# RasaGdKi

Zuerst env erstellen mit python 3.7.2

Wir hatten auch das neuste und 3.8 verwendet. Allerdings gab es hier immer irgendwelche pakackes, welche sich nicht mit den neueren Versionen verstanden haben.

1. conda create -n rasagdki python==3.7.2

env starten. Dies hat auch nicht in der Powershell funktiniert. Nach langen ausprobieren haben wir dann festgestellt, dases wirklich an der powershell liegt und wir sind auf den cmd terminal umgestiegen

2. conda activate rasagdki

Auch hatten wir mit pip3 sowie mit pip unsere Porbleme. Am Ende hat es am ebsten mit pip 20.2 funktiniert.

3. pip install --upgrade pip==20.2

Installation von rasa

4. pip install rasa

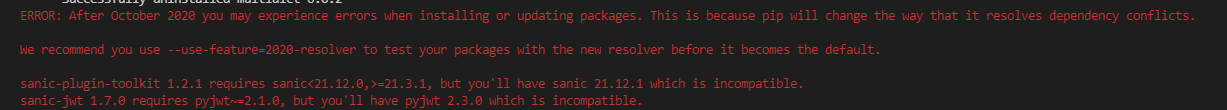
Jetzt werden erst einmal die ganzen Fehler ausgemerzt, da sanic und tensorflow auf einem zu neuen stand sind.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Zuerst wird sanic downgradet

Das hat dann jedoch nicht funktioniert. deshalb habe ich noch einmal rasa gelöscht und es doch mit pip3 versucht. Und es kam derselbe Fehler



5. pip uninstall rasa  
  
Auf stackoverflow haben wir folgende lösung gefunden

6. pip install rasa --use-feature=2020-resolver

Rasa wurde jetzt erfolgreich isntalliert.

Wir richten das directory ein mit dem Befehl

7. rasa init

Da die Configuration der yaml daten immer invalide war und wir ihn durch einen compiler jagen mussten haben wir die Erweiterung YAML von Red Hat installiert  
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Die NLU Daten sind ersteinmal automatisch auf englisch deshlab haben wir alles übersetzt und die confic auf de gesetzt

Dann wurde der bot neu trainiert

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Install spacy on rasa

Jetzt wird spacy installiert

python -m pip install spacy==3.0.6

python -m rasa init

python -m spacy project clone pipelines/ner\_demo\_replace

cd ner\_demo\_replace

python -m spacy download de\_core\_web\_lg

python -m spacy project assets

python -m spacy download de\_core\_web\_lg

python -m spacy project run

Ging wiederum nur mit der Powershell

python -m spacy project run all

python -m spacy project run package

python -m pip install packages/de\_ner\_demo\_replace-0.0.0/

Es wird über das spacy clone command geladen load-able via the spacy.load command.

rasa train nlu

Cd..

Jetzt muss noch die pipeline in rasa konfiguriert werden  
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Theoretisch ist die spacy nlu jetzt angebunden. Auf englisch funktniert das auch super. Allerdings sobald ich sie auf deutsch stelle wird der Assistent dumm!

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Einfache Discord connection py

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Discord

Zuerst Discord als py starten

Dafür die requests für discord installieren

Pip install discord.py python\_dotenv pytz requests

Dann discord file von Discord einfügen

Dann copy paste

Pip3 install requests 🡪 Als admin ohne vsc ausführen

Credinonals den connector activieren über myio input  
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Handler zwischen discord und rasa  
Ein Bild, das Text, Screenshot, Monitor, Bildschirm enthält.

Automatisch generierte Beschreibung

Enpoints webhook activ schalten



Costsum Channel.py Webhook settings über senic 🡪 Handler

Rasa starten mit rasa run. Jetzt kann über discord mit dem Rasa assistenten geredet werden.

Über den befehl exit kann die discord connection beendet werden.

Ein Bild, das Text, Monitor, Bildschirm, Katze enthält.

Automatisch generierte Beschreibung

Google calender api

pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib

**Anleitung um ClientSecret zu erstellen**

Unter Google Cloud Plattforms Anmeldedaten ersttellen. Brerechtigung <https://www.googleapis.com/auth/calendar> geben und OAuth 2.0-Client erstellen.

Dann OAuthClient runterladen und unter den Namen client\_secret.json ind order actions verschieben.

Dann CreateFirstToken.py ausführen