

# Kontext- und Koreferenzanalyse für gesprochene Sprache

Dokumentenart: Exposé für eine Masterarbeit  
Autor: Tobias Hey  
Matrikel-Nr.: 1609981  
Studiengang: Informatik Master  
Betreuer: Sebastian Weigelt  
Datum: 21. Januar 2016

## 1 Motivation

In der heutigen Zeit ist unser Alltag in nahezu allen Bereichen von Software durchdrungen, denn sie steuert unsere Geräte vom Wecker bis hin zum Smartphone. Doch nur wenige Personen sind in der Lage Software mithilfe von Programmiersprachen zu erstellen oder zu verstehen. Bedingt durch die fortwährende Automatisierung und den vermehrten Einsatz von Robotern wird diese Fähigkeit allerdings in Zukunft noch mehr an Bedeutung gewinnen. Es liegt also nahe das Programmieren von Software in der dem Menschen bereits eigenen Ausdrucksweise zu ermöglichen: Der natürlichen Sprache.

Allerdings sind die Herausforderungen, welche bei der Umsetzung von natürlicher Sprache in Quelltext entstehen, vielfältig. Die natürliche Sprache lässt viele Formulierungen zu, welche für einen Rechner nicht ohne weiteres eindeutig auflösbar sind. So greift der Mensch bei der Formulierung und auch dem Interpretieren häufig auf implizites Wissen über Zusammenhänge des Gesagten zurück, den sogenannten Kontext. Ein Beispiel hierfür sind Referenzen wie z.B. Anaphern (siehe Abbildung 2 „it“).

Würde man diese Referenzen nicht auflösen, wäre bei vielen Anweisungen nicht klar, auf welche Objekte oder welche Aktionen sie sich beziehen, wodurch es nicht möglich wäre eine korrekte und vollständige Abbildung der Anweisungen in Quelltext zu gewährleisten.

Das menschliche Gehirn löst diese Referenzen indem es auf die semantischen Zusammenhänge und das erlernte Wissen über die strukturellen Gegebenheiten in natürlicher Sprache zurückgreift [Fel13]. Gerade die semantischen Zusammenhänge hängen jedoch stark vom Kontext ab in dem die Konversation stattfindet. Dieser umfasst unter anderem Informationen wie das zuletzt Gesagte, das Thema der Konversation, den Gesprächspartner oder aber auch den Zustand der Umwelt in der man sich gerade befindet.

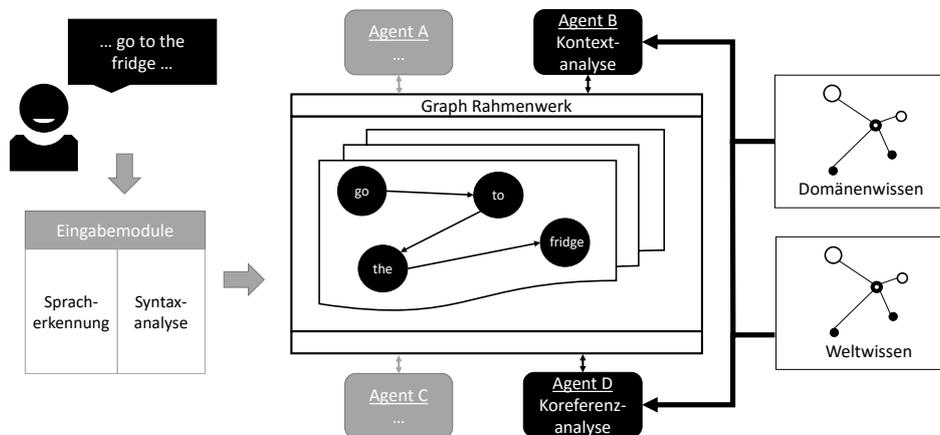


Abbildung 1: Vereinfachte Projekt Architektur (basierend auf [WT15])

Möchte man also eine vollständige Abbildung der Anweisungen in Quelltext durchführen, ist es nötig auch eine möglichst vollständige Vorstellung des Kontextes in dem die Aussagen getroffen werden, zu besitzen.

