

Masterarbeit

Synthese von Methodendefinitionen aus natürlichsprachlichen Äußerungen

Vanessa Steurer

Betreut von Sebastian Weigelt und Hans-Peter Zorn

IPD Tichy, Fakultät für Informatik



Motivation

- › Als Komponente in Projekt PARSE

- › Ziel von PARSE:
 - › Sprachbefehle → Funktionen des Zielsystems → Skript zur Ausführung

- › Bisher:
 - › Aufruf einprogrammierter Funktionen
 - › *“Open the fridge”*

- › Ziel der Thesis:
 - › Hinzufügen neuer Funktionen → *“Synthese von Methodendefinitionen”*
 - › *“Opening the fridge means ...”*

→ natürlichsprachliche Erweiterung intelligenter Zielsysteme

Verwandte Arbeiten

- › Programmierung & Funktionserweiterung in natürlicher Sprache
 - › SmartSynth [LGZ13] Skripte für Smartphones durch textuelle Beschreibungen
 - › Almond [CRX17] Dialogbasierter, programmierbarer virtueller Assistent
 - › Voxelurn [WM17] Sprach-Schnittstelle für den Aufbau von 3D-Pixelwelten
 - › Teaching Robots [SCC14] Dialogbasierte Robotererweiterung


- › Aber:
 - › kurze, einzelne Sätze
 - › fixe Signalwörter, Syntax und Vokabular
 - › domänenabhängig

Zielstellung der Arbeit

1. Erkennen einer Funktionserweiterung
 - › Klassifikation der Bestandteile
 - › Modellierung als Methodendefinition

2. Verknüpfung mit dem Zielsystem
 - › Domänenunabhängige API-Abbildung

Studiendetails

- › Ziel: Analyse der Formulierungen
- › Textuelle Datenerhebung
- › Crowdsourcing-Plattform  Prolific

Vorstudie


Klassifikation

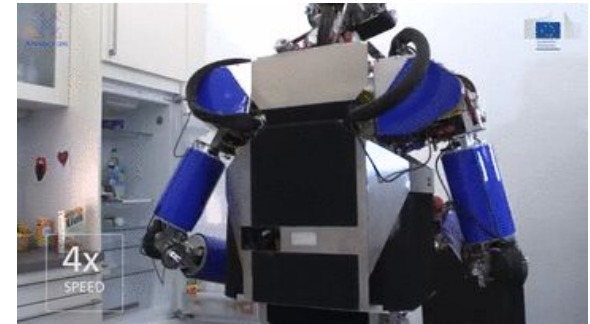
Repräsentation

API-Abbildung

Evaluation

Studiendetails

- › Ziel: Analyse der Formulierungen
 - › Textuelle Datenerhebung
 - › Crowdsourcing-Plattform  Prolific
 - › Texte zu vier unterschiedlichen Szenarien
 - › 870 Teilnehmer
- 3168 Datensätze (nach Bereinigung)
+ 242 Datensätze (PARSE-Korpus)
-
- 3410 Datensätze



Vorstudie

Klassifikation

Repräsentation

API-Abbildung

Evaluation

Fragestellungen der Klassifikation

- › Ziel: Erkennung einer Funktionserweiterung
 - › Wird eine Funktion aufgerufen oder hinzugefügt?
 - › Was ist der Funktionsname, was der Inhalt?
 - › Was sind die einzelnen Teilaufgaben?

Fragestellungen der Klassifikation

- › Ziel: Erkennung einer Funktionserweiterung
 - › Wird eine Funktion aufgerufen oder hinzugefügt? (Stufe 1)
 - › Was ist der Funktionsname, was der Inhalt? (Stufe 2)
 - › Was sind die einzelnen Teilaufgaben? (Stufe 3)

- hierarchische Klassifikation

- › Überwachtes maschinelles Lernen mittels neuronalen Netzen
 - › Umsetzung in Python (Keras, Tensorflow)
 - › Verwendung bestehender Agenten von PARSE

Erste Stufe der Klassifikation

- › Stufe 1: Wird eine Funktion **hinzugefügt** oder **aufgerufen**?

“in order to set the table take a plate and a fork and put them on the kitchen table please”

“we are eating soon i want you to set the table”

Erste Stufe der Klassifikation

- › Stufe 1: Wird eine Funktion **hinzugefügt** oder **aufgerufen**?
 - › binäres Entscheidungsproblem (2 Klassen)
 - › pro Eingabe

“in order to set the table take a plate and a fork and put them on the kitchen table please”

“we are eating soon i want you to set the table”

→ Training klassischer Lernverfahren sowie neuronaler Netze
Einsatz vortrainierter Wortvektoren
randomisierte Testdaten

Erste Stufe der Klassifikation

Modell	Architektur	Genauigkeit
Referenzwert	-	57.3
Logistische Regression	-	94.7
Random Forest	-	90.2
Einfaches Neuronales Netz	GMax, D(100)	89.6
Faltendes neuronales Netz	Conv(128,5), Max(2), Conv(128,5), GMax, D(10)	95.8
LSTM	LSTM(128), D(100)	62.5
Bidirektionales LSTM	BiLSTM(128), D(64)	95.9
Bidirektionales LSTM	BiLSTM(128), D(100), DO(0.3), D(50)	94.4

Vorstudie

Klassifikation

Repräsentation

API-Abbildung

Evaluation

Zweite Stufe der Klassifikation

- › Stufe 2: Wo ist der **Funktionsname** genannt? Wo der **Inhalt**? Gibt es **sonstige Äußerungen**?

“in order to set the table take a plate and a fork and put them on the kitchen table please”

Zweite Stufe der Klassifikation

- › Stufe 2: Wo ist der **Funktionsname** genannt? Wo der **Inhalt**? Gibt es **sonstige Äußerungen**?
 - › Mehrklassen-Entscheidungsproblem (3 Klassen)
 - › pro Wort (Sequenzmarkierungsproblem)

“in order to set the table take a plate and a fork and put them on the kitchen table please”

- Training von neuronalen Netzen
- Einsatz vortrainierter Wortvektoren
- randomisierte Testdaten

Zweite Stufe der Klassifikation

Modell	Architektur	Genauigkeit
Referenzwert	-	75.7
Einfaches Neuronales Netz	D(100)	84.8
LSTM	LSTM(128), D(64)	97.6
Bidirektionales LSTM	BiLSTM(128)	98.5
Bidirektionales LSTM	BiLSTM(128), DO(0.2)	98.8
Bidirektionales LSTM	BiLSTM(128), D(100), DO(0.3), D(50)	98.5
Bidirektionales LSTM	BiLSTM(256), D(100), DO(0.3), D(50)	98.5

Dritte Stufe der Klassifikation

- › Stufe 3: Was sind die einzelnen **Aktionen** inklusive zugehöriger **Entitäten**?

