Erkläre eine Robotik Konferenz... mit AI!

Storyboard

Für uns ist es immens wichtig, über die aktuellen Tätigkeiten anderer Forschungseinrichtungen informiert zu sein. In Österreich gibt es eine Plattform wo Forschungseinrichtungen die Ergebnisse ihrer Arbeit im Bereich Robotik und AI präsentieren können - diese Plattform ist der Austrian Robotics Workshop. Dort werden jedes Jahr zwischen 30 und 50 Paper vorgestellt, welche auf zwei bis sechs Seiten die wissenschaftlichen Ergebnisse präsentieren. Wir wollen diese Publikationen verwenden, um einen Überblick über den Stand der Wissenschaft in der österreichischen Szene zu gewinnen. Aber wie kann man mit so viel Text umgehen?

Um große Mengen an Text effizient zu erfassen, arbeiten wir mit Natural Language Processing und Regular Expressions (siehe die AIAV Videos zu Sprachverarbeitung, Textverarbeitung 1 und Textverarbeitung 2). Natural Language Processing (NLP) ist eine Mischung aus künstlicher Intelligenz und Linguistik. Das Ziel von NLP ist es Statistik zu verwenden, um große Mengen an Text zu analysieren [1]. Probleme die dabei von NLP addresiert werden sind vor allem maschinelle Übersetzung, Einholen von Informationen, Textkategorisierung und Datenextahierung aus Text [2].

Aber wie wendet man NLP in der Praxis an? Neben kompletten Toolkits, wie z.B. dem Python Natural Langugae Toolkit, können sogennante Regular Expressions (RegEx) verwendet werden. Regular Expressions sind in mehreren Programmiersprachen verfügbar - hier wurden sie innerhalb von Python mittels des re Moduls verwendet. RegEx können als eigene kleine Programmiersprache gesehen werden. Sie erlauben es, effizient nach Mustern in einem Text zu suchen. Dabei wird das Muster mittels sogenannter Matching Characters spezifiziert.

Das Python Regular Expressions HOWTO beschreibt die genaue Funktionalität der Matching Characters im Detail. Die Grundidee ist es, auf effiziente Weise ein Muster vorzugeben, welches im Text gefunden werden kann. RegEx erlauben es in ihrer einfachsten Form, aufeinanderfolgende Charaktere im Text zu finden. RegEx sind in den folgenden Absätzen und genauer erklärt.

In ihrer grundlegensten Form können RegEx bestimmte Wörter in einem Text finden. Im folgenden Beispiel setzen wir eine RegEx zur Suche des Wortes 'uns' ein.

In [20]:
```
# Wir importieren das re Modul zum Testen der Regular Expressions und definieren e
import re

text = "Das richtige Verwenden von AI erlaubt es uns, Probleme flexibel und nachvo
Wenn wir Probleme haben, wenden wir uns an die AIAV Wissensdrehscheibe."

# Die Regular Expression wird als 'uns' definiert.
# Die Suchfunktion sucht dann nach dem ersten Vorkommen der RegEx im Text.
expression = 'uns'

# Die Suchfunktion liefert uns den gefundenen Text und seine Position
print(re.search(re.compile(expression), text))
```

```
<re.Match object; span=(41, 44), match='uns'>
```

Wollen wir nun alle Vorkimnisse der RegEx finden, können wir anstatt re.search() re.findall()

verwenden. Im Beispieltext kommt 'uns' zwei mal vor.

In [10]:
```
print(re.findall(re.compile(expression), text))
```

```
['uns', 'uns']
```

Neben Buchstaben und Zahlen können auch Sonderzeichen in Regular Expressions verwendet werden. Während die meisten Zeichen einfach nur mit sich selbst matchen, bieten einige Sonderzeichen, die sogenannten Matching Characters, extra Funktionalität für den Ausdruck einer Regular Expression. So ist es möglich, bestimmte Gruppen an Charakteren von der Suche auszuschließen oder nach sich wiederholenden Sequenzen zu suchen. Beispiele für Matching Charaktere und ihre Funktionen sind in Tabelle 1 zusammengefasst.

