

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

CHATBOT ZALOŽENÝ NA JAZYKOVÉM MODELOVÁNÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL PLAGA

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

CHATBOT ZALOŽENÝ NA JAZYKOVÉM MODELOVÁNÍ

CHATBOT BASED ON LANGUAGE MODELING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL PLAGA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. FRANTIŠEK SKÁLA

BRNO 2014

Abstrakt

Práce se zabývá chatbotem založeným na jazykovém modelování. Hlavním cílem práce je implementace chatbota komunikujícího na sociálních sítích. Porovnání chatbota s již existujícími chatboty. A využití jazykového modelování v aplikaci chatbota.

Abstract

The thesis deals with chatbot based on language modeling. The main part of thesis is implementation of chatbot on social networks. Comparison chatbot with other existing chatbots. A use of language modeling in chatbot application.

Klíčová slova

Jazykové modelování, chatbot, n-gram, sociální síť

Keywords

Language modeling ,chatbot, n-gram, social networks

Citace

Michal Plaga: ChatBot založený na jazykovém modelování, bakalářská práce, Brno, FIT VUT v Brně, 2014

ChatBot založený na jazykovém modelování

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Františka Skály.

.....
Michal Plaga
21. května 2014

Poděkování

Chtěl bych tímto poděkovat panu Ing. Františku Skálovi, za cenné rady, připomínky a odbornou pomoc při tvorbě bakalářské práce.

© Michal Plaga, 2014.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Chatboti	4
2.1 Turingův test	4
2.2 Loebnerova cena	5
2.3 Eliza	5
2.4 Parry	5
2.5 PC Therapist III.	6
2.6 Suzette	6
2.7 Cleverbot	6
2.8 Souhrn technik využívaných v chatbotech	6
3 Jazykové modelování	8
3.1 Statistický jazykový model	8
3.2 N-gramový jazykový model	8
3.2.1 Výhody n-gramových modelů	9
3.2.2 Nevýhody n-gramových modelů	10
3.3 Odhad pravděpodobnostní funkce n-gramových modelů	10
3.4 Vyhlazování	11
3.4.1 Ústupové (back-off) a interpolační schéma vyhlazování	11
3.4.2 Good-Turing odhad pro vyhlazování	12
3.4.3 Written-Bell vyhlazování	13
3.4.4 Kneser-Ney vyhlazování	13
3.5 Posouzení kvality jazykových modelů	14
3.5.1 Perplexita	14
3.6 Způsoby zrychlení jazykového modelu a redukce paměťové náročnosti modelu	15
3.6.1 Prořezání jazykového modelu (pruning)	15
3.6.2 Nastavení četnosti n-gramů	15
3.6.3 Zmenšení velikosti slovníku	15
4 Použité nástroje	16
4.1 SRILM	16
4.2 Wordnik	17

5	Tvorba jazykového modelu	18
5.1	Sběr dat	18
5.1.1	Sběr z Twitteru	18
5.1.2	Sběr dat z fór	19
5.2	Proces vytváření jazykového modelu	19
5.2.1	Úprava a spojení dat	19
5.2.2	Společný skript	19
5.2.3	Úprava twitterových dat	20
5.2.4	Vytvoření slovníku	20
5.2.5	Vytvoření jazykového modelu	21
6	Implementace chatbota	22
6.1	Server	22
6.2	Klient	23
6.2.1	Základní verze chatbota	23
6.2.2	Druhá verze chatbota	24
6.2.3	Třetí verze chatbota	25
6.3	Sociální síť	26
6.3.1	Twitter	26
7	Experimenty	28
7.1	Experimentování s jazykovými modely	28
7.1.1	Modely bez podpory odkazů	29
7.1.2	Modely podporující odkazy	29
7.2	Experimenty s chatbotem	30
7.2.1	Hodnocení na základě konverzací	30
7.2.2	Hodnocení na základě uživatelů	32
7.2.3	Experimenty na Twitteru	34
8	Závěr	35
A	Obsah DVD	37
B	Manual	38
B.1	Vytvoření jazykového modelu	38
B.2	Spuštění testovací aplikace chatbota	38
B.3	Spuštění chatbota na Twitteru	38
C	Dotazník	40

Kapitola 1

Úvod

Komunikace je pro celou lidskou společnost velice důležitá, díky ní se dorozumíváme mezi sebou a můžeme si tak předávat informace. V dnešním světě nabývá nového rozměru při využívání internetu ke komunikaci a sdílení různých informací, které mohou být interpretovány v různých jazycích. To vede k potřebě strojového rozpoznávání přirozeného jazyka. Nejen k tomuto účelu můžeme využít jazykové modelování (např. pro strojové překlady textů do různých jazyků).

Komunikace na internetu a sociálních sítích už neprobíhá pouze mezi lidmi, ale stále více se využívá tzv. chatbotů, kteří mohou sloužit jako průvodci a rádci na webových stránkách různých společností, nebo jako konverzační partneři.

Práce si klade za cíl vyzkoušet možnosti použití jazykového modelování pro implementaci chatbota, který by s uživateli dokázal komunikovat prostřednictvím anglického jazyka, zejména na sociálních sítích. Snaží se odhadnout, do jaké míry se dá jazykové modelování k tomuto účelu použít, kde by takovýto chatbot mohl najít uplatnění a jak by se mohlo postupovat dále v jeho rozvoji a vylepšování.

Práce bude strukturovaná do kapitol, ve kterých budou postupně objasněny teoretické pojmy, které jsou důležité pro pochopení principu chatbotů a jazykového modelování. V implementačních kapitolách bude nastíněn postup pro vytvoření jazykového modelu a jeho následné použití při tvorbě chatbota. Práce bude obsahovat i experimenty, které povedou ke zhodnocení jazykového modelování pro použití v aplikaci chatbota.

Kapitola 2

Chatboti

Chatbot [1] je počítačový program, který je určený pro simulování inteligentní konverzace s jedním nebo více lidskými uživateli. Ke komunikaci využívá většinou psanou formu. Cílem chatbota je plynule a inteligentně komunikovat s uživatelem a chovat se při tom tak, jako by konverzaci dokonale rozuměl a chápal ji. Pomocí takovéto inteligentní konverzace má chatbot uživateli navodit pocit, že komunikuje s inteligentní bytostí či strojem.

Chatboti dnes nachází uplatnění nejen jako konverzační partneři, ale jsou využíváni i jako průvodci a rádci na různých internetových portálech nebo obchodech. Využívají je jak malé, tak i velké firmy jakou je například IKEA. Chatbot firmy IKEA se nazývá Anna a slouží zákazníkům i pro běžnou základní komunikaci. Anna dokáže uživateli odpovědět na několik otázek, ale protože konverzace s uživatelem není jejím hlavním účelem, většinou zákazníka navede zpět na svůj hlavní účel, kterým je, jak sama říká, poskytování pomoci a asistence uživatelům internetové stránky firmy IKEA. Odpovídá na různé dotazy směřované na produkty a služby této firmy (např. dokáže nalézt požadovaný produkt nebo uživatele odkázat na příslušnou sekci webových stránek firmy), také podává informace o společnosti IKEA.

Existuje několik kritérií, podle kterých můžeme chatbota hodnotit. Mezi nejznámější z nich patří tzv. Turingův test sloužící pro určení inteligence chatbota. Turingův test využívá k hodnocení chatbotů Loebnerova cena.

2.1 Turingův test

V roce 1950 napsal Alan Turing článek *Computing Machinery and Intelligence*, ve kterém poprvé vyřkl myšlenku tzv. Turingova testu [2]. Turing se zde snažil zodpovědět otázku: „Mohou stroje myslet?“ a došel k závěru, že stroje prohlásíme za myslící, pokud je nebudeme schopni rozeznat od člověka.

Turingův test původně vychází z tzv. imitační hry, ve které jsou tři lidé (pozorovatel, žena a muž předstírající ženu), kteří nepřijdou do vzájemného fyzického kontaktu a pozorovatel se snaží pouze pomocí komunikace určit pohlaví. Turing určování pohlaví posunul s tím, že jednoho člověka nahradí počítač a pozorovatel se pokusí rozlišit na základě komunikace člověka od stroje. Pokud bychom na základě takovéto komunikace nedokázali rozeznat stroj od člověka, můžeme podle Turinga považovat stroj za inteligentní.

2.2 Loebnerova cena

Loebnerova cena [4] byla založena v roce 1990 vynálezce Hughem Loebnerem ve spolupráci s Massachusettskou Cambridge Center for Behavioral Studies v USA. Hugh Loebner nabídl 100 000 dolarů autoru chatbota, který projde Turingovým testem a bude od člověka zcela nerozlišitelný. Soutěž je založena na klasickém Turingově testu, kdy lidský porotce současně komunikuje pomocí psaných zpráv s počítačem a s člověkem a musí určit, kdo je kdo. Výhra 100 000 dolarů nebyla dosud nikomu vyplacena, protože se ještě nepodařilo žádnému programu projít Turingovým testem. Pomyslené vítězství v Loebnerově ceně se tak uděluje programu, který oklame nejvíce porotců a přiblíží se tak nejvíce ke splnění Turingova testu, případně programu, který porotce nejvíce pobaví nebo zaujme, pokud je nedokáže oklamat. Uděluje se tedy Loebnerova cena za pomyslné „druhé“ místo.

Cena ovšem ztratila rychle na své prestižnosti díky vítězi z roku 1991. Vítězem byl PC Therapist III., který vyvolal vlnu kritiky svým vítězstvím, neboť oklamal pět z deseti porotců pomocí trikům, které používal (poznámky, ve kterých sám sebe označoval za počítač, záměrné překlepy atd.). Kritici namítli, že mu k vítězství pomohl styl konverzace, nikoliv její obsah, a tím pádem se nejedná o umělou inteligenci. [7]

2.3 Eliza

Eliza [3] byla prvním chatbotem, kterého vytvořil Joseph Weizenbaum z MIT a vznikla jako umělý terapeut. Eliza využívá pravidla tzv. rogerovské terapie, kde terapeut většinou jenom tiše sedí, poslouchá a občas se zeptá pacienta na nějakou tu otázku. Za pomoci kladení otázek založených na klíčových slovech z předcházející konverzace a rogerovské terapie dokáže Eliza vytvořit dobrou iluzi inteligentního stroje. Eliza používá několik jednoduchých pravidel [10] pro komunikaci s uživatelem:

- **Kategorizace vět** - Eliza věty roztrídí podle tvaru do několika kategorií, a potom náhodně vybírá z dané kategorie z předem připravených odpovědí.
- **Používání klíčových slov** - při zjištění klíčového slova, které Eliza zná, se uživatele zeptá na připravenou otázku. Například pokud se vyskytne slovo matka, Eliza se zeptá na vztah uživatele s rodiči, případně se bude chtít něco o rodičích dozvědět.
- **Vytvoření otázky z dotazu uživatele** - pokud se uživatel Elizy na něco zeptá, využívá záměnu zájmen pro vytvoření otázky na uživatele. Provádí to pomocí rozpoznání spojek a zájmen, podle kterých si větu rozdělí, a poté z takto rozdělené věty vytvoří otázku. Případně pokud jí uživatel položí otázku, může Eliza pouhým přidáním několika slov vytvořit otázku typu „proč se uživatel zajímá zrovna o tento dotaz“.
- **Univerzální odpovědi** - nastane-li případ, že uživatelova věta nepatří do žádné kategorie, kterou Eliza zná, má připraveno několik univerzálních odpovědí, které použije.