"in order to *set the table* *take a plate and a fork* and *put them on the kitchen table* please"

A1 A2 A3

Dritte Stufe der Klassifikation

- › Stufe 3: Was sind die einzelnen **Aktionen** inklusive zugehöriger **Entitäten**?
 - › Mehrklassen-Entscheidungsproblem
 - › pro Wort (Sequenzmarkierungsproblem)

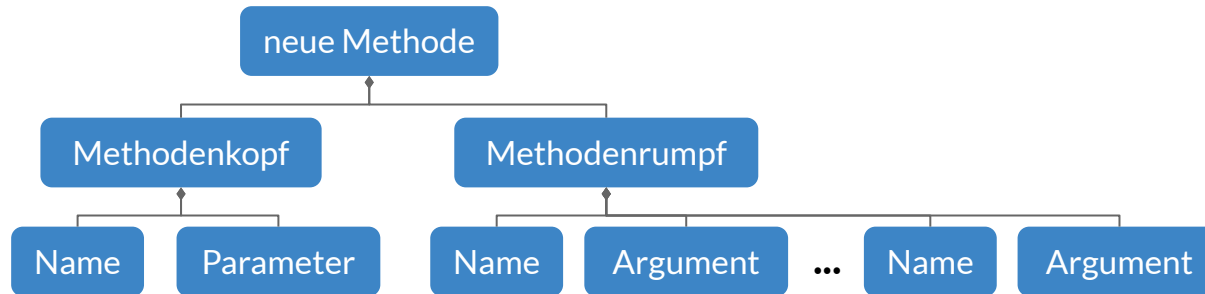
”in order to *set the table* *take a plate and a fork* and *put them on the kitchen table* please”

A1 A2 A3

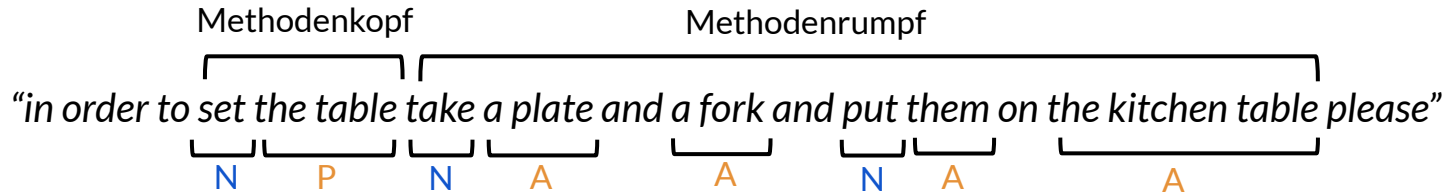
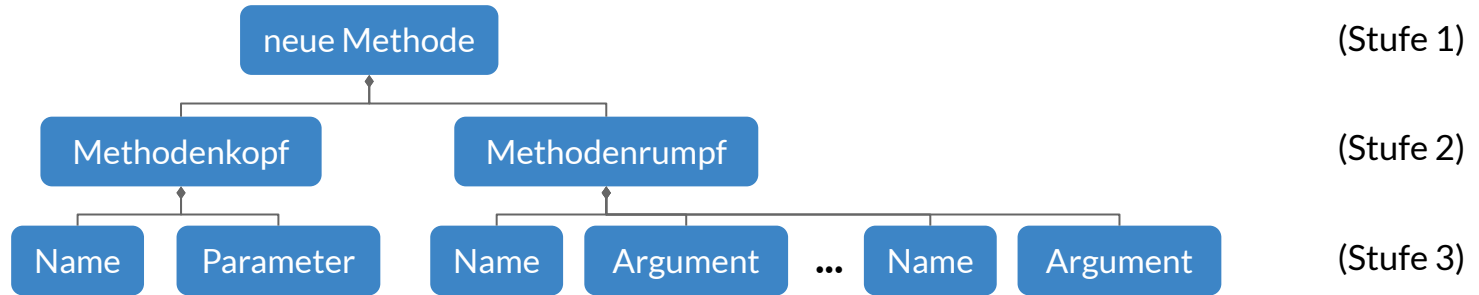
→ Verwendung der annotierten semantischen Rollen des SRL-Agenten von PARSE

Repräsentation der Klassifikationen auf Programm-Ebene

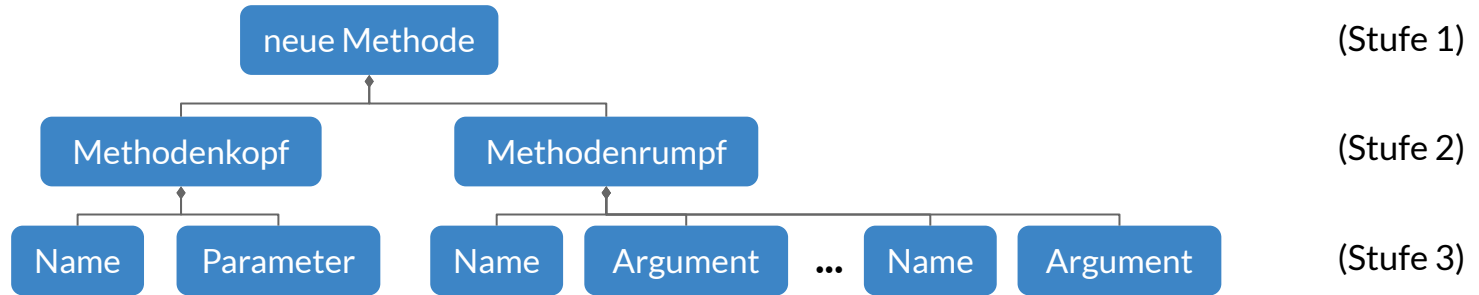
- › Ziel: Modellierung einer Funktionserweiterung als neue Methode
- › Repräsentation der Klassifikationen von **Sprach-Ebene** (Klassen) auf **Programm-Ebene** (Methoden)



