Informatik
Hauptcampus

H O C H
S C H U L E
T R I E R

Der Coaching Bot

The Coaching Bot

Maximilian Wellenhofer

Master-Projektstudium

Betreuer: Prof. Dr. Georg Schneider

Zürich, 28.02.2022

Kurzfassung

Provisionierung eines Bots, der die OnBoarding-Phase eines Coaching Programms automatisiert.

In der Kurzfassung soll in kurzer und prägnanter Weise der wesentliche Inhalt der Arbeit beschrieben werden. Dazu zählen vor allem eine kurze Aufgabenbeschreibung, der Lösungsansatz sowie die wesentlichen Ergebnisse der Arbeit. Ein häufiger Fehler für die Kurzfassung ist, dass lediglich die Aufgabenbeschreibung (d.h. das Problem) in Kurzform vorgelegt wird. Die Kurzfassung soll aber die gesamte Arbeit widerspiegeln. Deshalb sind vor allem die erzielten Ergebnisse darzustellen. Die Kurzfassung soll etwa eine halbe bis ganze DIN-A4-Seite umfassen.

Hinweis: Schreiben Sie die Kurzfassung am Ende der Arbeit, denn eventuell ist Ihnen beim Schreiben erst vollends klar geworden, was das Wesentliche der Arbeit ist bzw. welche Schwerpunkte Sie bei der Arbeit gesetzt haben. Andernfalls laufen Sie Gefahr, dass die Kurzfassung nicht zum Rest der Arbeit passt.

Abstract

The same in English.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

1

Einleitung und Problemstellung

1.1 Motivation

Warum einen Coaching Bot bauen?

Viele junge Menschen die nach einem abgeschlossenen Studium in die Arbeitswelt einsteigen möchten, haben keine oder wenig Ahnung davon, wie sie sich vorbereiten sollen oder welche Schritte erforderlich sind, um einen erfolgreichen Start zu schaffen. Persönliche Beratungsleistungen und speziell Einzel-Coaching können dabei helfen, sich effektiv vorzubereiten und einen erheblichen Wettbewerbsvorteil bieten, sind aber für Berufseinsteiger ohne signifikante finanzielle Mittel miest weder zugänglich noch erschwinglich. Das sind in Gründung befindliche Unternehmen -wavehoover consult AG- aus Zürich hat es sich daher zur Aufgabe gemacht, in dieser Hinsicht einen Beitrag zu leisten. Interaktionen mit jungen Absolventen auf klassischen Websiten waren bis dato wenig erfolgversprechend. Aus diesem Bedürfnis heraus und aufgrund meiner beruflichen Erkenntnis, dass die erste Kontaktaufnahme und die Vorbereitung auf die erste Sitzung meist sehr ähnlich oder gar einem Skript folgend ablaufen, ist die Idee zu einem standardisierten und automatisierten OnBoarding-Prozess entstanden. Vielen junge Menschen wenden sich ab vom traditionellen Web-Browser und sind stärker in Messenger-Diensten wie WhatsApp, Signal, Telegram oder vor Allem auf Social Media Portalen wie Instagram, Facebook, TikTok, SnapChat, etc. zu finden. Um sich auf diese Zielgruppe einzulassen, hat man sich entschieden, einen Bot zu programmieren, der Informationen vom User abfragt und Der Instant Messenger Telegram (https://telegram.org/) ist (neben vielen anderen) besonders unter jungen Menschen, ein beliebtes Kommunikations- und Interaktionsmedium, das Funktionen weit über das einfache Nachrichten-austauschen hinaus abdeckt. Ziel des Projekts ist es, primär jungen Absolventen die Möglichkeit zu bieten, sich auf eine erste Live Coaching Session vorzubereiten und nicht “völlig blank” in eine bezahlte Beratungsleistung zu investieren, ohne überhaupt bereits zu wissen, was sie erwartet oder wie man sich vorzubereiten hat, um das Meiste für sich selbst heraus zu holen.

