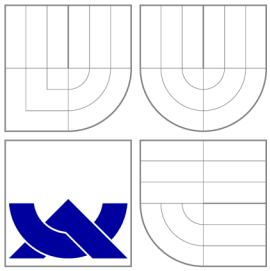
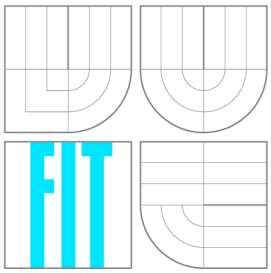


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

URČOVÁNÍ BLÍZKOST POJMŮ V OBLASTI INFORMAČNÍCH TECHNOLOGIÍ

IDENTIFYING TERM SIMILARITY IN INFORMATION TECHNOLOGY DOMAIN

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MILOSLAV SMUTKA

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2016

Zadání bakalářské práce/18688/2015/xsmutk00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Smutka Miloslav**
Obor: Informační technologie
Téma: **Určování blízkost pojmů v oblasti informačních technologií**
Identifying Term Similarity in Information Technology Domain
Kategorie: Umělá inteligence

Pokyny:

1. Prostudujte metody pro identifikaci jednoslovných a víceslovných odborných termínů a pro měření jejich sémantické podobnosti.
2. Seznamte se s existujícími jazykovými nástroji, které mohou být použity ke zpřesnění výše uvedeených metod, zejména v oblasti syntaktické a sémantického značkování.
3. Shromážděte data potřebná pro průběžné testování jednotlivých fází řešení problému
4. Na základě získaných poznatků navrhněte a realizujte systém, který dokáže z kolekce relevantních textů v oblasti informačních technologií extrahovat termíny a navrhout hierarchii pojmů vzhledem k jejich sémantice.
5. Vyhodnoťte realizované řešení a porovnejte zvolený přístup s alternativními přístupy.

Literatura:

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- funkční prototyp řešení

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrž Pavel, doc. RNDr., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá návrhem, implementací a vyhodnocením výsledků systému pro vyhledávání sémanticky blízkých slov. Pro určení vztahů mezi slovy systém využívá model word2vec z knihovny gensim.

Abstract

This bachelor thesis works with the idea, implementation and evaluation of resulting system for retrieval of semantically related words. For the determination of word relations, gensim library word2vec model is used.

Klíčová slova

zpracování přirozeného jazyka, sémantická podobnost, gensim, word2vec

Keywords

natural language processing, semantic similarity, gensim, word2vec

Citace

SMUTKA, Miloslav. *Určování blízkost pojmů v oblasti informačních technologií.* Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Smrž Pavel.

Určování blízkost pojmů v oblasti informačních technologií

Prohlášení

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

.....
Miloslav Smutka
18. května 2016

Poděkování

Chtěl bych poděkovat panu doc. RNDr. Pavlu Smržovi, Ph.D. za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Mé díky patří i Davidu Sloukovi za korekturu.

Obsah

Kapitola 1

Úvod

Zpracování přirozeného jazyka (v angličtině Natural Language Processing, odtud zkratka NLP) je obor na pomezí umělé inteligence a počítačové lingvistiky, který se zabývá problematikou analýzy jazyka za pomoci počítačových systémů. Zvláště v současné době, kdy se počítače užívají v širokém spektru oborů lidské činnosti a jejich výpočetní výkon prudce stoupá, je důležitá snaha, aby člověk s počítačem mohl komunikovat přirozeným jazykem a nikoliv pouze skrze speciální rozhraní.

Jedním z důležitých cílů NLP je shlukování slov podle významu. Cílem toho je určit, jak významově blízké si jsou dva pojmy. Jednou z možností získání této blízkosti jsou vektorové modely, které mapují slova sémanticky podobná blízko sebe v předem definovaném N-rozměrném prostoru.

Cílem práce je seznámit se se současnými metodami určování podobnosti slov a na základě těchto poznatků pak navrhnout systém, který umožní vyhledávat pro zadané slovo slova významově blízká. Práce je založena na modelu Word2vec a jako vzorek přirozeného jazyka je použit korpus Wikipedie. Na závěr jsou provedena vyhodnocení výsledků pro modely s různým nastavením.

Kapitola 2

Rozbor problematiky

Tato kapitola poskytuje teoretický základ, nezbytný pro pochopení bakalářské práce. Zabývá se všemi kroky potřebnými pro analýzu jazyka, počínaje předzpracováním korpusu anglické Wikipedie, jeho převodem na různé vektorové modely a dále možnostmi jeho zpracování pomocí algoritmů strojového učení.

2.1 Zpracování přirozeného jazyka

Zpracování přirozeného jazyka (NLP – Natural Language Processing) je obor, zkoumající, jak lze dosáhnout komunikace mezi člověkem a počítačem pomocí jazyka, ať už psaného, nebo mluveného. Výzkum je zaměřen na široké spektrum vědních oborů, počínaje sběrem poznatků o tom, jak lidé rozumí jazyku a jak jej používají. Tyto poznatky jsou pak užity pro vývoj nástrojů a technik, které umožní počítačům nalézt vztahy mezi slovy, slovními spojeními, větami i celými dokumenty. Výzkum se opírá o další obory, včetně informatiky, lingvistiky, matematiky, umělé inteligence, psychologie, strojového učení a dalších[4].

2.2 Předzpracování textu

Před samotným zpracováním textu je třeba předzpracovat soubor obsahující vzorek přirozeného jazyka, který bude podroben analýze. V závislosti na povaze vstupního textu je většinou nutné odebrat různé formátovací značky. Při předzpracování se zpravidla odstraňují interpunkční znaménka a text se rozděluje na jednotlivá slova (tokeny). Ty se mohou dále upravovat pro snížení počtu různých slov v textu, například převodem na základní tvar slova. Dále je možné text upravit odebráním nevýznamových a nespecifických slov. To jsou taková slova, která jsou příliš obecná a nenesou, z pohledu dokumentu, žádné důležité informace. Jedná se například o spojky, předložky a další. Jejich odstranění se provede na základě stoplistu, což je seznam slov, která se z korpusu odstraní. [7]

2.2.1 Tokenizace

Tokenizací se obecně rozumí proces rozdělení textu na jednotky, se kterými se bude dál pracovat. V případě tvorby korpusu se jedná o rozdělení na jednotlivá slova či fráze. Při tomto procesu se odstraní vše, co nenese význam textu; tedy zvláštní znaky, číslovky a interpunkční znaménka. Ne vždy je to však vhodné, někdy například můžeme chtít zachovat číslovky, či pouze data. Stejně tak někdy můžeme chtít ponechat spojovníky nebo některé

jiné zvláštní znaky. (Článek [7, 22, Kap. 2.2.1] uvádí pro příklad názvy jako M*A*S*H nebo C++, které by odstraněním zvláštních znaků ztratily svůj specifický význam). Stejně tak je někdy lepší text nedělit na jednotlivá slova, ale ponechat víceslovné názvy jako jeden token.

2.2.2 Tvorba n-gramů

Tvorba n-gramů je proces, kdy je na základě výskytu slov v textu generován seznam slovních spojení, která k sobě logicky patří, a jejichž samostatné rozdělení a následné zpracování by způsobilo ztrátu informace, kterou text původně nesl. Například slova „kontaktní“ a „čočky“ nesou sama o sobě jinou informaci, než při spojení do sousloví „kontaktní čočky“, které definuje zpřesnění informace. Taková slovní spojení je pak vhodné zpracovávat nikoliv zvlášť, ale jako jedno slovo, neboť se jedná o ustálené slovní spojení, které nese informaci jako jeden celek.

Jednou z možností extrakce slovních spojení je extrahování automaticky na základě relativní četnosti slov. Při tomto postupu jsou za ustálená slovní spojení považována ta slova, která se vyskytují v korpusu vedle sebe častěji, než by odpovídalo náhodnému rozložení pravděpodobnosti.

2.2.3 Truecasing

Truecasing je metoda, která se využívá pro opravu velikosti písma tam, kde byla nějakým způsobem znehodnocena [6]. U běžných textů se tato metoda hodí zejména pro první slova ve větě, která se píší s velkým písmenem na začátku; což není pro strojové učení vhodné. Obecně se však dá využít všude tam, kde není velikost písma odpovídající běžnému použití, například u textů, které jsou psány pouze velkým písmem. Pro provedení truecasingu se využívá statistických modelů, které odhadují velikost písmen v daném slově pomocí míry a způsobu jeho výskytu ve zbytku textu, případně podle slov ve stejné větě.

2.2.4 Stemming

Stemming je jednou z metod, jak snížit celkový počet slov v dokumentu a tím zjednodušit a zpřesnit zpracování. Jedná se o vytvoření základního tvaru slova za pomoci algoritmu, který odřezává předpony a přípony a vrací kmen slova. To s sebou nese riziko, že na stejný kmen budou převedena i slova nesouvisející.

2.2.5 Lemmatizace

Lemmatizace je dalším postupem snížení počtu slov, tentokrát převedením různých tvarů slova na slovníkový tvar slova neboli lemma. Lemmatizátor funguje na principu využití databáze slov a jejich tvarů. I lemmatizace nese riziko, že bude slovo převedeno na základ, který neodpovídá významu původního slova, nebo budou na stejný základ převedena dvě nesouvisející slova. Ale vzhledem k tomu, že lemmatizace nahrazuje slova pomocí slovníku a nikoliv automaticky pomocí algoritmu, dosahuje obecně přesnějších výsledků, než stemming.

2.3 Vektorové modely

Pro zachycení sémantických vztahů mezi slovy, větami či dokumenty se zavádí různé matematické modely, které dokáží tyto vztahy jednoduše reprezentovat matematickým vyjádřením, které navíc umožňuje snadnou manipulaci. V těchto modelech slova nahrazujeme čísly, s kterými lze dále pracovat jednodušeji a výpočetní náročnost výrazně klesá.

Vektorové modely jsou založeny na mapování slov do N-rozměrného prostoru. Každý rozměr pak v závislosti na konkrétním modelu může odpovídat určitému tématu. Souřadnice slova v rozměru je pak tím větší, čím více slovo tématu odpovídá. Témata jsou určena automaticky na základě analyzovaného vstupního vzorku přirozeného jazyka. Další zajímavou vlastností vektorových modelů je možnost počítat s vektory slov jako s matematickými vektory (viz 2.3.3).

2.3.1 Bag of words

Bag of words (BoW) je jedním z modelů, který reprezentuje dokumenty obsahující vzorky přirozeného jazyka, a to za pomoci matematické reprezentace. BoW reprezentuje text jako neuspořádaný soubor slov, kde není důležité jejich pořadí, nýbrž pouze společný výskyt a četnost. Informaci o výskytu slov a jejich počtu ukládá do uspořádané n-tice, kterou lze pro potřeby vektorového modelu interpretovat jako souřadnice vektoru.

