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

RYCHLÁ ADAPTACE POČÍTAČOVÉ PODPORY HRY KRYCÍ JMÉNA PRO NOVÉ JAZYKY

FAST ADAPTATION OF CODENAMES COMPUTER ASSISTANT FOR NEW LANGUAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. PETR JAREŠ**
AUTHOR

VEDOUCÍ PRÁCE — **doc. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání diplomové práce

Student: **Jareš Petr, Bc.**
Program: Informační technologie Obor: Strojové učení
Název: **Rychlá adaptace počítačové podpory hry Krycí jména pro nové jazyky**
Fast Adaptation of Codenames Computer Assistant for New Languages
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s knihovnami pro analýzu základních jazykových rovin, např. lemmatizaci textu a odhad společných kořenů, podporujících velkou šíři přirozených jazyků.
2. Prostudujte moderní metody reprezentace významu pomocí neuronových sítí typu transformer a způsoby extrakce vektorových modelů pro slovní asociace.
3. Navrhněte a implementujte systém, který bude schopen existující modely automatické podpory hry Krycí jména rychle adaptovat pro dosud nepodporované jazyky.
4. Ověřte kvalitu vytvořeného systému pro vybraný jazyk ve spolupráci se zahraničními partnery projektu enectCollect.
5. Vytvořte stručný plakát prezentující vytvořenou práci a její výsledky.

Literatura:

- Dle doporučení vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Funkční prototyp adaptačního systému

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Smrž Pavel, doc. RNDr., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 30. října 2020

Abstrakt

Tato diplomová práce rozšiřuje herní systém umělého hráče slovně-asociační hry Krycí jména o snadné přídání podpory pro nové jazyky. Systém je schopný hrát Krycí jména v rolích hádajícího hráče, zadavatele nápověd a jejich kombinací hráče verze Duet. K analýze různých jazyků byl použit neurální nástroj Stanza, který je jazykově nezávislý a umožňujě automatizované zpracování celé řady jazyků. Jednalo se především o lemmatizaci slov a určování slovních druhů pro výběr kandidátních nápověd ve hře. Pro vyhodnocení slovních asociací byla testována řada modelů, kde nejlepších výsledků dosahovala metoda Pointwise Mutual Information a prediktivní model fastText. Systém podporuje hraní Krycích jmen v 36 jazycích tvořených 8 různými abecedami.

Abstract

This thesis extends a system of an artificial player of a word-association game Codenames to easy addition of support for new languages. The system is able to play Codenames in roles as a guessing player, a clue giver or, by their combination a Duet version player. For analysis of different languages a neural toolkit Stanza was used, which is language independent and enables automated processing of many languages. It was mainly about lemmatization and part of speech tagging for selection of clues in the game. For evaluation of word associations were several models tested, where the best results had a method Pointwise Mutual Information and predictive model fastText. The system supports playing Codenames in 36 languages comprising 8 different alphabets.

Klíčová slova

Krycí jména, vícejazyčná podpora, zpracování přirozeného jazyka, lemmatizace, umělá inteligence, Pointwise Mutual Information, word embeddings, fastText, Python

Keywords

Codenames, multilingual support, natural language processing, lemmatization, artificial intelligence, AI, Pointwise Mutual Information, word embeddings, fastText, Python

Citace

JAREŠ, Petr. *Rychlá adaptace počítačové podpory hry Krycí jména pro nové jazyky.* Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Pavel Smrž, Ph.D.

Rychlá adaptace počítačové podpory hry Krycí jména pro nové jazyky

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Petr Jareš
14. května 2021

Poděkování

Rád bych poděkoval svému vedoucímu panu doc. RNDr. Pavlovi Smržovi, Ph.D. za odborné vedení práce.

Obsah

Kapitola 1

Úvod

V dnešní době se čím dál více deskových her přesouvá do online prostoru a hra Krycí jména není výjimkou. Hraní online umožňuje lidem se spojit přes velké vzdálenosti a také korektní kontrolu pravidel ve sporných situacích. Od doby, kdy byla vydána hra Krycí jména, byla hra zpracována v celé řadě dalších jazyků a neustále vycházejí další. Vytvoření podpory umělého hráče ve slovně-asociační hře Krycí jména pro více jazyků nabízí nejen možnost doplnit neúplný tým hráčů či roli pomocníka začínajícím hráčům, ale také možnost jak vyhodnocovat různé metody pro určování podobnosti slov a sběr relevantních dat pro různé jazyky formou hry.

V rámci předkládané práce navazuji na svoji bakalářskou práci Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11]. V bakalářské práci byl vytvořen sytém, který umí zaujmout jak roli hádajícího hráče, tak i roli hráče vytvářejícího nápovědy ve standardní verzi Krycích jmen. Kombinací těchto rolí byl vytvořen i hráč pro kooperativní verzi hry Duet. Systém podporoval jazyky češtinu a angličtinu.

Cílem předkládané diplomové práce je rozšířit systém tak, aby bylo možné do něj co nejjednodušeji přidávat nové jazyky, ve kterých hra Krycí jména vyšla. Dále je cílem vyzkoušet novější metody pro vyhodnocování podobnosti slov a případně je nahradit za stávající metody, pokud budou úspěšnější. Také je kladen důraz na vylepšení webové služby v oblasti zpětné vazby uživatele a lepšího sběru dat. Webová služba[1] poskytuje možnost testování zmíněných rolí umělého hráče a možnost jejich případného zdokonalení.

Kapitola 2 obsahuje rozbor přirozeného jazyka a jeho vazby pro hru Krycí jména. Dále popisuje rozdíly mezi různými skupinami jazyků, které mohou mít vliv na úspěšnost zpracování podpory pro tyto jazyky. Kapitola 3 prezentuje nástroje pro předzpracování textu a popisuje jejich fundamentální části, ze kterých se skládají. Představené nástroje jsou díky svých charakteristikám jazykově nezávislé, což nabízí možnost značné automatizace při přidávání nových jazyků do systému. Nástroje se využijí pro konkrétní jazykově závislé části systému. V kapitole 4 je stručný přehled technik použitých v referenční práci pro vyhodnocování podobnosti slov a také rozbor nových, které by měly pokrývat „slabiny“ stávajících. Techniky uvedené v této kapitole jsou jazykově nezávislé a aplikují se na text zpracovaný pomocí nástrojů z předchozí kapitoly. Kapitola 5 nejprve shrnuje stávající stav systému a dále popisuje návrh jeho rozšíření. Jsou zde popsány kroky, které je potřeba udělat pro přidání podpory nového jazyka. Také jsou zde popsány vylepšení systému v oblasti herní strategie a snahy dosažení vyšší úspěšnosti ve vytváření slovních asociací především pro český jazyk. V kapitole 6 jsou popsány implementační detaily zmíněného systému a imple-

[1] http://athena3.fit.vutbr.cz:8086/

mentovaná automatizace pro přidávání nových jazyků. V kapitole 7 jsou uvedeny úspěšnosti jednotlivých metod pro vyhodnocení sémantické podobnosti slov v rámci použití u Krycích jmen. Tyto metody jsou vyhodnocené na datech z reálných her v českém jazyce. Dále jsou zde porovnány výsledky jednotlivých jazyků, které jsou zpracovány ze záznamů her webové služby.

Kapitola 2

Rozbor jazyka

Tato kapitola obsahuje rozbor přirozeného jazyka na několika rovinách. Dále je zde uvedeno několik způsobů klasifikace různých jazyků do charakteristických skupin. V poslední řadě je zde stručně uvedena desková hra Krycí jména a k ní jazykové vztahy, které je potřeba brát v potaz při strojovém zpracovaní.

Přirozený jazyk – jazyk, kterým mluví lidé – lze v rámci analýzy rozdělit do několika rovin. Každá tato rovina má definované své vlastní vstupy a výstupy. Rozbor jazyka směrem od základních prvků až po interpretaci jazykových celků v reálném světe lze rozdělit na tyto roviny [27]:

- fonetická – určuje tvorbu zvuků (fónů) ze vstupního signálu.
- fonologická – definuje hlásky (fonémy) a jejich význam v relaci s písmeny abecedy.
- pravopisná – zajišťuje jednoznačnou interpretaci vícevýznamových jednotek jazyka.
- morfologická – zabývá se skloňováním a časováním slov.
- syntaktická – definuje slovosled a vztah mezi slovy ve větě.
- sémantická – zabývá se významem výrazů napříč všemi úrovněmi jazyka.

2.1 Morfologie

Tato práce řeší problém určování morfémů (minimálních sémanticky nedělitelných jednotek) slov, konkrétně lemmat (slovníkových tvarů) a kořenů slov. Zmíněná problematika spadá pod morfologickou rovinu analýzy jazyka.

Morfologie je lingvistická disciplína zabývající se ohýbáním slov, tedy skloňováním a časováním. Elementární prvky morfologie jsou morfémy. Jedná se o nejmenší části slova, které nesou vlastní sémantický a gramatický význam. Zde je uveden výčet typů morfémů [1].

Předpony

Předpony (prefixy) jsou morfémy, které se vyskytují před kořenem slova a mění jeho lexikální význam. V češtině existují dva typy předpon – slovotvorné a tvarotvorné.

Běžnější slovotvorné, které se vyskytují i v jiných jazycích (např. v angličtině), slouží k přesnějšímu určení nebo modifikaci hlavního významu slova a mohou být rozšířeny i dalšími příponami. Slovotvorné předpony jsou například vy-skoč-i-t, po-vy-skoč-i-t, roz-cest-í.

Tvarotvorné předpony se používají pro vyjádření nedokonavého vidu některých sloves. Jsou tvořeny buď morfémy *po* nebo *pů*, například po-let-í-m, pů-jd-e-me.

Kořen

Kořen je morfém, který nese základní význam slova. Tento význam může být blíže specifikován připojením předpon z levé strany nebo přípon z pravé.

Přípony

Přípony se vyskytují v rámci slovního tvaru za kořenem. V češtině existují přípony kmenotvorné, slovotvorné a tvarotvorné. Kmenotvorné přípony se vyskytují ve všech slovesných tvarech a u podstatných jmen. Slouží ke spojování kořenového morfému s tvarotvornými příponami. Podoba kmenotvorné přípony klasifikuje slovo do prézentních nebo do infinitivních slovených tříd. Příklady Kmenotvorných tříd mohou být následující: nes-e, děl-á, kup-ova-l.

Slovotvorné přípony upravují lexikální význam slova například zdrobňováním slova. Ukázkou může slovo loď. Pro zdůraznění, že se jedná o malé plavidlo, se použije loď-k-a. Dalšími příklady slovotvorných přípon jsou květ-in-áč-0, hrad-b-a.

Tvarotvorné přípony vyjadřují gramatický význam slov. Dělí se na dva typy, nefinální tvarotvorné přípony a finální tvarotvorné přípony. Za nefinálními se nutně vyskytují finální přípony. Finálním tvarotvorné přípony se také označují jako koncovky. Koncovky jsou většinou úplným koncem slova.

Jediný morfém, který se po koncovce může vyskytnout, je postfix. Příklad postfixu může být po-jď-0-me-ž. Poslední nezmíněný morfém je interfix. Ten slouží ke kompozici dvou slov do jednoho slova, například ve slově hlad-o-mor-0.

Kmen

Kmen je taková část slova, která zůstane po oddělení koncovky. Kmen slova může být totožný s kořenem, nebo je tvořen kořenem a kmenotvornou příponou. Procesu, který systematicky odstraňuje koncovky slov se říká stemming.

Lemma

Lemma slova je jeho slovníková podoba v základním tvaru. Někdy také označováno jako heslové slovo. V češtině jsou například lemmata podstatných jmen nominativy a lemmata sloves jejich infinitivy. Jelikož slovo tvarované a časované do různých podob nese stejný sémantický význam, je vhodné pro některé úkoly v rámci zpracování přirozeného jazyka používat lemmata slov namísto slov samotných. Slova *počítala* a *počítají* vyjadřují stejnou činnost *počítat*. Procesu vytváření lemmat z původních slov se říká lemmatizace.

2.2 Klasifikace jazyků

Přirozené jazyky lze klasifikovat různými přístupy na základě jejich společných charakteristik. Genetická klasifikace vychází z myšlenky, že některé jazyky jsou si příbuzné a vyvíjely se ze společného prajazyka. Touto klasifikací lze členit jazyky do jazykových rodin, jako jsou třeba indoevropské, uralské, austroasijské... Jazykové rodiny lze potom členit dále na podskupiny. Například indoevropské jazyky se dělí na germánské, románské, slovanské... [29]

Genetická klasifikace však nepřináší moc informací o skladbě jazyka. Mezi nejrozšířenější klasifikaci patří typologická, která klasifikuje jazyky v několika jazykových rovinách. Jedná se o typologie fonologické, morfologické, syntaktické a strukturní.

Fonologická typologie klasifikuje jazyky podle tvorby a charakteristik slabik.

Morfologická typologie rozděluje jazyky na základě různých kvantifikovatelných poměrových rysů nazvaných indexy. Jedná se třeba o index syntetičnosti (poměr morfémů a slov), index složení slova (poměr kořenů a slov) a další.

Syntaktická topologie vychází ze šesti možných slovosledných kombinací subjektu, verba a objektu (S, V, O). Nejčastěji jsou realizování pouze 3 kombinace, SVO, SOV a VSO.

Strukturní typologie třídí jazyky podle toho, jak jazyk pracuje s kořenem slova a afixy (předponami a příponami). Většina jazyků je směs těchto typů, avšak jeden typ v daném jazyku často převažuje.

Prvním typem jsou **aglutinační** jazyky. U těchto jazyků se kumulativně připojují afixy ke kořenu slova, kde každý afix má pravě jednu funkci u všech slov. Převážně aglutinační jazyk je například maďarština, kde ve slově *szomszédokra* (česky „na sousedy“) přípona -k vyjadřuje pouze množné číslo a přípona -ra mluvnický pád. [28]

Dalším typem jazyků jsou **flektivní**, také označované jako flexivní (flexe = ohýbání). Podobně jako u aglutinačních jazyků se také připojují afixy ke kořenu slova, ale afixy můžou mít napříč slovy různé významy. Příkladem flexivního jazyka je čeština. V češtině koncovka -e u tvaru stroj označuje buď jednotné číslo, druhý pád mužského rodu, nebo množné číslo v prvním nebo čtvrtém pádu. Dalším rozdílem oproti aglutinačním jazykům je, že daná kategorie slov může být označena různými afixy. Například k vyjádření množného čísla se používají různé koncovky u různých slov (*muži, pánové, stroje*). [29]

Izolační neboli analytické jazyky se vyznačují tím, že k vyjádření gramatické funkce neupravují dané slovo, ale přidávají k němu pomocná slova. Typickým příkladem je angličtina, kde se například podměty vyjadřují osobním zájmenem místo toho, aby se skloňoval přísudek.

Introflektivní (introflexivní) vyjadřují gramatické funkce úpravou hlásek uvnitř kořenu slova. Například v němčině pro vyjádření množného čísla slova *matka* se slovo změní z *die Mutter* na *die Mütter*.

Poslední skupinou jsou **polysyntetické** jazyky. Tyto jazyky mají oproti izolačním jazykům vysoký poměr morfémů vzhledem k počtu slov. U těchto jazyků se na slovo může vázat celá řada afixů označujících nejen zájmena, ale také i podstatná jména. Může tak vzniknout i slovo, které samotné představuje celou větu. Příkladem mohou být eskymácko-aleutské jazyky. [29]

2.3 Krycí jména

Krycí jména[1] je desková hra založená na vytváření slovních asociací. Hra vytvořená českým vývojářem Vladimírem „Vlaadou“ Chvátilem byla publikována ve 40 jazycích a získala několik světových ocenění. Pravidla a základní herní strategie původní verze krycích jmen jsou popsány v předchozí práci (Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11]). Po úspěchu původní verze této hry bylo vytvořeno (především v angličtině) několik variant. Velmi rozšířenou variantou je Duet[2].

[1] https://krycijmena.cz/

[2] https://www.mindok.cz/userfiles/files/pravidla/codenames-duet-rules-cz.pdf

Verze Duet je vytvořená pro dva hráče, kteří hrají kooperativně. Místo hraní proti soupeři se snaží odkrýt tajné agenty předtím, než jim vyprší počet kol. Opět se hraje s 25 slovy rozložených do mřížky 5×5, kde oba hráči dostanou vlastní herní plánek se svými 9 agenty. Platí, že 3 agenti jsou u obou hráčů společní a 6 rozdílných. Na plánku agentů jsou místo 1 nájemného vraha vrazi 3. Hra se hraje na 9 až 11 kol, podle toho jakou si hráči zvolí obtížnost. Hru začíná jeden z hráčů, kdy dá nápovědu druhému na své agenty. Druhý hráč v této fázi nebere v potaz vlastní plánek a snaží se označit zamýšlená slova. Tah končí ve chvíli, kdy se hádající hráč dobrovolně rozhodne ukončit tah po alespoň jednom správně označeném slovu, nebo označí náhodného kolemjdoucího. Po dokončení si hádající hráč vezme jeden žeton tahu a následuje jeho napovídání s ohledem na vlastní herní plánek. Takto se hráči střídají, dokud neuhádnou všech 15 agentů (6 vlastních každého hráče a 3 společní), nebo vyprší počet stanovených kol. Opět také platí pravidlo, že kdykoliv jeden z hráčů označí nájemného vraha, tak hra okamžitě končí a hráči prohráli. Stejně jako u původní verze nesmí mít nápověda stejný kořen jako kterékoliv nezakryté herní slovo.

Oproti původní verzi zde není omezen počet slov, který může hráč během svého tahu označit. Po označení všech slov k současné nápovědě může tedy zkusit označit libovolný počet slov ke všem nepovedeným předchozím tahům.

Krycí jména Duet mají vlastní herní slova, ale je klidně možné použít slova z původní verze nebo použít obě slovní sady dohromady. Minimálně původní verze Krycích jmen vyšla ve 40 jazycích[3]:

afrikánština	arabština	libanonská arabština
bulharština	katalánština	čeština
dánština	němčina	řečtina
angličtina	španělština	estonština
perština	finština	francouzština
hebrejština	chorvatština	maďarština
indonéština	islandština	italština
japonština	korejština	litevština
lotyština	holandština	norština
polština	portugalština	rumunština
rumunština	slovenština	slovinština
srbština	švédština	thajština
filipínština	turečtina	ukrajinština
čínština		

Jak již bylo v úvodu zmíněno, cílem této práce je rozšířit původní systém umělého hráče Krycích jmen [11] tak, aby šlo co možná nejjednodušeji přidávat další jazyky. Z jazykového pohledu se zde vyskytují dva hlavní problémy, identifikace lemmat a kořenů slov. Identifikace kořenů je nezbytná věc, bez které by nebylo možné použít umělého hráče při vytváření herních nápověd. Lemmatizace může u některých jazyků přispět k úspěšnosti a ke snížení výpočetní a paměťové náročnosti herního systému. Slovo v různých tvarech, které má stejné lemma, nese stejný sémantický význam. Modely, které jsou určené k vyhodnocování podobnosti slov, natrénované pouze na lemmatech mohou dosahovat lepších výsledků a zvláště u morfologicky bohatých jazyků se tím snižuje celkový počet slov modelu. Ovšem u různých

[3] https://czechgames.com/en/codenames/downloads/

jazykových typologií bývá algoritmický proces identifikace lemmat a kořenů slov odlišný, proto existují nástroje většinou pro daný specifický jazyk.

