

## TUM HALL OF SCIENCE AND TECHNOLOGY im Bereich Informatik

# PROBABILISTIC ROBOT ACTION CORES

Aufbereitet von Elena Hubwieser

In diesem Artikel wird ein Überblick über das Konzept Probabilistic Robot Action Cores (PRAC) gegeben. Dieses erleichtert die Kommunikation zwischen Menschen und Robotern. Die Menschen neigen dazu, sich ungenau und doppeldeutig auszudrücken, wenn sie Anweisungen geben, aber Roboter benötigen exakte und unmissverständliche Arbeitsaufträge. PRAC hilft, die menschlich formulierten Aufträge für Roboter verständlich zu machen.



**Abbildung 1** Der Roboter „TUM-Rosie“,  
 übernommen aus [1]

## EINLEITUNG

Roboter gibt es in vielen Bereichen des heutigen Lebens: Vom PKW-Montage-Roboter oder Tiefseetaucher bis hin zum Kosmonauten-Roboter. Leider gibt es noch nicht viele Roboter, die uns die alltäglichsten und mühsamsten Tätigkeiten abnehmen. Bisher können sie nur einfache Aufgaben übernehmen, wie zum Beispiel Rasenmähen, Staubsaugen, eine Waschmaschine Ein- und Ausräumen, Geschirr Abräumen oder Essen in der Mikrowelle Aufwärmen. Sie sind aber zu komplexeren Tätigkeiten wie Kochen, Bügeln, Putzen noch nicht fähig. Woran das liegt und inwieweit PRAC eine Lösung für dieses Problem bietet könnt ihr im Folgenden nachlesen. Dabei stütze ich mich hauptsächlich auf die Forschungsergebnisse von Michael Beetz und Daniel Nyga [1].

## DAS PROBLEM



Mit einer Anweisung sind immer Nebenbedingungen verknüpft, die teilweise sogar mehr aussagen, als die Anweisung selbst. Wenn Menschen sich gegenseitig Anweisungen geben, assoziieren sie damit automatisch diese Nebenbedingungen. Sie sind dazu in der Lage, weil sie die menschliche Kommunikation beherrschen und jahrelange Erfahrung mitbringen. Sie wissen genau, welche Aktionen und Objekte und welche Beziehungen unter ihnen miteinander verknüpft sind. Selbst wenn unser Gegenüber Informationen weglässt, können wir Menschen daher dennoch seine Anweisungen verstehen. Ein Roboter dagegen besitzt dieses umfassende Hintergrundwissen nicht ohne weiteres. Die richtige Interpretation menschlicher Anweisungen muss ihm erst beigebracht werden, damit er in jeder Situation so handeln kann, wie ein Mensch das ganz automatisch tun würde. Wenn ein Roboter menschliche Anweisungen erhält, stößt er auf folgende Probleme:

**Zweideutigkeit.** Dadurch, dass Informationen in Anweisungen weggelassen werden, kann eine Anweisung verschiedene Interpretationsmöglichkeiten haben. Heißt es zum Beispiel „*Den Speck schneiden.*“ in einem Rezept, so kann der Speck entweder in Würfel oder in Scheiben geschnitten werden. Mit der Hintergrundinformation, dass das Ergebnis Rührei mit Speck sein soll, ergibt sich die naheliegende Forderung, dass der Speck in Scheiben zu schneiden ist. Soll er aber Teil einer Nudelsauce werden, so wird der Speck gewürfelt.

**Beendigung einer Aktion.** Selten ist am Ende eines Kochrezeptes aufgeführt, was die abschließende Handlung ist. In einem Kochrezept wie Kartoffelgratin beispielsweise könnte das Ende so aussehen: „*Die Kartoffelmasse in die Form füllen und langsam bei 170 Grad mindestens 30 Minuten backen, dann mit geriebenem Käse (je nach Geschmack) bestreuen und goldbraun überbacken.*“ ([2]) Es wird mit keiner Silbe erwähnt, dass das Gratin nach dem Backen aus dem Ofen genommen werden und der Ofen ausgeschalten werden muss. Für einen geübten Koch ist das selbstverständlich und auch die meisten anderen Menschen denken an diese zusätzliche Aktion. Einem Roboter dagegen muss die zusätzliche Anweisung explizit mitgeteilt werden.

Ein Roboter, der in menschlichen Bereichen agiert, wird mit ungenauen, zweideutigen und unvollständigen Anweisungen konfrontiert. Der Schlüssel zum Verständnis von Alltagsaufgaben liegt im handlungsspezifischen Wissen.