2.3.2 TF-IDF

Model BoW zaznamenává absolutní počet výskytů slova ve větě, což zvýhodňuje slova, která se obecně v jazyce opakují často. Tato slova zároveň nesou menší podíl informace, než slova, která se objevují méně často. V anglickém jazyce můžeme jako příklad uvést slova jako „the“ a „a“. Těmto slovům je vhodné přikládat menší váhu. TF-IDF je metodou, která umožní přiřadit slovům váhu, která je nepřímo úměrná počtu výskytů slova v analyzovaném vzorku přirozeného jazyka.

TF-IDF je zkratka pro „Term Frequency“ a „Inverse Document Frequency“, což jsou dvě složky výpočtu této reprezentace váhy slova. První složkou (TF) je počet výskytů slova v celém dokumentu. Tato hodnota se dá vypočíst různými způsoby. Nejjednodušší variantou je přímý součet výskytů daného slova v dokumentu; dále se dá využít normalizace pomocí logaritmu, či vydělením celkové délky dokumentu, aby nedošlo k nadhodnocení dlouhých dokumentů, kde se obecně vyskytuje více slov. Jedním ze způsobů výpočtů TF může být následující vzorec [7]:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{2.1}$$

kde $n_{i,j}$ je počet výskytů slova t_i v dokumentu d_j. Jmenovatel je součtem výskytů všech slov v dokumentu d_j. Druhou složkou (IDF) je důležitost slova. Se vzrůstajícím počtem výskytů slova klesá hodnota informace, kterou nese. IDF pro slovo i spočítáme dle vzorce:

$$idf_i = log\frac{|D|}{|\{j : t_i \in d_j\}|} \tag{2.2}$$

kde $|D|$ je počet dokumentů, které zpracováváme a jmenovatel reprezentuje počet dokumentů, které obsahují slovo i.

Výsledná hodnota pro TF-IDF je pak dána vzorcem:

$$TF\text{-}IDF = TF * IDF \tag{2.3}$$

2.3.3 Aritmetika se slovy ve vektorovém prostoru

Vektorový prostor má tu vlastnost, že slova významově podobná leží blízko sebe a naopak slova, jejichž významy blízké nejsou, leží od sebe dál[15].

Další důležitou vlastností vektorového prostoru je možnost počítat se slovy, převedenými do vektorového prostoru, jako s běžnými vektory v matematickém slova smyslu. Podobnost slov či vět se dá nejjednodušeji zjistit tak, že se zobrazí jejich vektory do některého z vektorových modelů, čímž se zjistí se jejich blízkost. Přímá vzdálenost slov není reprezentativním ukazatelem podobnosti, neboť ji značně ovlivňuje četnost slova v korpusu. Proto se častěji pro určení podobnosti slov užívá tzv. kosinová podobnost, což je míra podobnosti dvou vektorů, která se zjistí výpočtem kosinu úhlu, který svírají vektory slov, pro které je hledána podobnost.

Neméně zajímavou vlastností je aritmetika s vektory, kdy můžeme pomocí vektorů vyjadřovat sémantický význam slov. Například článek [11] popisuje následující operaci:

$$vector(Paris) - vector(France) =$$

výsledkem pak bude vektor, obecně symbolizující vlastnost „být hlavním městem“. Jeho přičtení k jinému státu pak dá vektor, který bude pravděpodobně velmi blízký vektoru hlavního města onoho státu.

2.4 Word2vec

Word2vec je sada modelů pro zpracování přirozeného jazyka a vyhledávání sémantické podobnosti slov. Obsahuje několik různých algoritmů, které se dají vzájemně kombinovat pro konkrétní situace. Základ modelu Word2vec tvoří dvě neuronové sítě, ze kterých si může uživatel vybrat. Jsou jimi Continuous bag of words a Continuous Skip-gram [9]. Obě mají pouze vstupní, skrytou a výstupní vrstvu. Vstupní a výstupní vrstva je reprezentací slov, která se trénují. Softmax regrese určuje rozložení pravděpodobnosti výskytu jednotlivých slov ze slovníku v blízkém okolí vstupního slova. Váhy neuronů jsou na začátku nastaveny náhodně a postupným učením v iteracích se pomocí algoritmu backpropagation (viz 2.4.1) hledá co nejmenší odchylka od očekávaných hodnot z trénovacího korpusu. Vstupní vrstva neuronových sítí modelu je matice S x V, výstupní pak V x S, kde S je počet slov ve slovníku korpusu a V velikost vektoru. Skrytá vrstva má velikost V. [13]

2.4.1 Algoritmus Backpropagation

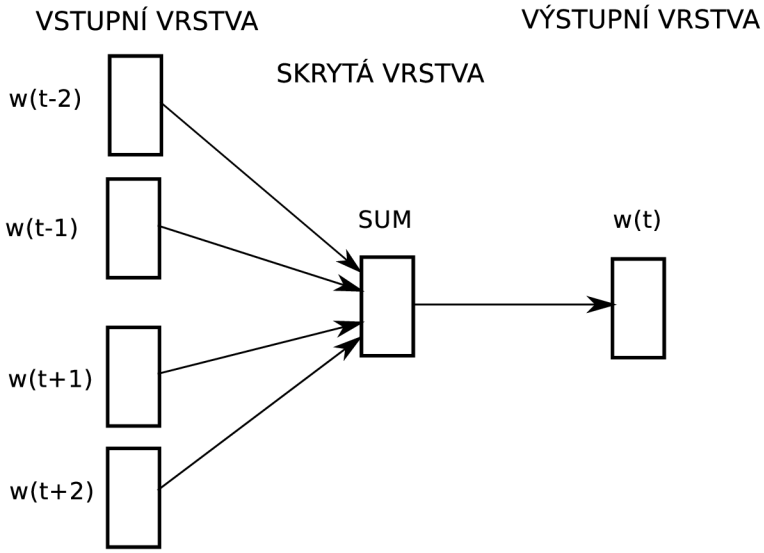
Backpropagation je algoritmus využívaný k učení neuronových sítí. Algoritmus porovnává vyhodnocené řešení s očekávaným a podle toho zjistí odchylku současného nastavení neuronové sítě. Na základě této odchylky spočte hodnotu, o kterou je třeba upravit váhy neuronů v síti, aby se odchylka co nejvíce minimalizovala. To je uskutečněno algoritmem Stochastic Gradient Descent[3]. Tato hodnota se nazývá gradient.

Použití tohoto algoritmu pro trénování neuronové sítě probíhá následovně. Nejdříve jsou váhy neuronové sítě nastaveny náhodně a neuronová síť pro všechna slova ve slovníku

vypočte pravděpodobnost, že se budou nacházet v blízkosti hledaného slova. Potom se tento výstup porovná s reálným rozložením slov v korpusu a váhy neuronů se upraví tak, aby se výsledek co nejvíce příblížil reálným datům. S každým opakováním tohoto procesu je neuronová síť v předpovídání slov přesnější. [14]

2.4.2 Continuous bag of words

Neuronová síť s názvem Continuous bag of words (CBOW) se snaží určit nejpravděpodobnější slovo podle několika okolních slov (kontextu).



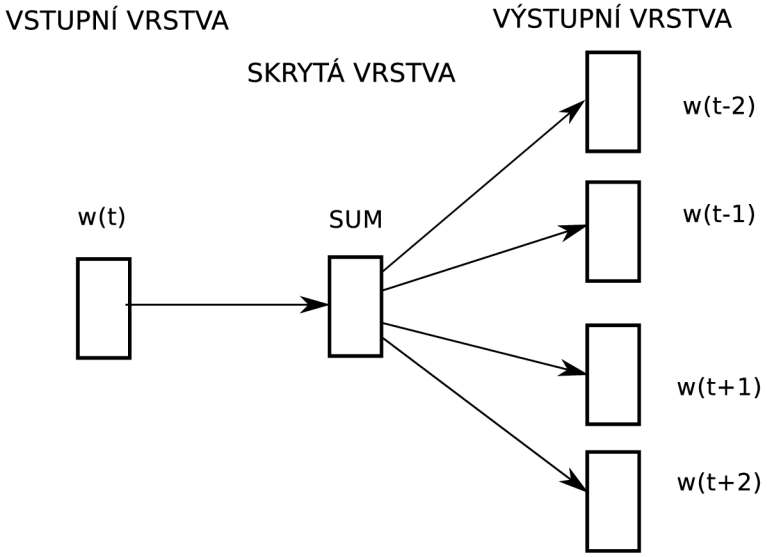
Obrázek 2.1: Schéma neuronové sítě CBOW

2.4.3 Continuous Skip-gram

Continuous Skip-gram (na obr. 2.2) se na rozdíl od CBOW snaží určit nepravděpodobnější slovo na základě aktuálního slova v okolí (kontext), je tedy náročnější na výpočet – z jednoho slova totiž počítá slov více, pro každé musí provést odhad pravděpodobnosti, že se bude nalézat v blízkosti aktuálního slova. Navíc v něm není stále stejný počet okolních slov, tzv. okno. Jeho velikost se určuje náhodně v předem stanoveném intervalu.

2.4.4 Hierarchical Softmax

Výpočet pravděpodobnosti, že se slovo bude nalézat v okolí hledaného slova, je prováděn pomocí funkce softmax. Klasická funkce softmax musí projít pro každé hledané slovo všechna slova ve slovníku, což by celý model neúměrně zpomalovalo. Proto word2vec využívá funkci *Hierarchical Softmax*. Tato funkce se od klasické softmax funkce liší zejména tím, že pro svůj běh nepotřebuje procházet celý slovník. Místo toho je při prvním spuštění modelu generován Huffmanův binární strom, což je stromová struktura, která mapuje slova tak, že často používaná slova jsou blíže kořenu stromu a méně častá dále. Tím přiřazuje binární kódy slovům na základě četnosti jejich výskytu tak, aby častěji užívaná slova měla kratší kód a méně často užívaná slova delší. Každé slovo ze slovníku je listem stromu a vede k němu jednoznačná cesta od jeho kořene. Při tvorbě tohoto stromu je každému uzlu určena



Obrázek 2.2: Schéma neuronové sítě Skip-gram

pravděpodobnost pro pravý a levý podstrom. Výsledná pravděpodobnost je pak součinem pravděpodobností na cestě k listu slova. Při použití funkce Hierarchical Softmax nejsou výstupní vektory neuronové sítě vektory slov nýbrž vektory náležející uzlům stromu. [13]

2.4.5 Negative Sampling

Funkce Negative Sampling aplikuje jiný postup pro výpočet pravděpodobnosti, že se slovo bude vyskytovat v okolí hledaného slova, bez potřeby procházet celý slovník. Místo upravování všech vektorů volí pouze vzorek, se kterým dále počítá. Velikost vzorku se pohybuje v rozmezí 2 až 5 slov pro velká vstupní data a 5 až 20 pro menší[5, 10]. Word2Vec užívá následující výpočet pro získání vzorku slov[13]:

$$E = -log\sigma({v'_{w_O}}^T h) - \sum_{w_i \in W_{neg}} log\sigma({v'_{w_i}}^T h), \tag{2.4}$$

kde w_O značí výstupní slovo a v'_{w_O} jeho výstupní vektor. h je výstupní hodnota skryté vrstvy, která se vypočítá následujícím způsobem pro CBOW model:

$$h = \frac{1}{C} \sum c = 1^C v_{w_c} \tag{2.5}$$

a pro skip-gram tímto způsobem:

$$h = v_{w_I} \tag{2.6}$$

Kapitola 3

Návrh systému

3.1 Požadavky

Cílem práce je systém, který pro zadané vstupní slovo či slova, nalezne slova a sousloví sémanticky podobná. Architektura systému musí umožňovat učení slov z libovolného korpusu a zároveň nabízet již předpřipravený natrénovaný model. Dále bude systém umožňovat hlubší analýzu vyhledaných pojmů na základě zpracovaných článků Wikipedie, což zahrnuje vyhledání příslušných článků a v nich klíčová slova pro výsledky vyhledávání. Systém dokáže pracovat i se zkratkami. Celý systém bude implementován v jazyce Python.