2.4 Lemmatizace

Lemmatizace je technika, která se používá pro nalezení slov v základním tvaru (lemmat). Lemmatizačních přístupů existuje několik, někdy se i kombinují dohromady. Jedním z přístupů jsou algoritmy využívající hrubou sílu (brute-force algorithms). Tyto algoritmy prohledávají slovník uložený ve vhodné datové struktuře. Tento slovník obsahuje vztahy lemmat slov a jejich skloňovaných tvarů. Nevýhodou těchto algoritmů je potřeba velkého množství záznamů ve slovníku, který musí obsahovat všechny možné tvary daného slova. Ovšem tato metoda lze zkombinovat s jiným přístupem, kdy prohledávání slovníku se použije pouze pro nepravidelná slova.

Další přístup funguje na základě určování slovního druhu (POS: Part of Speech), čímž se určí, jak se mají zpracovat koncovky slova pro získání lemmatu. Bez určování slovního druhu by docházelo u některých jazyků k velké chybovosti, protože různé slovní druhy mohou mít různá pravidla a používané koncovky.

Stochastické přístupy fungují na principu určování pravděpodobnosti lemmatu daného slova, kde pravděpodobnost se „učí“ na trénovací množině. Pravděpodobnost je dána podle tabulky slov a jejich lemmat, která se během učení upravuje. Při hledání lemmatu se vybere lemma s nejvyšší pravděpodobností. [15]

Novější algoritmy využívají strojové učení, konkrétně neuronové sítě. Zde jsou uvedeny dva nástroje trénující seq2seq neuronové sítě pro lemmatizaci slov a další analýzu přirozeného jazyka. Lemmatizace pomocí neuronové sítě slibuje dobrou přenositelnost mezi různými jazyky, neboť se síť může nezávisle natrénovat pro každý jazyk zvlášť.

Lemmatizační algoritmy lze také dělit podle toho, zda při lemmatizaci daného slova využívají kontextovou informaci z ostatních slov ve větě, nebo ne. Kontextovou informaci z věty ale u Krycích jmen nelze využít, protože nápovědy a herní slova jsou jednoslovné celky.

2.5 Stemming

Stemming je proces velmi podobný podobný lemmatizaci, pouze místo lemmatu slova hledá jeho kmen. Algoritmy pro stemming lze také podobně rozdělit do několika kategorií. První kategorií jsou algoritmy, které odstraňují koncovky slova. Oproti lemmatizaci se v tomto případě většinou nepoužívají POS informace, díky čemuž jsou tyto algoritmy jednodušší. Algoritmy mohou mít parametr, který určuje „agresivitu“ odstraňování. Agresivnější přístupy mohou vést k problému zvaný overstemming. Jedná se o případ, kdy je odstraněno příliš mnoho písmen, což může vést k tomu, že u slov s různým kmenem bude vyhodnocen stejný kmen. Naopak málo agresivní přístupy mohou vést k understemmingu, kde se neodstraní dostatečný počet písmen. Algoritmy odstraňující koncovky slov jsou většinou vázané k danému jazyku, který definuje morfologická pravidla.

Další kategorií jsou statisticky založené algoritmy. Tyto algoritmy vyhodnocují kmen slova s danou pravděpodobnostní procedurou. Například YASS Stemmer rozděluje slova do shluků na základě tolerované editační vzdálenosti. HMM Stemmer používá Skryté Markovovy modely. [12]

Kapitola 3

Nástroje pro předzpracování textu

V této kapitole jsou představeny některé nástroje pro jazykově nezávislé předzpracování textu. Jedná se o neurální nástroje, které se trénují pro jednotlivé jazyky. Použitím těchto nástrojů se nabízí značná automatizace při přidávání celé řady nových jazyků do herního systému. Nejprve jsou však představeny techniky, které tyto nástroje používají. Jedná se o LSTM, seq2seq a attention. Nástroje jsou využity u zpracování Krycích jmen především pro lemmatizaci slov a určování slovních druhů pro výběr kandidátních nápověd.

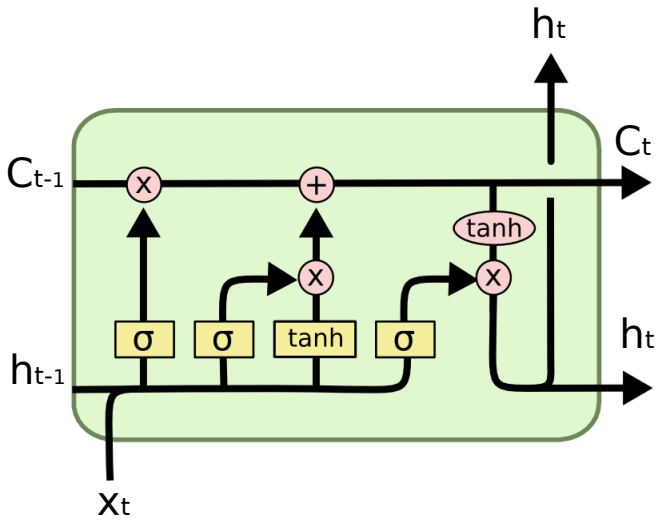
3.1 Architektura LSTM

Long short-term memory (zkráceně LSTM) je logická jednotka využívaná u rekurentních neuronových sítí (RNN: recurrent neural network). Oproti klasickým sítím, rekurentní sítě zpracovávají sekvenci vstupů, kde výstup sítě je ovlivněn nejen aktuálním vstupem (např. slovo), ale také všemi předcházejícími vstupy v dané sekvenci (např. předchozí slova ve větě). V případě obousměrného zapojení LSTM (Bi-LSTM), kde v jedné vrstvě jsou 2 zřetězení LSTM bloků (1 zleva a 1 zprava), lze využít kontextové informace z obou směrů v sekvenci dat.

Derivace ztrátové funkce se u rekurentních neuronových sítí počítá například pomocí algoritmu backpropagation through time (česky by se dalo přeložit jako zpětná propagace napříč časem). U delších vstupních sekvencí může při počítání zpětné propagace dojít k problému mizícího gradientu. Při dlouhých sekvencích nastane situace, kdy gradienty jednotlivých vah budou příliš malé na to, aby změnily svoji hodnotu, což znemožňuje, aby se síť učila. Tento problém řeší LSTM sítě. Na obrázku 3.1 je zobrazeno schéma LSTM bloku, kdy jsou tyto bloky v sekvenci na sebe napojené. Architektura LSTM bloku umožňuje vybírat s jakou váhou bude brán v potaz kontextovou informaci, což umožňuje vybírat pouze užitečné informace. Paměťová buňka C_t představuje stav v čase t. První brána forget gate na základě vstupu x_t a h_{t-1} určí, jak velkou část předchozího stavu C_{t-1} si ponechá. Input gate určuje, jak stav C_{t-1} a kandidátní stav $\tilde{C}_t$ vytvoří stav aktuální stav C_t. Nakonec output gate určí jaké informace by měly jít do skrytého stavu h_t.

3.2 Architektura Sequence-to-sequence

Sequence-to-sequence [25], zkráceně seq2seq, je architektura typu enkodér dekodér. Enkodér a dekodér jsou oba rekurentní neuronové sítě. Enkodér zpracovává sekvenci vstupu a jeho výstupem je vektor fixní délky. Tento výstupní vektor je potom předán dekóderu, který



Obrázek 3.1: Schéma LSTM bloku. Převzato z[2].

produkuje výstup variabilní délky. Výhoda tohoto přístupu je, že délka výstupní sekvence nemusí být stejná jako délka vstupu. Tento přístup je velmi výhodný a také se hojně využívá u strojového překladu.

3.3 Attention

Attention je mechanizmus, který řeší problém závislostí u dlouhých sekvencí dat. Česky by se dal přeložit jako mechanizmus pozornosti. Při klasickém rekurentním přístupu může u delších sekvencí k zahlcení informací a jejich nedostatečnému propagování. Myšlenka je taková, že dekodér při každém kroku využívá informaci pouze z vybraných částí vstupní sekvence, na které se zaměří.

Attention v seq2seq sítích funguje následujícím způsobem. V daném časovém kroku t spočítá dekodér hodnotu vnitřního stavu s_t. Nejprve se spočítá attention skóre e_t pro všechny vnitřní stavy enkodéru h_1 až h_N typicky jako skalární součin $e_t = [s_t^T h_1, \ldots, s_t^T h_N]$. Následně se spočítá pravděpodobnostní rozdělení jednotlivých složek vektoru pomocí funkce softmax $\alpha_t = softmax(e_t)$. Toto rozdělení se použije po spočítání vážené sumy vnitřních stavů enkodéru $a_t = \sum_{i=1}^{N} \alpha_{t,i} h_i$. Attention vektor a_t se potom zkonkatenuje s výstupem vnitřního stavu dekodéru s_t a pracuje se s ním dále stejně jako u seq2seq bez attention. [21]

Kromě skalárního součinu lze také použít multiplikativní nebo aditivní, které využívají k počítání matici váh, jejíž hodnoty jsou předmětem trénování.

3.4 Stanza

Stanza [19] je multifunkční nástroj pro zpracovaní přirozeného jazyka. Jedná se o sekvenci plně neuronových procesů, kde vstupem je obyčejný text a na výstupu se produkují jeho anotace. Konkrétně se jedná o tokenizaci, lemmatizaci, označení slovních druhů a dalších morfologických charakteristik (POS tagging), větný rozbor a rozpoznávání pojmenovaných entit. Architektura Stanzy je jazykově nezávislá, což umožňuje natrénovat modely na celé

[2] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

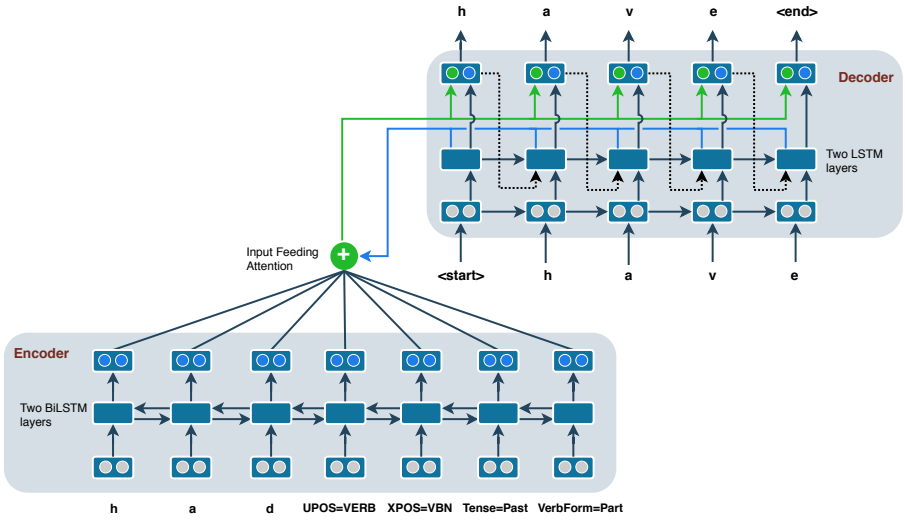
řadě jazyků. Autoři volně zpřístupnili modely pro 66 jazyků[1]. Tyto modely jsou natrénované na vícejazyčných korpusech a Universal Dependenices treebanks. UD treebanks je formát dat s anotovaným textem včetně jeho syntaktické analýzy neboli gramatických vztahů jednotlivých větných členů.

Architektura Stanzy se skládá z následujících modulů. První modul provádí tokenizaci. Po tokenizaci následuje POS tagging. Pro tento proces je použita Bi-LSTM neuronová síť. V dalším kroku se provádí lemmatizace slov ve větě. Lemmatizace je prováděna kombinací slovníkového lemmatizátoru a neuronového seq2seq lemmatizátoru. Po lemmatizaci se provádí analýza gramatické struktury ve větě, což je implementováno opět pomocí Bi-LSTM neuronové sítě. Nakonec je prováděno rozpoznávání pojmenovaných entit. Tento modul obsahuje jednovrstvou obousměrnou LSMT sít s CRF (conditional random field) dekóderem.

3.5 Turku neural parser pipeline

Turku neural parser pipeline [14] je také multifunkční nástroj pro předzpracování jazyka realizovaný pomocí neuronových sítí. Nástroj provádí segmentaci vět, tokenizaci, POS značkování, rozbor vztahů členů věty a lemmatizaci. Stejně jako u nástroje Stanza, autoři volně zpřístupnili natrénované modely pro více než 50 jazyků.

Lemmatizace je prováděná na principu strojového překladu. Na vstupu sítě je vždy jedno vstupní slovo s jeho POS značkami. Sekvence vstupu je tedy sekvence znaků daného slova. Sekvence znaků generovaných na výstupu celé sítě potom představuje lemma vstupního slova. Na obrázku 3.2 je zobrazena architektura sítě s ukázkovým anglickým slovem *had* (sloveso *mít* v minulém čase). Na výstupu je potom lemma tohoto slova *have*. Lemmatizace je implementována pomocí seq2seq architektury, kde enkodér tvoří dvouvrstvá Bi-LSTM síť využívající navíc mechanizmus attention. [13]



Obrázek 3.2: Schéma lemmatizátoru z Turku neural parser pipeline. Převzato z [13].

Dekodér je tvořen dvěma jednosměrnými LSTM vrstvami. Jeho vstupem je výstupní vektor z enkodéru zkonkatenovaný s attention vektory pomocí input-feeding přístupu. Jedná se o přístup, kdy jsou tyto vektory v daném kroku sekvence ovlivněny výstupy v předchozích

[1] https://stanfordnlp.github.io/stanza/available_models.html

krocích. Toto umožňuje, aby model bral v potaz předchozí „alignment“ [17]. Alignment ve strojovém překladu je termín, který určuje kolika slovy se přeloží vstupní fráze. Například jedno slovo z flektivního jazyka může být přeloženo na více slov v izolačním jazyce.

Kapitola 4

Sémantická reprezentace slov

V této kapitole jsou uvedeny různé modely, pomocí kterých lze vyhodnocovat sémantickou podobnost slov. Ve své bakalářské práci (Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11]) jsem pro určování sémantické podobnosti slov použil kombinaci dvou modelů. Model založený na počtu, který používá matici spoluvýskytů a počítá podobnost dvou slov pomocí Normalized Pointwise Mutual Information [5]. Jako prediktivní model byl použit Word2vec [6], respektivě jeho modifikace fastText [4] s architekturou Skip-gram a použití negativního vzorkování.

Normalized Pointwise Mutual Information (NPMI) počítá sémantickou podobnost dvou slov na základě poměru jejich společného výskytu v kontextu a jejich nezávislých výskytů. Vzorec pro výpočet podobnosti slov x a y je uveden v rovnici 4.1. Jednotlivé pravděpodobnosti jsou počítány z matice spoluvýskytů.

$$NPMI(x, y) = \frac{\log \frac{P(x \wedge y)}{P(x) \times P(y)}}{-\log P(x \wedge y)} \tag{4.1}$$

Word2vec je neuronová síť, která se trénuje na predikci slov na základě jejich kontextu. Pravděpodobnost predikce slova v při kontextu slova w se počítá pomocí jejich vektorových reprezentací pomocí rovnice 4.2.

$$p(v|w, \theta) = \frac{exp(in_w^T out_v)}{\sum_{v'=1}^{V} exp(in_w^T out_{v'})} \tag{4.2}$$

Kde θ představuje všechny trénovací parametry $\{in_v, out_v\}_{v=1}^{V}$, což jsou vstupní a výstupní reprezentace slov. V je velikost slovníku, všechny vektory mají předem daný počet dimenzí D.

Architektura Skip-gram predikuje celý kontext na základě jednoho vstupního slova, kde predikce jednotlivých slov jsou počítány nezávisle na sobě. Predikce kontextových slov $\boldsymbol{v}$ při velikosti kontextu C a vstupního slova w se počítá:

$$p(\boldsymbol{v}|w, \theta) = \prod_{i=1}^{C} p(v_i|w, \theta) \tag{4.3}$$

Model fastText je rozšířením původního modelu Word2vec, kde slova nejsou reprezentována jako atomické jednotky, ale rozdělují se na takzvané n-gramy, což jsou podslova o délce 3 až 6 znaků. Tyto n-gramy mají vlastní vektorovou reprezentaci. Při určení vektorové reprezentace slova se použije jak vektor slova samotného, tak i suma vektorů jeho n-gramů.

Díky této modifikaci lze aproximovat vektorové reprezentace slov, které nejsou ve slovníku modelu.

4.1 Latent Semantic Analysis

Jedna z dalších metod založených na počtu je Latent Semantic Analysis [7] (zkráceně LSA, česky přeloženo jako skrytá sémantická analýza), někdy také označována jako Latent Semantic Indexing (LSI). Oproti NMPI, kde se bere v potaz, jak často se dva termíny vyskytují ve společném kontextu, funguje LSA na principu, že porovnává podobnosti kontextů, ve kterých se dané termíny vyskytují.