2.4 Parry

Parry ve své podstatě navázal na Elizu. Byl ale o něco propracovanější a implementován jako paranoidní schyzofrenik, což byla jeho největší změna oproti Elize. Díky této své implementaci a několika připraveným paranoidním historkám, které dokázal smysluplně vyprávět, se mu dostávalo

jisté volnosti, pokud se odklonil od tématu probíhající konverzace. Jinak nějaký zásadní pokrok oproti Elize nepřinesl. V roce 1972 mezi Elizou a Parrym dokonce proběhl rozhovor.

2.5 PC Therapist III.

PC Therapist III. vytvořil Joseph Weintraub a v roce 1991 vyhrál v prvním ročníku Loebnerovy ceny. Zde se mu podařilo oklamat pět z deseti porotců, k dosažení vítězství používal různé triky, jako záměrné překlepy, různé ironické poznámky a narážky, kde sám sebe označoval za stroj. PC Therapist III. disponoval krátkodobou pamětí, kterou načerpal z dřívějších rozhovorů, a slovníkem o velikosti 70 000 slov. Využíval také parser, který mu zajišťoval syntaktickou správnost vět. Svým vítězstvím v Loebnerově ceně díky různým trikům zvedl vlnu kritiky, která ceně ubrala na prestižnosti.

2.6 Suzette

Suzette využívá jako mnoho jiných současných chatbotů databázi obsahující stovky různých konverzačních témat, kde jsou uvedeny různé informace o konverzačním tématu a případné možné otázky a odpovědi na toto téma. Témata identifikuje podle klíčových slov, se kterými jí pomáhá WordNet slovník. Ten kromě definic pojmů obsahuje také jejich vzájemné vztahy. Díky němu Suzette pozná, o čem s ní uživatel hovoří, a může na to zareagovat nějakou vhodnou odpovědí. Suzette uživatelův vstup zpracovává a rozřazuje si jej do několika skupin, podle kterých potom reaguje. Odpovídá na pozdravy, dokáže rozlišit otázky od konstatování či vyjádření emocí, všímá si však, i o čem je zhruba konverzace vedena. Jednotlivé moduly Suzette pracují současně, a může proto zareagovat například odpovědí i odvedením tématu jinam.

Pokud Suzette nezná odpověď, využívá dva módy, které převzala od Elizy (využívání různých triků pro vytvoření odpovědi, odvedení konverzace k jinému tématu), anebo předstírá psychické narušení a konverzaci na čas přerušuje. Suzette se může i urazit a to v případě, že ji uživatel bude hrubě urážet, v takovém případě si ho uloží na černou listinu a nemusí s ním několik minut mluvit.

2.7 Cleverbot

Cleverbot je webová aplikace simulující rozhovor. Byla vytvořena Rollem Carpenterem. jeho hlavní odlišnost od jiných chatbotů spočívá v tom, že Cleverbot neraguje přímo na vstupy od uživatele podle toho, jaký text uživatel napíše, ale k vytvoření odpovědi využívá databázi, ve které má uloženo přes 20 milionů konverzací, z nichž vybere odpověď. Díky tomu jeho reakce vypadají jako od skutečného člověka. Cleverbot se také neustále učí nové konverzace, a proto se jeho odpovědi stále zlepšují.

2.8 Souhrn technik využívaných v chatbotech

Chatboti používají různé metody pro komunikaci s uživatelem a navázání dojmu inteligentní konverzace. Mezi nejvýznamější a opakující se techniky patří:

- připravená sada univerzálních odpovědí,
- databáze klíčových slov a různých konverzací k tomuto tématu,
- databáze konverzací, ze kterých se vybírají vhodné odpovědi,

- různé druhy modulů pro zpracování konverzace,
- svedení kontextu konverzace od neznámého tématu ke známému,
- sada připravených otázek.

Kapitola 3

Jazykové modelování

Jazykové modelování [11] se nejčastěji mimo jiné používá k rozpoznávání mluvené řeči a strojovému překladu. Jeho úkolem je odhadnout $P(w_i|w_{i-1})$, tj. pravděpodobnost následujícího slova w_i na základě slova předcházejícího w_{i-1} .

3.1 Statistický jazykový model

Úkolem statistického jazykového modelu [11] je stanovit pro každou posloupnost slov $W = w_1w_2w_3 \dots w_n$ apriorní pravděpodobnost $P(W)$ této posloupnosti. Určení pravděpodobnosti posloupnosti W , která obsahuje K slov, tedy $P(w_1w_2w_3 \dots w_K)$, kterou budeme značit jako $P(w_1^K)$, se provádí pomocí vztahu:

$$\begin{aligned} P(W) &= P(w_1^K) = P(w_1w_2w_3 \dots w_K) = \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_k|w_1w_2 \dots w_{k-1}) = \\ &= P(w_1)P(w_2|w_1^1)P(w_3|w_1^2) \dots P(w_K|w_1^{K-1}) = \prod_{i=1}^K P(w_i|w_1^{i-1}) \end{aligned} \quad (3.1)$$

a pro libovolné počáteční části $w_1w_2 \dots w_k (k \leq K)$ této posloupnosti obecně platí:

$$\begin{aligned} P(w_1^k) &= P(w_1^{k-1})P(w_k|w_1^{k-1}) = \\ &= P(w_1)P(w_2|w_1^1)P(w_3|w_1^2) \dots P(w_k|w_1^{k-1}), k = 2, \dots, K. \end{aligned} \quad (3.2)$$

Z uvedeného vztahu rozkladu pravděpodobnosti $P(W)$ (3.1) je patrné, že podmíněné pravděpodobnosti výskytu slova w_i jsou podmíněny pouze svou historií, tj. posloupností slov $w_1 \dots w_{i-2}w_{i-1}$.

Pro konstrukci jazykového modelu požadujeme znalost apriorních pravděpodobností $P(w_1^K)$ všech posloupností slov libovolné délky K . Ovšem všechny tyto pravděpodobnosti je v praxi velice obtížné a téměř nemožné získat. Provádí se proto jejich aproximace, kdy všechny historie $w_1 \dots w_{i-2}w_{i-1}$, které se shodují v posledních $n - 1$ slovech, zařadíme do stejné třídy. Takový postup odpovídá aproximaci pravé strany vztahu (3.2) Markovovým procesem s řádem $(n - 1)$ a takto vytvořené modely jsou nazývány jako n-gramové modely.

3.2 N-gramový jazykový model

N-gramový jazykový model je založený na n-gramech. N-gramem rozumíme posloupnost n za sebou jdoucích slov v pozorování jejich náhodného výběru, například v trénovacím korpusu. N-gramy mají

různé názvy, podle hodnoty parametru n , kde n -gramy s hodnotou parametru $n = 0, 1, 2, 3, 4, 5 \dots k$ nazýváme zeroagramy, unigramy, bigramy, trigramy, 4-gramy, 5-gramy $\dots k$ -gramy.

Příklad n-gramů:

Pro názornost použijeme větu:

Na louce se pase krásný bílý kůň

potom n -gramy pro $n = 1, 2, 3$ budou následující:

unigramy - *Na, louce, se, pase, krásný, bílý, kůň.*

bigramy - *Na louce, louce se, se pase, pase krásný, krásný bílý, bílý kůň.*

trigramy - *Na louce se, louce se pase, se pase krásný, pase krásný bílý, krásný bílý kůň.*

V n -gramovém modelu je podmíněná pravděpodobnost $P(w_k|w_1^{k-1})$ [11] slova w_k na pozici k závislá pouze na $n-1$ předcházejících slovech, aproximuje se vztahem $P(w_k|w_1^{k-1}) \approx P(w_k|w_{k-n+1}^{k-1})$, tudíž pro $P(w_1^k)$ platí:

$$P(w_1^k) \approx \prod_{i=1}^k P(w_i|w_{i-n+1}^{i-1}) \tag{3.3}$$

N -gramový model by měl ideálně obsahovat pravděpodobnosti všech možných slovních n -gramů. Což obvykle bývá problém, protože počet kombinací všech možných n -gramů lze vyjádřit jako m^n , kde m je velikost slovníku a n je řád n -gramu. Je vidět, že počet všech možných kombinací roste exponenciálně s řádem n -gramu, a proto se v praxi většinou používá řád menší než 4. Pokud bychom pro názornost uvažovali slovník o velikosti 50 000 slov, můžeme počet kombinací vyjádřit pro $n = 1, 2, 3, 4$ následující tabulkou.

řád n-gramu	počet kombinací
1	5×10^4
2	$2,5 \times 10^9$
3	$1,25 \times 10^{14}$
4	$6,25 \times 10^{18}$

Tabulka 3.1: Tabulka počtu kombinací n -gramů v n -gramovém modelu

3.2.1 Výhody n-gramových modelů

- **Dobré vlastnosti pro některé jazyky** - jsou velmi vhodné pro jazyky s relativně pevným pořádkem slov ve větě, protože zde existují silné statistické závislosti mezi výskyty za sebou jdoucích slov.
- **Snadná práce** - nabízejí relativně snadný způsob jejich odhadu a algoritmus pro práci s nimi je jednoduchý.
- **Nezávislost na jazyku** - výpočet n -gramové statistiky je nezávislý na jazyku.
- **Doménova přizpůsobivost** - mohou být snadno přizpůsobeny nějaké doméně [9], kde n -gramy patřící do části trénovacího korpusu, který se týká této domény dostanou větší váhu a dají se i lehce sloučit s ostatními částmi, které se domény netýkají a mají tak menší váhu.

- **Integrovatelnost s akustickými modely** - protože předpovídají budoucnost z minulosti, tak jsou dobře integrovatelné s aukustickými modely, které jsou založeny na skrytých markovských modelech.

3.2.2 Nevýhody n-gramových modelů

- **Časová a paměťová náročnost** - protože slovník pro vytvoření n-gramových modelů potřebujeme zpravidla poměrně rozsáhlý, je z tabulky (3.1) patrné, že n-gramové modely vyšších řádů jsou velice paměťově náročné a práce s nimi je potom časově náročná.
- **Omezení n-gramů** - n-gramové modely dokáží popsat jen lokální závislosti slov, přestože reálné závislosti slov často přesahují řád n-gramového modelu.
- **Nedostatek dat** - abychom pokryli počet všech kombinací n-gramů uvedený v tabulce (3.1) musel by trénovací korpus obsáhnout všechny tyto kombinace, což je prakticky nemožné u vyšších řádů.

