Astronomy & Astrophysics manuscript no. output
July 2, 2020

Projektreport Data Exploration Project

Tante Elisa

Johannes Deufel, Jannik Fischer, Simone Marx and Simon Scapan

Duale Hochschule Baden-Württemberg Mannheim, Coblitzallee 1-9, 68163 Mannheim

Abgabe 17. Juli 2020

ABSTRACT

Vorgabe vom Wehner: min 3 Seiten und max 4 Seiten (jeweils ohne Abbildungen und Anhang), zitierte bücher müssen unten in die liste eingetragen werden und dann können sie wie in Kapitel "ursprüngliche ideen" referenziert werden , Alles was nach der Prozentzeile kommt, ist erstmal unwichtig

Key words. Seq2Seq – Chatbot – RNN

1. Einleitung

Idee, Konzept, Thema und Motivation beschreiben außreichend auf wirtschaftlichen Kontext eingehen

- erste Ebene
- Sprachassistent
- Tante Eliza
- normale sprachliche Unterhaltungen führen
- anwendbar auf jedem Endgerät
- einfache Bedienung -> Sprachsteuerung
- bezahlbar

In Work:

Der Mensch ist ein soziales Wesen, welches stets bemüht ist, in einer Gruppe zu agieren. Durch die globale Pandemie des Coronavirus stellen die Regierungen vieler Staaten drastische Maßnahmen. Um das Virus einzudämmen wird die Bevölkerung angehalten, soziale Kontakte zu meiden. Vor allem Rentner und pflegebedürftige Menschen leiden an den Folgen des Social Distancing, da diese als Risikogruppen ihre sozialen Kontakte auf ein Minimum reduzieren müssen. Es ergibt sich eine Marktlücke bei älteren Menschen, welche menschliche Kontakte haben möchten, jedoch wegen dem Virus keine Möglichkeit mehr dazu haben. Um die Situation bei den betroffenen Personen zu verbessern, wird ein Sprachbot entwickelt. Damit dieser bei der Zielgruppe gut vermarktet werden kann, ist es wichtig, dass die Anwendung einfach zu bedienen ist. Da in der Zielgruppe noch ein geringes Verständnis von technischen Anwendungen bzw. sogar eine Hemmschwelle herrscht, soll die Distribution vorwiegend über Business Customers, also über Pflegeheime geschehen. Der Sprachbot soll dabei über maschinelles Lernen sinnvolle Antworten auf die kurzen sprachlichen Konversionen des Users geben. Auch aus wirtschaftlichen Gesichtspunkten ist es wichtig, die Anwendung möglichst kostengünstig anzubieten und dennoch einen angemessenen Gewinn zu erwirtschaften.

Der vorliegende Projektbericht beschreibt unser wissenschaftliches Vorgehen bei der Durchführung unseres Projektes "Tante Eliza". Nachfolgend wird sowohl auf die bereits existierenden Wissenschaftlichen Werke im Bereich der Sprachbots eingegangen, als auch unsere Methoden und Bibliotheken empirisch untersucht. Zudem werden auch unsere Ergebnisse vorgestellt und kritisch hinterfragt.

:In Work

2. Related Work

Es existiert verschiedenste Literatur, die auf die Hintergründe und die Funktionen der verschiedenen Methoden der Computerlinguistik eingehen und diese erforschen. An dieser Stelle wird besonders auf die Theorie eines Deep-Learning Modells in der Computerlinguistik eingegangen, welches in der folgenden Arbeit verwendet wird. Hier können wir, wenn wir die theoretischen Papers gelesen haben grob auf den Inhalt eingehen, erklären was unter LSTM, RNN, Seq2seq und Teacher Forcing zu verstehen ist. Evtl ist dieses Buch auch hilfreich: Goyal (2018)

3. Verwendete Technologien und Bibliotheken

3.1. Systemarchitektur

Die Systemarchitektur ist wie in Abbildung 1 zu sehen aufgebaut. Sie besteht aus einem Client, dem React Frontend, einer Verbindung zum Backend via Flask und Python als Backend, welches den Chatbot beherbergt.

