



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra kybernetiky

**Tvorba doporučovacího chatbota pro humanoidního
robotu Pepper**

Recommendation Chatbot for the Humanoid Robot Pepper

Bakalářská práce

Studijní program: Otevřená informatika

Specializace: Základy umělé inteligence a počítačových věd

Vedoucí práce: Mgr. Michal Vavrečka, Ph.D.

Petr Schimperk

Praha, Leden 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Schimperk** Jméno: **Petr** Osobní číslo: **487010**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra kybernetiky**
Studijní program: **Otevřená informatika**
Specializace: **Základy umělé inteligence a počítačových věd**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Tvorba doporučovacího chatbota pro humanoidního robota Pepper

Název bakalářské práce anglicky:

Recommendation Chatbot for the Humanoid Robot Pepper

Pokyny pro vypracování:

Cílem práce je tvorba nového chatbota pro humanoidního robota Pepper, který je založen na doporučovacím systému, který porovnává uživatelské preference s ostatními uživateli a hodnocením na internetu. Systém využívá vlastní bázi znalostí a dokáže vybírat témata k hovoru na základě předchozích preferencí uživatele.

1. Tvorba báze znalostí o uživateli.
2. Tvorba systému pro řízení dialogu.
3. Tvorba algoritmů pro vyhodnocování a porovnávání báze znalostí.
4. Implementace chatbota do humanoidního robota.
5. Otestování robota v reálných podmínkách.

Seznam doporučené literatury:

- [1] Cerezo, Jhonny, et al. "Building an expert recommender chatbot." 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE). IEEE, 2019.
- [2] Ikemoto, Yuichiro, et al. "Conversation strategy of a chatbot for interactive recommendations." Asian Conference on Intelligent Information and Database Systems. Springer, Cham, 2018.
- [3] Li, Yu, et al. "DEUX: An Attribute-Guided Framework for Sociable Recommendation Dialog Systems." arXiv preprint arXiv:2105.00825 (2021).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Mgr. Michal Vavrečka, Ph.D., robotické vnímání CIIRC

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **20.05.2021**

Termín odevzdání bakalářské práce: **04.01.2022**

Platnost zadání bakalářské práce: **19.02.2023**

Mgr. Michal Vavrečka, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 4.1.2022

Petr Schimperk

Poděkování

Rád bych poděkoval panu Mgr. Michalu Vavrečkovi, Ph.D. a jeho kolegyni Mgr. Gabriele Šejnové za cenné rady, připomínky a odborné vedení práce. Také bych rád poděkoval své rodině za pomoc a podporu.

Abstrakt

V dnešní době je rychlý rozvoj ve všech oblastech interakce lidí se stroji. Velkému zájmu se kromě počítačového vidění těší i vývoj konverzačních programů, převážně v podobě osobních asistentů. Konverzační programy lze obecně nazvat chatboty a jejich přirozeným cílem je co nejlépe napodobit lidskou konverzaci ve všech možných směrech. Tato práce se snaží vytvořit spojení chatbota v humanoidním robotovi se systémem na doporučování tak, aby si uživatel zábavnou a interaktivní formou rozšířil znalosti a povědomí o různých tématech. Doporučování je založeno na porovnávání názorů uživatelů, se kterými chatbot dříve mluvil.

Abstract

There is a swift development in human-computer interaction these days. Among things like computer vision, there is also considerable interest in conversational agents, predominantly in the form of personal assistants. The primary purpose of conversational agents, generally called chatbots, is to mimic human conversation in all possible ways. This thesis tries to create a chatbot inside a humanoid robot that can recommend things in a fun and interactive way to expand users' knowledge about various topics. Recommending is based on users' opinions that chatbot has formerly conversed with.

Obsah

SEZNAM ZKRATEK.....	9
1 ÚVOD.....	10
1.1 Motivace.....	10
1.2 Cíle práce.....	11
1.3 Klíčová slova.....	11
2 CHATBOT.....	12
2.1 Základní vlastnosti.....	12
2.2 Struktura chatbota.....	14
2.3 Využití chatbotů.....	16
3 ROBOT PEPPER.....	18
4 IMPLEMENTACE CHATBOTA.....	20
4.1 Moduly.....	21
4.1.1 Rozhraní s uživatelem a Pepper API.....	21
4.1.2 Dialogové texty a jazykové třídy.....	22
4.1.3 Scraping informací.....	23
4.1.4 Databáze uživatelů.....	23
4.1.5 Chatbot.py.....	25
4.1.6 Konstrukce dialogu.....	26
4.1.7 Utility soubory.....	28
5 TESTOVÁNÍ APLIKACE.....	29
5.1 Rozbor poznámek ke konverzacím.....	30
ZÁVĚR.....	32
BIBLIOGRAFIE.....	33
SEZNAM PŘÍLOH.....	35
[A] PEPPER POHLEDY.....	36

Seznam zkratek

AIML	artificial intelligence markup language
API	application programming interface
CPP	jazyk C++
CPU	central processing unit
CSV	comma-separated values
GIL	global interpreter lock
HCI	human-computer interaction
HTML	hypertext markup language
IDE	integrated development environment
IPv4	internet protocol version 4
NLP	natural language processing
NLU	natural language understanding
RAM	random access memory
ROS	robot operating system
RPC	remote procedure call
SDK	software development kit
STT	speech to text
TTS	text to speech

1 Úvod

Chatboti jsou dnes běžně nabízeným produktem například v podobě domácích chytrých asistentů. Ve velkém je lze potkat také v eshopech nebo jako první úroveň uživatelské podpory. Jejich nespornou výhodou proti běžným počítačovým programům je způsob předávání informací, ať už textově, nebo hlasem. Není potřeba učit uživatele, jak chatbota používat, chatbot by sám měl umět uživateli sdělit, co vše dokáže.

Na druhou stranu vývoj chatbota je náročný a vyžaduje znalosti ve více oborech. Největším problémem je práce s jazykem při zpracování vstupu uživatele a při vytváření korektního výstupu chatbota. Problém zpracování vstupu se dělí na dvě části, NLP a NLU¹, které se v dnešní době překotně rozvíjí, přesto je stále velmi obtížné vyřešit tento problém pro všechny (nebo i pouze pár) jazyky naráz. Problém vytváření výstupu chatbota je možné řešit předem vytvořenými texty, které ale mohou být z části chybně použity (u češtiny například zájmena a koncovky sloves), nebo pomocí jazykových modelů, které jsou poměrně novou oblastí.

Systémy na doporučování podle dříve sebraných informací o uživateli jsou dnes všudypřítomné (např. reklamy a eshopy, filmové, hudební a knižní databáze nebo služby, seznamovací stránky). Jejich základem je porovnání profilů uživatelů a odhad možných zájmů uživatele.

Tato práce se zabývá tím, jak spojit chatbota a doporučovací systém tak, aby konverzace s uživatelem byla přirozená, zábavná a interaktivní, ale zároveň bylo možné získat dostatečné množství dat pro porovnání uživatelů.

Kapitola 2 se zabývá popisem chatbotů, jejich využitím a problematikou vývoje. Kapitola 3 popisuje humanoidního robota Pepper a to, co dokáže nabídnout. Kapitola *Implementace chatbota* se zabývá vlastní prací, rozebírá přístupy použité pro jednotlivé části. Poslední kapitola 5 krátce shrnuje výsledky testování.

1.1 Motivace

Motivací tohoto projektu je propojit dnes běžně samostatné domény: chatbot a doporučovací systém. Účelem je prozkoumání nové konverzační oblasti, která není pro dnešní chatboty běžná.

¹ Natural Language Processing a Understanding - zpracování a porozumění přirozenému jazyku

1.2 Cíle práce

Cílem této bakalářské práce je tvorba chatbota s přípravou na systém doporučování, který na základě dříve získaných preferencí jiných lidí dokáže uživateli navrhnout, co by pro něj mohlo být nějakým způsobem ideální volbou. Chatbot má být vytvořen pro humanoidního robota Pepper, což umožňuje získat víc informací o uživateli a doplnit konverzaci o nonverbální prvky. Zároveň by aplikace měla být snadno rozšiřitelná pro další budoucí vývoj.

1.3 Klíčová slova

Chatbot, HCI, doporučovací systém, systém pro doporučování, konverzace, dialog, humanoidní robot, personalizace robota, Pepper.