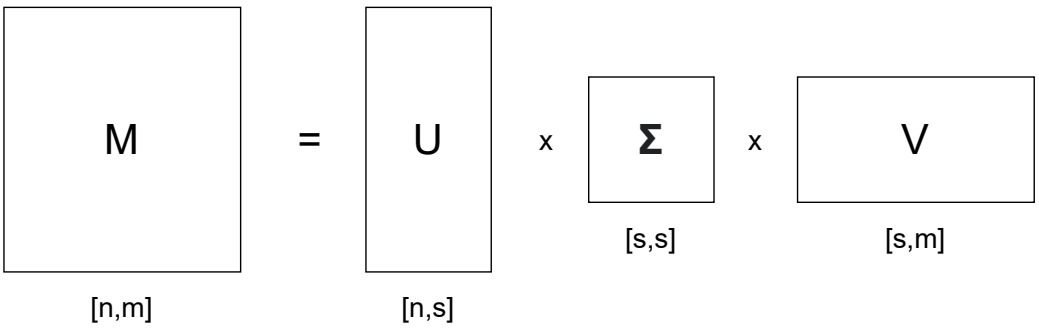
LSA vychází z řídké matice vytvořené z korpusu, kde řádky jsou jednotlivé termíny (slova) a sloupce dokumenty. Taková matice M má potom rozměry *počet unikátních slov* $\times$ *počet dokumentů*. Hodnoty jednotlivých elementů této matice mohou být buď počty výskytů jednotlivých slov v daných dokumentech, nebo hodnoty TF-IDF. TF-IDF (term frequency-inverse document frequency) je statistický údaj, který udává jakou má daný termín váhu v daném dokumentu.

$$\text{TF-IDF}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \log \frac{N}{|d' \in D : t \in d'|} \tag{4.4}$$

První zlomek v rovnici 4.4 udává frekvenci termínu t v dokumentu d. Počítá se jako počet výskytů t v d děleno celkovým počtem slov v d. Druhý zlomek počítá inverzní dokumentovou frekvenci. N je počet všech dokumentů v korpusu a jmenovatel je počet všech dokumentů, kde se t vyskytuje. Použití této hodnoty místo počtu výskytů jednotlivých slov lépe vystihuje jakou váhu má slovo t v dokumentu d, bez ohledu na četnost slova v celém textu. Slova jako předložky, spojky a třeba sloveso *být* se budou v jednom dokumentu vyskytovat relativně často, ale také se budou často vyskytovat v ostatních dokumentech, takže reálně nepřináší užitečnou informaci. Díky právě nízké inverzní dokumentové frekvenci nebude u těchto slov výsledná hodnota tak vysoká, než kdyby se použil čistě počet výskytů.

Singular Value Decomposition

Pro zjištění sémantických informací vytváří LSA skryté proměnné, které v podstatě interpretují témata. Ta jsou vytvořena faktorizací matice M pomocí Singular Value Decomposition (SVD). Faktorizace takovéto matice je demonstrována na obrázku 4.1.



Obrázek 4.1: LSA faktorizace matice.

Faktorizací vzniknou 3 matice U, Σ a V, které když se vynásobí, tak vznikne původní matice M. Matice M má rozměry $n \times m$, kde n jsou slova a m dokumenty. Matice U má tvar $n \times s$, kde s jsou skryté proměnné reprezentující témata. Hodnota prvku této matice udává, jak moc je dané slovo asociováno s daným tématem. Matice Σ je diagonální čtvercová matice tvaru $s \times s$. Prvky na diagonále udávají důležitost jednotlivých témat. Poslední matice U s tvarem $s \times m$, určuje poměr témat jednotlivých dokumentů reprezentovaných ve sloupcích. Tímto procesem lze redukovat informace do méně dimenzí, které jsou dány hyperparametrem s.

Pro účel získání sémantické podobnosti dvou samostatných slov se nabízí možnost vzít jejich vektorové reprezentace z matice U a porovnat jejich náležitost k jednotlivým tématům například pomocí kosinové vzdálenosti.

4.2 Adaptive Skip-gram

Jeden z problémů modelu Word2vec/fastText je, že ignoruje vícevýznamovost jednotlivých slov, pokud nejsou jejich významy explicitně vyznačeny v korpusu před trénováním. Například slovo *los* může mít význam jako zvíře, nebo jako lístek v loterii. Při trénování se potom tyto případné významy slova smísí dohromady, což může mít za následek zprůměrování vzdáleností vektorů k ostatním slovům daných významových skupin.

Jeden z modelů, který bere v potaz vícevýznamovost slov, je Adaptive Skip-gram (zkráceně AdaGram) [3]. Jedná se o architekturu, která rozšiřuje model Word2vec o učení jednotlivých významů slov pomocí Bayesovského neparametrického přístupu. Není tedy potřeba jednotlivé významy slov v korpusu dopředu nějak anotovat. Podle předpokladu, že slovo může mít více významů, je zde zavedena skrytá proměnná z, která určuje, o který význam se jedná. Každý význam slova má vlastní vektorovou reprezentaci. Původní věrohodnostní funkce Word2vec Skip-gram 4.3 se změní na rovnici 4.5. Symbol k udává jeden z významů vstupního slova.

$$p(\boldsymbol{v}|z = k, x, \theta) = \prod_{i=1}^{C} p(v_i|w_k, \theta) \tag{4.5}$$

Při trénování modelu se zohledňují jednotlivé významy pouze u vstupního prototypu slova. Na významy výstupních slov není brán ohled. Autoři článku tvrdí, že zohledňování významu výstupních slov by značně komplikovalo trénování a uvedený přístup je dostatečný k zachycení vícevýznamovosti slov [3].

Různá slova mohou mít různý počet významů, které navíc budou známy až v průběhu trénování modelu. Není tedy vhodné přidělit napevno každému slovu stejný počet významů. Adaptivní přidělování významů slov je vyřešeno pomocí varianty Dirichletova procesu zvané „stick breaking process“. Doslova lze přeložit jako lámání klacku, což obrazně odpovídá danému procesu. Celý „klacek“ představuje sumu pravděpodobností všech významů slova. Při vytvoření nové komponenty se z „klacku“ ulomí určitá část, jejíž délka odpovídá její pravděpodobnosti. Další „lámání“ se provádí na dosud ještě nepoužité části „klacku“. Takto je možno vytvářet části teoreticky do nekonečna a jejich suma pravděpodobností bude 1.

Pro určení délek, které se „odlomí“ se používá Beta pravděpodobnostní rozdělení uvedené v rovnici 4.6. Jedná o spojité rozdělení, které se v bayesovké inferenci používá k počítání posteriorní pravděpodobnosti Bernoulliho rozdělení.

$$Beta(x|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \tag{4.6}$$

Rozdělení má 2 parametry a a b, které definují křivku funkce. Gamma funkce v prvním zlomku se používá pro normalizaci. Pro stick-breaking process jsou použity parametry $Beta(1,\alpha)$, kde α je hyperparametr. Vyšší hodnota α způsobí, že se budou „odlamovat" kratší části, což povede k většímu počtu a vyšší granularitě významů slov. Naopak menší hodnota bude mít tendenci vytvářet méně významů. Teoreticky lze takto vytvořit nekonečně mnoho významů slov, ale maximální možný počet je počet výskytů daného slova v korpusu. Očekávaný počet významů slova je dán vzorcem $\alpha \log(n_w)$, kde n_w je počet výskytů slova w.

$$p(z=k|w,\beta) = \beta_{wk}\prod_{r=1}^{k-1}(1-\beta_{wr}), \quad p(\beta_{wk}|\alpha) = Beta(\beta_{wk}|1,\alpha), \quad k=1,\dots \tag{4.7}$$

Rovnice 4.7 udává apriorní pravděpodobnosti jednotlivých významů slov z. Symbol β_{wk} představuje k-tý vzorek z Beta rozdělení. Díky vynásobením s opačnými hodnotami předchozích vzorků bude vždy platit, že suma všech vzorků nebude přesahovat hodnotu 1.

Věrohodnostní funkci celého AdaGramu lze napsat následovně:

$$p(Y,Z,\boldsymbol{\beta}|X,\alpha,\theta) = \prod_{w=1}^{V}\prod_{k=1}^{\infty} p(\beta_{wk}|\alpha)\prod_{i=1}^{N}[p(z_i|x_i,\boldsymbol{\beta})\prod_{j=1}^{C} p(y_{ij}|z_i,x_i,\theta)] \tag{4.8}$$

Symboly X, Y představují všechna vstupní/výstupní slova. Počet všech slov je V. Symbol Z je množina o velikosti N a představuje významy všech slov. C je velikost kontextu. Při čtení rovnice odzadu se počítá pravděpodobnost výstupních slov v kontextu na základě vstupního slova a jeho významu. Tato hodnota se násobí pravděpodobností daného významu. To vše se vynásobí pro všechny významy vstupního slova x. Dále se výraz pronásobí pravděpodobnostmi vzorků z Beta rozdělení. Vše se vynásobí pro všechna vstupní slova z korpusu.

Jeden způsob, jak trénovat AdaGram je maximalizováním marginální věrohodnosti funkce 4.8 s ohledem k parametrům θ. Kvůli skrytým proměnným $\boldsymbol{\beta}$ a Z, nelze funkci zderivovat. Proto se model trénuje pomocí stochastic variational inference.

Po natrénování modelu lze spočítat pravděpodobnost daného významu slova x na základě vstupních kontextových slov $\boldsymbol{y}$.

$$p(z=k|x,\boldsymbol{y},\theta) \propto p(\boldsymbol{y}|x,k,\theta)\int p(k|\boldsymbol{\beta},x)p(\boldsymbol{\beta}|\mathcal{D},\theta,\alpha)d\boldsymbol{\beta} \tag{4.9}$$

Rovnice 4.9 počítá pravděpodobnost významu k slova x. $\mathcal{D}$ představuje množinu všech dvojic $x_i, \boldsymbol{y_i}$ z trénovacích dat.

4.3 Probabilistic FastText

Dalším modelem zohledňujícím vícevýznamovost slov je Probabilistic FastText [2]. Zde se jedná o rozšíření původního modelu fastText, takže oproti modelu AdaGram využívá n-gram podslova.

Slova jsou reprezentována jako směsice Gaussových rozdělení s K komponenty, kde jedna komponenta odpovídá jednomu významu daného slova. Slovo w je asociováno s funkcí hustoty 4.10,

$$f(x) = \sum_{i=1}^{K} p_{w,i}\mathcal{N}(x; \overrightarrow{\mu}_{w,i}, \Sigma_{w,i}) \tag{4.10}$$

kde $\{\mu_{w,i}\}_{k=1}^{K}$ jsou vektory středních hodnot, $\{\Sigma_{w,i}\}_{k=1}^{K}$ jsou kovarianční matice a $\{p_{w,i}\}_{k=1}^{K}$ jsou pravděpodobnosti jednotlivých komponent jejichž suma je 1. Kovarianční matice ve vícerozměrném Normálním rozdělení udává na diagonále variance jednotlivých dimenzí a ostatní prvky udávají korelaci mezi danými dvěma dimenzemi.

Vektor střední hodnoty (rovnice 4.11) se počítá principiálně podobně jako u původního fastTextu, což je suma vektoru celého slova a jeho n-gramů.

$$\mu_w = \frac{1}{|NG_w| + 1}(v_w + \sum_{g \in NG_w} z_g) \tag{4.11}$$

Symbol v_w představuje vektor celého slova w, NG_w je množina n-gramů a z_g je vektor n-gramu g z dané množiny.

Na obrázku 4.2 je demonstrace vektorů slov a jeho n-gramů ve 2D prostoru. V části a lze vidět vektory n-gramů a tučnou šipkou výsledný vektor střední hodnoty. V části b lze vidět vektorové umístění slova *bank* (anglicky banka nebo břeh) ve více významech a k nim vektory podobných slov při použití jedné Gaussove komponenty. V části c je zobrazen stejný případ, avšak slovo *bank* je reprezentováno dvěma Gaussovými komponentami.

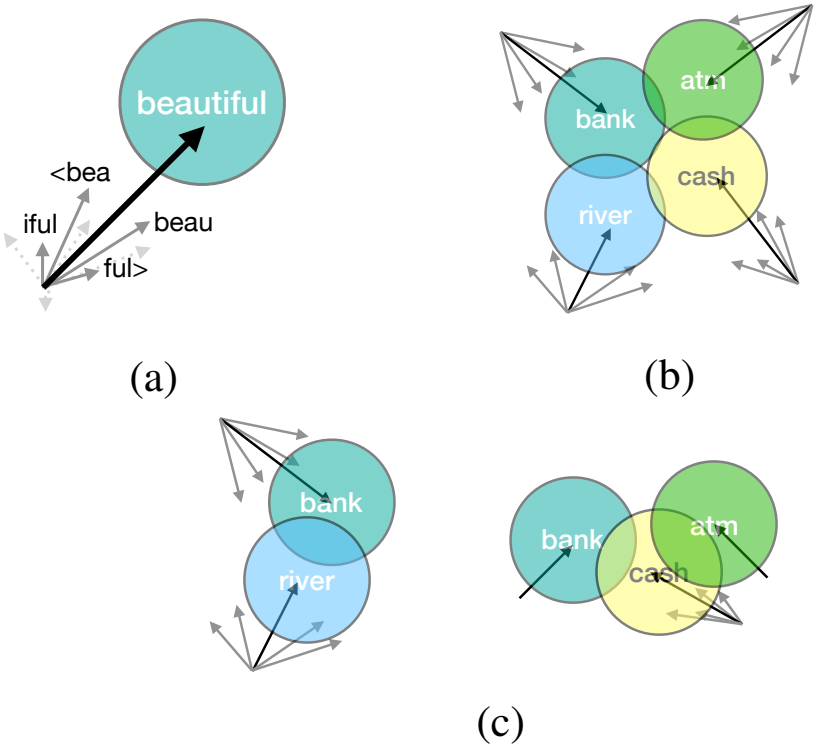
Běžně se podobnost slov u modelů, kde jsou slova reprezentována vektory, počítá pomocí skalárního součinu potažmo kosinovou vzdáleností. Zde jsou však slova reprezentována funkcemi pravděpodobnostního rozdělení, proto se podobnost počítá generalizovaným skalárním součinem v Hilberovském prostoru, který se nazývá expected likelihood kernel. Mezi dvěma slovy f a g je definovaná energie $E(f, g) = \log \int f(x)g(x)dx$, kde funkce f a g jsou funkce hustoty 4.10 daných slov. Energie má analytické řešení a lze spočítat následovně:

$$E(f, g) = \log \sum_{j=1}^{K} \sum_{i=1}^{K} p_{f,i}p_{g,i}exp(\xi_{i,j}) \tag{4.12}$$

kde $\xi_{i,j}$ je částečná energie odpovídající podobnosti komponentě i slova f a komponentě j slova g. Částečnou energii lze spočítat pomocí vektorů středních hodnot a kovariančních matic následujícím způsobem:

$$\begin{aligned}\xi_{i,j} &= \log \mathcal{N}(0; \overrightarrow{\mu}_{f,i} - \overrightarrow{\mu}_{g,i}, \Sigma_{f,i} + \Sigma_{g,i}) = \\ &= -\frac{1}{2}\log det(\Sigma_{f,i} + \Sigma_{g,i}) - \frac{D}{2}\log(2\pi) - \frac{1}{2}(\overrightarrow{\mu}_{f,i} - \overrightarrow{\mu}_{g,i})^T(\Sigma_{f,i} + \Sigma_{g,i})^{-1}(\overrightarrow{\mu}_{f,i} - \overrightarrow{\mu}_{g,i})\end{aligned} \tag{4.13}$$

kde D je počet dimenzí Normálních rozdělení jednotlivých komponent. Při trénování modelu se hledají parametry v_w pro každé slovo w a z_g pro každý n-gram g, které budou vhodně upravovat vektory středních hodnot (rovnice 4.11). Pro zjednodušení trénování jsou použity diagonální kovarianční matice pro všechny komponenty s hyperparametrem α, který určuje hodnotu na diagonále. Model se trénuje tak, že se snaží zvýšit energii slova w a jeho



Obrázek 4.2: Reprezentace Gaussových rozdělení jednotlivých významů slov. Převzato z [2].

kontextového páru c, aby byla vyšší, než vzorek negativního páru slova w a kontextu n o minimální hodnotu m. Ztrátová funkce je potom definovaná:

$$L(w, c, n) = max[0, m - \log E(w, c) + \log E(w, n)] \tag{4.14}$$

Autoři doporučují použít $K = 2$ komponenty, což dostatečně reprezentuje různé významy slov. Je sice možné trénovat model s více komponenty, to už však ale nezlepšuje úspěšnost modelu při vyhodnocování podobností slov. [2]

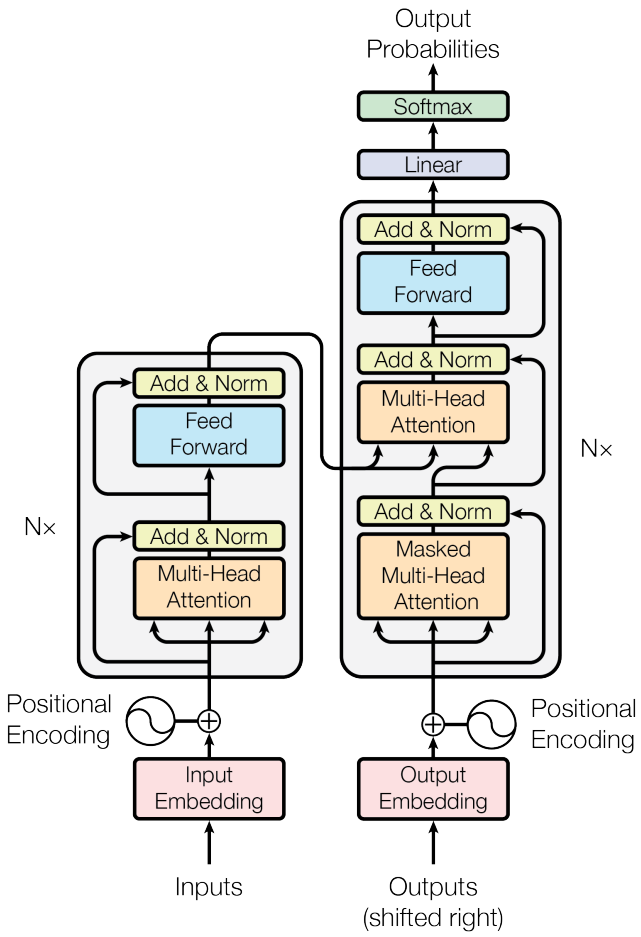
4.4 Transformers

Dosud zmíněné modely pracují na úrovni slov a jejich kontextů. Při predikci slova neuvažují skladbu věty ve smyslu pozic jednotlivých slov ve větě. Při používání slov ve větách může jejich význam ovlivňit to, jak jsou v rámci věty použitá. Nejnovější přístup ke zpracování celých vět jsou modely typu transformers [26]. Oproti dřívějším přístupům pomocí rekurentních neuronových sítí dosahují tranformers lepších výsledků a mnohem kratšího času trénování. Pro využití kontextových informací nepoužívají žádnou rekurenci a spoléhají se čistě na mechanizmus attention, což umožňuje značnou paralelizaci při trénování.

Architektura transformers (obrázek 4.3) se skládá z enkodéru a dekodéru. Obě části převádí vstup do vektorové podoby, která je analogická k vektorovým reprezentacím u modelu Word2vec. Na tyto vektorové reprezentace se aplikuje poziční kódování pomocí goniometrických funkcí, které zajistí, že se u jednotlivých slov bere v potaz jejich pozice ve větě.

