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

POČÍTAČ JAKO INTELIGENTNÍ SPOLUHRÁČ VE SLOVNĚ-ASOCIAČNÍ HŘE KRYCÍ JMÉNA

COMPUTER AS AN INTELLIGENT PARTNER IN THE WORD-ASSOCIATION GAME CODENAMES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **KATEŘINA CHOVANCOVÁ**
AUTHOR

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

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

154711

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Studentka: **Chovancová Kateřina**
Program: Informační technologie
Název: **Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména**
Kategorie: Zpracování řeči a přirozeného jazyka
Akademický rok: 2023/24

Zadání:

1. Seznamte se s principem hry Krycí jména a se strategiemi hraní, prostudujte oblast automatického určování sémantické příbuznosti slov se zaměřením na vazby mezi skupinami slov.
2. Zpracujte nově shromážděná data z her, identifikujte největší problémy používaných modelů, případně strategie hraní.
3. Navrhněte a implementujte systém, který zlepší kvalitu hraní v roli zadavatele asociovaných slov (hlavního agenta) i hadače (člena kontaktního týmu) oproti systémům vytvořeným v předchozích pracech.
4. Zaměřte se na zvýšení jazykové nezávislosti řešení a možnost snadné aktualizace modelů.
5. Vytvořte stručný plakát prezentující vytvořenou práci a její výsledky

Literatura:

- dle doporučení vedoucího

Při obhajobě semestrální části projektu je požadováno:

- funkční prototyp řešení

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, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 31.7.2024
Datum schválení: 21.12.2023

Abstrakt

Tato práce rozšiřuje systém pro určování sémantické podobnosti slov a vytvářením slovních asociací. K tomu je v práci využit prediktivní model fastText v kombinaci s metodou DETECT, a model založený na výpočtu Pointwise Mutual Information. Metoda DETECT využívá model Dict2vec, který je trénován na slovníkových definicích pojmů. Výsledný systém je schopen zastoupit hráče při hraní slovně asociační hry Krycí jména, a to jak na pozici člena operativy, tak v roli hlavního špiona. Zároveň byl v rámci práce vyvinut nástroj pro vytváření testů sémantické kontroly a znalosti, který pracuje se slovníkem českých synonym a slouží k výpočtu hodnoty TDS a určení četnosti výskytu slov. Poslední část práce se věnuje anlýze dat ze studie STST II., ve které bylo zkoumáno vzájemné myšlenkové napojení hráčů při hraní komunikační hry.

Abstract

This thesis extends a system for determining semantic similarity between words and creating word associations. For this purpose, the work uses the fastText predictive model in combination with the DETECT method, and model based on Pointwise Mutual Information calculation. DETECT method uses Dict2vec model, which is trained on dictionary definitions of terms. The resulting system is capable of replacing a player in the word association game Codenames, both as a member of an operative and as a spy. Furthermore, a tool for creating semantic control and knowledge tests was developed within the thesis, which uses a dictionary of Czech synonyms and is used to calculate the TDS value and determine the frequency of occurrence of words. The last part of the thesis is devoted to the analysis of data from the STST II. study, in which players' inter-thought connections while playing a communication game were investigated.

Klíčová slova

Krycí jména, slovní asociace, sémantická podobnost, word2vec, fastText, dict2vec, DETECT, sémantická znalost, sémantická kontrola, synonyma, TDS, STST II., komunikační hra, myšlenkové napojení

Keywords

Codenames, word association, semantic similarity, word2vec, fastText, dict2vec, DETECT, semantic knowledge, semantic control, synonyms, TDS, STST II., communication game, thought connection

Citace

CHOVANCOVÁ, Kateřina. *Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Pavel Smrž, Ph.D.

Počítač jako inteligentní spoluhráč ve slovně-asociační hře Krycí jména

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....
Kateřina Chovancová
30. července 2024

Poděkování

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

Obsah

Seznam obrázků

Kapitola 1

Úvod

Zpracování přirozeného jazyka (NLP, z anglického natural language processing) je soubor různých počítačových technik pro autimatickou analýzu a reprezentaci lidského jazyka [6]. Jeho cílem je umožnit počítačům porozumět lidské řeči, umožnit komunikaci s počítačem v přirozeném jazyce. Techniky NLP mají široké využití, např. při vyhledávání informací, kalsifikaci textu, automatickém překladu, generování textu atd. Hra Krycí jména je častým předmětem studií zabývajících se zkoumáním jazyka. Při hře Krycí jména vytvářejí hráči asociace se slovy tak, aby jim co možná nejlépe porozuměli jejich spoluhráči. Tato hra je tedy vhodným modelem pro zkoumání sémantické podobnosti slov [7].

Tato práce vychází z předešlých prací zabývajících se vytvořením systému pro zastoupení hráče při hraní Krycích jmen. Jedná se zejména o práci [11], která je v textu odkazována jako „předchozí práce“, a její rozšíření [12]. Cílem této práce je identifikovat problémy předešlého systému a navrhnout a implementovat potenciální řešení. Přestože předešlá práce implementuje systém pro hraní ve velkém množství jazyků, tato práce se věnuje pouze hraní Krycích jmen v češtině.

Pro vyhodnocené sémantické podobnosti slov je v roli člena operativy využit model založený na výpočtu Normalized Pointwise Mutual Information v kombinaci s prediktivním modelem fastText. Úkolem člena operativy je hádat na základě přijaté asociace odpovídající slova. Úkolem hlaního špiona je vytvářet vhodné jednoslovné asociace tak. V roli hlavního špiona je pro vyhodnocení sémantické podobnosti využit model fastText v kombinaci s metodou DETECT, která využívá model Dict2vec. Výsledný systém využívá webovou službu[1] z předešlé práce, která poskytuje uživatelské rozhraní pro hraní se systémem.

Druhá část této práce se zabývá vytvořením nástroje, který umožní vytvoření testovací sady otázek pro studii zkoumající sémantickou znalost a kontrolu. K tomu slouží vytvořená webová služba[2], která umožňuje především určování sémantické podobnosti slov a určování synonym. Systém využívá model fastText a slovník českých synonym.

Poslední část této práce se zabývá analýzou dat získaných v rímci studie STST II. Tato studie se zabývala měřením mozkové aktivity hráčů, kteří společně hráli komunikační slovně-asociační hru. Výsledkem analýzy je kvantifikace vzájemného myšlenkového napojení hráčů na základě sémantické podobnosti jejich odpovědí. Tato data mohou být výzkumníky následně porovnána s daty získanými měřením mozkové aktivity hráčů. Cílem je analyzovat, zda existují specifické vzorce mozkové aktivity, které odpovídají sémantické podobnosti při vzájemné komunikaci.

[1] Systém pro hraní Krycích jmen http://athena7.fit.vutbr.cz:8087
[2] Nástroj pro tvorbu testů sémantické kontroly a znalosti http://athena7.fit.vutbr.cz:8086

Kapitola 2

Hra Krycí jména

Historie využití umělé inteligence ve hrách sahá do počátků vývoje samotné umělé inteligence. Ještě dříve, než začala být umělá inteligence považována za samostatný obor, první průkopníci vytvářeli programy pro hraní her. Cílem bylo zjistit, zda počítače dokážou řešit úlohy, u kterých se předpokládalo, že je pro jejich řešení vyžadována „inteligence“. Nejrannější vývoj umělé inteligence pro hraní her byl soustředěn na klasické deskové hry, jako jsou šachy a dáma. Tyto hry vycházejí z jednoduchých pravidel, a přesto vytvářejí velmi složité situace představující výzvu i pro nejlepší lidské hráče. Existovala domněnka, že takové hry nějakým způsobem zachycují podstatu myšlení [24].

Po dekády byly šachy považovány za „modelový organismus“, na kterém bylo testováno nespočet nových metod umělé inteligence. V roce 1997 software Deep Blue, vyvinutý společností IBM, zvítězil nad úřadujícím šachovým velmistrem Garrym Kasparovem. Dnes si již může každý stáhnout do svého notebooku veřejně dostupný software, který hraje lépe než jakýkoli lidský hráč.

Dalším úspěchem IBM po Deep Blue byl Watson, softwarový systém schopný odpovídat na otázky v přirozeném jazyce. V roce 2011 Watson vyhrál 1 milion dolarů v televizní hře Jeopardy! svým vítězstvám nad bývalými vítězi hry. V roce 2017 porazil algoritmus AlphaGo nejlepšího lidského hráče ve hře Go a tím se tato hra stala stala poslední velkou klasickou deskovou hrou (hranou lidmi), ve které počítače dosáhly nadlidského výkonu.

I přes dosažení nadlidského výkonu ve všech klasických stolních hrách, které jsou založeny na jednoduchých pravidlech, výměně tahů a plánování komplexní strategie, však stále existuje mnoho herních výzev pro umělou inteligenci. Jednou z takových výzev jsou slovní hry. Slovní hry nejsou tak přesně definované a omezené jako klasické deskové hry. Strategie ve slovních hrách je naopak definována tím, jak hráči vnímají vzájemná propojení slov. Slovní hra Krycí jména poskytuje příležitost zkoumat chápání vztahů mezi slovy [14].

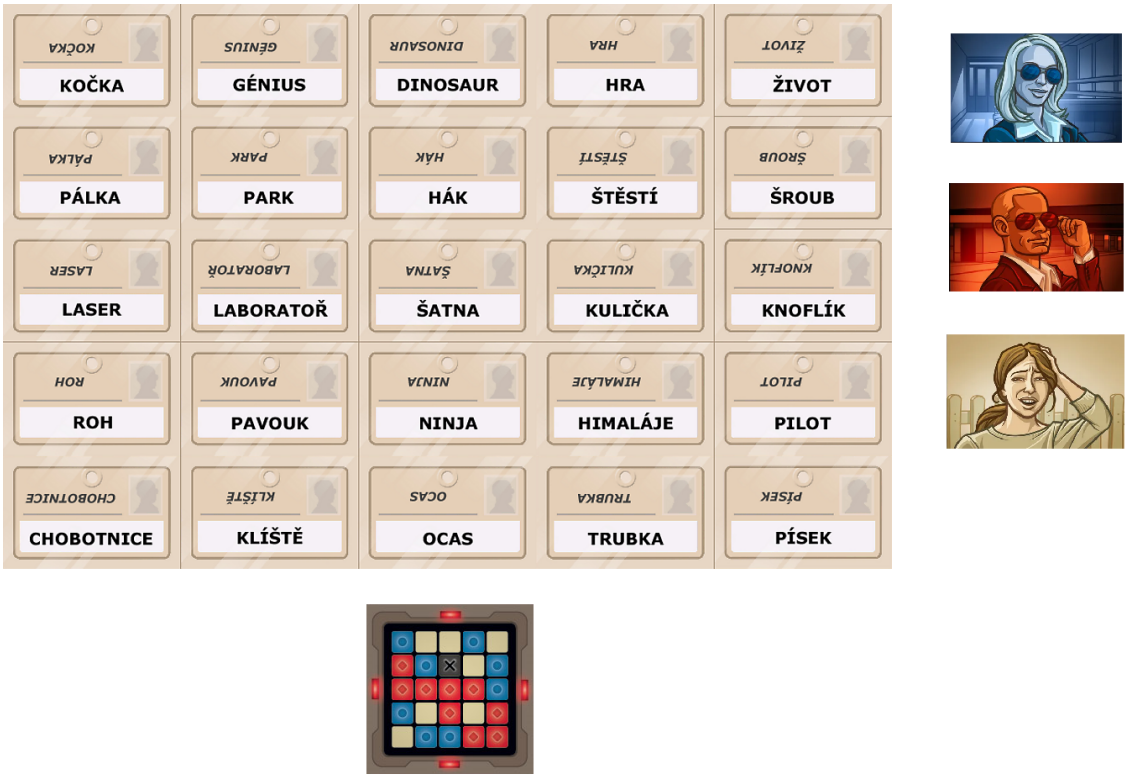
2.1 Krycí jména

Jak uvádí web [13], Krycí jména je slovně-asociační párty hra, ve které proti sobě soupeří dva týmy. Cílem hráčů je dříve než soupeři uhodnout, která slova (tajní agenti) patří jejich týmu. Každý z týmů má hlavního špiona a členy operativy. Hlavní špion se systémem správných nápověd snaží svůj tým dovést k cíli. Hru vytvořil český herní designér Vlaada Chvátil a v roce 2016 byla na prestižní soutěži deskových her Spiel des Jahres zvolena hrou roku.

Příprava hry

Jak uvádí oficiální pravidla hry [8], na začátku hry se hráči rozdělí do dvou přibližně stejně početných týmů. Týmy představují dva špionské útvary, červený a modrý, a každý tým zvolí svého hlavního špiona. Hlavní špioni se usadí na jednu stranu stolu a ostatní hráči, kteří představují členy operativy špionských útvarů, se posadí naproti nim. Poté je na stůl vyloženo 25 náhodně vybraných karet do mřížky o velikosti 5×5 karet, na každé kartě je jedno podstatné jméno. Tyto karty představují agenty.

Hlavní špióni náhodně vylosují jednu kartu klíče a umístí ji tak, aby její herní stranu viděli pouze oni. Klíč odpovídá mřížce slov rozložené na stole. Modrá pole odpovídají slovům, jež musí uhodnout modrý tým (modří agenti), červená pole má za úkol odhalit červený tým (červení agenti). Béžová pole představují náhodné kolemjdoucí a černé pole je nájemný vrah. Karta klíče také určuje, který tým bude hádat v prvním kole. Začínající tým musí uhádnout 9 slov (agentů), soupeři 8.



Obrázek 2.1: Hrací plocha před začátkem hry Krycí jména. Převzato[1] a upraveno

Pravidla

V každém kole nejdříve hlavní špion operativě poskytne nápovědu. Jeho úkolem je zvolit nápovědu tak, aby se vztahovala k jednomu či více slovům jeho barvy. Hlavní špion při výběru nápovědy musí dodržet tato pravidla:

- Nápověda musí být jedno slovo.

[1] Obrázek převzat z webu videacesky.cz, dostupný online: i.imgur.com/pP8gRXr.png

- Nápověda se musí týkat významu slov.
- Nápověda nesmí mít společný kořen se slovy, která leží na stole.

Pokud hlavní špión dá neplatnou nápovědu, tah týmu okamžitě končí. Jako dodatečný postih označí hlavní špion soupeřova týmu jednoho ze svých agentů v barvě svého týmu. Pak pokračuje zadáním další nápovědy.

Spolu s nápovědou hlavní špion operativě sdělí číslo, které určuje počet slov, ke kterým se nápověda vztahuje. Krom přirozených čísel může hlavní špion použít speciální čísla 0 nebo „nekonečno“. Použití čísla 0 znamená, že žádné z hledaných slov nikterak nesouvisí s nápovědou. Pokud špion použije „nekonečno“, chce, aby operativa hádala co nejvíce slov. To se může hodit, pokud operačnímu týmu z předchozích kol zbývá mnoho neuhodnutých slov.

Hlavní špion nesmí krom jendoslovné nápovědy nijak dále napovídat operativě (verbálně ani neverbálně).

Členové operativy po obdržení nápovědy společně postupně označují slova (agenty), o kterých si myslí, že náleží jejich týmu.