Der Client setzt eine Anfrage ab, welche er in gesprochener Sprache an das React Frontend sendet. Das Frontend empfängt die Anfrage und verarbeitet zuerst die Sprache und konvertiert diese Mithilfe von Mozilla SpeechRecognition SpeechRecognition (2020) in einen String. Die Begründung für diesen Schritt und auch dessen Position ist, dass ein String einfacher in der Handhabung ist als eine Soundfile, welches weiter verarbeitet wird. Der erzeugte String wird dann über die Backendconnection an das Backend gereicht. Hier wird es dem Chatbot übergeben. Dieser verarbeitet den Input und erzeugt einen Output, welcher wieder zurück an das Frontend gereicht wird. Im Frontend wird dieser String nun mit SpeechSynthesis SpeechSynthesis (2020) als Audio Feedback an den User zurück gegeben.

Unterstützt wird die Systemarchitektur von Docker. Das Projekt ist in zwei Container aufgeteilt, Frontend und Backend.

Der Frontend Container kann auf dem Port 3000 und der Backend Container auf Port 5000 angesprochen werden. Zum Starten des Systems wurde eine Docker-Compose Datei hinzugefügt, welche beide Container erstellt und lädt. Um die Software also nutzen zu können muss man lediglich den Docker-Compose im Wurzelverzeichnis erzeugen und starten. Näheres ist dem Readme Readme (2020) zu entnehmen.

3.2. Frontend

Das Frontend wurde auf der Basis von React aufgebaut. React bietet eine Grundstruktur und eine Lauffähige Webapplikation. Die Struktur des Frontend bei dieser Anwendung ist einfach gehalten. Sie besteht aus der Wurzel (App.js) und den beiden zweigen Chatbot und Landingpage. Die Landingpage beinhaltet die Startseite samt Animierten Log und einer Schaltfläche, welche einem zur Seite Chatbot gelangen lässt. Hier hat man zwei Schaltflächen, eine um eine Anfrage in Sprachform einzugeben und eine um die Antwort des Chatbots zu erhalten. Der Datentransfer hin zum Backend wird über die Datei Backendconnection realisiert.

3.3. Backend

Das Backend dockt über Flask an die React Anwendung an. Im Backend wird die Sprache Pyhon verwendet. Sobald eine Anfrage des Frontend eintrifft wird der gelieferte String in der Datei app.py aufgegriffen und an den Chatbot weiter gereicht. Dessen Response wird wiederum als String an das Frontend weiter geleitet, welches dann die weitere Bearbeitung des Prozesses vornimmt.

Der Chatbot ... hier setzen dann Simone und Johannes an ...

- Tenserflow Modell -> neuronales Netz
- Datasets: Cornell Movie -> Dialogue Corpus, Ubuntu Dialogue Corpus v2.0

Referenziere außerdem die Abbildung 3

3.4. Verworfene Ansätze

3.4.1. Weitere Methoden der Computerlinguistik

In der Computerlinguistik bestehen zur Implementierung eines Sprachbots neben des verwendeten Deep-Learning Ansatzes weitere Methoden, die den Sinn der Computerlinguistik, in diesem Falle, die Eingabesequenz in allen Einzelheiten zu verstehen und genauso zu reagieren wie ein Mensch (Vgl. Goyal 2018, S. 16), verfolgen. So bieten sich beispielsweise ein Naive Bayes Klassifikator oder ein Entscheidungsbaum an, der durch die Struktur des Wortes die jeweilige Wortart erkennt (Vgl. Bird 2009, S. 245). Darauf basierend kann im weiteren Verlauf der Verarbeitung der Sequenz beispielsweise mit einer vereinheitlichten Struktur gearbeitet werden. Unter genauerer Betrachtung der nötigen Methoden fällt jedoch auf, dass zahlreiche alternative Methoden notwendig sind, um nur mit jenen den Inhalt einer Sequenz zu extrahieren und dabei auf ein vergleichbares Ergebnis zu gelangen, wie es unter Verwendung des Deep-Learning Ansatzes der Fall wäre. Zudem bräuchte es einen sehr großen vielfach gelabelten Datensatz, um die einzelnen Modelle zu trainieren, was wiederum mit Hilfe von Deep-Learning wesentlich einfacher umgesetzt werden kann, da hier nur ein Datensatz mit einem Label, nämlich Frage oder Antwort benötigt wird. Dies sind Gründe, weshalb in dieser Arbeit stattdessen mit den Deep-Learning Methoden gearbeitet wird.