Zusammenführung der Klassifikationsergebnisse



Zusammenführung der Klassifikationsergebnisse



Methodenkopf Methodenrumpf

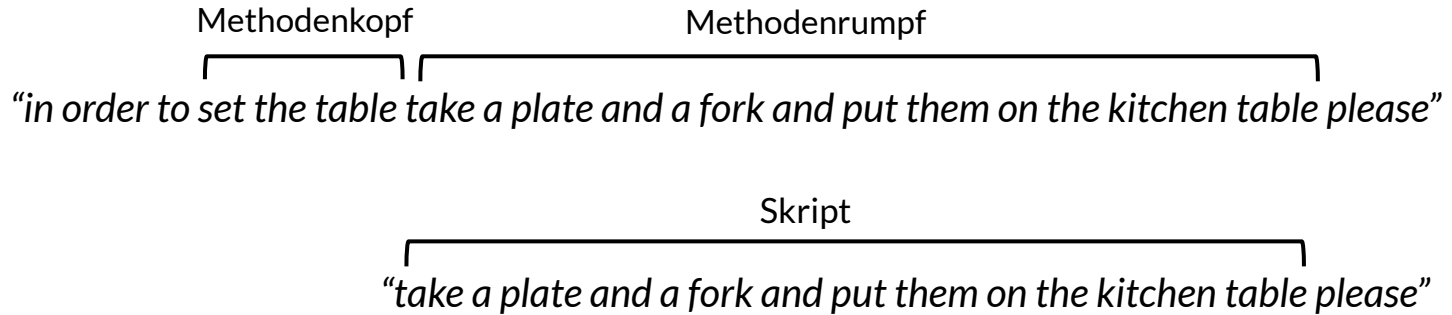
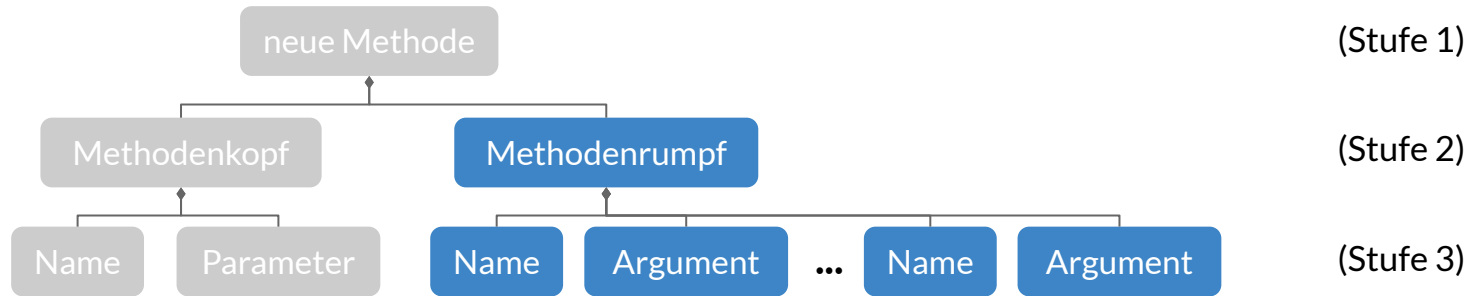
“in order to set the table take a plate and a fork and put them on the kitchen table please”

Funktions-
erweiterung

Kommando-
Sequenz

“take a plate and a fork and put them on the kitchen table please”

Zusammenführung der Klassifikationsergebnisse



Funktions-
erweiterung

Kommando-
Sequenz

Repräsentation der Klassifikationen auf Programm-Ebene

- › Modellierung der Klassifikationsergebnisse als Datenstruktur
 - › Ziel: Modellierung einer Funktionserweiterung als neue Methode
 - › Ziel: Modellierung einer Kommando-Sequenz als Skript
- › Integration als PARSE-Agent

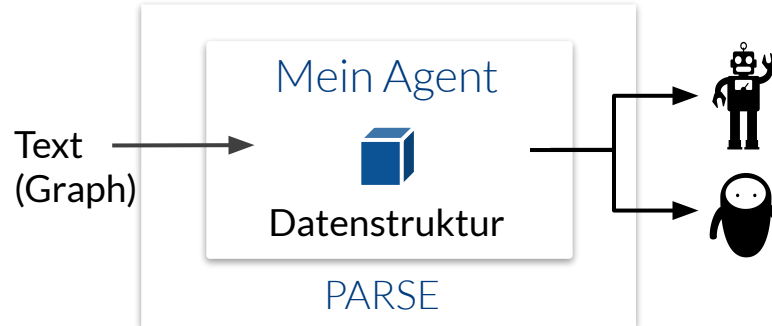
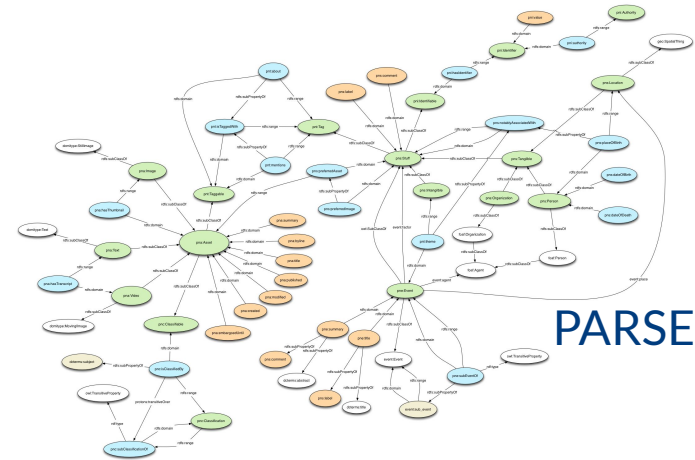


Abbildung auf die Zielsystem-API

- › Ziel: Verknüpfung der extrahierten Satzbestandteile mit Zielsystem-Funktionalitäten
 - › Zielsystem-API des Projekt PARSE ↔ Ontologie



Vorstudie

Klassifikation

Repräsentation

API-Abbildung

Evaluation

Abbildung auf die Zielsystem-API

- › Ziel: Verknüpfung der extrahierten Satzbestandteile mit Zielsystem-Funktionalitäten
 - › Zielsystem-API des Projekt PARSE ↔ Ontologie

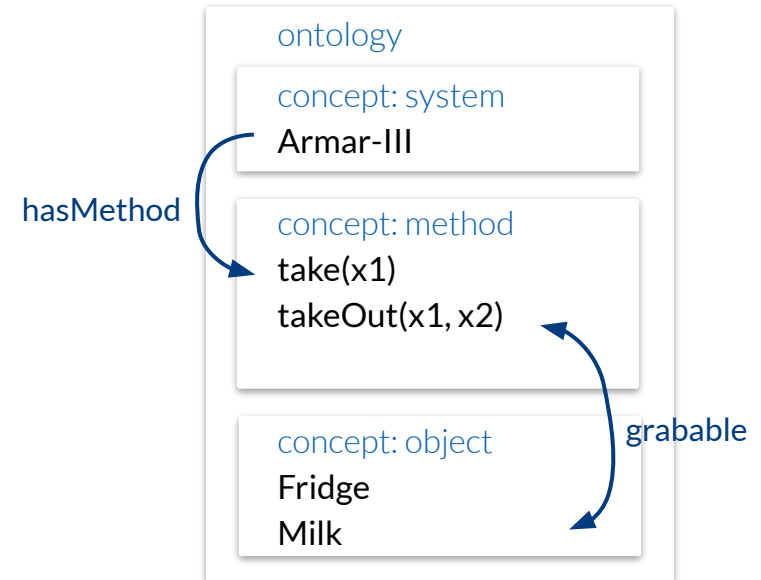



Abbildung auf die Zielsystem-API

- › Abbildung der Datenstruktur  auf Ontologie-Individuen
- › Methodenkopf (“neue Methode”) → neues Individuum

data structure 
new function: make cereals
subtask1_name: take
subtask1_params: milk, fridge
subtask2 ...

