Talk to me

Chatbots und digitale Assistenten

Anna Weißhaar

Köln, 26. März 2019

@findefain

Anna Weißhaar

(Sachen)Sucherin

“Was mit Sprache” gelernt

- Linguistik und Philosophie @unihd

“Was mit Computern” gelernt

- Informatik @hska

Kombination

- Search Engineer @inovex

Chatbots: Definition

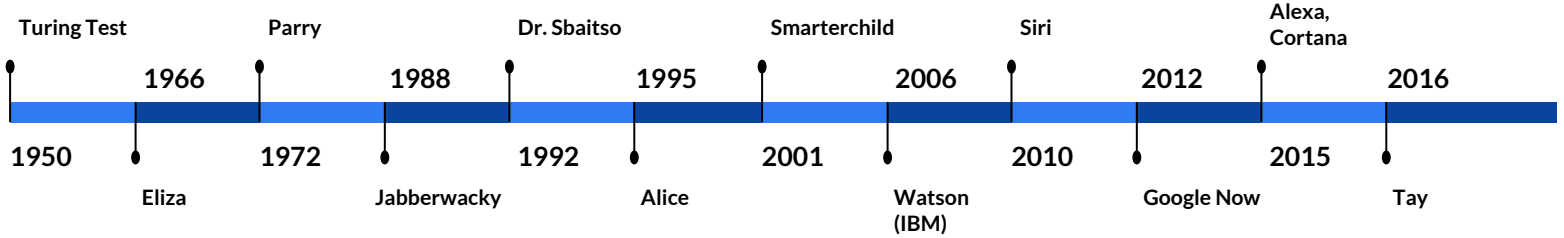
A **chatbot** (also known as a smartbot, conversational bot, chatterbot, interactive agent, conversational interface, Conversational AI, or artificial conversational entity) is a **computer program** or an artificial intelligence which **conducts a conversation** via auditory or textual methods. Such programs are often designed to **convincingly simulate** how a **human** would behave as a conversational partner, thereby passing the Turing test.

(Quelle: Internet, https://en.wikipedia.org/wiki/Chatbot)

Neither Aristotelian nor Russellian rules give the exact logic of any expression of ordinary language; for ordinary language has no exact logic.

(P. F. Strawson, *On Referring*)

Chatbots: Geschichte



https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

inovex

Chatbots: Versprechen

persönlich

natürliches Interface

besserer Kundenservice

Gespräche wie mit Menschen

Herausforderung

unstrukturierte Eingabe

Und dann würde mich halt auch noch interessieren, welche Größen es gibt.

Wie wird das Wetter am Samstag?

natürlichsprachige Antwort

Die **Sandalen** “Mallorca” haben wir in den **Größen** 36 bis 47 vorrätig.

Für **Karlsruhe** ist am **30.3.** sonniges **Wetter** bei Temperaturen bis 16 Grad vorhergesagt.

Absichten
Kontextinformationen

"Weltwissen"
bisheriger Kontext

ggf. Zustand
aktualisieren

inovex

Vorgehen traditionell: Regeln, Regeln, Regeln

Eliza: Pattern-Matching + Templates

```
key: computer 50
  decomp: *
    reasmb: Do computers worry you ?
    reasmb: Why do you mention computers ?
    reasmb: What do you think machines have to do with your problem ?
    reasmb: Don't you think computers can help people ?
    reasmb: What about machines worries you ?
    reasmb: What do you think about machines ?
```

```
key: forget 5
  decomp: * i forget *
    reasmb: Can you think of why you might forget (2) ?
    reasmb: Why can't you remember (2) ?
    reasmb: How often do you think of (2) ?
    reasmb: Does it bother you to forget that ?
    reasmb: Could it be a mental block ?
    reasmb: Are you generally forgetful ?
    reasmb: Do you think you are suppressing (2) ?
  decomp: * did you forget *
    reasmb: Why do you ask ?
    reasmb: Are you sure you told me ?
    reasmb: Would it bother you if I forgot (2) ?
    reasmb: Why should I recall (2) just now ?
    reasmb: goto what
    reasmb: Tell me more about (2).
```

A.L.I.C.E.: AIML

```
<?xml version = "1.0" encoding = "UTF-8"?>
<aiml version = "1.0.1" encoding = "UTF-8"?>
   <category>
      <pattern> HELLO ALICE </pattern>

      <template>
         Hello User
      </template>

   </category>
</aiml>
```

```
<category>
  <pattern> TELL ME ABOUT *</pattern>
  <template>
    Here is some information about <star/>:
    <url>https://en.wikipedia.org/wiki/<star/></url>
  </template>
</category>
```

Eliza in Aktion ...