3.2 Použité nástroje

3.2.1 Natural Language Toolkit

Pro práci s textem byla zvolena knihovna Natural Language Toolkit (NLTK), která poskytuje celou řadu nástrojů pro zpracování a analýzu textů. Stejně jako zbytek systému je napsána v jazyce Python. Kromě jiného obsahuje nástroje pro tokenizaci, lemmatizaci, stemming, určování slovních druhů, a mnoho dalších. [16]

3.2.2 Gensim

Pro vyhledávání sémantické podobnosti slov byla užita knihovna gensim. Ta je rovněž napsána v jazyce Python a obsahuje všechny nástroje potřebné pro tvorbu a následnou analýzu modelu Word2Vec. Navíc obsahuje velké množství dalších nástrojů pro analýzu textu ve vektorovém prostoru jako je sada nástrojů pro provedení Latentní Dirichletovy alokace, Latentní sémantické analýzy a další. [12]

3.3 Tokenizace

V rámci tokenizace je třeba převést vstupní texty, formátované stylem Wikipedie, na seznam vět složených z tokenů. Nejprve je nutné odstranit všechny formátovací značky, které tvoří různé tabulky, vnitřní odkazy Wikipedie, úrovně nadpisů a další. Zároveň je třeba vymazat symboly, které nenesou žádnou důležitou informaci, jako jsou čárky, středníky, hvězdičky a další. Takto upravený korpus bude dále rozdělen na jednotlivé věty a slova za použití nástrojů ze sady NLTK. Pro snadné zpracování bude každá věta uložena na nový řádek a slova budou oddělena mezerami.

3.4 Truecasing

Další podstatnou částí předzpracování textu je kontrola velkých a malých písmen ve slovech. Pro anglický jazyk je typické psaní velkých písmen na začátku věty. Zároveň ale začátek věty často obsahuje vlastní jméno, proto bude použit nástroj pro určování slovních druhů ze sady NLTK, který dokáže vlastní jména určit. U těchto potom bude ponechána původní velikost písmen, u ostatních bude první písmeno převedeno na malé.

3.5 Tvorba n-gramů

Text po tokenizaci bude převeden na jednotlivá slova. Analýzou korpusu pouze na základě jednotlivých slov by se však část informace ztratila. Proto bude provedena statistická analýza textu a slova, která se vyskytují vedle sebe s větší četností, než by odpovídalo statistické pravděpodobnosti, budou spojena podtržítkem. Tato analýza však musí brát v úvahu fakt, že například výčty slov ve stejné kategorii se mohou objevovat často, ale slova v nich nejsou souslovími. Tato analýza bude provedena pomocí modulu Phrase z knihovny gensim.

3.6 Lemmatizace

Pro omezení počtu slov ve slovníku bude použita Lemmatizace z knihovny NLTK. Ta by měla dokázat převést slova na základní tvary tak, aby se neztratila podstatná část informace z textu. Neznámá slova budou ponechána v původní tvaru.

3.7 Korpus Wikipedie

Jako korpus pro trénování byl zvolen korpus Wikipedie, dostupný na serverech Fakulty informačních technologií VUT. Verze ze 7. dubna 2016 má velikost necelých 50 GB. Zjednodušená struktura textu je následující:

```
<page>
<title>Nadpis</title>
<revision>
<text>
Toto je text. Text může obsahovat i více vět.
Toto je také text.
</text>
</revision>
</page>
```

Soubor obsahuje velké množství dalších XML tagů, určujících vnitřní uspořádání Wikipedie. Text samotný obsahuje formátovací značky pro tvorbu tabulek, odkazů dovnitř i vně Wikipedie a sekcí, kde některé jsou součástí textu, a jiné nesou informace o článku jako takovém – například seznam zdrojů. Pro příklad, citace jsou v textu ve složených závorkách, odkazy na články ve Wikipedii v hranatých, kde je před svislou čárou uvedena stránka, na kterou se text odkazuje, a za ním řetězec, který se zobrazí čtenáři. Pojmy odpovídající nadpisu jsou v textu uvedeny třemi apostrofy.

3.8 Výběr článků

Pro trénování korpusu je třeba vybrat vzorek dat, který bude pro cílová data v co nejmenším množství slov co nejlépe reprezentovat cílovou oblast. Proto je třeba navrhnout skript, který vybere takové články, jež vhodně reprezentují zkoumanou oblast informačních technologií. Tento skript by měl články vybírat na základě vnitřní kategorizace článků ve Wikipedii. To by však mohlo vést k nedostatečně reprezentativnímu korpusu, neboť by nemusela být dostatečně reprezentována témata, která přímo neodpovídají vybraným tématům Wikipedie, ale přesto spadají do kategorie informačních technologií. Proto je třeba k těmto článkům vybrat další, které se budou věnovat informačním technologiím na základě analýzy jejich obsahu.

3.9 Analýza obsahu článků

Na základě značkovacího jazyka Wikipedie bude třeba z článků vybrat části, které odpovídají definicím jednotlivých pojmů. Z tohoto bude dále třeba extrahovat pojmy samotné a uložit je do databáze, která obsahuje všechna synonyma daného pojmu a určit, zda k danému pojmu existuje zkratka. Tento systém musí umožnit jednoduché vyhledání kteréhokoliv ze synonym odpovídajícího pojmu a zkratku pojmu.

3.10 Vyhodnocení výsledků

Pro zhodnocení správnosti modelu je třeba provést vyhodnocení správnosti výsledků. Zejména je třeba určit, pro které nastavení parametrů je dosahováno pro toto zadání nejvyšší přesnosti ve vyhledávání významově podobných slov i v následném vyhledávání článků. Zkoumané parametry jsou typem použité neuronové sítě (CBOW nebo Skip-gram) a minimálním počtem výskytu slova, který je nutný k tomu, aby se slovo zahrnulo do modelu.

Vyhodnocení proběhne náhodným výběrem 20 slov a sousloví z oblasti informačních technologií. Ke každému bude vyhledáno 10 nejbližších slov ve všech testovaných modelech. Na základě mých znalostí, encyklopedie pojmů z oblasti informačních technologií [2], a případně pomocí internetového vyhledávače Google, bude o každém výsledném slově rozhodnuto, zda skutečně souvisí se vstupním slovem. Pro všechny modely pak bude určena procentuální úspěšnost.

Zároveň pro nejúspěšnější model bude provedeno vyhodnocení vyhledávání článků a zpracování zkratek. Opět bude rozhodnuto, na kolik se výsledek modelu shoduje s realitou.

Kapitola 4

Implementace

Tato část se věnuje konkrétní implementaci jednotlivých součástí systému. Celý systém je implementován v jazyce Python 2, který byl zvolen pro svou přehlednou strukturu a širokou podporu. Verze 2 byla zvolena proto, že knihovna gensim, nainstalovaná na školních počítačích, je napsána právě pro verzi 2.

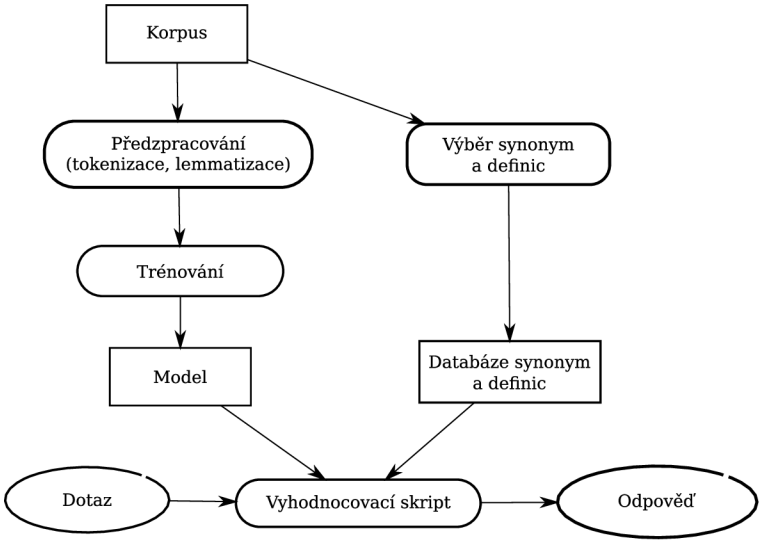
4.1 Požadavky na běh systému

Systém pro svou správnou funkci vyžaduje následující programy a knihovny:

- interpret jazyka python verze 2.7.6. nebo vyšší
- knihovnu gensim verze 0.12.1 nebo vyšší
- knihovnu NLTK verze 3.0.4 nebo vyšší
- balík NumPy verze 1.3. nebo vyšší
- balík SciPy verze 0.7. nebo vyšší

4.2 Schéma zpracování

Pro vytvoření modelu Word2Vec je třeba extrahovat vzorek přirozeného jazyka, který je rozdělen na jednotlivé věty a ty dále na slova. Pro další analýzu je pak třeba uložit definici jednotlivých pojmů a synonyma pojmů. Celé schéma přehledně zachycuje obrázek 4.1



Obrázek 4.1: Obecný způsob fungování systému

4.3 Zpracování korpusu

V této práci byl použit korpus anglické Wikipedie z data 7.4.2016 o celkové velikosti téměř 49 GB. Pro jeho zpracování je využit skript *createCorpus.py*, který využívá funkce *parseWiki*, *deleteMeta*, *deleteMetaDefinitions* a *synonyms*. Schéma tohoto skriptu je přehledně zobrazeno na obrázku 4.2. Tento skript čte po spuštění soubor *config.txt*, který musí obsahovat následující položky, každou na zvláštním řádku:

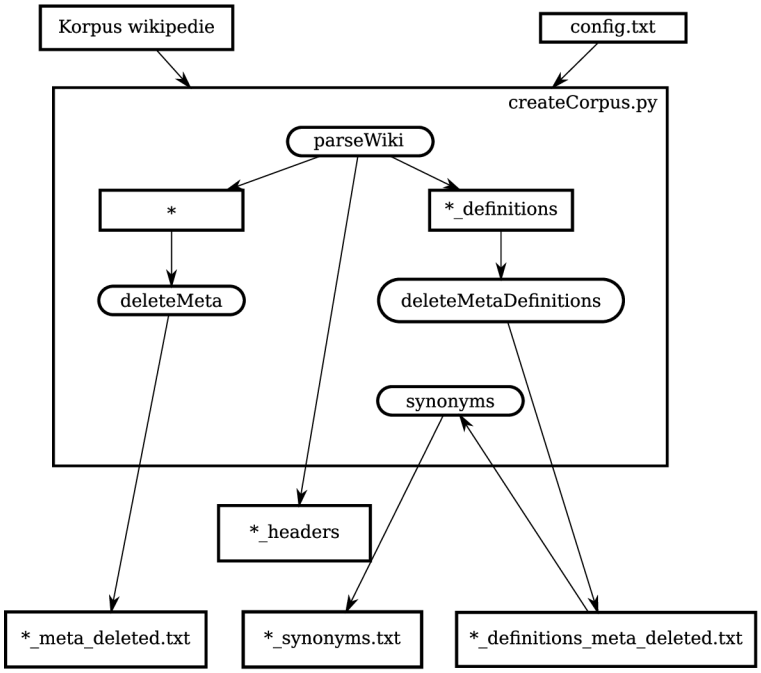
- path = 'cesta ke korpusu Wikipedie'
- name = 'název výstupních souborů skriptu'
- truecasing = (0|1)
  - může nabývat hodnot 0 nebo 1. 1 znamená, že bude použit truecasing prvních slov každé věty v korpusu.
- threshold = <0.2;0.004>
  - hodnota určuje, kolik článků bude zahrnuto do výsledného korpusu na základě klíčových slov ze seznamu concepts. Čím je hodnota vyšší, tím méně článků bude zahrnuto. Optimální nastavení se pohybuje v závislosti na počtu klíčových slov od 0.004 do 0.02.
- delete after = (0|1)
  - může nabývat hodnot 0 nebo 1. 1 znamená, že po skončení skriptu *createCorpus.py* budou smazány všechny nepotřebné soubory.
- concepts = {
  - následuje seznam klíčových slov, která definují zkoumanou oblast článků, které chceme z Wikipedie extrahovat. Každé slovo je na zvláštním řádku. Seznam je ukončen symbolem } na samostatném řádku.

- categories = {
  - následuje seznam názvů kategorií, které spadají do zkoumané oblasti. Články spadající do kterékoliv kategorie v tomto seznamu budou automaticky zahrnuty do vyhodnocení. Každé slovo je na zvláštním řádku. Seznam je ukončen symbolem *}* na samostatném řádku.

Na základě tohoto nastavení je volána funkce parseWiki, která vybírá články z korpusu Wikipedie. V každém článku je sečten počet výskytů slov ze seznamu *concepts* v konfiguračním souboru a vydělen celkovým počtem slov v článku. Je-li výsledná hodnota větší, než uvedená hodnota threshold, článek je do výsledku zahrnut; v opačném případě je zahozen. Pokud je článek v kategorii odpovídající některé z kategorií v seznamu *categories*, je automaticky vybrán a výpočet počtu slov se již neprovádí. Tento výstup je uložen do složky *resources* pod názvem zadaným v položce *name*. Dále je generován soubor s definicemi extrahovanými z jednotlivých článků. Tento soubor je pojmenován stejně jako soubor obsahující všechny extrahované články pouze s příponou *__definitions*. Definice jsou vybrány na základě značkovacího jazyka Wikipedie, kdy pojmy odpovídající nadpisu článku jsou zapsány ve třech apostrofech (”’takto”’). Za definici je považován první odstavec každého článku, který obsahuje alespoň jeden pojem ve třech apostrofech. Posledním generovaným souborem tohoto kroku je soubor s koncovkou *__headlines*. Ten uchovává informace o názvech všech článků zahrnutých do výsledného korpusu. Pokud byly vybrány na základě výskytu klíčových slov, na dalším řádku se nachází hodnota poměru klíčových slov oproti délce celého korpusu.

Dalším krokem ve zpracování korpusu je tokenizace. Tokenizaci provádí funkce *deleteMeta*. Vstupem této funkce je výstupní soubor, pojmenovaný podle položky *name* z předchozího kroku. Tato funkce odstraňuje z vybraných článků všechny formátovací značky, symboly a další prvky, které by mohly negativně ovlivnit učení korpusu. Z interpunkčních znamének jsou ponechány pouze čárky, které jsou důležité pro určení víceslovných výrazů a které budou odstraněny později. Dále jsou články pomocí nástroje *sent_tokenize* z knihovny NLTK rozděleny na jednotlivé věty, a ty jsou pomocí nástroje *word_tokenize* ze stejné knihovny rozděleny na jednotlivá slova. Na závěr je pro každé slovo provedena lemmatizace nástrojem wordnetLemmatizer, který je taktéž z knihovny NLTK. Výstupem je soubor s názvem uvedeným v položce *name* konfiguračního souboru *config.txt* a příponou *__meta__deleted.txt*, který obsahuje na každém řádku jednu větu. Slova ve větě jsou oddělena mezerami.

Podobný postup je proveden funkcí *deleteMetaDefinitions* pro soubor osbahující definice pojmů, pouze jsou ponechána všechna interpunkční znaménka a apostrofy, které jsou důležité pro výběr pojmů. Výstup je pak uložen do souboru s koncovkou *__definitions__meta__deleted.txt*. Z tohoto souboru jsou pak funkcí *synonyms* extrahovány pojmy a jejich synonyma, které se uloží do souboru s koncovkou *__synonyms.txt*. Extrakce synonym je provedena opět na základě značkovacího jazyka Wikipedie, kde je každé synonymum odpovídající názvu článku uvedo v prvním odstavci článku ve třech apostrofech. Pokud je některé ze synonym psáno celé velkým písmem, je považováno za zkratku pojmu, tato informace je zaznamenána.



Obrázek 4.2: Schéma skriptu createCorpus. Hvězdička je nahrazena názvem souborů, zadaným v souboru *config.txt* v položce *name*

4.4 Tvorba modelu

Pro určování podobnosti slov dle významu je třeba vygenerovat vhodný model, a to na základě souboru vygenerovaného například skriptem *createCorpus*, obsahujícího věty každou na jednom řádku, slova oddělená mezerami. To je provedeno skriptem *createModel.py*. Schéma skriptu je na obrázku 4.3. Tento skript má jediný povinný parametr, což je cesta k souboru s větami. Další, volitelné parametry, jsou následující:

- '-O' nebo '–output'
  - za tímto parametrem následuje název výstupních souborů. Pokud parametr není přítomen, užije se jako základ názvu výstupních souborů název souboru vstupního.
- '-SG' nebo '–skipGram'
  - je-li přítomen tento parametr, bude pro trénování užita neuronová síť skip-gram. V opačném případě bude užita síť CBOW.
- '-IW' nebo '–ignoreWords'
  - za tímto parametrem následuje číslo, udávající minimální počet výskytů slova v korpusu. Pokud se bude slovo vyskytovat méně často, systém jej ignoruje. Pokud není specifikováno, užije se hodnota 150.
- '-S' nebo '–size'
  - za tímto parametrem následuje číslo, určující dimenzionalitu výsledného vektorového prostoru. Standardní hodnota je 150.

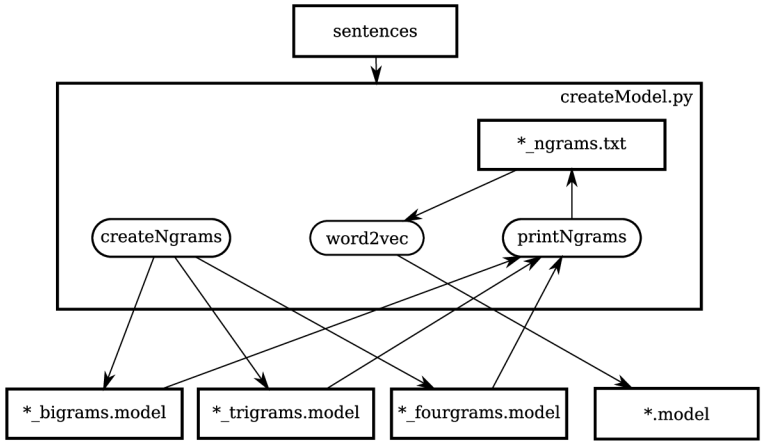
- '-C' nebo '–clean'
  - Je-li přítomen tento parametr, na závěr se smažou všechny nepotřebné soubory, vytvořené během běhu programu.

Tento skript volá postupně funkce *createNgrams*,*printNgrams* a *word2vec*. Funkce *createNgrams* vytváří objekty, které umožní zpracovávat sousloví jako jedno slovo. Jejím vstupem je korpus vět, pro který provede statistickou analýzu a vyhledá ta slova, která se vedle sebe objevují častěji, než by bylo statisticky pravděpodobné, což je provedeno objektem *phrases* z knihovny gensim. Pro kvalitní vyhledání takových sousloví je třeba zohlednit čárky, neboť jinak by slova, která se často objevují v seznamech či výčtech u sebe, byla zaznamenána jako sousloví. Pro získání více než dvouslovných termínů je tato analýza provedena opakovaně. Jejím výstupem jsou soubory s příponami *_ngram.model*, kde se nachází místo slova ngram slovo reprezentující maximální délku sousloví (bigram, trigram etc.). Analýza je provedena celkem třikrát po sobě pro nalezení sousloví až o délce 4 slov. Výstupem jsou tedy tři objekty. Spojení dvou slov je přijato jako n-gram, pokud splňuje nerovnost:

$$\frac{(ab - min) * N}{a * b} > práh \tag{4.1}$$

- a – počet výskytů slova a v korpusu
- b – počet výskytů slova b v korpusu
- ab – počet výskytů slov a a b vedle sebe
- min – minimální počet výskytů slovního spojení ab, aby bylo slovo započítáno jako slovní spojení, nastaven parametrem '-IW'
- N – celkový počet unikátních slov v korpusu
- $práh$ – hodnota, která musí být překročena, aby byla slova vybrána jako n-gram, nastavená na 15