2

VerwandteArbeiten

Schauen Sie nach – ob es bereits existierende Arbeiten und Systeme, die ähnliche Probleme bearbeiten gibt. Dies ist eigentlich das „Herzstück“ einer wissenschaftlichen Arbeit, weil man seine eigenen Beiträge zum aktuellen State-of-the-Art in Verbindung setzt. Beschreiben Sie (wenn möglich) mindestens 2 dieser Arbeiten. Danach, sozusagen als Resümee können Sie dann sagen, was Sie anders (besser!) machen wollen (also hier kommen Argumente hin, warum Sie nicht ein existierendes System nehmen, sondern selber eins programmieren und was Sie an Ideen übernehmen).

3

Grundlagen

Die folgenden Sprachen und Systeme dienen als Grundlage für die im Rahmen dieses Projekts entwickelte Applikation.

3.1 Python

Der größte Teil der Applikation ist in Python 3.8.6 [**?**] geschrieben. Das war zum Stand des Entwicklungsbeginns 2021 die aktuell stabile Python-Version.

3.2 Telegram API

Die Applikation basiert auf der API des Instant Messaging Dienstes Telegram [**?**] sowie deren Extension [**?**]. Als Grundgerüst der für den Bot erforderlichen State Machine wurde der ConversationBot von Leandro Toledo et. al. genutzt. [**?**] Das Repository enthält eine Vielzahl basaler Bot-Implementierungen, die als Startpunkt für jegliche Bot-Implementierung einen guten Überblick über das Grundgerüst ein die Funktionsweise eines Bot geben.

3.3 SQLite

Zur Speicherung von Nutzerdaten wird eine simple SQLite Datenbank [**?**] genutzt.

3.4 SQLite3 API

Als Schnittstelle zwischen dem Python basierten Backend und der SQLite Datenbank nutzt die Applikation den sqlite3 Database-Connector [**?**].

3.5 HMTL

HTML bietet uns die Möglichkeit die grafische Oberfläche in einem Standard Webbrowser auszugeben. [**?**]

3.6 PHP

Die HTML-GUI wird via PHP [**?**] an die Datenbank gebunden.

3.7 Google Calendar API

Die Applikation bindet die Google Calendar API [**?**] an, um es dem User zu ermöglichen, einen Termin mit dem Anbieter zu vereinbaren:

3.8 TheCoachingBot

Schließlich findet sich der gesamte Source-Code inklusive aller Abhängigkeiten in einem öffentlichen GitHub Repository: [**?**]

4

Konzept

Zum Beispiel: - Blockbild der Architektur Ihrer Anwendung - Pseudocode für Algorithmen - mathematische Formeln - evtl. Diagramme auf hohem Abstraktionsniveau.

Abbildung 4.1: Architektur für das Projekt "Der Coaching Botäuf hohem Abstraktionsniveau

Abbildung 4.2: Konversationsfluss des Bots

5

Realisierung

Hier kommt hin, wie es gemacht haben.

5.1 Telegram Bot Framwork

5.2 Generierung Telegram Bot

BotFather

5.3 Vanilla Bot Implementierung

conversationBot

5.4 Endlicher Automat und Zustände

Aufbohren ßtates"für ComeBack-Feature

5.5 Erstellung Datenbank

SQL Commands sqlite

5.6 Anbindung Datenbank an Python

sqlite3

5.7

6

Implementierung

Kann mit 5. Zusammenfallen. Manchmal eignet es sich, 2 Abstraktionsschritte zu machen (Realisierung und Implementierung getrennt).

Generell für 5. Und 6.: Wenig Quellcode (wenn überhaupt)! Maximal 2/3 Seite und immer begleitet von Erklärungen, was zu sehen ist. Kommentare im Quellcode sind nicht ausreichend. Dies gilt für UML Diagrammen analog.