Machine Learning to the Rescue (?)

Datenbasierte Chatbots

"The Rasa Stack is a set of **open source machine learning** tools for developers to create contextual text- and voice-based chatbots and assistants."

https://rasa.com/

Versprechen: Weniger hartcodierte Regeln, mehr AI

Rasa: Interpretieren und Reagieren

Interpretieren: Rasa NLU

Input: Nutzereingabe

Output: Intents, Entities

Reagieren: Rasa Core

Input: NLU-Output, Kontext

Output: Nächste Aktion

Rasa Grundidee: Domäne

entities => Dinge, über die gesprochen wird

intents => Was will die Nutzerin?

actions => Was kann der Bot tun?

---

templates => Was kann der Bot sagen?

slots => Gedächtnisinhalt

Domäne: Beispiel

https://github.com/RasaHQ/starter-pack-rasa-stack

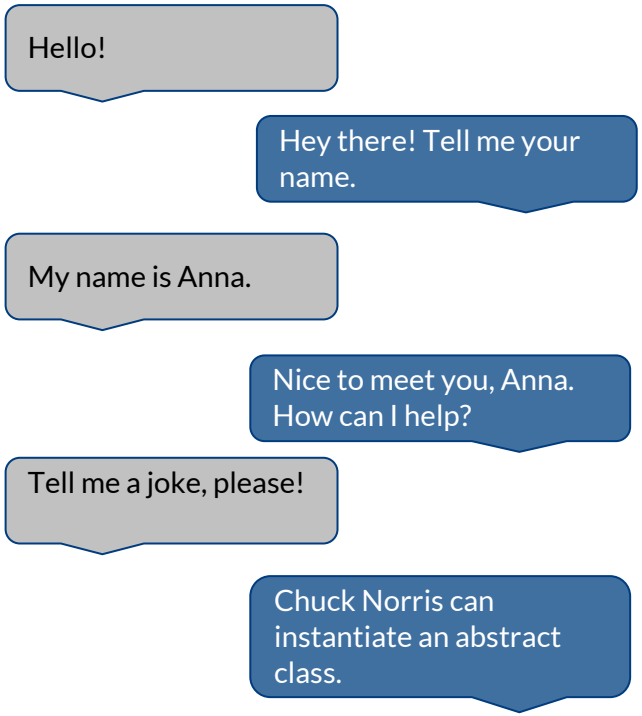
```
intents:
- greet
- name
- joke

entities:
- name

slots:
  name:
    type: text
```

```
actions:
- utter_greet
- action_joke
- utter_name

templates:
  utter_greet:
  - text: Nice to you meet you {name}. How can I help?
  utter_name:
  - text: Hey there! Tell me your name.
```



inovex

Interpretieren: Rasa NLU

My name is Anna


```
language: "en"

pipeline:
- name: "nlp_spacy"
- name: "tokenizer_spacy"
- name: "intent_featurizer_spacy"
- name: "ner_crf"
- name: "intent_classifier_sklearn"
```


```
{
 "intent": {
  "name": "name",
  "confidence": 0.859506104253725
 },
 "entities": [
  {
   "start": 11,
   "end": 15,
   "value": "anna",
   "entity": "name",
   "confidence": 0.9516187350992868,
   "extractor": "ner_crf"
  }
 ],
 "intent_ranking": [
  {
   "name": "name",
   "confidence": 0.859506104253725
  },
  {
   "name": "greet",
   "confidence": 0.07173887988512613
  },
...
 ],
 "text": "My name is Anna"
}
```


inovex

Rasa NLU: Trainingsdaten

```
## intent:greet
- Hi
- Hey
- Hi bot
- Hey bot

## intent:name
- My name is [Juste](name)
- I am [Josh](name)
- I'm [Lucy](name)
- People call me [Greg](name)
```

```
{
      "intent": "name",
       "entities": [
         {
           "start": 15,
           "end": 19,
            "value": "Greg",
           "entity": "name"
         }
       ],
       "text": "People call me Greg"
}
```

