



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**KONVERZAČNÍ AGENT PRO VÝUKU ANGLIČTINY**

CHATBOT FOR LEARNING ENGLISH

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MATEJ MELUŠ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Doc. RNDr. PAVEL SMRŽ, Ph.D.**

BRNO 2019

## Zadání bakalářské práce



18871

Student: **Meluš Matej**  
Program: Informační technologie  
Název: **Konverzační agent pro výuku angličtiny**  
**Chatbot for Learning English**  
Kategorie: Web

Zadání:

1. Seznamte se způsoby využití konverzačních agentů ve výuce a existujícími implementacemi agentů s otevřeným kódem.
2. Zpracujte přehled metod a nástrojů pro korekci gramatiky a dalších aspektů používání jazyka a korpusů s vyznačenými chybami.
3. Na základě získaných znalostí navrhnete a realizujete konverzační systém, který vhodnou formou zjistí uživatelské zájmy, bude s ním komunikovat na zvolené téma a současně zlepšovat jeho jazykové dovednosti.
4. Vyhodnoťte přínosy implementovaného řešení při reálném použití skupinou uživatelů.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Funkční prototyp řešení

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Smrž Pavel, doc. RNDr., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 1. listopadu 2018

## Abstrakt

Cielom tejto práce bolo vytvoriť konverzačný systém ktorého účel bude zlepšovanie jazykových schopností používateľa. Systém je zložený zo štyroch hlavných podsystémov kde má každý samostatnú funkciu. Získavanie dát používaných na tréning konverzácií, konverzácia samotná, korekcia gramatického stavu používateľského vstupu a samostatné cvičenia. Vytvorený systém 26 testujúcich používateľov ohodnotilo kladne až neutrálne a 80.8% z nich považovalo systém za použiteľný pri učení angličtiny.

## Abstract

Goal of this thesis was to create conversational system whose purpose will be improving users foreign language level. System is composed of four main subsystems, where each one has standalone functionality. Retrieving data used for training purposes, conversation, grammar correction of users inputs and exercises. Created system was tested by 26 users who scored system from positive to neutral. 80.8% users considered created system to be usable in process of learning English.

## Klíčové slová

konverzačný agent, chatbot, učenie jazyka, strojové učenie, gramatická kontrola, cvičenie angličtiny, Python

## Keywords

conversational agent, chatbot, language learning, machine learning, grammar control, English exercise, Python

## Citácia

MELUŠ, Matej. *Konverzační agent pro výuku angličtiny*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D.

# Konverzační agent pro výuku angličtiny

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Doc. RNDr. Pavla Smrža, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....  
Matej Meluš  
16. mája 2019

## Podakovanie

Rád by som poďakoval svojmu vedúcemu práce pánovi Doc. RNDr. Pavlovi Smržovi Ph.D za odbornú pomoc, rady a trpezlivosť pri riešení práce. Taktiež by som rád poďakoval svojej rodine za morálnu podporu a všetkým používateľom, ktorí sa zúčastnili testovania systému.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Rozbor riešenej problematiky</b>	<b>4</b>
2.1	História komunikačných agentov . . . . .	4
2.2	Súčasný stav komunikačných agentov . . . . .	5
2.3	Komunikační agenti v rámci elektronického vzdelávania . . . . .	6
2.4	Chatbot . . . . .	8
<b>3</b>	<b>Návrh</b>	<b>10</b>
3.1	Návrh implementácie . . . . .	10
3.1.1	Záujmy používateľa . . . . .	11
3.1.2	Dáta . . . . .	11
3.1.3	Konverzácia . . . . .	13
3.1.4	Gramatická kontrola . . . . .	14
3.1.5	Používateľské cvičenie . . . . .	15
3.1.6	Používateľské rozhranie . . . . .	15
<b>4</b>	<b>Implementácia</b>	<b>16</b>
4.1	Získanie počiatočných dát . . . . .	16
4.2	Konverzácia . . . . .	18
4.2.1	Výber odpovede . . . . .	18
4.2.2	Trénovanie chatbota počas konverzácie . . . . .	21
4.3	Gramatická kontrola a vyhodnotenie . . . . .	21
4.3.1	Gramatická kontrola . . . . .	21
4.3.2	Zobrazenie chýb používateľovi . . . . .	23
4.4	Cvičenie . . . . .	24
<b>5</b>	<b>Testovanie</b>	<b>27</b>
5.1	Návrh prototypu na testovanie . . . . .	27
5.2	Testovací protokol . . . . .	29
5.3	Spätná väzba . . . . .	30
5.4	Výsledky testovania . . . . .	31
5.5	Aplikácia získaných poznatkov z testovania . . . . .	32
<b>6</b>	<b>Záver</b>	<b>34</b>
	<b>Literatúra</b>	<b>35</b>
<b>A</b>	<b>Plagát</b>	<b>38</b>

<b>B</b>	<b>Obsah priloženého CD</b>	<b>39</b>
<b>C</b>	<b>Inštalácia a spustenie</b>	<b>40</b>

# Kapitola 1

## Úvod

Pri učení sa nového jazyka je praktické používanie jazyka jednou z najviac odporúčaných a najefektívnejších metód. Pri komunikácii je človek nútený vytvoriť zmysluplnú vetu či súvetie podľa potreby bez toho, aby mal prepísanú šablónu ako napríklad pri cvičeniach v školských učebniciach. Reálnu komunikáciu je teda možné brať ako cvičenie zamerané na všetko, čo človek z daného jazyka ovláda. Málokto má ale k dispozícii niekoho, s kým môže aktívne komunikovať v cudzom jazyku kedykoľvek a kolkokolvek. v tej situácii nahrádza druhého komunikanta konverzačný agent.

Pod pojmom konverzačný agent je možné predstaviť si program, ktorý podľa určitých pravidiel reaguje používateľa . Podnet je napríklad veta, ktorú používateľ povie či napíše a reakcia od agenta je iná veta. Takáto akcia a reakcia tvorí konverzáciu, pri ktorej sa agent snaží svojimi odpoveďami a správaním napodobniť reálne správanie človeka počas konverzácie.

Prví konverzační agenti sa spoliehali na veľké množstvo jednoduchých pravidiel s ktorým mohli pracovať. Čím väčší bol rozsah pravidiel s ktorým sa dalo pracovať, tým presnejšie bolo vyjadrovanie agentov. Vývoj konverzačných agentov spolu so strojovým učením pomohol v spracovaní prirodzeného jazyka. Súčasné technológie sa snažia docieľiť, že agent bude odpovedať na základe významu viet, nie podľa toho či danú vetu pozná alebo nie.

Vzdelávania je pre tému konverzačných agentov sľubná téma. Konverzačný agent ktorého cieľom je vzdelávanie používateľa je využiteľný napríklad v školách, kde by žiaci mohli trénovať cudzí jazyk v rámci konverzácie alebo vo firmách, kde by bol agent v pozícii trénera, ktorého cieľom je zaučiť nového zamestnanca na určité typy konverzácií s ktorými sa môže počas práce stretnúť.

V tejto práci sa venujem vytvoreniu konverzačného agenta, ktorého účelom je cvičenie a zlepšovanie používateľovej angličtiny. Agent by sa mal v rámci konverzácie sústrediť na témy, ktoré používateľ uviedol ako svoje záujmy. Súčasťou tejto práce je aj zber dát na základe preferencií používateľa, ktoré sú využité na tréningovanie agenta. Agent by mal byť schopný upozorniť na gramatické chyby ktoré používateľ počas konverzácie spravil a navrhnúť korekciu. Súčasťou systému by mali byť k dispozícii cvičenia na zdokonalovanie používateľovej angličtiny.

V druhej kapitole popisujem súčasný stav komunikačných agentov a ich využitie. V tretej kapitole popisujem princípy fungovania konverzačných agentov spolu s návrhom implementácie pre tvoreného konverzačného agenta. v štvrtej kapitole sa venujem implementácií jednotlivých častí konverzačného agenta a piata kapitola je venovaná testovaniu na používateľoch a vyhodnotení.

## Kapitola 2

# Rozbor riešenej problematiky

### 2.1 História komunikačných agentov

Pod pojmom konverzačný agent je možné predstaviť si program, ktorý reaguje na používateľa, napr. v rámci konverzácie, a reakciami sa snaží priblížiť čo najbližšie reakciám človeka.

V roku 1950 Alan Turing vyvinul test, ktorý hodnotil schopnosť strojov napodobniť správanie človeka tak, aby správanie stroja bola nerozoznatelné od správania človeka. Tento test bol počiatkom vývoja konverzačných agentov. Prví agenti však nemali väčší zámer ako výmena viet ktorá simulovala bežnú konverzáciu. Ak dokázali simulovať konverzáciu dostatočne dobre, prešli Turingovým testom.

Na Turingovom Teste bola v roku 1991 usporiadaná prvá súťaž s názvom *Loebner Prize* (Loebnerova Cena) v umelej inteligencii. Hugh Loebner, tvorca tejto ceny, sľúbil \$100,000 tomu konverzačnému agentovi, ktorého odpovede nebudú rozoznatelné od odpovedí poskytnutých človekom [5]. Skorý Turingov Test a Loebnerova cena obdržali kritiku ktorá vyplývala z faktu, že konverzační agenti fungujúce na podobnom princípe ako ELIZA dokážu v krátkom časovom rozsahu pôsobiť dostatočne „ľudsky“ aby prešli Turingovým Testom aj s úplnou absenciou porozumenia konverzácií [17].

### ELIZA

Prvý verejne známy konverzačný agent, ktorý mal mať aj iné využitie ako prázdna konverzácia, bol ELIZA. Agentu vytvoril v roku 1966 Joseph Weizenbaum na MIT (*Massachusetts Institute of Technology*).

ELIZA pracovala len v textovej úrovni. Vstup aj výstup bol v textovej forme, v ktorej hľadala určité vzory. Tieto vzory boli ručne napísané takže "znalosti" programu boli obmedzené. z toho vyplýva, že ELIZA nerozumela významu správam, ktoré prijímala alebo odosiela [21].

Cieľ použitia tohoto agenta mal byť v uľahčení práce psychológom či psychiatrom v tom, že na základe určitých vzorov by dokázala roztriediť pacientov podľa príznakov, ktoré vykazujú (napr. úzkosť, depresia, samovražedné sklony, atď.)



## PARRY

Konverzačný agent, ktorého vytvoril v skorých 70. rokoch na Stanforde Kenneth Mark Colby, bol prvý agent ktorý prešiel Turingovým testom.

PARRY mal simulovať pacienta s paranojou. Spochybňovanie úmyslov doktorov, zlá interpretácia motívov, domýšľanie si apod.

Fungoval na podobnom princípe ako ELIZA s tým rozdielom, že simuloval aj emócie (napr. hnev) [15].

## ALICE

Richard S. Wallace, autor jazyka AIML (*Artificial Intelligence Mark-up Language*), v roku 1995 vytvoril komunikačného agenta s názvom ALICE (*Artificial Linguistic Internet Computer Entity*), ktorý poskytoval ľuďom možnosť vytvoriť si vlastného komunikačného agenta.

ALICE nepracuje na princípe spracovania prirodzeného jazyka ale spolieha sa na veľkú zásobu pravidiel a vzorov podľa ktorých dokáže zmapovať vstup na určitý výstup. Základný systém agenta obsahuje približne 50 tisíc pravidiel.

Tieto pravidlá či vzory sú ukladané v AIML zložkách. AIML je odvodený od XML (*Extensible Mark-up Language*) [2].

## 2.2 Súčasný stav komunikačných agentov

Komunikační agenti sú často spájaní s pojmom *Artificial intelligence* (Umelá inteligencia, ďalej AI). Ako je v [12, 105–106] popísané, pre AI existuje aj termín *Artificial General Intelligence* (AGI), ktorý predchádza všeobecnej predstave o tom, čo AI naozaj je. Pre AGI je charakteristické riešenie problémov, učenie a vysporiadanie sa s nedostatkom informácií. To popisuje komunikačných agentov bližšie.

### Využitie komunikačných agentov

Uplatnenie komunikačných agentov je rôzne. Medzi najznámejšie je možné zaradiť odpovedanie na sociálnych sieťach pre rôzne firmy či stránky, spracovanie a vykonanie hlasových príkazov, elektronických sprievodcov či vzdelávanie.

Je mnoho komunikačných agentov a každý je definovaný tým, na čo je určený. No predsa niektoré mená vyčnievajú viac, ako iné.

### Siri

Siri je hlasový osobný asistent, ktorý bol prvý krát predstavený v roku 2011 firmou Apple. Za predchodcu Siri môže byť považované PDA (*Personal Digital Assistant*), ktoré bolo používané najmä v 90. rokoch 20. storočia.

Jedným z dôvodov, prečo bola Siri veľkým krokom v pred, je spracovanie prirodzeného jazyka. Siri dokázala otázku položenú v prirodzenom jazyku zanalyzovať, vykonať to, čo od nej používateľ žiadal a primerane odpovedať v prirodzenom jazyku. Týmto výrazne predčila konkurenciu v danom čase [9].

## Alexa

Firma Amazon vyvinula hlasového asistenta *Alexa*, ktorým používateľ ovláda *Amazon Echo* (inteligentný reproduktor). Skrz Alexu ako komunikačného agenta je možné zadávanie príkazov Echu a ten následne príkazy splní. Medzi dané príkazy patrí napr. púšťanie hudby, ovládanie určitých technológií v rámci domácnosti, hľadanie na internete a iné [7].

Druhá generácia hlasových asistentov sa líši v spôsobe aktivácie. Variácia je aktivácia asistenta vyslovením určitého slova (napr. „Alexa,“).

## Google assistant

V roku 2016 firma Google prišla na trh s vlastným hlasovým asistentom ktorý konkuruje asistentom ako Siri či Alexa. Momentálne je považovaný za jeden z najpokročilejších asistentov [22].

Spolu s Google asistentom google priniesol aj *Google home*. Ide o múdru domácnosť, ktorá je prepojená a ovládateľná skrz hlasového asistenta.

Asistenta je možné aktivovať hlasovo a následne mu zadať úlohy, ktoré má splniť. Týmto agent prispieva napríklad k zjednodušeniu rannej rutiny formou automatického zovretia vody či odostretie a otvorenie okien.