## 2 PARSE: Programming ARchitecture for Spoken Explanations

Das Projekt *PARSE* zielt darauf ab, in gesprochener Sprache formulierte Anweisungen und Erklärungen in von einem Zielsystem verständlichen Quelltext umzuwandeln. Dies soll es ermöglichen unterschiedliche Zielsysteme wie z.B. den Haushaltsroboter ARMAR-III [ARA<sup>+</sup>06] mittels natürlicher Sprache Anweisungen zu geben und ihm neue Funktionen beizubringen. Hierzu wird die agentenbasierte Architektur *ProNat* [WT15] (siehe Abbildung 1) eingesetzt, welche es ermöglicht verschiedene Schritte der Semantikerken- nung parallel durchzuführen. Vorgeschaltet sind hierbei die Eingabemodule, die variabel ausgetauscht werden können und dafür Sorgen, dass aus der Eingabe die Anweisungen in textueller Form extrahiert werden [Pas15], mit den Ergebnissen erster Rechnerlinguistik Verarbeitungsschritten, wie der Wortart-Markierung, angereichert werden [Koc15] und letztlich in den als geteilten Datenspeicher agierenden Graph überführt werden [Sch15]. Auf diesem Graph arbeiten daraufhin die Agenten, welche ihre Ergebnisse mittels Graphtransformationen in den Datenspeicher einfließen lassen. Momentan sind Agenten zur Erkennung von Kontrollstrukturen, zur Auflösung von sprachlichen Mehrdeutigkeiten von Wörtern und zur Erkennung von Aktionen in Arbeit.

Außerdem sollen unterschiedliche Domänen als Zielsystem zulässig sein, wes-

halb das Domänenwissen, repräsentiert in Domänen-Ontologien, über eine einheitliche Schnittstelle eingebunden wird. Diese Architektur erlaubt es somit sowohl die Eingabequellen als auch die Zielsysteme auszutauschen und durch den Graph als einheitlichen Datenspeicher die Agenten flexibel und transparent zu gestalten.

### 3 Zielsetzung

Ziel dieser Arbeit ist es den Kontext in den im *PARSE* Projekt vorkommenden natürlich sprachlichen Anweisungen so aufzubereiten, dass das gewonnene Wissen z.B. zur Auflösung von Referenzen genutzt werden kann. Hierzu soll ein weiterer Agent für *ProNat* erstellt werden, welcher die Eingabe auf Kontextinformationen hin analysiert und die gewonnenen Informationen geeignet in die Graphrepräsentation einfließen lässt.

Mit Kontextinformationen sind in diesem Fall Informationen gemeint, welche die Zusammenhänge des Gesagten darstellen. In erster Linie beinhaltet dies die Informationen aus den zuletzt gegebenen Anweisungen, wie die existenten Entitäten, deren Zustände und durchgeführten Aktionen, aber auch allgemeines Wissen über die Zusammenhänge der Entitäten und Aktionen, wie z.B. mögliche Teil-Ganzes Beziehungen (Meronymie) zwischen Entitäten.

Als zweiten Schritt nach dem Aufbau des Kontextwissens soll in dieser Arbeit ein weiterer Agent erstellt werden, der mithilfe der gewonnenen Kontextinformationen die Auflösung von Referenzen durchführt.

### 4 Vorgehen

Um die beschriebenen Kontextinformationen aus den Anweisungen zu extrahieren werden zwei Ansätze kombiniert. Zunächst lassen sich die Ergebnisse der vorherigen Arbeiten im Projekt heranziehen, um eine Art Gedächtnis des zuletzt Gesagten aufbauen zu können. Dies beinhaltet die Erkennung von Entitäten über die Ergebnisse der Komponente zur Syntaxanalyse und dem Agenten zur Aktionserkennung. Hierdurch lassen sich über strukturelle Regeln sowohl die handelnden Entitäten, als auch die möglichen Objekte der Aktionen extrahieren und pro Anweisung speichern.