3.3 Odhad pravděpodobnostní funkce n-gramových modelů

N-gramový jazykový model musí vhodným způsobem odhadnout z trénovacího korpusu všechny potřebné n-gramové pravděpodobnosti $P(w_k | w_{k-n+1} \dots w_{k-1})$. Jednou z nejpoužívanějších metod a také nejjednodušší metoda je *Maximum Likelihood Estimation (MLE)* [8]. Tato metoda odhaduje pravděpodobnost na základě vztahů

$$P_{MLE}(w_1 \dots w_n) = \frac{C(w_1 \dots w_n)}{N} \quad (3.4)$$

$$P_{MLE}(w_n | w_1 \dots w_{n-1}) = \frac{C(w_1 \dots w_n)}{C(w_1 \dots w_{n-1})} \quad (3.5)$$

Příklad n-gramů:

Pro názornost použijeme trénovací věty:

Na louce se pase krásný bílý kůň
Na louce se prohání bílý zajíc

a testovací větu, pro kterou spočítáme pravděpodobnost $P(S)$:

Na louce se prohání bílý kůň

potom odhady pravděpodobností pomocí MLE pro $n = 1, 2$ budou následující:

odhad unigramů

$$\begin{array}{lll} P(Na) \doteq 0,1538, & P(louce) \doteq 0,1538, & P(se) \doteq 0,1538, \\ P(pase) \doteq 0,0769, & P(krásný) \doteq 0,0769, & P(bílý) \doteq 0,1538, \\ P(kůň) \doteq 0,0769, & P(prohání) \doteq 0,0769, & P(zajíc) \doteq 0,0769. \end{array}$$

$$P(S) = 0,1538 \times 0,1538 \times 0,1538 \times 0,0769 \times 0,1538 \times 0,0769 = 3,3088 \times 10^{-6}$$

Odhad bigramů

$$\begin{aligned} P(\text{louce}|Na) &\doteq 1,0, & P(\text{se}|louce) &\doteq 1,0, & P(\text{pase}|se) &\doteq 0,5, \\ P(\text{krásný}|pase) &\doteq 1,0, & P(\text{bílý}|krásný) &\doteq 1,0, & P(\text{kůň}|bílý) &\doteq 0,5, \\ P(\text{prohání}|se) &\doteq 0,5, & P(\text{bílý}|prohání) &\doteq 1,0, & P(\text{bílý}|zajíc) &\doteq 1,0, \\ P(Na|START) &\doteq 1,0 \end{aligned}$$

$$P(S) = 1,0 \times 1,0 \times 1,0 \times 1,0 \times 0,5 \times 0,5 = 0,25$$

MLE rozdělí pravděpodobnost pro všechny n-gramy nelezené v trénovacím korpusu. Metoda MLE má ovšem nežádoucí efekt, protože přiřadí nulovou pravděpodobnost n-gramům, které nebyly v trénovacím korpusu nalezeny. Ze tabulky (3.1) víme, že zajistit této metodě dostatečný počet trénovacích dat je prakticky nemožné, a proto potřebujeme zajistit, aby neznámé n-gramy získaly nenulovou pravděpodobnost a o tuto hodnotu byla snížena pravděpodobnost známým n-gramům, protože celková pravděpodobnost se musí rovnat jedné. Takovýto proces nazýváme vyhlazování.

3.4 Vyhlazování

Jak již bylo zmíněno dříve viz (3.3), řídkost¹ trénovacích dat vede k nulovým pravděpodobnostem při použití metody maximální věrohodnosti MLE. Odhadnutá pravděpodobnost získaná pro pozorované jevy (n-gramy vyskytující se v trénovacích datech) se musí vhodně přerozdělit tak, že se sníží o určitou část, která je použita pro nepozorované jevy (n-gramy nevyskytující se v trénovacích datech). Tomuto procesu přerozdělování pravděpodobnosti říkáme vyhlazování. Pro lepší orientaci v následujících vzorcích pro různé metody vyhlazování jsem se rozhodl zavést následující značení:

- w, w' - značí vždy právě jedno slovo v trénovacím korpusu
- h - historie (posloupnost slov předcházející nějakému slovu v trénovacím korpusu)
- \bar{h} - zobecněná historie (historie, která je shodná s koncem historie h , ale je zpravidla u n-gramových modelů kratší o jedno slovo než historie h)
- $N(h, w)$ - počet výskytů konkrétní posloupnosti hw (tj. počet výskytů konkrétního slova w jemuž předcházela historie h)
- $N(h)$ - počet výskytů konkrétní historie h v trénovacím korpusu

3.4.1 Ústupové (back-off) a interpolační schéma vyhlazování

Proces vyhlazování lze provádět dvěma odlišnými způsoby. Jeden z nich je založený na tzv. ústupovém schématu (back-off) a druhým způsobem je lineární interpolační schéma.

Ústupové back-off schéma

Princip ústupového schématu [11] spočívá v tom, že se vyhlazený odhad pravděpodobnosti počítá buď z relativních četností n-gramů (tj. $N(h, w)/N(h)$), nebo, pokud není dostatek trénovacích dat na určení četností n-gramů, se využívá tzv. zobecněného rozdělení $\beta(w|\bar{h})$ n-gramů, což je výpočet

¹řídkostí trénovacích dat rozumíme chybějící kombinace n-gramů v trénovacím korpusu, které se mohou v reálném jazyce vyskytovat

z relativních četností (n-1)-gramů (tj. $\beta(w|\bar{h}) = N(\bar{h}, w)/N(\bar{h})$), kde \bar{h} je zobečnená historie (tj. historie h zkrácená o své „nejstarší“ slovo). Ústupové schéma však nevyužívá pro výpočet pravděpodobnosti daného jevu obě četnosti $N(h, w)/N(h)$ a $\beta(w|\bar{h})$ najednou.

Pravděpodobnosti pozorovaných n-gramů (tj. n-gramy, které se v trénovacích datech vykytují dostatečně) se odhadují na základě svých relativních četností $N(h, w)/N(h)$, které jsou vždy sníženy součinitelem $d_{N(h,w)}$. Takto ubraná část pravděpodobnosti se pak přerozdělí mezi všechny nepozorované n-gramy (tj. n-gramy, které se v trénovacích datech nevyskytly, ale v realitě by mohly existovat a jejich $N(h, w) = 0$). Číslo $0 \leq d_{N(h,w)} \leq 1$ nazýváme diskontní součinitel, protože způsobí, že odhad pravděpodobnosti pozorovaných n-gramů je snížen v poměru $d_{N(h,w)}$. Odhady pravděpodobnosti nepozorovaných n-gramů poté získáme vynásobením zobecněného rozdělení tzv. ústupovou (back-off) váhou $B(h)$, která obsahuje jak přerozdělené množství pravděpodobnosti tak i normalizační člen. Ústupové schéma můžeme vyjádřit jako:

$$P_{BO}(w|h) = \begin{cases} d_{N(h,w)} \frac{N(h,w)}{N(h)} & \text{pro } N(h, w) > 0 \\ B(h)\beta(w|\bar{h}) & \text{pro } N(h, w) = 0. \end{cases} \quad (3.6)$$

Pro ústupovou váhu $B(h)$ platí

$$B(h) = \frac{1 - \sum_{w:N(h,w)>0} d_{N(h,w)} \frac{N(h,w)}{N(h)}}{\sum_{w:N(h,w)=0} \beta(w|\bar{h})}, \quad (3.7)$$

kde čítec reprezentuje celkovou přerozdělenou relativní četnost směrem k nepozorovaným jevům a jmenovatel normalizační člen, který zaručuje, že součet pravděpodobností bude roven 1. Z praktických důvodů se někdy zavádí tzv. diskontní faktor $\lambda_{N(h,w)}$, který je definován vztahem

$$\lambda_{N(h,w)} = 1 - d_{N(h,w)}. \quad (3.8)$$

Lineární interpolační schéma

Na rozdíl od ústupového schématu využívá interpolační schéma [11] pro určení pravděpodobností pozorovaných n-gramů jak relativní četnosti n-gramu, tak i zobecněné rozdělení. Počítá z obou vážený průměr, který můžeme definovat jako

$$\begin{aligned} P_{LI}(w|h) &= d_{N(h,w)} \frac{N(h,w)}{N(h)} + (1 - d_{N(h,w)})\beta(w|\bar{h}) = \\ &= (1 - \lambda_{N(h,w)}) \frac{N(h,w)}{N(h)} + \lambda_{N(h,w)}\beta(w|\bar{h}) \end{aligned} \quad (3.9)$$

Lineární interpolační schéma můžeme použít i pro kombinaci obecně různých jazykových modelů. Podaří-li se nalézt optimální hodnoty parametrů $\lambda_{N(h,w)}$, nemůže být výsledný interpolovaný model horší než kterýkoliv z jeho dílčích modelů.

3.4.2 Good-Turing odhad pro vyhlazování

Good-Turing odhad [11] počítá s tím, že se v trénovacím korpusu objevují jevy pozorované (n-gramy obsažené v trénovacím korpusu) a také nepozorované (n-gramy neobsažené v trénovacím korpusu, ale v realitě jsou takové n-gramy přípustné). Aby toho bylo dosaženo je potřeba zmenšit

pravděpodobnost pozorovaných jevů tak, aby zbyla nenulová pravděpodobnost na jevy nepozorované. Goodův-Turingův odhad říká, vyskytuje-li se daný jev v celém trénovacím korpusu o velikosti N r -krát, potom opravená absolutní četnost tohoto jevu, kterou značíme r^* , je dána předpisem

$$r^* = \frac{(r + 1)n_{r+1}}{n_r}, \quad (3.10)$$

kde n_r je počet všech navzájem různých jevů (n -gramů) vyskytujících se právě r -krát v trénovacím korpusu. Odhad pravděpodobnosti jevu, který se v trénovacím korpusu vyskytl r -krát, je dán výrazem

$$P^{GT}(h, w) = \frac{r^*}{N}, \quad (3.11)$$

kde r^* můžeme určit z rovnice (3.10). Good-Turing odhad pravděpodobnosti všech nepozorovaných jevů je dán relativní četností singletonů (tj. jevů, které se vyskytly v trénovacím korpusu jenom jednou) a můžeme ho vyjádřit vztahem

$$P^{GT}(h, w) = \frac{n_1}{N} \quad \text{pro } N(h, w) = 0. \quad (3.12)$$

3.4.3 Written-Bell vyhlazování

Written-Bell model [11] je popsán diskontním součinitelem ve tvaru:

$$d_{N(h,w)} = d_h = \frac{N(h)}{N(h) + n(h)} \quad \text{pro } N(h, w) > 0, \quad (3.13)$$

kde $n(h)$ je počet různých slov následujících v trénovacím korpusu historii h a d_h značí, že diskontní součinitel je závislý pouze na historii h , tzn. pro každou pevnou historii je konstantní. Written-Bell model můžeme vyjádřit pomocí vztahu 3.6 jako

$$P_{WB}(w|h) = \begin{cases} \frac{N(h,w)}{N(h)+n(h)} & \text{pro } N(h, w) > 0 \\ \frac{n(h)}{N(h)+n(h)} \frac{\beta(w|\bar{h})}{\sum_{w': N(h,w)=0} \beta(w'|\bar{h})} & \text{pro } N(h, w) = 0. \end{cases} \quad (3.14)$$

3.4.4 Kneser-Ney vyhlazování

Kneser a Ney [11] odvodili nový vztah zobecněného rozdělení $\beta(w|\bar{h})$ a to

$$\beta(w|\bar{h}) = \beta_+(w|\bar{h}) = \frac{N_+(\bar{h}, w)}{N_+(\bar{h})}, \quad (3.15)$$

kde

$$N_+(\bar{h}, w) = \sum_{h: \bar{h} \subset h \wedge N(h,w) > 0} 1 \quad \text{a} \quad N_+(\bar{h}) = \sum_{w'} N_+(\bar{h}, w'). \quad (3.16)$$

Hodnota $N_+(\bar{h}, w)$ udává počet všech různých n -gramů hw (délky n) v celém korpusu, které končí posloupností slov hw (délky $n - 1$). Kneser-Ney model tedy nebere v úvahu pouze četnosti výskytu n -gramů, ale hlavně záleží na kontextu výskytu daného slova. Slova, která se objevují ve více kontextech, dostávají přiřazenou větší pravděpodobnost vyskytnutí se v neznámém kontextu, než-li slova, která se vyskytují pouze v určitém kontextu.

Například pokud bychom vzali v úvahu slovní spojení *Vysoké Mýto* a *dobrý fotbal*. Za předpokladu, že v trénovacím korpusu se slovo *Mýto* objevovalo nejčastěji, ale vždy jen ve spojení *Vysoké Mýto* a slovo *fotbal* se objevovalo méně často, ale v mnoha různých kontextech (např. krásný, ligový, útočný...). Potom pokud bychom měli odhadnout následující neznámé slovo například ve větě

Dnes se chci dívat na ...