V roku 2018 Google uviedol *Google Duplex*. Ide o nástroj, ktorý dokáže vytvoriť rezerváciu formou telefonátu. Agent situačne vkladá krátke medzery či citoslovčia ako napr. „hmm“ do konverzácie a tým tvorí presvedčivejšie vystupovanie podobajúce sa na človeka [6].

V máji roku 2018 bola predvedená úspešná rezervácia ku kaderníkovi.

## Cortana

Cortana je digitálny asistent vyvinutý firmou Microsoft dostupný na operačnom systéme Windows 10. Funkcia tohoto asistenta je zjednodušenie práce v systéme a prístupu k rôznym informáciám, na internete alebo v rámci počítača.

## 2.3 Komunikační agenti v rámci elektronického vzdelávania

*E-learning*, alebo vzdelávanie s použitím techniky má mnoho definícií. Tieto definície sa môžu odvíjať od použitia technického vybavenia (napr. inteligentné tabule – *smart board*) alebo od spôsobu dodania nových informácií (napr. online kurzy a prednášky).

V rámci vzdelávania už existujú určité systémy, ktoré sa zameriavajú na určité aspekty vzdelávacieho procesu alebo na vzdelávanie samotné.

Ako príklad je možné uviesť učenie sa Anglického jazyka ako cudzí jazyk. Väčšina hlasových asistentov pracuje primárne v angličtine, to znamená, že aby používateľ docielil vykonanie svojho príkazu asistentom, musí príkaz povedať v angličtine. Tento proces pomáha používateľovi k precvičovaniu anglickej výslovnosti a slovnej zásoby. Uvedený príklad je na asistentovi, ktorého primárnym cieľom nieje vzdelávanie. Existujú agenti, ktorí sú určené priamo na to.

Následujúce tri systémy sú postavené na princípe využitia prirodzeného jazyka v dialógu za účelom učenia sa. Agent je v pozícii učiteľa (*tutor*) a vedie žiaka určitým smerom.

## AutoTutor

AutoTutor je jeden z projektov v ktorom je využívané použitie dialógu s konverzačným agentom za cieľom učenia. AutoTutor pôsobí na pozícií učiteľa, ktorý zadáva žiakovi určité úlohy spolu s materiálom potrebným na zodpovedanie úloh. Agent pomáha žiakovi v aktívnom učení [12, 170–171].

## Ms Lindquist

Jeden zo systémov ktoré používajú princíp komunikačného agenta. Tento systém ponúka cvičenie s dozorom, pri ktorom má žiak možnosť učiť sa cvičením namiesto prijatia konečného riešenia. Autori tohoto systému zastávali prístup, kedy žiak čelí celkovo menej problémom no pri probléme sa naučia viac vďaka vedenej konverzácii o danom probléme [12, 170–171].

## Geometry Explanation Tutor

Ďalší systém, agenta za účelom učenia. Pri komunikácii s týmto agentom sú žiaci nútení riešenie na daný geometrický problém vysvetliť vlastnými slovami. Agent na základe znalostí ktoré má zanalyzuje odpoveď a vyhodnotí, či odpoveď žiaka je dostatočná. Štúdie za týmto systémom zistili, že žiaci ktorí sa naučia vysvetliť problém (definíciu) vlastnými slovami majú väčšiu šancu naučiť sa danú problematiku ako tí, ktorí sa učia definície [12, 170–171].

## Basilica

Všetci doteraz spomínaní agenti fungujú na princípe jeden žiak je vedený jedným agentom. Tento prístup je osvedčený, no v súčasnej dobe sú vyvíjané aj systémy, v ktorých je jeden alebo viac študentov učení jedným alebo viacerými agentmi. Tento prístup sa nazýva *collaborative learning*. Na tvorbu týchto systémov je možné použiť architektúru *Basilica* [18].

Konverzační agenti spomínaní v predošlých paragrafoch predstavujú jeden z najpoužívanejších prístupov využitia agentov vo sfére e-learningu. Agent zastáva pozíciu lektora alebo tútora, ktorý určitým spôsobom hodnotí používateľa na základe jeho odpovedí na tému, ktorá je cieľom učenia. Vo sfére vzdelávania existujú aj iné prístupy použitia konverzačných agentov. Niektoré z týchto prístupov sú popísané v nasledujúcich sekciách.

## Freudbot

Zatiaľ čo systém ELIZE bol vytvorený na analyzovanie a pomoc používateľom, cieľom Freudbota je konverzácia. Z mena je zjavné, ktorú osobnosť má agent predstavovať. Výber človeka na ktorého sa agent má ponášať bolo s cieľom vytvoriť konverzáciu pre používateľov zaujímavejšiu. Agent mal v rámci konverzácie pracovať s teóriami a konceptami spájanými alebo pripisovanými Sigmundovi Freudovi. Predpoklad väčšieho záujmu o konverzáciu zo strany používateľov bol potvrdený testovaním, kedy sa používatelia vyjadrili, že získanie informácií týmto spôsobom bolo zaujímavejšie ako čítanie čistého textu [11].

Princíp využitia tohoto konverzačného agenta v rámci e-learningu sa líši od predošlých v tom, že znalosti (v tomto prípade napr. teórie Sigmunda Freuda) sú poskytované ako súčasť konverzácie a nie formou skúšania.

## Využitie konverzačného agenta ako podpora pri hodnotení

V práci [14] autor popisuje využitie konverzačného agenta ako určitú formu sprievodcu pre študentov pri hodnotení ostatných študentov. Cieľom využitia agenta v tomto prípade je dosiahnutie primeranejšieho a presnejšieho hodnotenia. To bolo docielené tým, že agent spracoval predošlé hodnotenia študenta a využil ich ako zdroj tipov a návrhov pre ďalšie hodnotenie. Autor v rámci testovania poukazuje na zlepšenie hodnotení študenta pri využívaní tohoto konverzačného systému.

Väčšina agentov používaných v e-learningu používa angličtinu ako nástroj na dorozumievanie, ale primárny cieľ učenia nie je znalosť angličtiny (alebo iného jazyka). Cieľ týchto agentov je naučiť používateľa určitú tému alebo časť témy. Zameranie agenta tvoreného v rámci tejto práce je však znalosť a zdokonalovanie jazyka samotného. Jeden z nástrojov používaných na zdokonalovanie angličtiny je konverzácia. Mnou implementovaný konverzačný agent využíva podobný faktor ako bol spomínaný pri Freudbotovi s tým rozdielom, že obohatenie konverzácie pre používateľa nie je zaujímavá osobnosť ale témy, ktoré sú zaujímavé pre používateľa.

## Reprezentácia znalostí

Znalosti používateľa sú reprezentované vstupmi, ktoré zadáva v rámci komunikácie s agentom. Čím viac má používateľ znalostí jazyka, tým nižší je výskyt chýb v jeho vstupoch. Znalosti používateľa sú nepriamo prepojené so znalosťami agenta.

Znalosti konverzačného agenta sú primárne reprezentované rozsahom dát, ktoré sú pokryté v rámci databázy. Nepriame prepojenie znalostí je spôsobené pasívnym učením, ktorého sú v rámci konverzácie inteligentní agenti schopní.

## 2.4 Chatbot

*Chatbot* je konverzačný agent ktorý komunikuje s používateľom v prirodzenom jazyku [3]. Chatboty sú väčšinou založené na komunikácií skrz text a jedo z kritérií podľa ktorých sa rozdeľujú je implementácia konverzácie.

### Chatboty založené na súbore pravidiel

Tento prístup bol použitý pri prvom pokuse o vytvorenie chatbota, ktorým bola ELIZA. Princíp tohoto spôsobu implementácie spočíva v jednoduchom párovaní slov. Chatbot obsahuje sadu kľúčových slov, ktoré sú namapované na vopred definované výstupy. Vo vstupe používateľa sú vyhladané všetky kľúčové slová a podľa nájdených slov je vybraná odpoveď, ktorá je spojená s nájdenými slovami. Ak žiadne kľúčové slovo nebolo nájdené, je zvolená prednastavená odpoveď [20].

Chatboty fungujúce na tomto prístupe dokážu vhodne reagovať aj napriek tomu, že konverzácií nerozumejú. Dôkazom je ELIZE, pri ktorej niekoľko ľudí verilo, že komunikujú s ozajstným terapeutom [20]. Tento typ chatbotov je obmedzený rozsahom znalostí, ktoré sú reprezentované súborom kľúčových slov a definovaných odpovedí.

Využitie takéhoto typu chatbotov môže byť vo sférach alebo miestach, kde je predpokladaný výskyt určitých slov. Príkladom môže byť spoločnosť, ktorá má pravidelný kontakt so

zákazníkmi. Využitie takéhoto chatbota by mohlo nahradiť *FAQ – Frequently Asked Questions* (často kladené otázky). Zákazník by komunikoval s chatbotom, od ktorého by získaval potrebné informácie priamo z konverzácie formou otázky a odpovede (napr. „*Do you sell computer parts?*“).

### **Chatboty založené na strojovom učení**

Strojové učenie (*machine learning*) je kategória algoritmov ktoré umožňujú aplikáciám lepšie predvídať výstup zo daného vstupu bez toho, aby boli explicitne naprogramované. Algoritmy zanalyzujú existujúce vstupné a výstupné dáta a na základe získaných informácií dokážu predpokladať budúci výstup [19].

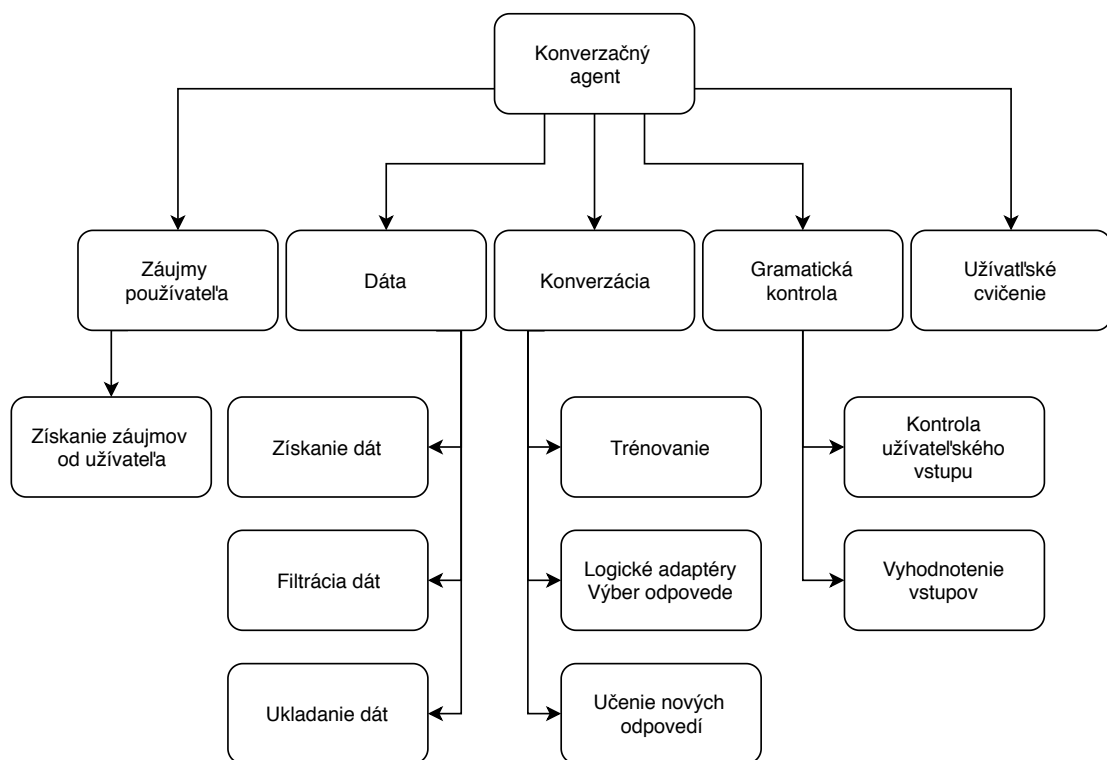
Chatboty založené na strojovom učení sú zložitejšie no v konečnom dôsledku presnejšie z pohľadu výberu odpovede. Znalosti týchto chatbotov pochádza primárne z veľkého objemu tréningových dát alebo z veľkého počtu konverzácií.

# Kapitola 3

## Návrh

### 3.1 Návrh implementácie

V schéme 3.1 je zobrazený návrh chatbota. Chatbot sa skladá z piatich hlavných častí, pričom sú pri každej časti zobrazené aj základné body, na ktoré je potrebné sa v rámci danej časti zamerať.



Obr. 3.1: Návrh riešenia

Jedny z hlavných častí implementovaného komunikačného systému sú postavené na existujúcom systéme s názvom *chatterbot (verzia=0.8.7)*<sup>1</sup>. Tento systém som zvolil kvôli rozhraniu, ktoré poskytuje pre tvorbu chatbota a prácu s databázou, ktorú chatbot používa.

<sup>1</sup><https://chatterbot.readthedocs.io/en/stable/>

### 3.1.1 Zájmy používateľa

Primárnym cieľom chatbota je zlepšovať angličtinu používateľa v rámci konverzácie. Aby bola konverzácia záživnejšia a zaujímavejšia, chatbot by sa mal v rámci danej konverzácie sústrediť na témy, ktoré sú pre používateľa zaujímavé. Získanie záujmov je možné viacerými spôsobmi, pričom každý spôsob má svoje výhody a nevýhody.

Jeden zo spôsobov je poverením získania záujmov chatbota z konverzácie. Jedna z výhod tohoto spôsobu je počínanie sa chatbota na ľudské správanie v rámci dialógu. v tomto prípade by bolo potrebné rozdeliť konverzáciu na dve časti, kedy v prvej časti by bola úloha chatbota získať záujmy a v druhej časti by bola samotná konverzácia na dané témy. Nevýhoda tohoto spôsobu spočíva v časti konverzácie kedy chatbot zisťuje záujmy používateľa. v prípade, že používateľ nespomenie v rámci dialógu témy, ktoré považuje za zaujímavé, táto konverzácia nikdy neskončí a k časti konverzácie sústredenej na dané záujmy vôbec nedôjde. Rovnaká situácia je aj v prípade, kedy chatbot nie je schopný vyčítať záujmy z používateľových vyjadrení.