6.1 Setup

6.1.1 Entwicklungsumgebung

6.1.2 Microsoft Visual Studio Code

6.1.3 pipenv

Die Applikation nutzt den Package Manager pipenv. Dieser bietet die Möglichkeit, ein projektspezifisches Dokument über alle Abhängigkeiten hinweg zu erstellen und im Projekt selbst zu speichern. So können andere Entwickler Abhängigkeiten leicht installieren und müssen dies nicht auf Systemebene tun, wo es ggf. zu Konflikten mit anderen Projekten kommen könnte. Um alle Abhängigkeiten einzusehen und automatisch zu installieren, pipenv [**?**] installieren und das coaching$_bot/Pipfilevia$

6.1.4 Konstanten und Schlüssel

Der Coaching Bot hat einige Abhängigkeiten zu Umsystemen, die Zugangsdaten voraussetzen. Diese sind im Repository [**?**] aus Sicherheitsgründen abstrahiert.

6.1.5 Dispatcher

Dispatcher liefern Nachrichten an den User aus. Pro Bot gibt es meist einen Dispatcher. Command- und ConversationHandler werden an diesen angehängt.

6.1.6 ConversationHandlers

ConversationHandler kontrollieren den Konversationsfluss zwischen dem User und dem Bot. Pro Bot kann es mehrere ConversationHandler geben. Der CoachingBot hat aber nut einen - den "conv$_handler$".$DerConversationHandlerkoordiniertalleCommandHandle$

6.1.7 CommandHandler

CommandHandler nehmen Nutzereingaben via eines CallbackContexts entgegen, prüfen diese auf vordefinierte Kriterien und führen prädefinierte Funktionen - sog. Handler Functions aus.

6.2 Coaching Bot Herzstück - main.py - State Machine

Die main.py ist das Herzstück des Coaching Bots.

Sie importiert alle Handler-Functions, authentifiziert sich durch den entsprechenden API-Schlüssel und beinhaltet den Dispatcher, an dem wiederum Conversation- sowie CommandHandler hängen. Darüber hinaus startet sie den Bot und aktualisiert die Handler in regelmäßigen Abständen via dem Updater.

Im Folgenden werden die CommandHandler für den Coaching Bot kurz aufgeführt:

6.2.1 start und cancel - Konversation starten und stoppen

Der in dieser Applikation umfangreichste ConversationHandler umfasst zwei Commands: /startünd /cancel". Solange der Bot ausgeführt wird, lässt sich eine Konversation mit ihm über den Befehlt /startßtarten und via /cancel"beenden. Zur Funktionsweise von /startünd /cancel", siehe ßtart.pyünd "cancel.pyünter "Handler Funktionen".

6.2.2 delete - Nutzerdaten löschen

Über den Befehl /delete"wird der CommandHandler "deleteäusgeführt. Zur Funktionsweise von /delete", siehe "delete.pyünter "Handler Funktionen".

6.2.3 help - Hilfe ausgeben

Über den Befehl /help"wird der CommandHandler "helpäusgeführt. Zur Funktionsweise von /help", siehe "help.pyünter "Handler Funktionen".

6.2.4 summary - Zusammenfassung ausgeben

Über den Befehl /summary"wird der CommandHandler ßummaryäusgeführt. Zur Funktionsweise von /summary", siehe ßummary.pyünter "Handler Funktionen".

6.2.5 status - Status Quo ausgeben

Über den Befehl /status"wird der CommandHandler ßtatusäusgeführt. Zur Funktionsweise von /status", siehe ßtatus.pyünter "Handler Funktionen".

