UNIVERSITY OF BREMEN

INSTITUTE FOR
ARTIFICIAL INTELLIGENCE

Actionable Knowledge Graphs

How Daily Activity Applications can Benefit from Embodied Web Knowledge

Michaela Kümpel

Vollständiger Abdruck der vom Fachbereich Mathematik und Informatik der Universität Bremen genehmigten Dissertation zur Erlangung des Grades

Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

Vorsitzender:	Prof. Dr. Dieter Hutter *Universität Bremen*
1. Prüfer:	Prof. Dr. hc. Michael Beetz, PhD *Universität Bremen*
2. Prüfer:	Prof. Frank van Harmelen, PhD *Vrije Universiteit Amsterdam*
Beisitzer:	Prof. Dr. Nico Hochgeschwender, *Universität Bremen*

Die Dissertation wurde am 20.12.2023 bei der Universität Bremen eingereicht und durch den Prüfungsausschuss am 19.03.2024 angenommen.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Bremen, den April 12, 2024

Michaela Kümpel

Abstract

Knowledge Graphs have been a research trend. The idea to acquire knowledge from the Web and make it accessible for users to answer queries has resulted in many famous knowledge graphs in industry (e.g. Google, Facebook or Amazon) as well as research (e.g. wikidata, WordNet or FrameNet). Due to their success, one can come to the conclusion that knowledge graphs might also help robots in answering complex queries in the same way that they help humans. In order to access knowledge graphs in robotic applications, one has to initially understand how robot control systems work. A robot can use a range of sensors to perceive its environment and build an internal belief state, a model of the environment, to make sense out of the perceived data. This means that a robot will use one of its sensors (e.g. a camera) to perceive an apple on the table in front of it. In its belief state it will then store the information that there is an apple and a table available in the environment (which are set in relation to the robot by storing exact object positions) but it might add the information that the apple is *on the table*, which more specifically relates the position of the apple to the position of the table as well as the robot itself in its belief state. If the robot is given the task to "cut the apple", the robot will access its belief state to reason about where it might find an apple to then perform the given task by relating the task to the environment information of its belief state. A belief state can also link to a knowledge base with additional knowledge that the robot might need. If we change the given task to "give me the milk" but do not change the environment so that the robot still only detects an apple on a table, the robot would not be able to fulfil this task without additional knowledge. However, if the belief state is linked to a knowledge base that offers the information that milk is a perishable product that usually is stored in a fridge, the robot can access this knowledge base to infer that it must search for a fridge in order to accomplish the given task.

Much of such additional knowledge that is needed to perform tasks is available on the Web. An agent that is enabled to access knowledge graphs containing Web

information can infer that the apple perceived on the table is a pome fruit of a food taxonomy, which has a core that usually is removed before eating. However, in order for a robot to access this information, we have to translate the object knowledge contained in a knowledge graph to environment information perceived by an agent in its belief state. What is more, if we want the robot to know how to hold the apple in order to cut it or how to remove the core of the apple, both the object and environment information needs to also be linked to action information. A knowledge graph that links object to environment and action information *makes the contained knowledge actionable*.

This Thesis proposes a five-step methodology for creating actionable knowledge graphs that follows existing knowledge engineering standards but links object knowledge to environment and action knowledge to enable various applications in daily environments, on different agents. The methodology is exemplary applied in two scenarios with different foci to create a product knowledge graph and a food cutting knowledge graph. The product knowledge graph aims at enabling omni-channel applications in unknown environments. It therefore contains product-related knowledge that is used by different agents such as smartphone, smart glass and robot, which aim at providing shopping assistance in a retail store. In order to provide user assistance like routing a customer to a searched product on different devices such as robot or smartphone, this scenario focuses on accessing relevant Web knowledge about products in a retail store that is linked to precise, reliable and agent-independent environment information. The food cutting knowledge graph aims at enabling robots to execute task variations of cutting actions. Here, the idea is to access Web knowledge to enable a robot to autonomously perform a range of cutting tasks. Therefore, this scenario focuses on how object information can influence action execution, how the needed knowledge can be acquired from the Web and how it can be modelled in a knowledge graph in such a way that a robot can use it to execute tasks. The methodology is validated by showcasing various applications that are enabled by the two exemplary knowledge graphs. The applications range from smartphone applications for shopping assistance that highlight interesting product features or route to a searched product over smart glass applications like shopping assistance and a recipe application to robot applications for shopping assistance and execution of cutting task variations on different fruits and vegetables.