Z tohoto dôvodu som sa rozhodol pre získanie záujmov formou otázky. Získanie záujmov formou explicitného zadania používateľom môže kaziť dojem aplikácie, pri ktorej sa od používateľa neočakáva nič iné len konverzácia, no predchádza to situácií spomínanej v predchádzajúcom paragrafe.

### 3.1.2 Dáta

Aby chatbot dokázal konverzovať, potrebuj určité dáta. Chatbot je trénovaný na dátach, ktoré reprezentujú konverzáciu dvoch osôb. Dáta ktorými je chatbot trénovaný sú kolekcia dvojíc, kde sa každá dvojica skladá z výpovede a reakcie. Takáto dvojica pre chatbota predstavuje správnu reakciu na daný vstup.

Dáta sú zložené z dvoch hlavných častí. Jedna časť je zložená z ukázkových dát, ktoré predstavujú rozhovor a tieto dáta sú prístupné v *corpusoch* (názov štruktúry v ktorej je uložený text v prirodzenom jazyku) ktoré sú uložené v rámci modulu *chatterbot*. Táto časť dát obsahuje základné konverzácie na všeobecné témy ako napr. počítače, jedlo, literatúra, filmy.

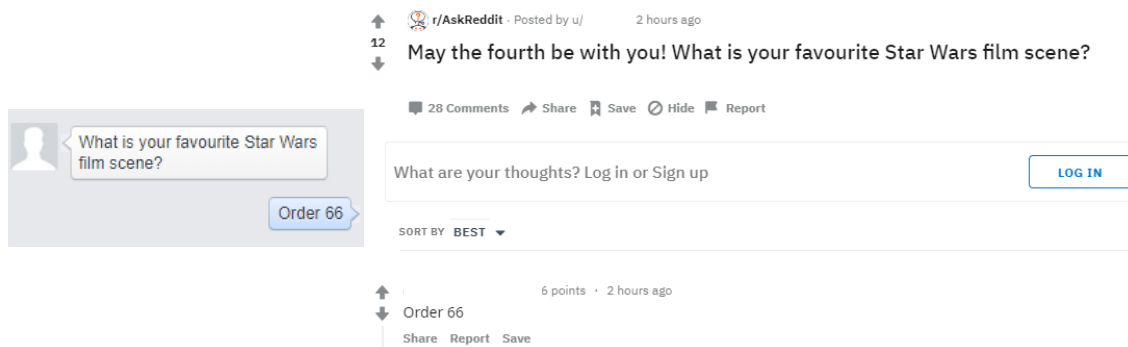
Druhá časť dát predstavuje dáta zamerané na záujmy používateľa. Získanie dát, ktoré najbližšie pripomínajú konverzáciu na konkrétnu tému nie je jednoduché. Predpokladáme, že záujem používateľa môže byť takmer všetko. Najst články zaoberajúce sa danou témou je možné, no väčšina týchto textov je vo forme monológu (napr. blog). Úspešná extrakcia dát ktoré predstavujú konverzáciu z takýchto textov nie je garantované. z tohoto dôvodu som ako zdroj týchto dát zvolil jedno z najpoužívanejších používateľských fór s názvom *Reddit*.

Ako je v článku [16] popísané, Reddit bola 18. najpopulárnejšia stránka celosvetovo v roku 2018. Je to kolekcia mnohých fór ktoré sú nazývané *subreddits*. Každé forum v rámci Redditu sa zameriava na konkrétnu tému, v rámci ktorej môže každý zaregistrovaný používateľ pridávať príspevky, komentovať a hodnotiť príspevky ostatných používateľov. Pre neregistrovaných používateľov je možné len prezeranie príspevkov bez žiadnej interakcie.

Dôvodov, kvôli ktorým som zvolil Reddit ako primárny zdroj dát zameraných na záujmy používateľa, je viac.

Prvým a hlavným dôvodom je podobnosť konverzácií ktorú je možné využiť. v rámci každého subredditu používateľa pridávajú príspevky (*Topic*), na ktoré ostatní užívatelia následne reagujú (*Comment*). Každý komentár (okrem komentárov na už existujúce kome-

táre) predstavuje reakciu na daný príspevok podobne, ako odpoveď chatbota predstavuje reakciu na vstup od používateľa.



Obr. 3.2: Porovnanie konverzácie a príspevku na Reddite

Druhým dôvodom je rozmanitosť fór. Pre väčšinu tém existuje určitý subreddit (napr. *Python*, *Fantasy*, *Cars*, *Programming*, *NBA*) kde sa užívatelia venujú výhradne danej téme. Vďaka veľkému množstvu subredditov je šanca na nájdenie toho, ktorý sa venuje záujmu používateľa relatívne vysoká a tým aj šanca na získanie dát zameraných na danú tému.

Tretím dôvodom je aktivita ľudí. Reddit je jedno z najpoužívanejších fór a na najznámejších subredditoch je denne pridaných niekoľko desiatok až stoviek príspevkov. To je výhodné pri získavaní, kedy je menšia pravdepodobnosť opakovaných dát.

Nevýhoda získavania dát z Redditu spočíva v tom, že všetky príspevky a komentáre sú vytvorené ľuďmi. To znamená, že v nich môže byť použitý určitý nespisovný žargón, vulgarizmy či nesprávna gramatika. Tento problém sa dá čiastočne obmedziť filtrovaním získavaných dát.

Získavanie dát z Redditu som sa rozhodol poňať dvomi spôsobmi. Prvý spôsob je zameranie sa na subreddit, ktorého cieľom je pokladanie otázok a ich zodpovedanie. v rámci tohoto fóra sú otázky filtrované a dáta sú získavané len z tých otázok, ktoré sa určitým spôsobom venujú téme, ktorú používateľ uviedol ako svoj záujem.

Druhý spôsob využíva to, že pri vyhľadávaní v rámci Redditu je odporúčaný aj subreddit, ktorý sa názvom najviac ponáša na záujem používateľa. Tento subreddit je následne využívaný ako zdroj dát.

Dĺžka príspevkov a komentárov sa líši a spadá do intervalu jedného charakteru až po niekoľko paragrafov. Aby sa predišlo situáciám, v ktorých je odpoveď chatbota jednoslovná alebo naopak je tvorená niekoľko súvetiami, odpovede sú rozložené na jednotlivé vety.

Dáta, ktoré sú získavané z používateľského fóra zvyčajne nemajú garantovanú gramatickú korektnosť. v otázkach aj odpovediach sa môžu nachádzať slangové slová, skratky, vulgarizmy či gramatické chyby. Tieto dáta majú predstavovať zmysluplné výroky ktoré budú možnou súčasťou konverzácie. z tohoto dôvodu je vhodná určitá filtrácia, aby boli získané dáta čo najvhodnejšie. Vety, ktoré považujem za vhodné musia spĺňať určité kategórie. Medzi tieto kategórie patrí napr. minimálna dĺžka vety, kapitalizácia začiatku vety, korektné ukončenie vety interpunkčnými znamienkami či iné gramatické pravidlá.

V rámci filtrácie získaných dát dochádza aj k cenzúre vulgarizmov obsiahnutých v dátach.

Pri získavaní dát z Redditu je použitá technika *web scrapingu*. Web scraping je technika ktorou je možné automaticky získať veľké množstvo dát z internetovej stránky [13].



Dáta ktoré sú v rámci chatbota používané sú uložené dvomi spôsobmi. Prvý spôsob používa modul chatterbot na vytvorenie databázy, v ktorej sú uložené dáta ktoré chatbot využíva v rámci konverzácie. Táto databáza je typu *SQLite*. SQLite je knižnica napísaná v jazyku C, ktorá poskytuje databázu ktorá nevyžaduje oddelené spracovanie serverom. v tejto databáze sú uložené všetky výroky a reakcie ktoré sa chatbot naučil spolu s väzbami medzi jednotlivými výroky a odpoveďami a počet výskytov.

Druhý spôsob ukladania je používaný pre všetky ostatné dáta ktoré sú získavané. Dáta získané z Redditu sú uložené v korpusoch, ktoré sú vo formáte *JSON* (JavaScript Object Notation). JSON je vhodný formát dát na prenášanie. Pre ľudí je ľahko čitateľný a jednoducho rozložiteľný a generovateľný pre stroje.

Databáza môže byť uložená lokálne alebo zdieľaná. Výhoda lokálne uloženej databázy je, že v rámci aktívnej konverzácie sa chatbot nenaučí nič, čo by používateľ nenapísal. Naopak nevýhoda lokálnej databázy je pomalá učiacia krivka chatbota. Konverzácia je jeden z hlavných zdrojov dát, na ktorých sa chatbot učí. Ak má k danej lokálnej databáze prístup len jeden používateľ, jediné konverzácie z ktorých sa chatbot učí sú tie, ktorých je používateľ súčasťou.

V prípade globálneho úložiska je databáza aktualizovaná každou konverzáciou v ktorej bola použitá. Nevýhoda online prístupu je riziko ukladania irelevantných dát, ktorých zdrojom môžu byť ľudia, ktorí chcú zámerne znehodnotiť funkčnosť chatbota. Vhodným príkladom, kedy je databáza zdieľaná medzi viacerými konverzáciami, je škola. Ak v rámci vyučovania bude  $n$  študentov konverzovať s chatbotom, databáza bude obohatená o nové dáta z  $n$  rôznych konverzácií.

### 3.1.3 Konverzácia

Jedna z dvoch častí, ktoré sú viditeľné aj pre používateľa. Konverzácia je proces akcie a reakcie, kde akcia je vstup od používateľa a reakcia je vybraná odpoveď chatbota. Konverzácia je jeden z prístupov k učeniu angličtiny. Pri konverzácií je používateľ nútený prepájať jednotlivé pravidlá so slovnou zásobou. Konverzácia je vhodný spôsob na opakovanie, precvičovanie a zlepšovanie angličtiny.

Ako základ, na ktorom je chatbot postavený, som si vybral modul chatterbot. Modul je veľmi dobre zdokumentovaný a poskytuje dostatočné rozhranie na prácu s chatbotom ktorý je implementovaný a s databázou ktorá je používaná. Modul taktiež poskytuje algoritmy na výber odpovede, porovnávanie výrokov, implementáciu vlastných logických adaptérov (skrz logické adaptéry je definovaná logika výberu odpovede) a nástroje, ktoré sú využiteľné v rámci testovania.

Konverzácia zároveň slúži pre chatbota ako zdroj nových dát na tréningovanie. Pri reakcií používateľa na predošlú správu od chatbota, reakcia používateľa je zaznamenaná ako vhodná reakcia na výstup chatbota, ktorý tomu predchádzal. Týmto sa zvyšuje vetná zásoba chatbota a tým presnejší je aj výber odpovedí. v prípade gramaticky nesprávnej reakcie od používateľa musí byť reakcia opravená a až následne posunutá chatbotovi ako materiál na tréningovanie. Týmto je možné predísť tréningovaniu s nesprávnymi dátami.

Tréningovanie chatbota je proces, počas ktorého sa tvoria väzby medzi jednotlivými výroky. Väzby definujú jednotlivé vzťahy medzi výroky ktoré sú na seba naviazané priamo i nepriamo.

Princíp výberu odpovede je výber najvhodnejšej odpovede z dvoch databáz. Jedna databáza je tvorená celkovou znalosťou ktorú chatbot má a druhá databáza je tvorená výroky,

ktoré sú zamerané na záujmy používateľa. Následne je vybraná finálna odpoveď tá, ktorá je z tejto dvojice vhodnejšia. Tým by sa malo docieľiť preferovanie výrokov ktoré sa sústreďujú na témy ktoré používateľ uviedol ako zaujímavé. Výber z celkovej znalosti by mal zabezpečiť, že v prípade keď odpoveď z databázy záujmov nieje dostatočne vhodná, chatbot má k dispozícii záložnú odpoveď.

Výber odpovede skrz jeden logický adaptér je proces, počas ktorého chatbot spracuje vstup od používateľa do požadovaného tvaru a následne hľadá v databáze výrok, ktorý je najviac podobný používateľskému vstupu. Po nájdení daného výroku vyberie z jednu odpoveď zo všetky výrokov ktoré sú definované ako legitímne odpovede pre daný vstup. Jeden logický adaptér vracia jednu odpoveď s jedným parametrom, ktorý sa nazýva *confidence*. Confidence je číslo ktoré reprezentuje presvedčenie chatbota, že daná odpoveď je najvhodnejšia. Tento parameter rozhoduje, ktorý logický adaptér vrátil najvhodnejšiu odpoveď.

Na vypočítanie podobnosti medzi používateľským vstupom a záznamom ktorý ma chatbot uchovaný môžu byť použité porovnávacie algoritmy ktoré poskytuje modul chatterbota. Jeden z algoritmov, ktorý som sa rozhodol použiť je algoritmus na výpočet *Levenshtein distance* (LD, Levenshteinova vzdialenosť). LD je metrika na meranie rozdielu medzi dvoma sekvenciami a vyjadruje minimálny počet úprav jednotlivých charakterov aby sa porovnávané slová zhodovali. Úprava predstavuje pridanie, odstránenie alebo náhradu charakteru [1].

Ďalší algoritmus na porovnávanie dvoch výrokov je pre ktorý som sa rozhodol je *Jaccard similarity index* (JS, Index Jaccardovej podobnosti). Index nadobúda hodnoty 0% – 100% a reprezentuje podobnosť dvoch sád (*set* - nezoradená kolekcia objektov). v prípade chatbota je ako sada braný výrok od používateľa a odpovede chatbota. Výpočet indexu je prienik dvoch porovnávaných sád podelený zjednotením tých istých sád [8].

### 3.1.4 Gramatická kontrola

Kontrolu používateľského vstupu je možné rozdeliť na dve základné časti, *syntax* a *sémantiku*. Sémantika sa zaoberá významom viet, syntax sa zaoberá skladbou viet. z dôvodu náročnosti kontroly sémantiky chatbot kontroluje iba syntax.

Kontrola gramatickej správnosti je druhá časť chatbota, ktorú používateľ vidí. Ako som popisoval v sekcii 3.1.3, konverzácia je vhodný spôsob aktívneho cvičenia angličtiny. Používateľ angličtinu používa, no nemá istotu, či sú jeho gramatika a pravopis správne. Gramatická kontrola vstupov zabezpečí, že používateľ bude vedieť kde sa dopustil chyby. Kontrola chýb je rozdelená na dve časti.