3.4.2. Weitere verworfene Ideen, auch bzgl. Deep Learning

4. Ergebnisse

Ergebnisse darstellen, die wir mit dem finalen Sprachbot und vielleicht vorgänger versionen erzielt haben

5. Bewertung der Ergebnisse

Kritische Bewertung der Ergebnisse: Was hat (nicht) funktioniert und warum?

6. Schlusswort

7. Anhang

(wie ist der Code auszuführen und was gibt es zu beachten?): usage of docker?

7.1. Anmerkungen zum Quellcode

- requirements
  - python
  - pycharm IDE
  - create project
  - virtualenv
  - flask
- split code into 3 parts:
  - app.py — the entry & exit point to our application
  - service.py — converts the request into a response.
  - models.py — handles everything that involves a Database.

...

7.2. Abbildungen

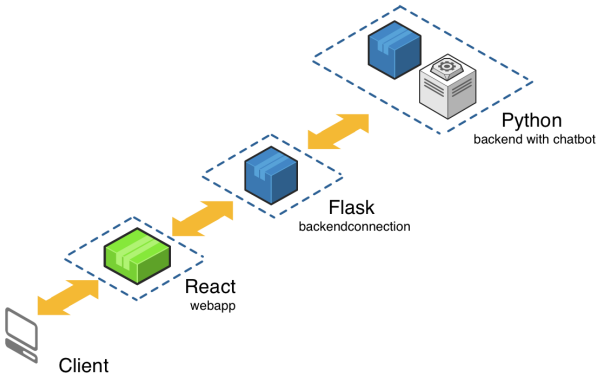

Fig. 1: Systemarchitektur

References

Bird, Steven; Klein, Ewan; Loper, Edward. 2009, Natural Language Processing with Python (O'Reilly Media Inc, Sebastopol) 504

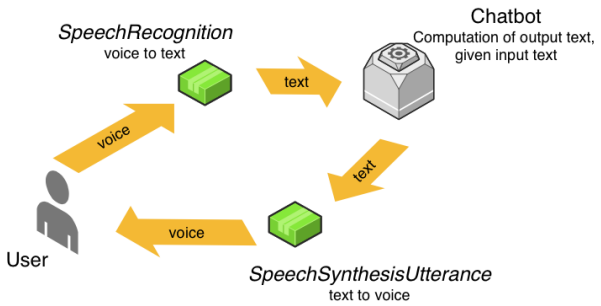

Fig. 2: Sprachverarbeitung

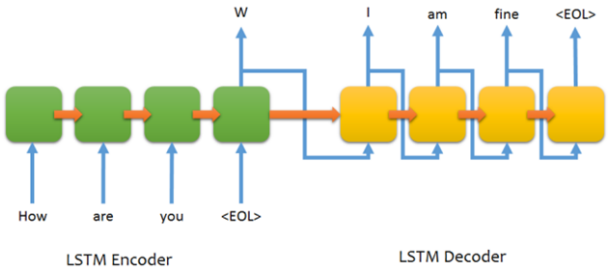

Fig. 3: LSTM Encoder und Decoder

Goyal, Palash; Jain, Karan; Pandey, Sumit. 2018, Deep Learning for Natural Language Processing: Creating Neural Networks with Python (Apress, Berkeley) 277

https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition (zugriff 02.07.2020 - 19:00)

https://developer.mozilla.org/de/docs/Web/API/SpeechSynthesis (zugriff 02.07.2020 - 19:00)

README.md