Zusammenfassung

Wissensgraphen sind ein Forschungstrend. Die Idee, Wissen aus dem Internet zu akquirieren und es Nutzern zur Beantwortung von Anfragen zugänglich zu machen, hat zu vielen berühmten Wissensgraphen in der Industrie (z.B. Google, Facebook oder Amazon) und in der Forschung (z.B. wikidata, WordNet oder FrameNet) geführt. Aufgrund ihres Erfolges kann man zu dem Schluss kommen, dass Wissensgraphen auch Robotern bei der Beantwortung komplexer Anfragen helfen könnten. Um Wissensgraphen in Roboteranwendungen nutzen zu können, muss man jedoch zunächst verstehen, wie Robotersteuerungssysteme funktionieren. Ein Roboter kann eine Reihe von Sensoren verwenden, um seine Umgebung wahrzunehmen und ein internes Modell der wahrgenommenen Umgebung aufzubauen, welches die wahrgenommenen Daten in einen Zusammenhang bringt. Das bedeutet, dass ein Roboter mit einem seiner Sensoren (z.B. eine Kamera) einen Apfel auf dem Tisch vor ihm wahrnehmen kann. In seinem internen Modell speichert er dann die Information, dass in der Umgebung ein Apfel und ein Tisch vorhanden sind (die durch die Speicherung der genauen Objektpositionen in Beziehung zum Roboter gesetzt werden), aber er könnte zusätzlich die Information hinzufügen, dass der Apfel *auf* dem Tisch liegt, was die Position des Apfels mit der Position des Tisches sowie des Roboters weiter spezifiziert. Wenn der Roboter nun die Aufgabe "Schneide den Apfel" erhält, greift er auf sein internes Modell zu, um zu überlegen, wo er einen Apfel finden könnte, um dann die gestellte Aufgabe auszuführen, indem er die Aufgabe mit den Umgebungsinformationen seines internen Modells in Beziehung setzt. Das interne Modell kann dabei auch mit einer Wissensbasis verknüpft sein, die zusätzliches Wissen enthält, welches der Roboter möglicherweise benötigt. Wenn wir die gegebene Aufgabe in "Gib mir die Milch" ändern, aber die Umgebung nicht verändern, so dass der Roboter weiterhin nur einen Apfel auf einem Tisch erkennt, wäre er ohne zusätzliches Wissen nicht in der Lage, diese Aufgabe zu erfüllen. Wenn das interne Modell jedoch mit einer Wissensbasis verknüpft ist, die die Information enthält, dass Milch ein verderbliches Produkt ist, das normalerweise in einem Kühlschrank

aufbewahrt wird, kann der Roboter auf diese Wissensbasis zugreifen und daraus ableiten, dass er nach einem Kühlschrank suchen muss, um die gegebene Aufgabe zu erfüllen.

Ein Großteil dieses zusätzlichen Wissens, das zur Erfüllung von Aufgaben benötigt wird, ist im Internet verfügbar. Ein Agent, der in der Lage ist, auf Wissensgraphen mit Web-Informationen zuzugreifen, kann daraus schließen, dass es sich bei dem Apfel auf dem Tisch um ein Kernobst einer Lebensmitteltaxonomie handelt sowie dass Kernobst ein Kerngehäuse hat, das normalerweise vor dem Verzehr entfernt wird. Damit ein Roboter jedoch auf diese Informationen zugreifen kann, muss das in einem Wissensgraphen enthaltene Objektwissen mit dem internen Modell des Agenten verbunden werden. Wenn der Roboter außerdem wissen soll, wie er den Apfel halten muss, um ihn zu schneiden, oder wie er das Kerngehäuse entfernen muss, müssen sowohl die Objekt- als auch die Umgebungsinformationen mit Aktionsinformationen verknüpft werden. Ein Wissensgraph, der Objekt-, Umgebungs- und Handlungsinformationen verknüpft, macht *das enthaltene Wissen handlungsfähig*.

