Recorder Bot Dokumentation

🡪 angefangen zusammenzufassen was in den verschiedenen Teilen des Codes passiert

**Main.py**

1. Config object wird erstellt

* Enthält die Informationen aus der confi.yaml Datei
* Diese Informationen können als Atribute aufgerufen werden, z.B.: Config.sequence\_model\_path

1. Storage object „store“ erstellt

* Enthält informationen wo / wie Nachrichten aus dem Chat gespeichert werden soll
* Message\_path aus config datei: „./store/messages.json“
* Methoden zum Speichern von Nachrichten
* Nutzt TinyDB database

…

1. AsyncClient object erstellt

* Benötigt um recorder bot mit matrix server zu verbinden
* AsyncClient bietet viele methoden um mit dem Server zu kommunizieren

1. Callbacks zum client hinzugefügt

* Eigene Methoden wie mit unterschiedlichen „events“ umgegangen sollen werden zum client „hinzugefügt“
* Methoden sind in callbacks.py definiert
* Recorder bot reagiert entsprechend wenn eines dieser Events auftritt:
  + Nachricht wird in einen Room geschickt
  + RecoderBot wird in einen Raum eingeladen
  + Eine Nachricht kann nicht entschlüsselt werden
  + Eine Nachricht mit „Unknown“ inhalt, z.B. emojis

**Callbacks.py**

* Language Class
  + Nimmt die entsprechende Datei aus dem language\_files Ordner (DE oder EN)
  + Der Rückgabe Text des Recorder Bots für die entsprechenden Situationen wird in einem Dictionary (texts) definiert.
  + Ein Language object wird als Atribut der Callback Class instanziiert sodas auf das texts dictionary zurückgegriffen werden kann
* **Callback Class** 
  + Methoden der Callback Class definieren wie der Recorder - Bot mit unterschiedlichen Aktionen / Nachrichten im Chat umgeht
  + Objekt wird genutzt um diese Methoden zum async client hinzuzufügen
* Callback Methoden:
* message Methode
  + Verhalten/Reaktion des Recoder-Bots wenn eine Nachricht in den Element-Raum geschickt wird
  + Nachrichten die nicht mit „:“ beginnen
    - Nachricht wird eingestuft als Problem, Ursache, Lösung oder Other (Prediction)
    - Nachricht wird gespeichert in der messages.json Datei im Store Ordner
    - Prediction wird als Nachricht in den Raum geschickt
    - 🡪 weiter wenn decryption funktioniert
* Invite Methode
  + Verhalten des Recorder-Bots wenn der Bot in einen Raum eingeladen wird
  + Gibt Rückgabe darüber ob Raum erfolgreich beigetreten
* \_get\_response
* \_reaction
* Decryption\_failure
* Unknown

Intelligence

* Methoden um Prediction zu treffen
* Sentence Class Predictor
* Token Class Predictor