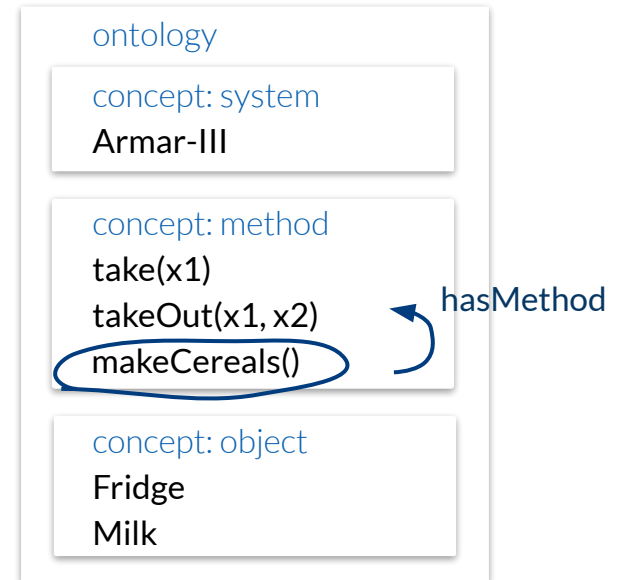




Abbildung auf die Zielsystem-API

- › Abbildung der Datenstruktur  auf Ontologie-Individuen
- › Methodenkopf (“neue Methode”) → neues Individuum
- › Methodenrumpf (“Funktionsinhalt”) → bestehende Individuen
- › Kommando-Skript (“Funktionsaufrufe”) → bestehende Indiv.

```

data structure 
new function: make cereals
subtask1_name: take
subtask1_params: milk, fridge
subtask2 ...
  
```

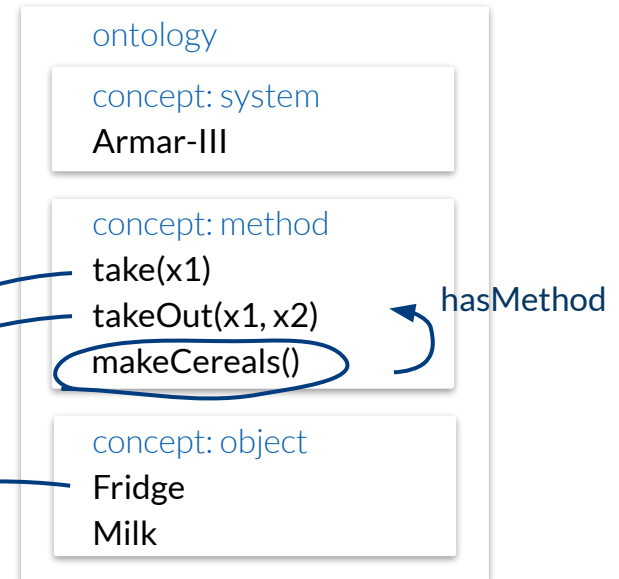


Abbildung auf bestehende Ontologie-Individuen

1. Vergleich der Bezeichner (*string matching*)

Vorstudie

Klassifikation

Repräsentation

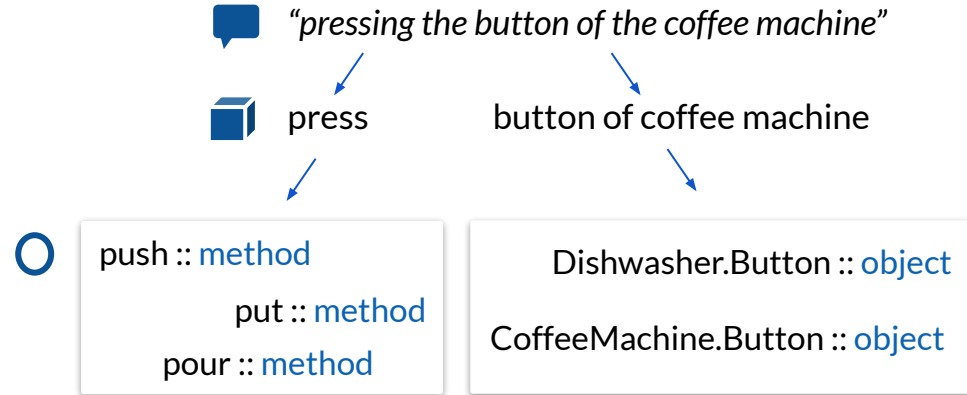
API-Abbildung

Evaluation

Abbildung auf bestehende Ontologie-Individuen

1. Vergleich der Bezeichner (*string matching*)

- › Zeichenketten-Vergleichsmetriken
- › Stopwörter & Wortart-Filter
- › Permutation
- › Lemmatisierung
- › Synonyme
- › Korreferenzen



→ **bezeichner-basierte** Kandidatenmenge an Funktionsnamen und Parameter

Abbildung auf bestehende Ontologie-Individuen

2. Vergleich der Datentypen

- › Bildung aller Kandidaten-Kombinationen
- › Vergleich der Datentypen von Methodenargument und **bezeichner-basiertem** Parameter-Kandidat

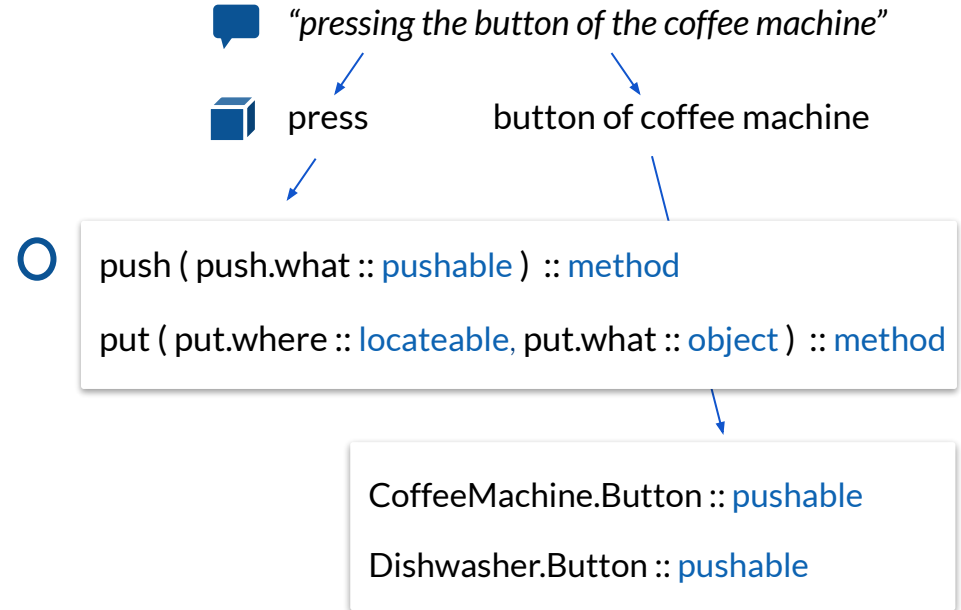
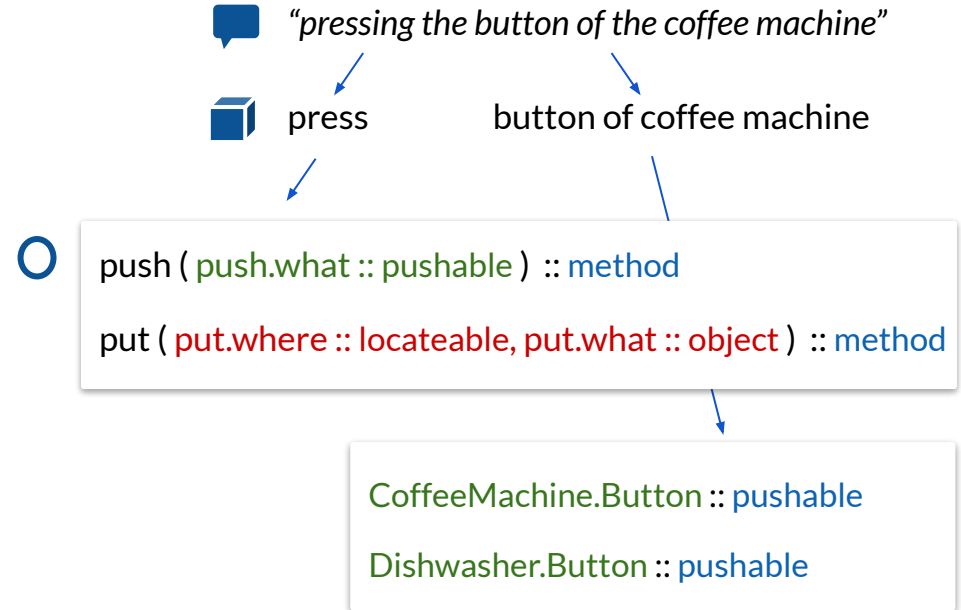