Chatbot, HCI, recommendation system, conversation, dialogue, humanoid robot, robot personalisation, Pepper.

2 Chatbot

Chatbot neboli konverzační agent je software, jehož cílem je nabídnout nějakou formu konverzace s uživatelem. Vstupem programu je zadání uživatele ve formě textu nebo hlasu a výstupem je taková odpověď, která se snaží co nejlíže přiblížit běžné lidské konverzaci.

2.1 Základní vlastnosti

Konverzace člověka se strojem může nabývat různých rozměrů, od strohé formy po významově složitou konverzaci s prvky neverbální komunikace. Čím více vlastností konverzace chceme sledovat, tím složitější chatbot bude a analýza odpovědi uživatele bude vyžadovat více času nebo výpočetního výkonu.

Základním prvkem konverzace je rozumět předané informaci od druhé strany. V případě, kdy chatbot uživateli často nerozumí, může být uživatel frustrován nebo chatbota přestane používat. Pro chatboty s více než jedním obratem (jednou replikou uživatele a jednou chatbota) je důležité pro porozumění záměru uživatele pochopení širšího kontextu konverzace. Do kontextu konverzace patří například téma, věc nebo entita, o které je konverzace vedena, a vztahy stran konverzace k danému tématu a věci.

Dalším důležitým prvkem konverzace jsou osobnosti jednotlivých stran. Chatbot by měl být pozitivní, zdvořilý a používat formální jazyk, ale zároveň i dodržovat stereotypy, které jsou běžné pro jazyk konverzace a pro zájmovou skupinu uživatelů. Pokud to účel chatbota vyžaduje, měl by chatbot umět komplexně pochopit osobnost uživatele a přizpůsobit mu konverzaci na míru. Budování vztahu s chatbotem může významně zlepšit uživatelský zážitek. [1, 2, 3]

Table 1. Conceptual model of chatbots social characteristics. Eleven characteristics found in the literature were grouped into three main categories, depending on whether the social characteristic relates to the chatbot's conversational behavior, social protocols, or virtual representation. For each characteristic, we point out the benefits, challenges, and strategies of implementing the characteristic in chatbots as reported in the literature.

	Social Characteristics	Benefits	Challenges	Strategies
Conversational Intelligence	<i>Proactivity</i>	[B1] to provide additional information [B2] to inspire users and to keep the conversation alive [B3] to recover from a failure [B4] to improve conversation productivity [B5] to guide and engage users	[C1] timing and relevance [C2] privacy [C3] users' perception of being controlled	[S1] to leverage conversational context [S2] to select a topic randomly
	<i>Conscientiousness</i>	[B1] to keep the conversation on track [B2] to demonstrate understanding [B3] to hold a continuous conversation	[C1] to handle task complexity [C2] to harden the conversation [C3] to keep the user aware of the chatbot's context	[S1] conversational flow [S2] visual elements [S3] confirmation messages
	<i>Communicability</i>	[B1] to unveil functionalities [B2] to manage the users' expectations	[C1] to provide business integration [C2] to keep visual elements consistent with textual inputs	[S1] to clarify the purpose of the chatbot [S2] to advertise the functionality and suggest the next step [S3] to provide a help functionality
	<i>Damage control</i>	[B1] to appropriately respond to harassment [B2] to deal with testing [B3] to deal with lack of knowledge	[C1] to deal with unfriendly users [C2] to identify abusive utterances [C3] to fit the response to the context	[S1] emotional reactions [S2] authoritative reactions [S3] to ignore the user's utterance and change the topic [S4] <i>conscientiousness</i> and <i>communicability</i> [S5] to predict users' satisfaction
Social Intelligence	<i>Thoroughness</i>	[B1] to increase human-likeness [B2] to increase believability	[C1] to decide how much to say [C2] to be consistent	Not identified
	<i>Manners</i>	[B1] to increase human-likeness	[C1] to deal with face-threatening acts [C2] to end a conversation gracefully	[S1] to engage in small talk [S2] to adhere turn-taking protocols
	<i>Moral agency</i>	[B1] to avoid stereotyping [B2] to enrich interpersonal relationships	[C1] to avoid alienation [C2] to build unbiased training data and algorithms	Not identified
Personification	<i>Emotional intelligence</i>	[B1] to enrich interpersonal relationships [B2] to increase engagement [B3] to increase believability	[C1] to regulate affective reactions	[S1] to use social-emotional utterances [S2] to manifest <i>conscientiousness</i> [S3] reciprocity and self-disclosure
	<i>Personalization</i>	[B1] to enrich interpersonal relationships [B2] to provide unique services [B3] to reduce interactional breakdowns	[C1] privacy	[S1] to learn from and about the user [S2] to provide customizable agents [S3] visual elements
	<i>Identity</i>	[B1] to increase engagement [B2] to increase human-likeness	[C1] to avoid negative stereotypes [C2] to balance the <i>identity</i> and the technical capabilities	[S1] to design and elaborate on a persona
	<i>Personality</i>	[B1] to increase believability [B2] to enrich interpersonal relationships	[C1] to adapt humor to the users' culture [C2] to balance the <i>personality</i> traits	[S1] to use appropriate language [S2] to have a sense of humor

Tabulka 1 - konverzační vlastnosti chabota [1]

Tabulka konceptuálně popisuje konverzační vlastnosti, které chatbot může mít. Sloupce *Benefits*, *Challenges* a *Strategies* detailněji popisují za jakým účelem chceme dané vlastnosti implementovat, jaké jsou výzvy při jejich implementaci a co nám vlastnosti nabídnou. *Conversational intelligence* (Konverzační inteligence) je soubor vlastností popisujících schopnosti vést konverzaci plynule a umět navázat tak, aby konverzace dávala smysl a měla nějakou pointu. Zároveň obsahuje vlastnosti, které umožňují vést konverzaci proaktivně a dynamicky, a to včetně řešení případných omylů kterékoliv ze stran konverzace. Do *social intelligence* (sociální inteligence) patří znaky popisující chování chatbota v různých situacích a jeho dlouhodobý postoj k uživateli. Tako skupina řeší výzvy, mezi které patří různá negativní chování uživatele a jak na ně korektně reagovat, dále jak obecně reagovat na emoce a jak zabránit tomu, aby chatbot nezněl příliš roboticky. Poslední kratší skupinou je *personification* (personifikace), která se zabývá osobností chatbota, a to včetně smyslu pro humor a možností částečnému přizpůsobení konverzace profilu uživatele. [1]

2.2 Struktura chatbota

Programování chatbota je komplexní problém vyžadující znalosti v mnoha oborech - primárně obory počítačových věd, sekundárně psychologie a částečně i jazykové vědy. Obor, do kterého tvorba chatbotů spadá, se nazývá Human-computer interaction (HCI, interakce člověka s počítačem).

Návrh chatbota se obecně skládá ze čtyř částí: příjem vstupu uživatele, převod vstupu na strojovou reprezentaci, porozumění vstupu a syntéza odpovědi, předání výstupu uživateli.

Vstup uživatele může být buď textový nebo hlasový, který se převádí na text. U textového vstupu by měl chatbot počítat s překlepy, u hlasového se špatným určením vyřčeného slova. U některých chyb (obzvláště překlepů) je možná korekce, u zbylých pak musí chatbot uživateli oznámit, že mu nerozuměl. Disponuje-li chatbot kamerovým vstupem, pak je možné získat obraz tváře, nebo i celého těla, uživatele.

Převod textového vstupu na strojově zpracovatelná data je úměrně složitý v závislosti na následné analýze dat. Nejjednodušším zpracováním je například odstranění interpunkčních znamének nebo převod textu na malá (velká) písmena. Složitější přístupy mohou vyžadovat tokenizaci a následně tagování (značkování). Tokenizace textu je jeho rozdělení na bloky, ke kterým je možné přiřadit tagy v závislosti na jejich obecném významu a pozici ve větě (například framework NLTK¹ pro Python, nebo algoritmus TF-IDF²). Dalšími převody jsou například určení emocí z textu, hlasu nebo tváře uživatele, objektivní vlastnosti uživatele (věk, pohlaví aj.) nebo určení nálady.