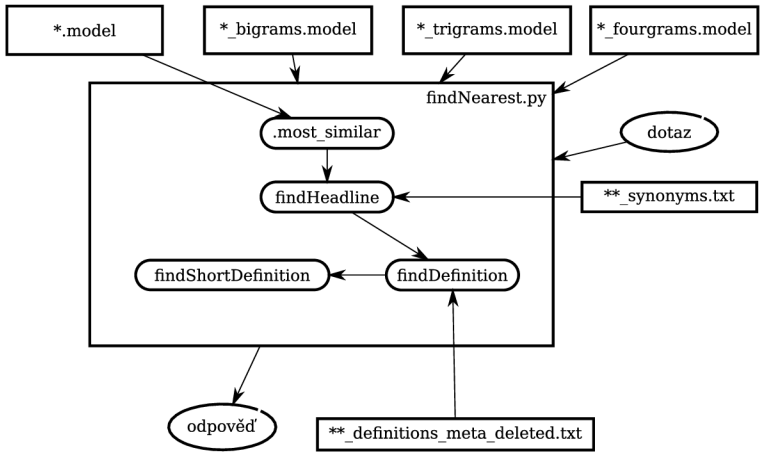
Dalším krokem je tvorba korpusu, který má sousloví zapsána tak, aby je mohl model zpracovávat jako jediné slovo. K tomu slouží funkce *printNgrams*, která za pomoci modelů vygenerovaných v předchozím kroku přepíše všechny výskyty víceslovných výrazů tak, že mezery mezi slovy nahradí podtržítky. Také ve výstupu odstraní čárky, které v této fázi již nejsou třeba. Zároveň je použit slovník nejčastějších anglických slov z knihovny NLTK, která jsou z korpusu odstraněna. Výstupem této funkce je soubor s příponou *_ngrams.txt*. Na závěr je volána funkce word2vec, která vytvoří z předem vytvořeného korpusu vektorový model. K tomu využívá objekt word2vec z knihovny gensim. Tento objekt je uložen pod názvem, zadaným prvním parametrem skriptu s koncovkou *.model*



Obrázek 4.3: Schéma skriptu createModel, místo hvězdičky je dosazeno pojmenování výstupních souborů

4.5 Vyhodnocování podobností

Podobnost slov či sousloví je vypočítána jako kosinová vzdálenost jejich vektorů. Pro vyhodnocení natrénovaných modelů byl vytvořen skript *findNearest.py*, který má dva vstupní parametry. Prvním je cesta k natrénovanému modelu Word2vec. Druhý parametr je předpona souborů vygenerovaných skriptem *createCorpus.py*. Tento skript provádí vyhodnocení natrénovaného modelu a vyhledávání definic a synonym na základě extrahovaných informací. Schéma skriptu je přehledně zachyceno na obrázku 4.4 a příklad spuštění a výpisu výsledků se nachází v příloze B.



Obrázek 4.4: Schéma skriptu findNearest. Místo * bude dosazen název modelu z prvního argumentu. Místo ** bude dosazen název vygenerovaných souborů z modulu *createCorpus.py*

Skript po načtení modelů přijímá slovo či sousloví, pro které, ve vektorovém prostoru modelu word2vec, najde nejbližší slova či sousloví pomocí metody objektu Word2Vec z knihovny gensim. Tento objekt je inicializován předem natrénovaným modelem a obsahuje metodu *most_similar*, která vrací slova, jež jsou nejblíže slovu zadanému. Její prototyp vypadá následovně:

```
most_similar(positive=[], negative=[], topn=10, restrict_vocab=None)
```

Argument *positive* obsahuje pole slov, pro která bude provedeno vyhledávání blízkých slov. Vektory slov v tomto poli se sečtou. Od nich budou odečteny vektory slov v poli *negative*. Argument *topn* určuje počet vyhledávaných slov. Při nastavení parametru *restrict_vocab* je zpracováno jen prvních N slov ze slovníku. Pro potřeby skriptu *findNearest.py* jsou nastavovány pouze parametry *positive* a *topn*. Ostatní jsou ponechány na implicitních hodnotách.

4.5.1 Vyhodnocení obsahu článků

Pro výsledné blízké pojmy pak skript dokáže najít zkratku, nebo, jedná-li se o zkratku, její úplné znění. Pokud se v databázi nachází článek, jehož název či některé ze synonym názvu odpovídá výsledku, či se mu blíží, skript vypíše název článku. Pokud je takových článků více, je vybrán ten, jehož název má nejmenší editační vzdálenost[1] od vyhledávaného termínnu. Funkce pro výpočet nejmenší editační vzdálenosti dvou slov byla převzata z [8], kde je distribuována pod licencí Creative Commons Attribution-ShareAlike 3.0 [1]. Dále, díky analýze textu definice článku, dokáže systém extrahovat klíčová slova související s pojmem.

Vyhledávání názvu je provedeno funkcí *findHeadline*. Ta prohledává soubor názvem odpovídajícím druhému argumentu skriptu s koncovkou *_synonyms.txt*, který obsahuje extrahované pojmy z vybraných článků Wikipedie. Pokud je vyhledaný pojem zkratkou, nebo naopak je v databázi poznámka o tom, že pojem má zkratku, je provedeno její vyhodnocení a vypsání.

Dle vyhledaných názvů článků jsou vyhledány funkcí *findDefinition* definice pojmů. Podle nich pomocí analýzy první věty definice vyhledána klíčová slova. To zajišťuje funkce *findShortDefinition*. Analýza věty je provedena pomocí nástroje *pos_tag* z knihovny NLTK. Tento nástroj určuje anglickým slovům jejich slovní druhy. Extrakce klíčových slov je provedena jako vyhledání první jmenné skupiny po prvním slovesu první věty definice. Jmennou skupinou je nepřerušená posloupnost podstatných a přídavných jmen spolu se spojkami.

[1] Minimální nutný počet záměn a přidání či odebrání znaků v jednom slově, aby bylo totožné se slovem druhým

Kapitola 5

Vyhodnocení

V této kapitole jsou popsány výsledky vyhledávání blízkých pojmů za použití různých nastavení, užitých při tvorbě vektorového modelu. Protože v dnešní době není k dispozici anotovaný korpus, zaměřený na počítačové vědy, bude vyhodnocení probíhat ručně. U modelů se zkoušelo nastavení parametrů neuronové sítě CBOW i skip-gram, a pro obě sítě byly použity dvě různé hodnoty minimálního výskytu slova pro jeho zahrnutí do modelu: 50 a 150 výskytů. Dále, pro nejúspěšnější model bylo vyhodnoceno vyhledávání klíčových slov a zpracování zkratek pro 5 vybraných slov a sousloví.

5.1 Postup vyhodnocování vyhledávání blízkých slov

Bylo náhodně vybráno 20 jednoslovných i víceslovných termínů z oblasti informačních technologií. Pro každý termín bude vybráno v každém modelu 10 nejbližších slov. Při vyhodnocení se zhodnotí, zda je nalezené slovo (nebo sousloví) opravdu významově blízké zadanému slovu.

graphical user interface	personal computer	Windows	USB
artificial intelligence	social network	Apple	Facebook
natural language processing	kernel	e-mail	algorithm
object-oriented programming	computer	PHP	extracting
wireless network	machine learning	cloud	CPU

Tabulka 5.1: Náhodně vybraná slova a sousloví

První sloupec tabulky obsahuje slova významově nejbližší ke slovu vyhledávanému. Druhý sloupec pak obsahuje vypočítanou podobnost se vstupním slovem. Následuje vyhodnocení vybraných výsledků, které zhodnocuje, proč byla slova započítána jako příbuzná či nikoliv. Pokud se v tabulce objevuje slovo, které není významově blízké slovu vyhledávanému, je ve vyhodnocení napsáno zdůvodnění, proč se toto slovo objevilo ve výčtu. Tabulky podobnosti pro všechy výsledky jsou v příloze A.

natural_language_processing	0.7048664093017578
AI	0.6982927322387695
theory	0.6941843628883362
neuroscience	0.6895197629928589
cognitive_science	0.6888583898544312
pattern_recognition	0.6255462169647217
information_retrieval	0.6168392896652222
machine_learning	0.6121153831481934
computer_science	0.6114770174026489
theoretical_computer_science	0.6035000681877136

Tabulka 5.2: artificial intelligence, CBOW, minimální počet výskytů slova 150

Pojem umělá inteligence *artificial intelligence* (zkratka *AI*) je poměrně rozsáhlý a obsahuje velké množství podoborů, proto mají sítě obecně problém definovat správná blízká slova. Pojmy *natural language processing*, *pattern recognition*, *information retrieval* a *machine learning* jsou jednotlivými obory umělé inteligence. Dále pak *computer science* a *theoretical computer science* jsou nadřazené vyhledávanému slovu, stejně jako *cognitive science*, která zkoumá vnímání a zpracování informací obecně. Slovo *theory* se do výsledků dostalo pravděpodobně proto, že se často objevuje v článcích o umělé inteligenci a různých zkoumaných teoriích. Pojem *neuroscience* je spjat s mnoha obory lidské činnosti a okrajově i s umělou inteligencí, proto se objevil v tomto výčtu.

AI	0.7854486107826233
cognitive_science	0.754462718963623
pattern_recognition	0.7279977202415466
natural_language_processing	0.7188439965248108
human-computer_interaction	0.7148535847663879
machine_learning	0.7141247987747192
artificial_life	0.6948615908622742
cybernetics	0.6910111308097839
theory	0.6765691637992859
robotics	0.6716815233230591

Tabulka 5.3: artificial intelligence, skip-gram, minimální počet výskytů slova 150

V této tabulce je navíc pojem *human-computer interaction*, který zahrnuje veškeré interakce mezi člověkem a počítačem a jedním z cílů umělé inteligence je právě co nejvíce zjednodušit tuto interakci. Dále je zde pojem *artificial life*, což je jedna z oblastí využití umělé inteligence. Pojem *cybernetics* je souhrné označení pro všechny řídící systémy, včetně těch, užívajících umělou inteligenci. Slovo *robotics* označuje vědu, která, mimo jiné, využívá v hojné míře i poznatků právě z oblasti umělé inteligence.