Rasa NLU: Training

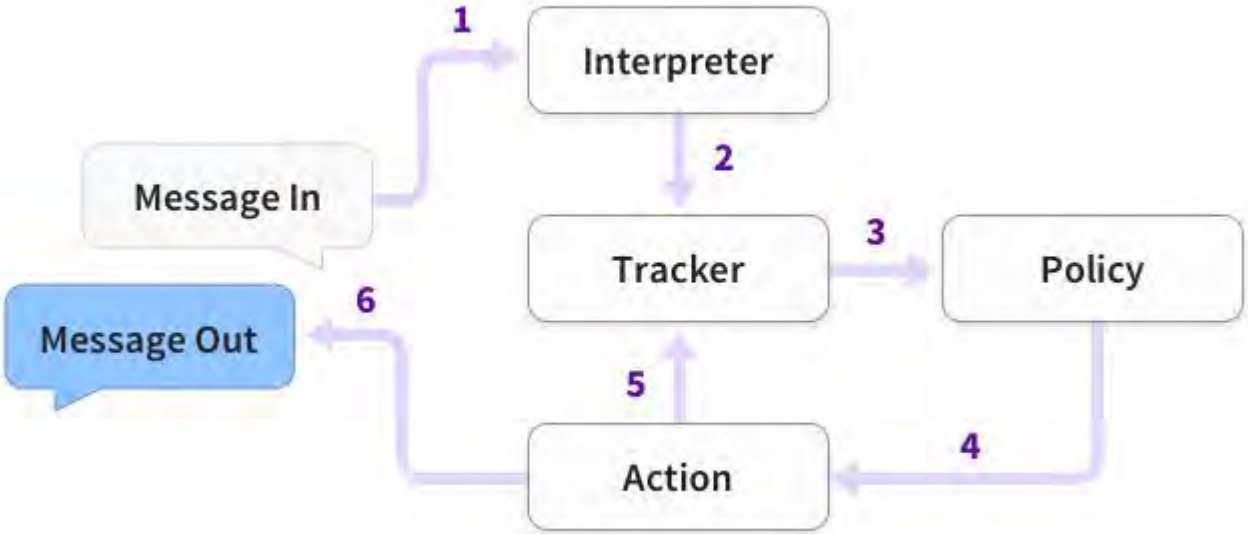
1) Preprocessing
   - Tokenisierung: Satz => Token
   - Features erzeugen: Token => Embeddings
2) NER: Trainiere CRF
   - Features: Transformierte Trainingsdaten (berücksichtige Nachbarn)
3) Intents: Trainiere SVM
   - Features: Word-Embeddings (gemittelt)
   - Labels: Intent-Namen

NLU Training in Aktion

Rasa NLU: Output


```
{
  "intent": {
    "name": "name",
    "confidence": 0.9501842025598839
  },
  "entities": [
    {
      "start": 13,
      "end": 17,
      "value": "anna",
      "entity": "name",
      "confidence": 0.9771150313654133,
      "extractor": "ner_crf"
    }
  ],
"text": "They call me Anna"
}
```