Enkodér se skládá z $N = 6$ identických vrstev, kde jedna vrstva obsahuje dvě podvrstvy. První je multi-head self-attention mechanizmus a druhá plně propojená dopředná síť. U každé podvrstvy je zapojeno reziduální připojení, po kterém následuje vrstvová normalizace. Vrstvová normalizace normalizuje hodnoty napříč všemi neurony v dané vrstvě před aplikováním aktivační funkce.

Dekodér má podobnou architekturu jako enkodér. Jeho vstupy jsou zpožděny o jednu pozici, aby se predikovalo slovo na i-té pozici z předchozích slov na pozicích 1 až $i-1$. Rozdíl oproti enkodéru je, že mezi 2 podvrstvy je vložená ještě jedna, která počítá multi-head self-attention přes výstup enkodéru. Dále jsou v dekodéru upraveny self-attention mechanizmy tak, aby počítaly pouze s pozicemi, které jsou na vstupu před aktuálně počítanou pozicí, což zajišťuje, že se pro predikované slovo berou v potaz pouze předchozí slova.



Obrázek 4.3: Architektura modelu transformers. Převzato z [26].

Attention

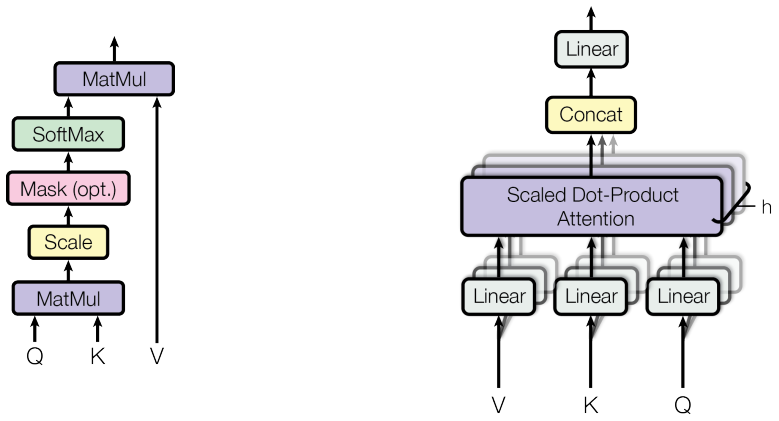
Mechanizmus attention slouží k určení na jakých částech vstupní sekvence je aktuální vstup závislý, neboli s jakou váhou definují jednotlivá slova ve větě kontextovou informaci aktuálně zpracovávaného slova. Schéma mechanizmu attention u modelu Transformers lze vidět v levé části obrázku 4.4. Pro každý vstupní vektor $\boldsymbol{x}$ se vytvoří vektory $\boldsymbol{q}$ pro query (*dotaz*),

$\boldsymbol{k}$ pro key (*klíč*) a $\boldsymbol{v}$ pro value (*hodnotu*) pomocí násobení vektoru $\boldsymbol{x}$ s danou maticí W_q, W_k nebo W_v. Vektory představující dotazy a klíče mají počet dimenzí d_k, vektory představující hodnoty mají počet dimenzí d_v. Každý jednotlivý dotaz $\boldsymbol{q}$ se skalárně vynásobí se všemi vektory klíčů $\boldsymbol{k}$, kde výsledné hodnoty se vydělí číslem $\sqrt{d_k}$ (proto má název škálovaný attention) a aplikuje se softmax funkce. Tato hodnota udává, jakou váhu mají slova na daných pozicích pro aktuální vstup. Většinou bude mít nejvyšší hodnotu slovo samotné, ale mohou se právě objevovat vyšší hodnoty i u ostatních slov. Tyto váhy se aplikují na vektory hodnot $\boldsymbol{v}$ pomocí maticového násobení. Při tomto procesu jsou parametry matic W_q, W_k a W_v předmětem trénování. V praxi se však všechny dotazy $\boldsymbol{q}$ reprezentovány maticí Q a attention se počítá paralelně jako:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}V) \tag{4.15}$$

Multi-Head Attention

Attention se však nepočítá pouze jednou, ale hned několikrát. Od toho název Multi-Head Attention (dalo by se přeložit jako vícehlavý attention). Princip spočívá v tom, že místo jedné sady transformačních matic W_q, W_k a W_v se jich použije několik, kde každá promítá vstupní vektor do jiného prostoru. Takto se spočítá několik attention hodnot, které se následně zkonkatenují do jednoho vektoru, který se ještě naposledy jednou transformuje. Tento postup je znázorněn v pravé části obrázku 4.4. Použití Multi-Head Attention umožňuje zachytit různé typy vazeb. Ty lze demonstrovat například na větě: *Kamarád hrál včera šachy.* Při počítání několika attention hodnot pro slovo *hrál*, může jedna reprezentovat vazbu „*Kdo?*“ a vyšší hodnota bude u slova *Kamarád*. Druhá hodnota může reprezentovat „*Co hrál?*“ a vyšší hodnota zde bude u slova *šachy*.[1]



Obrázek 4.4: Nalevo schéma škálovaného attention, napravo schéma multi-head attention. Převzato z [26].

BERT

Jedním z předtrénovaných vícejazyčných modelů typu transformers je BERT [8]. Jedná se o architekturu využívající Transformers bloky. Dostupné jsou dva typy modelů. $\text{BERT}_{\text{BASE}}$,

[1] Inspirace z https://towardsdatascience.com/transformers-141e32e69591

který má 12 Transformers bloků, počet dimenzí skrytých vrstev 768 a 12 attention hlav. $\text{BERT}_{\text{LARGE}}$ má 24 bloků, 1024 dimenzí a 16 hlav. Pro vektorovou reprezentaci vstupů využívá WordPiece embeddings. WordPiece spočívá v tom, že celé slovo nemá vlastní vektorovou reprezentaci (výjimka krátká slova), ale vektorově reprezentované jsou části slov, které jsou sdílené napříč všemi slovy. Při tokenizaci se například slovo „motorka“ rozdělí na části „moto“ a „##rka“, kde „##“ značí, že se jedná a pokračování předchozího slova. Speciální tokeny jsou `[CLS]`, který značí začátek vstupní sekvence a `[SEP]`, který slouží na oddělení vět v případě, že na vstupu je dvojice vět. Dvojice vět se může použít například pro úkol odpovídání na otázku, kde první věta je otázka a druhá odpověď.

BERT se trénuje následujícím způsobem. Nejprve se model předtrénuje na dvou úkolech současně. Jedná se o maskované jazykové modelování a predikci další věty. Při maskovaném jazykovém modelování se náhodně zakryje 15 % tokenů z každé vstupní sekvence a model se trénuje na jejich predikci. U predikce další věty se použije na vstupu dvojice vět tak, že v 50 % případů druhá věta opravdu následuje po první a v 50 % případů se druhá věta vybere náhodně s negativním označením. Po dokončení předtrénování modelu se mohou upravit vstupní a výstupní vrstvy pro konkrétní specifikaci úkolu a model se na daném úkolu dotrénuje.

Výhodou WordPiece embeddings je, že při trénování na vícejazyčném korpusu neroste počet vektorových reprezentací s každým slovem v jednotlivých jazycích. Vektorovou reprezentaci vět (případně slov) lze teoreticky získat ze skrytých stavů různých vrstev. Takových kombinací vektorů může být více, ať už čistě z jedné dané vrstvy nebo konkatenace, suma, průměr více různých vrstev[2]. Otázkou je, zda tyto vektory opravdu reprezentují sémantický význam slov, když jednotlivá slova jsou rozdělená na části, které jsou v případě vícejazyčného korpusu společné mezi jazyky. Další teoreticky možná varianta by byla využití vypočítaných attention hodnot z různých vrstev/hlav[3]. Zde nastává otázka, jak by se tato závislost v kontextu dala použít pro případ Krycích jmen, kde se jedná o samostatné jednoslovné celky, kdežto BERT pracuje s celými větami.

[2] https://colab.research.google.com/drive/1ZQvuAVwA3IjybezQOXnrXMGAnMyZRuPU

[3] Inspirace z https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1

Kapitola 5

Návrh

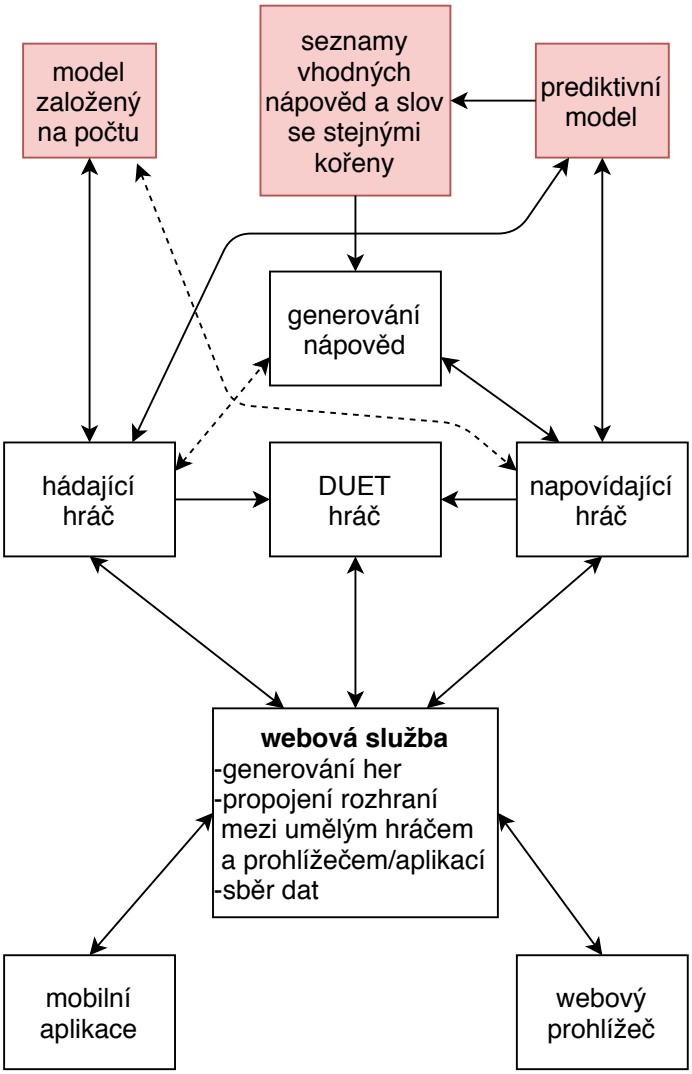
Tato kapitola obsahuje návrh sytému pro hraní umělého hráče Krycích jmen rozšířený o možnost přidávání dalších jazyků. Výchozí stav původního systému před zpracováním předkládané práce byl následující. Systém implementoval hráče obou rolí Krycích jmen (role napovídání a role hádání) jak ve standardní verzi, tak i ve verzi Duet. Implementovanými jazyky byla čeština a angličtina. Pro určování sémantické podobnosti slov byla použita kombinace modelů NPMI a fastText. Pro češtinu byly tyto modely natrénovány na korpusu CWC-2011 [22]. Pro angličtinu byla matice spoluvýskytů pro výpočet NPMI vytvořená z anglické wikipedie a natrénovaný fastText byl stažen z internetu[1] [18]. Vytvořená webová služba podporovala v obou jazycích testování hráče v obou rolích standardní verze, tak i variantu Duet, kde uživatel hraje kooperativně s počítačem.

5.1 Architektura systému

Schéma systému lze vidět na obrázku 5.1. Herní role využívají kombinaci modelů následujícím způsobem. Model založený na počtu je u hádajícího hráče využit k počítání podobností slov. U generování nápověd se podobnost počítá pouze z prediktivního modelu. Nemusí se jednat přímo o prediktivní model, musí však umět vyhodnocovat podobnost dvou slov a také hledat n nejpodobnějších slov k daným pozitivním a negativním slovům v přijatelném časovém intervalu. Pro generování nápověd je vhodné využít i frekvenci slov v korpusu. Tuto frekvenci má jak fastText, tak i model s maticí spoluvýskytů. Pomocí parametru při vytváření hráče lze nastavit, z kterého modelu tuto frekvenci používat. V situaci, kdy není pro daný jazyk dostupná matice spoluvýskytů, lze využít frekvenci z fastTextu. Naopak, pokud jako prediktivní model je použitý jiný než fastText, který nemá informaci o frekvenci slov, lze použít frekvenci z modelu založeného na počtu.

Vazba generování nápověd s napovídajícím hráčem je samozřejmá. Hádající hráč také může využívat generování nápověd. U příchozí nápovědy s číslem 2 nebo 3 provádí „úvahu“ nad vlastní nápovědou, kterou by použil, což může ovlivnit výslednou podobnost slov k aktuální nápovědě [11]. Hráč v roli Duet využívá obě role, které se pro něho každé kolo střídají.

[1] https://fasttext.cc/docs/en/english-vectors.html



Obrázek 5.1: Schéma systému, červeně zvýrazněné jsou jazykové závislé moduly.

5.2 Přidávání nového jazyka

Pro přidání nového jazyka je v první řadě zapotřebí získat jak herní slova v daném jazyce, tak model/y. Jako varianta se nabízí použít pouze jeden model, například pro angličtinu, a slova z jiných jazyků překládat do angličtiny a následně vyhodnotit podobnosti přeložených slov. Zde však nastává problém, že každý jazyk má jiná homonyma a mnohoznačná slova, jejichž použití může velmi dobře spojovat herní slova s odlišným významem. Například české slovo *hřeben* může být nástroj na česání vlasů (anglicky *hairbrush*) nebo skupina vrcholků hor ležících těsně vedle sebe (anglicky *ridge*). Dalším příkladem nevhodným pro překlad je slovo *koruna*, které se vždy přeloží do angličtiny na *crown*, avšak v českém textu bude mít často význam jako měna, zatímco v jiných jazycích bude pravděpodobně figurovat pouze jako panovnický klenot.

Prediktivní model je možno buď natrénovat na korpusu daného jazyka, nebo stáhnout například fastText z internetu. Výhodou fastTextu je, že umí aproximovat vektory neznámých slov a také jsou dostupné předtrénované modely pro 157 jazyků[2].

Pro model založený na počtu je potřeba vytvořit matici spoluvýskytů. Tento model však není nezbytně nutný pro vytvoření hráče. Hádající hráč by pro vyhodnocení podobnosti používal pouze prediktivní model. Pokud by byl použitý pouze prediktivní model, který nemá informaci o frekvenci slov, je vhodné místo něj použít alespoň nějaký objekt, který bude na dotaz vyhodnocení podobnosti jakýchkoliv slov vracet nulovou podobnost, ale bude mít uloženou frekvenci jednotlivých slov. Ignorování frekvence slov by vedlo k vytváření nápověd s „pochybnými“ málo frekventovanými slovy.

V rámci morfologického zpracování se u obou rolí provádí lemmatizace herních slov a u hádajícího hráče lemmatizace příchozí nápovědy. Tato činnost není nutná, ale může zlepšit úspěšnost, pokud jsou modely trénované na lemmatech. Nezbytně nutné je však určování kořenů slov při generování nápověd. Bez toho by napovídající hráč nemohl vytvářet pravidly povolené nápovědy. Také je potřeba vytvořit seznam „smysluplných“ slov s vhodnými slovními druhy slov pomocí POS značkování, který udává povolené nápovědy. V případě, že by tento seznam nebyl dostupný a při trénování modelu nebyl zvolený vhodný stoplist a požadovaná minimální frekvence slov, mohlo by dojít k tomu, že výsledné nápovědy by byly tvořeny slovy, která by člověk nikdy nepoužil (předložky, spojky, příslovce bez významného sémantického významu…).

Formát jazykově závislých statických dat

Všechny seznamy povolených a zakázaných slov jsou dopředu vytvořeny a uloženy v souborech v textovém formátu pro případné jednoduché ruční editování. Seznamy lze vytvořit dopředu, protože jsou statické a pouhé vyhledávání v seznamu je u generování nápověd výrazné rychlejší než samotné vytváření.

Tyto soubory je potřeba vytvořit před přidáním herní podpory v daném novém jazyce. Herní slova jsou uložena ve formátu slovo na řádek. To samé platí pro seznam povolených nápověd vzniklých z filtrování slovníku modelu pomocí POS značkování. Formát seznamu stejných kořenů herních slov má na řádku:
`herní_slovo:slova ze~slovníku modelu se stejným kořenem oddělená čárkou`.

Při generování nápovědy se nejprve vyberou z modelu kandidátní nejpodobnější slova k různým kombinacím cílených slov. Tato slova se filtrují seznamem povolených slov a poté se v závislosti na aktuální herní situaci zkontroluje, zda se nevyskytují v seznamech stejných kořenů daných herních slov.

Seznam vhodných slov pro nápovědy

Seznam vhodných slov pro nápovědy je filtrován ze slovníku modelu pomocí POS značkování. Nepovolené slovní druhy jsou: zájmena, číslovky, předložky, spojky, částice, citoslovce a ostatní tokeny jako interpunkce a slovní druhy vyhodnocené jako neznámé.

Dalším kritériem je frekvence slova v trénovacím korpusu. Při použití slov s velmi malou frekvencí výskytu může dojít ke generování nevhodných nápověd. Jedním z faktorů je, že běžný člověk nemusí mít tak obsáhle vědomosti a velkou slovní zásobu, aby znal definice všech vzácných pojmů. Dále se tímto odstraní slova, která zůstala v předzpracovaném

[2] https://fasttext.cc/docs/en/crawl-vectors.html

korpusu jako „smetí“ nebo případně je použit model, který byl natrénovaný na nevhodně předzpracovaném korpusu.

Odhad společných kořenů slov

To, jaká přesně slova jsou z podhledu pravidel zakázaná použít, se může jazyk od jazyku lišit[3] . V češtině a slovenštině je uvedeno, že nesmí být použitá nápověda, která má stejný kořen slova jako nějaké slovo aktuálně ve hře. V některých jazycích se zmiňuje, že nápověda nesmí být vytvořena skloňováním jiného slova ve hře a nebo nesmí mít společnou část slova. Často se udává příklad slova spojeného ze dvou slov. V češtině je to slovo *ptakopysk*, kde například nesmí být použité nápovědy jako *pták*, *ptákovina* a *pysk*. Z pravidel v různých jazycích tedy vyplývá, že nápověda by neměla mít na první pohled společnou část slova s některým ze slov ve hře.

U češtiny byly kořeny herních slov napsány ručně. Při vytváření seznamů slov se stejným kořenem u herních slov se pouze kontrolovalo, zda napsaný kořen není podřetězcem daného slova (s ignorováním diakritiky). Zpracování anglických herních slov bylo plně automatické. U angličtiny, jakožto izolačního jazyka, je stemmer dostatečně úspěšný.

U jazyků, které byly implementovány v rámci předkládané práce, byl použit následující algoritmus 1 pro aproximaci tvrzení, zda dvě slova mají společný kořen/základ. Rozhodnutí, zda se jedná o slova se stejným kořenem je dáno poměrem délek slov a délky nejdelšího společného podřetězce (s ignorováním diakritiky).