Hlavní částí chatbota je pochopení vstupu a syntéza odpovědi. Nejjednodušším (někdy i bezkontextovým) přístupem k tvorbě dialogu je pattern matching (porovnání textu a vzoru) společně s předem vytvořenými odpověďmi. Toto řešení je časově náročné na přípravu, v závislosti na složitosti očekávané konverzace, ale má zanedbatelné nároky na výpočetní výkon a je jednoduché na naprogramování. Pattern matching se dá spojit i s dynamicky vytvářenými odpověďmi, to ale vyžaduje významově a syntakticky korektní syntézu odpovědi (náročnost záleží na zvoleném jazyku konverzace). Někdy lze místo konkrétní odpovědi použít nějakou obecnou, například když si chatbot není jednoznačně jistý, že správně pochopil význam vstupu. Složitějším přístupem je použití jazykových modelů (např. rekurentní neuronové sítě), jejichž princip je stejný nezávisle na účelu chatbota, ale je potřeba velkého trénovacího datasetu, který už musí být specifický pro dané použití. Kromě specifického přístupu je možné využít i obecného omezení vstupu uživatele například tím, že chatbot proaktivně vede konverzaci, klade uzavřené otázky nebo odpověď je na výběr z několika možností.

Poslední částí je předání odpovědi uživateli. Tato část je inverzní první části - příjmu vstupu uživatele. Vždy se skládá z textového výstupu nebo hlasové syntézy, u robotů je možné nastavit jeho gestikulaci a celkový dojem promluvy. [4, 5, 6, 7]

¹ Natural Language Toolkit - <https://www.nltk.org/>

² Term Frequency - Inverse Document Frequency - určení relevance slova v textu podle výskytu daného slova v trénovacím datasetu (čím větší výskyt tím menší relevance)

2.3 Využití chatbotů

Chatboty lze rozlišit primárně podle oblasti použití. Dále je možné chatboty dělit podle jejich vlastností, například podle použití ze strany uživatele na jednorázové s nezávislými konverzacemi a dlouhodobé, kde je nutné udržovat širší profil uživatele. Většinu chatbotů lze roztrždit do tří hlavních kategorií: digitální asistent, informační zdroj nebo obecný/sociální chatbot.

Informačním zdrojem jsou takoví chatboti, kteří nabízejí shrnutí již dostupného obsahu (například noviny) typicky pomocí krátkého zadání (např. „Zobraz mi události z mého okolí.“). Interakce s uživatelem jsou na sobě nezávislé, někdy může být využita profilace uživatele pro účely nabízení personalizovaného obsahu.

U digitálních asistentů je typická konverzace složena z několika obrátů. Uživatel přistupuje k chatbotovi se záměrem dokončit nějakou úlohu - například koupě letenky nebo nahlášení vady při řešení reklamace. Konverzace se skládá z několika obrátů, kde druhá a další promluva ze strany chatbota jsou doplňující otázky k úloze uživatele. Příkladem digitálních asistentů jsou dnes běžně používané osobní asistenti (Apple - Siri, Microsoft - Cortana, Google Assistant, Amazon - Alexa, ...), kteří mohou mít jak otevřenou, tak i konkrétní doménu použití. Hojně jsou digitální asistenti využíváni v eshopech a prodeji či poskytování služeb obecně nebo jako první úroveň uživatelské podpory (např. call centra). V menším počtu je možné je potkat při výuce nebo v nemocnicích a hotelech při navigaci a odbavení lidí.

Poslední větší kategorií jsou sociální chatboti. Konverzace není omezená počtem obrátů a konverzovat lze na více než jedno téma. Někteří obecní chatboti disponují vlastní osobností, nebo i dokážou profilovat uživatele a na základě toho se s ním porovnávat a přizpůsobit konverzaci jeho zájmům. Chatboti v této skupině mohou významně využívat prvky neverbální komunikace, buď k doplnění jejich výstupu nebo k pochopení významu odpovědi uživatele. [1, 2, 4, 7, 8]

Pokud jsou sociální chatboti spojeni s robotem, pak konverzace může dosahovat až lidských rozměrů, ale i textoví chatboti mohou například rozpoznávat emoce. Vyspělý sociální chatbot měl být empatický, vnímat emoce uživatele a jejich změnu v čase, zároveň mít vlastní emoce a umět je v čase měnit. Konverzace by měla být přizpůsobena aktuální náladě uživatele, uživatel by měl být motivován k pokračování v konverzaci a měl by cítit, že ho chatbot chápe. Chatbot by sám o sobě měl mít silnou a konzistentní osobnost. Ze začátku by měl disponovat excelentní znalostí pouze některých témat a být schopen se učit z uskutečněných rozhovorů. [3, 7, 9]

Příklady obecných chatbotů:

- ELIZA (1966) - první obecný chatbot, vznikl na základě Turingova testu¹, založen na pravidlech a pattern matchingu, omezený tematický a znalostní rozsah
- PARRY (1972) - založen na ELIZA, přidává osobnost chatbota
- ALICE (1995) - tematicky a znalostně omezen podle dialogů napsaných v AIML², třikrát vyhrál Loebnerovu cenu³
- Jabberwacky (1997) - chatbot s osobností, který učí se z předchozích konverzací, dvakrát vyhrál Loebnerovu cenu, novější verze pod jménem Cleverbot
- Woebot (2017) - chatbot vytvořený za účelem pomoci lidem s jejich mentálním zdravím
- GPT-3 (2020) a LamDA (2021) - chatboti vytvoření pomocí jazykových modelů

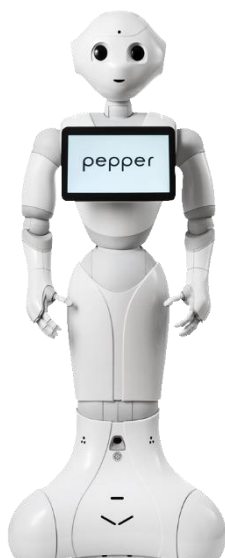
[5, 7, 10]

¹ Turingův test - testující osoba má za úkol rozlišit, zdali odpovědi na její libovolné otázky pochází od člověka nebo od počítačového programu. Pokud to nedokáže, pak program úspěšně splnit test. Více na: https://cs.wikipedia.org/wiki/Turing%C5%AFv_test

² Artificial Intelligence Markup Language - založen na XML, umožňuje vytvářet dialogy pomocí pattern matchingu, kde několik vzorů může být použito pro jednu odpověď

³ Loebnerova cena - soutěž založená na splnění Turingova testu

3 Robot Pepper



Obrázek 1 - robot Pepper
[11]

Pepper je humanoidní robot od společnosti SoftBank. Robot má člověku podobnou anatomii od pasu výš, místo nohou je tříkolová základna ve tvaru trojbokého jehlanu. Hlava robota není nijak pohyblivá, jediný možný pohyb je v krčném kloubu. Detailnější pohledy robota jsou v příloze A. Robot je na baterii s kapacitou 30Ah. Nabíjet lze pouze pokud je robot vypnut, je-li robot zapnut, pak je pouze napájen.

Robot umí mluvit mnoha jazyky, nemá ale plnohodnotné STT¹, jediná nabízená možnost je poslouchat programátorem předem nastavené fráze nebo slova. TTS a STT jsou ve většině jazyků od firmy NUANCE, pro japonštinu je dostupný software AiTalk od AI. Pro tvorbu dialogů je k dispozici speciální jazyk² a speciální příkazy³, které je možné vkládat do textu. Pro výstup zvuku jsou dva reproduktory v uších, pro vstup 4 mikrofony na hlavě (čtvercová konfigurace: pravý/levý-přední/zadní). Mluvení je možné současně doprovázet pohyby těla. Pohyb robota a jeho těla je možné řídit manuálně nebo automaticky, například je možné nechat robot se otáčet za člověkem podle vizuálního nebo zvukového kontaktu.

Robot je osazen dvěma 2D barevnými kamerami na čele a v ústech (typ OV5640, 2592x1944 px) a hloubkovou kamerou v levém oku (ASUS XTION, 1280x1024 px, zaostřovací rozsah 40 cm až 8 m). Robot disponuje mnoha senzory: tříosý gyroskop a akcelerometr, na noze: 3 nárazníková tlačítka, 2 sonary, 6 laserových a 2 infračervené senzory, hmatové senzory na hlavě a rukou. Tyto senzory nabízí mnoho dat pro dobrou orientaci v prostoru. Nárazníková tlačítka a hmatové senzory mohou sloužit i pro dotykovou interakci s uživatelem. Kamery a senzory vzdáleností umožňují například: vyhýbat se objektům při pohybu, sledovat objekty nebo lidi, učení tváří a jejich detekce v obraze nebo detekci speciálních Naomarks⁴.