- Pokud operativa označí slovo, které patří jejímu týmu, slovo je označeno jako agent daného týmu a tým smí pokračovat v hádání dalšího slova.
- Pokud operativa označí náhodného kolemjdoucího, slovo je označeno jako náhodný kolemjdoucí a tah tohoto týmu končí.
- Pokud operativa označí slovo, které patří soupeři, slovo je označeno jako agent soupeřícího týmu a tah tohoto týmu končí.
- Pokud operativa označí slovo, které představuje nájemného vraha, tým, který označil nájemného vraha, okamžitě prohrává.

Operativa vždy musí označit nejméně jedno slovo. Tým smí hádání kdykoli dobrovolně přerušit, ale obvykle se snaží uhodnout tolik slov, kolik určil hlavní špión. V každém kolemůže operativa hádat nanejvýš o 1 slovo víc. Výjimkou je, pokud hlavní špion spolu s nápovědou místo přirozeného čísla zvolil nulu nebo nekonečno. V takovém případě operativa smí hádat libovolný počet slov, dokud neudělá chybu.

Hra končí ve chvíli, kdy jsou označeni všichni agenti jednoho z týmů. Tento tým vítězí. Hra skončí předčasně, pokud operativa označí nájemného vraha. Tento tým prohrává.

2.2 Krycí jména předmětem výzkumu

Od vzniku hry Krycí jména v roce 2015 se zkoumáním této hry a vývojem systémů pro její hraní zabýval nespočet studií a různých vědeckých prací. Tato hra se hodí pro zkoumání vytváření slovních asociací a jiných aspektů zpracování přirozeného jazyka, stejně tak se v jejím kontextu dá zkoumat mezilidská komunikace a různé herní strategie.

Transformátory a velké jazykové modely

Modely založené na vektorových reprezentacích slov mohou mít problém při spolupráci s jinými typy modelů. Jedním z moderních přístupů při vytváření modelů k hraní Krycích jmen je využití velkých jazykových modelů (LLMs) a transformátorů. Jejich využitím se zabývá čím dál tím větší množství studií. Např. práce [10] se zabývala využitím modelu

GPT-2 pro hraní hry Krycí jména a jeho spoluprácí s jinými modely. Jejím závěrem bylo zhodnocení slibných výsledků pro další vývoj.

Práce [20] se zabývala využití modelu chat GPT-4 při hraní hry Krycí jména. Hlavním cílem Této studie bylo vytvořit systém, který bude schopný lépe spolupracovat s jinými modely. V této studii bylo vytvořeno několik různých modelů hlavního špiona, které využívaly různé techniky prompt engineeringu. Prompt engineering je obor zabývající se vytvářením a optimalizováním promptů tak, aby jazykové modely byly schopny vygenerovat požadované odpovědi.

Bylo implementováno 6 různých verzí hlavního špiona s různými instrukcemi při vytváření promptů.

- Základní model – nevyužívá žádnou specializovanou techniku. V rámci promptu předává všechna zbývající herní slova s jejich příslušností (tým, kolemjdoucí, vrah) a za úkol má vrátit 1 optimální nápovědu.
- Opatrný model – stejný, jako základní, ale vždy je instruován, aby vrátil nápovědu pro 1 slovo.
- Riskantní model – stejný, jako základní, ale má za cíl vždy cílit velké množství slov.

Další modely hlavního špiona využívají techniky prompt engineeringu:

- Chain of Thought (Řetězec myšlenek) – má za úkol vyřešit zadanou úlohu krok za krokem, vysvětlovat postupně asociace mezi slovy a jak si slova jsou a nejsou podobná.
- Self Refine (Sebezdokonalení) – tento model využívá sérii tří promptů. Nejdříve vygeneruje nápovědu stejným způsobem jako základní model, poté zhodnotí tuto nápovědu a nakonec zpětnou vazbu využije k vygenerování finální nápovědy.
- Solo performance – po přijetí základních instrukcí je vytvořeno několik „osobností“. Je vygenerována nápověda, poté je zhodnocená jednotlivými „osobnostmi“. Nápověda je upravovaná, dokud se všechny „osobnosti“ neshodnou na výsledku.

Žádný z implementovaných přístupů nakonec nedosáhl lepšího výsledku než základní model. Přesto studie dokazuje, že výhodou modelů využívajících velké jazykové modely oproti modelům využívajícím vektorové reprezentace slov je schopnost neustále pracovat s novými slovy bez dodatečného trénování.

Adaptivní model

Spolupráce a přizpůsobení se spoluhráčům je důležitou součástí úspěšné strategie při hraní Krycích jmen. Tímto aspektem se zabývala studie [2], která vytvořila adaptivního hráče nazvaného Adaptive Codenames Ensemble (ACE). ACE dokáže hrát v roli hlavního špiona i člena operativy. ACE využívala soubor různých modelů hráčů, takzvaných expertů, a úkolem ACE bylo pomocí zpětné vazby z jednotlivých herních tahů zlepšovat souhru s aktuálním spoluhráčem. ACE dosahoval výborných výsledků jak v roli hlavního špiona, tak v roli člena operativy. Jeho největší výhodou je schopnost se rychle adaptovat, přizpůsobit se spoluhráčům, zároveň je schopen využívat různé přístupy, modely a jazyky.

Strategický hierarchický model

Strategickými aspekty hry se zabývala mimo jiné studie [3]. V rámci ní byla podoba hry upravena tak, že místo slov hra pracuje s číselnými vektory. Tato verze hry byla nazvána Codenums a jinak funguje stejně jako Krycí jména. Umožňuje odstínit aspekty spojené se zpracováním jazyka, a tak zkoumat pouze herní strategii.

Tato studie popisuje hierarchický přístup k herní strategii. V rámci hierarchie existuje několik modelů hadačů (členů operativy), každý model představuje jinou úroveň strategického uvažování. Hadač na úrovni 0 je statický, tzn. že vybírá nápovědy pouze na základě aktuálního stavu hry a nebere v úvahu kontext předešlých kol. Model 1. úrovně využívá model na úrovni 0. Předvídá chování statického hadače a informace používá k optimalizaci svých tahů. Uchovává informace o předešlých tazích, které také využívá k optimalizaci rozhodnutí. Každá další úroveň modelu hráče využívá modely z nižších úrovní. Vyšší úrovně vždy používají komplexnější strategické uvažování než předešlé úrovně.

Modely na vyšších úrovních hierarchie vykazovaly lepší výsledky, byly schopny se lépe přizpůsobit změnám ve hře a vhodně upravovat svou strategii. To dokazuje, že využití herní strategie vede ke zlepšení výsledků modelů.

Kapitola 3

Jazykový korpus

Jazykový korpus je soubor autentických textů v určitém jazyce. Korpus je v elektronické podobě a jednotném formátu, aby jej bylo možné strojově zpracovávat. Korpus zobrazuje jazykové jevy v jejich přirozeném kontextu a maximálně reprezentuje daný jazyk, ideálně ve všech jeho oblastech [26].

Tradičně jsou korpusy pro lingvistický výzkum vytvářeny především z tištěných textů, jako jsou novinové články a knihy. S rozvojem World Wide Webu jako informačního zdroje jsou internetová data stále častěji používána jako tréninková data v oblasti zpracování přirozeného jazyka [17].

Stárnutí modelu

Stárnutí modelu je proces zhoršování kvality výsledků modelu v čase. Nastává u modelů, které jsou trénovány na dynamicky se měnících datech. Přirozený živý jazyk, tedy jazyk, který má rodilé mluvčí, se neustále mění a vyvíjí. Dle článku[27] do internetové databáze neologismů[1] ročně přibydou tisíce nových slov, z nichž některá zanikají a jiná začnou být používána v běžné komunikaci. Neologismy obvykle vznikají v souvislosti s aktuální událostí či problémem. Do databáze českých neologismů např. přibylo 2584 slov souvisejících se světovou pandemií choroby covid-19[2]. Jazyk se mění mnoha způsoby, kromě přibývání nových slov např. zanikají archaismy a mění se pravidla gramatiky. V oblasti zpracování přirozeného jazyka dochází k neustálému stárnutí modelů v souvislosti s měnící se povahou jazyka.

3.1 Předzpracování textu

Jak uvádí [21], předzpracování textu má za účel čištění dat. Zjednodušeně řečeno se jedná o převod textu do formátu, který je snadno srozumitelný, předvídatelný a analyzovatelný strojem pomocí algoritmů strojového učení. Vhodné předzpracování textu vede k rychlejšímu trénování a snížení paměťové náročnosti výsledného modelu.

[1] neologismus = nově utvořené nebo přejaté slovo, česky také novotvar
[2] Databáze Neomat dostupná z www.neologismy.cz

Segmentace vět

Segmentace vět je proces rozdělení textového korpusu na jednotlivé věty, aby další zpracování mohlo být prováděno nad jendotlivými větami. Segmentace se obvykle provádí v místě výskytu interpunkčních znamének pomocí tokenizéru.

Převod na malá písmena

Jedná se o jeden z nejjednodušších, ale přesto velmi účinných kroků předzpracování textu. Textový korpus obvykle obsahuje různé zkratky či slova psaná velkými písmeny. Pokud není text převeden jednotně na malá písmena, může dojít k tomu, že je více podob jednoho slova interpretováno jako více slov, např. slovo „ČESKO“ je považováno za jiné slovo než „Česko“. Proto je dobrou praxí v předzpracování textu převádění všech slov na malá písmena.

Tokenizace

Tokenizace je proces rozdělení textu na jednotlivé tokeny, kterými jsou jendotlivá slova, interpunkční znaménka či samostatné znaky. Hranice tokenu je obvykle v místě mezery či interpunkčního znaménka.

Odstranění bezvýznamových slov

Některá slova, často např. předložky a spojky, nejsou obvykle nositeli významu, proto je není nutné zpracovávat. Jejich vyfiltrování se provádí pomocí „stoplistu“, což je seznam slov, která mají být při zpracování textu přeskočena.

Odstranění interpunkce

Interpunkční znaménka vytvářejí šum při strojovém zpracování textu. k jejich odstraněné se obvykle používají regulární výrazy. Dalším způsobem rozlišení interpunkčních znaménem je použití taggeru, který označí takové znaky, a ty jsou pak při zpracování přeskočeny.

Stemming a lemmatizace

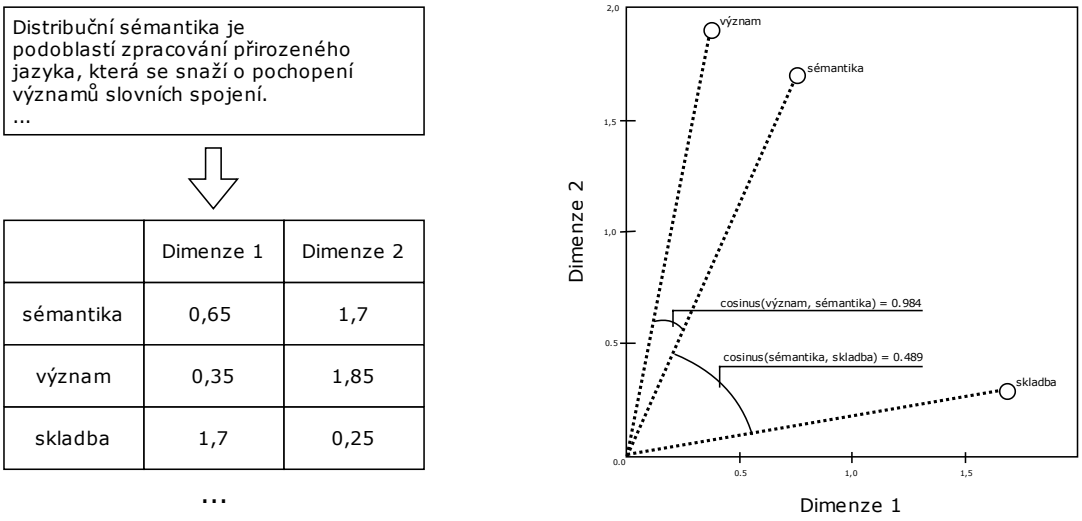
Stemming je technika zkracování slov pomocí odstranění prefixů a sufixů. Lemmatizace využívá slovníku a morfologické analýzy a snaží se slovo převést na jeho kořen. To vede ke zmenšení velikosti slovníku, neboť jsou různé formy jednoho slova převedeny na společný kořen. Obzvláště u morfologicky bohatých jazykyů, jako je čeština, je však určování kořenů slov náročné a nejednoznačné, a tak nemusí lemmatizace vždy poskytnout dobré výsledky.

Kapitola 4

Distribuční sémantika

Tato kapitola popisuje modely, které jsou použity k vyhodnocení podobnosti slov. V řešení je využit model zložený na počtu a modely fastText a Dict2vec, které oba vycházejí z prediktivního modelu Word2Vec. Využité modely jsou založeny na principu distribuční sémantiky.

Distribuční sémantika se zabývá modelováním sémantických vlastností a vztahů slov na základě jejich distribuce v kontextu. Jejím teoretickým základem je distribuční hypotéza, která říká, že „*Sémantická podobnost mezi dvěma lexémy je funkcí podobnosti jejich jazykových kontextů.*“[16] Ve své základní formě distribuční sémantika reprezentuje slova tak, že z velkého množství textových dat na vstupu pomocí abstrakce vytváří distribuční model, kde každé slovo je reprezentováno formou n-dimenzionálního vektoru. Souhrn všech vektorů slov v distribučním modelu tvoří n-dimenzionální prostor, ve kterém mohou být sémantické vztahy mezi jednotlivými slovy modelovány geometrickými vztahy mezi jejich vektory [5].



Obrázek 4.1: Princip distribuční sémantiky. Na vstupu je tetxový korpus, šipka představuje abstrakci, kdy je z textu vytvořen distribuční model. Slova jsou reprezentována jako body v n-dimenzionálním sémantickém prostoru. Geometrický vztah vektorů slov pak určuje sémantický vztah příslušných slov. Čím vyšší je hodnota cosinové podobnosti vektorů, tím sémanticky bližší si slova jsou.

4.1 Model založený na počtu

Model založený na počtu využívá matici spoluvýskytů, která obsahuje informaci o tom, jak často se jednotlivá slova vyskytují společně v kontextu. S využitím této matice je možné vyhodnotit sémantickou podobnost slov na základě jejich společného výskytu. Nevýhodou tohoto přístupu je paměťová náročnost, která se zvětšuje v závisloti na počtu unikátních slov ve slovníku modelu.

Normalized Pointwise Mutual Information

Pointwise Mutual Information (PMI) je hodnota, která udává míru asociace mezi dvěma slovy na základě jejich společného výskytu v porovnání s jejich jednotlivými výskyty.

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

kde $P(x \wedge y)$ je pravděpodobnost společného výskytu slov x a y, $P(x)$ je pravděpodobnost výskytu slova x a $P(y)$ je pravděpodobnost výskytu slova y. S použitím dat z matice spoluvýskytů lze upravit rovnici pro výpočet PMI takto:

$$PMI(x, y) \approx \log \frac{A \times N}{(A + C) \times (A + B)} \tag{4.2}$$

kde A udává počet spoluvýskytů slov x a y ve stejném kontextu, B udává počet výskytů slova x v kontextu bez slova y a C udává počet výskytů slova y v kontextu bez slova x. N je počet všech kontextů v korpusu. Postupnými úpravami lze rovnici pro výpočet hdonoty PMI (4.1) převést do tvaru.

$$NPMI(x, y) = \frac{\log \frac{P(x \wedge y)}{P(x) \times P(y)}}{-\log P(x \wedge y)} \tag{4.3}$$

Hodnoty NPMI jsou normalizovány do intervalu [-1,1]. Přibližnou hodnotu NPMI lze vypočítat s využitík dat z matice spoluvýskytů.

$$NPMI(x, y) \approx \frac{\log \frac{A \times N}{(A+C) \times (A+B)}}{-\log \frac{A}{N}} \tag{4.4}$$

4.2 Word2Vec

Word2Vec je skupina modelů, která funguje na principu neuronové sítě. Byla vyvinuta výzkumníky ve společnosti Google v roce 2013. Cílem modelů je zachycení sémantických vlastností slov na základě jejich výskytů ve velkém textovém korpusu.