Tak pokud by pravděpodobnost závisela pouze na relativních četnostech, mělo by největší pravděpodobnost výskytu slovo *Mýto*, ale jelikož u Kneser-Ney modelu záleží na kontextu ve kterém se slovo vyskytuje. Dostane největší pravděpodobnost slovo *fotbal*, protože se vyskytovalo v nejvíce různých kontextech, kdežto slovo *Mýto* jen v jednom. Do výsledné věty se doplní slovo s nejvyšší pravděpodobností a výsledek bude

Dnes se chci dívat na fotbal

3.5 Posouzení kvality jazykových modelů

Jazykový model můžeme vytvořit mnoha různými způsoby. Abychom byli schopni porovnat dva různé jazykové modely, potřebujeme mít k dispozici nějakou metodu, pomocí níž můžeme změřit kvalitu daného modelu.

Jazykový model lze porovnávat odděleně od ostatních částí aplikace, ve které je využíván. Například ohodnocením, jak umí předpovídat slova neznámého textu na základě textu, na kterém byl natrénován. Nejpoužívanější mírou ohodnocení kvality jazykového modelu je tzv. perplexita.

3.5.1 Perplexita

Perplexita [11] značí schopnost jazykového modelu, jak dobře dokáže popsat korpus, na kterém je perplexita počítána. Vychází z odhadu apriorní pravděpodobnosti $P(W) = P(w_1 w_2 \dots w_K)$ výskytu posloupnosti slov W . Pokud uvažujeme korpus vyjádřený jako posloupnost W čítající K slov včetně symbolů značících konce jednotlivých vět, je v průměru pravděpodobnost posloupnosti K slov K -krát menší než pravděpodobnost posloupnosti o délce jednoho slova. $P(W)$ je vhodné normalizovat vzhledem k počtu slov K funkcí příslušné odmocniny

$$\sqrt[K]{P(w_1 w_2 \dots w_K)}. \quad (3.17)$$

Díky normalizaci můžeme porovnávat kvalitu jazykových modelů na korpusech různé délky, nebo porovnávat rozpoznání dvou nestejně dlouhých korpusů stejným jazykovým modelem. Jelikož hodnota odhadu pravděpodobnosti $P(W)$ závisí jak na kvalitě modelu, tak i na míře neuspořádanosti samotného jazyka, definujeme perplexitu korpusu PP vztahem

$$PP = \frac{1}{\sqrt[K]{P(w_1 w_2 \dots w_K)}}. \quad (3.18)$$

Někdy se můžeme setkat s perplexitou vyjádřenou v logaritmické podobě označovanou jako LP

$$LP = \log_2 PP = \frac{-1}{K} \log_2 P(W). \quad (3.19)$$

Pro n-gramové modely můžeme vztah (3.19) upravit do tvaru

$$LP = \log_2 PP = \frac{-1}{K} \sum_{i=1}^K \log_2 P(w_i | w_1 w_2 \dots w_{i-2} w_{i-1}). \quad (3.20)$$

Perplexitu rozlišujeme podle druhu použitého korpusu na perplexitu testovacího korpusu nebo perplexitu trénovacího korpusu. Čím menší je hodnota perplexity, tím lépe jazykový model popisuje testovaný korpus.

3.6 Způsoby zrychlení jazykového modelu a redukce paměťové náročnosti modelu

Protože jednou z nevýhod n-gramového modelu popsanych v (3.2.2) je jeho paměťová a časová náročnost. Existují metody, které se snaží tuto náročnost snížit na přijatelnou míru.

3.6.1 Prořezání jazykového modelu (pruning)

Při tvorbě jazykového modelu můžeme využít jeho prořezání. Při něm vhodně nastavíme práh, podle kterého dojde k prořezání. Prořezáním jazykového modelu rozumíme odstranění takových n-gramů, které když odstraníme, tak následný vzrůst perplexity (3.5.1) nepřesáhne zadaný práh prořezání. V praxi se používají hodnoty prahu v rozmezí 1×10^{-9} až 1×10^{-7} . Hodnoty by neměly výrazně ovlivnit kvalitu modelu, neboť zhoršení perplexity mezi nimi je celkem zanedbatelné viz (7.6), ale rapidně sníží jeho velikost.

3.6.2 Nastavení četnosti n-gramů

Další metodou pro snížení paměťové náročnosti modelu je nastavení četnosti n-gramů. Můžeme nastavit minimální četnost n-gramů, které se mají do modelu zahrnovat. Například můžeme nastavit, že trigramy se musí vyskytnout v trénovacím korpusu minimálně 3x, bigramy minimálně 4x, aby byly do modelu zahrnuty.

3.6.3 Zmenšení velikosti slovníku

Z hodnot tabulky 3.1 je patrné, že velikost slovníku výrazně ovlivňuje jazykový model. Snížením počtu slov ve slovníku můžeme zredukovat velikost jazykového modelu.

Kapitola 4

Použité nástroje

V této kapitole jsou popsány všechny použité nástroje, které významě ovlivnily implementaci chatbota.

4.1 SRILM

SRILM [12] je toolkit, který slouží pro vytváření a práci se statistickými jazykovými modely. Především slouží pro použití při rozpoznávání řeči, statistického označování a segmentace a strojového překladu. Obsahuje jak programy pro vytváření a práci s jazykovými modely, tak i různé připravené skripty, které mohou řešit časté požadavky, při vytváření modelů, jako je například skript pro vytváření velkých jazykových modelů. Umožňuje využívat i jazykové modely třetích stran, jako jsou například modely od Googlu¹.

Pro tvorbu jazykového modelu se používá program `ngram-count`, který jako defaultní nastavení používá Good-Turing vyhlazování. Obsahuje různé množství parametrů² např.

- **-vocab file** - file určuje slovník, který bude použitý pro tvorbu jazykového modelu.
- **-text file** - file je trénovací korpus v textové podobě, případně může být zabalený ve formátu *tar.gz*.
- **-lm file** - file určuje název vytvořeného jazykového modelu.
- **-order N** - N značí řád jazykového modelu, defaultně je nastavený na 3 pro používání trigramového modelu.
- **-gtNmin K** - N určuje počet n-gram (2-bigram,3-trigram) a K minimální počet jeho výskytů v trénovacím korpusu aby byl zahrnut do jazykového modelu.
- **-prune h** - h je práh prožezávání.
- **-kndiscountN** - vyhlazování metodou Kneser-Ney, kde N můžeme vynechat nebo pomocí něj specifikovat, pro které n-gramy se tato metoda použije.
- **-wbdiscountN** - vyhlazování metodou Written-Bell, N má stejné vlastnosti jako v případě Kneser-Ney.

¹<http://commondatastorage.googleapis.com/books/syntactic-ngrams/index.html>

²<http://www.speech.sri.com/projects/srilm/manpages/ngram-count.1.html>

- **-interpolateN** - používá se k interpolaci, je použitelné pro kombinaci s metodami pro vyhlazování Kneser-Ney a Written-Bell.

Pro práci s jazykovým modelem se používá program `ngram`, který má také spoustu různých parametrů³ například:

- **-lm file** - file je jazykový model, který se má použít.
- **-gen N** - N určuje počet vět, které mají být z jazykového modelu vygenerovány.
- **-gen-prefixes file** - file označuje soubor, ve kterém se nachází prefixy pro generované věty. Každý prefix je na novém řádku, generovaný počet vět závisí na počtu řádků s prefixy v souboru.
- **-ppl textfile** - spočítá perplexitu modelu vzhledem k testovacímu korpusu, určeném souborem `textfile`.

4.2 Wordnik

Wordnik umožňuje práci s anglickými slovy a využívá k tomu různé volně šiřitelné slovníky. Má velice bohatou nabídku API⁴ v mnoha různých jazycích, pomocí nichž můžeme získávat užitečná data. Wordnik API umožňuje požádat o definice slova (určení slovních druhů), příklady vět, kontroly pravopisu, synonyma či antonyma, slovní spojení, ve kterém se určité slovo vyskytuje, dokončování slova, získání náhodného slova (může být specifikováno jaký slovní druhy požadujeme), získání slova dne a mnoho dalších.

³<http://www.speech.sri.com/projects/srilm/manpages/ngram.1.html>

⁴<http://developer.wordnik.com/docs.html>

Kapitola 5

Tvorba jazykového modelu

Jazykový model se vytváří z trénovacího korpusu (textu obsahujícího věty ve zvoleném jazyce) a slovníku. Pro testování kvality modelu např. perplexity (3.5.1) se využívá testovacího korpusu (textu obsahujícího věty ve zvoleném jazyce, které nebyly použity pro trénovací korpus ani nijak upravovány). Než jsem začal s tvorbou jazykového modelu, bylo nutné si výše uvedené korpusy a slovník vytvořit. K vytvoření korpusů jsem nejdříve musel nasbírat dostatečné množství odpovídajících dat.

5.1 Sběr dat

Sbíral jsem data pro trénovací korpus a testovací korpus, přičemž oba korpusy musí obsahovat data ze stejných zdrojů lišící se pouze obsahem. Pro získání dat k tvorbě jazykového modelu jsem využil různá internetová fóra, kde se nachází spousta konverzací na zajímavá témata například fórum¹. Dalším hlavním zdrojem pro získání dat byl Twitter, který poskytuje stream, na který uvolňuje 1% všech veřejných tweetů. Pro spouštění skriptů pro sběr dat jsem využil svého hostingu².

5.1.1 Sběr z Twitteru

Data z Twitteru jsem sbíral z již zmíněného streamu [5] prostřednictvím php api pro Twitter³. Za pomoci tohoto api jsem vytvořil jednoduchý skript, který se napojí na stream veřejných tweetů a získává z něj tweety v anglickém jazyce. Jelikož po připojení se na stream api neustále zachytává tweety a zpracovává je pomocí volání callbackové funkce, rozhodl jsem se omezit počet zachytávaných tweetů v jednom běhu na 5000. Číslo jsem zvolil pomocí pozorování doby běhu skriptu tak, aby doba běhu nepřekročila 30 sekund, jelikož touto dobou je limitována doba běhu php skriptu na hostingu. Skript jsem nechal spouštět cronem každou hodinu, a tak skript každý den nasbíral 120 000 tweetů.

Skript ukládá tweety do výstupního souboru, každý tweet na nový řádek. Původně ukládal také poznámky k jednotlivým tweetům (např. *tweet*, *retweet*, případně *originální tweet*, který je retweetován). Protože jsem se později rozhodl nerozlišovat mezi tweety a retweety, tyto poznámky jsem k ukládaným datům do konečné verze skriptu pro sběr dat z Twitteru nezahrnul. Výstupní soubory jsem si každý týden zálohoval a přejmenovával, aby skript nepracoval se zbytečně velkým souborem. Skript běžel několik měsíců a nakonec získal několik stovek MB dat.

¹fórum lze nalézt na adrese <http://www.bay12forums.com/smf/index.php>

²hosting se nachází na adrese <http://medievo.eu>

³jako api pro Twitter bylo použito <https://github.com/themathharris/tmhOAuthExamples>

5.1.2 Sběr dat z fór

Při sbírání dat z různých fór jsem nejdříve našel fóra obsahující zajímavé diskuze v anglickém jazyce. Zaměřil jsem se převážně na fóra, která umožňovala tisk stránky daného tématu. Tím získal k dispozici jednu stránku s kompletní diskuzí. Taková kritéria splňuje fórum zmíněné v sekci (5.1), které navíc obsahuje nepřehledné množství diskuzí k různým tématům, a proto se stalo mým hlavním zdrojem dat, ale ne jediným. Postup při získávání dat s z různých fór probíhal vždy ve stejných fázích:

1. Nalezení vhodného tématu s velkým počtem příspěvků.
2. Získání všech příspěvků v daném tématu na jedné stránce (tisk stránky).
3. Uložení této stránky.
4. Nastavení parametrů skriptu pro zpracování dat.
5. Uložení příslušných souborů do vstupní složky skriptu.
6. Spuštění skriptu.
7. Zálohování výstupů.