¹ Speech To Text - software pro transkripci zvuku, opačný směr než TTS - Text To Speech

² <http://doc.aldebaran.com/2-8/naoqi/interaction/dialog/aldialog.html>

³ http://doc.aldebaran.com/2-8/naoqi/interaction/dialog/dialog-syntax_full.html

⁴ <http://doc.aldebaran.com/2-8/naoqi/vision/allandmarkdetection.html>

Počítačová část robota je složena z CPU Intel Atom E3845 (64bitové, čtyři jádra, 1,91 GHz) a 4 GB DDR3 RAM. Robota lze připojit k internetu pomocí ethernetu RJ45 nebo Wi-Fi standardu 802.11 a/b/g/n, konektivitu doplňuje také Bluetooth do verze 4.0. Na hrudi robota se nachází dotykový tablet (IPS, až 5 dotyků, 1280x800 px, vlastní CPU [ARM Cortex-A7, čtyři jádra, 1,3 GHz] a 1 GB RAM). Robot má 3 nezávisle ovladatelné segmenty LED: na očích a ramenou (24bitové RGB spektrum) a na uších (16 úrovní modré). Zapínací tlačítko se nachází na tabletu, za krkem je total-stop tlačítko pro případy nouze.

Robota lze ovládat pomocí aplikace Choregraphe, vlastního kódu a oficiálního SDK¹ nebo pomocí ROS² rozhraní. SDK je dostupné pro CPP, Javu a Python, případně i starší verze pro JavaScript. Dokumentace k SDK a API³ je dostupná pouze jako webová stránka. Všechny hardwarové součásti je možné ovládat (zapisovat či číst data) buď pomocí API nebo přímého nastavení paměti pomocí textového klíče. API je rozděleno na služby, každá služba pak kromě metod nabízí signály a eventy (události). Služby API lze rozdělit na primitivní a komplexní/nadstavbové. Účelem primitivních služeb je nabídnout snazší ovládání hardwaru a poskytnout nezměnná data ze senzorů. Komplexnější služby pak poskytují ucelené nadstavby, například: rozpoznávání tváří, jejich vlastností a možnost pojmenování uživatelů, rozpoznávání nálady, sedících osob, mávání rukou, tvorbu dialogů, automatické základní reakce apod. Některé služby nepřímo ovládají i jiné služby nebo vyžadují změnu jejich nastavení, tyto vedlejší efekty a požadavky jsou zmíněny v popisu služeb nebo metod. [11]

¹ Software Development Kit - sada nástrojů pro vývoj softwaru, běžně obsahuje nějaké API

² Robot Operating System - <https://www.ros.org/>

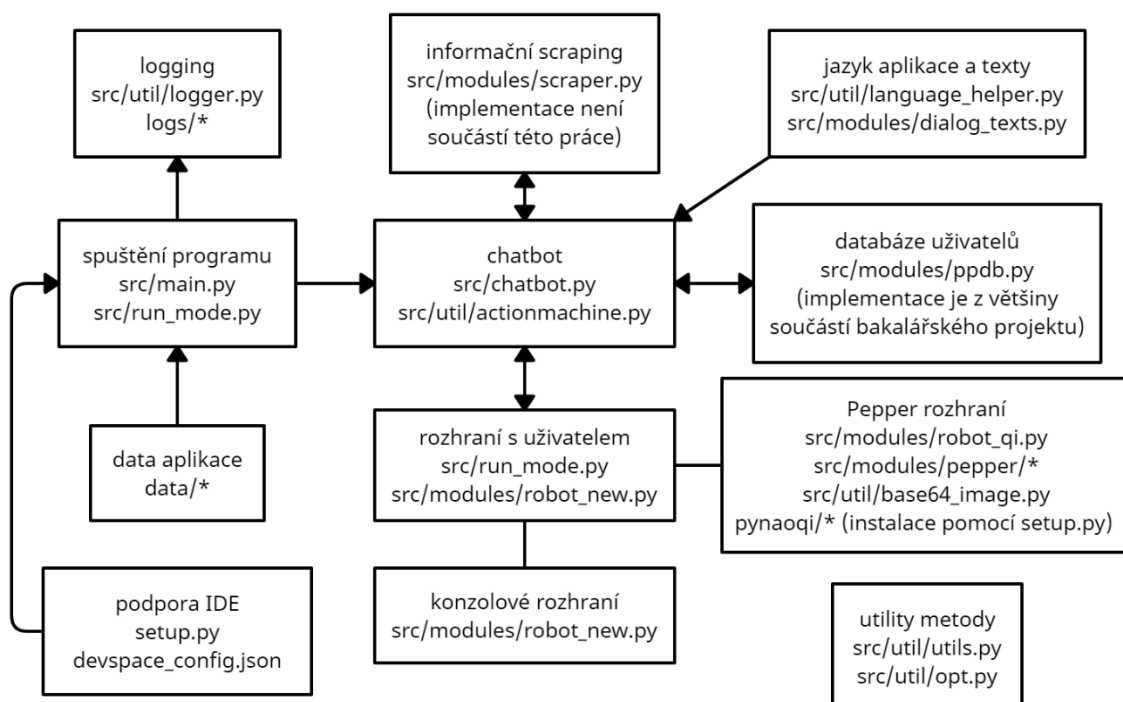
³ Application Programming Interface - rozhraní aplikace, které slouží k jejímu použití pomocí jiného softwaru

4 Implementace chatbota

Aplikace chatbota pro humanoidního robota Pepper je implementována v Pythonu 2.7. Aplikace využívá knihovny popsané pomocí `/requirements.txt`¹ (lze snadno nainstalovat pomocí `pip2 install -r requirements.txt`) a `sqlite` (minimální verze 3.8.11.1) bindings pro Python 2. Z pohledu teoretické části se jedná o sociálního chatbota s `pattern matchingem`, předepsanými texty a rozpoznáváním tváří uživatelů.

Projekt se spouští pomocí souboru `/src/main.py` buď z příkazové řádky nebo pomocí metody `main`. Poté aplikace běží samostatně, dokud není vypnuta, od uživatele se očekává vstup v podobě odpovědi, kdykoli je během konverzace položena otázka. Argumenty jsou popsány v nápovědě (přepínač `-h`) nebo v souboru `/src/main.py`, popis speciálních argumentů, které slouží k jemnějšímu nastavení a podléhají častým změnám, lze získat spuštěním `/dump_kwargs.py`. Až na jeden jsou všechny argumenty volitelné a mají nastavenou výchozí hodnotu. Jediným povinným argumentem je režim (`mode`), kterým se vybírá, v jakém režimu aplikace poběží. Režimy jsou tři:

- `local` - přizpůsobeno pro běh v konzoli, vstup i výstup je textový
- `robot` - běží výhradně na robotovi, tj. musí být spuštěno přímo z robota
- `robot_remote` - aplikace se vzdáleně připojí na robota, je třeba znát jeho IP



Obrázek 2 - popis modulů a souborů, které pod ně spadají. Šipky ukazují tok dat [vlastní]

¹ Všechny cesty k souborům v kapitole 4 jsou vztaženy ke kořenovému adresáři přílohy C

4.1 Moduly

Implementace aplikace je rozdělena do hlavní třídy, modulů a utility tříd. Nejdříve jsou popsány samostatné moduly, v kapitolách 4.1.5 a 4.1.6 je poté popsán chatbot samotný, jako poslední jsou krátce shrnuty utility funkce.

4.1.1 Rozhraní s uživatelem a Pepper API

V kapitole 3, ve které je popsán robot Pepper, je zmíněno Pepper SDK pro Python ve verzích 2 a 3. Pepper SDK pro Python 3.6 je dostupné pouze v jedné vydané verzi¹ (označené jako release), ačkoliv ho někteří programátoři úspěšně používají, mně se nepodařilo zprovoznit některé potřebné metody (mimo jiné například většinu použitých callbacků). SDK pro verzi Python 2 je sice stabilní, ale jedná se pouze o dynamický wrapper CPP metod. Oficiální dokumentace existuje pouze v HTML provedení s CPP deklaracemi metod, neexistuje žádná oficiální podpora pro Python IDE (např. stub soubory vytvořené pomocí Mypy²). Kvůli dvěma předchozím faktům bylo velmi obtížné hledat a často i pochopit metody a správně přeložit jejich signature z CPP do Pythonu. Např. obecný wrapper `ALValue`, který je použit pro většinu složitějších objektů, nemá přímý ekvivalent v Pythonu a je většinou přeložen jako strom z vnořených listů a primitivních typů.