Algoritmus 1: Odhad zda mají slova společný kořen.

Vstup : slova v a w
Výstup: ANO: slova mají společný kořen, NE: slova nemají společný kořen

```
v' = remove_diacritics(v)
w' = remove_diacritics(w)
min_len = minimum(length(v'), length(w'))
max_len = maximum(length(v'), length(w'))
match = find_longest_substring(v', w')
if length(match) >= 5 then
    return ANO
if length(match) == 4 then
    if min_len <= 6 and max_len <= 8 then
        return ANO
    else
        return NE
if length(match) == 3 then
    if min_len <= 4 and max_len <= 6 then
        return ANO
    else
        return NE
return NE
```

[3] https://czechgames.com/en/codenames/downloads/

Hodnoty jednotlivých rozhodovacích prahů byly experimentálně testovány se zběžnou kontrolou agresivity na výsledných seznamech společných kořenů herních slov. Tento algoritmus byl testován pro vyhodnocení společných kořenů ve slovenštině, němčině, francouzštině a vypadá, že by mohl být vhodný i pro celou řadu dalších jazyků (minimálně jazyků používajících latinskou abecedu) s případnou úpravou rozhodovacích prahů.

5.3 Strategie herních rolí

Strategie herních rolí zůstává stejná, jak v předchozí bakalářské práci (Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11]), až na generování nápověd. Samotná strategie herních rolí, až na výjimky přidané ke zlepšení úspěšnosti v češtině, je jazykově nezávislá.

Člen operativy

Člen operativy (hádající hráč) jednak využívá informaci z vlastních předchozích nepovedených kol, aby označil zbylá cílená slova, tak také při neúspěšném protivníkově tahu si váhuje potencionální nepřátelská slova, kterým se ve sporných situacích spíše vyhne. Dále u některých obdržených nápověd provádí „úvahu“ nad tím, jakou nápovědu by použil on, což může potencionálně ovlivnit konečné vyhodnocení podobností obdržené nápovědy s herními slovy. napovídající hráč si také pamatuje vlastní předchozí tahy a v případě, že spoluhráč nezvládl označit zamýšlená slova v předchozích kolech, preferují se nápovědy na slova, která ještě nebyla v rámci hry cílená.

Nově také může hádající hráč využít informace z článků z Wikipedie v reálném čase. Pro příchozí nápovědu se vyhledá její článek v daném jazyce, pokud existuje. Následně se hledá v daném článků počet výskytů všech aktuálních herních slov. Vyhodnocená podobnost pomocí sémantických modelů pro dané slovo se zvýší o n %, kde n je počet výskytů slova v článku. Pokud je příchozí nápověda vícevýznamová a existuje více článků, vezme se maximálně prvních 5 článků, aby vyhodnocení netrvalo příliš dlouho. Výhodou použití Wikipedie je, že lze použít pro jakýkoliv jazyk. Při vyhodnocení se ukázalo, že použití Wikipedie při hádání slov mírně zvyšuje celkovou úspěšnost na testovací sadě (viz kapitola 7). Nevýhodou je, že se článek stahuje v reálném čase a tah může trvat až 3 sekundy.

Hlavní špión

Hlavní špión (napovídající hráč) používá novou hodnotící funkci kvality nápovědy, která lépe integruje četnost výskytů kandidátního slova v textu do celkové kvality. Pro každou n-tici, kde n je 1 až 5, se vybere 1000 nejpodobnějších slov, kde cílená slova jsou pozitivní vzorky a nájemný vrah negativní vzorek. Tato slova jsou filtrována seznamem vhodných nápověd a seznamem kořenů aktuálních herních slov. Slova, která splňují zmíněná kritéria jsou vyhodnocena pomocí rovnice 5.1.

$$U_h = |T|^{\frac{|M|}{|E|}} \cdot (\min(\{\mathrm{sim}(h,w)|w \in T\}) - \max(\{\mathrm{sim}(h,w')|w' \in (N \cup E \cup A)\})) \\ +\min(\log_{10} \mathrm{freq}(h); 5) \cdot 0{,}1 \tag{5.1}$$

Množina M představuje slova vlastního týmu, množina $T \subseteq M$ jsou slova cílená nápovědou h. Množiny E, N, A představují slova nepřátelského týmu, náhodné kolemjdoucí

(neutrální slova) a nájemného vraha (případně více vrahů ve verzi Duet). Funkce sim představuje normalizovanou podobnost dvou slov v intervalu <0,1>. Prostřední část vzorce počítá, jaký je rozdíl mezi podobnostmi nejméně podobného cíleného slova a nejvíce podobného slova, které nepatří do vlastního týmu. V případě, že nápověda cílí tři a více slov, může být jedno neutrální slovo v seřazeném pořadí podobností mezi cílenými slovy a nejvíce podobné necílené slovo se bere buď nájemný vrah nebo slovo nepřátelského týmu.

Čím více bude cílených slov, tím bude slabší celková asociace mezi slovy, ale na druhou stranu není vhodné používat nápovědy jenom na jedno nebo dvě slova. Proto je rozdíl podobností násoben počtem cílených slov. Výjimka je u cílení jednoho slova, potom se rozdíl nenásobí číslem 1, ale číslem 0,5. Nápověda na jedno slovo bývá většinou synonymum cíleného slova a výsledná podobnost je velmi vysoká. Vynásobení číslem 0,5 se výrazně zvýší preference volby nápovědy pro více slov a nápověda na jedno slovo se použije v případech, kdy se nepodaří nalézt přiměřeně uspokojivou nápovědu na více slov.

Pro lepší adaptabilitu k aktuální herní situaci je číslo udávající počet cílených slov umocněno poměrem *počet zbývajících vlastních slov / počet zbývajících nepřátelských slov*. Toto odráží situaci jak moc hráč vyhrává, nebo prohrává. Pokud hráč vyhrává, nevyplatí se tolik riskovat a používat nápovědy, které cílí více slov, jejichž asociace s nápovědou by nemusely být tak silné, což by mohlo vést k tomu, že by spoluhráč zbytečně označil špatná slova. Takto budou mít preferenci nápovědy s vyšší vyhodnocenou podobností s cílenými slovy. Naopak pokud hráč prohrává, vyplatí se spíše riskovat a použít nápovědu cílící více slov, aby byla možnost dohnat nepřátelský tým.

Posledním parametrem je frekvence výskytů nápovědy v korpusu. Čím je slovo frekventovanější, tím je vyšší šance, že ho bude spoluhráč znát s jeho případnými asociacemi. Dalším faktorem je, že výsledná nápověda bude spíše vypadat jako slovo, které by použil člověk a nebyla by příliš „robotická“. Například při cílení slov *pohádka* a *mech* by model vyhodnotil, že slovo *křemílkovský* bude mít nepatrně vyšší podobnost než slovo *křemílek*. Slovo *křemílek* ale bude mít značně vyšší frekvenci výskytu, což potom jako výsledná nápověda bude vypadat lépe.

Pokud by však frekvence slova měla v rovnici příliš vysokou váhu, výsledné nápovědy by byly velmi frekventovaná slova s nízkou podobností s cílenými slovy. Je proto brána v potaz maximální hodnota počtu výskytů 100 000 a výsledná hodnota zlogaritmována a vydělena 10. Toto nastavení se ukázalo v rámci manuálních experimentů jako vhodný poměr mezi frekvencí a podobností s cílenými slovy.

Duet hráč

Duet hráč používá obě role jak hlavního špióna, tak člena operativy s pár drobnými rozdíly. Fakt, že tři slova (agenti) jsou společná pro oba dva spoluhráče, je brán v potaz. Pokud Duet hráč označí ve svém hádajícím tahu tři slova, která měl v seznamu agentů pro vlastní napovídání, potom ví, že určitě žádné z jeho slov není společné se slovy spoluhráče a ke konci hry už neoznačí žádné slovo, které patří do vlastního seznamu agentů. Tím, že pravidla umožňují označovat neomezený počet slov, tak hráč v roli člena operativy se snaží „dohádat“ všechna zbylá slova z předchozích nepovedených kol a v roli hlavního špióna již nepoužívá nápovědu typu nekonečno.

Jelikož ve variantě Duet nejsou slova nepřátelského týmu, jako měřítko aktuální situace se použije počet zbývajících kol. Místo počtu zbývajících nepřátelských slov $|E|$ se v rovnici 5.1 použije počet zbývajících kol hry. Při výběru kandidátních slov se použijí tři slova (nájemní vrazi) jako negativní vzorky.

Úpravy hráče v českém jazyce

V rámci hraní v českém jazyce, pro který byl herní systém primárně vyvíjen, jsou použity jazykově specifické úpravy. První je použití slovníku spisovné češtiny, který je použit v rámci člena operativy pro dodatečnou úpravu vyhodnocených podobností nápovědy s herními slovy. Pokud se u dvojice vyhodnocovaných slov objeví jedno v definici druhého ve slovníku spisovné češtiny, zvýší se celková podobnost těchto slov o 0,25. Číslo bylo zvolené na základě experimentů na vyhodnocovací sadě záznamů her. Tato úprava již byla implementována v předchozí bakalářské práci Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11].

Druhá úprava, která byla implementována v rámci této práce, je použita při generování nápověd. Jedná se o použití slovníku současné češtiny[4]. Zde je zapsáno přes 3000 unikátních slov do kategorií ve stromové struktuře. Každá kategorie může mít více podkategorií, které blíže specifikují danou skupinu slov. V uvedených 3000 slovech se vyskytuje 230 ze 400 herních slov klasické verze Krycích jmen a 188 ze 400 herních slov verze Duet. Jelikož jsou si některé sousední kategorie ve slovníku velice podobné, byly tyto kategorie spojeny a případně ručně upraveny.

Po vygenerování nápovědy standardním způsobem se zkontroluje, zda existuje kategorie ze slovníku, která by spojovala minimálně stejný počet cílených slov. Pokud se tato kategorie najde a její název (případně více názvů) nemá společný kořen s žádným herním slovem, použije se jako nápověda místo původně vygenerované. Podmínkou je, že se v dané kategorii nesmí vyskytovat žádné nepřátelské slovo a nájemný vrah. Náhodný kolemjdoucí se může vyskytovat, pokud by nápověda cílila alespoň tři slova. Jelikož by se tato nápověda vynutila „natvrdo“ bez kontroly vyhodnocených podobností slov a některé sousedské kategorie jsou si podobné, musí platit, že žádná sousední kategorie v dané úrovni podstromu nesmí obsahovat nájemného vraha. Dále sousední kategorie může obsahovat slovo nepřátelského týmu pouze v případě, že hráč aktuálně prohrává.

Použití zmíněných kategorií také částečně řeší jeden z jevů modelu fastText, s jehož použitím je preference jako nápovědu použít slovo „jedno z druhu“, neboli místo pojmu, který by generalizoval cílená slova, vybere jedno další, které k nim patří do skupiny. Například na slova *banán* a *maso* bude výsledná nápověda třeba *brambora*. Člověk naopak má tendenci asociovat takové pojmy nějakým generalizovaným výrazem jako například *jídlo*.

[4] https://www.nechybujte.cz/slovnik-soucasne-cestiny-temata

Kapitola 6

Implementace

Tato kapitola popisuje výsledný systém z pohledu implementace. Jsou zde popsány použité technologie a implementační detaily úprav jednotlivých částí systému. Především se jedná o úpravy z pohledu vícejazyčné podpory.

6.1 Použité technologie

Sytém byl implementovaný v programovacím jazyce Python 3. Pro prvotní trénování modelu fastText byla použita implementace skupiny Facebook Research[1]. Pro další využívání modelu fastText a jeho případného dotrénování byla použita knihovna gensim [20], která byla také využita pro vytvoření modelu LSA. Model Probabilistic FastText byl natrénovaný autory dostupnou implementací[2] v jazyce Python 2. Pro spuštění v Python 3 stačilo upravit některé konstrukce a definice konstant. Model AdaGram také má od autorů dostupnou implementaci[3], která je v programovacím jazyce Julia. Model byl v jazyku Julia natrénován a pomocí dalšího dostupného repozitáře[4] byl výsledný model převeden do formátu JSON a z něho do formátu, který je možný načíst v jazyce Python 3.

Pro přidání podpory v některých jazycích byly použity články z Wikipedie v daném jazyce. Ke zpracování byl použit nástroj WikiExtractor[5]. K implementaci webové služby byl použit framework Flask[6]. Dokumentace zdrojových kódů byla vygenerována pomocí nástroje Sphinx[7].

Jazykově zaměřené technologie

V rámci původní bakalářské práce Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11] byly jazyky čeština a angličtina zpracovány dedikovanými nástroji pro daný jazyk. Lemmatizace a POS značkováni je pro češtinu prováděno pomocí nástroje MorphoDiTa [24]. Lemmatizace a stemming anglického jazyka je prováděno pomocí knihovny

[1] https://github.com/facebookresearch/fastText
[2] https://github.com/benathi/multisense-prob-fasttext
[3] https://github.com/sbos/AdaGram.jl
[4] https://github.com/lopuhin/python-adagram
[5] https://github.com/attardi/wikiextractor
[6] https://flask.palletsprojects.com/en/1.1.x/
[7] https://www.sphinx-doc.org/en/master/

NLTK[8]. Knihovna spaCy[9] byla použita k POS značkování pro účel výběru vhodných slov pro nápovědy.

Pro podporu dalších jazyků byl použit nástroj Stanza. Jedná se konkrétně o lemmatizaci a POS značkování. Nástroje Stanza nebo Turku neural parser pipeline dosahují velmi podobných výsledků, Stanza je však výrazně rychlejší. Oba nástroje podporují implementaci na CPU i GPU. Na CPU zvládne Stanza zpracovat zhruba 12 jednotlivých slov za sekundu. Turku neural parser pipeline zpracuje jedno slovo za 3 až 4 sekundy na CPU. Oba nástroje jsou vhodné pro zpracování celých vět, případně odstavců, kde můžou značně využít paralelizaci. U Krycí jmen se zpracovávají pouze jednotlivá slova. Další výhodou nástroje Stanza je, že stačí pouze importovat knihovnu a načíst model v rámci spuštěného skriptu. Turku neural parser pipeline je potřeba spouštět zvlášť v serverovém módu a pomocí http dotazů zasílat text ke zpracování, což by přinášelo vyšší údržbu při správě webové aplikace. Především kvůli vícenásobné rychlosti je ve výsledném systému využíván nástroj Stanza.

6.2 Herní role

Herní role implementované v bakalářské práci byly upraveny tak, aby byly jazykově nezávislé. Jazykově závislá lemmatizace je předána rolím formou parametru. Jedná se o objekt, který má implementovanou metodu `__call__()`. Při jejím zavolání se vstupním slovem se vrátí seznam lemmat daného slova. Pro češtinu je lemmatizátor implementovaný třídou `Czech_Lemmatizer`, pro angličtinu `English_Lemmatizer` a pro ostatní jazyky `Multilingual_Lemmatizer`. Lemmatizátor pro ostatní jazyky vrací nejen lemma slova, ale také i slovo samotné. V případě, že by byl použit model, který nebyl natrénovaný na lemmatech, mohlo by používání pouze lemmat k vyhodnocení snížit úspěšnost modelu. Na druhou stranu, pokud by model byl natrénovaný na lemmatech vytvořených stejným nástrojem, je potřebné pro vyhodnocení podobností použít lemma slova. Lemmatizátory nefungují na 100 %, takže je potřeba mít slova z korpusu a potom i slova při hře lemmatizována stejně. Při vyhodnocování podobností slov jsou použita všechna lemmata vrácená lemmatizátorem a použita kombinace s nejvyšší podobností.

Role člena operativy je implementována třídou `Operative`, role hlavního špióna třídou `Spymaster`.

Pro roli hráče Duet jsou nejprve implementované dvě třídy. Třída `Operative_DUET`, která dědí ze třídy `Operative` a třída `Spymaster_DUET`, která dědí ze třídy `Spymaster`. Obě tyto třídy mají překryté některé metody pro fungování ve verzi Duet. Samotný Duet hrát je implementovaný třídou `DUET_player` a vytváří jednu instanci z obou zmíněných tříd.

6.3 Webová služba

Webová služba běží na školním serveru athena3 a je přístupná na adrese http://athena3.fit.vutbr.cz:8086/. Webová služba nabízí uživateli hrát ve třech rozhraních. V prvním hraje uživatel za oba týmy hlavního špióna a systém vytvoří dva členy operativy na hádání slov. V druhém rozhraní jsou role prohozeny, kde uživatel hraje za oba členy operativy a systém vytvoří dva hlavní špióny. Třetí rozhraní je kooperativní hra Duet, kdy uživatel a systém hrají spolu a snaží se odkrýt všechna slova, dokud nevyprší zvolený

[8] https://www.nltk.org/
[9] https://spacy.io/

počet tahů. Ve všech rozhraních jsou volitelné sady herních slov, pokud jsou v daném jazyku dostupné. Lze tedy použít jenom slova ze standardní verze nebo jenom z verze Duet nebo případně obě sady sloučené dohromady.

V rámci této práce byla webová služba modularizována a přepsána tak, aby jednotlivé směrovací funkce (routes) byly jazykově nezávislé. Pro práci se všemi potřebnými daty pro různé jazyky slouží třída `Data_Loader`. Tato třída načítá data jednotlivých jazyků zadaných v konfiguračních souborech a poskytuje je v rámci celé webové služby.

Konfigurační soubor

Konfigurační soubor jazyka pro webovou službu má následující formát, kde jednotlivé položky jsou cesty k jednotlivým souborům:

- VEC_MODEL=<prediktivní model>
- COUNT_MODEL=<matice spoluvýskytů>
- MORPHO=<v případě češtiny morfologický slovník>
- DEF_DICT=<v případě češtiny zpracovaný slovník spisovné češtiny>
- HYPERNYMS=<manuálně definovaná hyperonyma>
- VEC_VOCAB=<seznam povolených nápověd>
- CODENAMES_WORDS=<herní slova standardní verze>
- CODENAMES_BLACKLIST=<seznamy společných kořenů herních slov standardní verze>
- DUET_WORDS=<herní slova Duet>
- DUET_BLACKLIST=<seznamy společných kořenů herních slov Duet verze>
- STANZA_CODE=<v případě ostatních jazyků zkratka jazykového modelu Stanza>
- PERSISTENT_LOAD=<yes/no>

Povinné parametry webové služby v daném jazyce jsou prediktivní model, lemmatizátor (Stanza kód v případě nových jazyků) a minimálně jedna slovní zásoba pro alespoň hádání slov. Pro generování nápověd je také potřeba seznam povolených nápověd a seznam kořenů pro danou slovní zásobu. V případě nedostupného lemmatizátoru lze použit jednoduchý objekt, který při zavolání bude vracet slovo ze vstupu. Dále je pro generování nápověd vhodné, aby jeden z definovaných modelů měl přístup k frekvenci výskytů slov.