V prvej časti kontroly je na detekciu chýb používaná Python trieda na detekciu a korekciu gramatiky. v druhej časti je vstup používateľa testovaný voči pravopisným chybám ktoré boli zozbierané z reálnych zdrojov (napr. chyby spravené v študentských práca či chyby spravené pri úprave textu na wikipédií).

Pri nájdení chýb je používateľ oboznámený s nálezom a je mu poskytnuté vysvetlenie danej chyby s možnými opravami danej chyby. v prípade nájdenia pravopisnej chyby pri testovaní oproti zozbieraným chybám, je používateľovi zobrazená množina slov, ktoré by mohli byť správnym tvarom zle zapísaného slova. Jedna z nevýhod kontroly pravopisných chýb voči zozbieraným chybám je možnosť podobnosti slov. Pravopisná chyba v slove môže vytvoriť slovo, ktoré je pravopisne správne len s iným významom. Taktiež môže nastať situácia, kedy je v zázname uvedené chybne napísané slovo ktoré je s iným významom pravopisne správne. To môže viesť k nesprávnemu označeniu pravopisnej chyby v používateľskom vstupe. Kvôli týmto možným situáciám je nájdenie pravopisnej chyby podané

používateľovi ako možnosti v ktorých mohol spraviť chybu. Používateľ tak vidí aké chyby boli už zaznamenané pri podobnom slove a potenciálne si môže všimnúť vlastnú chybu.

Gramatická kontrola je jedna z hlavných častí chatbota, čo je dôvodom prečo je vykonávaná stále. Používateľ má ale možnosť zvoliť si čas zobrazovania opráv podľa svojich preferencií.

V prvom móde konverzáciu neprerušuje. Chatbot skontroluje všetky vstupy ktoré používateľ počas konverzácie zadal a po ukončení konverzácie zobrazí správnosť všetky vstupov. Tento mód je nastavený ako predvolený.

V druhom móde je kontrola vykonaná hneď po zadaní používateľského vstupu a následne je výsledok kontroly zobrazený. v tomto móde je používateľ o chybe, ktorú vykonal, informovaný počas konverzácie.

Tento mód je možné nastaviť pri spustení programu.

### 3.1.5 Používateľské cvičenie

Cvičenie pravopisu je ďalšia z použiteľných metód na tréning a zlepšovanie angličtiny. Cvičenie už nie je súčasť konverzácie s chatbotom a je dobrovoľná. Po ukončení konverzácie dostáva používateľ možnosť precvičiť si znalosť pravopisu a slovnej zásoby. Takýto typ cvičenia je možné vytvoriť viacerými spôsobmi.

Jeden zo spôsobov je využitie dát, ktoré obsahujú zozbierané chyby. Výhoda tohoto spôsobu je relatívne veľké množstvo zozbieraných pravopisných chýb, čím by bola docieľaná variabilita chýb na precvičenie. Nevýhoda tohoto prístupu je podobná, ako pri kontrole pravopisu voči týmto zozbieraným dátam. Pravopisné chyby sú zaznamenané bez kontextu. Keďže tieto zozbierané chyby pochádzajú od reálnych ľudí, chyba mohla byť spôsobená vynechaním písmena (*knife* → *nife*) ale taktiež mohla byť spôsobená neznalosťou daného slova (*knife* → *naf*). To znamená, že pri cvičení používateľ nevie, z akého slova tento chybný zápis vznikol a jeho oprava môže vytvoriť gramaticky správne slovo, no ak sa nezhoduje s tým, ktoré je spojené s daným chybným zápisom, oprava je vyhodnotená negatívne.

Ďalší spôsob je založený na podobnom princípe, no chyby sú umelo vytvorené programom. Chyba je vytvorená napríklad vymenením dvoch písmen v slove, vynechanie alebo pridanie písmena. Toto cvičenie nie je náročné a používateľ si pri ňom opakuje pravopis slov.

Tento spôsob je možné využiť aj v cvičení, kedy používateľ opravuje pravopisnú chybu v celej vete. Využitie vety v rámci cvičenia je možné aj na opakovanie a cvičenie syntaxe. Cvičenie využívajúce vety má vyššiu náročnosť, pretože pri pravopisnej chybe používateľ nevie, v ktorom slove je chyba a pri zmiešanej syntaxi musí aplikovať pravidlá, ktoré platia pre tvorbu viet v angličtine.

### 3.1.6 Používateľské rozhranie

Chatbot nemá implementované žiadne grafické rozhranie a používateľ s chatbotom komunikuje skrz príkazovú riadku. Grafické používateľské rozhranie by bolo možné dodatočne implementovať rôznymi spôsobmi podľa toho, ako by bol chatbot prístupný.

Ak by bol chatbot spustený na serveri, ktorý je prístupný online, je možné vytvoriť webové grafické rozhranie a chatbot bude fungovať ako aplikačné rozhranie.

V prípade, že bude chatbot používaný ako samostatná aplikácia bežiacia na používateľskom počítači, je možné vytvoriť grafické rozhranie priamo pre aplikáciu.

## Kapitola 4

# Implementácia

Celý chatbot je implementovaný v jazyku Python, verzii 3.7.0, čím by mala byť zabezpečená nezávislosť od operačného systému. z hľadiska implementácie sa aplikácia chatbota skladá zo štyroch hlavných podsystémov:

- Získanie počiatočných dát použitých na trénovanie a konverzáciu
- Konverzácia a učenie
- Gramatická kontrola a vyhodnotenie používateľových vstupov
- Generovanie cvičení

V nasledujúcich podkapitolách popisujem implementáciu jednotlivých častí chatbota spolu s príkladmi ich použitia. Pri popisovaní použitých modulov a programov sú v rámci názvu uvedené aj verzie s ktorými pracujem.

### 4.1 Získanie počiatočných dát

#### Web-scraping

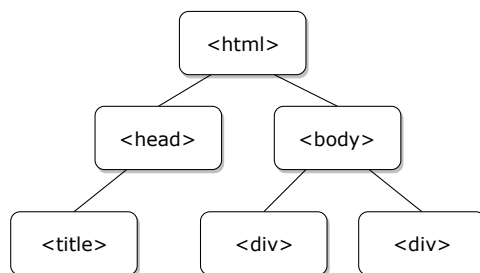
Ako som popísal v 3.1.2, dáta použité na počiatočné trénovanie chatbota sú získané z používateľského fóra Reddit s využitím metódy web-scrapingu. V rámci web-scrapingu je využívaný *Hypertext Transfer Protocol* (HTTP), kedy program odošle na server HTTP požiadavku (HTTP request) na danú webovú stránku a server mu obratom pošle HTTP odpoveď (HTTP response), ktorá obsahuje zdrojový kód stránky v jazyku HTML, s ktorým bude následne program pracovať. Na prácu s HTTP je použitý modul *requests*(verzia=2.19.1)<sup>1</sup>.

Na parsovanie zdrojového kódu stránky používam Python modul *beautifulsoup4* (verzia=4.6.3)<sup>2</sup>, ktorý vracia stromovú štruktúru v ktorej sú uložené jednotlivé elementy stránky (príklad takejto štruktúry zobrazuje 4.1). Následne je možné stromom prechádzať a vyhľadávať elementy, ktoré obsahujú hľadané dáta podľa *tagu* a príslušných atribútov elementu (*id*, *class*).

Dáta sú z Redditu získavané dvomi prístupmi. Jeden prístup získava dáta z fóra ktoré je zamerané na otázky a odpovede a v ňom vyhľadáva záujmy, ktoré používateľ uviedol. Druhý prístup vyhľadá subreddit ktorý je najbližší tomu záujmu, ktorý používateľ uviedol.

<sup>1</sup><https://docs.python.org/3/library/http.client.html>

<sup>2</sup><https://pypi.org/project/beautifulsoup4/>



Obr. 4.1: Príklad stromovej štruktúry HTML kódu

Prístupy majú rozličné postupy, no pracujú na podobnom princípe. Program získá počiatočnú stránku ktorú parsuje a získa stromovú štruktúru v ktorej následne vyhledá elementy, ktoré sú potrebné (napr. elementy ktoré sa odkazujú na konkrétny subreddit). Následne sa program cyklí cez dané odkazy a v sa cyklí cez jednotlivé topiky. Cyklus je ukončený v prípade, že už nie sú ďalšie dáta získateľné alebo je dosiahnutý požadovaný počet výrokov (tento počet je určený programátorom). Nepresiahnutie maximálneho počtu odpovedí kontroluje trieda *ResponseCounter*.

Aby boli odpovede získané z Redditu čo najvhodnejšie, sú spracované a filtrované. Na eliminovanie komentárov s nevhodnou dĺžkou (napr. vety s jedným slovom alebo početné súvetia) je použitý parser, ktorý celý komentár rozloží na jednotlivé vety. Tieto vety sú následne filtrované.

Ako nástroj na filtrovanie je použitý objekt triedy *TrainingSentenceFilter*. Cieľom tejto triedy je eliminovať všetky vety, ktoré nespĺňajú ani základné gramatické pravidlá. Trieda používa modul na kontrolu gramatiky s názvom *language-check* (verzia=1.1)<sup>3</sup> a kontroluje splnenie základných pravidiel, ktoré považujem za nutné. Medzi tieto pravidlá patrí kapitalizovanie vety a správne ukončenie vety (interpunkčné znamienka). Počas filtrácie sú vety kontrolované taktiež na výskyt vulgarizmov. Kontrola výskytu a cenzúra sú implementované pomocou modulu *profanity* (verzia=1.1)<sup>4</sup>.

Dáta získané z Redditu sú ukladané do korpusov v JSON formáte. Pre každý záujem, ktorý používateľ zadal je použitý samostatný korpus. v JSON formáte sú dáta ukladané ako dvojice (kľúč, hodnota). v rámci korpusu je ako kľúč nastavený názov príspevku a v hodnote je uložený zoznam všetkých komentárov, ktoré prešli filtrom. Príklad uložených dát pre záujem „Star Wars“ je zobrazený na 4.2.

```

{"If Star Wars were a Broadway Production, what would some of the musical numbers be?": [
  "Someone needs a Hand?",
  "Who Has the High Ground Now?",
  "A New Hope."
]},
{"What are your feelings on the new Star Wars movies?": [
  "7 and Rogue one are awesome.",
  "8 and Solo sucked.",
  "A fun ride but seems to go nowhere."
]},

```

Obr. 4.2: Dáta získané z Redditu uložené vo formáte JSON

<sup>3</sup><https://pypi.org/project/language-check/>

<sup>4</sup><https://pypi.org/project/profanity/>

## Dáta poskytované modulom

Modul *chatterbot-corpus* (verzia=1.1.2) poskytuje korpusy ktoré obsahujú vzorové dáta, ktoré môžu byť použité na počítačové tréning chatbota. Každý z týchto korpusev obsahuje konverzáciu vo formáte *yml*. Príklad dát obsiahnutý v základných korpusech poskytovaných modulom je zobrazený na 4.3.

```
categories:  
  - food  
conversations:  
  - do you drink  
  - my brain does not require any beverages.  
  - electricity  
  - electricity is nutrition for robots.  
  - are you experiencing an energy shortage?  
  - my processor requires very little power.
```

Obr. 4.3: Príklad konverzácie uloženej v corpuse modulu pre kategóriu jedlo

Modulové korpusy sú použité len na prvotné tréning chatbota. Po prvom tréningu sú tieto konverzácie uložené v databáze a nie je potrebné ich tréningovať znova.

## 4.2 Konverzácia

Celkový koncept konverzácie je tvorený opakovaným reagovaním na predošlý výrok. Reagovanie chatbota pozostáva z viacerých krokov, ktoré sú vykonané pri každom výbere odpovede.

Chatbot je implementovaný v triede s názvom *JeffBot*, ktorá dedí z triedy *ChatBot*.

### 4.2.1 Výber odpovede

Cieľom chatbota z hľadiska konverzácie je reagovanie na výroky od používateľa čo najvhodnejšími odpoveďami. o výber najvhodnejšej odpovede sa používajú logické adaptéry, ktoré poskytuje modul *chatterbot*. Logický adaptér je algoritmus, ktorý podľa nastavených parametrov vyberie odpoveď, ktorá je podľa určených podmienok najvhodnejšia odpoveď. Použitý modul poskytuje niekoľko prednastavených logických adaptérov ako aj možnosť implementácie vlastných logických adaptérov.

Chatbot by sa mal v rámci konverzácie sústrediť na témy, ktoré pred začiatkom konverzácie používateľ zadal ako svoje záujmy. Úspešnosť tejto funkcionality z veľkej časti závisí na výbere odpovedí za použitia logických adaptérov.

Každý logický adaptér má určité parametre, ktoré rozhodujú o postupe pri výbere odpovede. Medzi tieto parametre patrí algoritmus na porovnávanie reťazcov a spôsob výberu odpovede.

## InterestLogicAdapter

Tento logický adaptér som implementoval s cieľom preferencie odpovedí, ktoré sú spojené so záujmami používateľa. InterestLogicAdapter dedí z triedy *LogicAdapter* a pracuje s dátami zozbieranými z fóra Reddit. Príklad získaných dát je zobrazený na obrázku 4.2.

Prvý krok vyhľadávania najvhodnejšej odpovede v tomto logickom adaptéri je vyhľadanie výroku, ktorý je najpodobnejší tomu ktorý zadal používateľ (t.j. ten na ktorý chatbot hľadá reakciu). Logický adaptér cyklom prechádza všetky výroky v danom corpuse pričom má uložený výrok, ktorý má najväčšiu podobnosť. Čím väčšia je podobnosť daného výroku s výrokom od používateľa, tým vyšší je koeficient *confidence*. Na výpočet podobnosti reťazcov je použitá Levenshteinova vzdialenosť (metóda popísaná v 3.1, sekcia: *Konverzácia*).

Po skončení cyklu logický adaptér pracuje s výrokom, ktorý je považovaný za najbližší možný ekvivalent výroku, ktorý zadal používateľ. Následne adaptér vyberie zo všetkých odpovedí, ktoré sú označené ako validné odpovede na daný výrok, jednu konkrétnu odpoveď. Vybraná je tá odpoveď, ktorá je používaná najčastejšie.