Des Weiteren lässt sich eine Art Zusammenhangwissen über das Einbinden von Weltwissen und Domänenwissen erzeugen. Hiermit sind Informationen gemeint, die über die syntaktischen Zusammenhänge der Anweisungen hinausgehen. So kann untersucht werden inwieweit Zustände und Zustandsübergänge der Entitäten mithilfe der Informationen über die ausgeführten Aktionen ermittelt werden können. Hierbei könnten Anfragen an die Wissensontologien dabei helfen, mögliche Zustände pro Entität und Zustandsübergänge pro Aktion zu ermitteln. So könnte aus einer *open* Aktion auf dem *dish-*

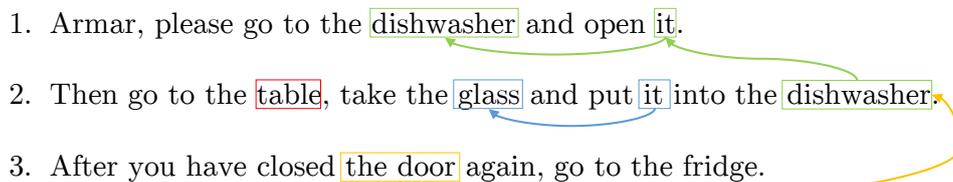
1. Armar, please go to the dishwasher and open it.
  2. Then go to the table, take the glass and put it into the dishwasher.
  3. After you have closed the door again, go to the fridge.
- 

Abbildung 2: Beispiel für natürlichsprachliche Referenzen

*washer* geschlossen werden, dass dieser in den Zustand „Geöffnet“ übergeht und vermutlich ebenso einen komplementären Zustand „Geschlossen“ besitzt, wenn auf ihm auch eine `close` Aktion ausgeführt werden kann.

Eine weitere Komponente des Zusammenhangwissens stellen mögliche Teil-Ganzes Beziehungen zwischen den Entitäten dar. Hierbei werden zu den erkannten Entitäten Meronyme gesucht und daraufhin bewertet, wie wahrscheinlich eine Teil-Ganzes Beziehung zu anderen bereits erkannten Entitäten ist. Mit diesen Informationen können dann folgende Agenten entscheiden, ob sie diese Beziehungen in ihre Betrachtung mit einfließen lassen wollen.

Die erkannten Entitäten und Aktionen werden daraufhin als Knoten in den Graph eingefügt und die zusätzlichen Informationen, wie z.B. die Zustände, als Attribute dieser Knoten modelliert. Außerdem kann jederzeit über Kanten nachverfolgt werden, welche Wörter diese Informationen geliefert haben. Hierdurch können alle folgenden Agenten auf das Kontextwissen über den geteilten Datenspeicher zugreifen.

Ist dann die Repräsentation des Kontextes existent, lassen sich die erhaltenen Informationen nutzen, um die Ergebnisse von Koreferenzanalyse-Werkzeugen zu validieren und weitere Referenzen, wie Referenzen zwischen Aktionen, in den Anweisungen zu finden. Hierzu wird ein geeignetes Koreferenzanalyse-Werkzeug gewählt und ein Agent erstellt, der dessen Ergebnisse in die Graphrepräsentation einfließen lässt und dabei abhängig vom vorhandenen Kontextwissen versucht die Auflösung der Referenzen zu verbessern.

## 5 Beispiel

Das Beispiel in Abbildung 2 beinhaltet verschiedene natürlichsprachliche Referenzen, welche mit Hilfe des Kontextwissens aufgelöst werden können.

Betrachtet man zunächst die ersten beiden Sätze so ergibt sich als extrahierbarer Kontext: Die handelnde Entität ist *Armar* und es existieren die Entitäten *table*, *glass* und *dishwasher*. Außerdem wurden die Aktionen `open(it)`, `take(glass)` und `put_into(dishwasher, it)` ausgeführt. Über den Zustand des *dishwasher* kann noch keine Aussage getroffen werden, da sich die Aktion `open` zunächst auf das Pronomen „*it*“ bezieht und erst nach der

Auflösung durch den Koreferenzanalyse-Agenten eindeutig als sich auf den *dishwasher* beziehend interpretiert werden kann. Hier hilft das agentenbasierte Design der Architektur, dass es ermöglicht die Agenten parallel bzw. wieder aufrufbar auszuführen und je nach neuem Wissensstand die gleichen Anweisungen erneut zu analysieren. So kann, sobald der Koreferenzanalyse-Agent die Referenz aufgelöst hat, der Kontext des *dishwasher* um den Zustand geöffnet erweitert werden.