Abbildung auf bestehende Ontologie-Individuen

2. Vergleich der Datentypen

- › Bildung aller Kandidaten-Kombinationen
- › Vergleich der Datentypen von Methodenargument und **bezeichner-basiertem** Parameter-Kandidat



→ **typ-sichere** Kandidaten-Kombinationen

⊆ **bezeichner-basierten** Kandidaten

Abbildung auf bestehende Ontologie-Individuen

3. Bewertung der Kandidaten

- › **Bezeichner-Ähnlichkeit** auf Basis der Vergleichsmetriken
 - › des Methoden-Kandidaten *MK* zB “*press*” mit *push* :: **method**
 - › aller typ-sicheren Parameter-Kandidaten *PK* zB “*refrigerator*” mit *Fridge* :: **object**
- › **Abdeckung** der benötigten Methoden-Argumente *AMP* zB *put*(**put.where**, **put.what**)
- › **Abdeckung** der klassifizierten Aktionsparameter *AKP* zB “*put the bowl on the table*”

$$\text{Bewertungsfunktion } S = 0.6 \cdot MK + 0.4 \cdot \sum PK \cdot AMP - 0.3 \cdot AKP$$

Abbildung auf bestehende Ontologie-Individuen

3. Bewertung der Kandidaten

- › **Bezeichner-Ähnlichkeit** auf Basis der Vergleichsmetriken
 - › des Methoden-Kandidaten *MK*
 - › aller typ-sicheren Parameter-Kandidaten *PK*
- › **Abdeckung** der benötigten Methoden-Argumente *AMP*
- › **Abdeckung** der klassifizierten Aktionsparameter *AKP*

zB “press” mit push :: **method**

zB “refrigerator” mit Fridge :: **object**

zB put(**put.where**, **put.what**)

zB “put the **bowl** on the **table**”

☑ “pressing the button of the coffee machine”

○ K1: push (**CoffeeMachine.Button**) → Platz 1

K2: push (**Dishwasher.Button**) → Platz 2

K3: put () → Platz 3

→ Rangfolge **bewerteter** Kandidaten
pro extrahierter Aktion

Agentenausgabe: Synthese von Methodendefinitionen

- › Ziel: Modellierung einer Funktionserweiterung als neue Methode
- › Ziel: Modellierung einer Kommando-Sequenz als Skript

*“hi armar i am going to show you how to **start the dishwasher** you have to **close the door** to the machine and **press the blue button** afterwards”*

*“good morning armar when you need to **start the dishwasher** you have to first **close its door** and then **press the blue button**”*

```
new startDishwasher (Dishwasher) :  
    close (Dishwasher.Door)  
    press (Dishwasher.BlueButton)
```


Agentenausgabe: Synthese von Kommando-Skripten

- › Ziel: Modellierung einer Funktionserweiterung als neue Methode
- › Ziel: Modellierung einer Kommando-Sequenz als Skript

“carefully take the milk out of the fridge then take it to the kitchen table and pour the cereals into the bowl first then carefully pour the milk into bowl”

“retrieve the milk from the fridge, take it to the table and pour the cereals in the bowl then pour the milk on them”

```
retrieve(Milk, Fridge)
take(Milk, Table)
pour(Cereal, Bowl)
pour(Milk, Bowl)
```

Ende-zu-Ende-Evaluation

- › Ziel: Beurteilung der synthetisierten Methodendefinitionen / Skripte
 - › “Entsprechen die Agentenausgaben den korrekten API-Elementen?”
 - › Erhebung neuer Testdaten: **50** Eingaben zu je **zwei neuen** Szenarien (∅ 4.7 Instruktionen / Eingabe)

	Genauigkeit
Synthese Methodenname	

```
new startDishwasher()
```

```
new startDishwasherHiArmar()
```

Ende-zu-Ende-Evaluation

- › Ziel: Beurteilung der synthetisierten Methodendefinitionen / Skripte
 - › “Entsprechen die Agentenausgaben den korrekten API-Elementen?”
 - › Erhebung neuer Testdaten: **50** Eingaben zu je **zwei neuen** Szenarien (Ø 4.7 Instruktionen / Eingabe)

	Genauigkeit
Synthese Methodenname	90.4

```
new startDishwasher()
```

```
new startDishwasherHiArmar()
```

Top-1-Ergebnisse	Präzision	Ausbeute	F1-Maß
Abbildung Methodenname			
Abbildung Signatur			

```
take (Milk, Fridge)
```

```
take (Milk, Fridge)
```

Ende-zu-Ende-Evaluation

- › Ziel: Beurteilung der synthetisierten Methodendefinitionen / Skripte
 - › “Entsprechen die Agentenausgaben den korrekten API-Elementen?”
 - › Erhebung neuer Testdaten: **50** Eingaben zu je **zwei neuen** Szenarien (∅ 4.7 Instruktionen / Eingabe)

	Genauigkeit
Synthese Methodename	90.4

```
new startDishwasher()
```

```
new startDishwasherHiArmar()
```

Top-1-Ergebnisse	Präzision	Ausbeute	F1-Maß
Abbildung Methodename	88.6	93.4	90.9
Abbildung Signatur	66.8	72.1	69.4

```
take (Milk, Fridge)
```

```
take (Milk, Fridge)
```

Vorstudie

Klassifikation

Repräsentation

API-Abbildung

Evaluation

Zusammenfassung

- › **Ziele:** Funktionserweiterung von intelligenten Zielsystemen
 - › Synthese von Methodendefinitionen und Skripten aus natürlichsprachlichen Eingaben
- › **Anforderungen:**
 - › Abbildung unterschiedlicher Syntax bei gleicher Semantik → gleiche Ausgabe
 - › Zielsystem- und Szenarienunabhängigkeit → Domänenunabhängigkeit