Dalším problémem tohoto SDK jsou eventy a signály, které jsou propagované z CPP prostředí pomocí vláken. Jelikož některé Python interprety využívají GIL pro správu paměti (tzn. že pouze jedno vlákno může v daný okamžik vykonávat Python kód), musí vlákna eventů čekat na přepnutí z hlavního vlákna. Vlákna eventů pak zbytečně zabírají hardwarové prostředky. V případě jiných Python interpretů je naopak potřeba dbát na správnou synchronizaci. Na straně aplikace je tento problém řešen synchronizačními zámky a frontou, která slouží k předávání dat z vláken eventů do hlavního vlákna aplikace.

Třídy uživatelského rozhraní jsou `/src/modules/robot_new.py` (pro konzolové rozhraní) a `/src/modules/robot_qi.py` (pro Pepper SDK), v adresáři `/src/modules/pepper` jsou třídy Pepper API sloužící k otypování Pepper SDK a třídy pro převod některých tříd wrapperu na jejich Python ekvivalenty.

¹ <https://github.com/aldebaran/libqi-python/releases>

² <https://mypy.readthedocs.io/en/stable/stubs.html>

Rozhraní nabízí následující metody:

- hlasové metody `say` a `listen` - tyto metody slouží k asynchronnímu vstupu nebo výstupu hlasu. Součástí zpracování hlasové vstupu je převod získané hlasové stopy na text, k tomu se využívá knihovna `speech_recognition`¹ a její součást pro Google STT službu. Je možné nastavit dobu ticha od posledního zaznamenaného úseku, po kterou se má čekat, než bude hlasový záznam poslán do STT služby. TTS obsluhuje sám robot.
- metody pro ovládání tabletu robota - nabízí přehrávání videa/zvuku, zobrazení libovolného HTML souboru (HTML soubor je načten a poslán do robota ze strany aplikace, vyžaduje speciální Choregraphe aplikaci pro příjem poslaného souboru - příloha D) a vrácení tabletu do základního stavu
- metody pro sledování tváře člověka - hlavní metodou je připojení callbacku z rozpoznávání tváří, který je předává informace o detekovaných osobách před robotem. Dále jsou k dispozici metody na učení tváří a jejich sledování - robot otáčí svoji hlavu za nejbližší osobou či zvukem.

Speciálním souborem je `/src/run_mode.py`, který zajišťuje izolaci Pepper SDK, které není potřeba pro konzolové rozhraní, a správné pořadí importů při vytváření uživatelského rozhraní.

4.1.2 Dialogové texty a jazykové třídy

V aplikaci je možné přepnout za běhu jazyk vstupu a výstupu. V `/data/locales.json` jsou definovány jazykové profily², které obsahují parametry pro nastavení jazyku v různých modulech (například TTS pro Pepper).

Části dialogových textů (názory chatbota, obecné odpovědi a otázky pro uživatele) jsou předem sepsané v `/data/<označení jazyka3>/topics.csv`. CSV⁴ soubor je rozdělen do kategorií, témat a sentimentů pomocí prvních 4 sloupců (například kinematografie - film - nejoblíbenější). Kategorie, témata a sentimenty jsou definovány jako anglický klíč stejný napříč jazyky = překlad v daném jazyce. V dalších sloupcích jsou věty, které se použijí při tvorbě konkrétních bloků dialogu. Ve sloupci 8 je hodnota, která určuje názor robota.

¹ <https://pypi.org/project/SpeechRecognition/>

² Data pro profil en-US nejsou v současné verzi aplikace úplná, tj. aplikace funguje pouze v cs-CZ.

³ Není jednoznačně definováno, ale je nutné použít stejné označení, které je použito v jazykovém profilu. V práci byla použita kombinace ISO 639-1 a ISO 3166-1.

⁴ Comma Separated Values - tabulková data oddělená pomocí čárek a řádků

Obecné odpovědi robota, regulární výrazy sloužící k pochopení odpovědí uživatelů jsou uloženy v `/data/<označení jazyka>/dictionary.json` jako dvojice klíč-hodnota, kde hodnota je buď jedna možnost nebo seznam možností.

Obsah `topics.csv` a `dictionary.json` není součástí tvorby této práce.

Třídy v `/src/modules/dialog_texts.py` slouží k načtení CSV souboru s dialogový texty. Jednotlivé kategorie, témata a sentimenty jsou uspořádány do stromové hierarchie. Ze sentimentů lze pomocí enumu `Sentences` získat jednotlivé věty. Metody v `/src/utils/language_helper.py` umožňují načíst jazykové profily, dalším modulům měnit jazyk celé aplikace, překládat klíče z `dictionary.json` (viz výše) nebo registrovat callbacky pro odposlouchávání změny jazyka aplikace.

4.1.3 Scraping informací

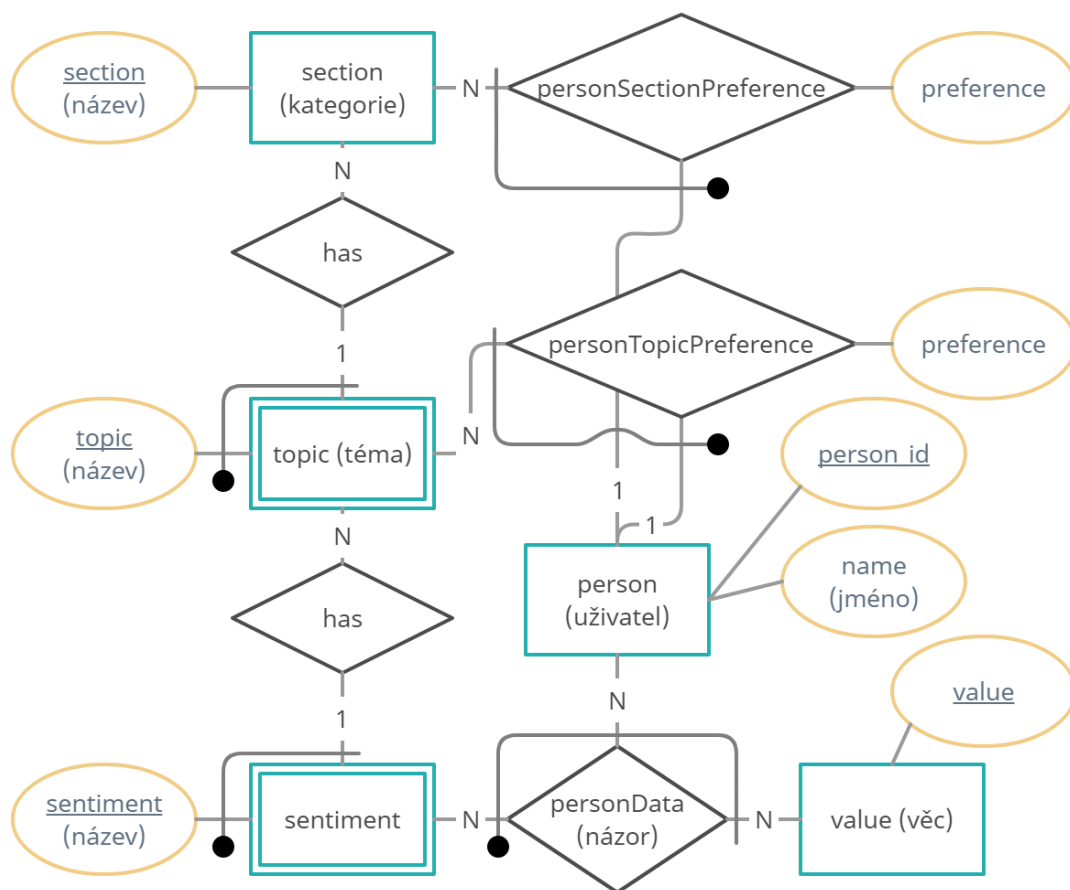
V rámci konverzace o nějakém tématu chatbot tvoří odpověď z informací získaných na internetu (např. Wikipedia nebo ČSFD). Soubor `/src/modules/scrapper.py`¹ obsahuje třídu `Scrapper`, která umožňuje pro různá témata získat informace z internetu. Metodě `web_scrape` se předává celá odpověď uživatele a aktuální téma, metoda pak vrátí instanci `ScrapeResult`, která obsahuje získané informace nebo indikuje, že se scraping skončil neúspěchem.