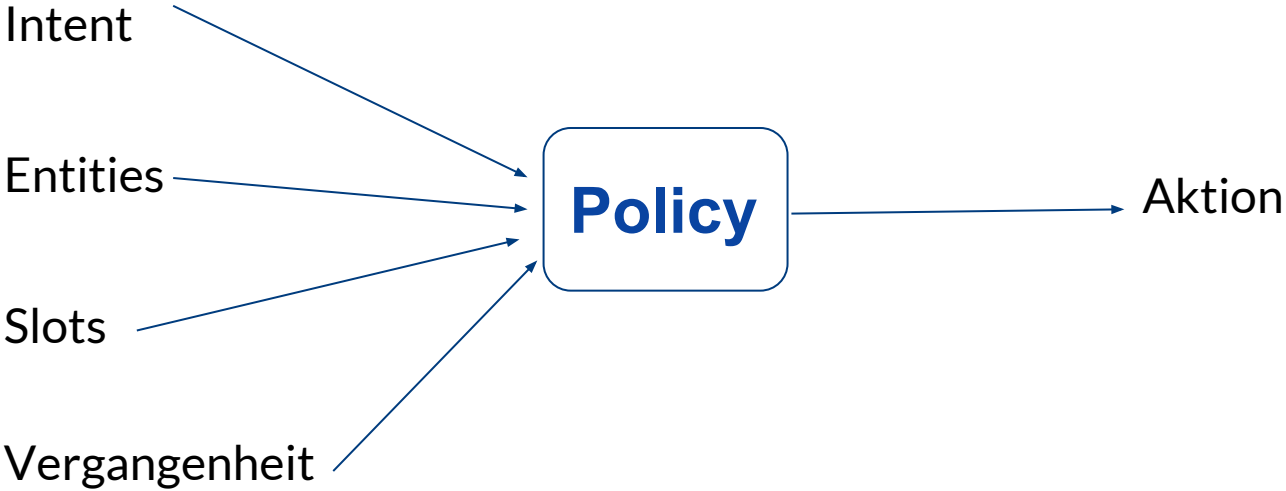
Reagieren: Rasa Core



https://rasa.com/docs/core/architecture/

inovex

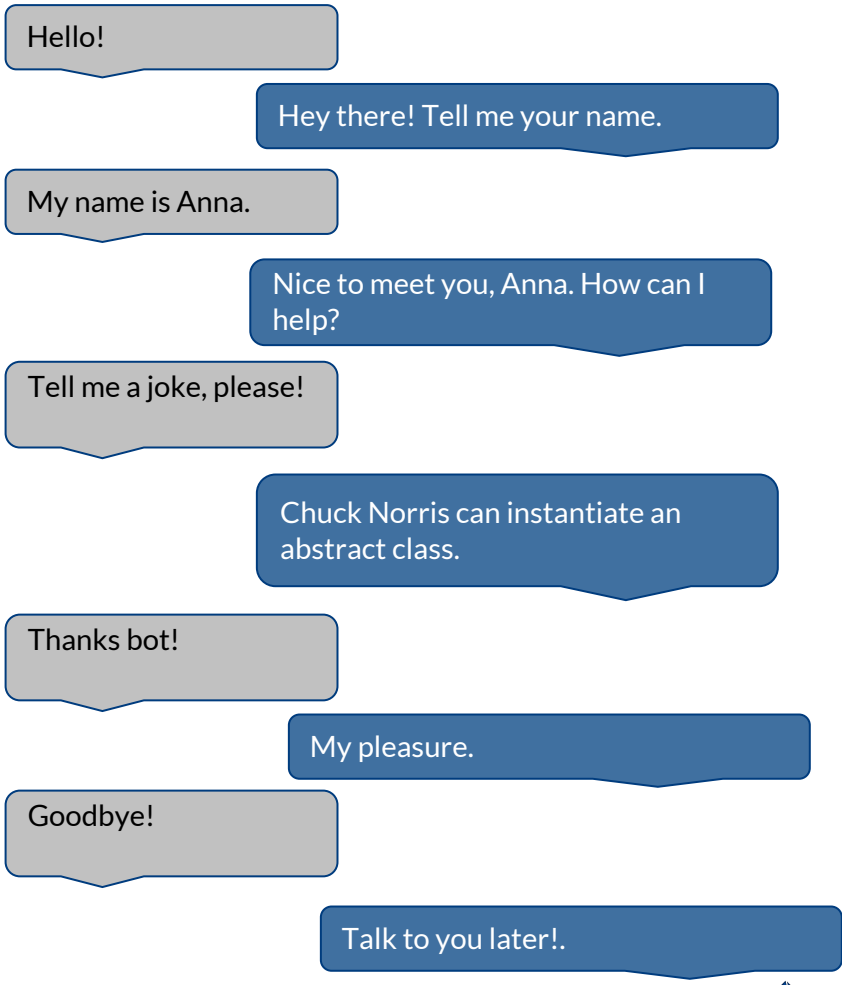
Reagieren: Rasa Core



Rasa Core: Trainingsdaten

Abläufe beschreiben

```
## story_joke_02
* greet
 - utter_name
* name{"name":"Anna"}
 - utter_greet
* joke
 - action_joke
* thanks
 - utter_thanks
* goodbye
 - utter_goodbye
```