Využitie tohoto adaptéra je za účelom sústredenia konverzácie okolo záujmov používateľa. Preto v prípade, že sa nájde v dátach výrok, ktorý je podobný tomu ktorý užívateľ zadal, je hodnota koeficientu *confidence* zvýšená o 0.1. Ak je nájdený výrok navyše otázka, koeficient je zvýšený o 0.2.

## BestMatch

Logický adaptér, ktorý je prednastavený v module *chatterbot* a pracuje na podobnom princípe, ako InterestLogicAdapter. Rozdiel je v dátach, s ktorými adaptér pracuje. BestMatch adaptér pracuje s databázou, ktorá uchováva všetky dáta ktoré sa chatbot naučil. Databáza je typu *SQLite*. Príklad uložených výrokov je zobrazený na obrázku 4.4 a príklad uložených odpovedí spolu s výrokom, na ktorý odpovedajú a počtom výskytov je zobrazený na obrázku 4.5.

```
-- statement --
['id', 'text', 'extra_data']
(1, 'What is AI?', b'\x80\x04'\x94.)
(4, 'Are you sentient?', b'\x80\x04'\x94.)
(13, 'What language are you written in?', b'\x80\x04'\x94.)
(14, 'Python.', b'\x80\x04'\x94.)
(15, 'I am written in Python.', b'\x80\x04'\x94.)
```

Obr. 4.4: Uložene výroky

```
-- response --
['id', 'text', 'created_at', 'occurrence', 'statement_text']
(3, 'Are you sentient?', '2019-03-30 18:42:20', 1, 'Sort of.')
(6, 'Are you sapient?', '2019-03-30 18:42:20', 1, "In all probability, I am not. I'm not that sophisticated.")
(7, 'Are you sapient?', '2019-03-30 18:42:20', 1, 'Do you think I am?')
(8, 'Are you sapient?', '2019-03-30 18:42:20', 1, 'How would you feel about me if I told you I was?')
(9, 'Are you sapient?', '2019-03-30 18:42:21', 3, 'No.')
(10, 'What language are you written in?', '2019-03-30 18:42:21', 1, 'Python.')
(18, 'You are not immortal', '2019-03-30 18:42:21', 1, "As long as I'm backed up I am.")
```

Obr. 4.5: Uložené odpovede s výrokmami a počtom výskytov

V 4.5 sú v druhom riadku vypísané jednotlivé elementy, ktoré každý záznam obsahuje. Jedná sa o text, s ktorým je spojená daná odpoveď, dátum vytvorenia záznamu, počet výskytov danej odpovede a odpoveď samotnú. Adaptér BestMatch pri výbere najfrekvencovanejšej odpovede používa informáciu o počte výskytov zo záznamu danej odpovede.

Adaptér na výpočet podobnosti používa Jaccardovu podobnosť (metóda popísaná v 3.1, sekcia: *Konverzácia*). Vo verzii modulu ktorý je použitý na implementáciu chatbota program vypisuje upozornenia pri každom porovnávaní. To ruší celkový dojem z konverzácie. Implementoval som triedu, ktorá dedí z triedy *JaccardSimilarity* z modulu *comparisons* a upravil som potrebné metódy tak, aby upozornenia nerušili konverzáciu.

## LowConfidenceAdapter

Adaptér ktorý je poskytnutý v module *chatterbot*. Tento adaptér je použitý v prípade, že vybraný výrok ma koeficient confidence nižší, ako je nastavená hranica adaptéru (0.5). v prípade, že koeficient nedosiahne túto hranicu, chatbot zvolí prednastavenú odpoveď („I am sorry, but I do not understand.“).

Zvýšením tejto hranice môže doceliť presnejšie odpovede chatbota vďaka tomu, že chatbot použije danú odpoveď len keď si je veľmi istý správnosťou (o tom ako je presvedčený o správnosti odpovede hovorí koeficient confidence). Zvýšenie môže taktiež viesť k situácií, kedy chatbot nebude odpovedať nijak inak len prednastavenou odpoveďou z dôvodu, že databáza neobsahuje dostatočné množstvo dát aby sa vo väčšine prípadov našiel výrok s minimálnou podobnosťou.

Logické adaptéry sú používané všetky pri každom výbere odpovede. Každý adaptér vráti jednu odpoveď s určitou confidence hodnotou. Následne chatbot použije tú odpoveď, ktorá má najvyššiu hodnotu confidence. v prípade zhody hodnoty je vybraná tá odpoveď, ktorá pochádza z adaptéru ktorý je v inicializácii nastavený ako prvý. Aby chatbot primárne používal výroky z dát zameraných na záujmy užívateľa, InterestLogicAdapter je v inicializácii nastavený ako prvý.

## Problémy pri výbere odpovede

Pri hľadaní výroku podobného tomu, ktorý zadal používateľ môže nastať situácia, kedy vybraný výrok nie je takmer vôbec podobný výroku používateľa. Táto situácia nastáva v prípade, kedy používateľ napíše výrok, ktorý sa nepodobá žiadnemu chatbotu známemu výroku. Aj na príek tomu, že podobnosť je nízka (napr. podobnosť reťazcov spočíva len v jednom slove), je výrok označený za ten s najväčšou podobnosťou. To spôsobí, že vybraná odpoveď nie je odpoveď na používateľov výrok, ale na výrok ktorý mal najväčšiu podobnosť aj keď spolu tieto dva výroky vôbec nesúvisia.

Na kvalitné odpovede od chatbota je potrebné veľká sada overených dát, na ktorých sa môže chatbot učiť alebo veľké množstvo konverzácií s ľuďmi, ktorí nebudú cielene používať nezmyselné odpovede. Po poskytnutí týchto dát je perspektíva implicitnej filtrácie nesprávnych a nevhodných odpovedí. Odpovede sú vybrané na základe počtu výskytu. Ak je určitá odpoveď správna, je používaná opakovane čo zvyšuje jej počet výskytov a tým sa zvyšuje šanca na znovu použitie danej odpovede. Filtrácia nezmyselných odpovedí je implicitná, pretože nezmyselné dáta sú používané minimálne a preto sa počet ich výskytu nezvyšuje a pri výbere odpovede budú preferované zmyselné odpovede ktorých koeficient je vyšší.



## 4.2.2 Trénovanie chatbota počas konverzácie

Konverzácií sa skladá z postupnosti výrokov, ktorých autor sa mení. Konverzáciu je teda možné rozdeliť na kolekciu dvojíc, kde je jedna dvojica zložená z výroku a z odpovede. Tieto dvojice sa prelínajú tak, že v jednej dvojici je reťazec  $M$  v pozícií výroku, na ktorý sa reaguje a v inej dvojici je ten istý výrok  $M$  v pozícií reakcie. Tieto dvojice sú tvorené súbežne s konverzáciou a slúžia tak ako materiál na trénovanie chatbota.

Počas konverzácie si chatbot uchováva posledný výrok, ktorý použil ako odpoveď. Následný výrok od používateľa poskytuje reakciu, ktorú chatbot automaticky považuje za validnú.

Trénovanie chatbota predstavuje pridávanie nových záznamov do databázy. Počas jedného výroku od používateľa sú do databázy pridané dva záznamy. Prvý záznam je tvorený poslednou odpoveďou, ktorú chatbot vybral a nasledujúcim výrokom ktorým používateľ reagoval. Druhý záznam je tvorený vstupom používateľa a odpoveďou, ktorú chatbot vybral.

## Problémy pri trénovaní chatbota

Hlavný problém, ktorý je spojený s trénovaním chatbota je prítomnosť chybných dát. Keďže sa chatbot trénuje z dvojíc, kde jednu časť dvojice poskytuje chatbot a druhú používateľ, chyba sa môže objaviť v oboch častiach.

Chyba na strane chatbota môže byť spôsobená nezmyselnosťou odpovede. Chatbot vyberie odpoveď na výrok, ktorý sa najviac podobá na výrok zadaný používateľom. Ak je podobnosť nízka, odpoveď nemusí dávať vôbec zmysel. Po výbere odpovede je vytvorený záznam ktorý je tvorený výrokom ktorý užívateľ zadal a vybranou odpoveďou a tento záznam je pridaný do databázy ako validný. Kontrolovanie, či odpoveď chatbota je dostatočne zmyselná, by urýchlila trénovanie chatbota validnými odpoveďami. Táto kontrola by ale musela byť vykonaná za každou odpoveďou chatbota a opakovaná až kým chatbot neodpovie správne. Pri kontrole odpovedí chatbota ale nie je hranica, na ktorý pokus chatbot nájde uspokojivú odpoveď, čo môže znamenať neznámy počet opakovaní. Takéto trénovanie ale nie je cieľom používania chatbota používateľom, preto chatbot túto kontrolu neobsahuje.

Použitie chybných dát získaných používateľa vyplýva z celkovej myšlienky chatbota. Keďže je cieľom chatbota učiť a zlepšovať angličtinu používateľa, očakáva sa, že vstupy od používateľa budú obsahovať chyby. Tento vstup je použitý pri trénovaní chatbota a preto je vhodné ho opraviť pred tým, ako je použitý na trénovanie. To je docielené balíčkom *language\_check*, ktorý poskytuje triedu *LanguageClass*. Použitím tejto triedy je do určitej miery možné docieľiť opravu vstupu používateľa pred jeho použitím na trénovanie.

## 4.3 Gramatická kontrola a vyhodnotenie

Gramatická kontrola sa skladá z dvoch častí a je aplikovaná na každý výrok, ktorý používateľ zadal. Vyhodnotená správnosť výroku je užívateľovi zobrazená automaticky počas konverzácie alebo konverzácia nie je prerušovaná a výsledky sú zobrazené používateľovi po ukončení konverzácie.

### 4.3.1 Gramatická kontrola

Prvá časť kontroly gramatiky je používa *language\_check*. Tento Python balíček bol najvhodnejšia možnosť hlavne kvôli dostupnosti. Najväčšia výhoda používania tohoto balíčku je

bezplatnosť. Existujú rôzne organizácie, ktoré poskytujú aplikačné rozhranie ktoré je možné využiť na korekciu gramatiky, no zdarma poskytujú len určitý počet dotazov v rámci skúšobného obdobia. Všetko navyše je následne za poplatok.

Táto kontrola je implementovaná v triede *CorrectClass*, ktorá dedí z triedy *LanguageTool*. Táto trieda implementuje dve metódy, ktoré sú v rámci chatbota použité. Metóda *check()* a metóda *correct*. Metóda *check()* vracia slovníky chýb, ktoré boli nájdené vo vete. Slovník chýb obsahuje informácie o pozícií chyby, typu chyby a potenciálnych opravách chyby. Metóda *correct* opraví chybnú vetu aplikovaním náhrad ktoré sú uložené v slovníku.

Aby bola gramatická kontrola touto triedou najúčinnějšía, je potrebné vykonávať kontrolu vety v cykle, pretože opravenie chýb, ktoré nájde v prvej iterácii môže odhaliť ďalšie gramatické chyby. Príklad na postupnú korekciu zobrazuje obrázok 4.6.

```

1. iterácia
Veta      : "Does yu lik rainy dajs?"
Chyby     : "Does yu lik  rainy dajs?"
           ^ ^  ^^^  ^^^^
Opravené  : "Does you like rainy days?"

2. iterácia
Veta      : "Does you like rainy days?"
Chyby     : "Does you like rainy days?"
           ^^^^
Opravené  : "Do you like rainy days?"

```

Obr. 4.6: Korekcia gramatiky použitím triedy *LanguageCheck*

Druhá časť gramatickej kontroly spočíva kontrolovanie vstupu voči chybným dátam, ktoré boli zozbierané. Tieto dáta sú uložené v korpuse chýb vo formáte JSON. Dáta obsahujú chyby, ktoré ľudia spravili v jednotlivých slovách. v rámci korpusu sú uložené ako hlavné kľúče písmená abecedy kvôli základnému rozdeleniu a rýchlejšiemu prístupu. Následne každá časť obsahuje kolekciu slovníkov, v ktorých je ako kľúč gramaticky správne napísané slovo a ako hodnota je zoznam všetkých gramaticky zle napísaných tvarov daného slova. Na obrázku 4.7 je zobrazený príklad uložených pravopisne zle zapísaných slov.

```

{"n": {
  "negligible": [
    "negligable",
    "neglible"
  ],
  "negotiate": [
    "negociate"
  ],
},
{"p": {
  "pain": [
    "pane",
    "paine"
  ],
  "painful": [
    "painfull",
    "pianful"
  ],
},

```

Obr. 4.7: Chybné dáta uložené v korpuse

V tejto časti kontroly program kontroluje každé slovo, ktoré sa nachádza vo výroku používateľa. Ako prvé sa program cyklí cez korektné zápisy slov a porovnáva, že kontrolované slovo je zaradené ako pravopisne správne. v prípade, že slovo bolo nájdené, program usudzuje pravopisnú správnosť slova a pokračuje na ďalšie slovo. Ak sa vyhľadávané slovo nenašlo medzi korektnými, program sa cyklí cez chybné zapísané slová. Ak sa vyhľadávané slovo nachádza v zozname nesprávne zapísaných slov, program si zaznamená správny tvar slova ku ktorému nesprávny tvar patrí a pokračuje vo vyhľadávaní. Po dokončení cyklu má program ku kontrolovanému slovo uložené všetky slová z ktorých sú v korpuse zaznamenané chybné tvary ktoré sa zhodujú s vyhľadávaným slovom.

Ak sa vyhľadávané slovo nenašlo v korpuse chýb, nemusí to nutne znamenať korektnosť daného slova. Znamená to, že nebola zaznamenaná chyba žiadnom slove, ktoré by sa zhodovalo s vyhľadávaným slovom.

### 4.3.2 Zobrazenie chýb používateľovi

Zobrazenie nájdených chýb vo výrokoch používateľa je možné automaticky počas konverzácie alebo po skončení konverzácie. Čas zobrazenia závisí od používateľa.

Chatbot je určený pre ľudí, ktorí sa angličtinu učia ako druhý jazyk a predpokladá sa, že prvý jazyk je slovenský alebo český. Všetky balíčky a moduly ktoré sú v rámci implementácie použité používajú na vyhodnotenia a komunikáciu angličtinu. z toho dôvodu je potrebný preklad správ, ktoré popisujú nájdené chyby. Na preklad je použitý balíček *translate (verzia=3.5.0)*<sup>5</sup>.