Tabelle 1: Zusammenfassung der wichtigsten Matching Charaktere:

Symbol	Funktion
^	Negiert den vorhergehenden Teil der Expression
*	Spezifiziert 0 oder mehr Wiederholungen
+	Spezifiziert 1 oder mehr Wiederholungen
?	Spezifiziert mindestens 0 und maximal 1 Wiederholungen
{m,n}	Spezifiziert mindestens m und maximal n Wiederholungen
[ ]	Definieren eine Klasse von Ziffern
\	Definiert ein set von Ziffern oder macht aus einem Metacharakter eine normale Ziffer
()	Definiert eine Gruppe

Praktische Implementierung

Für den praktischen Usecase wurden Regular Expressions zusammen mit anderen Textverarbeitungsmethoden zur Klassifizierung der Veröffentlichungen vom Austrian Robotics Workshop (ARW) verwendet. Wir wollen die Veröffentlichungen den vom Workshop angegebenen Themengebieten zuordnen um uns einen Überblick über aktuelle Forschungstrends zu verschaffen. Da es in der Formatvorlage des ARW keine händisch angegebenen Schlagwörter gibt, müssen wir zunächst potentiell wichtige Wörter aus den Texten ermitteln. Anhand dieser Schlagwörter werden die Texte dann den Themengebieten zugeordnet.

Der Ablauf dafür ist wie folgt: Zunächst wird der rohe Text aus den Artikeln importiert. Da nicht alle Teile des rohen Textes für die Schlagwörter Interessant sind, wird der Text zunächst mittels RegEx formatiert. Das Literaturverzeichnis wird entfernt, indem die Überschrift "REFERENCES" im Text gesucht und alle folgenden Charaktere weggelassen werden. Dannach werden Textformatierende Charaktere (wie z.B. "\n" für neue Zeilen) und Zitationen (bestehend aus einer Zahl in eckigen Klammern) entfernt. Kapitelüberschriften und Zahlen im Text werden ebenfalls mittels Regular Expressions gesucht und entfernt. Idealerweise erhalten wir so einen Text, welcher nur Wörter aus dem Fließtext des Artikels und Satzzeichen beinhaltet.

Nun wird der Text in Sätze aufgeteilt. Das passiert anhand der immer noch im Text vorhandenen Satzzeichen. Die Sätze werden wiederum in Token aufgeteilt. Diese Token stellen Schlagwort Kandidaten dar. Die Aufteilung in Kandidaten erfolgt wiederum anhand der Satzzeichen (Satzende heißt auch Abschluss des aktuellen Kandidaten) und auf Basis einer Stoppliste. Die Stoppliste ist eine Liste aus Wörtern welche die Kandidaten voneinander trennen. Solche Stopplisten können

automatisch generiert werden und sind innerhalb der gleichen Sprache für verschiedene Anwendungsgebiete unterschiedlich [3].

Nach der Generierung der Kandidaten wird jeder Kandidat bewertet. Dafür wird gezählt, wie oft jedes Wort im Text vorkommt (Wortgrad) und in wie vielen Sätzen es vorkommt (Wortfequenz). Aus diesen beiden Metriken wird jeder Schlagwort Kandidat nummerisch bewertet (Wortgrad/Wortfrequenz). Dadurch ergibt sich eine nach ihrer Wichtigkeit sortierte Liste an potentiellen Schlüsselwörtern.