Prediktivní model je implicitně načítaný pomocí `gensim.models.FastText.load()`. Model fastText, který je uložený pomocí knihovny gensim jako Numpy datové pole (.npy formát) umožňuje výrazně rychlejší načítání do paměti. Načítání stejného modelu uloženého binárně pomocí knihovny fastText od Facebook Resarch trvá kolem 200 s. Načítání stejného modelu uloženého jako npy trvá necelých 30 s.

Práci s maticí spoluvýskytů implementuje třída `CountModel`, která počítá hodnotu NPMI. Pro český lemmatizátor potřeba načíst morfologický slovník MorfFlex CZ [23]. Dále je možnost načíst hyperonyma pro lepší kvalitu kvalitu nápověd, které byly ručně sepsány pro češtinu ze slovních zásob Krycích jmen. Ta jsou použita na stejném principu jako témata

ze slovníku současné češtiny (sekce 5.3). Při načítání seznamů společných kořenů herních slov se zkontroluje, zda seznamy obsahují všechna poskytnutá herní slova. Pokud seznam neodpovídá, nepovolí se použití dané slovní zásoby.

Plánovače

Pro zvládnutelnou podporu většího množství jazyků jsou pouze modely pro češtinu, angličtinu a slovenštinu načtené v paměti po celou dobu běhu webové služby. Ostatní modely jsou načítány při požadavku hraní v daném jazyce. Pro uvolňování nepoužívaných jazyků je spuštěný plánovač, který každých 8 minut nastaví značku každého jazyka jako nepoužívaný. Pokud není daný jazyk v rámci webové služby použit, uvolní se patřičné modely během dalšího běhu plánovače. I když Python explicitně nezaručuje vrácení alokované paměti zpět operačnímu systému, modely pro jeden jazyk zabírají v paměti 10 až 20 GB, což je dostatečně velké pro Python, aby danou paměť vrátil. Pokud je aktuální virtuální paměť RAM použitá z více než 70 %, webová služba nepovolí načítání nových jazyků, dokud se paměť neuvolní.

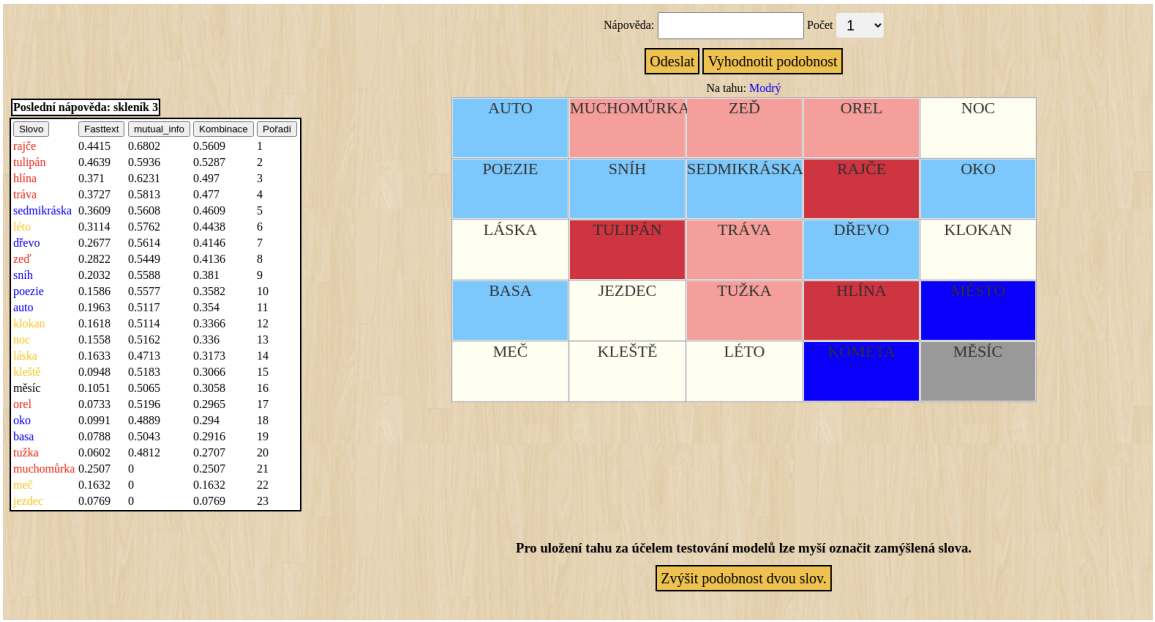
Jelikož je generování her v prvním kole relativně časově náročné kvůli předpočítání nejpodobnějších slov jednotlivých kombinací, jsou herní role s generováním nápověd u perzistentně načtených jazyků předgenerované. Začínající hráč má rovnou připravenou první nápovědu. Hráč, který nezačíná, má alespoň předpočítaná nejpodobnější slova. Generování probíhá každý den ve 2 hodiny ráno, kdy se generuje do 30 her pro roli hlavního špióna a hráče Duet pro každý ze zmíněných jazyků.

Pokud se uživateli během jakéhokoliv vyhodnocení podobnosti slov počítačem nebude zdát, že podobnost dvou slov není vyhodnocená jako dostatečně vysoká, má možnost odeslat žádost o dotrénování modelu na dané dvojici. Tyto žádosti se ukládají a jednou denně v rámci plánovaného předgenerování nápověd pro nové hry se také modely na těchto žádostech dotrénují. Z každé žádosti dvojice slov se vytvoří 10 stejných vět obsahujících tato dvě slova a podvzorkování nejčastějších slov se nastaví na 0 (jinak by se nic nedotrénovalo). Bylo zjištěno, že toto dotrénování zvýší normalizovanou podobnost slov o zhruba 0,08 až 0,15. Při následné kontrole na vyhodnocovací sadě bylo zjištěno, že dotrénování nemá prakticky žádný vliv na vyhodnocenou podobnost ostatních slov v modelu. Většina vyhodnocených podobností zůstala stejná a některé se změnily maximálně o hodnotu 0,001.

Záznamy her

U webové služby bylo implementováno zlepšené ukládání záznamů, kde se ukládá každý provedený tah, a u nápověd generovaných počítačem i zamýšlená slova pro srovnání. Uživatel má ve všech rozhraních, kde zadává nápovědu, možnost i označit zamýšlená slova. Pokud tak učiní, uloží se další záznam přímo ve formátu pro vyhodnocování modelů na sadě herních tahů. V rámci sbírání dat se u her Duet ukládají výsledky her, zda hráči vyhráli a případně kolik tahů potřebovali k odkrytí všech potřebných slov. U těchto výsledků se ukládá i IP adresa pro možnost rozlišení úspěšností jednotlivých uživatelů v rámci jednotlivých jazyků.

Dále byla webová služba rozšířena o možnost, kdy uživatel může označit vygenerovanou nápovědu jako neplatnou kvůli stejnému kořenu s některým herním slovem. V tomto případě dostane možnost vybrat dané herní slovo a jeho seznam slov se stejným kořenem se aktualizuje.



Obrázek 6.1: Ukázka rozhraní webové služby v češtině, kde uživatel zadal poslední nápovědu *skleník 3* a systém zaujímá roli člena operativy.

6.4 Automatizace podpory nových jazyků

Pro přidání podpory nového jazyka je nejprve potřeba získat herní slova. Pro všechny chybějící jazyky byla herní slova získána z online rozhraní od Czech Game Editon[10] pomocí poloautomatického skriptu `get_game_words.py`, který v cyklu resetuje „stůl“ pro daný jazyk, dokud nenarazí na všech 400 herních slov.

Po získání herních slov je potřeba mít alespoň prediktivní model, buď stažený z internetu nebo vlastnoručně natrénovaný. Pro vytvoření seznamu vhodných nápověd je nutné provést POS analýzu slovníku prediktivního modelu. Pro analýzu a vytvoření seznamu jsou implementované funkce v modulu `save_hint_game_vocab.py`. Po vytvoření seznamu vhodných nápověd je dále potřeba vytvořit seznamy společných kořenů herních slov se seznamem vhodných nápověd. Funkce pro vytváření seznamu společných kořenů pro češtinu, angličtinu a algoritmus 1 jsou implementované v modulu `save_game_stemms.py`. Poslední věc je vytvoření konfiguračního souboru pro webovou službu (sekce 6.3).

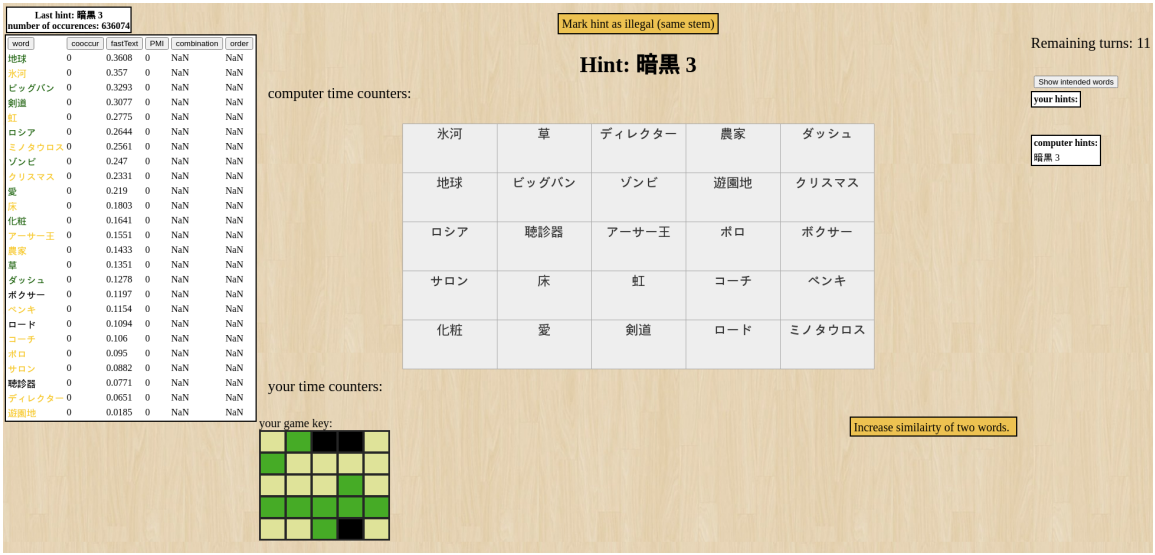
Pro přidání podpory nových jazyků byl vytvořen skript `add_language.py`, který automatizuje všechny výše zmíněné kroky. Jako parametry dostane url adresu pro stažení fastText modelu, kód jazyka a herní slova. Skript stáhne potřebný Stanza model pro POS analýzu, vytvoří potřebné seznamy, uloží fastText model ve formátu npy a vytvoří konfigurační soubor pro webovou službu.

V rámci zjišťování, zda mají dvě slova společný kořen, byla pro odstranění diakritiky použita knihovna `unidecode`[11], která převádí řetězec do ASCII formátu. To však má značný vliv při používání pro jazyky, které nepoužívají latinskou abecedu, protože jejich řetězce do ní převádí. Při stahování modelů bylo zjištěno, že některé jazyky mohou také obsahovat i slova zapsaná latinskou abecedou. Například japonský fastText má ve slovníku slova jako

[10] https://codenames.game/
[11] https://pypi.org/project/Unidecode/

かがみ (zrcadlo), Kagaku (věda). Aby byla všechna slova zkontrolována ve stejné abecedě, jsou převáděna do ASCII formátu. Vzhledem k nedostatečné znalosti ostatních jazyků byly prahy algoritmu 1 ponechány tak, jak jsou původně navrženy.

Výsledný systém aktuálně podporuje 36 jazyků. Ze 40 jazyků zmíněných v sekci 2.3 nebyla implementována podpora pouze pro libanonskou arabštinu, islandštinu, thajštinu a filipínštinu. Pro tyto jazyky nejsou dostupné natrénované modely Stanza. Turku neural parser pipeline sice má dostupný model pro thajštinu, ale vzhledem k jeho rychlosti na CPU pro samostatné jednoslovné vstupy by nešlo provést POS analýzu celého slovníku v rozumné době.



Obrázek 6.2: Ukázka rozhraní webové služby verze Duet v japonštině, kde systém vygeneroval první nápovědu ve hře.

Kapitola 7

Vyhodnocení

V této kapitole jsou popsány dosažené výsledky. Je zde uvedeno vyhodnocení úspěšnosti jednotlivých modelů v českém jazyce. Pro vyhodnocení herních rolí v různých jazycích jsou uvedeny statistiky ze záznamů her nasbíraných pomocí webové služby.

7.1 Datové sady

Ve výsledném systému byly modely natrénovány na korpusu CWC-2011 [22]. Jedná se o kolekci zpráv, článků, blogů a dalších literárních celků. Celkem obsahuje 218 653 428 vět s 957 728 unikátními slovy. Pro testování úspěšnosti LSA byl použit korpus All.vert, který se skládá ze starých novinových článků a knih. Tento korpus je rozdělený na jednotlivé dokumenty, což umožňovalo snadné vytvoření *termín* × *dokument* matice. U korpusu CWC-2011 to nebylo tak jednoduše možné, protože jsou jednotlivé věty v korpuse zamíchané. Pro pokus dotrénování výsledného modelu na aktuálnějším korpuse byl použit Czech Text Document Corpus v 2.0 [16] (zkráceně CTDC), který obsahuje přes 11 tisíc novinových článků.

Pro ostatní jazyky byly staženy předtrénované modely fastText dostupné od Facebook AI Research [9]. Tyto jsou natrénované na korpusech Wikipedie a Common Crawl. Všechny modely obsahují 2 milióny slov, což může značit dvě věci. Trénovací korpus pro daný jazyk je buď velmi obsáhlý nebo není dostatečně předzpracovaný, mohou platit i obě tvrzení dohromady. Ve výsledném systému je výjimka u němčiny, pro kterou byl použit předtrénovaný model fastText od deepset GmbH[1], který je natrénován na německé Wikipedii. Hlavním důvodem, proč byl použit, je, že obsahuje všechna herní slova z německých Krycích jmen. Oproti tomu fastText od Facebook Research neobsahuje 17 německých herních slov.

Ve výsledném systému jsou použité matice spoluvýskytů pro češtinu, slovenštinu a angličtinu. Pro podporu ostatních jazyků je použit pouze model fastText. Pro češtinu je matice spoluvýskytů vytvořená na korpusu CWC-2011, pro slovenštinu a angličtinu jsou matice vytvořené z lemmatizované Wikipedie.

7.2 Výsledky modelů pro češtinu

Představené modely v kapitole 4 společně s modely vyhodnocovanými v rámci předchozí bakalářské práce (Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména [11]) byly vyhodnoceny v češtině na ručně sepsaných záznamech her Krycích jmen. Vyhodnocovací sada je tvořena 680 tahy, které obsahují člověkem zadanou nápovědu a jeho zamýšlená

[1] https://deepset.ai/german-word-embeddings

slova a slova označená jeho týmovým spoluhráčem. Použitou metrikou pro vyhodnocení úspěšnosti modelů je mAP (mean Average Precision). Při vyhodnocení se spočítají podobnosti jednotlivých slov ve hře s použitou nápovědou a slova se seřadí sestupně podle vyhodnocené podobnosti. Úspěšnost jednoho tahu je počítaná jako:

$$\text{mAP} = \frac{\sum_{w \in T} \frac{\text{očekávané pořadí}(w)}{\text{reálné pořadí}(w)}}{|T|} \tag{7.1}$$

kde T je množina slov, která byla označena spoluhráčem, případně slova zamýšlená napovídajícím hráčem. Na konkrétní pořadí cílených slov mezi sebou není brán ohled, tedy pokud cílená slova jsou A, B, C a vyhodnocené pořadí bude C, A, B, pořád bude mAP $= 1$.

model	**specifikace**	**mAP (%)**
NMPI		**76,79**
LSA	počty výskytů (CWC-2011)	47,31
	TF-IDF (CWC-2011)	**53,36**
	TF-IDF (All.vert)	53,17
Word2vec	sg, ns	**71,68**
GloVe	dostupný z [10]	**68,76**
fastText	sg, ns (CWC-2011)	**74,99**
	sg, ns (CWC-2011 + CTDC)	69,92
	stažený předtrénovaný	62,86
AdaGram	nejvyšší kombinace	70,35
	vážená suma	**72,37**
	nejpravděpodobnější - kontext	66,88
	vážená suma - kontext	71,39
Probabilistic FastText	2 komponenty	**64,98**
BERT	1. vrstva	**34,29**
	poslední 4 vrstvy - průměr	32,14
	poslední 4 vrstvy - suma	31,86
	attention hodnoty	17,92
NMPI & fastText	aritmetický průměr	**77,81**
	překlad cs → en	56,32

Tabulka 7.1: Tabulka úspěšnosti modelů na vyhodnocovací sadě

V tabulce 7.1 jsou uvedeny úspěšnosti jednotlivých variant modelů. Zkratky „sg, ns“ znamenají architektura Skip-gram a použití negativního vzorkování. Pokud není uvedeno jinak, modely byly natrénovány na korpusu CWC-2011. Výjimka je u modelu BERT, který byl stažený předtrénovaný. BERT byl zpracovávaný pomocí knihoven od Google Research[2] a Hugging Face[3]. Testované varianty byly $\text{BERT}_{\text{BASE}}$ *cased* a *uncased*. Výsledky různých variant byly téměř totožné. Při používání modelu BERT jsem zkoušel použít embeddings vektory různých vrstev po průchodu sítí, kde vstup byl samostatné slovo ve větě. Dané slovo

[2] https://github.com/google-research/bert
[3] https://github.com/huggingface/transformers

by správně mělo být v kontextu nějaké věty. V tomto případě by bylo potřeba spočítat dopředný průchod několika kandidátních vět, kde hodnoty embeddings vektorů budou závislé nejen na kontextu, ale i na pozici daného slova ve větě. Už takhle je počítání dopředného průchodu mnohem výpočetně náročnější, než pouhý výběr vektorové reprezentace z prediktivního modelu. Výběr a počítání dopředného průchodu několika vět by při vyhodnocování podobnosti slov v samotné hře bylo příliš časové náročné a u generování nápověd nereálné. Další věcí, která nejspíše měla vliv na použití embeddings vektorů je, že všechny vektory mají tendenci korelovat v prostoru podobným směrem. Nejmenší kosinová podobnost dvou slov v testovací sadě z intervalu <0,1> byla kolem hodnoty 0,6. Většina vyhodnocených dvojic měla podobnost kolem hodnoty 0,9. Dále jsem zkoušel využít hodnot z attention matic, kde vstupem byla věta obsahující nápovědu a všechna slova ve hře. Při počítání vybraných matic, které vypadaly, že by mohly nést nějakou užitečnou informaci, byla však úspěšnost velmi nízká.