Zusammenfassung

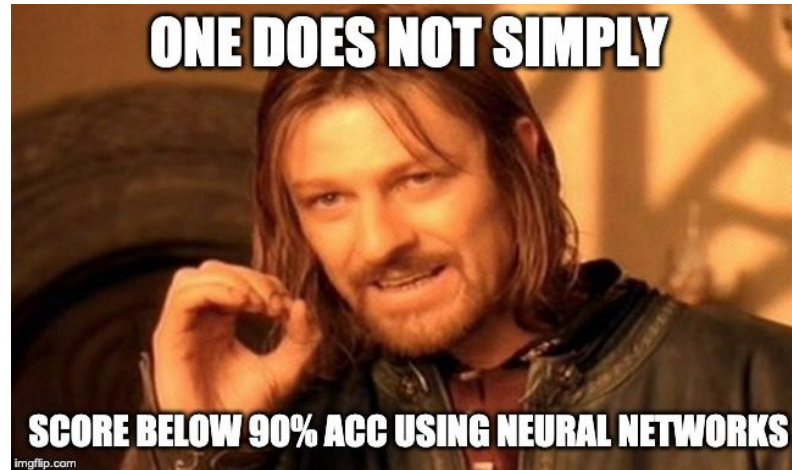
- › **Ziele:** Funktionserweiterung von intelligenten Zielsystemen
 - › Synthese von Methodendefinitionen und Skripten aus natürlichsprachlichen Eingaben
- › **Anforderungen:**
 - › Abbildung unterschiedlicher Syntax bei gleicher Semantik → gleiche Ausgabe
 - › Zielsystem- und Szenarienunabhängigkeit → Domänenunabhängigkeit
- › **Umsetzung:**
 - › Klassifikation von Funktionserweiterungen (3 Stufen, jeweils >95.9% Genauigkeit)
 - › Abbildung auf die Zielsystem-API
 - › Synthese von Methodennamen (90.4% Genauigkeit)
 - › Abbildung von Methodennamen (90.9% F1) und vollständiger Methodensignaturen (69.4% F1)
- › **Ausblick:** Alternative Klassifikation mit Google's BERT

Anhang

Erkenntnisse für inovex

- › Menge, Qualität & Diversität der Datengrundlage >> Modellkonfigurationen
- › Daten-Vorverarbeitung von Textdaten:
 - › tokenized >> lemmatized (→ größeres Vokabular)
 - › Stopwords, Ausreisser, Padding bei variabler Eingabelänge
- › Vortrainierte Wortvektoren (hier FastText) >> selbst trainierte Wortvektoren
 - › FastText >> GloVe >> Word2Vec
- › Sequenzmarkierungsprobleme → bidirektionale Architekturen (auch komplett ohne Attention!)
- › binäre Textklassifikation → bidirektionale Architekturen & CNNs
- › Potential von Google's BERT

Danke für eure Aufmerksamkeit!



New skill: Bring some beverage

Intermediate steps:

- All beverages are in the fridge
- Glasses are located on the kitchen counter
- Pour the selected beverage into a glass
- Hand it over to the user



How would you **teach** the robot this new skill?
"Speak" as naturally as possible. Be creative!

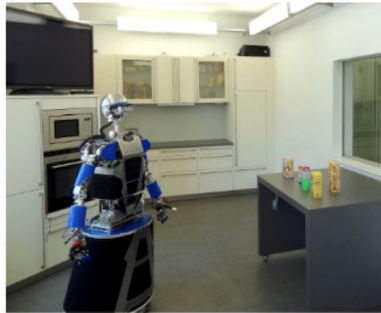
Synthetisierte Methodennamen (Ende-zu-Ende-Eval)

New Skill:
“Start the Dishwasher”



- startDishwasher
- startDishwasherHello
- startDishwasherHelloFirst
- turnOnDishwasher
- useDishwasher
- pushTheBlueButton

New Skill:
“Prepare Some Cereals”



- prepareCereals
- prepareSomeCereal
- prepareABowlOfCereal
- makeCereal
- makeCereals
- prepareCerealFirstThing
- makeSomeCereals
- makeBowlOfCereal

Klassifikator-Evaluation versus Ende-zu-Evaluation

binäre Stufe-1-Klf.	Genauigkeit	F1-Maß
Stufe-1-Klassifikator	95.9	93,3
Ende-zu-Ende	85.0	90.1

Details zu Testdaten:

- 300 Eingaben
je 4 Szenarien
- 2,97 Aktionen/E

- 50 Eingaben
je 2 Szenarien
- 4,73 Aktionen/E

(jeweils zufällig
gezogen)

Vorstudie

Klassifikation

Repräsentation

API-Abbildung

Evaluation

Top-1- und Top-3-Ergebnisse (Ende-zu-Ende-Eval)

	Top-1-Ergebnisse	Präzision	Ausbeute	F1-Maß
Top-1	Abbildung Methodename	88.6	93.4	90.9
	Abbildung Methodensignatur	66.8	72.1	69.4
Top-3	Abbildung Methodename	31.0	97.3	47.0
	Abbildung Methodensignatur	23.5	75.8	35.8

Ende-zu-Ende- versus API-Abbildungs-Evaluation

	Top-1-Ergebnisse	Präzision	Ausbeute	F1-Maß
Ende-zu-Ende	Abbildung Methodename	88.6	93.4	90.9
	Abbildung Methodensignatur	66.8	72.1	69.4
API-Abbildung	Abbildung Methodename	80.7	85.4	83.0
	Abbildung Methodensignatur	65.3	68.9	67.0

- 50 Eingaben je 2 Szenarien
- 4,73 Aktionen/E

- 25 Eingaben je 3 Szenarien
- 2,97 Aktionen/E

(jeweils zufällig gezogen)

Vorstudie

Klassifikation

Repräsentation

API-Abbildung

Evaluation

Häufigste Fehlerursachen des Agenten (Ende-zu-Ende-Eval)

Semantik

- “*make sure* you close the door”
- “*put* the cup *down*”
- “bring it the person *who asked you*”
- “then *it starts washing*”
- “when *I walked into the room*”
- “*the machine fills* the cup”

Fehlende Synonyme

- “*click* the button”
- “*load* the dishwasher”
- “pour a *portion* into the bowl”

Überblick Fehlerursachen des Agenten (Ende-zu-Ende-Eval)


Tabelle 10.9: Fehlerursachen für die Abbildungssuche von Methoden-Individuen


Ursache	Eingabe	Parameter	Musterlösung	Ergebnis
1 Bezeichner	„close the door“	[the door]	close(Cupboard.Door)	close(Fridge.Door)
	„go to the dishwasher“	[the dishwasher]	go(Dishwasher)	go(Dishes)
2 Synonyme	„click the button“	[the button]	push(Button)	-
	„load the dishwasher“	[the dishwasher]	fill(Dishwasher, Dishes)	lowerHead
	„pour a portion into a bowl“	[a portion, into a bowl]	pour(Milk,Bowl)	put(Popcorn,Bowl)
3 SRL	„the milk is found in the fridge“	[in the fridge]	find(Milk,Fridge)	find(?,Fridge)
	„pour milk onto cereals“	[milk onto cereals]	pour(Milk,Cereals)	pour(Milk,?)
4 Semantik	„press the button so <i>it starts washing</i> “	[]	-	wash(?)
	„ <i>make sure</i> the door is closed“	[sure, the door]	-	make(Fridge.Door)
5 Erklärung	„press the button <i>to start it</i> “	[it]	-	start(?)
	„press the button <i>to turn it on</i> “	[it, on]	-	turnOn(Kitchen)
6 Korref.	„put it on the table“	[it, table]	put(Bowl,Table)	put(Kitchen,Table)
	„put some in the bowl“	[some, the bowl]	put(Milk,Bowl)	put(?,Bowl)
7 Benutzung	„pour the milk“	[the milk]	pour(Milk,Bowl)	pour(Milk,?)
	„carrying to the table“	[the table]	carry(Bowl,Table)	carry(?,Table)
8 Klassif.	„to make cereals“	[]	-	make(Cereals)
	„ <i>push the button</i> to turn on the dishwasher“	[dishwasher]	push(Button)	turnOn(Dishwasher)