## DIE LÖSUNG



Die Idee hinter der Modellierung von handlungsspezifischem Wissen ist die Verallgemeinerung alltäglicher Ereignisse, so dass sie auf mehrere Fälle übertragbar sind. Dabei soll der so stark wie möglich abstrahiert werden, die Beschreibung soll aber gleichzeitig so spezifisch wie möglich sein. Das bedeutet eine verallgemeinerte Handlung soll auf so viele Situationen wie möglich anwendbar sein, darf aber keine nötigen Informationen verlieren, damit verschiedene spezielle Handlungen nicht verwechselt werden. Menschen führen solche Abstraktionen unbewusst und schnell durch. Sie können Begriffe miteinander assoziieren und so schnell deren Bedeutungen und weitere involvierte Begriffe erschließen. Dieser Vorgang findet bei den Menschen schon in frühester Kindheit statt. So können die Abstraktionsebenen mit über die Jahre gewonnener Erfahrung stetig erweitert werden.

Ein Roboter muss natürlich keine Erfahrung aufbauen, sondern bekommt einen umfassenden Wissensstand übertragen. Dazu gehören auch Assoziationen, die ihm dabei helfen können, Begriffe einzuordnen und zu verstehen. Ein Roboter muss zum Beispiel über die Beziehungen, die mit einem Begriff verbunden sind, Bescheid wissen. Damit sind sowohl Beziehungen zwischen verschiedenen Begriffen als auch Beziehungen innerhalb eines Begriffs gemeint. Eine Sammlung solcher Begriffsbeziehungen nennt man *Robot Action Core*, also den Kern einer von einem Roboter ausgeführten Handlung. Die gesammelten Beziehungen eines *Robot Action Cores* bilden einen abstrakten Handlungstypen. Jede Entität, also jede Informationseinheit, die von einem Handlungstyp betroffen ist, bekommt eine Rolle in der Handlung zugewiesen. Die Handlungsrollen spezifizieren also die Beziehungen zwischen den in eine Handlung involvierten Entitäten. Ein Roboter kann eine Aktion erst erkennen und einordnen, wenn er alle beteiligten Handlungsrollen kennt.

### Idee

Die Idee hinter dem *Probabilistic Robot Action Core (PRAC)* ist, eine Wahrscheinlichkeitsverteilung zu schaffen, um die Rückschlüsse auf die eigentliche Bedeutung menschlicher Anweisungen zu ermöglichen. Mit Hilfe der folgenden Formel kann ein *Action Core* passend abstrahiert werden, so dass er auf viele verschiedene Situationen anwendbar ist:

$$\text{argmax}_{\text{benötigte Rollen}} P_{\text{Handlung}}(\text{benötigte Rollen} | \text{vorgegebene Rollen})$$

$P_{Handlung}$  (benötigte Rollen | vorgegebene Rollen) berechnet die Wahrscheinlichkeit, dass bei einer Handlung unter der Voraussetzung, dass die vorgegebenen Rollen auftreten, die benötigten Rollen auftreten.  $argmax$  berechnet das Maximum über diese Wahrscheinlichkeit bei allen benötigten Rollen. Vorgegebene Rollen beinhalten auch Vererbungsbeziehungen oder Taxonomien (mit  $\sqsubseteq$  dargestellt) und Teil-Ganzes Beziehungen (mit  $\preceq$  dargestellt).

Es genügen schon ein paar Informationen zu einer Handlung, um die Beziehungen zwischen den Objekten zu erkennen und so in der Vererbungshierarchie auf der passenden Ebene eine abstrakte Handlung zu finden, die zu der Situation passt. Folgendes Anweisungsbeispiel dazu ist aus [1] übernommen:

„Fülle<sub>Handlung</sub> Milch<sub>Leitmotiv</sub> in eine Schüssel<sub>Zielort</sub>“  
 und „Fülle<sub>Handlung</sub> ein Glas<sub>Zielort</sub> mit Wasser<sub>Leitmotiv</sub>“

Dabei steht  $R = \{Handlung, Leitmotiv, Zielort\}$  für die verschiedenen Rollen des Handlungsverbs „füllen“. Im Rahmen von PRAC wird zu folgender Anweisung abstrahiert:

„Fülle<sub>Handlung</sub> eine Flüssigkeit<sub>Leitmotiv</sub> in ein Behältnis<sub>Zielort</sub>“

Der Roboter kann mit dieser Abstraktion auch zum Beispiel einen Saft in ein Behältnis füllen.