AI	0.7196123003959656
neuroscience	0.6899906992912292
natural_language_processing	0.6690598726272583
theory	0.6644940376281738
computer_science	0.6460037231445312
robotics	0.6429806351661682
theoretical_computer_science	0.6368385553359985
cognitive_science	0.6360152959823608
linguistics	0.6244781613349915
biology	0.621038556098938

Tabulka 5.4: artificial intelligence, CBOW, minimální počet výskytů slova 50

Opět zde můžeme nalézt stejná slova jako v předchozích tabulkách, která nejsou příbuzná vyhledávnanému pojmu. Navíc se sem dostala slova *biology* a *linguistic*. Druhé slovo má vzdálenou souvislost s pojmem pro její využití při zpracování přirozeného jazyka umělou inteligencí.

cognitive_science	0.751270592212677
AI	0.7413702011108398
robotics	0.7168495059013367
machine_learning	0.7082957029342651
natural_language_processing	0.7046661376953125
theory	0.6985126733779907
neuroscience	0.6947837471961975
computational	0.694147527217865
subfield	0.6903601288795471
human-computer_interaction	0.688478946685791

Tabulka 5.5: artificial intelligence, skip-gram, minimální počet výskytů slova 50

V této tabulce se navíc objevují slova *subfield* a *computational*, která jsou příliš obecná na to, aby mohla být považována za příbuzná, stejně jako pojem *theory*. Je však jasné, že umělá inteligence má mnoho podoborů a proto se zde slovo *subfield* vyskytuje.

data_mining	0.7227692604064941
artificial_intelligence	0.7048665285110474
pattern_recognition	0.6916810274124146
machine_learning	0.6772722005844116
information_retrieval	0.6694695949554443
machine_translation	0.6475945711135864
neuroscience	0.606237530708313
cognitive_science	0.6048372387886047
modeling	0.5956696271896362
mathematical	0.5919747352600098

Tabulka 5.6: natural language processing, CBOW, minimální počet výskytů slova 150

V tabulce lze pozorovat, že výstupní slova velmi dobře odpovídají příbuzným tématům k sousloví *natural language processing* (zpracování přirozeného jazyka). Sousloví *data mining*, *pattern recognition*, *machine learning* označují postupy, kterých se při zpracování přirozeného jazyka využívá. Pojmy *artificial intelligence*, *information retrieval* jsou pak nadřazené oblasti zpracování přirozeného jazyka. Pojem *machine translation* označuje oblast, ve které nachází zpracování přirozeného jazyka široké uplatnění. Slovo *modeling* se vztahuje k tvorbě různých matematických modelů pro zpracování jazyka, ale je příliš obecné na to, aby se dalo počítat jako úspěšně přiřazené. Sousloví *coginitive science* označuje vědu, zabývající se zpracováváním informací a se zpracováním přírozeného jazyka souvisí. Pojem *mathematical* se do výstupu dostal zřejmě pro matematické modely využité pro zpracování přirozeného jazyka. Pojem *neuroscience* je zde pravděpodobně proto, že má blízko analýze procesu zpracování jazyka jako takového.

machine_learning	0.7737637758255005
pattern_recognition	0.7557234168052673
information_retrieval	0.7465181946754456
data_mining	0.7227152585983276
artificial_intelligence	0.7188440561294556
cognitive_science	0.7004119753837585
machine_translation	0.6819087862968445
linguistics	0.6732376217842102
natural_language	0.6675568222999573
computational	0.6585089564323425

Tabulka 5.7: natural language processing, skip-gram, minimální počet výskytů slova 150

Tato tabulka nabízí podobné výsledky jako ta předchozí. Chybí zde však slova *modeling*, *neuroscience* a *mathematical*, která nebyla uznána jako podobná. Místo nich se zde nachází slova *linguistic*, které označuje analýzu jazyka jako takového, *natural language*, který je vlastně podmnožinou vstupního pojmu a slovo *computational*, které je zde kvůli častému spojení zpracování přirozeného jazyka s výpočetními modely, jako blízké jej však zařadit nelze.

pattern_recognition	0.7389746308326721
data_mining	0.7228318452835083
machine_learning	0.7161930799484253
machine_translation	0.690359354019165
artificial_intelligence	0.669059693813324
computer_vision	0.6475123167037964
information_retrieval	0.637970507144928
bioinformatics	0.6320317387580872
knowledge_representation	0.6229563355445862
computational_linguistics	0.6217935681343079

Tabulka 5.8: natural language processing, síť CBOW, minimální počet výskytů slova 50

Oproti předchozím tabulkám zde přibyla položka *computer vision*, která označuje problematiku na stejné úrovni, jako je *natural language processing*. Pojmy *knowledge representation* a *computational linguistic* označují podproblémy v rámci zpracování přirozeného

jazyka. Pojem *bioinformatics* je zde navíc, neboť se zabývá zejména shromažďováním biologických dat. Je to způsobeno pravděpodobně nízkým minimálním počtem výskytů slova pro zahrnutí do modelu. Slova, která se vyskytují málo často, pak mohou být zvýhodňována.

computational_linguistics	0.7585175037384033
machine_learning	0.7458779215812683
machine_translation	0.7439097762107849
pattern_recognition	0.7121661901473999
artificial_intelligence	0.7046661972999573
text_mining	0.701724112033844
information_retrieval	0.7016003131866455
computational_biology	0.6997258067131042
data_mining	0.6964676976203918
computational_chemistry	0.6885359883308411

Tabulka 5.9: natural language processing, síť skip-gram, minimální počet výskytů slova 50

V této tabulce přibyla navíc položka *computational chemistry*, která se sem dostala pravděpodobně také díky nízkému minimálnímu vyžadovanému počtu výskytu slova pro zahrnutí do modelu, stejně jako *computational biology*.

machine	0.6769828200340271
PC	0.6525312066078186
personal_computer	0.6289916634559631
laptop	0.5756670236587524
pc	0.5315970182418823
my_laptop	0.5241783261299133
microcomputer	0.5199245810508728
mainframe	0.4825839698314667
device	0.48196345567703247
emulator	0.46295225620269775

Tabulka 5.10: computer, síť CBOW, minimální počet výskytů slova 150

Z tabulky vidíme, že určování příbuzných slov pro takto obecný pojem dělají modelu největší problémy. Pojmy *PC*, *personal computer*, *microcomputer*, *laptop* a *mainframe* odpovídají poměrně úzce pojmu *computer*. Naopak pojmy *machine* a *device* jsou již vzdálenější, označují obecně jakékoliv zařízení. Pojem *my laptop* je způsoben špatným zpracováním modelu a nedá se započítat. Slovo *emulator* pak označuje počítačový programem, což je však význam natolik vzdálený, že se také nedá počítat.

machine	0.6883833408355713
personal_computer	0.6330452561378479
hardware	0.5833504796028137
remotely	0.5507045984268188
PC	0.5499809980392456
laptop	0.5473527908325195
scanner	0.5473410487174988
hooked	0.545274555683136
desktop	0.5384228825569153
modern	0.5310214757919312

Tabulka 5.11: computer, síť skip-gram, minimální počet výskytů slova 150

V této tabulce se objevují navíc pojem *hardware*, který s počítačem opravdu souvisí, ale také pojmy *remotely*, *scanner*, *hooked* a *modern*, které se pochopitelně v blízkosti slova *computer* hojně objevují, ale významově k němu mají daleko.

PC	0.6702527403831482
personal_computer	0.6418827772140503
machine	0.6410247087478638
laptop	0.562408983707428
microcomputer	0.5280351042747498
my_laptop	0.5261626839637756
device	0.5225568413734436
my_pc	0.5198904275894165
window_vista	0.49010568857192993
thin_client	0.4813291132450104

Tabulka 5.12: computer, síť CBOW, minimální počet výskytů slova 50

V této tabulce jsou navíc slova *windows vista* a *thin client*, které mají k vyhledávanému pojmu blízko. Na druhou stranu se zde však objevuje další pojem, který vznikl chybou při zpracování a tím je *my pc*.

machine	0.6798543334007263
personal_computer	0.6567084193229675
laptop	0.6371073722839355
PC	0.6112921237945557
hardware	0.5959345698356628
window_vista	0.5936201810836792
desktop	0.5913107395172119
my_laptop	0.5868101119995117
ethernet_cable	0.5820971131324768
my_pc	0.5789662003517151

Tabulka 5.13: computer, síť skip-gram, minimální počet výskytů slova 50

Tato tabulka obsahuje navíc pouze sousloví *ethernet cable*, který se opět určitě často objevuje poblíž slova *computer*, ale významově k němu má daleko.

Apple_Inc	0.6418497562408447
iPhone	0.6331830620765686
iPad	0.6263080835342407
iPod	0.6236118078231812
App_Store	0.6102774739265442
iPod_Touch	0.6026948094367981
MacBook	0.5750130414962769
Apple_Computer	0.5676814913749695
4th_generation	0.5661998987197876
iPhone_iPad	0.5580594539642334

Tabulka 5.14: Apple, síť CBOW, minimální počet výskytů slova 150

Výsledky velmi dobře odpovídají příbuzným pojmům pro *Apple*. Pojmy *iPhone*, *iPad*, *iPod*, *App Store*, *iPod Touch*, *MacBook* a *Apple Computer* odpovídají výrobkům či službám společnosti *Apple*, která se celým názvem jmenuje *Apple Inc*. Pouze poslední dva termíny nemůžeme počítat jako správné výsledky. Pojem *4th generation* je zde zejména díky nové nabídce společnosti *Apple*, kterou je 4th generation Apple TV. Spojení *iPhone iPad* je způsobeno špatným předzpracováním, kdy se někdy může stát, že takováto slova, která se objevují vedle sebe často ve výčtech, se uloží jako ustálené slovní spojení.

iPhone	0.7904753088951111
App_Store	0.7724881172180176
iPad	0.7416660785675049
iPhone_iPod	0.7115271091461182
iPod_Touch	0.7076528072357178
Blackberry	0.7002319693565369
iPhone_iPad	0.700178861618042
iPod_touch	0.6795750260353088
iOS	0.6775909066200256
Apple_Inc	0.6701352000236511

Tabulka 5.15: Apple, síť skip-gram, minimální počet výskytů slova 150

Oproti předchozí tabulce je v této navíc pojem *iOS*, který odpovídá operačnímu systému zařízení společnosti *Apple* a *Blackberry*, což je konkurenční firma na poli mobilních telefonů. Opět si zde můžeme povšimnout pojmů *iPhone iPod* a *iPhone iPad*, které vznikly kvůli špatnému předzpracování a nedají se počítat jako úspěšně nalezená příbuzná slova.

iPhone	0.627647340297699
Apple_Inc	0.6177367568016052
1st_generation	0.6088740229606628
iPad	0.605204701423645
Microsoft	0.6008187532424927
iPhone_iPad	0.5755926966667175
Amazon	0.5740593075752258
Samsung	0.5719084143638611
app	0.5650652050971985
Applecom	0.5632712841033936

Tabulka 5.16: Apple, síť CBOW, minimální počet výskytů slova 50

Zde navíc přibyly konkurenční společnosti *Microsoft*, *Samsung* a společnost *Amazon*, která pro Apple distribuuje některé jeho produkty. Výraz *Applecom* je pravděpodobně způsoben špatným zpracováním odkazu apple.com. Pojem *app* je zkratkou ke slovu application a obecně se často vyskytuje jako souhrné pojmenování aplikací na mobilní telefony. Se společností *Apple* však souvisí pouze okrajově. Pojem *1st generation* se zde objevil pravděpodobně jako odkaz na první generace všech možných produktů firmy *Apple*.

iPhone	0.7786290049552917
iPad	0.7361947894096375
iPod_touch	0.7148223519325256
Apple_Inc	0.7047652006149292
Apple_Watch	0.7020667195320129
iPhone_iPad	0.6978839039802551
App_Store	0.6839324235916138
iPod_Touch	0.6772884726524353
Apple_iPhone	0.6767436861991882
iPad_iPhone	0.6742246747016907

Tabulka 5.17: Apple, síť skip-gram, minimální počet výskytů slova 50

Jen pro doplnění bez započítání do celkových statistik je uvedena tabulka 5.18, která ukazuje výsledky modelu skip-gram s minimálním počtem výskytů slova 50 pro slovo *apple*. Tato síť měla pro slovo *Apple* 90% úspěšnost v určení příbuzných pojmů.

fruit	0.8412255644798279
grape	0.8177860975265503
tart	0.8056857585906982
pear	0.7994844317436218
juice	0.7957667708396912
sweet	0.7916285395622253
dessert	0.7886976003646851
pineapple	0.7861086130142212
potato	0.7856360077857971
cider	0.7813228964805603

Tabulka 5.18: apple, síť skip-gram, minimální počet výskytů slova 50

I přesto, že se korpus zaměřuje zejména na články z oblasti počítačových technologií, dokáže pro nízkou hodnotu minimálního počtu výskytů vyhledávat příbuzné pojmy i pro obecnější výrazy. Tato tabulka ukazuje, jak důležitý je dobře provedený truecasing. Pokud by se vůbec neprovedl, bylo by slovo *apple* ve významu jablko na začátku věty započítáno stejně jako slovo *Apple* ve smyslu společnosti. Slova z této tabulky by se pak přiblížila výsledkům pro vyhledávání pojmu *Apple*. Pokud by byla v korpusu převedena všechna písmena na malá, výsledky pro vyhledávání slov *Apple* a *apple* by se zamíchaly mezi sebe.