Beispiel: Bewertung der Kandidaten

- › **Bezeichner-Ähnlichkeit** auf Basis der Vergleichsmetriken
 - › des Methoden-Kandidaten **MK**
 - › der typischeren Parameter-Kandidaten **PK**
- › **Abdeckung** der benötigten Methoden-Argumente **AMP**
- › **Abdeckung** extrahierter Aktionsparameter **AEP**

$$S = 0.6 \cdot MK + 0.4 \cdot \sum PK \cdot AMP - 0.3 \cdot AEP$$

 “put the bowl on the table carefully”

 [put] [the bowl, the table , carefully]

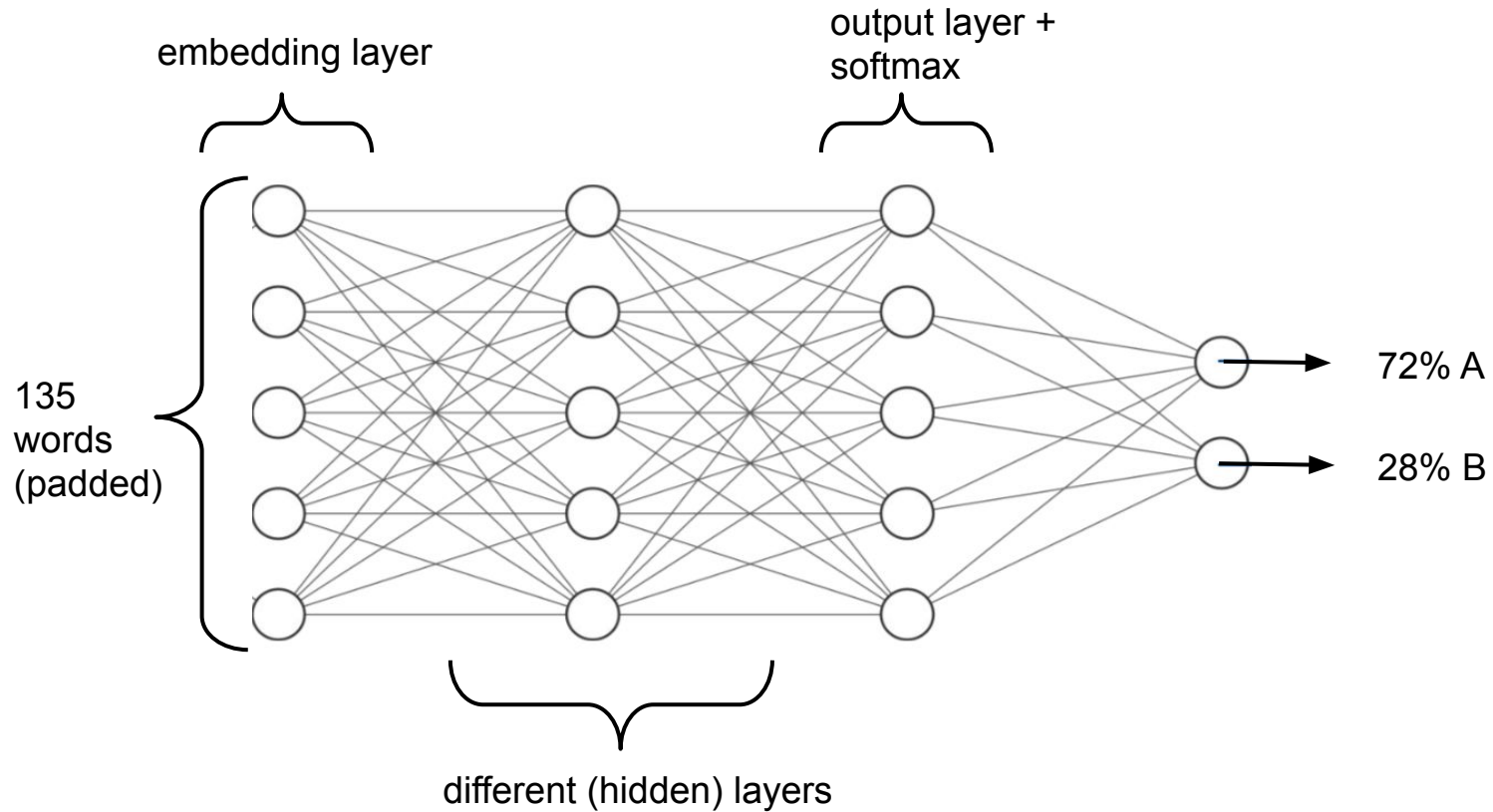
○ K1: put (put.what, put.where) :: MK = 0.85 ; Bowl :: PK1 = 0.9 ; Table :: PK2 = 0.88

K2: put (put.what, put.where) :: MK = 0.85 ; Bottle :: PK1 = 0.74 ; Table :: PK2 = 0.88

$$S1 = 0.6 \cdot 0.85 + 0.4 \cdot \sum (0.9 + 0.88) \cdot (2/2) - 0.3 \cdot (2/3) = 0.51 + 0.4 \cdot 1.78 - 0.2$$

$$S2 = 0.6 \cdot 0.85 + 0.4 \cdot \sum (0.74 + 0.88) \cdot (2/2) - 0.3 \cdot (2/3) = 0.51 + 0.4 \cdot 1.62 - 0.2$$

Binäre Klassifikation



Stufe-1-Klassifikation (binär “Lehrsequenz”)

Neuronale Netze

Tabelle 9.13: Genauigkeiten auf den randomisierten Validierungsdaten verschiedener neuronaler Netze für Stufe 1

Name	Erweiterung	Ep	SW _v	VW _v
ANN1	-	99	90.3	90.3
ANN2	Flat, D(100)	12	91.6	84.6
ANN3	GMax, D(100)	11	89.9	87.9
CNN1	Conv(128, 5), Max(2), Conv(128, 5), GMax, D(10)	7	95.2	95.4
RNN1	GRU(128), D(100)	12	56.2	56.2
RNN2	BiGRU(32), D0(0.2), D(64), D0(0.2)	12	94.7	95.2
RNN3	LSTM(128), D(100)	12	56.2	56.2
RNN4	BiLSTM(128), D(64)	5	95.1	95.6
RNN5	BiLSTM(128), D(100), D0(0.3), D(50)	10	93.6	94.5

Stufe-1-Klassifikation (binär “Lehrsequenz”)