In einer unbekanntem Situation können Informationen aus der Umgebung dazu genutzt werden, um herauszufinden, welche Informationen fehlen und die möglichen Handlungsstrukturen auf ein kleines Maß einzugrenzen.

### Definition

Aufbauend auf dieser Idee, wird PRAC als eine abhängige Wahrscheinlichkeitsverteilung  $P(R \times A \times C \mid \sqsubseteq; \preceq)$  definiert. Dabei ist  $R$  die Sammlung aller Handlungsrollen,  $A$  die Sammlung aller Aktionsverben,  $C$  die Sammlung aller Klassenkonzepte,  $\sqsubseteq$  die taxonomischen Beziehungen über  $C$  und  $\preceq$  die Teil-Ganzes Beziehungen über  $C$ .

Mit Hilfe dieses Modells können auch zweideutige Anweisungen interpretiert werden. Es wird einfach die wahrscheinlichste der möglichen Lösungen verwendet. Auch fehlende Informationen können damit ergänzt werden. PRAC ermöglicht die Kommunikation zwischen Menschen und Robotern, so dass Roboter Befehle von Menschen verstehen und ausführen können. Im folgenden Beispiel aus [1] ist die Anweisung „Wende den Pfannkuchen“ gegeben mit der Handlung „wenden“ und dem Objekt „Pfannkuchen“. An dieser Stelle wird PRAC benutzt, um das Instrument zu finden, das wahrscheinlich am meisten nützt:

$$\begin{aligned} argmax_{c \in K_{onzepte}} P(i \sqsubseteq c \mid \\ p \sqsubseteq Pfannkuchen, Leitmotiv(a, p), Handlungsverb(a, Wenden), Werkzeug(a, i)) \\ = Pfannenwender \end{aligned}$$

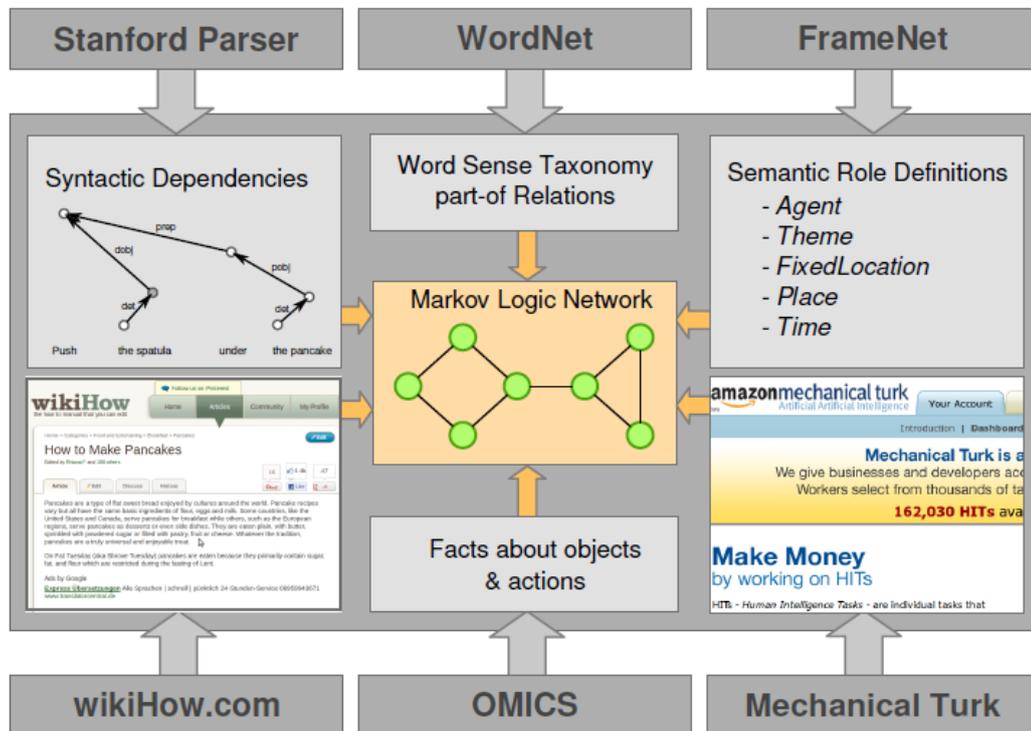
## IMPLEMENTIERUNG

Hier wird die Implementierung von PRAC vorgestellt, die von Nyga und Beetz in [1] beschrieben wird. Das Programm ist so konzipiert, dass es auf menschliche Anweisungen anwendbar ist. Es werden 5 Quellen für menschliche Anweisungen verwendet:

- ✓ Das [WordNet](#), die lexikalische Datenbasis für konzeptuelles und taxonomisches Wissen,
- ✓ das [FrameNet](#), die Datenbasis für Definitionen von Handlungen und Rollen,
- ✓ der [Stanford Parser](#), um syntaktischen Informationen aus Anweisungen von Menschen zu filtern,
- ✓ die Website [wikihow.com](#), für domänenspezifisches Wissen über Handlungen und

- ✓ der Marktplatz von *Amazon Mechanical Turk*, um semantisch beschriftete Grundinformationen zu erhalten.

Die **ersten drei Punkte** enthalten Wissen, dessen Elemente zueinander in Beziehung gesetzt und daher direkt in das Modell PRAC übertragbar sind. Wir nennen diese Quellen **Wissensquellen**. Die **beiden letzten Punkte** betreffen Quellen, aus denen Daten gewonnen werden, die sogenannten **Datenquellen**.



**Abbildung 2:** Architektur der Implementierung von PRAC, übernommen aus [1]

### Wissensquellen:

In *WordNet* werden Wörter mit ähnlicher Bedeutung in sogenannten *Synsets* gruppiert. Synset ist eine Abkürzung für „sets of cognitive synonyms“, was so viel wie Synonymsammlungen heißt. Zwischen diesen Synsets gibt es starke taxonomische und Teil-Ganzes Beziehungen. Das Programm WordNet kann kostenlos aus dem Internet heruntergeladen werden, aber die Dienste können auch online genutzt werden. Man kann in WordNet ein beliebiges englisches Wort eingeben und erhält als Ausgabe eine Liste sämtlicher Synsets, mit denen das Programm das eingegebene Wort verbinden kann. Das Ergebnis sind also sowohl Synonyme als auch mögliche Wortbedeutungen. Diese Ergebnisse entsprechen der Menge C. [3]

*FrameNet* ist eine englischsprachige Plattform, auf der Bedeutungen von Wörtern in semantischen Rahmen gefunden werden können. Man kann auf der Website ein Wort in das Suchfeld eingeben und erhält als Antwort eine Liste verschiedener Variationen des Wortes und einer Information über die jeweilige Wortart jeder Variation, das heißt n für Nomen, v für Verb, usw. Ein Mausklick auf eine Variation öffnet einen Link zu der dazugehörigen Worterklärung. Eine Worterklärung besteht aus den beteiligten Rollen, also den Satzbausteinen, die stets mit dem Wort verbunden sein müssen und einer Umschreibung des Wortes sowie einigen Beispielsätzen. [4]

Nicht nur die Bedeutung der Wörter eines Satzes kann bei der Übersetzung menschlicher Anweisungen helfen, sondern auch der Satzbau des Handlungsauftrags. So kann zum Beispiel eine Präposition den Sinn einer Anweisung entscheiden. Außerdem ist die Anordnung der Wörter in einem Satz wichtig zu beachten.

Solche Informationen können durch einen Parser aus einer menschlichen Anweisung herausgefiltert werden. Für PRAC wird der *Stanford Parser* verwendet. Dieser kann aus dem Internet heruntergeladen werden. Dieser erkennt Abhängigkeiten innerhalb eines Satzes und erstellt dazu Baumdiagramme. So kann vorhergesagt werden, welche Bedeutung kombinierte angewandte Wörter, die an bestimmten Stellen im Satz stehen, einem Satz verleihen. Der Stanford Parser repräsentiert die Menge  $R$  im PRAC-Modell. [5]

## Datenquellen

Auf *wikihow.com* werden kostenlos Fragen zu alltäglichen Problemen öffentlich gestellt und beantwortet. Diese Antworten werden für alle User zugänglich gemacht und zur erleichterten Suche in Kategorien wie „Technik & Elektronik“, „Familienleben“, „Sport & Fitness“ und viele weitere unterteilt. Die Antworten sind stets anleitend gestellt und geben daher genau das Vorgehen zur Lösung des beschriebenen Problems vor. Für PRAC werden hauptsächlich Kochanleitungen benötigt, die zudem in englischer Sprache verfasst sind, daher betrachtet man hier nur die Kategorie „Food & Entertainment“ der englischen Plattform, welche der deutschen Kategorie „Essen & Ausgehen“ entspricht. Außerdem werden nur Anweisungen betrachtet, in denen die Verben „add“ (=hinzufügen), „cut“ (=schneiden), „fill“ (=füllen), „flip“ (=wenden), „mix“ (=mischen), „place“ (=stellen), „pour“ (=eingießen) und „put“ (=legen) vorkommen. Diese Verben sind auch zugleich als die meistverwendeten aus den Analysen der Wissensquellen hervorgegangen. Insgesamt konnten 1400 Anweisungen verwendet werden, die zusätzlich mit semantischen Anmerkungen versehen werden. [6]