Při trénování je textový korpus procházen po jednotlivých větách. Předem je určena velikost kontextového okna, v jehož středu se nachází zpracovávané slovo. Ostatní slova v kontextovém okně nazýváme kontextovými slovy. Pro každé konetxtové okno jsou vektorové váhy kontextových slov upraveny tak, aby odpovídaly pravděpodobnosti výskytu slov v kontextu se vstupním slovem. Poté je kontetxové okno o jedno slovo posunuto a proces několikrát iteruje přes celý korpus. Po dokončení trénování modelu jsou k vektorové reprezentaci slov používány natrénované vektory.

Word2Vec používá jednoduchou neuronovou síť s jednou skrytou vsrtvou. Vstupní vrstva je tvořena takovým počtem neuronů, jaká je velikost slovníku, kde každý jeden neuron

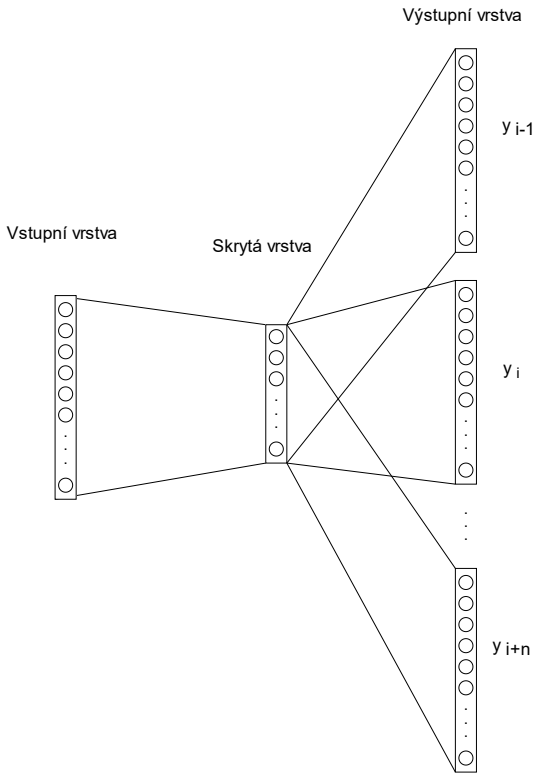
reprezentuje jedno slovo. Skrytá vrstva obsahuje tolik neuronů, jaká je dimenzionalita výstupních vektorů slov. Počet dimenzí výstupních vektorů je parametr, který je nastaven před samotným trénováním. Výstupní vrstva má stejný počet neuronů jako vstupní.

Modely Word2Vec využívají jednu ze dvou architektur: CBOW nebo skip-gram [18].

Skip-gram

Model skip-gram funguje na principu předvídání kontextuálních slov na základě cílového slova. Jedná se v podstatě o opak architektury CBOW - na vstupu je cílové slovo a kontextová slova jsou na výstupní vrstvě.

Obecně bývá tato architektura vhodnější pro práci s velkým množstvím dat, lépe se hodí při zpracovávání málo četných slov.

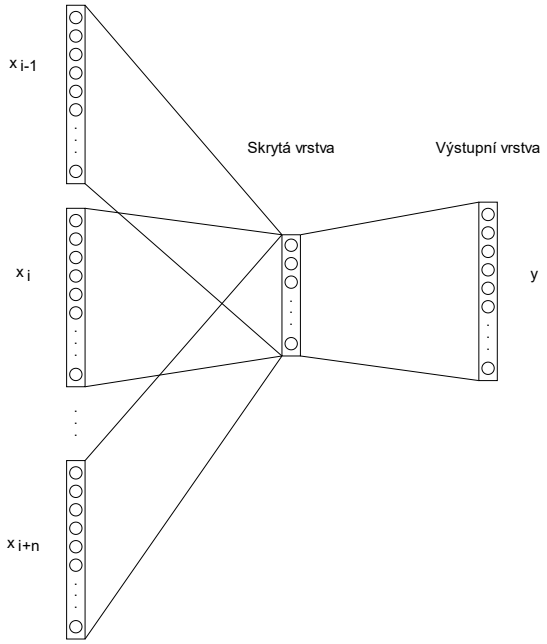


Obrázek 4.2: Zjednodušené schéma architektury skip-gram

CBOW

Model CBOW (z anglického *Continuous Bag of Words*) funguje na principu předpovídání cílového slova na základě slov v jeho kontextu. Kontextová slova tedy tvoří vstupní vrstvu a cílové slovo je na výstupu.

Obecně bývá tato architektura vhodnější pro práci s menším množstvím dat.



Obrázek 4.3: Zjednodušené schéma architektury CBOW

Princip fungování

Podstatu trénování modelu je možné vysvětlit na nejjednoduším příkladu modelu, jak vysvětluje článek[19]. Model má na vstupní vrstvě příjmá právě 1 kontextové slovo (jedná se tedy o velmi jednoduchý model CBOW). Velikost slovníku označíme V, velikost skryté vrstvy N.

Vstupem je vektor o velikosti slovníku, v němž index konkrétního vstupního slova i je nastaven na 1 a ostatní indexy na 0 (tzv. *one-hot encoding*).

Váhy mezi vstupní a skrytou vrstvou jsou reprezentovány maticí *W1* o velikosti $V \times N$. Každý řádek matice *W1* je tedy N-dimenzionální vektor v reprezentující odpovídající slovo ze vstupní vrstvy. Při vynásobení *one-hot encoded* vektoru vstupního slova i ze vstupu s maticí *W1* je výsledkem vektor v_i , což je hodnota skryté vrtsvy h.

$$h = W1^T x := v_i^T \tag{4.5}$$

Váhy mezi skrytou a výstupní vrstvou jsou reprezentovány maticí *W2* o velikosti $N \times V$. Každý sloupec této matice je vektor v'. Slovo j je tedy reprezentováno j-tým sloupcem matice *W2*, vektorem v'_j. Pro každé slovo ve slovníku můžeme vypočítat skóre u_j.

$$u_j = v_j'^T h \tag{4.6}$$

Abychom získali pravděpodobnost výskytu slova j v kontextu slova i, využijeme funkci softmax pro převedení skóre do pravděpodobnostního rozložení:

$$p(w_j|w_i) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})} \tag{4.7}$$

kde y_j je výstupem j-té jednotky výstupní vrstvy. Výstupem je tedy vektor y o velikosti V, který pro každé slovo ze slovníku udává hodnotu pravděpodobnosti jeho výskytu v kontextu vstupního slova.

Na základě vektoru y je poté vybráno slovo s nejvyšší pravděpodobností jako predikované slovo. Dojde k urční ztrátové funkce E.

$$E = -\log(w_o|w_i) \tag{4.8}$$

kde w_o je skutečné cílové slovo. Poté je vypočítána chyba predikce e_j a váhy neuronové sítě jsou pomocí zpětné propagace upraveny tak, aby byla minimalizovaná chyba.

Optimalizace

V modelech Word2Vec je každé slovo reprezentováno dvojicí vektorů - vstupním vektorem v_w a výstupním vektorem v'_w. Učení výstupních vektorů je výpočetně náročné - aby byl aktualizován vektor v'_w, je nutné iterovat každé slovo w_j ve slovníku, vypočítat jeho skóre u_j, pravděpodobnost y_j, chybu pedikce e_j a nakonec využít chybu predikce k aktualizaci výstupního vektoru v'_w. Náročnost těchto výpočtů, které musí být prováděny nad každým slovem ve slovníku, je limitující obzvláště u trénování s velkým množstvím dat, na velkých tetxových korpusech s rozsáhými slovníky. Word2Vec využívá k optimalizaci 2 metody: hierarchický softmax a negativní vzorkování.

Při využití funkce softmax (v její základní formě) je složitost výpočtu $O(V)$, kde V je velikost slovníku. Hierarchický softmax je obdobou této funkce, využívá však binární strom, ve kterém listové uzly reprezentují jednotlivá slova. Pro každý listový uzel existuje jedinečná cesta ke kořeni, která je používána k určení pravděpodobnosti daného slova. Namísto vah všech slov pak matice $W2$ obsahuje váhy všech vnitřních uzlů stromu. Využití hierarchického softmaxu snižuje složitost výpočtu na $O(\log(V))$.

Při trénování neuronové sítě jsou váhy pozitivních vzorků (kontextová slova či cílové slovo v závislosti na architektuře) upraveny tak, aby se jejich pravděpodobnost blížila 1. Váhy všech ostatních slov jsou upraveny, aby se jejich pravděpodobnost blížila 0. Při využití negativního vzorkování místo upravování všech vah matice $W2$ je vybráno S náhodných slov ze slovníku, která budou sloužit jako negativní vzorky. Spolu s vahami pozitivních vzorků jsou váhy upraveny pouze u několika negativních vzorků namísto všech slov ze slovníku. To vede k výraznému snížení časové náročnosti trénování.

4.3 FastText

FastText je open-source knihovna vyvinutá výzkumníky z Facebook AI Research. Vychází z modelu Word2Vec a jejím cílem je také učení vektrových reprezentací slov. Předchůdce fastTextu, model Word2Vec, při trénování pracuje s jedotlivými slovy jako s celky - nebere v potaz jejich vnitřní strukturu. To vede k ignorování syntaktických vztahů slov. Obzvlášť u morfologicky bohatých jazyků, jako je čeština, vede práce s informacemi na úrovni menších celků, než jsou celá slova, ke zlepšení kvality vektorových reprezentací slov. Model fastText pracuje se slovy po částech, které jsou nazývány n-gramy. Každé slovo je rozloženo na skupinu n-gramů. Vektorová reprezentace je vytvořena pro každý n-gram (kde jedním z n-gramů je i celé samotné slovo). Slovo je pak reprezentováno jako průměr vektrorových reprezentací všech jaho n-gramů. To modelu fastText narozdíl od jiných modelů, jako je Word2Vec, umožňuje vytvářet i reprezentace slov, která se nenacházejí v trénovacích datech (tzv. out-of-vocabulary). Model takové slovo jednoduše reprezentuje jako průměr vektorů jeho n-gramů. Tato metoda také modelu umožňuje zachytit vztah mezi slovy základovými a odvozenými (syntaktická a lexikální derivace). Model fastText, podobně jako Word2Vec, implementuje architektury CBOW a skip-gram [4].

4.4 Dict2Vec

Při vytváření vektorových reprezentací slov pracují výše zmíněné modely na principu distribuční sémantiky. Slova, která se v textovém korpusu nacházejí často ve stejném kontextu, budou reprezentována jako sémanticky blízká a naopak slova, která se ve stejném kontextu nevyskytují, nebudou nebudou mít podle těchto modelů velkou souvislost. V přirozeném jazyce je však pravděpodobné, že některá slova vyskytující se ve stejném kontextu spolu nemusejí souviset. Zároveň pouze fakt, že se určitá slova v tréninkovém korpusu nevyskytují dostatečně často ve stejném kontextu, ještě není zárukou, že tato slova nejsou sémanticky blízká. Architektura modelu Dict2vec vychází z modelu Word2Vec, model ale představuje novou strategií učení vektorových reprezentací slov zapomocí slovníkových definic slov. Tyto definice mohou obsahovat skryté informace o podobnosti a souvislosti slov [22].

Slabé a silné páry

Model Dict2vec rozlišuje dva typy párů slov: slabé páry a silné páry. Dvě slova w_a, w_b tvoří slabý pár, pokud se slovo w_a nachází v definici slova w_b, ale slovo w_b se nenachází v definici slova w_a (nebo naopak).

Pokud se slovo w_a nachází v definici slova w_b a zároveň se slovo w_b nachází v definici slova w_a, tato slova tvoří silný pár a zároveň K sémanticky nejbližších slov ke slovu w_a (resp. w_b) tvoří silný pár se slovem w_b (resp. w_a).

Pozitivní vzorkování

Pro každé cílové slovo w_t ze slovníku jsou sestaveny slabé a silné páry. z $W(w_t)$, množiny slov, která s cílovým slovem tvoří slabé páry, je náhodně vybráno n_w slov, která vytvoří množinu $V_w(w_t)$. Z množiny slov $S(w_t)$, která s cílovým slovem tvoří silné páry, je náhodně vybráno n_s slov, která vytvoří množinu $V_s(w_t)$. Poté je pro slovo w_t spočítána cena pozitivního vzorkování J_{pos}.

$$J_{pos}(w_t) = \beta_s \sum_{w_i \in V_s(w_t)} \ell(v_t \cdot v_i) + \beta_w \sum_{w_j \in V_w(w_t)} \ell(v_t \cdot v_j) \tag{4.9}$$

kde ℓ je logistická ztrátová funkce definovaná jako $\ell : x \mapsto \log(1+e^{-x})$ a v_t, v_i, v_j jsou vektory asociované se slovy w_t, w_i, w_j. Nastavení koeficientů β_s, β_w a počtu vybraných vzorků n_s, n_w určuje důležitost slabých a silných párů při učení. Cílem je minimalizovat cenu a tím k sobě přiblížit vektory slov, která tvoří slabé a silné páry.

Negativní vzorkování

Pro každé cílové slovo w_t je vytvořena množina $F(w_t)$ z k náhodně vybraných slov ze slovníku, která nenáleží do množin $W(w_t)$, $S(w_t)$. Cílem je oddělit vektory z množiny $F(w_t)$ od vektoru slova w_t minimalizováním hodnoty J_{neg}.

$$J_{neg}(w_t) = \sum_{\substack{w_i \in F(w_t) \\ w_i \notin W(w_t) \\ w_i \notin S(w_t)}} \ell(-v_t \cdot v_i) \tag{4.10}$$

kde ℓ, v_t, v_i mají stejný význam jako v rovnici (4.9).

Globální cílová funkce

Hodnoty získané pozitivním a negativním vzrkováním jsou poté zahrnuty do výpočtu ceny pro každou (cíl,kontext) dvojici (w_t, w_c).

$$J(w_t, w_c) = \ell(v_t \cdot v_c) + J_{pos}(w_t) + J_{neg}(w_t) \tag{4.11}$$

Globální cílová funkce je definována jako součet cen pro všechny dvojice (cíl,kontext) v celém korpusu.

$$J = \sum_{t=1}^{C} \sum_{c=-n}^{n} J(w_t, w_{t+c}) \tag{4.12}$$

Poté dochází k aktualizaci vah modelu, aby bylo dosaženo co nejnižší hodnoty J.

4.5 Metoda DETECT

Metoda DETECT byla představena výzkumníky z Duke University[15]. V rámci své práce jsem převzala tuto metodu s úpravami, jak je popsáno níže. Tato metoda slouží ke zvýšení kvality nápověd generovaných hlavním špionem a byla vyvinuta za účelem minimalizování 3 zjištěných problémů.

Prvním problémem je výběr velmi málo četných slov jako nápověd. Pokud člen operativy dostane jako nápovědu slovo, které se v trénovacích datech vyskytovalo jen minimálně, je pravděpodobné, že hráč toto slovo vůbec nebude znát. Nápověda, kterou člen operativy nepochopí, protože nezná její význam, je zcela zbytečná.