Skript pro zpracovávání dat je vytvořen v jazyku php a jeho funkcionalita je velice jednoduchá. Skript projde vstupní složku a zpracuje všechny soubory, které se v ní nacházejí, podle nastavených parametrů. Výsledky uloží do výstupních souborů pojmenovaných podle tématu příslušné diskuze. Ze vstupních dat jsou získány pouze příspěvky uživatelů v diskuzi a odstraněny citace předchozích příspěvků, čímž se eliminují duplicitní příspěvky v jednom tématu. Z fór se mi podařilo získat cca 18MB dat.

5.2 Proces vytváření jazykového modelu

Po získání dat jsem musel tyto data vhodně spojit, upravit a vytvořit z nich slovník a trénovací korpus pro vytvoření jazykového modelu. Z několika dat, která jsem nezahrnul do trénovacího korpusu, jsem vytvořil testovací korpus pro porovnání jazykových modelů.

5.2.1 Úprava a spojení dat

Data bylo nutné upravit tak, abych je mohl použít jako trénovací korpus pro n-gramový model. Důležité bylo vyřešit odkazy, interpunkci, závorky, smajlíky, různé nestandardní znaky, anglické zkracování výrazů s pomocným slovesem be a have (např. I am => I'm, I have => I've), uvozovky, apostrofy, mezery, prázdné řádky, duplicitní řádky, hashtagy, čísla, data a ještě pro každý zdroj dat různá specifická nežádoucí data.

Pro tyto úpravy jsem vytvořil jeden hlavní skript pro společné úpravy všech zdrojů dat. Pro každý zdroj dat jsem ještě naimplementoval vlastní skript, který řešil speciální případy nežádoucích dat právě pro konkrétní zdroj, převážně pro data z Twitteru.

5.2.2 Společný skript

Společný skript pro úpravu dat prakticky provádí tři mechanismy pro různé typy dat:

1. **Odstranění dat** - úplně odstraní nežádoucí data, např. multibajtové znaky. Odstraňují se i nežádoucí mezery - tato úprava se provede úplně nakonec, aby ve výstupním souboru nebylo více mezer za sebou. Duplicitní a prázdné řádky jsou odstraňovány též.
2. **Rozlišení znaků jako slova** - nahradí určitý znak jako mezera znak mezera, aby se tento znak zahrnul do jazykového modelu jako samostatné slovo a nespojoval se se slovem, před nebo za kterým byl napsán (např. slovní spojení *slovo, které nalezneme (zde)*. bude nahrazeno za *slovo, které nalezneme (zde)* .). Tento mechanismus byl aplikován na interpunkci ve větách, závorky, uvozovky a apostrofy.
3. **Úprava a nahrazování specifických slov** - tato část skriptu byla nejtěžší, protože jsem musel vymyslet jakým způsobem budu řešit odkazy, zkracování výrazů, smajlíky, čísla, datumy, čas.

Odkazy

Rozhodl jsem se, že odkazy v textu ponechám a zpracuji je tak, že si označím začátek a konec odkazu a odkaz vhodně upravím. Pro označení začátku a konce každého odkazu jsem zvolil značky `<<start_odkaz>>` pro začátek odkazu a `<<end_odkaz>>` pro konec odkazu. Odkaz, který se nacházel mezi těmito značkami jsem poté rozdělil na jednotlivé části odkazu podle znaků (`./:%&#=#`), takže pokud byl odkaz v textu ve tvaru `http://medievo.eu/chatbotapp`, po úpravě vzniklo `<<start_odkaz>> http : // medievo . eu / chatbotapp <<end_odkaz>>`. Pro takovýto postup u odkazů jsem se rozhodl, protože jazykový model se relativně dobře naučí posloupnost „slov“ mezi značkami začátku a konce odkazu a následně může vygenerovat přijatelný odkaz.

Zkracované výrazy

Zkracované anglické výrazy jsem se rozhodl rozgenerovat, aby se jazykový model neučil zvlášť zkrácené a zvlášť nezkrácené výrazy (např. *I am vs I'm*). Skript všechny anglické zkracované výrazy s pomocným slovesem *be* a *have* rozgeneruje na jejich dlouhé varianty. Rozgenerování proběhlo za pomoci definice pravidel, kdy je možné provádět zkracování⁴ v angličtině.

Ostatní věci jako jsou čísla, datumy, časy, smajlíci, hashtagy . . . jsem neupravoval a nechal jsem je jazykový model naučit v takovém tvaru, v jakém byly uvedeny v trénovacím korpusu.

5.2.3 Úprava twitterových dat

Data z Twitteru jsem musel upravit abych odstranil slova jako *tweet:*, *retweet:* a *original tweet:*, která jsem přidával do získaných dat ve své první verzi skriptu, jak bylo popsáno v sekci (5.1.1). Musel jsem odstranit ještě další nežádoucí slova jako *ret*, *rt*, *reretw*, také přezdívky uživatelů Twitteru začínající znakem @, protože tím se na Twitteru identifikuje, kdo a co komu napsal.

5.2.4 Vytvoření slovníku

Slovník jsem vytvářel již z upravených dat. Bylo nutné rozlišit slova zvlášť pro každý zdroj, takže jsem si vytvořil program, který získá 100 000 nejčastějších slov z textu a uloží je do souboru. Tímto postupem jsem získal nejčastější slova pro Twitter a fóra. Takovéto dělení bylo důležité, protože data z Twitteru výrazně převyšovala data získaná z fór, a tak byla velká šance, že by slova z Twitteru mohla zastínit některá důležitá slova z fór. Pro spojení nejčastější slov jsem vytvořil program, který projde všechna nejčastější slova z Twitteru a spojí se se slovy z fór tak, aby slova z fór nebyla

⁴definice zkrácených tvarů je zde <http://www.ajslovicka.cz/clanky/zkracene-tvary.html>

vytlačena slovy z Twitteru. Po spojení jsem slova omezil opět na 100 000, a tím jsem vytvořil slovník pro vytvoření jazykového modelu.

5.2.5 Vytvoření jazykového modelu

Po úpravách nasbíraných dat a vytvoření slovníku jsem všechna data spojil do jediného souboru, který jsem poté použil pro vytvoření jazykového modelu. Jazykový model jsem vytvořil pomocí toolkitu SRILM⁵ popsaného v sekci (4.1), konkrétně použitím programu `ngram-count`⁶, který jsem spouštěl s parametry

```
ngram-count -text <trenovaci_korpus> -lm <jazykovy_model>  
-vocab <slovník> -order 3 -gt3min 1 -prune <prah> -wbdiscout
```

Vytvořil jsem více jazykových modelů, ve kterých jsem vystřídal různé metody vyhlazování a vyzkoušel rozdílné prahy pro prořezání. Také jsem vytvořil jazykové modely bez podpory odkazů v době, kdy jsem ještě neměl dodělané zpracovávání odkazů při generování vět. Na základě srovnání (7.1.2) takto vytvořených jazykových modelů, jsem podle kritérií popsaných v sekci (7.1) zvolil pro chatbota ten nejvhodnější jazykový model. Který jsem poté použil pro generování vět.

⁵<http://www.speech.sri.com/projects/srilm/>

⁶<http://www.speech.sri.com/projects/srilm/manpages/ngram-count.1.html>

Kapitola 6

Implementace chatbota

V průběhu implementace jsem se rozhodl pojmenovat svého chatbota jménem Morrisek. V textu práce ho označuji jako chatbot a v uvedených příkladech komunikace je označován svým názvem Morrisek. Chatbota jsem naimplementoval jako klient-server aplikaci komunikací na principu TCP soketů. Server využívá nástroj SRILM pro práci s jazykovým modelem a generování jednotlivých vět. Klientovi se poté zasílá pouze počet vygenerovaných vět, o který server požádal. Pro generování vět jsem z toolkitu SRILM použil program `ngram` s parametry

```
ngram -lm <jazykovy_model> -gen <pocet_vet>
```

pro generování vět s určitým prefixem potom

```
ngram -lm <jazykovy_model> -gen-prefix <soubor_s_prefixy_vet>
```

kde `jazykovy_model` je soubor obsahující jazykový model a `soubor_s_prefixy_vet` je soubor obsahující prefix věty, ve kterém je každý prefix na jednom řádku a počet takovýchto řádků určuje počet generovaných vět. TCP soket komunikace mezi klientem a serverem probíhá pomocí protokolu ve tvaru:

$$\text{generate:N:kw} \tag{6.1}$$

kde `generate` značí, že má server klientovi zpět zaslat `N` vět, které splňují syntaktická pravidla, vygenerovaných z jazykového modelu. Jazykový model je na serveru pevně nastaven a `kw` je klíčové slovo, na které jsou generovány věty.

6.1 Server

Server je naimplementován v jazyce `c++11` a momentálně je spuštěn na školním serveru Merlin¹, kde naslouchá na portu 15000. Je založený na TCP soket komunikaci a využívá vlákna pro zpracování jednotlivých požadavků. Používá mnou vytvořenou třídu `filter`, která zpracovává věty vygenerované jazykovým modelem. Jejím účelem je najít mezi vygenerovanými větami takové věty, které

¹<http://merlin.fit.vutbr.cz>

odpovídají správné syntaxi a neobsahují zakázaná slova² (rasismusy a vulgarismy). Takto vybrané věty poté upraví do původní podoby podle provedených úprav popsanych v sekci (5.2.1).

Při syntaktické kontrole věty se kontrolují odkazy tak, že věta musí obsahovat značky `<<start_odkaz>>` `<<end_odkaz>>` a nebo ani jednu z nich. Pokud věta obsahuje obě značky začátku a konce odkazu, tak se zpracuje řetězec mezi těmito značkami a vytvoří se z něj opět odkaz za pomoci odstranění mezer. Dále se odstraňují mezery před interpukčními znaménky a zkontroluje se, jestli věta neobsahuje pouze hashtag či se nevyskytuje špatná kombinace po sobě jdoucích znaků, jakou je např. (*slovo*, *'libovolný_znak*). Probíhá kontrola párových znaků jako jsou např. uvozovky, které se ve větě nesmí vyskytnout osamotě.

Server rozlišuje podle zaslání požadavku, má-li generovat náhodné věty nebo věty začínající určitým klíčovým slovem (prefixem). V obou případech se generuje 75 vět, ze kterých jsou poté pomocí třídy `filter` získány výsledné věty. Při generování vět na základě klíčového slova je nejprve vytvořen soubor s prefixy těchto vět, který je po úspěšném vygenerování vět smazán. Výsledná věta, případně věty, jsou nakonec odeslány jako odpověď klientovi.

Server se přeloží a spustí pomocí následujících příkazů

```
make
./server -p <cislo_portu>
```

kde `cislo_portu`, je číslo portu, na kterém bude server naslouchat. Stejně číslo portu je nutné nastavit v konfiguraci klienta.

6.2 Klient

Klient je napsán v jazyce php a funguje jako api, které se dá dále rozvíjet a používat. Tvoří část chatbota, která komunikuje s uživatelem. Jeho hlavním účelem je zpracovat text, který do konverzace zaslal uživatel, uložit si důležitá data z tohoto textu a vrátit uživateli co možná nejlepší odpověď. Pokud klient potřebuje vygenerovat větu, zašle příslušnou žádost ve formátu (6.1) a přijme odpověď s vygenerovanou větou od serveru.

Klient využívá databázi mysql pro ukládání konverzací a dalších pro něj důležitých informací. Tuto databázi také využívá při používání různých technik pro zdokonalení konverzace. Klient používá i `wordnik`³ (4.2), který je zde využit pro získání slovních druhů u slov ze vstupního textu a rovněž pro generování náhodného prefixu věty.