Auf dem Marktplatz von *Amazon Mechanical Turk* können Unternehmer Anfragen für Gelegenheitsarbeiten stellen, auf die Arbeiter antworten können. Man findet auf der Plattform bezahlte Gelegenheitsarbeiter, die Arbeiten für Unternehmer durchführen, die nicht von Maschinen erledigt werden können. Diese Aufgaben werden „Human Intelligence Tasks“ (HIT) genannt. [7, 8]

Für PRAC wurde ein öffentlicher Aufruf auf Amazon Mechanical Turk gestartet. Darin werden Arbeitskräfte angeheuert, die die gefundenen menschlichen Anweisungen Wort für Wort mit Anmerkungen versehen haben, in denen die Wortbedeutungen enthalten sind. Die Antwortmöglichkeiten hierfür wurden aus WordNet gewonnen und vorgegeben. Daraufhin wurden Kommentare zu den Handlungsrollen in jeder Anweisung erstellt. So wurde eine große Menge an Informationen gesammelt, die Anmerkungen zu der Semantik von menschlichen Anweisungen enthalten. Auf diese Weise konnte das PRAC-Modell eigearbeitet werden.

## Markov Logic Networks

Das logische System der Implementierung von PRAC basiert auf den *Markov Logic Networks (MLN)*. MLNs verknüpfen First-Order-Logik mit wahrscheinlichkeitstheoretischen, ungerichteten Graphen. Um mehr Informationen darüber zu erhalten folgt dem Link zu [9].

Weitere Informationen zur Verbindung zwischen MLNs und PRAC können in [1] gefunden werden. Außerdem werden dort auch einige Experimente zu PRAC dargestellt.

## LITERATURLISTE

[1] Beetz, M., Nyga, D.: *Everything Robots Always Wanted to Know about Housework (But were afraid to ask)*, In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2012), verfügbar unter: <https://ias.in.tum.de/people/nyga/publications> (letzter Zugriff am 01.09.2013)

[2] Tenesse: *Kartoffelgratin* (2013), verfügbar unter <http://www.kuechengoetter.de/rezepte/Auflauf/Kartoffelgratin-2582882.html> (letzter Zugriff am 01.09.2013)

- [3] Princeton University: *WordNet – A lexical database for English* (2012), verfügbar unter <http://wordnet.princeton.edu/> (letzter Zugriff am 01.09.2013)
- [4] International Computer Science Institute: *FrameNet*, verfügbar unter <https://framenet.icsi.berkeley.edu/fndrupal/home> (letzter Zugriff am 01.09.2013)
- [5] The Stanford National Language Processing Group: *The Stanford Parser: A statistical parser* (2013), verfügbar unter <http://nlp.stanford.edu/software/lex-parser.shtml> (letzter Zugriff am 01.09.2013)
- [6] Herrick, J: *wikiHow – to do anything* (2013), verfügbar unter <http://www.wikihow.com/Main-Page> (letzter Zugriff am 01.09.2013)
- [7] Amazon.com, Inc.: *amazonmechanical turk – Artificial Artificial Intelligence* (2013), verfügbar unter <https://www.mturk.com/mturk/> (letzter Zugriff am 01.09.2013)
- [8] kas: *Mechanical Turk: Amazon-Jobs für Hungerlöhne* (2012), verfügbar unter [http://business.chip.de/news/Mechanical-Turk-Amazon-Jobs-fuer-Hungerloehne\\_53643285.html](http://business.chip.de/news/Mechanical-Turk-Amazon-Jobs-fuer-Hungerloehne_53643285.html) (letzter Zugriff am 01.09.2013)
- [9] Domingos, P., Richardson, M.: *Markov Logic Networks* (2006), erschienen in *Machine Learning* (2006) 62: S.107-136, verfügbar unter <http://link.springer.com/article/10.1007/s10994-006-5833-1#page-1> (letzter Zugriff am 01.09.2013)