5.2 Shrnutí výsledků vyhledávání příbuzných slov

Zde se nachází souhr výsledků a vypočítaná přesnot pro jednotlivé modely. V tabulce 5.19 je zaznamenán počet slov z výsledku, která jsou opravdu příbuzná vyhledávanému slovu.

neuronová síť	CBOW		skip-gram	
minimální počet výskytů slova	50	150	50	150
graphical user interface	9	10	9	9
artificial intelligence	6	8	6	8
natural language processing	8	7	8	9
object-oriented programming	10	7	8	7
wireless network	10	9	10	9
personal computer	7	8	8	8
social network	9	9	9	10
kernel	10	10	10	10
computer	7	8	7	6
machine learning	10	10	8	9
Windows	10	10	10	10
Apple	7	8	9	8
e-mail	10	10	10	10
PHP	10	10	10	10
cloud	9	10	9	10
USB	10	10	10	10
Facebook	9	10	9	9
algorithm	10	10	10	10
extracting	9	10	7	9
CPU	10	10	10	10
průměr	9	9,2	8,85	9,05
procentuální úspěšnost	90%	92%	88,5%	90,5%

Tabulka 5.19: Výsledky pro náhodně vybraná slova

Všechny modely celkem dosáhly průměrné úspěšnosti 90% bez statisticky významných rozdílů mezi nimi. Takto vysoká úspěšnos může být z části způsobena tím, že se celé modely zaměřují pouze na úzký okruh témat. Při takto malém rozdílu výsledků nelze určit, které nastavení je nejlepší, neboť takto drobné odchylky mohou být způsobeny i občasnou chybou ve vyhodnocení, či náhodou. Malý rozdíl v úspěšnosti jednotlivých modelů může být způsoben nedostatečně rozdílným nastavením testovaných parametrů.

5.3 Vyhodnocení vyhledávání článků podle nalezených slov a zpracování zkratek

Pro výsledky 5 z 20 vybraných slov bylo užito vyhledávání článků z Wikipedie a zpracování zkratek. Jejich vyhodnocení se věnuje následující odstavec. Pro každé relevantní slovo je započítáno, zda byla nalezena ekvivalentní zkratka a odpovídající článek. V každé tabulce je pro jednotlivý výsledek uvedena v druhém sloupci buď jeho zkratka, nebo naopak rozepsán celý název zkratky (v závislosti na tom, zda je v prvním sloupci zkratka nebo plný název). Ve třetím sloupci se nachází název článku, který byl výsledku přiřazen. Pokud je plný popis zkratky zároveň názvem článku, jsou sloupce dva a tři sloučeny. Do výsledku jsou zahrnuta jen slova, která byla v předchozím hodnocení určena jako příbuzná s vyhledávaným pojmem. Vyhledávání je provedeno na modelu CBOW s minimálním počtem výskytu slov 150, který měl v předcházejícím textu největší úspěšnost vyhodnocování blízkých slov.

toolkits	-	BASIC extension
GUI	Graphical user interface	
widget	-	Widget (GUI)
command_line_interface	-	-
user_interface	UI	User interface
graphical_interface	-	-
frontend	-	-
toolkit	-	Fox toolkit
window_manager	-	Window manager
UI	User interface	

Tabulka 5.20: Vyhladené články pro pojem graphical user interface

V této tabulce je hned několik nesprávně určených článků. Pro pojem *toolkits* byl určen článek *BASIC extension* z důvodu, že žádný článek nesoucí název *toolkits* se na Wikipedii nevyskytuje. Článek *BASIC extensions* však nese jako jeden z podnázvů i název *BASIC toolkits*, který má ze všech možností nejbližší editační vzdálenost ke slovu *toolkits*. Podobný problém nastal i u vyhledávání článku k pojmu *toolkit*. Článek k pojmu *command_line_interface* nebyl vůbec nalezen, což je způsobeno v pomlčce v názvu článku na Wikipedii na což není vyhledávací skript připraven. Protože nebyl nalezen článek, nebylo možné dohledat ani odpovídající zkratki *CLI*. Článek pro pojem *graphical_interface* nebyl nalezen, protože článek na Wikipedii nese název *Graphical user interface*. Pojem *frontend* odpovídá nejlépe článku *Front and back ends*, který nemohl být nalezen, protože žádný z pojmů v definici tohoto článku přesně neodpovídá vyhledávanému pojmu. Ostatní články byly určeny správně.

data_mining	-	Data mining
artificial_intelligence	AI	Artificial intelligence
pattern_recognition	-	Pattern recognition
machine_learning	-	Machine learning
information_retrieval	IR	Information retrieval
machine_translation	MAHT	Machine translation
cognitive_science	-	Embodied cognitive science
modeling	-	3D modeling

Tabulka 5.21: Vyhladené články pro pojem natural language processing

V této tabulce můžeme vidět, že většina nalezených pojmů přímo odpovídá článkům na Wikipedii, proto nebyl s jejich určením žádný větší problém. Články *Embodied cognitive science* a *3D modeling* byly vybrány proto, že články odpovídající přímo výsledným pojmům nebyly zahrnuty do zpracovávaného modelu. Zkratky byly vyhodnoceny dobře až na zkratku *MAHT*, která odpovídá pojmu *machine-aided human translation*. Byla takto určena, protože se vyskytuje v prvním odstavci článku *Machine translation* jako varování na možnou záměnu pojmů.

OOP	Object-oriented programming	
declarative	-	Declarative programming
inheritance	-	inheritance
higher-level	-	-
functional_programming	-	Functional programming
imperative	-	Imperative programming
Smalltalk	-	Smalltalk

Tabulka 5.22: Vyhledané články pro object-oriented programming

Jediná zkratka *OOP* v tomto výstupu byla vyhodnocena správně. Článek Inheritance může odpovídat více článkům na Wikipedii. Jinak vyhledané pojmy odpovídají článkům, takže vyhledávání bylo přímočaré.

Python	-	Python (programming language)
Perl	-	Perl
JavaScript	-	JavaScript
Javascript	-	JavaScript
Java	-	Java
VBScript	-	VBScript
Objective-C	-	Objective-C
Tcl	Transaction Control Language	
script	-	SCRIPT (markup)
Lua	-	Lua (programming language)

Tabulka 5.23: Vyhledané články pro PHP

V této tabulce, kromě pojmů odpovídající článkům, můžeme vidět, že skript nalezl odpovídající články k pojmu i přesto, že Wikipedia obsahuje více definic konkrétního pojmu v závislosti na zkoumané oblasti. To je způsobeno tím, že ostatní definice nebyly zahrnuty do korpusu při jeho tvorbě. Konkrétně se jedná o pojmy *Lua* a *Python*, respektive

jim odpovídající články *Lua (programming language)* a *Python (programming language)*. Chybou v tomto vyhodnocení je přiřazení článku *SCRIPT (markup)* k pojmu script. Toto nastalo proto, že byla nalezena přesná shoda pojmu s pojmem před závorkou, a tak se již nepokračovalo v prohledávání.

processor	-	Processor
GPU	Graphics processing unit	
microprocessor	-	Microprocessor
cpu	Central processing unit	
chipset	-	Chipset
graphic_card	-	-
superscalar	-	Superscalar processor
FPU	Floating-point unit	
ARM	ARM architecture	
x86_processor	-	-

Tabulka 5.24: Vyhledané články pro CPU

Zkratky jsou v tomto případě vyhledány dobře, až na případ ARM, kde výraz *ARM architecture* není plným názvem zkratky, jen je zkratka jeho částí. Článek pro pojem *graphic card* není nalezen, protože na Wikipedii je odpovídající článek zapsán pod nadpisem *Video card*. Článek pro *x86_processor* nebyl nalezen, protože tento pojem se spíše pouze pojí s ostatními výrazy. Na Wikipedii proto ani odpovídající článek neexistuje. Pojem *superscalar* se pojí v článcích Wikipedie pouze s článkem *Superscalar processor*, a proto je jeho určení správné.

5.4 Shrnutí výsledků vyhledávání článků

Zde se nachází souhrn výsledků vyhodnocení vyhledávání článků a zpracování zkratek.

	Celkem	Určeno špatně	Neurčeno	Určeno správně
Určené zkratky	12	2	1	10
Určené články	39	6	5	34

Tabulka 5.25: Výsledky vyhodnocení vyhledávání článků a zpracování zkratek

Z tabulky vyplývá, že vyhledávání článků je poměrně přesné, zejména proto, že vyhledané pojmy většinou přesně odpovídají článkům z Wikipedie. Celkem bylo vyhledávání provedeno pro 45 pojmů, zbylé pojmy byly z analýzy vypuštěny, protože vyhledané pojmy nebyly příbuzné se zadanými slovy. Z nich byl správně článek vyhledán pro 86,7% článků. Celkem se mezi zkoumanými pojmy objevilo 13 prvků, pro které měla být určena buď zkratka, nebo plný název odvozený ze zkratky. To se povedlo v 10 případech.

Kapitola 6

Závěr

6.1 Dosažené výsledky

S použitím knihovny gensim a jazykového nástroje NLTK byl v jazyce Python vytvořen systém, umožňující zpracovávat korpus Wikipedie. Tento systém umožňuje jeho analýzu pomocí vektorového modelu Word2vec a statické analýzy, která určuje ustálená slovní spojení. Zároveň je systém modulární, takže je možné jej použít na libovolné texty, které budou převedeny do formátu zpracovatelného systémem. Systém dále obsahuje vyhodnocovací část, která umožní nad zpracovaným modelem pokládat dotazy na významově podobná slova a pro výsledky vyhledávat odpovídající články na Wikipedii.

Dále byly vygenerovány modely s různým nastavením a jejich výstupy byly porovnány. Úspěšnost všech modelů byla neočekávaně podobná, což může být způsobeno nedostatečně rozdílnými parametry nastavení jednotlivých modelů. Celková vysoká úspěšnost může být částečně způsobena tím, že se všechny modely zaměřují pouze na úzký okruh témat. Průměrná úspěšnost modelu v určování významově blízkých slov je 90,5%, nejúspěšnější model dosahuje úspěšnosti 92%, nejméně úspěšný pak 88,5%.