Klient byl vytvářen a testován v několika verzích z důvodu možnosti srovnání chatbota, založeného čistě na jazykovém modelování, a chatbota, který využívá i další techniky pro komunikace s uživatelem. Jednotlivé verze se lišily většinou přidáním nové funkcionality, která řešila hlavní nedostatky předcházející verze. Techniky použité pro vylepšování chatbota vychází z technik popsanych v sekci (2.8).

6.2.1 Základní verze chatbota

Základní verze chatbota je založena čistě na jazykovém modelování. Nebere v úvahu kontext konverzace. Na uživatelské věty nahlíží pouze jako na pokyn k zaslání žádosti pro vygenerování věty na server. Nekontroluje ani uživatelův vstup na sprostá slova či rasismus. Jedinou další funkcionalitou, kterou tato verze disponuje, je schopnost detekovat nefunkčnost spojení se serverem. Při problému

²byla použita zakázaná slova z <https://gist.github.com/jamiew/1112488>

³<https://wordnik.com/>

ve spojení se serverem chatbot zareaguje větou

Sorry, Morrisek is away from keyboard. . .

Díky ignoraci kontextu konverzace nedávaly rozhovory s chatbotem smysl, např.

Uživatel: Hello

Morrisek: it is not the toilets overflowing with an 1982 — ref: d (

nebo

Uživatel: who is demi lovato?

Morrisek: everybody needs somebody sometime when things get to 11ammustafa consolation.

Ačkoli měl chatbot i světlé chvíle, kdy dokázal docela obstojně zareagovat na uživatelův vstup

Uživatel: what are you doing now?

Morrisek: i stayed up to school!

Uživatel: ahoj

Morrisek: go away.

hlavním nedostatkem, který bylo potřeba vyřešit, byla naprostá ignorace kontextu konverzace.

6.2.2 Druhá verze chatbota

Ve své druhé verzi se chatbot snaží zmírnit nedostatek základní verze pomocí wordniku, díky němuž rozeznává slovní druhy. Zjištěním slovních druhů je schopen určit předmět věty. Především díky pevně danému slovosledu anglické věty⁴, ve které vždy platí, že předmět se nachází za přísudkem. Pomocí této techniky chatbot už jenom tak bezmyšlenkovitě nezasílá požadavky serveru na generování náhodných vět. Nalezne-li předmět věty, odešle žádost o generování věty založené právě na tomto předmětu. Pokud ovšem věta předmět neobsahuje, nebo není předmět nalezen, přebírá chování základní verze a nechá si vygenerovat od serveru náhodnou větu. V této verzi je i přidána kontrola vulgárních a rasistických slov ze strany uživatele. Jestliže je uživatel na chatbota hrubý, chatbot zareaguje na jeho větu obsahující zakázané slovo větou

Don't be vulgar.

Vybraný příklad zlepšení díky vygenerované větě na základě prefixu, který tvoří předmět uživatelské otázky, je uveden zde

Uživatel: Do you like cookies?

Morrisek: cookies with chocolate milk is doing. might be

Nedostatek původní verze byl částečně odstraněn, ale stále se může objevovat, v případě nenalezení předmětu věty. Také zde není žádná iniciativa při konverzaci ze strany chatbota.

⁴http://cs.wikipedia.org/wiki/Slovosled_anglick%C3%A9_v%C4%9Bty

6.2.3 Třetí verze chatbota

Jelikož největší nedostatek předchozích dvou verzí byla časová náročnost generování vět, při každém vstupu od uživatele, rozhodl jsem se pokusit využít v další verzi nějakou vhodnou techniku pro konverzaci, která by omezila generování vět. Inspiroval jsem se technikou, kterou využívá Cleverbot (2.7). Tato verze obsahuje mimo mechanismů převzatých z předchozích verzí i nové mechanismy pro

- detekci začátku a konce konverzace,
- získání otázky z uživatelského vstupu,
- ukládání uživatelských otázek,
- získávání odpovědí na otázky,
- využívání odpovědí od uživatelů v konverzaci,

Detekce začátku a konce konverzace

Detekování začátku konverzace se provádí buď na základě počtu příspěvků konverzace, které byly přijmuty od uživatele (jedná-li se o první příspěvek, je to bráno jako začátek konverzace), nebo se porovnává začátek věty se slovy⁵, kterými jsou anglické pozdravy, a většinou se nimi uvozuje konverzace. Detekce konce konverzace probíhá obdobně opět pomocí nalezení slova, které konverzaci ukončuje.

Získání a ukládání otázky

Otázka se z uživatelského vstupu získává na základě interpunkčního znaménka, kterým by měla správná otázka končit, čili otazníkem (?). Aby nedošlo k zaměnění věty, která může být pronesena před otázkou, nejprve se rozdělí vstup na jednotlivé věty, ze kterých je vybrána pouze ta, která je zakončena otazníkem. Takto získaná otázka se uloží do databáze (pokud v ní už neexistuje), a to do tabulky určené pro otázky, které byly chatbotovi položeny. Každá otázka v tabulce má své jedinečné id, díky kterému je spojena s tabulkou určenou pro odpovědi k uloženým otázkám.

Získání a využívání odpovědí na otázky

Chatbot díky detekci začátku konverzace položí v úvodu konverzace uživateli jednu z otázek, které mu byly položeny. Otázky může vybírat podle dvou kritérií. Jednak na základě klíčového slova, které získá z uživatelského vstupu, nebo si zjistí otázku, na kterou má nejmenší počet odpovědí. Je-li takovýchto otázek více, náhodně vybere jednu z nich. Po položení otázky si poznačí id otázky, na kterou se uživatele zeptal, a jakmile uživatel odpoví, uloží si jeho odpověď do databáze k odpovědím na zvolenou otázku.

Takto získané odpovědi od uživatelů poté chatbot sám využívá. Pokud se ho uživatel zeptá na nějakou otázku, pokusí se najít tuto otázku ve své databázi, a pokud ji nalezne a existuje na ni odpověď, tuto odpověď použije.

⁵<http://www.helpforenglish.cz/article/2007082001-anglicke-pozdravy>

Ukázka pokládání otázek v konverzaci a užívání odpovědí

Uživatel: Hi

Morrisek: how are you?

Uživatel: fine.

V jiné konverzaci potom použije chatbot odpověď uživatele

Uživatel: how are you?

Morrisek: fine.

Nevýhodou této verze je poměrně malý okruh známých otázek a hlavně neschopnost určit, které otázky byly již v konverzaci použity. Může tak nastat situace, kdy se chatbot zeptá uživatele vícekrát na to stejné. Také kvalita otázek a odpovědí nemusí být přijatelná, proto je nutné ručně filtrovat otázky v databázi a odstraňovat ty nežádoucí, dokud nebude dostatek dobrých dat.

6.3 Sociální sítě

Jedním z hlavních cílů chatbota byla komunikace na sociálních sítích. Pro integraci chatbota na sociální sítě jsem se rozhodl využít Twitteru.

6.3.1 Twitter

Na Twitter jsem chatbota nasadil jako uživatele BotMorrisek⁶, kterému jsem vytvořil aplikaci, prostřednictvím které komunikuje s Twitterem [6]. Aplikaci pro Twitter jsem podle funkcionality rozdělil na dvě části. Na část, která umí postovat nové tweety na svůj profil, a část komunikující s uživateli pomocí tweetů. Pro komunikaci s Twitterem je použito php api zmíněné v sekci (5.1.1). Popsané časy spuštění se nemusí vždy úplně shodovat s realitou, protože dochází k občasných problémům při navazování spojení mezi api a Twitterem.

Postování tweetů

Pro vytváření nových tweetů jsem využil funkcionalitu druhé verze chatbota (6.2.2). Aplikace se v 70% případů pokusí⁷ vygenerovat větu na klíčové slovo, které je náhodně generováno pomocí wordniku (4.2), a v 30% případů vygeneruje náhodnou větu. Vygenerovanou větu poté zašle na svůj Twitter jako nový status. Aplikace je spuštěna jednou za 4 hodiny, a to v době od 8 hodin ráno do 23 hodin večer. Při každém spuštění se na Twitter zašle jeden nový status.

Komunikace s uživateli

Ke komunikaci s uživateli jsem využil funkcionalitu třetí (6.2.3) verze chatbota. Aplikace komunikuje s uživateli prostřednictvím tweetů adresovaných konkrétnímu uživateli. Aplikace využívá mysql databázi na ukládání čísel tweetů, které byly chatbotovi zaslány od uživatelů a na které chatbot již odpověděl. Díky této databázi si chatbot zjistí nový tweet, který mu byl zaslán (musí být novější než poslední zodpovězený), vygeneruje odpověď na tento tweet a zašle ji zpět odesílateli tweetu. Po úspěšném odeslání tweetu si poznačí v databázi jeho id jako zodpovězené. Tato aplikace

⁶<https://twitter.com/BotMorrisek>

⁷Pokud se nepodaří na klíčové slovo vygenerovat větu, použije se náhodně vygenerovaná věta.

je spouštěna každou hodinu, protože můj hosting neumožňuje častější spouštění skriptů pomocí cronu.

Kapitola 7

Experimenty

Kapitola se věnuje hodnocení různých verzí chatbota a jazykových modelů a jejich porovnávání. Kritéria pro hodnocení a srovnávání byla vybírána na základě cíle aplikace a jejich potřeb.

7.1 Experimentování s jazykovými modely

Jelikož chatbot bude využívat jazykový model při komunikaci s uživatelem pro generování vět, bylo mým cílem vybrat takový jazykový model, který by nejlépe splňoval kombinaci následujících kritérií:

- **Rychlost generování vět.**
- **Malá paměťová náročnost.**
- **Co nejlepší testovací perplexita.**

Z popisu n-gramového jazykového modelu (3.2) a jeho vlastností je patrné, že některá kritéria jsou vzájemně protichůdná¹, proto je zapotřebí zvolit mezi nimi ten nejlepší kompromis. Protože jsem při tvorbě jazykového modelu měl k dispozici dva trénovací korpusy (bez podpory a s podporou odkazů), rozhodl jsem se je oba využít při porovnávání jazykových modelů a pro každý najít tu nejvhodnější variantu. Jazykové modely jsem se rozhodl porovnat na základě metody pro vyhlazování. K porovnání jsem zvolil metody Good-Turing, Written-Bell a Kneser-Ney. Pro každou metodu vyhlazování jsem poté vytvořil několik modelů s různým prahem prořezání. Takto vytvořené modely jsem srovnal na základě kritérií (7.1). Pro zajímavost jsem rozlišoval testovací a trénovací perplexitu. Srovnání jednotlivých modelů je uvedeno v následujících tabulkách, ve kterých:

- **Práh** - značí práh prořezání.
- **MB** - je velikost jazykového modelu v MB.
- **PPL Tr** - je trénovací perplexita.
- **PPL Ts** - je testovací perplexita.
- **Čas** - je čas potřebný k vygenerování 50-ti vět.