In dieser Arbeit wird eine fünfstufige Methodik zur Erstellung von handlungsfähigen Wissensgraphen vorgeschlagen, die sich an bestehenden Standards der Wissensmodellierung orientiert, dabei aber Objektwissen mit Umgebungs- und Handlungswissen verknüpft, um verschiedene Anwendungen in alltäglichen Umgebungen und auf unterschiedlichen Agenten zu ermöglichen. Die Methodik wird beispielhaft in zwei Szenarien mit unterschiedlichen Schwerpunkten angewandt, um einen Produktwissensgraphen und einen Wissensgraphen zum Schneiden von Lebensmitteln zu erstellen. Der Produktwissensgraph zielt darauf ab, Omnichannel-Anwendungen in unbekannten Umgebungen zu ermöglichen. Er beinhaltet daher Produktinformationen, die von verschiedenen Agenten wie Smartphones, Smart Glasses und Robotern genutzt werden, um Nutzer beim Einkauf in einem Einzelhandelsgeschäft zu unterstützen. Um z.B. den Nutzer auf verschiedenen Geräten wie Roboter oder Smartphone zu einem gesuchten Produkt zu leiten, konzentriert sich dieses Szenario auf den Zugriff auf relevante Produktinformationen aus dem Internet, die mit präzisen, zuverlässigen und agentenunabhängigen Umgebungsinformationen verknüpft ist. Der Wissensgraph für das Schneiden von Lebensmitteln zielt darauf ab, Roboter in die Lage zu versetzen, Aufgabenvariationen von Schneideaktionen auszuführen. Daher konzentriert sich dieses Szenario auf die Frage, wie Objektinformationen die Ausführung von Aktionen beeinflussen können, wie das benötigte Wissen aus dem Internet gewonnen werden kann und wie es in einem Wissensgraphen so modelliert werden kann, dass ein Roboter es zur Ausführung von Aufgaben nutzen kann. Die Methodik wird anhand verschiedener Anwendungen, die durch die beiden beispielhaften Wissensgraphen ermöglicht werden, validiert. Die Anwendungen reichen von Smartphone-Anwendungen zur Einkaufsunterstützung, die interessante Produktmerkmale hervorheben oder zu einem gesuchten Produkt führen, über Smart-Glass-Anwendungen wie Einkaufsunterstützung und eine Rezeptanwendung bis hin zu Roboteranwendungen zur Einkaufsunterstützung und Ausführung von Schneideaufgaben auf verschiedenen Obst- und Gemüsesorten.

Acknowledgements

I am not the type to really read acknowledgements but rather dive into the text. If you do, this is for you. And if you don't know me, even more so. Me, I am a mom of three. Why do I tell you this? Because it still makes a difference. I was naive enough to think that I could just as easily get my PhD as before kids, or as men. Unfortunately, even in the 2020s, it is not easy. You need nerves, and people that support you. If by any chance you are a mother reading this, please feel encouraged to pursue your goals. Also, feel free to contact me. I will try to support you in the ways I can. This already leads me to thank the person that helped me through my journey: Sabine Veit. Without you I most likely would have given up due to university bureaucracy. Thank you so much, I will try my best to be a similar help to others! Next, thank you Michael Beetz for believing in me. I remember the day when you asked me if I really wanted to work towards a PhD (to which I nodded enthusiastically). You could have questioned me, like others, but you believed in me and supported me. Thank you for that. You are an inspiration and I appreciate all the 5 minute talks in front of the coffee machine where you just shared the brightest ideas. Due to both our restricted time schedules I learned to pack all important things into such a 5 minute discussion - this will surely be very helpful in the future as well! Next I want to thank my family, and most importantly my kids. You actually deserve the most gratefulness. Thank you for your patience, for dealing with me being in submission mode and not having time for your problems, for me being late, not watching your sports game but rather working or not preparing lunch, for your support. I love you. Last but not least, I want to thank all my students for their works and my colleagues and outside collaborators for the discussions and cooperation that have led to this Thesis.

Acronyms

API Application Programming Interface

AR Augmented Reality

bfo Basic Formal Ontology

cdno Compositional Dietary Nutrition ontology

ChEBI Chemical Entities of Biological Interest

CRAM Cognitive Robot Abstract Machine

DL Description Logic

doid human disease ontology

DUL DOLCE+DnS Ultralite

ETL extract load transform

FDA Food & Drug Administration

FoodKG Food Knowledge Graph

FoodOn Food Ontology

GTIN Global Trade Identification Number

hpo human phenotype ontology

KnowRob Knowledge Processing for Robots

LLM Large Language Model

NDRF National Drug File

NEEM Narrative enabled episodic memory

OBO Foundry Open Biological and Biomedical Ontology Foundry

OWL Web Ontology Language

QR code Quick response code

RDF Resource Description Framework

RDFS Resource Description Framework Schema

RT-2 Robotics Transformer 2

semDT semantic Digital Twin

SOMA Socio-physical Model of Activities

SPARQL SPARQL Protocol And RDF Query Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

Contents

List of Figures

List of Tables