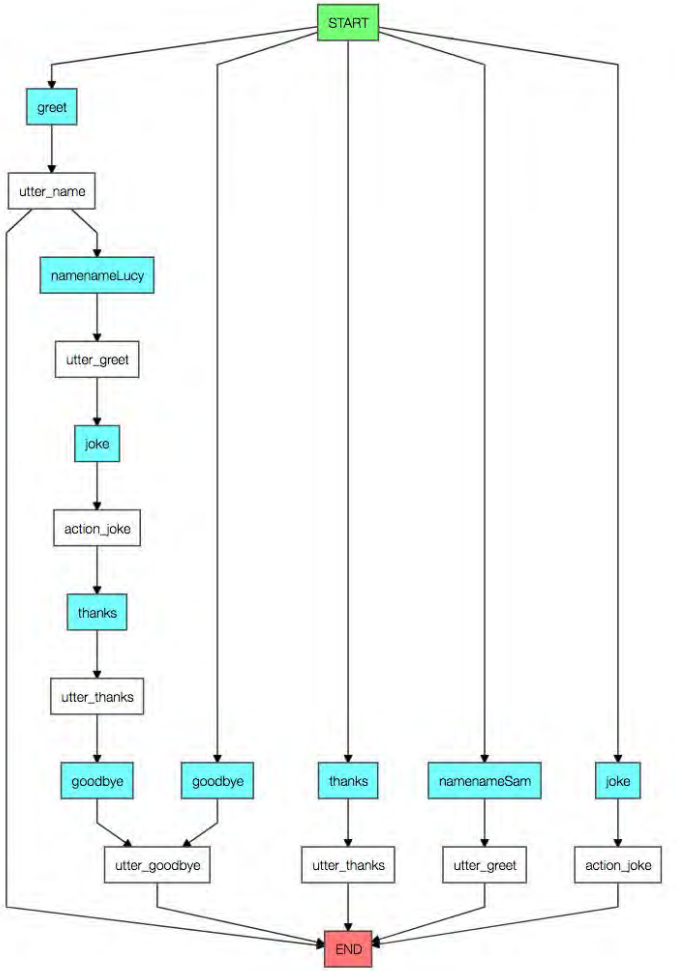


inovex

Rasa Core: Trainingsdaten

Stories => Vektordaten

- generiere Graph
- tracke Features: Intents, Entities, Slotwerte, letzte Aktion
- erzeuge Featurevektoren: One-Hot oder BoW (Labelnamen)
- berücksichtige Vergangenheit



inovex

Rasa Core: Training

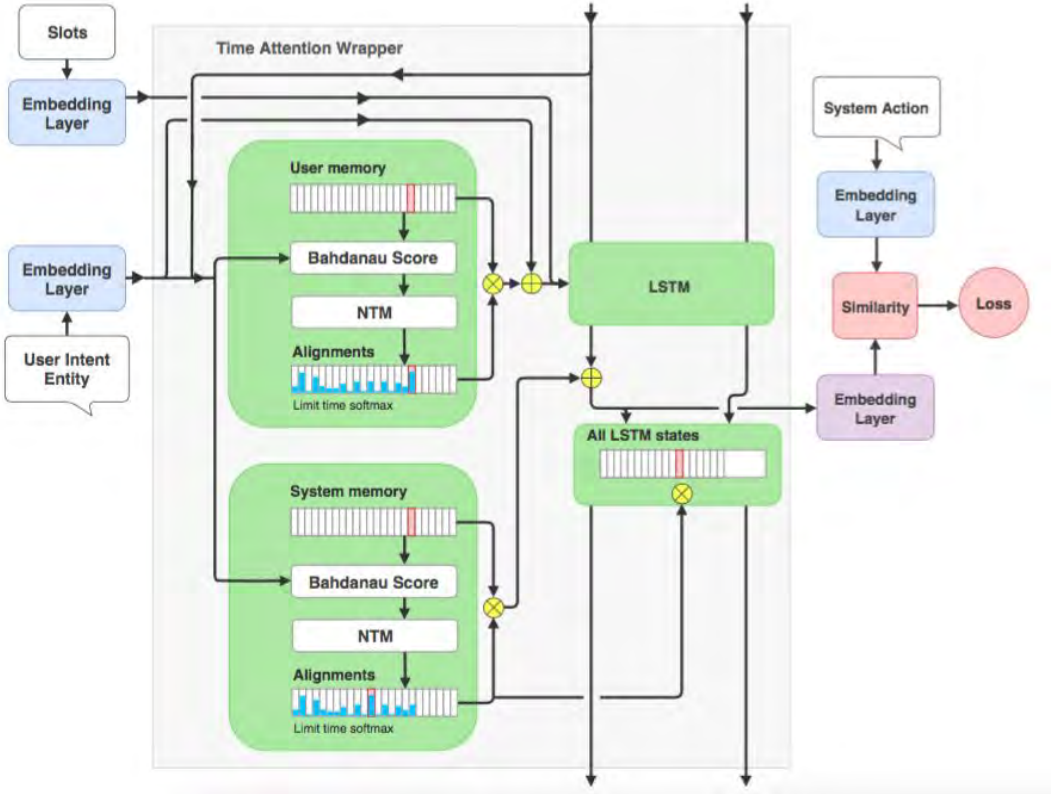
Policies:

- Memoization:
  - Exakter Match auf Trainingsdaten
- Keras: Neuronales Netz
  - Default: LSTM
- Recurrent Embedding Dialogue
  - Embeddings + Attention!

=> **Grundidee**: Typische Verläufe lernen, wichtige Informationen "merken"

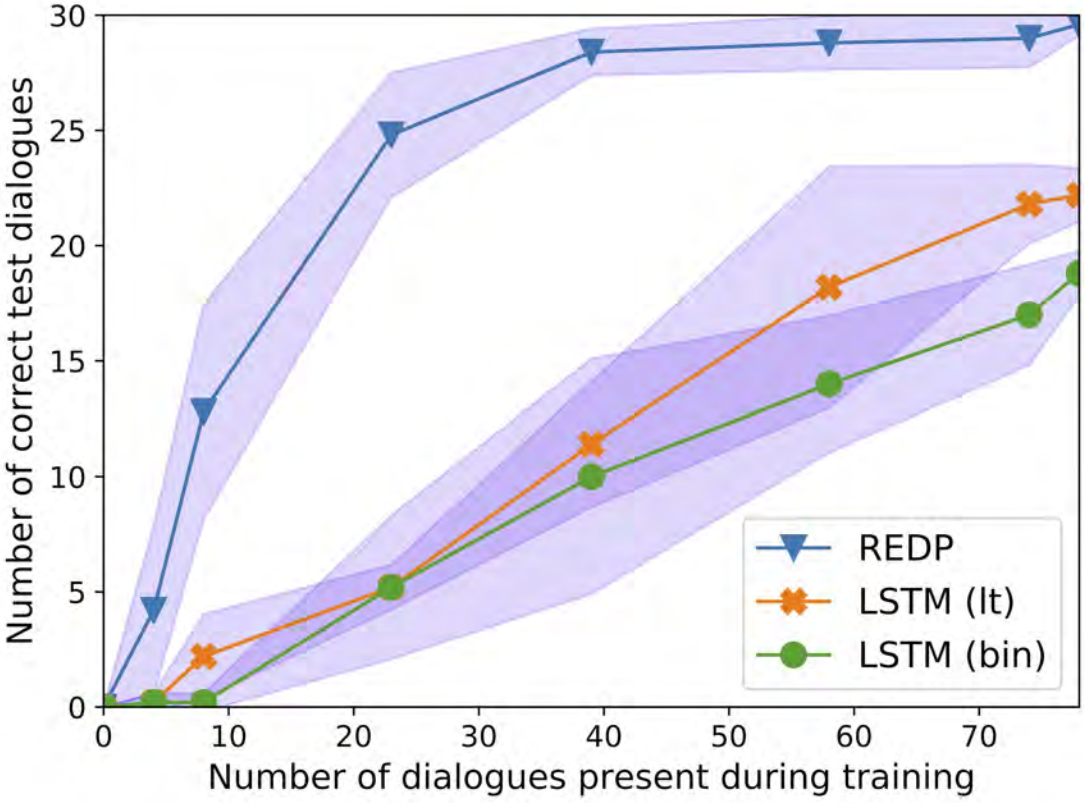
Rasa Core in Aktion

Rasa Core: Recurrent Embedding Dialogue



https://blog.rasa.com/attention-dialogue-and-learning-reusable-patterns/

Rasa Core: Recurrent Embedding Dialogue



inovex

ML to the Rescue (?): Fazit

+ Weniger explizites Pattern-Matching
+ Verstehen: Semantische Ebene
+ Reagieren: "Gedächtnis"

- Regelhunger => Datenhunger
- Supervised Learning: gelabelte Daten
- hohe Modellqualität nicht trivial

- Weltwissen/Kontextinformationen auch mit ML nicht gratis
- Gute Aktionen wollen implementiert werden

Vielen Dank

Anna Weißhaar
Search Engineer

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

anna.weisshaar@inovex.de
0173 3181 097