Zobrazovanie nájdených chýb je priamo napojené na gramatické kontroly. z toho vyplýva, že pre jednu vetu sú zobrazené dva výsledky kontroly. Zobrazovanie nájdených chýb má slúžiť používateľovi ako prostriedok na učenie a zlepšovanie, preto je dôležité aby boli zobrazované informácie prínosné.

Pre každú chybu, ktorá bola nájdená vo vete ktorú používateľ zadal, je zobrazené:

- Vysvetlenie chyby
- Pozícia chyby
- Zoznam možných opráv
- Príklad opravy používateľovej vety

Pre každé slovo, ktoré sa zhodovalo s aspoň jedným chybným tvarom v korpuse chýb:

- Zoznam slov, ktoré môžu byť správny zápis chybného slova

Na obrázku 4.8 je zobrazená vyhodnotenie správnosti vety spolu s potenciálnou opravou.

---

<sup>5</sup><https://pypi.org/project/translate/>

```

Vaša veta:
  Does yu lik rainy dajs?
Vyhodnotenie:
  -- Oprava podľa modulu --
    Našla sa možná pravopisná chyba. Pozícia: 5. Možné náhrady: ['you', 'EU', 'nu', 'ye', 'yo', ...]
    Našla sa možná pravopisná chyba. Pozícia: 8. Možné náhrady: ['like', 'link', 'lie', 'lid', 'lip', ...]
    Našla sa možná pravopisná chyba. Pozícia: 18. Možné náhrady: ['days', 'dams', 'dais', 'dads', ...]
    Mali ste na mysli 'do'? „robí“ sa používa iba pre tretiu osobu, ktorá je jednotná (on / ona / ona)..
      Pozícia: 0. Možné náhrady: ['Do']
    Možný správny tvar: Do you like rainy days?

  -- Oprava podľa zozbieraných chýb --
    lik - možný zlý zápis zo slov : ['like', 'lick']

```

Obr. 4.8: Výpis nájdených chýb v používateľovej vete

## 4.4 Cvičenie

Princíp cvičení je opačný tomu v konverzácií a preto je cvičenie dostupné mimo konverzácie. Používateľovi je poskytnutá pravopisná chyba a jeho úlohou je chybu opraviť.

Cvičenia sú implementované dvomi spôsobmi. Náhodné vyberanie chybných zápisov z existujúceho korpusu chýb a umelé generovanie chýb. Všetky typy cvičenie prebiehajú v cykle a v každej iterácii cyklu prebehne výber alebo vytvorenie chybného slova.

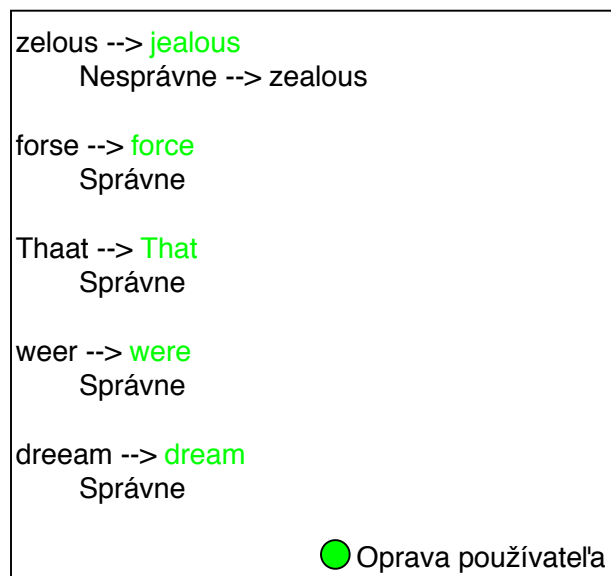
### Cvičenie so slovom

Pri používaní slov z korpusu chýb je výber slova výsledkom troch krokov. v každej iterácii je náhodne vybrané prvé písmeno. v rámci tohoto písmena je náhodne vybrané správne slovo a k nemu náhodne vybraný chybný zápis zo všetkých chybných zápisov daného slova. Nevýhoda tohoto typu cvičenia je bezkontextovosť. Používateľ nevie, v akom kontexte sa slovo nachádzalo v čase keď bolo zapísané chybné. Záleží v tom prípade len na používateľovi a jeho slovnej zásobe. Náročnosť tohoto cvičenia je rôzna kvôli veľkosti zaznamenaných chýb. Chyby boli zozbierané od ľudí z rôznych situácií a rôznych znalostných úrovní. To znamená, že korpus obsahuje chyby, ktoré vznikli zámenou dvoch slov s podobnou skladbou (napr. *tough* a *though*, ako aj chyby ktoré vznikli absenciou znalosti daného slova (napr. „*kíu*“ zo slova *queue*).

Pri chybách ktoré vzniknú generovaním existuje rovnaký problém s kontextom. Používateľ nevie v akom kontexte bolo slovo použité, no oproti používaniu korpusu je náročnosť cvičenia o niečo nižšia a stabilnejšia. Stabilnosť je docielená tým, že chybný zápis sa od korektného stále líši maximálne v dvoch písmenách. Chybný zápis, ktorý vzniká generovaním, je vytvorený jednou z troch možností. Pridanie (zdvojenie) písmena, odstránenie písmena alebo výmena dvoch susedných písmen. Táto možnosť je vyberaná náhodne.

Tým, že si chybný zápis zachováva podobnú štruktúru, používateľ má určitú predstavu ako slovo vyzeralo pri správnom zápise. To redukuje situácie ako napr. so slovo *queue* v predchádzajúcom paragrafe.

Dáta v ktorý sú generované chyby sú získané zo korpusov, ktoré sú zamerané na záujmy užívateľa. z týchto dát sú následne použité len slová, ktoré spĺňajú minimálnu dĺžku slov (3 písmená). Po každej oprave, ktorú používateľ zadal je správnosť opravy vyhodnotená. v prípade, že používateľ opravil nesprávne, je vypísaný aj tvar ktorý je správny. Príklad cvičenia používajúce len jedno slovo je zobrazený na obrázku 4.9.



Obr. 4.9: Cvičenie založené na jednom slove

### Cvičenie s vetou

V rámci tohoto cvičenia sú použité vety, ktoré boli získané pre konverzáciu ktorá sa zameriava na záujmy používateľa.

Pri tomto cvičení pracuje používateľ s celou vetou. Veta môže obsahovať jedno slovo chybné zapísané alebo môže mať zlú syntax. Na používateľovi je opraviť tieto chyby.

Pri pravopisnej chybe sa jedná o jedno slovo v celej vete. Náročnosť opravy chyby v celej vete je náročnejšie, pretože používateľ nevie, o ktoré slovo ide. Na vytvorenie pravopisnej chyby je použitý rovnaký postup ako pri cvičení s jedným slovom. Chybná syntax je vytvorená náhodným zamiešaním slov vo vete. Náročnosť opravy teda závisí od zložitosti a dĺžky vety.

Náhodné zamiešanie môže vytvoriť vetu, ktorá bude od pôvodnej rozdielna len v pozícií dvoch slov. To núti používateľa sústrediť sa na pravopis jednotlivých slov vety a tým si podvedome precvičuje pravopis daných slov.

Aby bol používateľ nútený všímať si, o ktorú chybu sa jedná, chyby sú generované náhodne. Pri každej vete ktorá je získaná je generované náhodné číslo, ktoré rozhoduje o tom aká chyba bude obsiahnutá v danej vete. Príklad na cvičenie používajúce celé vety je zobrazené na obrázku 4.10. v príklade boli ako záujmy uvedené „Star Wars“ a „Harry Potter“. Po ukončení cvičení je používateľovi taktiež zobrazená úspešnosť ktorú dosiahol pri jednotlivých cvičeniach.

Cvičenia sú dostupné v rámci aplikácie a sú dobrovoľné. o tom, či budú alebo nebudú spustené rozhoduje používateľ, takisto ako aj o jednotlivých opakovaníach v rámci cvičení.

parked this only just spot. in no, Why officer, I've -->  
Why no, officer, I've just parked only in this spot.  
Nesprávne --> Why no, officer, I've only just parked in this spot.

Empire, she kissed You way right? in the him saw -->  
You saw the way she kissed him in Empire, right?  
Správne

Shooting ai seems like the best option. -->  
Shoting aim seems like the best option.  
Nesprávne --> Shooting air seems like the best option.

the Potter Post Long and Office Harry Queue. -->  
Harry Potter and the Long Post Office Queue.  
Správne

I mean think abouut all you could do with it!! -->  
I mean think about all you could do with it!!  
Nesprávne --> I mean think about all you could do with it!!

● Oprava používateľa

Obr. 4.10: Cvičenie založené na celej vete

# Kapitola 5

## Testovanie

Táto kapitola je zameraná na testovanie ktoré bolo vykonané s cieľom získania spätnej väzby od používateľov po práci s chatbotom. Testovaná je funkčnosť chatbota a cvičení pri použití používateľom.

### 5.1 Návrh prototypu na testovanie

Dôležitý faktor pri návrhu prototypu na testovanie bola prístupnosť. Bolo nutné brať do úvahy, že cieľový používateľ nemusí mať dostupné všetky nástroje, ktoré sú potrebné na fungovanie chatbota a taktiež nemusí byť ochotný stráviť čas navyše získavaním týchto nástrojov. Z pohľadu dostupnosti bolo najvhodnejšie testovanie dostupné online.

#### Webové rozhranie

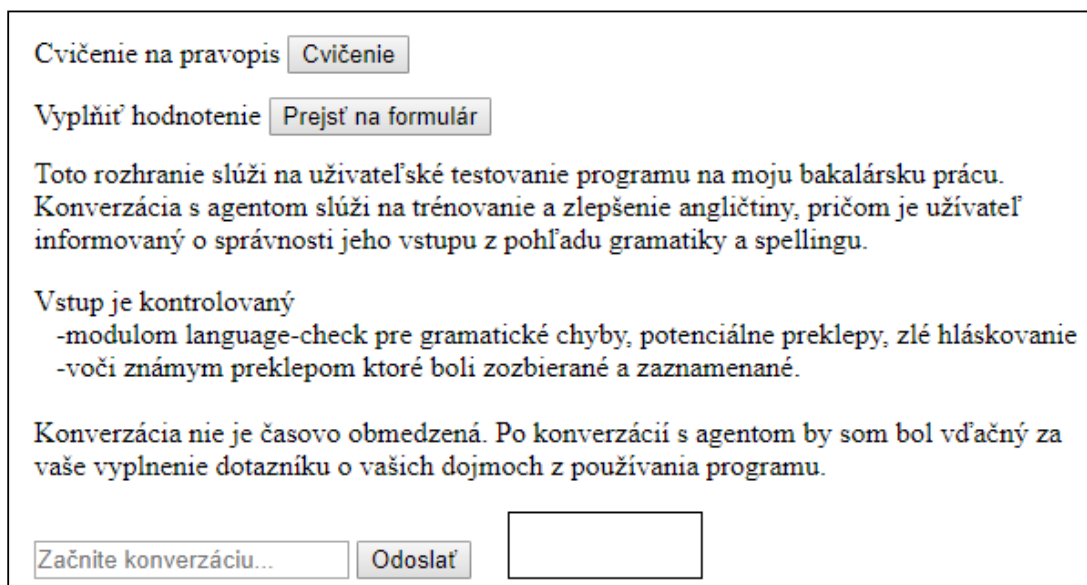
Dostupnosť online bolo možné dosiahnuť úložiskom, z ktorého by používateľ získal všetko čo je potrebné na fungovanie programu alebo aplikáciou, ktorá by bolo priamo dostupná z webového prehliadača. Aby som uľahčil používateľom prácu pri testovaní, zvolil som možnosť dostupnosti priamo vo webovom prehliadači.

Vytvorenie programu napísaného v jazyku Python dostupného cez webové rozhranie je možné viacerými spôsobmi a závisí od náročnosti programu. Na vytvorenie webového rozhrania pre testovací prototyp je použitý *Django*. Django je webový framework napísaný v jazyku Python, pomocou ktorého je možné tvoriť webové stránky podľa architektúry MVT (model–view–template) [10].

Implementácia webového rozhrania s využitím webového frameworku vyžadovala modifikáciu chatbota. Zatiaľ čo v implementácii chatbot funguje ako jeden celok, pri testovanom prototypu bolo potrebné jednotlivé časti rozdeliť a použiť ako aplikačné rozhranie. v architektúre MVT, template predstavuje rozhranie, s ktorým pracuje používateľ. Kostra rozhrania je napísaná v jazyku HTML s využitím CSS na rozloženie elementov na stránke. View predstavuje vrstvu ktorá slúži na komunikáciu používateľského rozhrania s chatbotom. Používateľské rozhranie nie je predmetom testovania a slúži čisto ako nástroj na komunikácie, preto je graficky strohé. Pomocou webového frameworku je možné

vytvoriť aplikáciu, ktorá je dostupná cez webové rozhranie, no nie je dostupná online. Na sprístupnenie som použil *Heroku*. Heroku je *Cloud PaaS*. *PaaS* (Platform as a Service), je služba ktorá poskytuje zákazníkovi vyvíjať, testovať a spúšťať aplikácie bez potreby tvorby a údržby infraštruktúry [4]. Heroku poskytuje bezplatné nasadenie obmedzeného počtu ap-

likácií, no na testovanie to bolo dostačujúce. Príklad prototypu na testovanie prístupného skrz webové rozhranie zobrazuje obrázok 5.1. Testovaný prototyp je dostupný na adrese <https://jeffbot-testing.herokuapp.com/>.



Cvičenie na pravopis

Vyplniť hodnotenie

Toto rozhranie slúži na užívateľské testovanie programu na moju bakalársku prácu. Konverzácia s agentom slúži na tréning a zlepšenie angličtiny, pričom je užívateľ informovaný o správnosti jeho vstupu z pohľadu gramatiky a spellingu.

Vstup je kontrolovaný

- modulom language-check pre gramatické chyby, potenciálne preklepy, zlé hláskovanie
- voči známym preklepom ktoré boli zozbierané a zaznamenané.