Dále lze v tabulce vidět, že samostatně natrénovaný fastText má vyšší úspěšnost, něž fastText stažený z internetu, alespoň pro češtinu. Což ukazuje, že vhodně předzpracovaný korpus může mít znatelný vliv pro výslednou úspěšnost modelu, především u morfologicky bohatých jazyků. Na českém fastTextu staženého z internetu je vidět, že nebyl pořádně tokenizován. Model má ve slovníku slova jako „Auto“, „auto“, „auto.“, „auto-“. Dále byla testována varianta, kde česká slova byla překládána do angličtiny[4] a podobnost vyhodnocena na anglických modelech. Používání překladu se ukazuje jako nevhodné, protože i samotný stažený český fastText má o více než 6 % vyšší úspěšnost.

Další varianta byla dotrénovat fastText na novějším korpusu, ne z důvodu získání co nejlepší úspěšnosti, ale pro naučení aktuálnějších termínů a jejich sémantického kontextu. Očekával jsem, že úspěšnost po dotrénování bude víceméně stejná, případně maximálně klesne o zhruba 2 %. Nicméně po dvou epochách trénování klesla podobnost na testovací sadě až o 5 %.

Pro model AdaGram bylo otestováno několik variant vyhodnocování podobností různých významů slov. Nejvyšší kombinace znamená, že se u dvojice slov vybrala taková kombinace jejich významů, která byla vyhodnocena jako nejvyšší. U vážené sumy se sečetly podobnosti všech kombinací významů slov, kde každá vyhodnocená podobnost byla vynásobená apriorními pravděpodobnostmi jednotlivých významů. Při vyhodnocování podobnosti slov v rámci kontextu bylo pro jedno slovo z dvojice použito jako kontext pro spočítání podmíněných pravděpodobností významů. Pro druhé slovo bylo použito totéž s první slovem jako kontext. Při testování v rámci kontextu jsou v tabulce uvedeny 2 varianty. U první se pro každé slovo použil jenom jeden význam, který byl v rámci kontextu vyhodnocen jako nejpravděpodobnější. Druhá varianta počítala totéž jako vážená suma, akorát se použily zmíněné podmíněné pravděpodobnosti místo apriorních. Při namátkovém testování nejpodobnějších slov jednotlivých významů u modelu AdaGram bylo vidět, že model umí dobře rozlišovat jednotlivé významy slov a k nim podobná slova. Přesto ale měl fastText pořád vyšší úspěšnost, což ukazuje, že pro Krycí jména je dostatečné jednotlivé významy slov průměrovat do jedné reprezentace.

I s novějšími přístupy stále platí, že nejúspěšnější metoda je počítání podobností slov na základě jejich společného výskytu (NPMI). Použitím artitmetického průměru hodnot NMPI a fastText se úspěšnost zvýší na 77,81 %. Při použití počtu výskytů z článků Wikipedie se dosáhlo celkového zlepšení na 77,96 %. Použití Wikipedie zlepšilo úspěšnost u 2,5% tahů a zhoršilo u 0,74% tahů. Při použití Wikipedie, slovníku spisovné češtiny a úvahy nad

[4] https://github.com/lushan88a/google_trans_new

vlastní nápovědou implementovanou v bakalářské práci [11] se celková úspěšnost dostala na 78,78 %.

Testovací sada však obsahuje „pouze“ 680 tahů, což v porovnání celkovým počtem slov v českém jazyce (s výjimkou předložek, spojek…) nemusí adekvátně pokrývat velký počet různých situací. Další aspekt vyhodnocování sémantické podobnosti slov je, že podobnost dvou slov nemusí být jednotlivými lidmi brána stejně a neexistuje ucelený pravdivý výrok. Každý člověk může mít jiný pohled a vyhodnotit slovní asociace podle sebe. Navíc při hře si člověk ne vždy důkladně projde všechna herní slova a někdy označí první slovo, kterého si všimne, že by mohlo být cílené. To samé platí při dávání nápověd spoluhráči.

7.3 Úspěšnost vícejazyčné podpory

Úspěšnost systému v rámci různých jazyků byla vyhodnocena na nasbíraných datech ze záznamů webové služby. Záznamy se ukládají po jednotlivých tazích, kde se zaznamená nápověda, případně zamýšlená slova, pokud se zvolí, slova označená v daném tahu a týmové příslušnosti jednotlivých herních slov. Při vyhodnocení systému v roli člena operativy byly jednotlivé tahy vyhodnoceny modely znovu, aby bylo možné seřadit slova podle vyhodnocených posloupností a určit, jak daleko v pořadí bylo všech n požadovaných slov ze správného týmu, kde n bylo číslo nápovědy. Nasbíraná data byla pro jazyky čeština, angličtina, slovenština a němčina. Dostatečné množství dat bylo nasbíráno pro češtinu, u ostatních jazyků mohou být výsledky spíše orientační.

Tabulka 7.2 obsahuje statistická data ze záznamů her, kde byl systém v roli člena operativy. Poslední řádky udávají, kolik slov by celkem systém označil, aby našel požadovaný počet správných slov na příchozí nápovědu. Při srovnání češtiny s ostatními jazyky lze vidět, že systém měl podobnou úspěšnost při hádání slov. Mezi standardní verzí a verzí Duet je vidět, že ve verzi Duet bylo častěji označováno neutrální slovo. Je to především tím, že v Duetu nejsou slova nepřátelského týmu, místo nichž je o 5 neutrálních slov více. Navíc ve verzi Duet lze v jednom tahu označit libovolný počet slov, což umožňuje použít agresivnější nápovědu, která může mít tendenci je sílit a případná správná slova lze jednoduše „dohádat“ v dalších tazích. Dále lze vidět, že nebyl označen prakticky žádný nájemný vrah ani v jedné verzi. Z toho vyplývá, že se uživatelé opravdu snažili slovům představující nájemného vraha vyhýbat.

Tabulka 7.3 skoro stejné statistické údaje, kde však systém byl v roli hlavního špióna a uživatelé hádali. V porovnání s nápovědami uživatelů lze vidět, že generované nápovědy jsou v průměru o něco agresivnější, neboli cílí více slov. Úspěšnosti hádání slov uživateli je o něco nižší u standardní verze, než při prohozených rolích. Avšak ve verzi Duet bylo hádání uživatelů úspěšnější, než hádání systému na uživatelem zadané nápovědy. Opět porovnání mezi jazyky je téměř srovnatelné. Avšak pro reálnější odhad by bylo vhodné mít více dat.

Při zběžné revizi záznamů bylo vidět, že při používání webové služby se nemusí uživatelé pořádně soustředit a neprojdou si všechna slova ve hře, což v některých případech vedlo k nevhodně zvolené nápovědě nebo označení méně podobných slov.

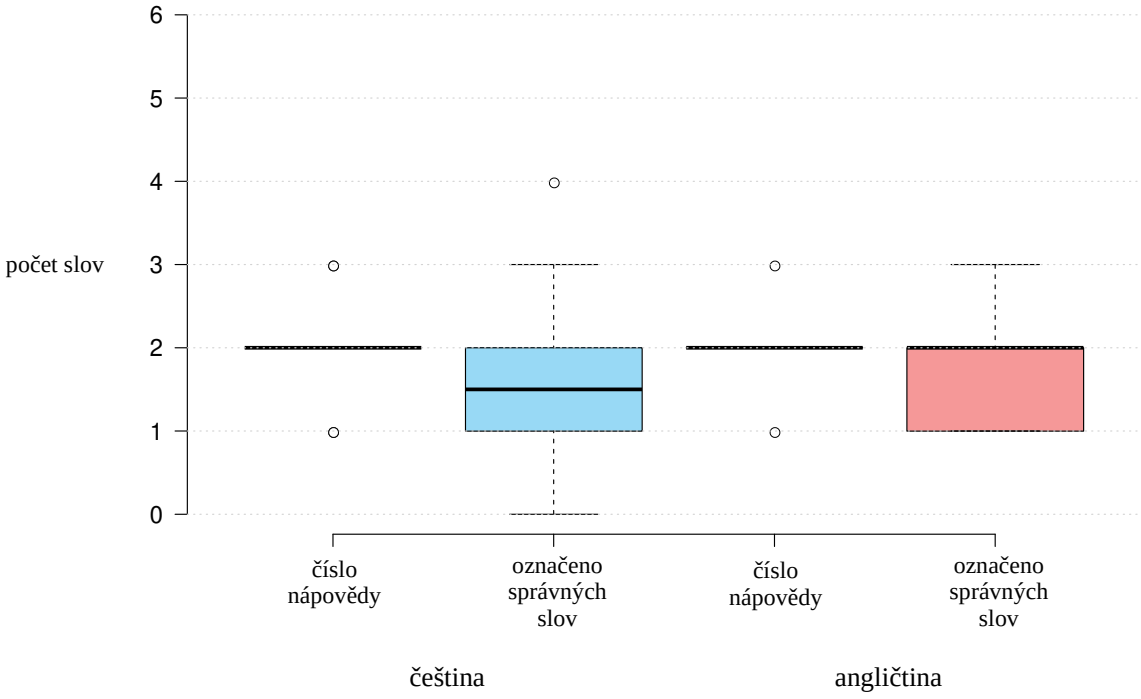
		čeština		**angličtina**		**slovenština**		**němčina**	
		standard	Duet	standard	Duet	standard	Duet	standard	Duet
počet standardních nápověd		55	125	10	32	1	9	2	-
průměrný počet cílených slov		2,036	2,152	2,0	2,281	1,0	2,111	2,0	-
průměrný počet správně označených slov		1,563	1,256	1,7	1,406	1,0	1,667	1,5	-
průměrná úspěšnost označení slov správného týmu		73,63 %	55,53 %	83,33 %	57,45 %	100 %	66,30 %	75 %	-
počet kol kde byl označen:	**neutrální slovo**	15	82	1	24	0	5	1	-
	nepřátelský tým	9	N/A	2	N/A	0	N/A	0	N/A
	nájemný vrah	1	0	1	0	0	0	0	-
počet nápověd typu nekonečno		0	N/A	0	N/A	0	N/A	0	N/A
průměrný počet správně označených slov		-	N/A	-	N/A	-	N/A	-	N/A
průměrný počet potřebných slov k označení na nápovědu číslo:	**1**	1,125	1,0	1,0	1,667	1,0	1,0	-	-
	2	2,946	3,84	3,375	3,0	-	3,167	3,0	-
	3	3,8	5,227	3,0	5,833	-	7,0	-	-
	4	-	8,667	-	7,333	-	-	-	-

Tabulka 7.2: Tabulka vyhodnocení záznamů webové služby, kde počítač zaujímal roli člena operativy a uživatel hlavního špióna.

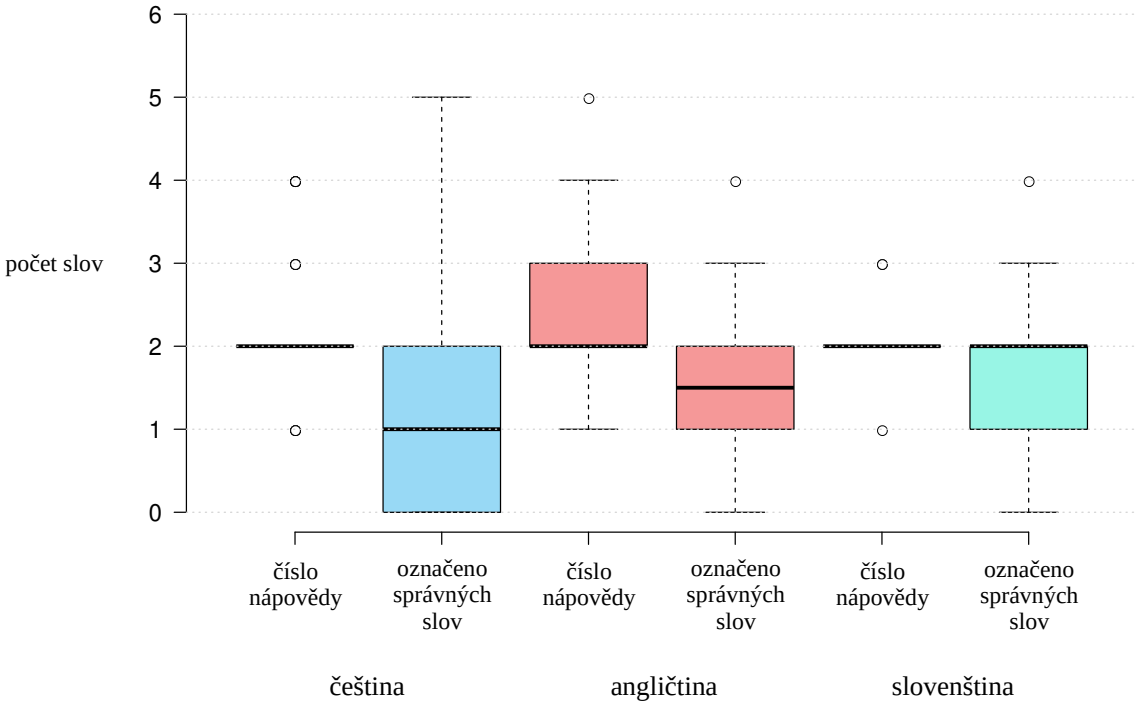
		čeština		**angličtina**		**slovenština**		**němčina**	
		standard	Duet	standard	Duet	standard	Duet	standard	Duet
počet standardních nápověd		1131	1656	68	271	43	14	6	-
průměrný počet cílených slov		2,39	2,4	2,544	2,273	2,535	2,143	2,0	-
průměrný počet správně označených slov		1,33	1,333	1,338	1,207	1,14	1,571	0,75	-
průměrná úspěšnost označení slov správného týmu		61,87 %	57,73 %	60,56 %	60,32 %	55,88 %	77,38 %	52,78 %	-
počet kol kde byl označen:	**neutrální slovo**	118	951	17	126	8	6	1	-
	nepřátelský tým	54	N/A	19	N/A	16	N/A	2	N/A
	nájemný vrah	16	166	1	22	1	0	1	-
počet nápověd typu nekonečno		214	N/A	9	N/A	8	N/A	2	N/A
průměrný počet správně označených slov		1,523	N/A	1,333	N/A	1,625	N/A	2,5	N/A

Tabulka 7.3: Tabulka vyhodnocení záznamů webové služby, kde počítač zaujímal roli hlavního špióna a uživatel člena operativy.

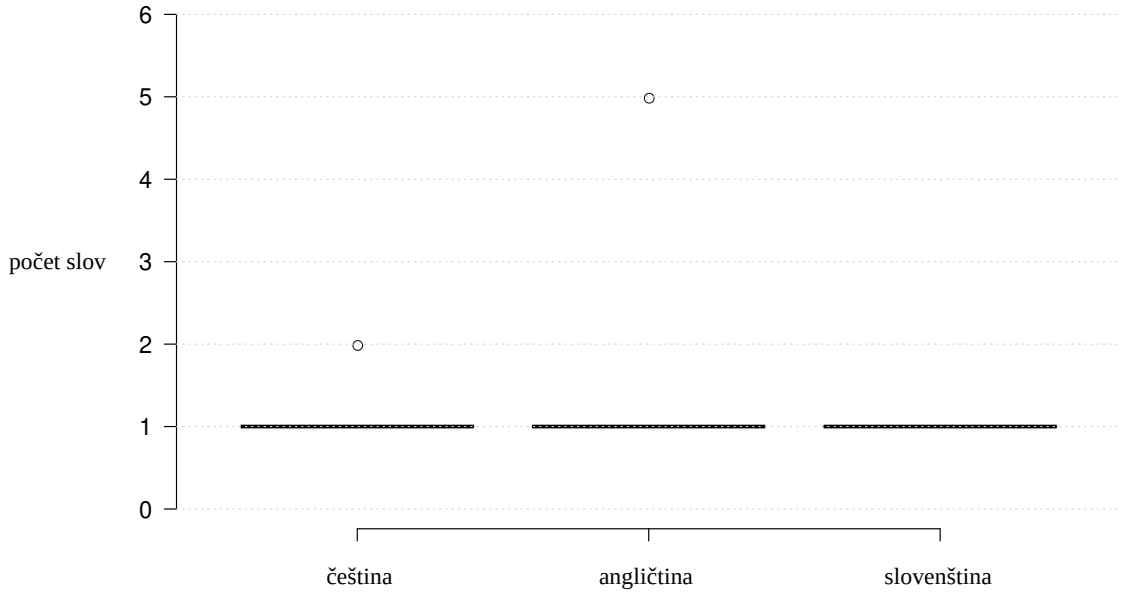
Obrázky 7.1 a 7.2 ukazují četnosti nápověd cílící různý počet slov a četnosti správně označených slov, kde systém byl v roli clena operativy ve standardní verzi a ve verzi Duet pro jazyky, kde bylo nasbíráno alespoň minimální množství dat. Obrázky 7.3, 7.4 a 7.5 ukazují četnost počtu slov, kolik by systém označil slov v roli člena operativy, dokud by neoznačil požadovaný počet správných slov na nápovědy cílící 1, 2 a 3 slova. Poslední dva obrázky 7.6 a 7.7 ukazují stejné četnosti jako první dva, s tím rozdílem, že zde jsou role prohozené a systém byl v roli hlavního špióna a uživatelé hádali slova. V grafech mohou hodnoty počtu označených správných slov být vyšší než čísla nápověd. Během tahu může hráč označit jedno slovo (v Duetu více slov) navíc, aby zkusil „dohádat“ předchozí nepovedené tahy.



