse von Text ist die *Anpassung von Data Mining Ansätzen*, welche auf strukturierten Daten basieren. Diese müssen an natürliche Sprache angepasst werden. Dabei sollten die Data Mining Ansätze zunächst auf ihre *Tauglichkeit im NLP Bereich* hin geprüft werden. Insbesondere, da es für die deutsche und englische Sprache wenig fundierte Erfahrungen gibt.

Eine weitere Herausforderung ist die Aus- und *Belastung der Domänenexperten*, welche die Informationen extrahieren, bewerten und kategorisieren müssen. Ab einer bestimmten Größe ist die manuelle Bewertung der Artefakte nicht mehr zeit- und kosteneffizient [3]. Erfahrungsgemäß haben diese Experten wenig Zeit, um zeitaufwändige Reviews durchzuführen, da sie zumeist stark in viele unterschiedliche Projekte eingebunden sind. Zusätzlich belastet dieser Review-Prozess über längere Zeit die Konzentrationsfähigkeit des Experten und geht oftmals mit einer *steigenden Fehlerquote* einher. Das *subjektive Expertenwissen* beruht auf der Erfahrung des jeweiligen Experten und prägt bei einem manuellen Review die Ergebnisse der einzelnen Bearbeitungsschritte und kann zu Fehleinschätzungen führen. Darüber hinaus sind die Ergebnisse nicht reproduzierbar und unterscheiden sich zwischen den Experten. Nichtsdestotrotz bieten die fachlichen Kompetenzen der Experten unschätzbaren Mehrwert und können erfolgsentscheidend sein. Um den größtmöglichen Nutzen zu generieren sollte das manuelle und automatisierte Vorgehen kombiniert werden, um die Vorteile beider Ansätze zu vereinigen.

Oftmals ist die *Reproduzierbarkeit* von vorgestellten vollautomatisierten Verfahren nicht gegeben, da die konkrete Umsetzung nicht hinreichend dokumentiert ist. So sind entweder Tools, Algorithmen, Konfigurationen oder vollständige Ergebnisse nicht einsehbar.

Verfolgte NLP Lösungsansätze

Aus Platzgründen fokussieren wir uns im Folgenden auf AF2. Zu dessen Unterstützung wurde eine Mapping-Studie durchgeführt, in der Darting [4] fünf mögliche NLP-Verarbeitungsketten identifiziert hat, um relevante Informationen aus natürlich sprachlichen Texten zu extrahieren. Diese wurden zunächst prototypisch implementiert und anhand von Dokumenten aus zwei Projekten im Rahmen einer expertenbasierten Case Study evaluiert.

Um eine möglichst einfache Verarbeitung der Dokumentinhalte in den darauffolgenden Schritten zu ermöglichen, wurden die zu verarbeiteten Dokumente zuerst in Paragraphen aufgespalten. Grundsätzlich wurden zwei Vorgehensarten unterschieden: einerseits die vektorbasierten Ansätze Doc2Vec und Word2Vec, sowie das Bag-of-Words Modell TF-IDF [4]. Diese Ansätze wurden gewählt, da sie die Daten auf unterschiedliche Art und Weise verarbeiten. Zusätzlich wurde bei TF-IDF die Auswirkung von PCA und LSA, zwei Verfahren zur Reduktion von Dimensionalität, untersucht. Um eine möglichst aussagekräftige Darstellung der resultierenden Daten zu identifizieren, wurden verschiedene Sortierungskriterien und Filteralternativen angewendet. Hierzu zählen unter anderem Schablonen basierend auf Mustern von Part-of-Speech Tags oder n-Gram Teilmengenbetrachtung.

Erfahrungen und Ausblick

Wir haben in den verschiedenen Anwendungsfällen die Erfahrung gemacht, dass die verwendeten Ansätze und Verarbeitungsketten von vielen Faktoren abhängig sind. Von zentraler Bedeutung ist dabei die Strukturierung der Daten, ihre Qualität bzw. deren Vorverarbeitung und die Menge der Daten.

Um eine effizientere Verarbeitung sicherzustellen sollten daher die Rohdaten zunächst von nicht-relevanten Elementen bereinigt werden. Eine weitere Erkenntnis ist, dass dies erfahrungsgemäß ein notwendiger Vorverarbeitungsschritt darstellt, dessen Einfluss sich auf die Qualität der Ergebnisse überträgt. Darüber hinaus muss beachtet werden, dass Ansätze, die auf vektorbasierten Modellen beruhen, nur mit einer signifikanten Menge an Daten trainiert werden können. Auf Grund der unterschiedlichen Ausprägungen von Satzstrukturen, Qualität und Vielfalt der Rohdaten, sowie der verwendeten Sprache und weiteren projektspezifischen Faktoren, müssen die Methoden aktuell für jede einzelne Analyse iterativ geprüft und individuell angepasst werden. Momentan gibt es wenig Erfahrung auf dem Gebiet und es wird empfohlen, möglichst viele verschiedene Ansätze sowie Konfigurationen zu evaluieren. Dabei sollte ein Datensatz gewählt werden, für den die gewünschten Ergebnisse bereits vorliegen oder das notwendige Domänenwissen abrufbar ist.

Da unsere Daten hauptsächlich aus einer geringen Domänenanzahl stammen, könnte die Auswertung von Artefakten aus weiteren Domänen tieferliegende Ergebnisse zu Tage führen.

Quellen

[1] Wnuk, K.; Regnell, B.; Berenbach, B.: Scaling Up Requirements Engineering – Exploring the Challenges of Increasing Size and Complexity in Market-Driven Software Development. In Requirements Engineering: Foundation for Software Quality (REFSQ). Springer Berlin Heidelberg, Berlin, Heidelberg, S. 54-59, 2011.

[2] John, I.: Pattern-based Documentation Analysis for Soft-ware Product Lines. In PhD Theses in Experimental Software Engineering. Fraunhofer Verlag, Stuttgart, Band 30, 2010.

[3] Cheng, B. H. C.; Atlee, J. M.: Research Directions in Requirements Engineering. In Future of Software Engineering (FOSE). IEEE Computer Society, Minneapolis, DC, USA, S. 285-303, 2007.

[4] Darting, S.: Concept Extractor – Automatic Extraction of Product Concepts from Different Natural Language Product Artefacts. Masterarbeit, Technische Universität Kaiserslautern, 2018.