Neuronale Netze

Tabelle 9.16: Test-Genauigkeiten der Stufe-1-Klassifikation

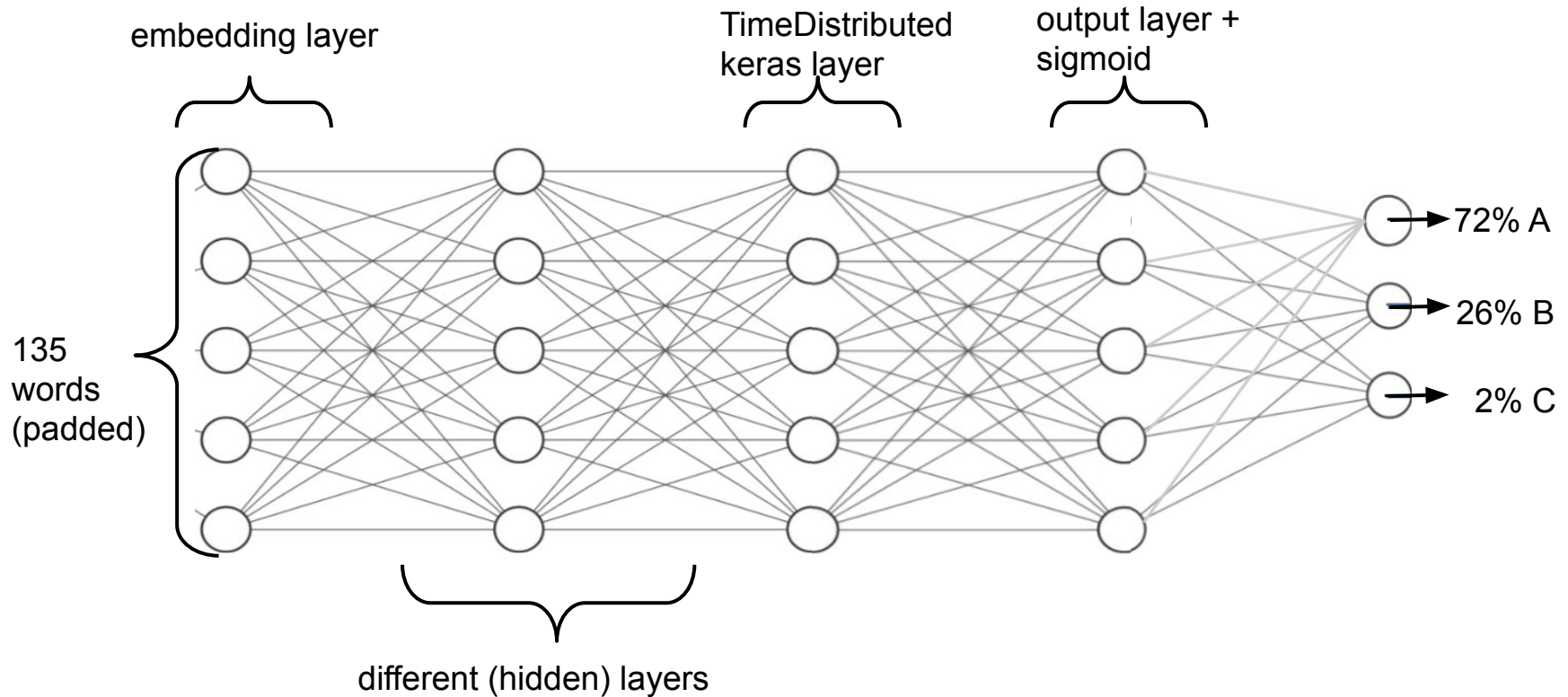
	Randomisiert		Szenarienbezogen	
	Validate	Test	Validate	Test
Referenzwert	57.3	-	54.7	-
Random Forest	92.1	90.2	90.1	37.4
Logistische Regression	92.7	94.7	92.9	71.9
CNN1 + VW _v	95.5	96.6	97.7	86.2
RNN4 + VW _v	95.6	95.9	96.2	91.9
BERT3	99.6	98.8	99.2	97.7

Stufe-1-Klassifikation (binär “Lehrsequenz”) Klassische Lernverfahren

Tabelle 9.9: Erzielte Genauigkeiten auf den Validierungsdaten der Stufe-1-Klassifikation mit klassischen Lernverfahren

	randomisiert
Referenzwert	57.3
Entscheidungsbaum	89.0
Random Forest	92.1
Support Vektor Machines	84.8
Naïve Bayes	77.1
Logistische Regression	92.7

Mehrklassen-Klassifikation (Sequenzmarkierung)



Stufe-2-Klassifikation (Mehrklassen “Lehrsequenz-Bestandteile”) Neuronale Netze

Tabelle 9.22: Genauigkeiten auf den randomisierten Validierungsdaten verschiedener neuronaler Netze für Stufe 2

Name	Erweiterung	SW _v	VW _v
ANN1	-	85.1	85.1
ANN2	D(50)	85.3	85.3
RNN1	LSTM(128)	97.4	97.8
RNN2	LSTM(128), D(64)	97.3	97.7
RNN3	BiLSTM(128)	98.6	98.7
RNN4	BiGRU(128)	98.4	98.5
RNN5	BiLSTM(128), D(100), DO(0.3), D(50)	98.2	98.2
RNN6	BiLSTM(128), DO(0.2)	98.5	98.8
RNN7	BiLSTM(256), DO(0.2)	98.6	98.7

Stufe-2-Klassifikation (Mehrklassen “Lehrsequenz-Bestandteile”) Neuronale Netze

Tabelle 9.24: Test-Genauigkeiten der Stufe-2-Klassifikation

	Randomisiert		Szenarienbezogen	
	Validate	Test	Validate	Test
Referenzwert	75.9	-	75.7	-
RNN6 + VW _v	98.8	98.8	98.1	97.5
RNN3 + SW _v	98.6	98.3	98.2	96.0
RNN6 + SW _v	98.5	98.4	98.2	95.8
RNN7 + VW _v	98.7	98.5	98.2	97.5

Literatur

- [LEZ13] Le, Vu ; Gulwani, Sumit ; Su, Zhendong: SmartSynth: synthesizing smartphone automation scripts from natural language, 11th Annual International Conference on Mobile Systems, Applications, and Services, 2013
- [MAW17] Wang, Sida I. ; Ginn, Samuel ; Liang, Percy ; Manning, Christopher D.: Naturalizing a Programming Language via Interactive Learning, Association for Computational Linguistics, 2017
- [SHE14] She, Lanbo; Cheng, Yu; Chai, Joyce Y.; Jia, Yunyi; Yang, Shaohua; Xi, Ning: Teaching Robots New Actions through Natural Language Instructions, 23rd IEEE International Symposium on Robot and Human Interactive Communication, 2014
- [CAM17] Campagna, Giovanni ; Ramesh, Rakesh ; Xu, Silei ; Fischer, Michael ; Lam, Monica S.: Almond: The Architecture of an Open, Crowdsourced, Privacy-Preserving, Programmable Virtual Assistant, 26th International Conference on World Wide Web, 2017