Druhým identifikovaným problémem je používání velmi častých slov jako nápovědy. Pokud se slovo v trénovacích datech vyskytuje velmi často, dá se předpokládat, že je generické a nemá dostatečně konkrétní význam. Taková nápověda není dostatečně konkrétní a člen opertivy si ji může spojit s větším množstvím herních slov. To povede k nepřesné odpovědi.

Třetím problémem nápověd může být, že nevyužívají běžné, obecně známé znalosti, necílí na hráčův „selský rozum“.

Nápovědy, které trpí těmito nedostatky, nemusejí být nutně špatné ve smyslu efektivního propojení cílených slov. Pokud však hlavní špion při výběru nápověd nebere v potaz, jaké jsou znalosti a jaká je slovní zásoba jeho spoluhráčů, ani nápověda, která je dobrá „na papíře“ nemusí vést k úspěšnému uhodnutí cílených slov.

Metoda DETECT se skládá ze dvou částí - funkce DICT a funkce FREQ. Pro každou kandidátní nápovědu (tj. slovo, které hlavní špion zvažuje při výběru nápovědy) je pomocí metody DETECT vypočítáno skóre. To je pak zahrnuto do celkového skóre, na základě kterého je z kandidátních nápověd vybrána nápověda.

FREQ

K penalizaci málo četných a příliš četných slov využívá metoda DETECT funkci FREQ. Tato funkce přiřadí každé kandidátní nápovědě hodnotu na základě její četnosti výskytu v korpusu.

$$FREQ(w) = -\begin{cases} \frac{1}{f_w} & \text{when} \quad f_w \leq \alpha \\ 1 & \text{when} \quad f_w > \alpha \end{cases} \tag{4.13}$$

kde f_w je frekvence výskytu slova w v korpusu a α je parametr algoritmu. Funkce FREQ' penalizuje málo četná slova více než častá, až do chvíle, kdy je slovo tak časté, že je hodnota jeho frekvence výskytu vetší než parametr α.

DICT

Funkce DICT(w_1, w_2) je definovaná jako podobnost slov w_1 a w_2 dle modelu Dict2vec. Podobnost je vypočtena na základě geometrického vtahu vektorů těchto slov vytvořených modelem Dict2vec.

$$\text{similarity}(w1, w2) = \frac{v1 \cdot v2}{\|v1\| \ \|v2\|} \tag{4.14}$$

DETECT skóre

Pro každou kandidátní nápovědu $\tilde{c}$ je vypočítáno DETECT skóre.

$$\begin{aligned} DETECT(\tilde{c}) = & \lambda_F FREQ(\tilde{c}) + \lambda_D \Bigg(\sum_{t \in T} 1 - DICT(\tilde{c}, t) - \\ & - \max \left(\max_{e \in E} \left(1 - DICT(\tilde{c}, e)\right), 1.5 \times \left(1 - DICT(\tilde{c}, a)\right) \right) \Bigg) \end{aligned} \tag{4.15}$$

kde T je množina všech slov týmu hlavního špiona, E je množina všech slov soupeřícího týmu, a je slovo představující nájemného vraha, jehož výběr představuje větší chybu než výběr nepřátelského slova. Pro jeho větší penalizaci je hodnota $(1 - DICT\,(\tilde{c}, a))$ násobena hodnotou 1,5. λ_F, λ_D jsou parametry algoritmu.

Kapitola 5

Sémantická znalost a kontrola

Studie Paula Hoffmana a Wei Wu z Edinburské univerzity z roku 2022 zkoumá dvě základní složky sémantického poznání - sémantickou znalost a sémantickou kontrolu. Celá studie byla prováděna v anglickém jazyce [23].

Výzkumníci shromáždili data od 127 mladých dospělých ve věku 18-42 let a 106 starších dospělých ve věku 60-85 let. Respondenti absolvovali 3 typy testů - test zaměřený na měření sémantické kontroly, sémantické znalosti a kontrolní test nesémantické kognitivní kontroly (kterému se ve své práci dále nevěnuji). Cílem studie bylo odhalit, jak se v průběhu lidského života mění způsob sémantického poznávání okolního světa v závisloti na věku.

Výsledky této studie ukazují, že starší dospělí mají větší rozsah sémantických znalostí než mladší respondenti. Mladí dospělí oproti tomu vykazují lepší výsledky při testování sémantické kontorly.

5.1 Sémantická znalost

Sémantická znalost je schopnost uchovávat velké množství informací o významu objektů, konceptů a slov, které člověk shromažďuje v průběhu života. Některé studie poukazují na to, že lidské reprezentace pojmů a objektů jsou proměnlivé, mění se na základě kontextu, které si každý člověk nese s sebou např. na základě svých cílů, aktuálních a dlouhodobých zkušeností nebo nervové degenerace [25]. Sémantická znalost bývá obvykle zkoumána pomocí testů slovní zásoby, kde účastníci definují významy slov, identifikují slova nebo vybírají synonyma.

Test šíře sémantické znalosti

V rámci studie účastníci v každém testu šíře sémantické znalosti dostali na výběr 2 slova. Jejich cílem bylo určit, zda mají slova stejný význam. U všech použitých slov byla zaznamenána frekvence jeho výskytu, která byla získána z databáze Subtlex-UK[1]. Frekvence výskytu méně častého slova z dvojice určovala míru náročnosti daného testu - čím méně časté slovo, tím náročnější test, neboť se dá předpokládat, že význam tohoto slova nebudou znát všichni respondenti.

[1] Subtlex-UK – databáze frekvencí výskytů slov pro britskou angličtinu

5.2 Sémantická kontrola

Sémantická kontrola je schopnost získávat a využívat sémantické znalosti vhodně v závislosti na kontextu a úkolu. Úkoly zaměřené na zkoumání sémantické znalosti zahrnují porovnívání pojmů na základě společných vlastností. Tento typ úlohy klade vysoké nároky na schopnost sémantické kontroly, protože automatická aktivace sémantických znalostí často nestačí k určení správné možnosti.

Test sémantické kontroly

V rámci studie byly testy sémantické kontroly zaměřeny na spojování slov na základě společné vlastnosti. Účastník dostal 3 slova - sondu a další 2 nabízená slova, tj. cílené slovo a distraktor. Účastník pak měl určit, které z nabízených 2 slov je podobnější sondě v určené vlastnosti, buď ve velikosti, nebo barvě.

Účastníci museli zaměřit svou pozornost na cílové sémantické vlastnosti a ignorovat ostatní sémantické vlastnosti a další asociace. Potřebu kontroly lze ovlivnit změnou sémantického vztahu sondy k cíli a distraktoru. Když má sonda silný vztah k cíli (např. „Má okurek stejnou barvu jako salát nebo myčka nádobí?“), je potřeba jen malá kontrola, protože stávající sémantická asociace podporuje výběr správné odpovědi. Naopak když je sonda sémanticky příbuzná s distraktorem, ale ne s cílem (např. „Má sůl stejnou barvu jako mrak nebo pepř?“), jsou nároky na kontrolu vysoké, protože silná existující asociace mezi sémanticky blízkými slovy musí být potlačena.

Ke kvantifikaci obtížnosti testu byla vypočtena hodnota TDS (z anglického „target versus distractor strength“). TDS bylo vypočítáno pro každý test.

$$TDS = sim(s, t) - sim(s, d) \tag{5.1}$$

kde s je sonda, t je cílené slovo, d je distraktor a funkce $sim(w_1, w_2)$ určuje kosinovou vzdálenost vektorové reprezentace slov w_1 a w_2. Čím nižší je hodnota TDS, tím větší nároky na sémantickou kontrolu test klade.

Kapitola 6

Analýza předchozích řešení

Základem mé práce byla analýza návrhu a implementace řešení z předchozích prací[11][12]. Tu jsem prováděla pomocí analýzy logů z her odehraných se systémem z předchozích prací a také hraním her s tímto systémem. V této kapitole jsou popsány identifikované problémy předchozího řešení, které má moje práce za cíl minimalizovat či odstranit.

6.1 Zpracování logů

sem napíšu jaký je formát logů hry, popíšu, že jsem prostudováním logů a odehráním několika partií odhalila tyhle problémy.

Stárnutí modelů

Jedním z identifikovaných problémů předchozího řešení je stárnutí modelu. v předchozí práci[11] je pro český jazyk využit korpus CWC-2011. Jedná se o soubor zpráv, článků, blogů a dalších literárních celků. Tento korpus byl vytvořen v roce 2011. Nereflektuje tedy události, které nastaly po jeho vzniku.

V korpusu z roku 2011 se některá slova, která se dnes běžně používají, nevyskytují vůbec nebo jen zřídka. V tabulce 6.1 můžeme vidět příklad četnosti výskytu slov v korpusu CWC-2011 a v korpusu BUbiNG crawl z roku 2022.

Slovo	cwc2011	CZ OURS
streamování	378	12361
covid	0	134832
pandemie	4397	85355
Krym	1231	10908
tiktok	12	4322

Tabulka 6.1: Četnost některých slov v korpusech z roku 2011 a 2022

Krom frekvence výskytu slov se stárnutí modelu projevuje také v sémantických vztazích mezi slovy. Postupem času slova mění kontext, ve kterém jsou používána, což vede ke změně sémantických vztahů. To demonstruje tabulka 6.2, která ukazuje 10 semanticky nejbližších slov ke slovu „kryptoměna“ dle modelu natrénovaného na korpusech CWC-2011 a BUbiNG crawl z roku 2022. v obou případech byla podobnost vyhodnocena pomocí modelu fastText s architekturou skip-gram.

10 nejbližších slov ke "kryptoměna" fastText skipgram model			
cwc2011		**CZ OURS**	
slovo	**příbuznost**	**slovo**	**příbuznost**
vědec	0.3711	bankéř	0.4355
biolog	0.3642	robot	0.4074
démon	0.3586	zlato	0.4051
programátor	0.3253	ropa	0.3956
podvodník	0.3166	podvodník	0.3921
nepřítel	0.3137	počítač	0.3581
upír	0.3117	programátor	0.3396
buňka	0.3072	podnikatel	0.3364
počítač	0.3044	bublina	0.3073
klíč	0.2993	diamant	0.3044

Tabulka 6.2: Deset sémanticky nejbližších slov ke slovu "kryptoměna"dle fastText skipgram modelu natrénovaného na korpusu z roku 2011 a 2022

Podřetězce slov

Při analýze logů her jsem identifikovala několik herních situací, kdy byla zadaná nápověda označena za neplatnou, přestože neporušovala žádné z pravidel. V těchto případech bylo systémem chybně vyhodnoceno, že má zadaná nápověda stejný kořen jako některé z herních slov. Jednalo se například o nápovědu „průmysl“, u níž bylo vyhodnoceno, že má společný slovní základ s herním slovem „rum“. Obdobná situace nastala u nápovědy „pes“, u níž bylo vyhodnoceno, že má společný slovní základ s herním slovem „pěšák“.

Analýzou řešení jsem zjistila, že při vyhodnocování validity nápovědy je jedním z pravidel, že žádné herní slovo nesmí být podřetězcem nápovědy a naopak. Toto pravidlo vede k vyhodnocení platných nápověd jako neplatných. Proto bylo ze systému odstraněno.

Příslučníci národnosti, obyvatelé měst, zeměpisné názvy

U názvů států a měst a jejich obyvatel stemming mnohdy neodhalí společný kořen slov. Takovým příkladem mohou být dvojice slov Řecko a Řek, Itálie a Ital, Praha a Pražák.

Slovo	**Kořen určený stemmerem**
Praha	prah
Pražák	pražá
Řecko	řec
Řek	řek
Itálie	itáli
Ital	ita

Tabulka 6.3: Problematická slova a jejich kořeny dle stemmeru knihovny sumy

Četnost slov

Při výběru nápovědy je nutné, aby bral hlavní špion v úvahu znalosti svých spoluhráčů. Při používání velmi málo četných slov jako nápověd může dojít k situaci, kdy spoluhráč nezná nápovědu. Poté není schopen správně označit cílená slova. Nízká frekvence výskytu slova také může znamenat, že model při trénování neměl dostatek informací, aby správně modeloval jeho vztahy s ostatními slovy. To může vést k nepřesné reprezentaci slova. Tuto skutečnost předešlá práce řeší rozdělením slov podle četnosti výskytu do kategorií. V případě pužití strategie je skóre slov váhováno hodnotou $(0,1 * \log_{10}(\text{počet výskytů slova}))$. Pokud není použita strategie, je ke skóre slov, která se v testovacích datech vyskytla více než 1000krát, přičtena konstanta 0,1.

Obdobným problémem, který předešlá práce neřeší, je využití velmi častých slov jako nápověd. V takovém případě může mít nápověda natolik obecný význam, že si hráč nadokáže vytvořit konkrétní sémantickou asociaci s určitými herními slovy. To může vést k označení nesprávných slov. Ve své práci jsem se rozhodla oba výše popsané problémy řešit s využitím metody DETECT, která je popsána v kapitole 4.5.

Kapitola 7

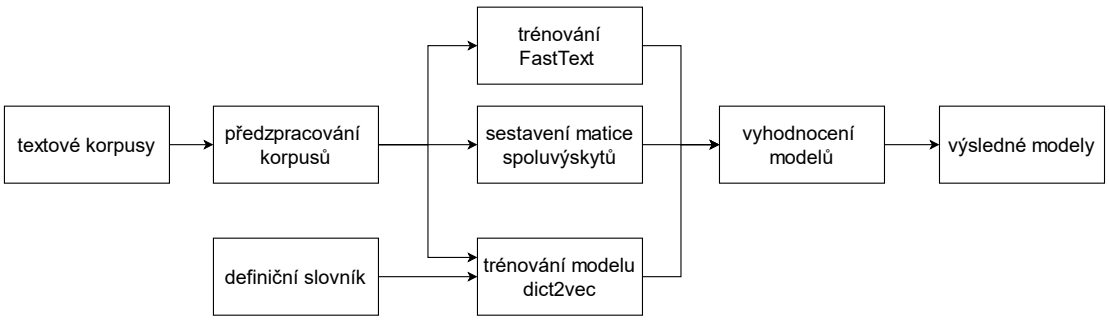
Návrh a implementace systému pro hraní Krycích jmen

Výsledný systém pro hraní hry Krycí jména vychází z minulých prací[11][12]. Ve své práci se věnuji systému pracujícímu s českým jazykem. V rámci své práce jsem prostudovala implementaci minulých řešení a odhalila jsem nedostatky a problémy, jak je popsáno v kapitole 6. Tato kapitola popisuje návrh a implementaci výsledného systému.

7.1 Architektura systému

Původní systém z předchozích prací vypadal následovně. Implementoval hráče v roli hlaního špiona i člena operativy. K určování sémantické podobnosti slov moduly operativy i špiona využívaly kombinaci modelu založeného na počtu NPMI a prediktivního modelu fastText. Pro češtinu byly tyto modely natrénovány na korpusu CWC-2011. Vytvořená webová služba umožňovala testování hráče v obou rolích ve standardní verzi i ve verzi Duet.