Konverzácia nie je časovo obmedzená. Po konverzácií s agentom by som bol vďačný za vaše vyplnenie dotazníku o vašich dojmoch z používania programu.

Obr. 5.1: Používateľské rozhranie v prototypu na testovanie

## Úprava chatbota

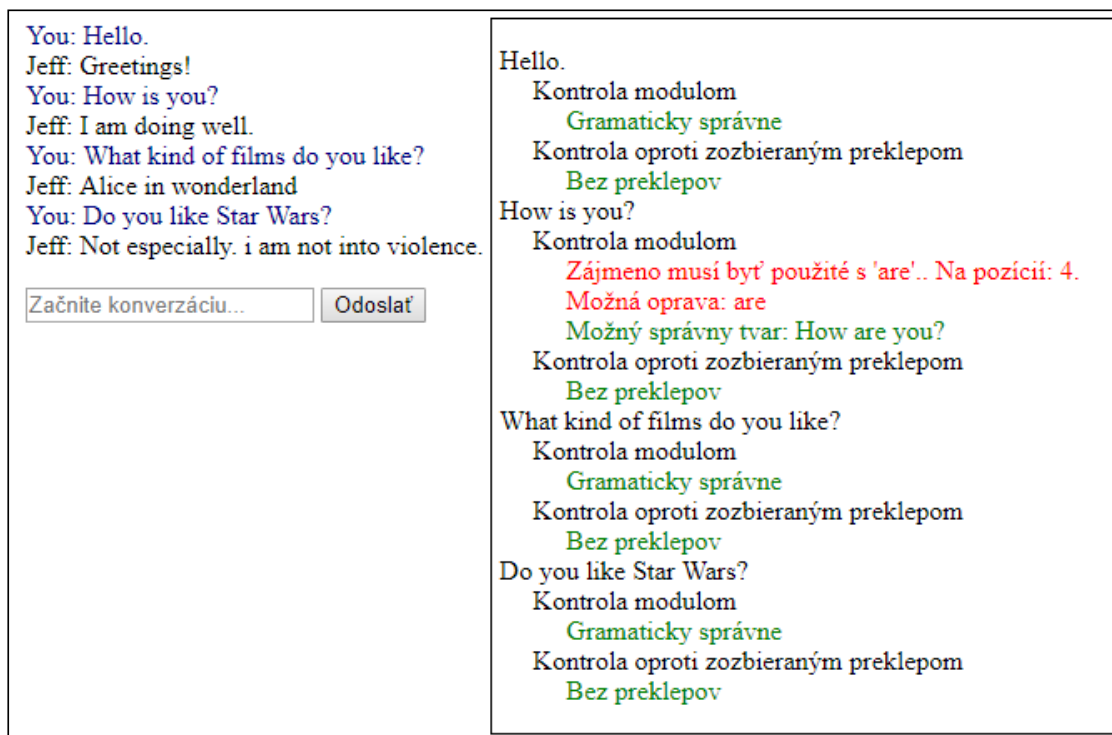
Testovanie sa sústreďuje na prácu používateľov s chatbotom a preto nie je testované získavanie dát. Namiesto získavania dát pri každom používateľovi sú v chatbotovi nastavené ako záujmy univerzálne témy „movies“, „books“ a „games“. k týmto záujmom sú vopred získané dáta a sprístupnené chatbotovi. Dáta, ktoré boli na novo získané mali nevýhodu, že pred testovaním neboli použité a to mohlo zapríčiniť náhodný výber odpovedí pri konverzácií.

Keďže konverzácia je časovo neobmedzená a priestoru vo webovom rozhraní je viac, ako v príkazovom riadku, rozhodol som sa nastaviť mód kontroly a vyhodnotenia správnosti viet automaticky po každom vstupe používateľa. Na informovanie používateľa o gramatickom stave jeho vety je vyhradený samostatný priestor, takže vyhodnocovaním neprerušuje konverzáciu.

V rámci testovania bola poskytnutá možnosť využiť prototyp cvičenia, no nebolo to požadované. Cvičenie bolo upravené tak, že boli generované iba jednotlivé slová a cvičenie nebolo ukončené programom, ale používateľom.

Príklad použitia prototypu pre komunikáciu je zobrazený na obrázku 5.2 a príklad cvičenia a jeho použitia je zobrazený na obrázku 5.3.





Obr. 5.2: Konverzácia a korekcia pri testovaní

## 5.2 Testovací protokol

Chatbot je aplikácia stavaná na prácu s používateľom, preto je dôležitá prívetivosť aplikácie z pohľadu používateľa. Cieľom testovania je teda zistiť názory používateľov na hlavné časti chatbota.

Vďaka spôsobu implementácie, ktorý je popísaný v 5.1, je jediná úloha používateľa chatbota použiť. Použitie by malo byť intuitívne – konverzovať s chatbotom. Konverzácia nie je časovo obmedzená takže dĺžka konverzácie závisí jedine od používateľa. v rámci testovania je možné prepínať sa medzi chatbotom a cvičením.

### Zamerania testovania

Medzi hlavné časti chatbota patrí konverzácia, korekcia viet a cvičenie. Toto sú jediné časti chatbota, ktoré sú viditeľné pre používateľa a preto je potrebné testovanie zamerať na ne.

Ako testovanie konverzačnej časti je úlohou testujúceho používateľa viesť konverzáciu s chatbotom. Hodnotenie tejto časti je veľmi subjektívne, pretože závisí od pohľadu používateľa na zmysluplnosť konverzácie a odpovedí chatbota.

Korekcia a informovanie používateľa o gramatickom stave vety, ktorú zadal je úzko spojená s konverzáciou. Po každej vete ktorou používateľ reaguje na chatbota je veta skontrolovaná a informácie z kontroly sú zobrazené. Testovanie korekcie a hodnotenia spočíva v správnosti opráv a zrozumiteľnosti zobrazených informácií.

Cvičenie je dobrovoľné ale doporučené. Testovací prototyp obsahuje len cvičenie používajúce jedno slovo. Cieľom testovania tohoto cvičenia je zistiť, či to používatelia vidia ako nápomocnú funkčnosť chatbota.

Preklep	Vaše riešenie	Správne riešenie
youngeist	youngest	youngest
overshaddowed	overshadowed	overshadowed
in sis	in sin	inside
desplay	display	display
yuth	youth	youth

Preklep v slove: football

Obr. 5.3: Použitie cvičenia pri testovaní

V rámci spätnej väzby je zisťovaný aj celkový dojem z používania chatbota, názor na jeho reálne využitie a komentáre obsahujúce kladné či záporné pripomienky používateľa.

## Používatelia

Cielová skupina používateľov nie je vekovo obmedzená. Minimálna potrebná technická znalosť predstavuje používanie klávesnice, myši a webového prehliadača na používateľskej úrovni.

Chatbot je tvorený pre ľudí ktorých rodný jazyk je slovenčina alebo čeština a angličtinu sa učia alebo používajú ako druhý jazyk. Minimálna úroveň angličtiny nie je určená, no na minimálny profit z používania chatbota sú vhodné aspoň základy. Maximálna úroveň určená nie je a spätná väzba od používateľov s vyššou úrovňou angličtiny je vhodná na ohodnotenie gramatickej stránky konverzácie a korekcie.

## 5.3 Spätná väzba

Získanie spätnej väzby je formou krátkeho dotazníku ktorý je dostupný z testovacieho prototypu.

Dotazník zisťuje od používateľa:

- Celkový dojem z používania chatbota
- Akú úspešnosť mal chatbot pri hľadaní a opravovaní chýb
- Čitateľnosť, pochopiteľnosť a nápomocnosť zobrazených opráv a vysvetlení
- Zmysluplnosť odpovedí chatbota
- Použitie cvičení
  - Použitelnosť takéhoto typu cvičení
- Názor na použitie chatbota v škole a samoštúdiu
- Vlastné pripomienky

## 5.4 Výsledky testovania

Testovania sa celkovo zúčastnilo 26 ľudí. Úroveň angličtiny testovanej skupiny bola v rozsahu A1 až C1. Konkrétna reprezentácia úrovni angličtiny bola: A1(7.7%), A2(0%), B1(30.8%), B2(46.2%), C1(15.4%).

### Zmyslupnosť odpovedí

Ohodnotenie zmyslupnosti odpovedí bolo formou stupnice, kde 5 značí zmyslupnú, logickú reakciu chatbota a 1 značí nezmyselnú odpoveď.

Celkové ohodnotenie zmyslupnosti je priemerné, prevažujúce na negatívnu stranu. Extrémy označilo rovnomerne 15.4% a 15.4% používateľov, a neutrálne ohodnotilo 26.9% používateľov. 30.8% používateľov sa priklonilo na negatívnu stranu hodnotenia a 11.55% používateľov označili odpovede za prevažne zmyslupné.

### Korekcia viet

Úspešnosť agenta pri vyhľadávaní chýb bola hodnotená stupnicou, kde 1 označuje úplný neúspech a 5 označuje vysokú úspešnosť.

Úspešnosť chatbota pri vyhľadávaní chýb žiaden používateľ neohodnotil číslom nižším ako 3. 61.5% používateľov ohodnotilo úspešnosť vyhľadávania chýb maximálnym počtom bodov a číslom 4 ohodnotilo 26.9% používateľov.

Čitateľnosť vysvetlení pri výpise žiaden používateľ neohodnotil negatívne a kladne ju ohodnotilo 92.3% používateľov. 92.3% taktiež opovažovalo vysvetlenia za nápomocné.

### Cvičenie

Zo všetkých používateľov použilo v rámci testovania cvičenie 73.1%. z týchto používateľov to 60.8% považovalo za nápomocné, 17.4% to považovalo za neutrálne, 17.4% to označilo skôr ako zbytočné než ako nápomocné a 4.3% považovalo toto cvičenie za zbytočné.

### Celkové dojmy a komentáre

96.2% používateľov ohodnotila svoj celkový dojem z používania agenta ako neutrálny až kladný, z čoho kladný dojem chatbot zanechal pri 42.3% testovaných používateľoch.

Z 26 testovaných používateľov si 80.8% z nich vie predstaviť používanie chatbota pri výuke a cvičeniach v škole či ako pomôcku pri samoštúdiu angličtiny.

V komentároch mali používatelia možnosť vyjadriť svoje pripomienky, ktoré neboli zahrnuté v dotazníku. Väčšina komentárov predstavovala pripomienky k dĺžke čakania na odpoveď chatbota, zmyslupnosť odpovedí a celkové smerovanie konverzácie.

### Vyhodnotenie testovania

Výsledky testovania prevažne splnili moje očakávania a predpoklady. Väčšina aspektov ktoré boli testované boli prijaté a ohodnotené neutrálne až kladne, pričom najkritickejšie bola hodnotená konverzácia.

Výber odpovede v rámci konverzácie bolo ohodnotený pozitívne aj negatívne. Rovnomernosť hodnotenia značí, že záležalo od toho aké vstupy zadával používateľ. Tento nedostatok bol spôsobený dátami, ktoré boli použité na tréning testovacieho prototypu.

Na účely testovania boli na novo zozbierané dáta pre tri všeobecné témy (knihy, filmy, hry). Za účelom získania dát na trénovanie bol upravený algoritmus, ktorý zbieral veľké množstvo odpovedí ale malé množstvo výrokov, na ktoré sa odpovedá. To spôsobilo, že pri výbere odpovede chatbot vyhladal výrok, ktorý sa najviac podobá výroku od používateľa, no k tomuto výroku neboli viazané žiadne odpovede. V tomto prípade chatbot použije odpoveď z iného algoritmu a tá odpoveď môže dávať menší až žiadny zmysel. Predpokladá sa, že toto bola príčina negatívnej časti hodnotenia. V prípadoch, kedy používateľ zadal vety, ktoré chatbot dokázal rozoznať a spojiť s možnými odpoveďami, konverzácia a odpovede mali byť zmysluplné. Tomuto tvrdeniu nasvedčuje pozitívna časť hodnotenia.

Zmysluplnosť konverzácie mohla byť ovplyvnená aj počtom záujmov. Na testovanie boli použité tri hlavné témy, ktoré chatbot pri výbere odpovedí uprednostňoval. Chatbot za účelom preferencie týchto tém používal tri sady dát, ktoré nie sú rozdelené, čo mohlo spôsobiť časté zmeny tém. Zmenou témy je myslený výber odpovede zo sady, ktorá prislúcha inému záujmu ako ten, ku ktorému patrila predošlá odpoveď.

Pripomienky boli aj k dĺžke čakania na odpoveď. Dlhé čakanie na odpoveď bolo zapríčinené jednou z metód porovnávania reťazcov. Pri výbere odpovede sú používané viaceré adaptéry spolu s rozličnými porovnávacími metódami, aby bol výber odpovede presnejší. V sekcii 5.5 je popísané čiastočné vyriešenie tohoto problému.

Cvičenie bolo používateľmi hodnotené prevažne pozitívne. V komentároch jeden z používateľov poukázal na bezkontextovosť v rámci cvičenia. Testovací prototyp neposkytoval všetky typy cvičení, ktoré sú obsiahnuté vo finálnej implementácii. Finálna implementácia poskytuje cvičenie, v ktorom je používateľovi kontext známy.

Pri testovaní cvičenia išlo o zistenie využiteľnosti takéhoto spôsobu trénovania angličtiny. Pri hodnotení prospešnosti takéhoto typu cvičenia, na stupnici od 1 (minimum) do 5 (maximum), ho z 23 používateľov ohodnotilo číslom 3 a vyššie 78.2%.

Jedno z najdôležitejších hodnotení je celkové vyhodnotenie práce s chatbotom a jeho využitia. Z pohľadu celkového využitia testovaného chatbota sa 80.8% používateľov vyjadrilo, že si vedia predstaviť jeho využitie ako pomôcku pri zlepšovaní svojich jazykových schopností či už v škole alebo v rámci samostatného vzdelávania.

## 5.5 Aplikácia získaných poznatkov z testovania

Výsledky testovania poukázali na chybný prístup, ktorý som používal pri získavaní dát. Problém bol v reprezentácii dát na trénovanie. Dáta sú získavané vo forme dvojíc reprezentujúce časti konverzácie ako výrok–odpoveď. Pri získavaní bol pred testovaním dôraz na počet odpovedí, pričom mal byť podobný dôraz na počet výrokov, s ktorými sú odpovede spojené.