Obecný scraping začíná dotazem ve vyhledávači, který je sestaven z uživateli odpovědi a téma nebo omezení stránky, na které se má hledat. Tento způsob umožňuje relativně efektivně extrahovat název věci z odpovědi. Poté následuje získání informací o věci, které je buď specializované (např. filmy/knihy - databáze filmů/knih) nebo obecné (pomocí Wikipedie). Většina specializovaných zdrojů je analyzována pomocí parsování HTML.

4.1.4 Databáze uživatelů

Databáze uživatelů slouží k uložení jmen uživatelů, kategorií, témat a sentimentů, jmen věcí (názvy filmů, jídel apod.) a názorů uživatelů na kategorie, témata a věci. Také je v ní uložen názor chatbota.

¹ Implementace `scrapper.py` nebyla součástí této práce.



Obrázek 3 - ER diagram databáze uživatelů [vlastní]

V diagramu databáze je vidět centrální entita **personData**, která spojuje tři hlavní větve: uživatel (entita **person**), kategorie-téma-sentiment (entita **section-topic-sentiment**) a věc (entita **value**). U uživatele je, na rozdíl od ostatních entit, důležitý i atribut **id**, který slouží k identifikaci uživatele napříč aplikací. Entity kategorie a téma mají navíc slabé entity **personSectionPreference** a **personTopicPreference**, které vyjadřují na celočíselné škále -10 až 10 (negativní až pozitivní) uživatelův vztah k dané sekci nebo tématu. V obrázku není uvedena entita **PPDB**, která obsahuje následující atributy: verze schéma databáze, verze sqlite (ve které byla databáze naposledy otevřena), název a čas posledního importu dat.

Kategorie, témata, sentimenty a věci lze hromadně importovat pomocí funkce `import_data`, věci lze také přidat při vkládání názoru uživatele (funkce `put_value_into_person`). Funkce `select_section` a `select_topic` slouží k nastavení kategorie a tématu, tyto hodnoty nejsou nastavitelné pomocí parametrů ostatních funkcí, protože se předpokládá, že bude běžně prováděno více volání, kde se kategorie a téma nemění. Funkce `get_histo_for_person_sentiment` (četnost sentimentů pro zvolenou osobu a téma), `get_histo_for_sentiment` (četnost věcí pro sentiment a téma) a `get_histo_for_value` (četnost sentimentů pro konkrétní věc) vrací třídu `_Histogram`, která nabízí základní statistické funkce (sumu a průměr četností a počet tříd) a řazení podle četnosti jednotlivých tříd histogramu.

Metoda `recommend_person_by_topic` je základem pro budoucí obecné doporučování. Metoda porovnává názory zvoleného uživatele se všemi ostatními uživateli a chatbotem. Porovnávají se názory na věci, ke kterým má zvolený uživatel přiřazen jeden ze sentimentů, jejichž seznam (společně s jejich hodnotami) je argumentem metody. Výsledkem je seznam uživatelů seřazených podle míry podobnosti se zvoleným uživatelem. Míra podobnosti dvou uživatelů je suma invertovaných rozdílů hodnotí sentimentů. Rovnice pro výpočet míry podobnosti uživatele pu a u ($mp_{pu,u}$):

$$mp_{pu,u} = \sum_v^{V_{pu,u}} rhs - |hs_{u,v} - hs_{pu,v}|$$

- $V_{pu,u}$ - průnik množin věcí, ke kterým mají uživatelé pu a u přiřazený jeden ze sentimentů, které byly předány jako argument metody
- rhs - maximální rozsah hodnotí sentimentů
- $hs_{u,v}$ a $hs_{pu,v}$ - hodnota sentimentu, který uživatel u nebo pu přiřadil věci v

4.1.5 Chatbot.py

Soubor `/src/chatbot.py` je hlavní soubor aplikace, který spojuje všechny moduly a řídí veškerý běh konverzace. Třída `Chatbot` je navržena jako stavový automat, kde každá funkce je nezávislá na druhé a funkce se volají nepřímou pomocí zásobníku akcí. Tento přístup byl zvolen kvůli problémům popsaným v 4.1.1. Při přechodu mezi dvěma akcemi jsou volány metody, které zajišťují správný běh robota Pepper.

Aplikace po startu čeká na nalezení a rozpoznání tváře nějaké osoby. Pokud tvář nezná, pak přejde do části pro založení nového profilu osoby. Při zakládání nového profilu probíhá učení tváře robotem, poté se robot ptá uživatele na jméno. Když pozná tvář známého uživatele, pozdraví ho a aktualizuje rozpoznávání jeho tváře (dle dokumentace¹ rozpoznávání probíhá na základě až 10 předchozích obrázků).

Kdykoliv je nějaká osoba rozpoznána, aplikace přechází do dialogové části. Nejprve se náhodně volí sekce, při volbě je uživatel tázán, zdali se chce o zvolené sekci bavit nebo ne. Tento dotaz je položen s určitou pravděpodobností v závislosti na preferenci dané sekce, čím je absolutní hodnota preference vyšší, tím je vyšší šance, že dotaz bude přeskočen. Pokud uživatel odpoví neutrálně nebo kladně či je preference dostatečně vysoká, přechází se k náhodné volbě tématu. K další volbě sekce dojde tehdy, když proběhne konverzace pro náhodně zvolený počet témat.

Po volbě téma se náhodně volí sentiment, vybírá se z množiny sentimentů, které byli nejméně konverzovány s aktuálním uživatelem. V aktuální verzi aplikace je možné diskutovat o každém sentimentu pouze jednou. Jakmile je nějaký sentiment vybrán, přejde se ke konstrukci dialogu.

4.1.6 Konstrukce dialogu

Dialog pro konkrétní sentiment je konstruován z na sobě nezávislých bloků tak, aby dialog dával smysl, ale zároveň byl rozmanitý a neunavoval uživatele. Dialog je vždy veden tak, že robot řekne nějakou informaci a/nebo položí uzavřenou otázku, na kterou by uživatel měl krátce odpovědět (typicky výčet možností - např. ano/ne, nebo název nějaké věci), jakékoliv otázky nebo souvětí jako odpověď mohou být špatně interpretovány.

V aktuální verzi aplikace jsou implementovány pouze 4 dialogové bloky (název bloku odpovídá názvu metody v `chatbot.py`):

- `say_info_robot` - v tomto bloku robot seznamuje uživatele se svým názorem, který se skládá z jeho věci (názoru na sentiment) a v 50 % případů i ze zdůvodnění jeho názoru. Pokud někdy (v rámci jednoho sentimentu) před tímto blokem byl blok `ask_value` a zároveň odpověď uživatele je stejná jako názor robota, pak robot uživateli oznámí, že jsou stejného názoru.

¹ http://doc.aldebaran.com/2-8/naoqi/peopleperception/alfacedetection-api.html#ALFaceDetectionProxy::reLearnFace_ssCR

- `ask_value` - je klíčovým blokem pro získávání názoru uživatele. Robot se ptá uživatele (např. „Jaký je tvůj nejoblíbenější film?“) a uživatel odpoví (např. „Nevím.“ nebo „Můj nejoblíbenější film je XY.“), poté následuje problematické zjištění, kde a jaká věc se v odpovědi nachází. V současné verzi je pro všechna témata použito scrapování (viz kap. 4.1.3), pro případné porovnání je odpověď zbavena běžných nepísmenných znaků a přebytečných mezer.
- `say_info` - tento blok sdělí uživateli výsledek scrapování. Pokud scrapování skončilo neúspěšně, pak robot řekne pouze obecnou větu (např. „Ten film se mi taky (ne)líbil.“). V budoucnu je plánováno rozšíření, kde, pokud je tento blok zařazen před dotazem, je nejdříve vybrána náhodná věc a poté je uživateli sdělena ve formě „Věděl si že, [téma] [věc] ...“.
- `handle_same_opinion` - blok je typicky umístěn po bloku `ask_value`. Jeho prací je informovat uživatele tehdy, když uživatelův názor je shodný s názorem chatbota. Pokud k tomu dojde vícekrát během jedné konverzace, pak chatbot uživateli sdělí, že je rád, že má podobné názory jako on.