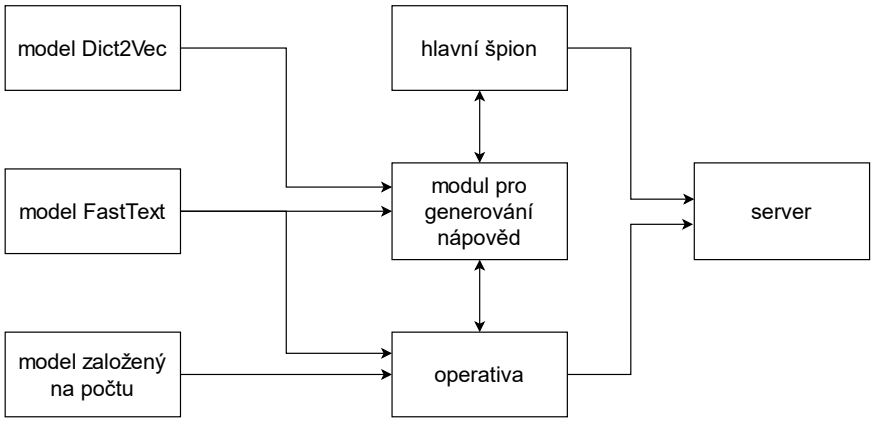
Schéma architektury výsledného systému je uvedeno na obrázcích 7.1 a 7.2. První část se věnuje předzpracování nových textových korpusů a natrénování sémantických modelů. Tyto modely jsou následně využity moduly člena operativy a hlavního špiona.



Obrázek 7.1: Architektura části systému pro předzpracování textových korpusů a trénování sémantických modelů

Systém pro hádání byl přejat z předešlých prací a využívá kombinaci modelů fastText a NPMI. Modul pro generování nápověd byl upraven tak, aby při generováni nápověd využíval kombinaci modelu fastText a metody DETECT, která využívá model Dict2Vec. Poslední

částí systému je serverová aplikace, která propojuje rozhraní umělého hráče s webovým prohlížečem. Tato část byla přejata z předešlých prací.



Obrázek 7.2: Architektura části systému pro hraní Krycích jmen

7.2 Trénování modelů

Nejdříve byly předzpracovány 3 české korpusy. Následně byly tyto korpusy využity k natrénování sémantických modelů. Modely byly vyhodnoceny a nejúspěšnější z nich jsou použity v systému pro hraní Krycích jmen.

Použité korpusy

Prvním použitým korpusem je korpus Wikipedia 20230901 CZ. Jedná se o kolekci textů získaných z kompletní databáze české Wikipedie. Tento korpus byl vytvořen 1. 9. 2023 s využitím nástroje wikiextractor[1]. Druhým korpusem je korpus CZ OURS vytvořený s využitím webového prohledávače BUbiNG[2]. Korpus byl vytvořen 8. 4. 2022 a obsahuje velké množství textových dat z 312701 různých webových domén.

Posledním použitým korpusem je KKorpus 2.0. Jedná se o největší ze zmíněných korpusů. Tento korpus obsahuje data extrahovaná z české wikipedie k 1. 1. 2023 a webová data získaná prohledávačem BUbiNG k 1. 9. 2020 a 24. 2. 2023. Dále obsahuje data z korpusu csTenTen[3] vytvořeného v roce 2017, korpusů HPLTDatasets v1[4], CulturaX[5] a Historical Corpora[6], z 807 elektonických záznamů knih a více než půl milionu komentářů z domény idnes.cz[7].

Předzpracování korpusů

Každý z použitých korpusů je uložen v jiném formátu. Z toho důvodu byl pro předzpracování každého z nich vytvořen samostatný skript.

[1] https://github.com/attardi/wikiextractor
[2] BUbiNG: Massive Crawling for the Masses
[3] csTenTen: Corpus of the Czech Web
[4] HPLTDatasets v1
[5] CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages
[6] Historical Corpora
[7] https://www.idnes.cz

K předzpracování korpusu Wikipedia 20230901 CZ slouží 2 skripty. Prvním je skript `parse_wiki_from_extractor.py`, který odstraní html značky a další nepotřebné znaky. Poté je použit skript `preprocess_cz.py`, který pomocí knihovy MorphoDiTa[8] provede lemmatizaci každého slova, odfiltruje interpunkci a bezvýznamová slova.

K předzpracování korpusu CZ OURS slouží skript `preprocess_cs_crawl.py` a k předzpracování korpusu KKorpus 2.0 slouží skript `preprocess_knot_corpus.py`. Oba tyto skripty také slouží k odfiltrování nežádoucích znaků a bezvýznamových slov a využívají knihovnu MorphoDiTa k lemmatizaci.

Pro vytváření slabých a silných páru při trénování modelu Dict2Vec je dále použitý Slovník spisovné češtiny, který obsahuje definici více než 28 tisíc pojmů.

Název korpusu	**Velikost**	**Velikost slovníku**
Wikipedia 20230901 CZ	906 MB	386581
CZ OURS	17.07 GB	1598303
KKorpus 2.0	252 GB	5076785

Tabulka 7.1: Velikost jednotlivých použitých korpusů a velikost jejich slovníku po předzpracování

Model založený na počtu

K vytvoření matice spoluvýskytů, kterou následně využívá model založený na počtu, slouží skript `create_cooccurrence.py`. Tento skript byl převzat z předešlé práce[11]. Kvůli velikosti korpusu KKorpus 2.0 a velkému množství unikátních slov nebyla z tohoto korpusu vytvořena matice spoluvýskytů. Stejný problém byl řešen i v předchozí práci, kde nebylo možné sestavit matici z korpusu seznam2017. Autor v práci uvedl, že „Kvůli velkému počtu unikátních slov a omezení operační paměti nebylo možné vytvořit tímto způsobem matici spoluvýskytů z korpusu seznam2017. (...) I kdyby se podařilo matici vytvořit, byla by příliš velká pro rozumné použití“[11]. Korpus seznam2017 obsahoval 8697736 unikátních slov, zatímco korpus KKorpus 2.0 obsahuje 5076785 unikátních slov.

Model fastText

Při trénování prediktivního modelu jsem vycházela z předchozí práce[11], kde byl model fastText vyhodnocen jako nejúspěšnější prediktivní model z trojice modelů fastText, Word2Vec a GloVe. V této práci byla vyhodnocena nejúspěšnější kombinace parametrů pro trénování modelu fastText. Jedná se o architekturu Skip-gram s použitím negativního vzorkování.

Skript `trainFasttext.py` slouží k trénování modelu pomocí Facebook knihovny fastText[9]. Natrénovaný model je uložen v textovém formátu, který obsahuje pouze vektory slov, a v binární formátu, který obsahuje celý model. Textový formát je méně paměťově náročný, ale neobsahuje vektorové reprezentace jednotlivých n-gramů slov. Neumožňuje tedy aproximaci vektorů slov, která nejsou ve slovníku. V předchozí práci byl model fastText s aproximací slov, která nejsou ve slovníku, vyhodnocen jako úspěšnější, než model bez aproximace. Z tohoto důvodu je dále využíván model uložený v binárním formátu.

[8] MorphoDiTa: Morphologický slovník a tagger je open-source nástroj pro morfologickou analýzu textů v přirozeném jazyce. Dokumentace: ufal.mff.cuni.cz/morphodita

[9] knihovna fastText

Model Dict2vec

Pro natrénování modelu Dict2vec je nejdříve model natrénován bez využití silných a slabých párů. Je tedy natrénován Word2Vec model s architekturou skip-gram. Tento model je vytvořen, protože natrénované vektorové reprezentace slov poté slouží k určení K nejbližších slov při vytváření silných párů (viz kapitola 4.4).

Poté jsou vytvořeny slabé a silné páry na základě slovníkových definic. K tomu byl využit Slovník spisovné češtiny. Nejdříve byl využit skript `preprocess_cz.py` k lemmatizaci definic. Poté byl vytvořen skript `load_def_dict.py`, který slouží k načtení definičního slovníku a jeho převedení do formátu vyžadovaného pro trénování modelu. K vytvoření párů slouží skript `generate_s.py`, který s využitím natrénovaného Word2Vec modelu a slovníkových definic vytvoří dva tetxové soubory – jeden obsahující slabé páry, jeden obsahující silné páry. Páry jsou uloženy ve formátu dvojice slov oddělených mezerou na řádek.

K natrénování samotného modelu Dict2vec slouží program `dict2vec.c`, který je napsaný v jazyce C a je dostupný z oficiálního github repozitáře projektu[10]. Model je natrénován s využitím slabých a silných párů a vektory slov získané trénováním jsou uloženy v textovém formátu. Při trénování modelu je nastaveno několik parametrů. Bylo natrénováno několik modelů s různými kombinacemi parametrů:

- strong-draws: počet silných párů, které jsou vybrány pro pozitivní vzorkování
- weak-draws: počet slabých párů, které jsou vybrány pro pozitivní vzorkování
- epochs: počet epoch při trénování

Ostatní parametry byly ponechány ve výchozím nastavení. Každý model byl poté vyhodnocen, jak je popsáno v kapitole 7.3, aby byla zjištěna nejúspěšnější kombinace parametrů.

Pro využití modelu při generování nápověd byl vytvořen skript `dict2vec_model.py`, který obsahuje definici třídy `Dict2vecModel`. Objekt třídy je instancován s jedním parametrem, kterým je název souboru, v němž jsou uložené vektorové reprezentace slov vytvořené modelem Dict2vec. Třída implementuje metodu `similarity()`, která slouží k výpočtu podobnosti slov. K tomu využívá funkci `cosine()` z modulu scipy.spatial.distance knihovny SciPy[11].

7.3 Vyhodnocení modelů

K vyhodnocení modelu založeného na počtu a modelu fastText byl obdobně jako v předchozí práci[11] využit kód dostupný v github repozitáři NLP-FIT[12] s úpravou pro evaluaci modelu založeného na počtu. Testovací data tvoří záznamy 680 tahů reálných odehraných partií hry Krycí jména. Testovací sada obsahuje záznamy z let 2018 a 2019. U každého tahu je zaznamenána nápověda hlavního špiona spolu se slovy, která byla nápovědou cílena. Vyhodnocení probíhá porovnáním slov, která model vyhodnotí jako nejpodobnější nápovědě, se slovy, která byla cílena a která byla operativou označena. Pokud model vyhodnotí cílená slova jako nejpodobnější, je úspěšnost 100%. Čím dále v pořadí podobnosti s nápovědou cílená slova jsou, tím nižší úspěšnost model vykazuje.

[10] Veřejný repozitář se zdrojovým kódem pro trénování modelu Dict2vec
[11] knihovna SciPy
[12] veřejný repozitář NLP-FIT

Vyhodnocení NPMI

V tabulce 7.2 jsou uvedeny výsledky vyhodnocení modelu založeného na počtu s využitím metody NPMI. Metoda NPMI byla v předešlé práci vyhodnocena jako nejúspěšnější z metod, které implementuje model založený na počtu. Proto ve své práci dále využívám pouze tuto metodu.

Při využití matice spoluvýskytů sestavené z korpusu CZ OURS vykazuje model asi o 0,72% větší úspěšnost než model z předchozí práce, který využívá korpus cwc2011. Zároveň došlo ke snížení počtu slov, která nejsou ve slovníku (OOV), zhruba o polovinu.

Při využití matice spoluvýskytů sestavené z korpusu Wikipedia 20230901 CZ vykazuje model zhoršení oproti korpusu cwc2011 asi o 7%. Zároveň výrazně větší část slov z testovací sady, více než 5%, není ve slovníku modelu.

Korpus	**Počet slov OOV**	**Úspěšnost NPMI**
CZ OURS	4 (0,5%)	77,41%
cwc2011	10 (1%)	76,69%
Wikipedia 20230901 CZ	47 (5,3%)	69,36%

Tabulka 7.2: Vyhodnocení NPMI na různých korpusech

Vyhodnocení modelu fastText

V tabulce 7.3 jsou uvedeny výsledky vyhodnocení modelu fastText natrénovaného na různých korpusech. Nejnižší úspěšnost vykazuje model, který byl natrénován na korpusu Wikipedia 20230901 CZ. Žádný z natrénovaných modelů nedosáhl lepšího výsledku než model z předešlé práce natrénovaný na korpusu cwc2011. Úspěšnost modelu natrénovaného na korpusu KKorpus 2.0 je však téměř srovnatelná, nový model vykazuje asi o 0,1% horší výsledek.

Korpus	**Úspěšnost modelu fastText**
cwc2011	74,94%
KKorpus 2.0	74,84%
CZ OURS	74,34%
Wikipedia 20230901 CZ	68,53%

Tabulka 7.3: Vyhodnocení modelu fastText na různých korpusech

Jelikož je hlavní motivací trénování nových modelů reakce na stárnutí modelu, rozhodla jsem se ve své práci využít model natrénovaný na korpusu KKorpus 2.0. Přestože model vykazuje mírně horší výsledek, je natrénovaný na výrazně větším množství dat, která jsou navíc mnohem aktuálnější.

Vyhodnocení modelu Dict2vec

Účelem využití modelu Dict2vec v řešení je zlepšení výběru nápovědy v kombinaci s modelem fastText. Z toho důvodu bylo vyhodnocení modelu Dict2vec provedeno hodnocením nápověd vygenerovaných hlavním špionem. Cílem testování bylo najít model, který bude nejvíce zlepšovat generované nápovědy.

Testovací sadu tvořilo 86 záznamu prvních tahů reálných her. Nápovědy byly vygenerovány modulem hlavního špiona s využitím modelu fastText a algoritmu DETECT.

V každém testu modelů Dict2vec byl využit stejný model FastText natrénovaný na korpusu CZ OURS. V každém testu byly pro metodu DETECT nastaveny stejné parametry. Úspěšnost jednotlivých testů je tedy závislá pouze na využití různých natrénovaných modelů Dict2vec.

K vygenerování nápověd slouží skript `get_hints.py`. Nápovědy vygenerované špionem s využitím jednotlivých modelů Dict2vec jsou následně předány skriptu `eval_detect_hints.py`, kde je příjmá modul operativy. Ten na základě přijaté nápovědy označí předpokládaná cílená slova.

Pokud je označeno cílené slovo, skóre dané nápovědy je 1 (úspěch). Pokud operativa označí slovo, které nebylo cíleno, ale jedná se o slovo daného týmu, nápovědě je přiřazeno skóre 0,3. Při označení neutrálního slova je skóre 0, při označení slova nepřátelského týmu je skóre -0,5 a při označení nájemného vraha -1.

Celkové skóre modelu Dict2vec (v rámci metody DETECT) je poté spočteno jako aritmetický průměr skóre všech nápověd.

Kvůli časové náročnosti trénování jednotlivých modelů byly nejdříve všechny kombinace parametrů použity pro natrénování modelu Dict2vec na nejméně paměťově náročném korpsu Wikipedia 202309011 CZ. Pro modely natrénované s různými kombinacemi parametrů pak bylo vyhodnoceno skóre úspěšnosti nápověd, jak je uvedeno v tabulce 7.4.

Strong-draws	**Weak-draws**	**Epoch**	**Skóre**
3	2	2	86,32%
3	2	3	84,73%
2	1	2	84,42%
2	1	1	83,86%
3	2	1	83,31%

Tabulka 7.4: Úspěšnost nápověd s využitím modelů Dict2vec natrénovaných na korpusu Wikipedia 202309011 CZ s různými hodnotami parametrů strong-draws, weak-draws a epoch.

Jako nejúspěšnější kombinace parametrů trénování bylo vyhodnoceno: strong-draws 3, weak-draws 2, epoch 2. S touto kombinací parametrů byl následně natrénován model i na korpusu CZ OURS. Z důvodu velké časové a paměťové náročnosti model Dict2vec nebyl trénován na korpusu KKorpus 2.0. Srovnání úspěšnosti nápověd modelů natrénovaných s nejúspěšnějšími paramtery trénování ukazuje tabulka 7.5.