Im dritten Satz tritt dann eine Referenz auf, die ohne Kontextwissen nicht auflösbar wäre. Hier bezieht sich „*the door*“ auf die Tür des *dishwasher*. In diesem Fall sollte das Kontextwissen einen Knoten für die Entität *door* enthalten, der eine Kante zum Knoten *dishwasher* besitzt, welche angibt, dass es sich hier um eine Form von Meronymie handelt. Außerdem handelt es sich bei der ausgeführten Aktion um eine `close` Aktion, welche auf eine Entität hindeutet, die den Zustand „Geöffnet“ hat oder zumindest mit einer `open` Aktion in Verbindung gebracht werden kann. Diese Informationen helfen der Koreferenzanalyse diese Referenz aufzulösen und stellen ein Beispiel für einen Fall dar, der mit herkömmlichen Lernverfahren basierten Koreferenzanalysen nicht ohne weiteres aufgelöst werden kann.

## 6 Evaluation

Um den in dieser Arbeit gewählten Ansatz zu Evaluieren, können zunächst die in [Gü15] gesammelten Anweisungsbeschreibungen herangezogen werden. Allerdings enthalten diese nur bestimmte Fälle von natürlichsprachlichen Referenzen und eignen sich deshalb nur bedingt für die Evaluation. Daher werden im Rahmen dieser Arbeit weitere Anweisungsbeschreibungen gesammelt, welche den Gebrauch von Referenzen provozieren.

Für die Evaluation der Extraktion des Kontextwissens lässt sich bestimmen, für wie viele der in den Beschreibungen erwähnten Anweisungen Kontextinformationen extrahiert werden konnten und ob die untersuchten Kontextinformationen zuverlässig erkannt wurden. Dies ließe sich mithilfe von Musterlösungen (Gold-Standard) evaluieren, gegen die die erkannten Informationen geprüft werden.

Um daraufhin die Auflösung der Referenzen evaluieren zu können, lassen sich ebenso Musterlösungen heranziehen. Als Bewertungskriterien eignen sich hierbei die Präzision und Ausbeute sowie das F-Maß. Außerdem lässt sich hiermit auch ein Vergleich mit der Variante des Koreferenzanalyse-Werkzeugs ohne Anpassungen und Kontextwissen durchführen und somit evaluieren wie groß das Verbesserungspotenzial durch Kontextwissen ist.

## Literatur

- [ARA<sup>+</sup>06] ASFOUR, T. ; REGENSTEIN, K. ; AZAD, P. ; SCHRODER, J. ; BIERBAUM, A. ; VAHRENKAMP, N. ; DILLMANN, R.: ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, S. 169–175
- [Fel13] FELIX, Sascha W.: *Kognitive Linguistik: Repräsentation und Prozesse*. Springer-Verlag, 2013. – ISBN 978-3-663-05399-6
- [Gü15] GÜNES, Zeynep: *Aufbau eines Sprachkorpus zur Programmierung autonomer Roboter mittels natürlicher Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, Mai 2015
- [Koc15] KOCYBIK, Markus: *Projektion von gesprochener Sprache auf eine Handlungsrepräsentation*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, Juli 2015
- [Pas15] PASKARAN, Dinesh: *Evaluation unterschiedlicher Spracherkennungssysteme in der Domäne Humanoide Robotik*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, November 2015
- [Sch15] SCHNEIDER, Michael: *Entwurf einer Handlungsrepräsentation für gesprochene Sprache*, Karlsruher Institut für Technologie (KIT) – IPD Tichy, Bachelor’s Thesis, Mai 2015
- [WT15] WEIGELT, S. ; TICHY, W.F.: Poster: ProNat: An Agent-Based System Design for Programming in Spoken Natural Language. In: *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on* Bd. 2, 2015, S. 819–820