Bloky, které jsou plánovány, ale nebyly ještě implementovány:

- `say_info_other_ppl` - má podobný význam jako `say_info_robot`, ale zdrojem není názor robota, nýbrž uživatelů v databázi chatbota nebo, pokud je pro aktuální téma dostupný, názor lidí na internetu.
- `ask_like_or_sentiment` - robot se uživatele zeptá, zdali se mu líbí nějaká věc. Očekávané odpovědi jsou „ano“, „ne“, vyjádření sentimentem nebo že o dané věci neví/nezná ji. Zdrojem by měly být věci obecně známé, aby šance, že uživatel vybranou věc nezná, byla minimální.
- `ask_playing_game` - na robotovi jsou dostupné aplikace/hry přizpůsobené nějaké tématice (např. uhadni písničku nebo slepé mapy). Robot by se zeptal uživatele, jestli chce nějakou aplikaci spustit, pouze když je preference téma vysoká a zároveň by počet vyplněných sentimentů byl nad určitou úroveň.
- `ask_like_x_or_y` - uživateli je položen dotaz, jestli má radši X nebo Y. Uživatelova odpověď by měla být X, Y nebo že nezná ani jednu z nabízených možností. Typově se tento blok podobá `ask_like_or_sentiment`.

4.1.7 Utility soubory

Projekt využívá různé utility soubory ze složky `/src/util`, které jsou shrnuty v následujícím seznamu:

- `actionmachine.py` - slouží jako vývojový předstupeň stavového automatu pro `chatbot.py`. Místo stavů se systému předávají lambda funkce nebo pointery na metody. Zároveň obsluhuje „pauzové akce“, které jsou volány mezi každým stavem. Zároveň umožňuje logovat volání stavů nebo okamžité ukončení aplikace (nezávisle na hloubce call stacku).
- `base64_image.py` - načítá obrázky z `/data/html/img` a převádí do base64 kódování pro použití na tabletu robota.
- `language_helper.py` - popsán v kapitole 4.1.2
- `logger.py` - slouží k logování chodu a knihoven aplikace. Problémem logování knihoven je to, že knihovny běžně používají standardní výstupy operačního systému (`sys.stdout` a `sys.stderr`). Oba systémové výstupy jsou přemostěny do Python logovacího frameworku a ten je nastaven tak, aby logoval do souboru i do konzole. Přemostění vyžaduje kopírování bufferu ve vedlejších vláknech, které sleduje systémové výstupy.
- `opt.py` - implementace `Optional` třídy z Javy, `opt_apply` metody pro řetězení metod nad jednou instancí, jež mohou vrátit `None`
- `utils.py` - sada metod pro clampování (ohrazení v intervalu) číselných hodnot, parsování IPv4 a portu, konvertování vnořených struktur (listů, map a stromů) ze `str` do `unicode` třídy a obráceně

5 Testování aplikace

Testování se zúčastnili tři lidé ve věkové skupině 20-25 let. Testující vedli dvě různě dlouhé konverzace, druhá konverzace využívala data získaná z první konverzace. Všechny konverzace byly vedeny pomocí konzolového rozhraní. Záznamy konverzací společně s poznámkami k nim a soupis všech poznámek jsou obsahem přílohy B.

Před konverzací bylo účastníkům sděleno, aby odpovídali, pokud možno, formálně, spisovně a pouze jednou větou, aby nekladli otázky a konverzaci vedli co možná nejpřirozeněji. Po absolvování konverzací testující doplnili známky (jako ve škole) na otázky do následující tabulky a měli možnost se obecně vyjádřit k průběhu konverzací.

Číslo účastníka	1	2	3
Pohlaví	Ž	M	M
Suma časové délky konverzací (v minutách)	40	35	40
Poznámka k účastníkovi	bez předchozí znalosti chatbotů	-	-
Jak příjemná byla konverzace s chatbotem?	4	2	2
Jak důvěryhodně na vás chatbot působí?	5	3	3
Jak přínosný pro Vás byl rozhovor s chatbotem?	2	3	1
Jak zábavný pro Vás byl rozhovor s chatbotem?	2	1	3
Jak moc je podobná konverzace s chatbotem rozhovoru s člověkem?	4	3	3
Jak celkově byste zhodnotil schopnosti chatbota?	3	2	2
Kvalita komunikace člověk	2/3	1-	1
Kvalita komunikace chatbot	3/4	2/3	2

Tabulka 2 - hodnocení chatbota účastníky testování

Z hodnocení vyplývá, že chatbot nepůsobí dostatečně přirozeně a důvěryhodně, naopak relativně dobré známky dostal za přínosnost a zábavnost. U zábavnosti bylo hodnocení pravděpodobně zatíženo tím, že některé chyby chatbota působily komicky. Z kvality komunikace lze usoudit, že kvalita chatbota je přímo závislá na kvalitě vstupu uživatele.

Ve slovním hodnocení se objevily následující body:

- přehlcení uživatele informacemi, pokud dlouho nedošlo ke změně téma
- nemožnost opravy vstupu uživatele

- scraping informací u témat jídlo, státy a města je nezáživný
- chybí detekce humorných odpovědí a zpětná vazba na ně
- konverzace je dlouhodobě monotónní
- názor chatbota „dušené jídlo“ na „nejnudnější kuchyni“ není ideální
- konverzace o objektivních věcech (např. „nejdelší film“) by měly být vedeny jiným způsobem, hlavně by měl být uživatel opraven při špatné odpovědi
- kontextově zcestné informace (ekvivalent ke kontextově špatnému scrapingu níže)
- schopnost chatbota poradit si s překlepy ve vstupu uživatele
- velké číselné údaje (např. počet obyvatel státu) by bylo lepší hrubě zaokrouhlit, než je vypisovat/vyslovovat na jednotky přesně

Testování na robotovi probíhalo pouze během vývoje rozhraní s robotem. U STT a TTS se závažné nedostatky neobjevily, pouze u TTS jsou některá slova špatně vyslovována (např. „krk“ nebo cizojazyčné výrazy). Tento problém lze napravit přidáním speciálních značek do textu, který má robot vyslovit. Rozpoznávání tváří při druhé až čtvrté/páté konverzaci může trvat déle, než je uživatelsky přívětivé, důvodem je ještě neúplný profil tváře uživatele. Výjimečně se stává, že robot označí jiného uživatele, než který je právě před ním.

5.1 Rozbor poznámek ke konverzacím

V souboru seznam_vsech_poznamek.txt v příloze B jsou poznámky ke všem 6 konverzacím. Poznámky jsou rozděleny do 4 částí:

- validní odpovědi uživatele jsou takové odpovědi, kterými je možné odpovědět na otázku chatbota tak, že je zachován aktuální kontext konverzace a odpověď přirozeně navazuje na otázku. Chybou chatbota pak je, když tyto odpovědi špatně interpretuje.

Konkrétní příklady z konverzací:

- „nevím“ nebo „počkej, přemýšlím“ v delších a složitějších variantách, než je možné obsáhnout pattern matchingem nebo regulární výrazem
- obecná odpověď, která popisuje širokou skupinu věcí, např. „nějaký dokument“
- konkrétní odpověď, která ale není jedinečnou věcí, např. „obraz - můj portrét“
- konkrétní odpověď, ale uživatel si nevzpomene na název, tak ji nepřímou popisuje
- různé vtipné odpovědi

Řešením tohoto problému by mohlo být vylepšení scrapingu tak, aby dokázal lépe detekovat výstupy, které jsou výrazně v rozporu se vstupem. Poté by došlo k použití obecné odpovědi, oproti současné verzi aplikace by se ale musel zvýšit jejich počet, aby konverzace nebyla příliš repetitivní.