Korpus	**Skóre nápověd**
Wikipedia 202309011 CZ	86,32%
CZ OURS	85%

Tabulka 7.5: Úspěšnost nápověd s využitím modelů Dict2vec natrénovaných na korpusu Wikipedia 202309011 CZ a na korpusu CZ OURS. Oba modely byly natrénovány s nejúspěšnější kombinací parametrů trénování – strong-draws 3, weak-draws 2, epoch 2.

Model natrénovaný na korpusu CZ OURS dosáhl horších výsledků. To může být způsobeno tím, že je se povaha korpusu získaného z internetové encyklopedie Wikipedia lépe hodí pro trénování modelu Dict2vec. V původní práci, která tento model představila, je také využit korpus získaný z Wikipedie. Proto je model natrénovaný na korpusu Wikipedia 202309011 CZ využit v řešení hlavního špiona v implementaci metody DETECT.

7.4 Implementace člena perativy

Implementace člena operativy byla přejata z předešlé práce[11]. Člen operativy je implementován třídou `Operative`, jejíž difinice se nachází ve skriptu `operative.py`. Modul člena operativy má za cíl na základě přijaté nápovědy vyhodnotit podobnost herních slov s nápovědou a označit slova, která považuje za cílená. Člen operativy implementuje strategii hraní, která udržuje kontext mezi jednotlivými tahy a využívá informace o kontextu při výběru slov.

Zpracování nápovědy

Nápověda je po přijetí lemmatizována s využitím knihovny MorphoDiTa, která vyhodnotí i Part-of-speech (POS) tag nápovědy. V některých případech nemusí být lemmatizace jednoznačná. V takových případech je přihlédnuto k POS tagům a je definováno několik pravidel výběru lemmat:

- Pokud nápověda může být v prvním i jiném pádu, použije se pouze lemma pro první pád.
- Pokud se může jednat o sloveso i jiný slovní druh, lemma slovesa je použito pouze pokud má sloveso přiřazený čas.
- Pokud se jedná o sloveso s přiřazeným časem, které má stejný tvar jako významově jiné slovo, je vyhodnoceno, zda je lemma jiného slovního druhu stejné jako vstupní slovo (před lemmatizací). Pokud je lemma jiného slovního druhu shodné se vstupním slovem, je použito toto lemma. Jinak je použito sloveso.
- Pokud se jedná o vlastnost chrakterizovanou podstatným jménem ženského rodu končící na „ost“, dojde k záměně koncovky „ost“ na „ý“. Výjimkou jsou některá pevně definovaná slova jako „událost“ či „kost“

Všechna lemmata, která zůstanou po vyhodnocení těchto pravidel, jsou použita pro vyhodnocení podobnosti. Podobnost nápovědy se slovem je nakonec dána nejpodobnějším lemmatem. Všechna herní slova jsou také lemmatizována.

Použitá kombinace modelů

Operativa využívá k vyhodnocení podobnosti kombinaci skóre z modelu fastText a normalizovaného skóre NPMI. Použití kombinace záleží na tom, jestli se lemma nápovědy nachází ve slovnících obou modelů.

- Pokud je nápověda ve slovníku obou modelů, použije se aritmetický průměr nenulových hodnot
- Pokud je nápověda pouze v modelu NPMI, seřadí se nenulové hodnoty podle skóre NPMI. Slova s nulovou hodnotou NPMI jsou seřazena podle aproximace vektoru OOV slova modelu fastText.
- Pokud je nápověda pouze ve slovníku fastText, použijí se jeho hodnoty.
- Pokud není nápověda v žádném modelu, použije se aproximace vektoru OOV slova z modelu fastText.

V případě použití aritmetického průměru hodnot je průměr vypočten, pokud je hodnota NPMI nenulová. Pokud je hodnota NPMI rovna 0, použije se pouze skóre fastText. Další váhování nápovědy je provedeno s využitím definičního slovníku. Pokud je při vyhodnocení podobnosti zjištěno, že se slovo nachází v definici druhého slova, skóre podobnosti se zvýší o 0,25.

Strategie operativy

Operativa využívá systém váhování možných slov nepřítele. Na začátku hry jsou váhy všech herních slov nastaveny na nulu. Při zadání nápovědy nepřátelským špionem je vyhodnocena podobnost slov s nápovědou stejně jako u vlastní nápovědy. Pro nápovědu s číslem N je vypočítána nová váha u N nepodobnějších slov.

$$w_i = w_i + \frac{sim(h, i)}{N \cdot \alpha} \tag{7.1}$$

kde w_i je váha slova i, $sim(h, i)$ je skóre podobnosti slova s nápovědou h a parametr α je nastaven jako konstanta 0,75. Pokud je hádání nepřátelské operativy ukončeno předčasně špatnou odpovědí, ze seznamu váhovaných slov seřazeného podle podobnosti je vybráno tolik slov, kolik protivníkovi pro danou nápovědu zbývalo označit. Pro tato slova jsou aplikovány vypočítané váhy.

V tahu operativy je pak při vyhodnocování skóre slov využita váha slova k penalizaci. Celkové skóre slova je spočteno tak, že od vyhodnocené podobnosti každého slova s nápovědou je odečtena jeho aktuální váha. Na konci každého kola obou týmů jsou váhy všech slov, která nebyla pozitivně váhována v daném kole, sníženy o hodnotu 0,05 (nejmenší hodnota váhy je 0). Tím je zaručeno, že slovo, které bylo nesprávně označeno za napřátelské na začátku hry, může být později vyhodnoceno jako slovo mého týmu.

Pokaždé když operativa vybírá slovo k označení ve vlastním tahu, je tah uložen ve formátu: N označovaných slov, další 2 slova s nejvyšším skóre navíc a případně slovo, které bylo přidáno na dokončení předchozích tahů. Pokud dojde k označení slova, které není týmu operativy, a toto slovo nebylo přidáno jako dokončování předchozího tahu, uloží se nápověda jako nedokončená s počtem zbývajících slov k označení.

Pokud operativa dostane nápovědy s běžným číslem, k cíleným slovům je přidáno první slovo ze seřazeného seznamu nedokončených tahů. Při obdržení nápovědy s číslem 0 je váha nejpodobnějšího slova upravena tak, aby nebylo označeno do konce hry. Jako cílená slova je pužit seznam slov z nedokončených tahů. Pokud operativa dostane nápovědu s číslem „nekonečno“, použije seznam slov z nedokončených tahů a přidá jedno nejpodobnější slovo k nápovědě.

Seznam slov z nedokončených tahů je seřazen následovně: nejdříve všechna původně zamýšlená slova, která nejsou váhována, poté záložní slova, která nejsou váhována, a nakonec ostatní slova seřazená vzestupně podle váhy. Z každé nápovědy se bere dohromady nejvíce tolik slov, kolik zbývalo označit. Nápověda je odstraněna ze seznamu nedokončených, pokud v seznamu nejsou žádná slova nebo zbývá 0 slov k označení.

Pokud se nepřátelský tým blíží vítězství, použije operativa agresivnější přístup při obdržení nápovědy typu „nekonečno“. Když protivníkovi zbývají dvě slova, k nejpodobnějšího slovu jsou postupně přidávána další slova, u kterých platí, že jejich podobnost s předchozím slovem v pořadí je vyšší než s následujícím. Pokud protivníkovi zbývá už jedno slovo, je jeho výhra téměř zaručena. V takovém případě je použito tolik slov, aby celkový počet odpovídal tomu, kolik týmu operativy zbývá uhádnout.

V rámci hádání slov je také dobré zamyslet se nad zadanou nápovědou. Pokud pro N předpokládaných cílených slov existuje lepší nápověda, než zadaná nápověda, může být důvodem zadání příslušné nápovědy snaha cílit jiné slovo. Při výběru slov je nejdříve vyhodnocena podobnost nápovědy ke slovům, jak je popsáno výše. Poté je vybráno N nejpodobnějších slov, kde N je počet cílů. K těmto slovům je vyhodnoceno 1000 nejpodobnějších slov modelem fastText pomocí funkce `most_similar()`. Pokud se lemma zadané nápovědy nenachází v nejpodobnějších slovech, je vygenerována nejlepší nápověda pomocí těchto nejpodobnějších slov. Za tuto nápovědu je označena ta, u které je největší rozdíl podobnosti mezi posledním cíleným slovem a prvním necíleným. Pro každé necílené slovo je spočítáno skóre, které je rozdíl mezi průměrem podobnosti cílených slov s nápovědou a jeho podobností s nápovědou. Toto skóre je spočítáno pro původní zadanou i nově vygenerovanou nápovědu. U slova, kde je toto skóre o největší hodnotu vyšší pro původní nápovědu oproti vygenerované, se dá předpokládat, že na ni hlavní špion chtěl cílit původní nápovědou. Rozdíl skóre pro původní a nově vygenerovanou nápovědu je přičten k hodnotě podobnosti slova vůči zadané nápovědě. Toto zvýšení hodnoty může pomoci dané slovo označit.

7.5 Implementace hlavního špiona

Implementace hlavního špiona vychází z předešlé práce[11]. Hlavní špion je implementován třídou `Spymaster`, jejíž definice se nachází ve skriptu `spymaster.py`. Úkolem hlavního špiona je generování vhodných nápověd pro operativu. Při volbě nápovědy je nutné vhodně zvolit, na kolik slov bude nápověda cílit. Nápověda cílící větší množství slov je méně konkrétní, čímž se zvyšuje šance na označení chybného slova. Pokud však nepřátelskému týmu zbývá uhodnout méně slov, je vhodné pokusit se náskok nepřítele dohnat volbou riskantnějších nápověd, které cílí větší množství slov.

Metoda DETECT

Metoda DETECT při výpočtu počítá s několika parametry: α pro metodu FREQ, λ_D a λ_F pro výpočet celkového skóre. Parametry metody DETECT jsou uloženy jako atributy objektu třídy `Spymaster`. Metoda DETECT je implementována ve skriptu `select_hint.py`, který slouží k uchování jendotlivých nápověd a jejich vyhodnocení. Skript obahuje definici třídy `Hint` a metoda DETECT je implementována jako metoda této třídy.

Při testování bylo dosaženo lepších výsledků, když nebyly příliš časté nápovědy penalizovány tolik, jak je popsáno v kapitole 4.5. Skóre DETECT je v řešení počítáno jako:

$$FREQ(w) = -\begin{cases} \frac{1}{f_w} & \text{when} \quad f_w \leq \alpha \\ \frac{1}{100} & \text{when} \quad f_w > \alpha \end{cases} \tag{7.2}$$

Nastavení parametrů metody DETECT

Nastavení parametrů metody ovlivňuje úspěšnost metody. K vyhodnocení nejlepšího nastavení těchto parametrů byl využit stejný způsob, který je popsána v sekci 7.3. Při testování tedy byl pro jednotlivé testy objekt třídy Spymaster inicializován s různými hodnotami těchto atributů a bylo provedeno vyhodnocení úspěšnosti nápověd s využitím modulu operativy.

Jako nejlepší kombinace parametrů bylo vyhdonoceno: $\alpha = 100000$, $\lambda_D = 2$, $\lambda_F = 5$.

Parametr α	Skóre
100 000	0.847481
1 000 000	0.845155
500 000	0.840504
10 000	0.837209
5 000 000	0.833527
5 0000	0.832171

Tabulka 7.6: Skóre nápověd při využití různých hodnot parametru alpha

Parametr λ_D	Skóre
2	0.836628
1	0.836047
5	0.833527
3	0.829070
4	0.829068

Tabulka 7.7: Skóre nápověd při využití různých hodnot parametru λ_D

Parametr λ_D	Skóre
5	0.845155
4	0.840698
1	0.840698
2	0.836628
3	0.829071

Tabulka 7.8: Skóre nápověd při využití různých hodnot parametru λ_F

Seznam použitelných nápověd a kořenů slov

V rámci optimalizace využívá modul hlavního špiona předem sestavený seznam všech slov ze slovníku modelu fastText, která jsou vhodnou nápovědou. K sestavení tohoto seznamu slouží skript `get_valid_hints.py()`, který vyfiltruje všechna nevhodná slova na základě POS značky slovního druhu.

Zároveň je pro každou kandidátní nápovědu nutné kontrolovat, zda nemá stejný kořen jako některé z herních slov. K optimalizaci této operace slouží soubor vygenerovaný skriptem `gat_stems.py` Ten obsahuje seznam všech henních slov a ke každému z nich seznam všech slov ze slovníku fastText, která mají stejný kořen. Obdobně jako v předešlé práci je porovnána řetězcová vzdálenost kořenů slov. Pokud jsou si kořeny podobné alespoň na 85%, je vyhodnoceno, že slova mají stejný kořen. Do seznamu slov se stejným kořenem byla ručně přidána problematická slova popsaná v kapitole 6.1.

Výběr nápovědy

K výběru nápovědy implementuje třída `Spymaster` metodu `select_best_hint()`. Ze všech slov daného týmu jsou vytvořeny kombinace až 5 slov. Poté jsou pomocí funkce knihovny fastText `most_similar()` vyhodnocena nejpodobnější slova pro každou kombinaci. Slovo představující nájemného vraha je použito jako negativní vzorek při výběru. V rámci optimalizace je pro každou kombinaci vyhodnoceno 2000 nejpodobnějších slov.

Každé získané slovo je poté porovnáno se seznamem slov vhodných pro nápovědu. Všechna slova, která se nacházejí v tomto seznamu, se stávají kandidátní nápovědou.

Poté jsou ze seznamu kandidátních nápověd vyfiltrována všechna slova, která mají společný kořen s některým z herních slov. K tomu je využit uložený seznam, který pro každé herní slovo definuje seznam všech slov ze slovníku modelu, která mají stejný kořen.

Poté je nad seznamem vyfiltrovaných kandidátních nápověd provedena evaluace jednotlivých nápověd. Pro každou kandidátní nápovědu je spočteno skóre DETECT. Pokud se nápověda nenachází ve slovníku modelu Dict2vec, je skóre dáno pouze hodnotou FREQ. Hodnota FREQ závisí pouze na počtu výskytů daného slova v korpusu, takže ji lze vypočítat pro každou nápovědu. Pokud je nápověda ve slovníku modelu, je vypočtena také suma podobnosti nápovědy se všemi slovy příslušného týmu a maximální hodnota podobnosti s nepřátelským slovem (kde podobnost s nájemným vrahem je váhována x1,5) a je vypočítáno celkové skóre DETECT, jak je popsáno v kapitole 4.5.

Poté je pro každé herní slovo spočítáno skóre. To je dáno podbností s nápovědou dle modelu fastText, ke které je připočteno skóre dle metody DETECT. Zapojení DETECT skóre do skóre každého herního slova namísto celkového skóre nápovědy během testování vykazovalo nejlepší výsledky. Pokud nápověda cílí na 3 a více slov, je dovoleno, aby jedno z cílených slov byl náhodný kolemjdoucí (nesmí to však být nejpodobnější cílené slovo). Pro kandidátní nápovědu je spočteno skóre score_diff. Pokud je povoleno cílit náhodného kolemjdoucího, score_diff je rozdíl skóre posledního cíleného a prvního nepřátelského slova. Jinak je score_diff rozdíl skóre posledního cíleného slova a prvního slova, které nenáleží danému týmu. Celkové skóre kandidátní nápovědy je pak spočteno jako

$$score = N^{\frac{M}{E}}(score\_diff) \tag{7.3}$$

kde N je počet cílených slov, M představuje počet zbývajících neuhodnotých slov daného týmu a E počet zbývajících neuhodnotých slov nepřátelského týmu. Násobení skóre nápovědy hodnotou N zaručí, že nebudou příliš vybírány nápovědy cílící malé množství slov. Takové nápovědy jsou sice snadno uhodnutelné, ale označení jednoho slova v každém tahu pravděpodobně nepovede k vítězství. Prametr $\frac{M}{E}$ udává míru rizika, které je nutné při výběru nápovědy podstoupit. Pokud je poměr slov příslušného týmu ke slovům protivníků velký, příslušný tým prohrává a je nutné zvolit nápovědu cílící na větší množství slov.

