

Projektplan Text Mining Praktikum WiSe 18/19 - Rasa Chatbot

Lucas Schons, Lukas Gehrke, Jonas Wolff, Leonard Haas, David Fuhry

28.11.2018

Gruppenmitglieder

Name, Matrikel-Nummer

- Jonas Wolff, 3720558
- Leonard Haas,
- David Fuhry, 3704472
- Lukas Gehrke, 3757499
- Lucas Schons, 3711400

Aufgabenteilung Rasa-Chatbot

- **David Fuhry**
 - Beschaffung von Wikipedia-Daten zu Physikern
 - Weiterverarbeitung der Daten in R
- **Jonas Wolff**
 - Pflegen eines Rasa-Beispiel-Setups im repo
 - Auseinandersetzung mit Rasa-Slots für Erkennung von Namen in Intents
- **Leonard Haas**
 - Weiterverarbeitung der Daten in R; Fokus auf Cleaning
- **Lucas Schons**
 - Weiterverarbeitung der Daten in R; Fokus auf Named Entity Recognition mit dem Stanford NER
- **Lukas Gehrke**
 - Weiterverarbeitung der Daten in R; Fokus auf Extraktion von Geburtsdatum für Intent *Birth*
 - Beschäftigung mit Antwortgenerierung in Rasa Core

Auflistung der Aufgaben und beabsichtigter Lösungsschritte

Allgemeine Zielsetzung und Aufteilung

In dem Projekt soll die Frage geklärt werden, inwieweit Chatbots in der Lage sind, einerseits natürlichsprachliche Anfragen zu verstehen und andererseits Antworten aus einem natürlichsprachlichen Korpus zu generieren. Dazu soll sich mit der Open-Source Software Rasa zur Erstellung von Chatbots auseinandergesetzt werden.

Es soll am Beispiel eines englischsprachigen Chatbots, der Fragen zu berühmten Physikern beantworten kann, gearbeitet werden.

Die Aufgaben werden grob in zwei Bereiche aufgeteilt: Die Akquise und Extraktion von Daten und die Konfiguration des Rasa-Chatbots.

Gewinnung und Information Extraction zu Physiker-Daten

Zielsetzung

Für den Chatbot sollen Daten zu Physikern auf Englisch gewonnen werden. Als Quelle soll Wikipedia genutzt werden. Von dort wurde bereits eine Liste mit etwa 1000 Physikernamen gewonnen. Zu den Namen sollen die Wikipedia-Artikel als HTML akquiriert werden. Der so gewonnene Daten-Korpus soll dann in R für die Intents, mit denen der Chatbot arbeiten soll (siehe unten), aufbereitet werden. Im Idealfall werden R-Scripts geschrieben, die fertige Rasa-Trainingsfiles ausgeben.

Aufgaben

- Englischsprachige Wikipedia-Einträge zu berühmten Physikern als Korpus speichern
- Die Daten bereinigen (Entfernung von Tags, Hyperlinks etc)
- Für die Intents Skripte erstellen, um jeweils benötigte Informationen zu extrahieren.
 - Dazu gegebenenfalls weitere Aufbereitung (Named Entity Recognition, Part of Speech Tagging, Tokenization)
- Gewonnene Informationen in ein zu Rasa kompatibles Datenformat bringen
- Automatisierungspotenziale in den oben genannten Aufgaben bestimmen
 - Rausfinden, inwieweit die Erzeugung von Trainingsdaten vom Bot selbst übernommen werden kann

Arbeitsstand

- Es sind bereits 983 Physikernamen gewonnen worden:

```
Jules Aarons
Ernst Karl Abbe
Derek Abbott
Hasan Abdullayev
Alexei Alexeyevich Abrikosov
Robert Adler
Stephen L. Adler
Franz Aepinus
...
```

- Die Wikipedia-Artikel zu allen Namen sind in einem XML-File gespeichert.

```
<page>
<title>Jules Aarons</title>
<ns>0</ns>
<id>48756809</id>
<revision>
  <id>867484885</id>
  <parentid>866458180</parentid>
  <timestamp>2018-11-06T01:00:04Z</timestamp>
  <contributor>
    <username>MopTop</username>
    <id>5027336</id>
  </contributor>
  <comment>citations</comment>
  <model>wikitext</model>
  <format>text/x-wiki</format>
...
```

- Es wurde ein R-Skript geschrieben, das Namen und zugehörige Wikipedia-Artikel in einem .csv File speichert

```
library(xml2)
```

```
data <- read_xml("../data/Wikipedia-20181120103842.xml")
```

```
title.nodes <- xml_find_all(data, "../title")
```

```
titles <- sapply(title.nodes, xml_text)
```

```

text.nodes <- xml_find_all(data, ";//text")

texts <- sapply(text.nodes, xml_text)

df.out <- data.frame(Title = titles,
                     Text = texts)

saveRDS(df.out, "../data/texte.RDS")

write.table(df.out, "../data/texte.csv")

```

Konfiguration des Rasa-Chatbots

Anmerkung: Dieser Aufgabenbereich wird in der ersten Phase des Projekts (Dezember 2018) hinter die Datenakquise zurückgestellt. Der Bot soll zunächst so einfach wie möglich implementiert werden.