Tento poznatok som využil na pozmenenie algoritmu, ktorý sa nebude sústreďovať na maximálny počet odpovedí k jednému výroku ale na rozumnejší pomer výrokov a odpovedí. Touto úpravou by sa mala znížiť pravdepodobnosť situácie, kedy chatbot generuje odpoveď náhodne.

Aby sa predišlo častému meneniu sád, z ktorých sú vyberané odpovede, znížil som počet záujmov, ktoré používateľ zadáva na začiatku programu na jeden záujem. To by malo zlepšiť sústredenie konverzácie okolo jednej témy, ktorá používateľa zaujíma.

Experimentovaním som zistil, že dĺžka čakania je extrémne zvýšená pri výpočte podobnosti výrokov použitím Jaccardovej podobnosti. Použitie tejto metódy bolo výhodné pre rôznorodosť vybraných odpovedí, z ktorých bola následne vybraná najvhodnejšia. Pri nahradení Jaccardovej podobnosti Levenshteinovou vzdialenosťou klesla priemerná dĺžka čakania z 90 sekúnd na 30 sekúnd. Z podstaty funkcie, ktorú plní chatbot je pre používateľa prijateľnejšia doba čakania nižšia o  $\frac{2}{3}$  pôvodného času na úkor možného zníženia presnosti odpovede.

Počas experimentovania som ďalej zistil, že výber odpovede cez *LowConfidenceAdapter* trval priemerne 10 sekúnd, pričom táto odpoveď nie je takmer nikdy použitá. Tento adaptér je použitý ako poistka v prípade, že chatbot nenájde žiadnu odpoveď. Nepoužitie toho adaptéru zrýchli výber odpovede chatbota v priemere o 1/3 celkového času odpovede a odstráni možnosť prednastavenej odpovede v prípade nulovej zhody pri hľadaní reakcie.

## Kapitola 6

### Záver

Cieľom práce bolo vytvoriť konverzačný systém, ktorého cieľom bude zlepšovanie jazykových schopností používateľa. Cieľ práce bol splnený využitím konverzácie, korekcie a cvičení ako prostriedky na zlepšenie jazykovej úrovni používateľa. Dáta používané v konverzácií sú získané z používateľského fóra Reddit podľa zvolených záujmov používateľa. Konverzácia bola implementovaná sadou algoritmov strojového učenia využívajúc podobnosť reťazcov. Korekcia gramatiky vstupov používateľa bola implementovaná využitím existujúcich nástrojov a taktiež voči sade zozbieraných chybných dát. Pri cvičení robil korekciu chybných dát používateľ. Chybné dáta použité na cvičenie boli kombináciou zozbieraných chýb a umelo vytvorených chýb.

Výsledný program bol testovaný 26 používateľmi, pričom väčšina hodnotení bolo kladná až neutrálna. Po testovaní sa 80.8% používateľov vyjadrilo, že si vedia predstaviť reálne využitie implementovaného chatbota ako pomocný prostriedok na zlepšovanie jazykových schopností v škole alebo v rámci samostatného vzdelávania.

V prípade pokračovania v práci by bolo možné zamerať sa na ďalšie zdroje dát používaných na tréning chatbota, ako napr. Twitter. Rozšírenie by bolo možné aj v oblasti konverzácií pridaním ďalších algoritmov na výber odpovedí. V časti cvičení by bolo možné použiť iné typy cvičení.

# Literatúra

- [1] Babar, N.: *The Levenshtein Algorithm*. Január 2017, [Online; navštívené 03.05.2019].  
URL <https://www.cuelogic.com/blog/the-levenshtein-algorithm>
- [2] Bayan Abu Shawar, E. A.: *ALICE Chatbot: Trials and Outputs*. [Online; navštívené 07.04.2019].  
URL <https://pdfs.semanticscholar.org/6cd1/153c8da1d4694fe9d5b5d3bb283bc9e999ca.pdf>
- [3] Bayan Abu Shawar, E. A.: *Using dialogue corpora to train a chatbot*. [Online; navštívené 30.04.2019].  
URL [https://www.researchgate.net/profile/Eric\\_Atwell/publication/267575064\\_Using\\_dialogue\\_corpora\\_to\\_train\\_a\\_chatbot/links/54536a9b0cf2bccc490a0890/Using-dialogue-corpora-to-train-a-chatbot.pdf](https://www.researchgate.net/profile/Eric_Atwell/publication/267575064_Using_dialogue_corpora_to_train_a_chatbot/links/54536a9b0cf2bccc490a0890/Using-dialogue-corpora-to-train-a-chatbot.pdf)
- [4] Butler, B.: *PaaS Primer: What is platform as a service and why does it matter?* Február 2013, [Online; navštívené 11.05.2019].  
URL <https://www.networkworld.com/article/2163430/paas-primer--what-is-platform-as-a-service-and-why-does-it-matter-.html>
- [5] Buttazzo, G.: *Artificial consciousness: Utopia or real possibility? Computer*, ročník 34, č. 7, Júl 2001: s. 24–30, ISSN 0018–9162, doi:10.1109/2.933500.
- [6] Callaham, J.: *What is Google Duplex and how do you use it?* Marec 2019, [Online; navštívené 30.04.2019].  
URL <https://www.androidauthority.com/what-is-google-duplex-869476/>
- [7] Clauser, G.: *What Is Alexa? What Is the Amazon Echo, and Should You Get One?* Apríl 2019, [Online; navštívené 30.04.2019].  
URL <https://thewirecutter.com/reviews/what-is-alexa-what-is-the-amazon-echo-and-should-you-get-one/>
- [8] DATAASPIRANT: *IMPLEMENTING THE FIVE MOST POPULAR SIMILARITY MEASURES IN PYTHON*. Apríl 2015, [Online; navštívené 03.05.2019].  
URL <https://dataconomy.com/2015/04/implementing-the-five-most-popular-similarity-measures-in-python/>
- [9] Dormehl, L.: *Today in Apple history: Siri debuts on iPhone 4s*. Október 2018, [Online; navštívené 29.04.2019].  
URL <https://www.cultofmac.com/447783/today-in-apple-history-siri-makes-its-public-debut-on-iphone-4s/>

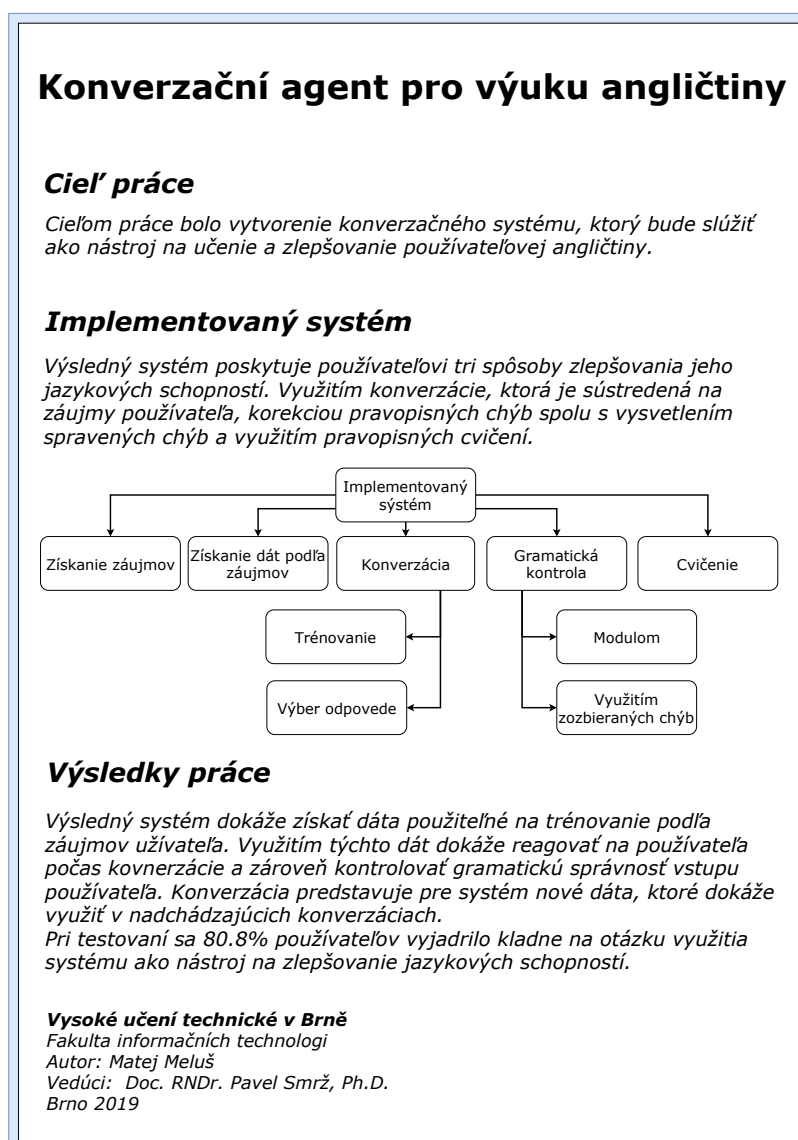
- [10] Foundation, D. S.: *Django*. 2013, [Online; navštívené 25.04.2019].  
URL <https://djangoproject.com>
- [11] Heller, B.; Proctor, M.; Mah, D.; aj.: *Freudbot: An Investigation of Chatbot Technology in Distance Education*. 2005.
- [12] Kerry, A.; Ellis, R.; Bull, S.: *Conversational Agents in E-Learning*. In *Applications and Innovations in Intelligent Systems XVI*, editácia T. Allen; R. Ellis; M. Petridis, London: Springer London, 2009, ISBN 978-1-84882-215-3.
- [13] Kho, J.: *How to Web Scrape with Python in 4 Minutes*. September 2018, [Online; navštívené 02.05.2019].  
URL <https://towardsdatascience.com/how-to-web-scrape-with-python-in-4-minutes-bc49186a8460>
- [14] Lee, Y.-C.; Fu, W.-T.: *Supporting Peer Assessment in Education with Conversational Agents*. In *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion, IUI '19*, New York, NY, USA: ACM, 2019, ISBN 978-1-4503-6673-1, s. 7–8, doi:10.1145/3308557.3308695.  
URL <http://doi.acm.org/10.1145/3308557.3308695>
- [15] van Lun, E.: *Parry*. [Online; navštívené 07.04.2019].  
URL <https://www.chatbots.org/chatbot/parry/>
- [16] Nicol, W.: *What is Reddit? A beginner's guide to the front page of the internet*. Júl 2018, [Online; navštívené 02.05.2019].  
URL <https://www.digitaltrends.com/web/what-is-reddit/>
- [17] Powers, D. M. W.: *The Total Turing Test and the Loebner Prize*. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP3/CoNLL '98*, Stroudsburg, PA, USA: Association for Computational Linguistics, 1998, ISBN 0-7258-0634-6, s. 279–280.  
URL <http://dl.acm.org/citation.cfm?id=1603899.1603947>
- [18] R. Kumar, C. P. R.: *Architecture for Building Conversational Agents that Support Collaborative Learning*. *IEEE Transactions on Learning Technologies*, ročník 4, č. 1, Január 2011: s. 21–34, ISSN 1939-1382, doi:10.1109/TLT.2010.41.
- [19] Rouse, M.: *machine learning (ML)*. Máj 2018, [Online; navštívené 30.04.2019].  
URL <https://searchenterpriseai.techtarget.com/definition/machine-learning-ML>
- [20] Shawar, B. A.; Atwell, E.: *Chatbots: Are they Really Useful? LDV Forum*, ročník 22, 2007: s. 29–49.
- [21] Shum, H.-y.; He, X.-d.; Li, D.: *From Eliza to XiaoIce: challenges and opportunities with social chatbots*. *Frontiers of Information Technology & Electronic Engineering*, ročník 19, č. 1, Jan 2018: s. 10–26, ISSN 2095-9230, doi:10.1631/FITEE.1700826.  
URL <https://doi.org/10.1631/FITEE.1700826>
- [22] Tillman, M.; Grabham, D.: *What is Google Assistant and what can it do?* Január 2019, [Online; navštívené 30.04.2019].



URL <https://www.pocket-lint.com/apps/news/google/137722-what-is-google-assistant-how-does-it-work-and-which-devices-offer-it>

# Príloha A

## Plagát



Obr. A.1: Plagát so stručným popisom projektu

## Príloha B

# Obsah priloženého CD

- `/mistake_corpuses/` - zozbierané chybné dáta
- `/reddit_corpuses/` - úložisko dát získaných z používateľského fóra Reddit používané na tréning chatbota, zložka obsahuje už existujúce súbory pre určité záujmy
- `/src/exercise_generator` - zdrojové kódy používané pri spravovaní cvičení
- `/src/web_scraping/` - zdrojové kódy používané pri získavaní dát z používateľského fóra Reddit
- `/src/` - zdrojové kódy implementujúce hlavné časti systému
- `/latex/` - zdrojové kódy pre vytvorenie technickej správy
- `xmelus01-chatbot-for-learning-english.pdf` - technická správa

## Príloha C

# Inštalácia a spustenie

Na inštaláciu a spustenie aplikácie je potrebný Python 3.7 a prítomnosť určitých balíčkov v systéme. Nainštalovanie potrebných balíčkov je možné príkazom

```
$pip install <názov balíčku>==<verzia>
```

 Balíčky potrebné na beh aplikácie:

- requests==2.19.1
- bs4==0.0.1
- profanity==1.1
- language-check==1.1
- chatterbot==0.8.7
- translate==3.5.0
- SQLAlchemy==1.2.12
- sqlparse==0.3.0

Chatbot sa spúšťa z príkazovej riadky:

```
$python english_agent.py [ew=N] [es=M] [display=DIS] [initlearn=True]
```

kde

- `english_agent.py` je názov hlavného súboru, ktorým je aplikácia spustiteľná
- `[]` sú voliteľné argumenty
- `ew=N` je spustenie chatbota s cvičením slov na konci, v počte N
- `es=M` je spustenie chatbota s cvičením viet na konci, v počte M
- `display=DIS` je nastavenie módu, kedy sa budú používateľovi zobrazovať opravy, pričom DIS môže byť ['after', 'during']
- `initlearn=True` je nastavenie počiatočného tréningu dátami poskytnutých v korpusu