V prvním herním kole je zvolena agresivní strategie. Po vyhodnocení nejvhodnější nápovědy, jak je popsáno výše, je vyhodnocena nejlepší nápověda cílící o jendo slovo více. Pokud je skóre této nápovědy alespoň 80% skóre vyhodnocené nápovědy, použije se tato agresivnější nápověda namísto původní.

Hyperonyma

Pro zlepšení nápovědy je definovaný pojem hyperonym. Po vybrání nejvhodnější nápovědy je zkontrolováno, zda se v seznamu hyperonym nenachází slovo, které by cílilo na více herních slov. Pravidlem je, že pod dané hyperonymum nesmí spadat nepřátelská slova ani vrah. Pokud je takové slovo nalezeno, je použito jako nápověda.

Webová služba

Webový služba byla převzata z předešlé práce [11]. Webová služba běží na školním serveru athena7 a je přístupná na adrese http://athena7.fit.vutbr.cz:8087/. Slouží k hraní Krycích jmen ve webovém prohlížeči. Počítač může zastoupit roli hlavního špióna nebo člena operativy. Služba implementuje velké množství jazyků, má práce se věnuje pouze češtině. K

implementaci byl použit mikroframework Flask. Na straně webového prohlížeče komunikaci zajišťuje jazyk JavaScript.

Vyhodnocení

Vyhodnocení kvality nápověd byl využit stejný způsob, jako je popsán v kapitole 7.3. Stejným způsobem byl vyhodnocen i model z předchozí práce pro srovnání. Systém z předchozí práce dosáhl 89,15% úspěšnosti. Systém představený v této práci dosáhl úspěšnosti 86,42%. Nedošlo tedy ke zlepšení výkonu.

Nápovědy vygenerované systémem z předešlého systému a novým systémem byly také testovány lidskými hráči. Testovací rozhraní zobrazilo uživatelovi sadu 35 vybraných herních situací a uživatel měl za úkol vždy označit tolik slov, jaké číslo nesla nápověda.

	Systém z předešlé práce	**Nový systém**
Označeno cílených slov	336	328
Označeno necílených slov týmu	67	90
Označeno kolemjdoucích	72	76
Označeno nepřátelských slov	27	17
Označeno vrahů	10	10

Tabulka 7.9: Srovnání výběru slov při hraní lidských hráčů s oběma systémy.

Vyhodnocení nápověd bylo počítáno stejně, jako je popsáno v kapitole 7.3, jediným rozdílem zůstává, že hadaná slova v reakci na nápovědu negeneroval modul operativy, ale označoval je člověk. Nápovědy vygenerované systémem z předchozí práce dosáhly při tetsování reálných uživatelů průměrného skóre 63,8%. Nápovědy vygenerované novým systémem dosáhly průměrného skóre 64,6%. Pro oba systémy bylo celkem označeno stejné množství nájemných vrahů, každý hráč v průměru označil 2.

Z toho vyplývá, že přestože systém z předešlé práce je schopen lépe spolupracovat sám se sebou, když zastupuje roli špiona i operativy, nový systém je schopen mírně lepší spolupráce s člověkem. Je však nutné zmínit, že rozsah testování s lidskými hráči nebyl velký, účastnilo se pouze 5 hráčů, všichni ve věku mezi 20 a 30 lety.

Kapitola 8

Studie STST II.

Různé obdoby komunikačních her mohou být prostředkem pro zkoumání lidské komunikace. Tím se zabývá i studie STST II. V této studii hrály dvojice účastníků komunikační hru. Během experimentu byla účastníkům měřena mozková aktivita pomocí funkční magnetické rezonance (fMRI). Jedná se o moderní zobrazovací metodu, která se snaží mapovat funkční oblasti mozku aktivované při provádění určitého úkolu či stimulace [9]. Cílem studie bylo zkoumání vzájemného propojení hráčů při hraní komunikační hry,

8.1 Pravidla

Každý herní blok se skládal z deseti kol. Na začátku herního bloku viděli oba účastníci na obrazovce počítače jiné slovo. Bloky byly označeny typem „HÁDÁNÍ“ nebo „ASOCIACE“.

V bloku typu „HÁDÁNÍ“ měli účastníci za cíl uhodnout slovo, které jejich partner řekne jako další. Toho měli dosáhnout na základě slov, která oba vyslovili nebo viděli v předchozím kole. Pokud se oběma hráčům podařilo říci stejné slovo před koncem bloku, pokračovali ve hře a snažili se nalézt další slovo, které jejich partner řekne. V bloku typu „ASOCIACE“ bylo úkolem účastníků říci v každém kole slovo, které je významově příbuzné s předchozím slovem druhého hráče.

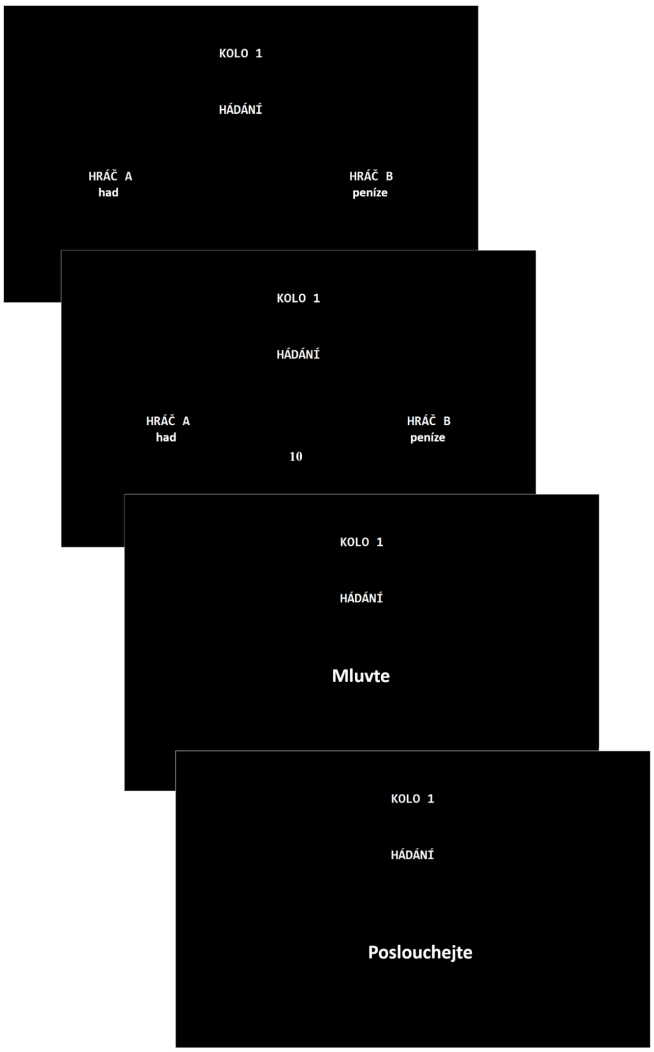
Každý blok obsahoval 10 herních kol. V každém kole měli účastníci říct jedno podstatné jméno, sloveso nebo přídavné jméno. V obou typech bloků museli účastníci v každém kole říci nové slovo, opakování slov bylo zakázáno.

8.2 Způsob vyhodnocení dat

K vyhodnocení podobností slov získaných studií STST II. byl využit model fastText natrénovaný na korpusu KKorpus 2.0. K práci s daty byla využita knihovna pandas. K výpočtu trendu byla využita knihovna scikit-learn.

Filtrace dat

Nejdříve byly odfiltrovány ty odpovědi, při kterých účastníci nedodrželi pravidla. Prvním pravidlem, které měli účastníci dodržovat, bylo v každém kole říct pouze 1 slovo. Pokud účastníci použili slovní spojení, tuto odpověď nebylo možné zpracovat, protože model fastText pracuje s vektorovými reprezentacemi jednotlivých slov (a jejich částí). Výjimkou bylo použití slovesa se zvratným zájmenem „se“ nebo „si“. V takovém případě bylo k vyhodnocení použito samotné sloveso.



Obrázek 8.1: Příklad instrukcí, které viděli účastníci na obrazovce v průběhu jednoho kola

Dalším z pravidel, které účastníci měli dodržet, bylo použití podstatných jmen, přídavných jmen nebo sloves. Vyhodnocení slovního druhu odpovědi bylo provedeno pomocí POS taggeru knihovny Morphodita. Data byla ručně překontrolována, aby bylo zabráněno nesprávnému určení slovního druhu u slov, jejichž slovní druh nelze určit jednoznačně (např. slovo ráno lze vyhodnotit jako příslovce, ale také se může jednat o podstatné jméno). Dále bylo kontrolováno, zda v průběhu jednoho bloku hráči nezopakovali slova, která v tomto bloku již řekli sami nebo je řekl jejich spoluhráč.

Z analýzy dat musely být vynechány odpovědi párů 3, 7, 9, 16, 18, 24, 26, 33 a 40. Tyto páry totiž mluvily slovensky a vyhodnocení dat s využitím modelu fastText je jazykově závislé (model fastText využitý k analýze je natrénovaný na českém korpusu).

120 odpovědí muselo být analýzy vyřazeno, protože hráč vyslovil více než 1 slovo. Dalších 47 odpovědí bylo vyřazeno kvůli nepovolenému slovnímu druhu, 124krát hráči zopakovali slovo, které již v daném bloku zaznělo, a 117krát nebyla odpověď zaznamenána, protože

hráč špatně slyšel, žádné slovo nevymyslel nebo jeho odpovědi ve zvukovém záznamu hry nejde dobře rozumět.

8.3 Výsledky

U bloků označených jako „HÁDÁNÍ“ určuje hodnota podobnosti slov, která účastníci řekli v jednom kole, kvalitu jejich odhadu. Pokud oba účastníci řekli stejné slovo, znamená to, že uhodli slovo, které spoluhráč řekne. U bloků označených jako „ASOCIACE“ určuje hodnota podobnosti slov, která účastníci řekli v jednom kole, jak moc se při výběru asociací vzájemně ovlivňují. V obou případech se tedy dá říci, že v ideálním případě, kdy se na sebe při hraní účastníci „napojují“, jejich odpovědi jsou sémanticky čím dál tím bližší, až nakonec řeknou stejné slovo.

Hodnoty podobnosti odpovědí jsou jednoznačně uspořádanou časovou řadou. Pro kvantifikaci hodnoty vzájemného „napojování“ hráčů byla v rámci každého herního bloku spočítána hodnota lineárního trendu podobnosti jejich odpovědí v 1 kole. Trend je obecná tendence vývoje zkoumaného jevu v čase. Trend může být rostoucí, klesající nebo může existovat řada bez trendu [1].

Z 670 zkoumaných bloků byl výsledný trend kladný zhruba u poloviny (336) a zhruba u poloviny záporný (334). To naznačuje, že zvýšit hodnotu vzájemného napojení během jednoho bloku se účastníkům povedlo zhruba v polovině případů

Dále byla z podobnosti slov, která účastníci řekli v každém kole, vypočítána průměrná hodnota pro každý blok. Předpoklad byl takový, že pokud se na sebe účastníci v průběhu několika herních bloků dokážou lépe napojit, průměrná hodnota podobnosti jejich odpovědí za blok se bude zvyšovat. Ze 48 zkoumaných párů se pouze u 16 z nich projevil stoupající trend průměrné hodnoty podobnosti za blok. To naznačuje, že vzájemné napojení hráčů se u většiny z nich v průběhu hraní několika herních bloků nezvýšilo.

Dále bylo zjištěno, kolika párům se během 1 herního bloku, který má 10 kol, povedlo alespoň v polovině kol říci slova, jejichž podobnost je alespoň 0,5. Tohoto výsledku dosáhlo 11 párů ze 48. Nejúspěšnějšími páry byly 49, 50 a 57, kterým se povedlo dosáhnout tohoto výsledku ve 2 blocích.

38 párům se povedlo aspoň 1x v rámci bloku říct stejné slovo. 22 párům se povedlo v jendom herním kole říci stejné slovo alepsoň 2x. I v tomto ohledu patří páry 49 a 57 k nejúspěšnějším. Páry 49 a 50 řekl alepsoň 2x stejné slovo ve 4 herních blocích. Stejného výsledku dosáhly páry 35 a 43.

Podrobné výsledky jsou uvedeny v příloze A

Kapitola 9

Nástroj pro tvorbu testů sémantické znalosti a kontroly

Studie, zabývající se testováním sémantické znalosti a kontroly, jak je popsána v kapitole 7.3, byla vytvořena v anglickém jazyce. Všechna slova v testech sématické kontroly a znalosti byla anglicky a respondenti odpovídali v angličtině. Existuje snaha vytvořit podobnou studii v českém jazyce. Pro sestavení jednotlivých testů semanticky kontroly i sémantické znalosti jsem v rámci své práce vytvořila jednoduchý webový nástroj. Tento nástroj má za cíl výzkumníkům, kteří se budou zabývat obdobou studie pro český jazyk, usnadnit tvorbu testovacích otázek.

9.1 Použité technologie

Webový nástroj pro tvorbu testů na backendu využívá jazyk python s využitím frameworku Flask[1]. Jedná se o mikroframework sloužící k vývoji webových aplikací. Na frontendu využívá aplikace jazyk JavaScript pro dynamické zobrazení dat uživateli. V aplikaci je také využita správa sezení s využitím Flask (Flask session[2]).

Slova zadaná uživatelem jsou lemmatizována pomocí knihovny MorphoDiTa. K vyhodnocování podobnosti slov a jejich četnosti v korpusu je využit model fast text natrénovaný na korpusu KKorpus 2.0. Seznam českých synonym je uložen ve formátu .xml. Dále aplikace využívá seznam českých podstatných jmen k filtrování slov. Tento seznam je uložen v textovém souboru ve formátu 1 slovo na řádek. Podstatná jména byla získána ze souboru, který je zdrojempro morfologický analyzátor, na základě značky slovního druhu

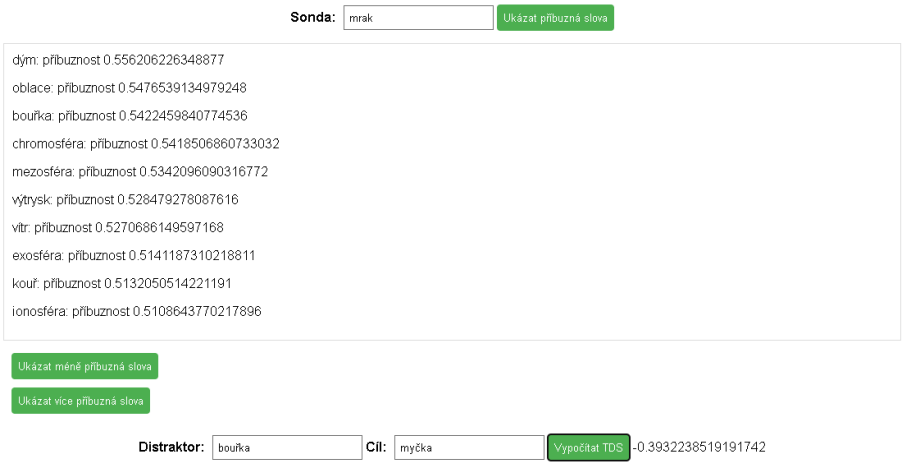
9.2 Nástroj pro tvorbu testů sémantické kontroly

V testu sémantické kontroly je testovací otázka vždy tvořena trojicí slov a důležitou metrikou je jejich sémantická podobnost. Vytvořila jsem jednoduchou webovou stránku na které uživatel zadá slovo, které má být sondou. S využitím metody `most_similar()` modelu FastText je získáno 1000 nejpodobnějších slov k sondě.