- chyby uživatelů - do této skupiny patří stavy, kdy uživatel nedostatečně pochopil otázku chatbota, nevnímal konzervaci nebo odpověděl významově špatně. Takové chyby lze obecně těžko detekovat, ale je jim možné snadno předcházet pomocí mechanismu, kdy uživatele může opravit svoji poslední odpověď.
- chyby chatbota - kategorie chyb způsobených chybou v programu nebo slabinou v použitém přístupu k nějakému problému. Nejčastějšími chybami byly kontextově špatné scrapiny (hledání informací mimo aktuální téma konverzace) a slabé nebo vadné regulární výrazy. Mezi další chyby scrapingu patří například vyhledání špatné věci nebo špatná extrakce informace z webové stránky. Patří sem i problém chybějící extrakce věci z odpovědi uživatele, tento problém je v současné verzi aplikace řešen použitím internetového vyhledávače na celou odpověď uživatele. Pokud uživatel zadá dlouhou větu nebo souvětí, pak vyhledávač nemusí najít správnou věc. Poslední významnou skupinou jsou jazykové chyby, např. špatné zájmeno (věc má jiný rod než téma) nebo matoucí text použití při vyzvání uživatele, aby zopakoval svou odpověď.
- subjektivní chyby chatbota jsou názory na to, jak vylepšit již existující dialogové bloky a jejich odpovědi tak, aby konverzace byla dynamičtější a přirozenější. Evidentním problémem je scraping na téma recepty, města a státy, který po několikátém opakování je silně nezáživný a nudný. Obecným problémem jsou vyplňující texty, někdy se opakují nebo je jich příliš, jindy naopak kompletně chybí. Oba problémy se dají snadno opravit a zároveň jejich vyřešení významně zvedne kvalitu konverzace.

Závěr

Cílem práce bylo vytvořit aplikaci chatbota pro humanoidního robota Pepper. Vytvořená modulární aplikace je založena na stavovém automatu pro dynamicky vytvářený dialog, díky modulárním prvkům je solidním základem pro budoucí vývoj. Aplikace je napsána v Pythonu 2 a je možné ji spustit jak na lokálním počítači, tak i na robotovi.

Konverzace je vedena chatbotem, což umožňuje pokládat uzavřené otázky ze strany chatbota tak, aby očekávané odpovědi byly co nejsnazší na analyzování. Dialog je vytvářen z předem napsaných dialogových textů na základě informací uložených v profilu uživatele. Z těchto informací lze vyčíst, jaká jsou oblíbená témata a k nim oblíbené věci uživatele a zároveň na jejich základě nabízet uživateli věci, které by se mu mohly líbit.

Současná verze aplikace byla otestována 3 lidmi. Jejich hodnocení přibližně odpovídá stavu vývoje chatbota. Hlavním nedostatkem je nepřírozenost a nedůvěryhodnost odpovědí chatbota, vyzvednout lze přínosnost informací získaných z internetu. Některé nalezené problémy byly již opraveny, jiné vyžadující komplexnější přepracování určitých přístupů. Problém složitějších obecných odpovědí uživatele je takřka neřešitelný pomocí pattern matchingu a regulárních výrazů, řešením je vyvinout lepší systém pro tvorbu a použití obecných tematických odpovědí.

Směru budoucího vývoje je několik. Prvním limitujícím problémem je Python 2 sám o sobě, který již není podporován a novější knihovny jsou vyvíjeny výhradně pro Python 3, je tedy potřeba vylepšit oficiální verzi Pepper SDK pro Python 3 nebo případně vyvinout nové SDK. Druhým směrem je snížit počet předem vytvořených textů tak, aby jich bylo co nejméně (např. odstraněním duplicit u opakovatelných otázek typu „Jaký je tvůj nejoblíbenější [téma]?“) nebo, pokud to bude možné, vytvořit automatickou tvorbu textů. Třetím, v této práci ne příliš popsáním, cílem je vytvoření systémů doporučení vázaných na konkrétní témata (např. u filmů/seriálů/knih se nabízí uživatelské recenze v různých databázích, které se dají snadno porovnat s databází uživatelů chatbota). Posledním větším problémem současné verze je nahodilost malý počet bloků dialogů. U nahodilosti je řešením použít informované algoritmy - např. volit kategorii nebo téma podle předchozích částí konverzace. Počet konverzačních bloků a jejich možných kombinací se zvýší po dodělání již připravovaných bloků.

Bibliografie

- 1 CHAVES, Ana Paula a Marco Aurelio GEROSA. How Should My Chatbot Interact? A Survey on Social Characteristics in Human–Chatbot Interaction Design. *International Journal of Human–Computer Interaction* [online]. 2021, **37**(8), 729-758 [cit. 2022-11-22]. ISSN 1044-7318. Dostupné z: [doi:10.1080/10447318.2020.1841438](https://doi.org/10.1080/10447318.2020.1841438)
- 2 DUIJST, Daniëlle. *Can we Improve the User Experience of Chatbots with Personalisation?* [online]. University of Amsterdam, 2017 [cit. 2021-11-22]. Dostupné z: [doi:10.13140/RG.2.2.36112.92165](https://doi.org/10.13140/RG.2.2.36112.92165). Magisterská práce.
- 3 FORLIZZI, Jodi, John ZIMMERMAN, Vincent MANCUSO a Sonya KWAK. How interface agents affect interaction between humans and computers. In: *Proceedings of the 2007 conference on Designing pleasurable products and interfaces* [online]. New York, USA: ACM Press, 2007, s. 209-221 [cit. 2021-11-22]. ISBN 9781595939425. Dostupné z: [doi:10.1145/1314161.1314180](https://doi.org/10.1145/1314161.1314180)
- 4 ABDUL-KADER, Sameera a Dr. John WOODS. Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications* [online]. 2015, **6**(7), 72-80 [cit. 2021-11-22]. ISSN 21565570. Dostupné z: [doi:10.14569/IJACSA.2015.060712](https://doi.org/10.14569/IJACSA.2015.060712)
- 5 BROWN, Tom B. a Benjamin MANN, Nick RYDER, et al. Language models are few-shot learners [online]. 2020 [cit. 2021-11-22]. arXiv preprint arXiv:2005.14165. Dostupné z: <https://arxiv.org/abs/2005.14165>
- 6 PICHL, Jan a Petr MAREK, Jakub KONRÁD, Martin MATULÍK, Jan ŠEDIVÝ. Alquist 2.0: Alexa Prize Socialbot Based on Sub-Dialogue Models [online]. 2020 [cit. 2021-11-22]. arXiv preprint arXiv:2011.03259. Dostupné z: <https://arxiv.org/abs/2011.03259>
- 7 SHUM, Heung-yeung, Xiao-dong HE a Di LI. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering* [online]. 2018, **19**(1), 10-26 [cit. 2021-11-22]. ISSN 2095-9184. Dostupné z: [doi:10.1631/FITEE.1700826](https://doi.org/10.1631/FITEE.1700826)

- 8 FØLSTAD, Asbjørn, Marita SKJUVE a Petter Bae BRANDTZAEG. Different Chatbots for Different Purposes: Towards a Typology of Chatbots to Understand Interaction Design. BODRUNOVA, Svetlana S., et al. Internet Science [online]. Cham: Springer International Publishing, 2019, s. 145-156 [cit. 2021-11-22]. Lecture Notes in Computer Science, 11551. ISBN 978-3-030-17704-1. Dostupné z: [doi:10.1007/978-3-030-17705-8_13](https://doi.org/10.1007/978-3-030-17705-8_13)
- 9 SILVERVARG, Annika a Arne JÖNSSON. Iterative development and evaluation of a social conversational agent. In: *6th International Joint Conference on Natural Language Processing (IJCNLP 2013)* [online]. Nagoya, Japan, 2013, s. 1223-1229 [cit. 2021-11-22]. Dostupné z: <https://www.diva-portal.org/smash/get/diva2:686914/FULLTEXT01.pdf>
- 10 DILMEGANI, Cem. Top 30 Chatbots & Reasons for Why They Are The Best. *AIMultiple* [online]. 2021 [cit. 2021-12-28]. Dostupné z: <https://research.aimultiple.com/top-chatbot-success/>
- 11 Pepper - Documentation [online]. SoftBank [cit. 2021-12-20]. Dostupné z: http://doc.aldebaran.com/2-8/home_pepper.html

Seznam příloh

- A Pepper pohledy (str. 36) - obrázky ukazující vzhled robota a umístění kloubů
- B Záznamy konverzací z testování (zip) - konverzace tří účastníků testování
- C Zdrojový kód aplikace (zip) - bez souborů, jejichž obsah nebyl součástí tvorby této práce, aplikace tedy není ve funkčním stavu
- D Kód aplikace Choreographe pro přenos HTML na tablet robota (zip)

[A] Pepper pohledy - zdroj: 11