6.3 Handler Funktionen

Handler-Funktionen (siehe coaching$_bot/handler_functionsin[?])sindFunktionen, dieaufEingabenre
$Funktionenein, beschreibenderenUmfangundAufbauunderklärenihreFunktionsweise.$

6.3.1 start.py

6.3.2 birthdate.py

6.3.3 bio.py

6.3.4 cancel

6.3.5 confirmation$_mail.py$

6.3.6 email.py

6.3.7 gender.py

6.3.8 help.py

6.3.9 location.py

6.3.10 photo.py

6.3.11 states.py

6.3.12 status.py

6.3.13 summary.py

6.3.14 telephone.py

6.3.15 validation.py

6.4 Datenbank

6.5 E-Mail-Versand

6.6 Tests

6.7

6.8

6.9

6.10

6.11

6.12 Kalender

Google Cloud Console: https://console.cloud.google.com/apis/credentials/consent?project=coachin
bot-339115 Calendar API Python Quickstart Guide: https://developers.google.com/calendar/api/qu

Verify own website: https://www.google.com/webmasters/verification/home?hl=en

7

Beispiele

Hier kommen einige Bildschirmabzüge hin, damit man sich die Arbeit mit dem System vorstellen kann (falls es eine visuelle Komponente gibt). Natürlich mit kurzem erklärenden Text. Man kann sich hier auch eine Art Drehbuch überlegen, wie man mit der Anwendung umgeht und diese dann hier mit Screenshots umsetzen.

8

Anwendungsszenarien

Der Coaching Bot kann in allerlei Szenarien angewandt werden. Die in den letzten Jahren stark angewachsene Zahl an Personal Coaches kann den Bot mit basalen Programmierfähigkeiten an die eigenen Bedürfnisse anpassen. Vor allem für Nebenerwerbstätige Coaches mit einem kleinen Kundenportfolio stellt der Coaching Bot eine einfache Möglichkeit dar, Neukunden onzuboarden. Der Prozess ist unkompliziert, unverbindlich und einfach zu adaptieren.

Ausbaupotenzial

Sollte der Coaching Bot über die ersten Monate vielversprechende Ergebnisse liefern, sind folgende Ausbaustufen geplant:

1. Verteilung und Verlagerung in die Cloud für konstante und hohe Verfügbarkeit 2. Skripten und Automatisierung weiterer Coaching-Stufen. i.e. könnte die erste Session, die oft ähnlich abläuft auch vom Bot abgehandelt werden. 3. Ton-Aufnahmen für das Biography-Modul

9

Zusammenfassung und Ausblick

In diesem Kapitel soll die Arbeit noch einmal kurz zusammengefasst werden. Insbesondere sollen die wesentlichen Ergebnisse Ihrer Arbeit herausgehoben werden. Erfahrungen, die z.B. Benutzer mit der Mensch-Maschine-Schnittstelle gemacht haben oder Ergebnisse von Leistungsmessungen sollen an dieser Stelle präsentiert werden. Sie können in diesem Kapitel auch die Ergebnisse oder das Arbeitsumfeld Ihrer Arbeit kritisch bewerten. Wünschenswerte Erweiterungen sollen als Hinweise auf weiterführende Arbeiten erwähnt werden.

Nochmal kurz sagen, was Sie gemacht haben - am besten die Ziele der Arbeit aus 1.2. nochmals nennen und kurz erklären, wie sie das in Ihrem System realisiert haben. (So was wie ein "management summary)

appendix

A

Glossar

DisASTer	Distributed Algorithms Simulation Terrain, eine Plattform zur Implementierung verteilter Algorithmen [**?**]
DSM	Distributed Shared Memory
AC	Atomic Consistency (dt.: Linearisierbarkeit)
RC	Release Consistency (dt.: Freigabekonsistenz)
SC	Sequential Consistency (dt.: Sequentielle Konsistenz)
WC	Weak Consistency (dt.: Schwache Konsistenz)

B

Selbstständigkeitserklärung

☐ Diese Arbeit habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

☐ Diese Arbeit wurde als Gruppenarbeit angefertigt. Meinen Anteil habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Namen der Mitverfasser:

Meine eigene Leistung ist:

Datum

Unterschrift der Kandidatin/des Kandidaten