Poté jsou s využitím seznamu podstatných jmen vytříděna slova, která podstatným jménem nejsou. Seznam slov je uložen do session a uživateli je zobrazeno 10 nejpodobnějších podstatých jmén spolu s jejich hodnotou podobnsoti se sondou. Uživatel může procházet

[1] Dokumentace frameworku Flask: flask.palletsprojects.com
[2] Dokumentace rozšíření Flask Session flask-session.readthedocs.io

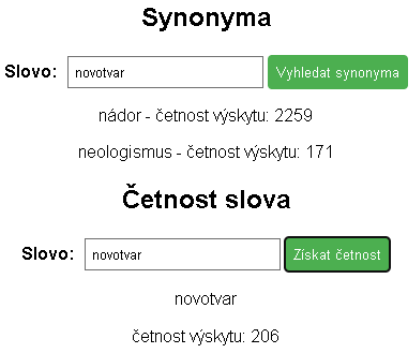
seznamem seřazeným dle podobnosti a vybrat si ze slov distraktor. Po vyplnění distraktoru a cíleného slova je vypočítána hodnota TDS. To uživateli umožní vytvářet vhodné trojice slov pro testování sémantické kontroly.



Obrázek 9.1: Ukázka webového nástroje pro tvorbu testu sémantické kontroly

9.3 Nástroj pro tvorbu testů sémantické znalosti

V testu sémantické znalosti je testovací otázka tvořena dvojicí slov, která buď jsou nebo nejsou synonyma. Metriku obtížnosti u tohoto testu je četnost výskytu méně častého slova v korpusu. Pro účely sestavení těchto testů byla vytvořena webová stránka, která uživateli k zadanému slovu ve slovníku českých synonym vyhledá a následně zobrazí slova stejného významu. Pro jakékoliv zadané slovo stránka vrátí četnost jeho výskytu. To uživateli umožní vytvářet vhodné dvojice synonym a zjistit četnost výskytu slov pro testování sémantické kontroly.



Obrázek 9.2: Ukázka webového nástroje pro tvorbu testu sémantické znalosti

Kapitola 10

Závěr

V rámci práce byl implementován systém pro hraní Krycích jmen. Systém se zaměřuje především na nedostatky předchozího systému. Nový systém využívá modely natrénované na korpusch KKorpus 2.0 a CZ OURS v odpovědi na stárnutí modelů z předešlé práce. V modulu hlavního špiona je implementována metoda DETECT, která má za cíl zlepšit kvalitu generovaných nápověd. Při testování kvality nápověd s lidskými hráči dosáhli nápovědy generované novým systémem o 0,8% lepšího výsledného skóre v porovnání se systémem z předešlé práce. Součástí systému je i webová služba[1], která umožňuje hrát Krycí jména s implementovaným systémem.

V rámci práce byl také vytvořen nástroj k usnadnění vytvoření sady otázek pro studii sémantické kontroly a znalosti. Tento nástroj umožňuje vyhledávání sémanticky podobných slov, výpočet hodnoty TDS a vytváření párů sysonym. Nástroj je dostupný na webu[2].

Součástí práce byla také analýza dat ze studie STST II. Data byla nejdříve vyfiltorvána na základě definovaných pravidel a poté byly vypočítány hodnoty sémantických podobností slov, kzeré hráči řekli. Tyto hodnoty by měly reflektovat vzájemné myšlenkové napojení hráčů.

Námětem pro budoucí práci může být natrénování modelu Dict2vec na větším vzorku dat. Použitý slovník pro český jazyk obsahuje definici 28000 pojmů, což je pouze 0,55% velikosti slovníku použitého modelu fastText. Extrakcí dat z internetových slovníků by bylo možné získat větší testovací sadu. Dalším námětem pro budoucí práce by mohlo být vytvoření adaptivního modelu hráče, který by se dokázal automaticky přizpůsobit jazyku, strategii a stylu hraní svého spoluhráče.

[1] Systém pro hraní Krycích jmen http://athena7.fit.vutbr.cz:8087
[2] Nástroj pro tvorbu testů sémantické kontroly a znalosti http://athena7.fit.vutbr.cz:8086

Literatura

[1] Ambrožová, K. *Anaýza trendu (odhad lineárního trendu a klouzavé průměry)*. Masarykova univerzita, duben 2016. Dostupné z: https://is.muni.cz/el/1431/jaro2016/Z2069/um/54271982/55862544/Prez_Stcv5_pondelni_skup.pdf. Studijní podklady k předmětu Statistické metody II, cvičení č. 5.

[2] Archibald, C. a Brosnahan, S. *Adapting to Teammates in a Cooperative Language Game*. 2024. Dostupné z: https://arxiv.org/abs/2403.00823.

[3] Bills, J. a Archibald, C. A Deductive Agent Hierarchy: Strategic Reasoning in Codenames. In: *2023 IEEE Conference on Games (CoG)*. 2023, s. 1–8.

[4] Bojanowski, P. et al. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*. Cambridge, MA: MIT Press, 2017, sv. 5, s. 135–146. Dostupné z: https://aclanthology.org/Q17-1010.

[5] Boleda, G. Distributional Semantics and Linguistic Theory. *Annual Review of Linguistics*. Annual Reviews, 2020, sv. 6, Volume 6, 2020, s. 213–234. ISSN 2333-9691. Dostupné z: https://www.annualreviews.org/content/journals/10.1146/annurev-linguistics-011619-030303.

[6] Cambria, E. a White, B. Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *IEEE Computational Intelligence Magazine*, 2014, sv. 9, č. 2, s. 48–57.

[7] Chowdhary, K. R. Natural Language Processing. In: *Fundamentals of Artificial Intelligence*. New Delhi: Springer India, 2020, s. 603–649. ISBN 978-81-322-3972-7. Dostupné z: https://doi.org/10.1007/978-81-322-3972-7_19.

[8] Chvátil, V. *Krycí jména*. Říjen 2015. Dostupné z: https://www.mindok.cz/userfiles/files/pravidla/8595558302239_50.pdf. Pravidla hry Krycí jména.

[9] fMRI Brno. *Co je funkční magnetická rezonance fMRI?* online. 2008. Dostupné z: http://fmri.mchmi.com/main_index.php?strana=5. Výzkumná skupina při LF MU v Brně.

[10] Jaramillo, C. M. et al. Word Autobots: Using Transformers for Word Association in the Game Codenames. In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. říjen 2020, sv. 16, s. 231–237. Dostupné z: https://ojs.aaai.org/index.php/AIIDE/article/view/7435.

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

[12] Jareš, P. *Rychlá adaptace počítačové podpory hry Krycí jména pro nové jazyky.* Brno, CZ, 2021. Master's thesis. Brno University of Technology, Faculty of Information Technology. Dostupné z: https://www.fit.vut.cz/study/thesis/21497/.

[13] K., O. *Krycí jména* online. Dostupné z: https://www.zatrolene-hry.cz/spolecenska-hra/kryci-jmena-4961/.

[14] Koyyalagunta, D. et al. Playing Codenames with Language Graphs and Word Embeddings. *Journal of Artificial Intelligence Research*, Červen 2021, sv. 71, s. 319–346. Dostupné z: https://dl.acm.org/doi/pdf/10.1613/jair.1.12665.

[15] Koyyalagunta, D. et al. Playing Codenames with Language Graphs and Word Embeddings. *Journal of Artificial Intelligence Research*, Květen 2021. Dostupné z: https://arxiv.org/abs/2105.05885.

[16] Lenci, A. a Sahlgren, M. From Usage to Meaning: The Foundations of Distributional Semantics. In: *Distributional Semantics*. Cambridge University Press, 2023, s. 3–25. Studies in Natural Language Processing.

[17] Liu, V. a Curran, J. R. Web Text Corpus for Natural Language Processing. In: School of Information Technologies, University of Sydney. *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, Duben 2006, s. 233–240. Dostupné z: https://aclanthology.org/E06-1030.

[18] Nicholson, C. V. *A Beginner's Guide to Word2Vec and Neural Word Embeddings* online. 2023. Dostupné z: https://wiki.pathmind.com/word2vec.

[19] Rong, X. Word2vec Parameter Learning Explained. *ArXiv*, 2014, abs/1411.2738. Dostupné z: https://api.semanticscholar.org/CorpusID:2154019.

[20] Sidji, M. a Stephenson, M. Prompt Engineering ChatGPT for Codenames. In:. Srpen 2024.

[21] Tabassum, A. a Patil, D. R. R. A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing. In: *International Research Journal of Engineering and Technology (IRJET)*. červen 2020, sv. 7, č. 6, s. 4864–4867. Dostupné z: https://api.semanticscholar.org/CorpusID:235211496.

[22] Tissier, J. et al. Dict2vec : Learning Word Embeddings using Lexical Dictionaries. In: Palmer, M. et al., ed. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Září 2017, s. 254–263. Dostupné z: https://aclanthology.org/D17-1024.

[23] Wu, W. a Hoffman, P. Validated measures of semantic knowledge and semantic control: normative data from young and older adults for more than 300 semantic judgements. *Royal Society Open Science*, Únor 2022, sv. 9, č. 2.

[24] Yannakakis, G. N. a Togelius, J. *Artificial Intelligence and Games*. 1. vyd. Springer Cham, 2018. ISBN 978-3-319-63519-4.

[25] Yee, E. Fluid semantics: Semantic knowledge is experience-based and dynamic. *The speech processing lexicon: Neurocognitive and behavioural approaches*. Walter de Gruyter GmbH & Co KG, 2017, sv. 22, s. 236.

[26] Škrabal, M. *Korpus* online. Říjen 2021. Dostupné z: https://wiki.korpus.cz/doku.php/pojmy:korpus. Wiki Českého národního korpusu.

[27] Šprinclová, Z. Jazykovědkyně Michaela Lišková zkoumá neologismy: Nová slova patří mladým. *AΩ / Věda pro každého*, 2022, 1/2022. ISSN 2570-7566.

Příloha A

Výsledky analýzy dat studie STST II.

V této příloze jsou formou tabulek uvedeny konkrétní výsledky vyhodnocení dat ze studie STST II.

Podrobná tabulka obsahující vypočítaný trend pro každý herní blok zde není uvedevna kvůli její velikosti (671 řádků).

Číslo páru	**Trend**
49	0.023823
27	0.012117
43	0.007795
36	0.006913
32	0.006499
51	0.006124
34	0.005415
57	0.004702
29	0.003246
55	0.002901
28	0.002739
25	0.002429
11	0.002386
52	0.001443
12	0.001129
45	0.000690

Tabulka A.1: Trendy průměrné podobnosti slov vyslovených párem v jednom kole v průběhu herních bloků. Páry, které dosáhly pozitivního trendu, seřazené podle velikosti trendu sestupně. Hodnoty trendů jsou zaokrouhleny na 6 desetinných míst.

Číslo páru	Trend
38	-0.000200
37	-0.000554
17	-0.000758
1	-0.001033
19	-0.001121
42	-0.001235
39	-0.001347
56	-0.001663
10	-0.001793
54	-0.002153
44	-0.002708
48	-0.003670
5	-0.003763
2	-0.004223
53	-0.004295
20	-0.005494
41	-0.006380
23	-0.006731
6	-0.006810
4	-0.006928
35	-0.007905
14	-0.007929
46	-0.008921
30	-0.009108
13	-0.009740
22	-0.010081
8	-0.010310
47	-0.010610
31	-0.012223
21	-0.012562
15	-0.014672
50	-0.016243

Tabulka A.2: Trendy průměrné podobnosti slov vyslovených párem v jednom kole v průběhu herních bloků. Páry, které dosáhly negativního trendu, seřazené podle velikosti trendu sestupně.

Číslo páru	**Počet bloků se skóre >0.6 alespoň v 5 kolech**	**Čísla bloků**
49	2	5, 14
50	2	1, 7
57	2	3, 11
13	1	3
23	1	6
27	1	1
30	1	1
37	1	11
43	1	7
47	1	10
51	1	10

Tabulka A.3: Páry, které v rámci jednoho bloku alepsoň v polovině kol dosáhli podobnosti odpovědí > 0.6, a příslušné bloky

Číslo páru	Počet bloků s alespoň 1 shodou	Čísla bloků
28	9	1, 2, 4, 5, 6, 9, 11, 13, 14
31	8	1, 3, 4, 5, 6, 9, 11, 12
55	8	3, 4, 5, 6, 9, 10, 12, 14
22	7	1, 2, 4, 5, 9, 10, 12
43	7	1, 5, 7, 8, 12, 13, 14
49	7	4, 5, 7, 8, 10, 11, 14
54	7	1, 4, 5, 8, 9, 10, 12
13	6	1, 3, 6, 7, 8, 9
35	6	2, 4, 6, 7, 11, 12
48	6	2, 7, 8, 9, 10, 11
50	6	1, 2, 3, 5, 7, 8
57	6	3, 6, 10, 11, 12, 13
12	5	3, 5, 8, 9, 12
27	5	6, 7, 10, 12, 13
46	5	1, 6, 7, 10, 12
47	5	1, 2, 8, 9, 10
51	5	3, 5, 8, 10, 14
37	4	1, 2, 10, 11
56	4	1, 7, 9, 10
6	3	1, 4, 5
17	3	2, 9, 14
29	3	5, 11, 12
42	3	2, 4, 12
45	3	3, 7, 10
52	3	1, 9, 10
32	2	8, 11
34	2	7, 10
38	2	5, 6
39	2	4, 12
41	2	4, 14
44	2	4, 11
53	2	3, 5
14	1	1
15	1	1
21	1	6
23	1	6
30	1	3
36	1	14

Tabulka A.4: Páry, které v rámci jednoho bloku alepsoň jednou vyslovili stejné slovo, a odpovídající bloky

Číslo páru	**Počet bloků s alespoň 2 shodami**	**Čísla bloků**
Pár 57	4	3, 10, 11, 13
Pár 35	4	6, 7, 11, 12
Pár 43	4	5, 7, 13, 14
Pár 49	4	7, 8, 11, 14
Pár 31	3	4, 5, 9
Pár 27	2	12, 13
Pár 50	2	1, 7
Pár 56	2	1, 9
Pár 48	2	7, 11
Pár 51	2	10, 14
Pár 22	1	2
Pár 15	1	1
Pár 47	1	1
Pár 52	1	1
Pár 54	1	8
Pár 32	1	8
Pár 34	1	7
Pár 45	1	7

Tabulka A.5: Páry, které v rámci jednoho bloku alepsoň dvakrát vyslovili stejné slovo, a odpovídající bloky