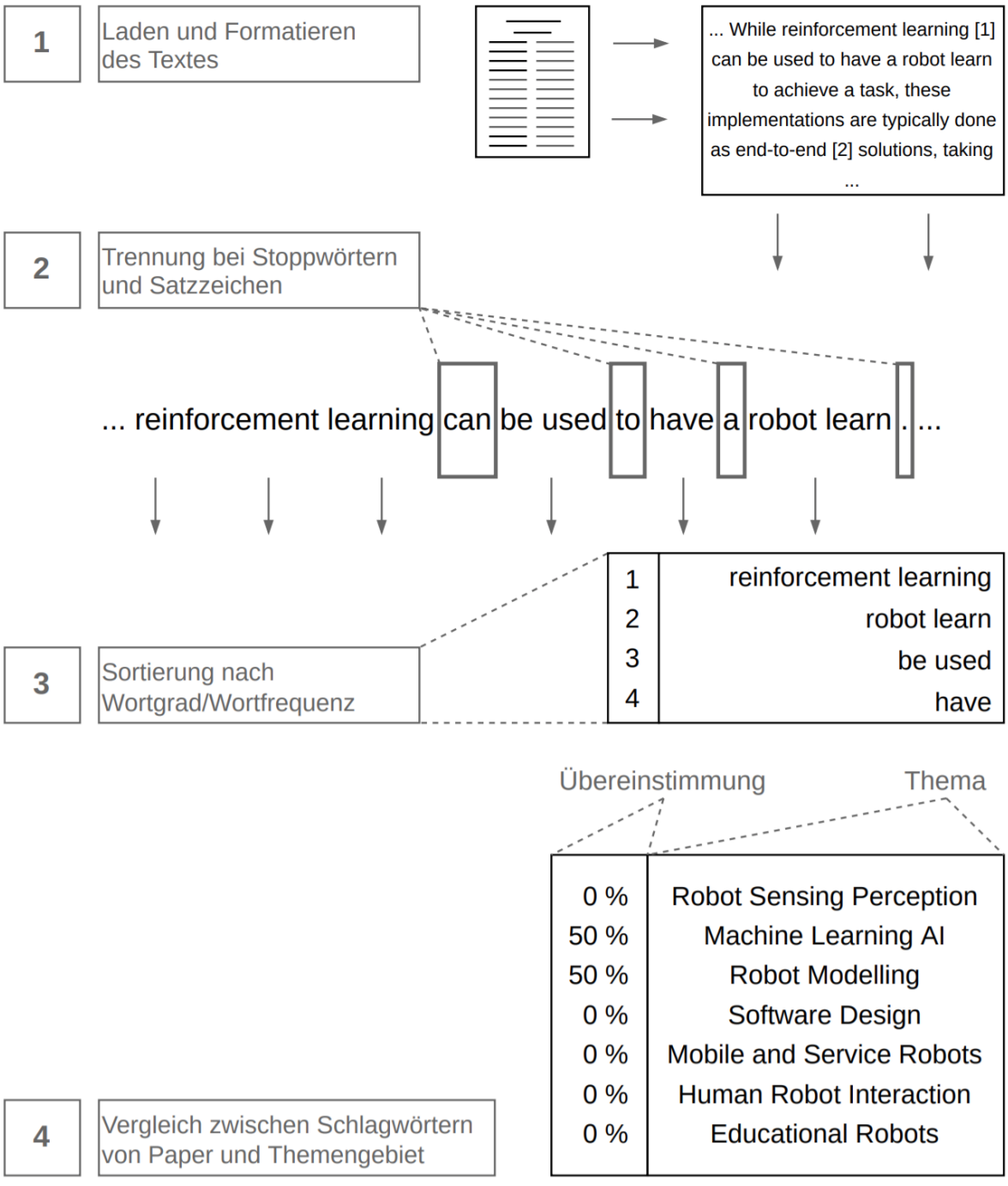

Abbildung 1: Damit wir uns über die Veröffentlichungen des Austrian Robotics Workshops einen Überblick verschaffen können, wollen wir die Themen der Publikationen automatisch ermitteln. Dafür wird der Text aus jeder Publikation zuerst geladen und formatiert. Stoppwörter teilen den Text dann in Phrasen auf. Diese Phrasen werden gezählt und anhand der Anzahl, wie oft sie im Text (Wortgrad) und in wievielen Sätzen die vorkommen (Wortfrequenz), bewertet. Die Phrasen mit der besten Bewertung werden dann zur Klassifizierung verwendet.

Diese Extrahierung wird nun für alle Publikationen des Austrian Robotics Workshops durchgeführt

(siehe Abbildung 1). Die Schlagwörter werden dabei für alle Publikationen aus einem Jahr gruppiert. Jedes Paper wird anschließend anhand seiner automatisch generierten Schlagwörter innnerhalb der Themengebiete in Tabelle 2 klassifiziert. Die Themengebiete spiegeln die auf roboticsworkshop.at angegebenen Themengebiete für den Austrian Robotics Workshop wieder. Für jedes Themengebiet wurde eine Liste an Schlagwörtern mittels Google Keyword Planner ermittelt und händisch überarbeitet. RegEx wurde hier wiederum verwendet, um zu überprüfen, welche Schlagwörter aus welchen Publikationen mit den Schlagwörtern in Tabelle 2 übereinstimmen.