Zielsetzung

Es soll ein Rasa Chatbot entwickelt werden, der auf Englisch Fragen zu Physikern beantwortet. Dazu sollen die Rasa-Technologien Rasa NLU (Natural Language Understanding) und Rasa Core verwendet werden. Rasa NLU soll dafür genutzt werden, dass Rasa die Fragen und dahinterliegenden Intents (Absichten) eines Chatbot-Nutzers versteht. Rasa Core ist für die Antwortgenerierung zuständig und steuert den Gesprächsverlauf. Um beide Teile erfolgreich zu implementieren, soll sich mit der Dokumentation von Rasa auseinandergesetzt werden. Besonderes Augenmerk liegt auf der Einspeisung der aufbereiteten Physikerdaten in den Rasa-Bot. Im Laufe des Projekts soll zunächst ein Chatbot erstellt werden, der durch die Teammitglieder trainiert wird und einfach Fragen beantworten kann. (5 Intents siehe unten). Im weiteren Verlauf des Projekts soll geklärt werden, inwieweit der Bot selbstständig Trainingsdaten aus dem Korpus erstellen kann. Dazu sollen zB Rasa Custom Actions über `rasa_core_sdk` oder die Erstellung von Bot Antworten über einen externen Server in Betracht gezogen werden.

Aufgaben

- Lauffähige Version des Rasa-Bots erstellen
- Intents für den Bot ausformulieren
- Den Bot mit den erstellten Intents trainieren
- Slots in den Intents nutzen, damit der Bot zwischen verschiedenen Entitäten unterscheiden kann
- Den Bot mit aufbereiteten Physikerdaten trainieren
- später: Möglichkeiten zum selbstständigen Training des Bots sichten

Arbeitsstand

- Der Rasa Bot läuft auf den Notebooks aller Teilnehmer in `pip` oder `miniconda`-Umgebungen
- Für den Physiker-Prototypen sollen folgende Intents implementiert werden:
 - *birth* - "Where and when was **\$name** born?"
 - *isAlive* - "Is **\$name** still alive?"
 - *education* - "Where did **\$name** go to school?"
 - *researchArea* - "What did **\$name** discover?"
 - *hasNobelPrize* - "Did **\$name** win the Nobel Prize?"

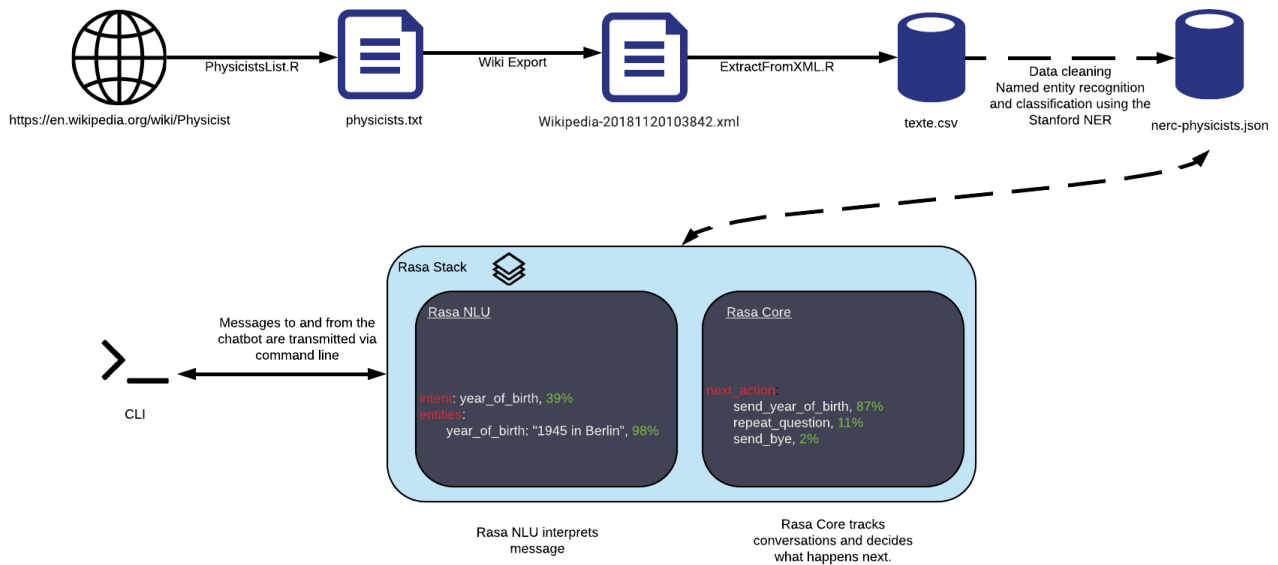


Figure 1: Rasa Chatbot Architektur

Architekturdiagramm

Absehbare und mögliche Probleme sowie Lösungsansätze

Problem 1: Daten sind nicht vorhanden

Da sich bei den Physikern Nicht alle Wikipedia Seiten von Physikern enthalten alle gewünschten Daten. Beispielsweise könnte es sein, dass nicht immer das Geburtsjahr oder der Geburtsort bekannt sind.

Lösung

Standardsätze für nicht vorhandene Daten, wie "X/Y ist nicht bekannt"

Problem 2: Aufsetzen von Rasa gibt Probleme

Da Rasa von Tensorflow abhängt und Tensorflow noch kein Python3.7 unterstützt ist Python3.6.6 die letzte unterstützte Version.

Lösung

Die einfachste Lösung wäre Python auf Version 3.6.6 zu downgraden. Dies kann jedoch zu Problemen vor allem in Linux Systemen führen, da Python für viele Systemprozesse benötigt wird.

Besser ist es eine virtuelle Python Umgebung mit Programmen wie Conda zu erstellen. Damit kann gezielt Python3.6.6 mit allen benötigten Packages installiert werden, unter denen man dann Rasa laufen lassen kann.

Zeitplan

Phase 1 bis zum 30.11

- Recherche RASA
- Daten sichten
- Generieren der Daten (bspw. durch crawling)
- Intents formulieren
- RASA aufsetzen

- Klassifikations-Algorithmen auswählen

Phase 2 bis zum 14.12

- Daten aufbereitet haben
- Prototypen erstellen
- Leitfragen (teilweise) beantwortet

Phase 3

- Feedback einarbeiten
- Finale Version entwickeln