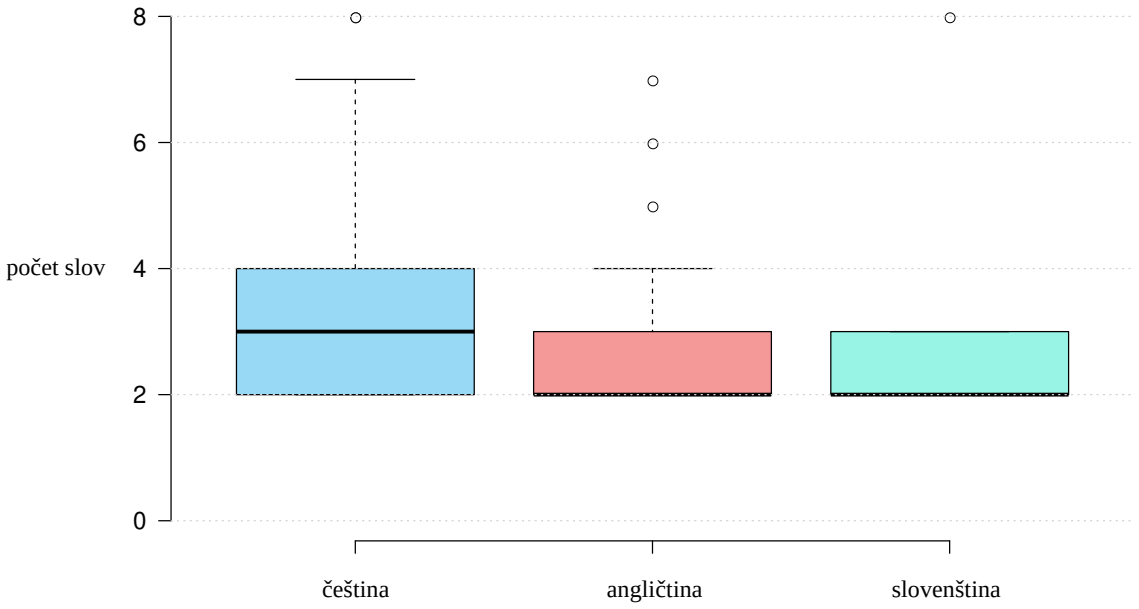
Obrázek 7.1: Krabicový graf znázorňující četnost počtu cílených slov a počtu správně označených slov kde systém byl v roli člena operativy standardní verze.



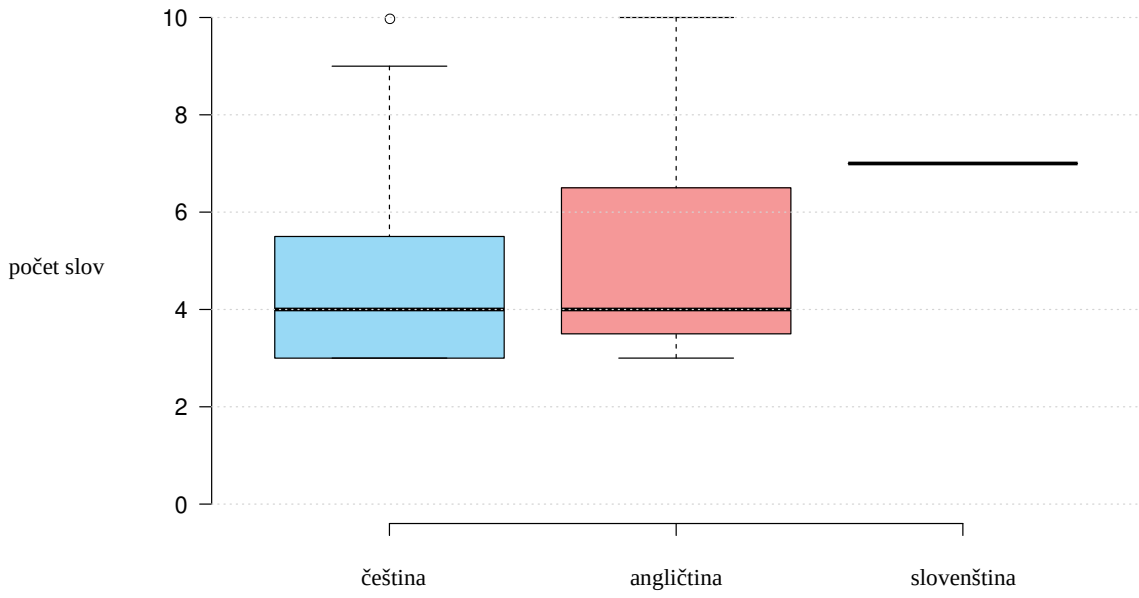
Obrázek 7.2: Krabicový graf znázorňující četnost počtu cílených slov a počtu správně označených slov, kde systém byl v roli člena operativy Duet verze.



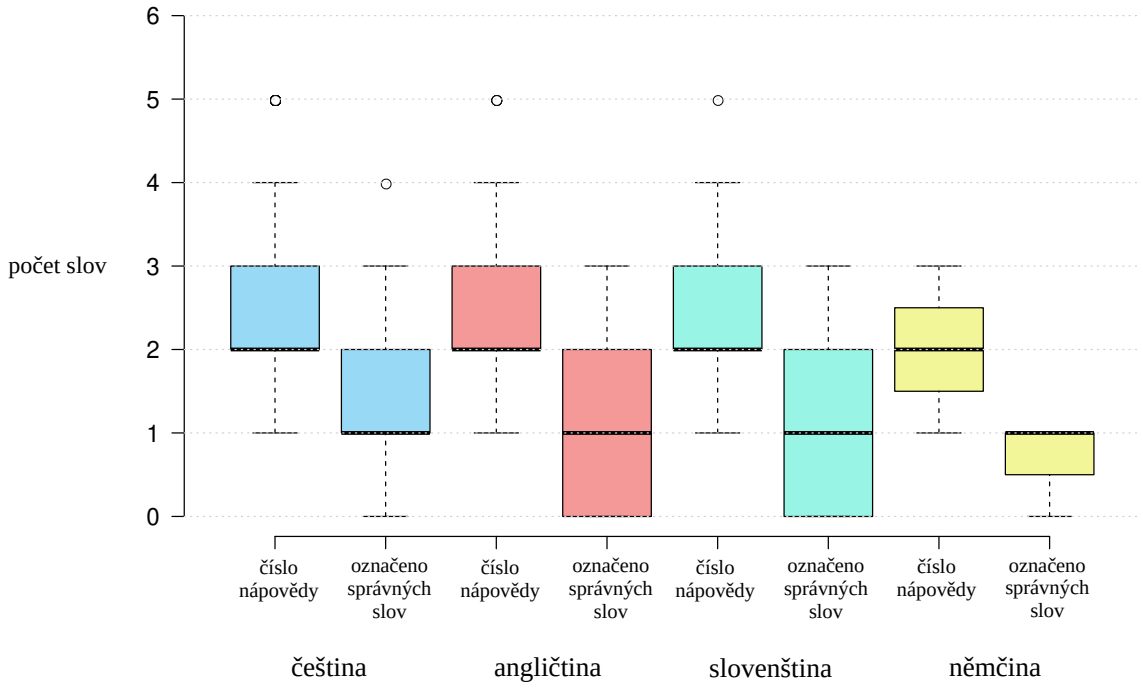
Obrázek 7.3: Krabicový graf znázorňující četnost, kolik slov by systém celkem označil, aby uhodl všechna správná slova na nápovědy, které cílí 1 slovo. Data jsou sloučená z obou verzí.



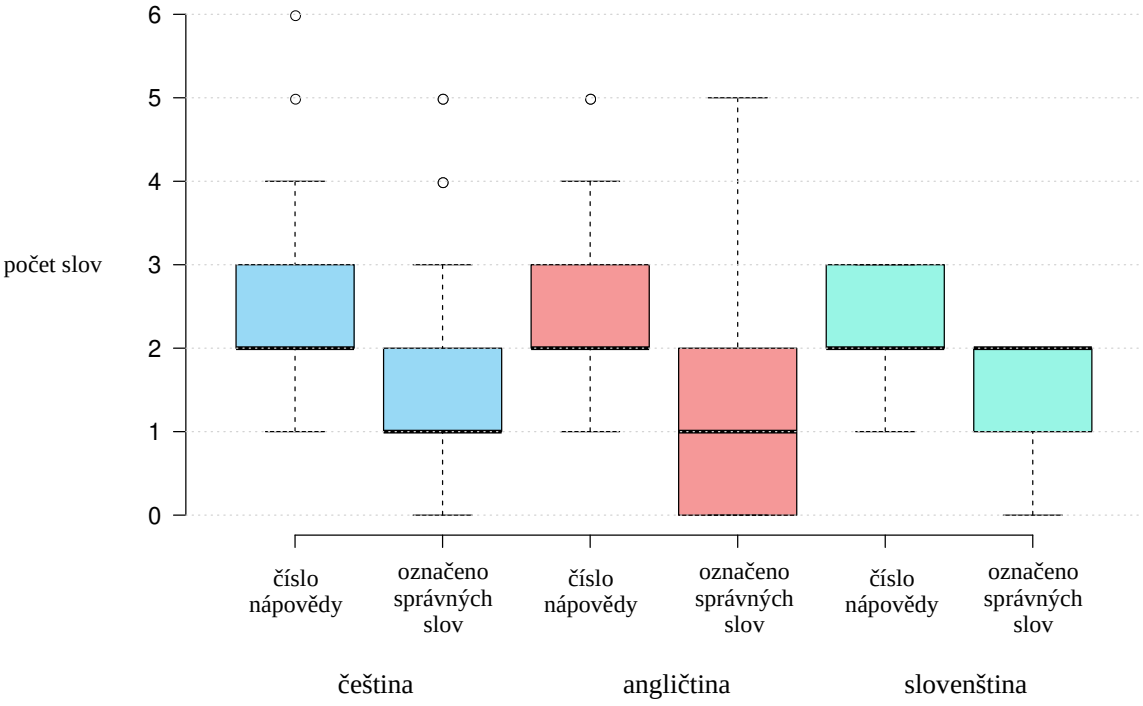
Obrázek 7.4: Krabicový graf znázorňující četnost, kolik slov by systém celkem označil, aby uhodl všechna správná slova na nápovědy, které cílí 2 slova. Data jsou sloučená z obou verzí.



Obrázek 7.5: Krabicový graf znázorňující četnost, kolik slov by systém celkem označil, aby uhodl všechna správná slova na nápovědy, které cílí 3 slova. Data jsou sloučená z obou verzí.



Obrázek 7.6: Krabicový graf znázorňující četnost počtu cílených slov a počtu správně označených slov, kde systém byl v roli hlavního špióna standardní verze.



Obrázek 7.7: Krabicový graf znázorňující četnost počtu cílených slov a počtu správně označených slov, kde systém byl v roli hlavního špióna Duet verze.

Kapitola 8

Závěr

Cílem překládané práce bylo rozšířit systém implementovaný v jazyce Python, který je schopný zastoupit hráče ve hře Krycí jména v rolích hádajícího hráče, napovídajícího hráče a hráče ve verzi Duet. Rozšíření spočívala v automatizaci podpory systému pro nové jazyky v podobě použití nástrojů podporujících celou řadu jazyků a jednoduché integrace do systému. Dále bylo cílem prozkoumat nové metody pro určování sémantické podobnosti slov, rozšíření sběru dat webové služby[1] a snaha vylepšit implementované herní role.

V rámci překládané práce byly vyhodnoceny novější přístupy pro určování sémantické podobnosti slov pro hru Krycí jména. Dále byly představeny nástroje pro analýzu a zpracování přirozeného jazyka podporující velké množství jazyků, které byly využity pro přidání podpory nových jazyků do implementovaného systému. Výsledný systém podporuje hraní Krycích jmen v 36 jazycích tvořených 8 různými abecedami. Při optimalizaci herních rolí byl však kladen důraz především na češtinu, která navíc využívá 2 speciální slovníky pro zlepšení určování slovních asociací. Všechny podporované jazyky využívají prediktivní model fastText pro určování sémantické podobnosti slov, případně data z Wikipedie při hádání slov. Jazyky čeština, slovenština a angličtina navíc využívají matici spoluvýskytů slov, která je použita pro vyhodnocení Pointwise Mutual Information.

V českém jazyce byla dosažena úspěšnost 77,96 % při hádání slov na testovací sadě. Nabíraná data pomocí webové služby ukazují, že v různých jazycích se průměrná úspěšnost systému pohybovala kolem dosažené úspěšnosti v češtině.

Jelikož byla většina nových jazyků přidávána automatizovaně se stejnými parametry a použitím pouze předtrénovaného modelu fastText, nabízí se možnost jako rozšíření důkladně zpracovat jednotlivé jazyky s použitím i matice spoluvýskytů, k čemuž by však bylo vhodné mít hlubší znalosti daného jazyka. Pokud by webová služba měla být vystavena výrazně vyšší zátěži, bylo by vhodné ji distribuovat mezi několika servery, kde by každý server měl na starosti část podporovaných jazyků. Případně by šlo využít několik GPU pro práci s vektorovými reprezentacemi slov, kde by se dobře paralelizovalo počítání kandidátních nápověd. Právě větší používání webové služby by vedlo k získání mnohonásobně většímu počtu záznamů her, které by potom mohly být využity ke specifickému trénování modelu pro určování slovních asociací.

[1] http://athena3.fit.vutbr.cz:8086/

Literatura

[1] *Internetová jazyková příručka* [online]. Praha: Ústav pro jazyk český AV ČR, v. v. i. 2008-2020 cit. [2020-12-07]. Dostupné z: https://prirucka.ujc.cas.cz/.

[2] Athiwaratkun, B., Wilson, A. G. a Anandkumar, A. Probabilistic FastText for Multi-Sense Word Embeddings. *CoRR*. 2018, abs/1806.02901. Dostupné z: http://arxiv.org/abs/1806.02901.

[3] Bartunov, S., Kondrashkin, D., Osokin, A. a Vetrov, D. P. Breaking Sticks and Ambiguities with Adaptive Skip-gram. *CoRR*. 2015, abs/1502.07257. Dostupné z: http://arxiv.org/abs/1502.07257.

[4] Bojanowski, P., Grave, E., Joulin, A. a Mikolov, T. Enriching Word Vectors with Subword Information. *CoRR*. 2016, abs/1607.04606. Dostupné z: http://arxiv.org/abs/1607.04606.

[5] Bouma, G. Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proceedings of the Biennial GSCL Conference 2009*. Leden 2009.

[6] Dean, T. M. andIlya Sutskever andKai Chen andGreg Corrado andJeffrey. Distributed Representations of Words and Phrases and their Compositionality. *CoRR*. 2013, abs/1310.4546. Dostupné z: http://arxiv.org/abs/1310.4546.

[7] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. a Harshman, R. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*. 1990, sv. 41, č. 6, s. 391–407.

[8] Devlin, J., Chang, M., Lee, K. a Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*. 2018, abs/1810.04805. Dostupné z: http://arxiv.org/abs/1810.04805.

[9] Grave, E., Bojanowski, P., Gupta, P., Joulin, A. a Mikolov, T. Learning Word Vectors for 157 Languages. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[10] HOŠTÁK, V. S. *Shlukování slov podle významu*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D.

[11] Jareš, P. *Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D. Dostupné z: https://www.fit.vut.cz/study/thesis/21503/.

[12] Jivani, A. A Comparative Study of Stemming Algorithms. *Int. J. Comp. Tech. Appl.* Listopad 2011, sv. 2, s. 1930–1938.

[13] Kanerva, Jenna, Ginter, Filip, Salakoski et al. Universal Lemmatizer: A Sequence to Sequence Model for Lemmatizing Universal Dependencies Treebanks. *Natural Language Engineering*. Cambridge University Press. 2020, s. 1–30. DOI: 10.1017/S1351324920000224. Dostupné z: http://dx.doi.org/10.1017/S1351324920000224.

[14] Kanerva, J., Ginter, F., Miekka, N., Leino, A. a Salakoski, T. Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, 2018.

[15] Karásek, J., Morský, O., Šanda, P. a Burget, R. Strojové učení základem pro hybridní lemmatizační algoritmus. *Elektrorevue*. 2012, sv. 14. ISSN 1213-1539.

[16] Kral, P. a Lenc, L. Czech Text Document Corpus v 2.0. In: chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S. et al., ed. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Paris, France: European Language Resources Association (ELRA), May 2018. ISBN 979-10-95546-00-9.

[17] Luong, M., Pham, H. a Manning, C. D. Effective Approaches to Attention-based Neural Machine Translation. *CoRR*. 2015, abs/1508.04025. Dostupné z: http://arxiv.org/abs/1508.04025.

[18] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. a Joulin, A. Advances in Pre-Training Distributed Word Representations. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[19] Qi, P., Zhang, Y., Zhang, Y., Bolton, J. a Manning, C. D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2020. Dostupné z: https://nlp.stanford.edu/pubs/qi2020stanza.pdf.

[20] Řehůřek, R. a Sojka, P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, s. 45–50.

[21] See, A. a Lamm, M. *Machine Translation, Sequence-to-sequence and Attention* [online]. Standford University, 2019 cit. [2021-01-03]. Dostupné z: http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf.

[22] Spoustová, J. a Spousta, M. *CWC2011*. 2012. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. Dostupné z: http://hdl.handle.net/11858/00-097C-0000-0006-B847-6.

[23] Straka, M. a Straková, J. *Czech Models (MorfFlex CZ 161115 + PDT 3.0) for MorphoDiTa 161115*. 2016. LINDAT/CLARIN digital library at the Institute of Formal

and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. Dostupné z: http://hdl.handle.net/11234/1-1836.

[24] Straková, J., Straka, M. a Hajič, J. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, s. 13–18. Dostupné z: http://www.aclweb.org/anthology/P/P14/P14-5003.pdf.

[25] Sutskever, I., Vinyals, O. a Le, Q. V. Sequence to Sequence Learning with Neural Networks. *CoRR*. 2014, abs/1409.3215. Dostupné z: http://arxiv.org/abs/1409.3215.

[26] Vaswani, A., Shazeer, N., Parmar, N., N. Gomez, J. U. andLlion Jones andAidan, Kaiser, L. et al. Attention Is All You Need. *CoRR*. 2017, abs/1706.03762. Dostupné z: http://arxiv.org/abs/1706.03762.

[27] Zeman, D. *Lingvistická terminologie* [online]. 1999 cit. [2020-12-06]. Dostupné z: http://ufal.mff.cuni.cz/~zeman/vyuka/podklady/pzpj02-roviny.pdf.

[28] Čermák, F. *Jazyk a jazykověda*. 4. vyd. Karolinum, 2011. ISBN 978-80-246-2360-3.

[29] Černý, J. *Úvod do studia jazyka*. 1. vyd. Rubico, 1998. ISBN 80-85839-24-5.

Příloha A

Obsah paměťového média

DVD nosič obsahuje:

- Diplomovou práci ve formátu pdf
- Zdrojové soubory LaTeX – adresář: /dp_src
- Zdrojové soubory implementovaných modulů – adresář: /src
- Dokumentaci zdrojových kódů – adresář: /src/docs
- Virtuální prostředí s Python 3.6.7 a potřebnými knihovnami – adresář: /src/xjares00env
- Zdrojové soubory webové služby – adresář: /src/web
- Vytvořená statická data využívaná při běhu systému – adresář: /src/data
  Kvůli paměťové náročnosti nebyly přidány sémantické modely. Ty jsou dostupné v projektovém adresáři na školním serveru VUT FIT.
- Skripty a data spojená s vyhodnocením úspěšnosti – adresář: /src/evaluation
- Vytvořený plakát ve formátu pdf