Themengebiet	Schlagwörter
RobotSensingPerception	sensor, filter, sensing, vision, camera
MachineLearningAI	machine learning, artificial intelligence agent, feature, classification, network
RobotModelling	model estimation, pose, kinematic, structure
SoftwareDesign	software middleware, simulation, communication, verification
MobileAndServiceRobots	mobile, plan, rescue, ontology, constraint
HumanRobotInteraction	human user, interact, safe, workspace
EducationalRobots	education, show, demonstrate, workshop, school

Tabelle 2: Regular Expressions ordnen die Paper anhand ihrer Schlagworte verschiedenen Themengebieten zu. Dazu wurde zu jedem Thema eine Liste von Schlagworten basieren auf dem Google Keyword Planner erstellt.

Die Ergebnisse der einzelnen Klassifikationen werden pro Jahr aufsummiert und der prozentuale Anteil jedes Themas für das jeweilige Jahr wird berechnet. Letzendlich werden diese Anteile, wie in Abbildung 2 gezeigt, als Balkendiagramm dargestellt.

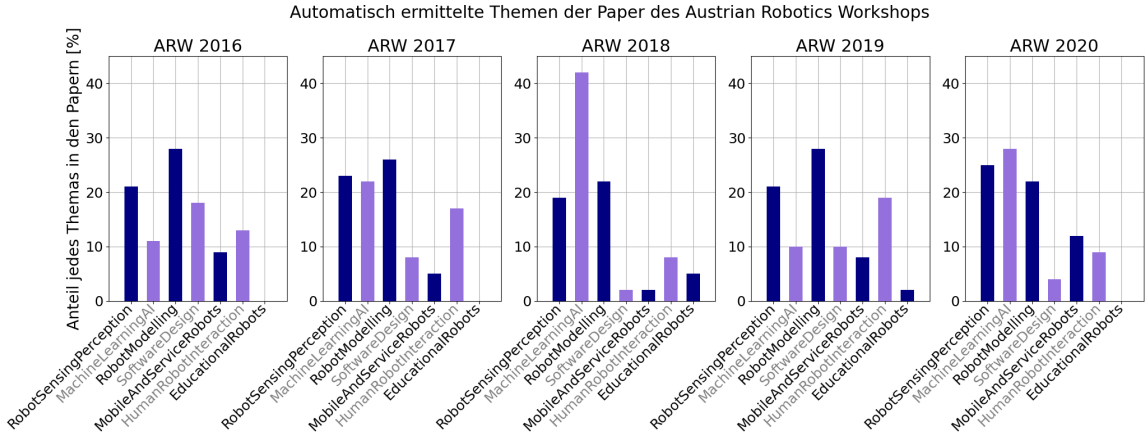

Abbildung 2: Die Veröffentlichungen des Austrian Robotics Workshops wurden mittels Regular Expressions auf Schlagwörter untersucht. Diese extrahierten Schlagwörter erlauben es uns, die Paper anhand ihres Themengebiets zu klassifizieren um einen Überblick über aktuelle Trends in der Forschung in Österreich zu erlangen.

Fazit

Regular Expressions erlauben es uns, Natural Language Processing schnell und effizient zu betreiben. Dadurch das RegEx so optimiert sind, können Untersuchungen von großen Mengen Text lokal, ohne viel Rechenleistung durchgeführt werden.

Die Themenbereiche der Veröffentlichungen zeigen einen klaren Trend zur Verwendung von Künstlicher Intelligenz in Robotik. Besonders im Jahr 2018, dem einzigen Jahr zwischen 2016 und

2020, in dem der Austrian Robotics Workshop nicht zusammen mit einer anderen Konferenz abgehalten wurde, fällt der Anteil an Publikationen mit Fokus auf AI sehr hoch aus. Dies zeigt uns, dass AI in der österreichischen Forschung an Robotik Anwendung findet.

Quellen

[1] Nadkarni, P.M., Ohno-Machado, L. and Chapman, W.W., 2011. Natural language processing: an introduction. Journal of the American Medical Informatics Association, 18(5), pp.544-551.

[2] Russell, S. and Norvig, P., 2002. Artificial intelligence: a modern approach.

[3] Rose, S., Engel, D., Cramer, N. and Cowley, W., 2010. Automatic keyword extraction from individual documents. Text mining: applications and theory, 1, pp.1-20.