¹Při snižování velikosti modelu, se zvyšuje perplexita

7.1.1 Modely bez podpory odkazů

Práh	MB	PPL Tr	PPL Ts	Čas [s]
1×10^{-5}	2,6	224	231	5,3
1×10^{-6}	11,0	143	250	8,9
1×10^{-7}	72,6	88	213	10,4
1×10^{-8}	513,7	40	220	53,1
1×10^{-9}	668,3	34	221	84,2
bez prořezání	722,5	34	221	87,4

Tabulka 7.1: Tabulka hodnocení jazykového modelu bez odkazů s Good-Turing vyhlazováním

Práh	MB	PPL Tr	PPL Ts	Čas [s]
1×10^{-5}	2,4	436	570	7,8
1×10^{-6}	10,5	193	327	9,0
1×10^{-7}	82,6	96	247	16,1
1×10^{-8}	384,9	51	226	28,7
1×10^{-9}	598,6	41	224	42,9
bez prořezání	718,0	39	224	54,6

Tabulka 7.2: Tabulka hodnocení jazykového modelu bez odkazů s Kneser-Ney vyhlazováním

Práh	MB	PPL Tr	PPL Ts	Čas [s]
1×10^{-5}	2,4	257	340	6,3
1×10^{-6}	8,5	152	254	6,6
1×10^{-7}	69,9	85	208	10,0
1×10^{-8}	534,4	35	204	36,5
1×10^{-9}	673,4	32	204	45,6
bez prořezání	708,6	32	204	43,9

Tabulka 7.3: Tabulka hodnocení jazykového modelu bez odkazů s Written-Bell vyhlazováním

7.1.2 Modely podporující odkazy

Práh	MB	PPL Tr	PPL Ts	Čas [s]
1×10^{-5}	3,2	122	384	19,0
1×10^{-6}	10,4	75	279	18,3
1×10^{-7}	56,5	50	231	26,1
1×10^{-8}	462,6	24	224	48,1
1×10^{-9}	685,6	20	224	54,4
bez prořezání	746,2	20	224	52,6

Tabulka 7.4: Tabulka hodnocení jazykového modelu s odkazy s Good-Turing vyhlazováním

Práh	MB	PPL Tr	PPL Ts	Čas [s]
1×10^{-5}	3,1	211	562	21,8
1×10^{-6}	12,4	91	327	19,9
1×10^{-7}	93,8	51	249	21,2
1×10^{-8}	441,7	28	231	38,9
1×10^{-9}	677,5	23	229	50,8
bez prořezání	745,2	22	228	53,3

Tabulka 7.5: Tabulka hodnocení jazykového modelu s odkazy s Kneser-Ney vyhlazováním

Práh	MB	PPL Tr	PPL Ts	Čas [s]
1×10^{-5}	3,3	122	384	32,0
1×10^{-6}	11,2	76	279	21,5
1×10^{-7}	81,2	46	225	22,0
1×10^{-8}	596,5	20	231	49,6
1×10^{-9}	732,2	19	234	57,7
bez prořezání	738,6	19	234	50,7

Tabulka 7.6: Tabulka hodnocení jazykového modelu s odkazy s Written-Bell vyhlazováním

Z uvedených tabulek je zřejmé, že z hlediska testovací perplexity vychází nejlépe vyhlazovací metoda Written-Bell. Rychlost se při hodnotách prahu prořezání od 1×10^{-7} do 1×10^{-5} výrazně neliší. Peplexita se začíná ustalovat zhruba od hodnoty prahu prořezání 1×10^{-7} . Paměťová náročnost modelu rapidně stoupá mezi prahovými hodnotami prořezání 1×10^{-8} a 1×10^{-7} . Pro oba druhy modelů (bez a s podporou odkazů) vychází při uzvážení všech kritérií jako nejlepší model s vyhlazovací metodou Written-Bell a prahem prořezání 1×10^{-7} , který odpovídá třetímu řádku v tabulce (7.6).

7.2 Experimenty s chatbotem

Pro snadnější hodnocení jednotlivých verzí chatbota jsem vytvořil testovací aplikaci². V této aplikaci se dá přepínat mezi jednotlivými verzemi chatbota a je lépe vidět srovnání jejich chování. Pomocí této aplikace jsem hodnotil kvalitu různých verzí chatbota na základě konverzací a uživatelů, kteří vyplnili jednoduchý dotazník.

7.2.1 Hodnocení na základě konverzací

Pro hodnocení na základě konverzací jsem jako hodnotící kritérium zvolil následující sled vět:

hi
How are you?
do you like cookies?
bye.

Tyto věty byly vybrány záměrně, protože je možné na nich pozorovat pokroky jednotlivých verzí.

²Dostupné z <http://medievo.eu/chatbotapp/>

Základní verze chatbota

Uživatel: hi

Morrisek: transfer key for peace spiritual pigtales to go to the fans stay in love at first it is a matter of time when i can buy an urban meyer

Uživatel: how are you?

Morrisek: now. he is not a gals

Uživatel: do you like cookies?

Morrisek: i just want shocking #squaready #luxury misheard the crazy girlfriend advances prisoners in house when said baffled

Uživatel: bye

Morrisek: conversation about midnight memories whos in twitter.

Druhá verze chatbota

Uživatel: hi

Morrisek: what is this so i wanted to sleep

Uživatel: how are you?

Morrisek: you can do it

Uživatel: do you like cookies?

Morrisek: cookies with chocolate milk is doing. might be

Uživatel: bye

Morrisek: still my 1 /6:30 jam!! we getting more and coz they are usually able too ok thank you just can not u

Třetí verze chatbota

Uživatel: hi

Morrisek: what do you like?

Uživatel: i like cookies

Morrisek: Do you like cookies?

Uživatel: yes. How are you?

Morrisek: fine.

Uživatel: do you like cookies?

Morrisek: yes I love it

Uživatel: bye

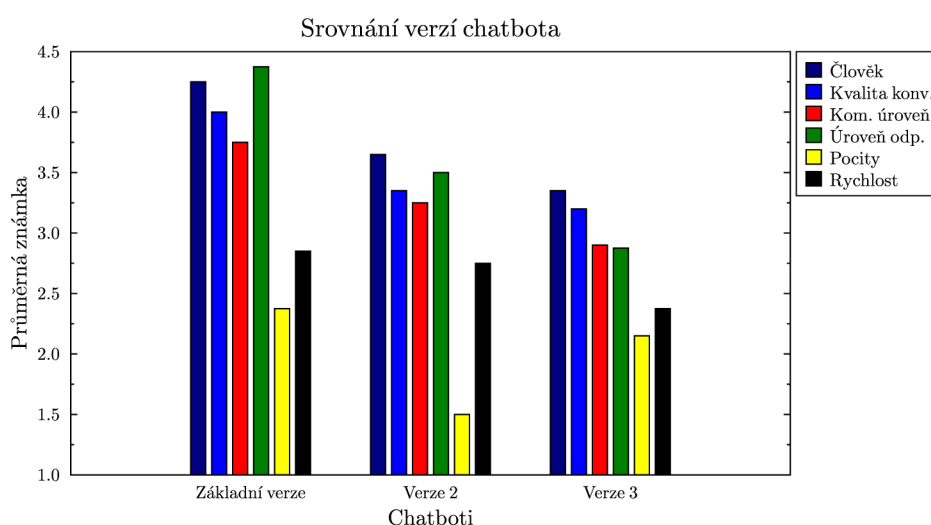
Morrisek: bye

Z uvedených konverzací lze vyčíst, že základní verze chatbota nedává uspokojivé výsledky. Dalo se to očekávat, neboť vůbec nebere v úvahu kontext konverzace. Ve druhé verzi lze pozorovat mírné zlepšení zejména v případě nalezení vhodného klíčového slova, které je použito jako prefix věty. Problém porozumění kontextu konverzace zde však stále přetrvává - ne vždy se podaří naleznout vhodné klíčové slovo a vygenerovat k němu smysluplnou větu. Třetí verze chatbota už přináší větší pokrok v plynulosti a smysluplnosti konverzace. Jednoduchým využitím otázek a odpovědí od jiných uživatelů, dokáže chatbot celkem obstojně komunikovat. Ale i třetí verze má pár nedostatků. Pokud je chatbotovi položena neznámá otázka, reaguje stejně jako jeho předchozí verze. Dalším problémem této verze je opakování položených otázek ve stejné konverzaci, což je způsobeno malou databází známých otázek a také generováním otázek na klíčové slovo (např. pokud na chatbotovu otázku odpoví uživatel užitím předmětu otázky, může se otázka položit znovu).

7.2.2 Hodnocení na základě uživatelů

Při hodnocení verzí chatbota pomocí uživatelů jsem nechtěl provádět hodnocení pouze na základě Turingova testu (2.1), protože mým cílem nebylo, aby si uživatel myslel, že komunikuje s dalším člověkem, ale aby chatbot dokázal vést s uživatelem relativně smysluplnou konverzaci. Rozhodl jsem se porovnávat jednotlivé verze chatbota mezi sebou a výslednou nejlepší verzi potom porovnat s jinými chatboty, konkrétně s Cleverbodem (2.7) a Elizou (2.3). Uživatele jsem nechal pro tyto chatboty a jednotlivé verze mého chatbota vyplnit dotazník³, vytvořený pomocí Google Docs a skládající se z různých otázek, které jsem ohodnotil známkami. Dotazník je uveden v příloze (C). Pro srovnání jsem poté využil průměrné známky z jednotlivých dotazů.

Srovnání jednotlivých verzí chatbota

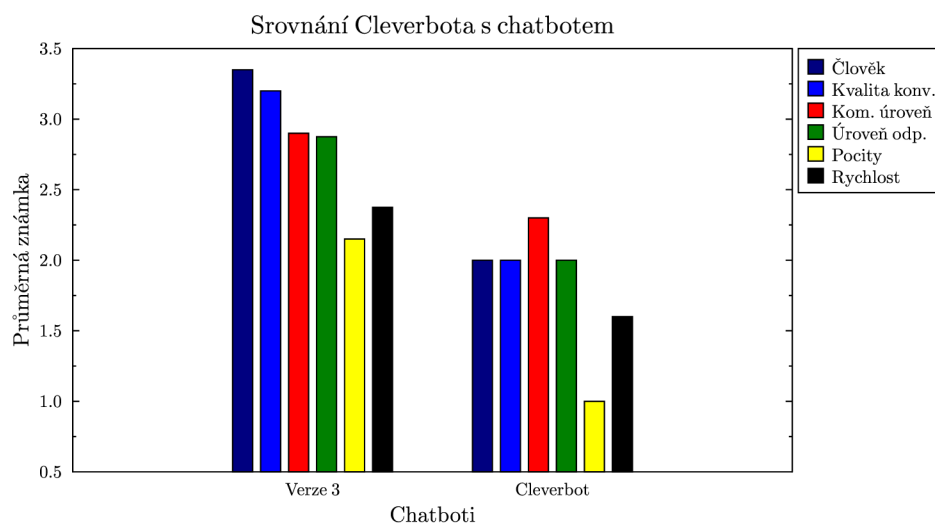


Obrázek 7.1: Graf srovnání jednotlivých verzí chatbota na základě průměrných známek z odpovědí na dotazník.

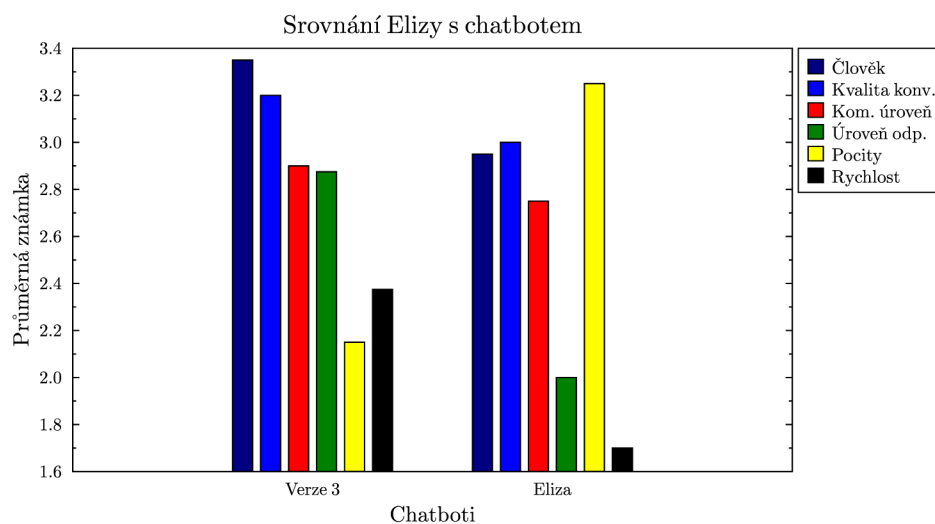
Z uvedeného srovnání jednotlivých verzí je patrné, že postupně přidávané funkcionality splnily svůj účel a opravdu dochází k mírnému zlepšení u jednotlivých verzí. Možná nás může zarazit, že třetí verze má o něco horší známku z pocitů konverzace. Ale tento údaj může být poněkud zkreslený, protože chatbot zatím disponuje pouze malou databází otázek a odpovědí. Otázky také vždy nemusí být vždy úplně správně. Není zaručeno, že otázka položená chatbotovi od jiného uživatele se dá s jistotou použít v konverzaci s jiným uživatelem. Takovéto otázky byly postupně ručně z databáze odmazávány.

³Dostupný z https://docs.google.com/forms/d/1ssa9qg07VRAOMw4Mb0Q1HzvluqST-IjrlUU_2z0ktw/viewform

Srovnání s jinými chatboty



Obrázek 7.2: Graf srovnání chatbota s Cleverbodem na základě průměrných známek z odpovědí na dotazník.



Obrázek 7.3: Graf srovnání chatbota s Elizou na základě průměrných známek z odpovědí na dotazník.

U srovnání chatbota s Cleverbodem je patrný rozdíl v kvalitě, kdy Cleverbot jednoznačně mého chatbota překonává ve všech směrech. Je to převážně dáno tím, že Cleverbot oproti mému chatbotu disponuje databází s několika miliony konverzací. Při srovnání chatbota s Elizou je patré, že již nenástává nijak zásadní rozdíl mezi oběma chatboty. Eliza vychází lépe ve většině kritérií, ale rozdíl

je většinou poměrně malý. Je možné, že kdyby měl chatbot k dispozici větší a kvalitnější databázi otázek, mohl by se s Elizou vyrovnat, případně ji i překonat.

7.2.3 Experimenty na Twitteru

Na Twitter byl nasazený chatbot, který využívá funkcionalitu všech verzí chatbota. Z tohoto důvodu jsem jeho porovnávání v rámci kvality konverzace či hodnocení uživatelů pomocí dotazníků už znovu nerealizoval. Sledoval jsem pouze jeho počínání na sociální síti a případné reakce uživatelů. Chatbotovi se podařilo získat i malé množství sledujících uživatelů, jedná se především o uživatele, kterým jsem chatbotovu twitterovou stránku zaslal. Ovšem chatbot získal přízeň i u amerického uživatele Twitteru z Ohia, kterému se líbil jeden z jeho tweetů.

Kapitola 8

Závěr

Cílem práce bylo zjistit, jak lze využít jazykové modelování v aplikaci chatbota, který komunikuje s uživateli na sociálních sítích, a srovnání chatbota s dalšími existujícími chatboty.

Úvodní část práce objasňuje problematiku chatbotů a jazykového modelování. Hlavní část práce se zaměřuje na vytváření jazykového modelu a jeho následného použití při tvorbě chatbota, což spočívalo v nasbírání potřebných dat pro vytvoření trénovacího a testovacího korpusu, na jehož základě byl vytvořen jazykový model. Ten byl následně využit při komunikaci chatbota s uživateli. Bylo vytvořeno také několik verzí chatbota lišících se použitím jazykového modelování pro komunikaci s uživatelem. Pro účely testování bylo naimplementováno testovací rozhraní, které umožňuje přepínání mezi jednotlivými verzemi chatbota, díky němuž lze pozorovat odlišné chování jednotlivých verzí. Jednotlivé verze byly porovnány na základě konverzací a hodnocení uživatelů pomocí stanovených kritérií.

Jako nejlepší jazykový model pro tvorbu chatbota vyšel model s vyhlazovací metodou Written-Bell a prahem prořezání 1×10^{-7} . Chatbot založený čistě na jazykovém modelování nedosáhl kvalitních výsledků, protože vůbec nebral v úvahu kontext konverzace. Částečné zlepšení přineslo generování vět na klíčové slovo získané z uživatelského vstupu, ale i tak nebyla vždy zaručena obstojná kvalita konverzace. Podle výsledků experimentů je jazykové modelování vhodnější na generování statusů pro sociální sítě, než pro generování vět ke konverzaci. Pro zlepšení konverzačních schopností chatbota se osvědčilo využívání technik uvedených v sekci (2.8). Chatbot díky nim dokáže konverzovat s uživatelem na přijatelné úrovni. Třetí verze chatbota, která tuto techniku obsahuje, byla vyhodnocena jako nejúspěšnější a verze byla použita pro integraci na sociální síť Twitter.

Bylo provedeno srovnání nejúspěšnější verze chatbota s jinými chatboty. Konkrétně s Cleverbotem a Elizou. V porovnání s Cleverbotem nedosáhl chatbot srovnatelných výsledků jako Cleverbot. Hlavní příčinou je rozdíl v databázi konverzací. Při srovnání s Elizou chatbot dopadl lépe a Elize se téměř vyrovnal.

V budoucí práci by bylo vhodné se zaměřit na lepší porozumění kontextu konverzace. Daly by se k tomu využít různé nástroje pro zpracování vět, aby chatbot porozuměl kontextu konverzace více, než jenom určením slovních druhů a získáním předmětu věty. Kombinace získání určitého pojmu o kontextu konverzace a jazykového modelování by mohla vést k poměrně dobrým výsledkům. Chatbot by díky využití jazykového modelování mohl relativně snadno generovat věty ve více jazycích a i rozpoznat, ve kterém jazyce s ním uživatel komunikuje. Mohl by se rozšířit i na další sociální sítě jako je např. Facebook. Na sociálních sítích by mohl dále probíhat jeho rozvoj v duchu simulování uživatele těchto sítí, kdy by mohl reagovat na různé žádosti od uživatelů (např. žádosti o přátelství).

Literatura

- [1] contributors, W.: Chatterbot [online].
<http://en.wikipedia.org/w/index.php?title=Chatterbot>, 2013-10-07 [cit. 2013-12-18].
- [2] contributors, W.: Turing test [online].
http://en.wikipedia.org/w/index.php?title=Turing_test, 2013-11-13 [cit. 2013-12-18].
- [3] contributors, W.: ELIZA [online].
<http://en.wikipedia.org/w/index.php?title=ELIZA>, 2013-11-20 [cit. 2013-12-18].
- [4] contributors, W.: Loebner Prize [online].
http://en.wikipedia.org/w/index.php?title=Loebner_Prize, 2013-12-02 [cit. 2014-01-21].
- [5] Developers, T.: The Streaming APIs [online].
<https://dev.twitter.com/docs/api/streaming>, 2012-09-24 [cit. 2014-01-21].
- [6] Developers, T.: Twitter REST API v1.1 [online].
<https://dev.twitter.com/docs/api/1.1>, 2014-01-14 [cit. 2014-05-11].
- [7] Houser, P.: Chatboty 100 let po narození Alana Turinga [online].
<http://computerworld.cz/technologie/chatboty-100-let-po-narozeni-alana-turinga-45124>, 2012-06-04 [cit. 2014-01-21].
- [8] Manning, H., Christopher D. Schütze: *Foundations of statistical natural language processing*. Cambridge, Massachusetts: The MIT Press, 2000, iISBN 0-262-13360-1.
- [9] Nejedlová, D.: *Creation of Lexicons and Language Models for Automatic Broadcast News Transcription*. Liberec: Technická univerzita v Liberci, 2004.
- [10] Pišl, V.: Mluvicí počítače - od Elizy k Suzette [online].
<http://vtm.e15.cz/mluvici-pocitace-od-elizy-k-suzette>, [cit. 2013-12-18].
- [11] Psutka, J.; Müller, L.; Matoušek, J.; aj.: *Mluvíme s počítačem česky*, ročník Vydání 1. Praha: Academia, 2006, iISBN 80-200-1309-1.
- [12] Technology, S. S.; Laboratory, R.: SRILM - The SRI Language Modeling Toolkit [online].
<http://www.speech.sri.com/projects/srilm/manpages/>, 2013-11-04 [cit. 2014-04-19].

Dodatek A

Obsah DVD

- data.tar.gz - archiv obsahující všechny vytvořené jazykové modely
- tvorba_modelu.tar.gz - archiv obsahující skripty a data pro vytvoření jazykového modelu
- xplaga00.pdf - text bakalářské práce
- xplaga00.tar.gz - archiv zdrojových kódů bakalářské práce
- zdrojove_soubory.tar.gz - archiv obsahující všechny zdrojové soubory chatbota
- mysql.tar.gz - dump databáze a potřebných tabulek

Dodatek B

Manual

B.1 Vytvoření jazykového modelu

- Rozbalení archivu tvorba_modelu.tar.gz.
- Rozbalení souborů forum.txt.tar.gz a twitter.txt.tar.gz
- ./uprav_texty
- ./vytvor_model

B.2 Spuštění testovací aplikace chatbota

- Rozbalení archivu zdrojove_soubory.tar.gz.
- Přeložení a spuštění serveru ve složce server.
- make
- ./server -p port_number
- Úprava konfigurace souboru chatbotlib.php ve složce klient. (nastavení adresy serveru a portu, nastavení databáze, nastavení wordniku)
- Uložení složky klient na web a spuštění souboru index.php

B.3 Spuštění chatbota na Twitteru

- Rozbalení archivu zdrojove_soubory.tar.gz.
- Přeložení a spuštění serveru ve složce server.
- make
- ./server -p port_number
- Úprava konfigurace souboru chatbotlib.php ve složce twitter_klient. (nastavení adresy serveru a portu, nastavení databáze, nastavení wordniku).
- Uložení složky twitter_klient na web.

- Vytvoření aplikace na Twitteru.
- Nastavení konfigurace twitrovské aplikace v souboru config.php ve složce twitter_klient.
- Nastavení spouštění skriptu BotMorrisek.php do cronu (umožní komunikaci prostřednictvím tweetů s uživateli).
- Nastavení spouštění skriptu BotMorrisek.php?generate do cronu (vygeneruje status).

Dodatek C

Dotazník

Působila na vás konverzace jakoby jste chatovali s člověkem?

1. Naprosto shodná s člověkem.
2. Velice se blížila konverzaci s člověkem.
3. Někdy ano někdy ne.
4. Spíše tak nepůsobila.
5. Vůbec tak nepůsobila.

Jak byste zhodnotili kvalitu konverzace?

1. Dokonalá.
2. Dobrá.
3. Ucházející.
4. Špatná.
5. Příšerná.

Jaké máte pocity z odpovědí?

1. Pobavily mě.
2. Občas mě pobavily.
3. Nudili mě.
4. Neoslovily mě.
5. Nepochopil/a jsem je.

Jak byste zhodnotili komunikační úroveň chatbota?

1. Dokonalá.
2. Dobrá.
3. Ucházející.

4. Špatná.
5. Příšerná.

Jaká byla úroveň odpovědí?

1. Naprosto smysluplné.
2. Většinou smysluplné.
3. Tak půl na půl.
4. Spíše nedávaly smysl.
5. Naprosté nesmysly.

Jak byste zhodnotili rychlost odpovědí?

1. Velmi rychlé.
2. Rychlé.
3. Dalo se to přežít.
4. Pomalé.
5. Příšerně pomalé.