6.2 Přínos práce

V současné době neexistuje žádný podobný systém který by umožňoval tvorbu a zpracování korpusu zaměřeného na určitou tematickou oblast a jeho použití pro tvorbu modelu Word2vec. Přínosem je tedy samotný systém. Dále práce přináší srovnání modelů, které byly naučeny za použití různého nastavení.

6.3 Možnosti rošíření

Systém je složen z modulů, a to umožňuje jeho další vylepšení či nahrazení některých částí. Například je možné přidat podporu pro jiný korpus, nebo naopak tvorbu jiného vektorového modelu pro jeho zpracování. Stejně tak je možné upravovat či nahrazovat zvlášť i samostatné funkce a například navrhnout lepší funkci pro vyhledávání klíčových slov či jinou funkci pro statistickou analýzu a vyhodnocení ustálených slovních spojení.

Literatura

[1] CreativeCommons.
URL https://creativecommons.org/licenses/by-sa/3.0/

[2] Technology Definitions and Cheat Sheets from Whatls.com The Tech Dictionary and IT Encyklopedia. TechTarget, [Online; navštíveno 16.5.2012].
URL http://whatis.techtarget.com/

[3] Bottou, L.: Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, Springer, 2010, s. 177–186.

[4] Chowdhury, G. G.: Natural language processing. *Annual review of information science and technology*, ročník 37, č. 1, 2003: s. 51–89.

[5] Goldberg, Y.; Levy, O.: word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[6] Lita, L. V.; Ittycheriah, A.; Roukos, S.; aj.: Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, Association for Computational Linguistics, 2003, s. 152–159.

[7] Manning, C. D.; Raghavan, P.; Schütze, H.; aj.: *Introduction to information retrieval*, ročník 1. Cambridge university press Cambridge, 2008.

[8] Matthews, C. P.: Algorithm Implementation/Strings/Levenshtein distance Wikibooks, open books for an open world. online, [Online; navštíveno 2.5.2012].
URL https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance

[9] Mikolov, T.; Chen, K.; Corrado, G.; aj.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Mikolov, T.; Sutskever, I.; Chen, K.; aj.: Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013, s. 3111–3119.

[11] Mikolov, T.; Yih, W.-t.; Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, 2013, s. 746–751.

[12] Řehůřek Radim: gensim Topic modelling for humans. online, [Online; navštíveno 16.5.2012].
URL https://radimrehurek.com/gensim/

[13] Rong, X.: word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[14] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J.: Learning representations by back-propagating errors. *Cognitive modeling*, ročník 5, č. 3, 1988: str. 1.

[15] Sahlgren, M.: The distributional hypothesis. *Italian Journal of Linguistics*, ročník 20, č. 1, 2008: s. 33–54.

[16] Steven Bird and Ewan Klein and Eduard Loper: *Natural Language Processing with Python*. O'Reilly, 2009, ISBN 0596516495.

Přílohy

Seznam příloh

Příloha A

Tabulky podobností

A.1 CBOW síť, minimální výskyt slova 150x

Následují tabulky pro vyhledávaná slova pro neuronovou síť CBOW a nastavení minimálního počtu slov 150 výskytů.

toolkits	0.7354674935340881
GUI	0.7315068244934082
widget	0.6922416090965271
command_line_interface	0.67087721824646
user_interface	0.6708413362503052
graphical_interface	0.6483340263366699
frontend	0.6443015933036804
toolkit	0.6379028558731079
window_manager	0.6355262994766235
UI	0.630246639251709

Tabulka A.1: graphical user interface

natural_language_processing	0.7048664093017578
AI	0.6982927322387695
theory	0.6941843628883362
neuroscience	0.6895197629928589
cognitive_science	0.6888583898544312
pattern_recognition	0.6255462169647217
information_retrieval	0.6168392896652222
machine_learning	0.6121153831481934
computer_science	0.6114770174026489
theoretical_computer_science	0.6035000681877136

Tabulka A.2: artificial intelligence

data_mining	0.7227692604064941
artificial_intelligence	0.7048665285110474
pattern_recognition	0.6916810274124146
machine_learning	0.6772722005844116
information_retrieval	0.6694695949554443
machine_translation	0.6475945711135864
neuroscience	0.606237530708313
cognitive_science	0.6048372387886047
modeling	0.5956696271896362
mathematical	0.5919747352600098

Tabulka A.3: natural language processing

OO	0.7566062808036804
object_oriented	0.7130606770515442
object-oriented	0.7107299566268921
OOP	0.6844682097434998
declarative	0.6575562953948975
inheritance	0.6468064188957214
higher-level	0.6413199305534363
functional_programming	0.6306642889976501
imperative	0.6205624938011169
Smalltalk	0.6134539246559143

Tabulka A.4: object-oriented programming

wireless	0.7560281753540039
network	0.7474242448806763
wifi	0.7119653224945068
telephone_line	0.668020486831665
wireless_router	0.6670366525650024
wirelessly	0.6657297611236572
gateway	0.6653194427490234
hotspot	0.6625641584396362
internet_connection	0.6621872782707214
wireless_LAN	0.6615471839904785

Tabulka A.5: wireless network

microcomputer	0.7688449025154114
PC	0.6887520551681519
video_game_console	0.6384503245353699
computer	0.6289917230606079
mainframe	0.626389741897583
low-end	0.6216510534286499
mainframe_computer	0.6209869980812073
minicomputer	0.6174967288970947
IBM_PC	0.6155154705047607
high-end	0.6035906672477722

Tabulka A.6: personal computer

social_medium	0.7170822620391846
social_networking	0.6947005987167358
social_networking_site	0.6621918082237244
discussion_forum	0.6001526117324829
e-commerce	0.5793967843055725
portal	0.547183632850647
Foursquare	0.5459847450256348
online_dating	0.5434122681617737
Reddit	0.5299060344696045
online	0.5276097655296326

Tabulka A.7: social network

user-mode	0.6814433336257935
Linux_kernel	0.6423948407173157
bootloader	0.6003885269165039
Kernel	0.5976055860519409
hypervisor	0.5958376526832581
boot_loader	0.585541844367981
microkernel	0.5702061653137207
monolithic	0.5652428269386292
FreeBSD	0.5536016225814819
virtual_machine	0.5517433881759644

Tabulka A.8: kernel

machine	0.6769828200340271
PC	0.6525312066078186
personal_computer	0.6289916634559631
laptop	0.5756670236587524
pc	0.5315970182418823
my_laptop	0.5241783261299133
microcomputer	0.5199245810508728
mainframe	0.4825839698314667
device	0.48196345567703247
emulator	0.46295225620269775

Tabulka A.9: computer

supervised_learning	0.6924189925193787
data_mining	0.68646240234375
natural_language_processing	0.6772719621658325
statistical	0.6534110903739929
information_retrieval	0.6445767283439636
stochastic	0.6372207999229431
pattern_recognition	0.6343125700950623
computational	0.6210997104644775
artificial_intelligence	0.6121153235435486
analytic	0.6088345646858215

Tabulka A.10: machine learning

Windows_XP	0.8313494324684143
XP	0.8267267942428589
Microsoft_Windows	0.8077729940414429
OS	0.7953020930290222
Vista	0.7910439372062683
Mac	0.7712206840515137
Windows_Vista	0.7704282999038696
Mac_OS	0.7393714785575867
Windows_98	0.7337403297424316
Ubuntu	0.7300837635993958

Tabulka A.11: Windows

Apple_Inc	0.6418497562408447
iPhone	0.6331830620765686
iPad	0.6263080835342407
iPod	0.6236118078231812
App_Store	0.6102774739265442
iPod_Touch	0.6026948094367981
MacBook	0.5750130414962769
Apple_Computer	0.5676814913749695
4th_generation	0.5661998987197876
iPhone_iPad	0.5580594539642334

Tabulka A.12: Apple

email	0.8895456790924072
mail	0.799058198928833
message	0.7570008039474487
inbox	0.6779130697250366
Email	0.6663457751274109
SMS	0.6511695384979248
unsolicited	0.628455400466919
mail_server	0.6265214085578918
fax	0.6232337355613708
gmail	0.6217942237854004

Tabulka A.13: e-mail

Python	0.7832421660423279
Perl	0.7773458361625671
JavaScript	0.7697815299034119
Javascript	0.7687159776687622
Java	0.7326694130897522
VBScript	0.7231535911560059
Objective-C	0.7080575823783875
Tcl	0.7078874111175537
script	0.6889826655387878
Lua	0.6856329441070557

Tabulka A.14: PHP

cloud-based	0.6876357793807983
cloud_computing	0.6793816685676575
SaaS	0.6614523530006409
enterprise	0.61008220911026
web_hosting	0.5811865329742432
provisioning	0.5749561786651611
provider	0.5590795278549194
Internet-based	0.5562770962715149
VOIP	0.5534878969192505
web-based	0.5521165728569031

Tabulka A.15: cloud

usb	0.739787757396698
SATA	0.7233502268791199
USB_port	0.7228453755378723
adapter	0.710137128829956
Bluetooth	0.7035708427429199
ethernet	0.7031010985374451
Firewire	0.7023100256919861
OTG	0.697553813457489
parallel_port	0.6926457285881042
SCSI	0.6918773651123047

Tabulka A.16: USB

Twitter	0.7594847083091736
MySpace	0.7218582630157471
Instagram	0.7104974985122681
LinkedIn	0.6950193643569946
YouTube	0.6938547492027283
Tumblr	0.6710572242736816
Yahoo	0.6691644787788391
Myspace	0.6617641448974609
Reddit	0.6443433165550232
Flickr	0.6418399810791016

Tabulka A.17: Facebook

heuristic	0.7828882336616516
sorting_algorithm	0.772415280342102
method	0.7570831775665283
approximation_algorithm	0.7239567637443542
decision_tree	0.7107872366905212
depth-first_search	0.7043591737747192
technique	0.6953004002571106
quicksort	0.6823855042457581
formulation	0.6779823303222656
optimization_problem	0.6698823571205139

Tabulka A.18: algorithm

storing	0.7054656147956848
extracted	0.6965122818946838
extract	0.6918213367462158
compressing	0.6807516813278198
retrieving	0.6389070153236389
extraction	0.6369398236274719
stored	0.6253810524940491
manipulating	0.5932835340499878
containing	0.5891938805580139
collecting	0.5862526297569275

Tabulka A.19: extracting

processor	0.8655214905738831
GPU	0.7658922076225281
microprocessor	0.7230755686759949
cpu	0.695073664188385
chipset	0.6908401250839233
graphic_card	0.6636040806770325
superscalar	0.6527196168899536
FPU	0.6440349817276001
ARM	0.6418676972389221
x86_processor	0.6409181952476501

Tabulka A.20: CPU

A.2 Síť skip-gram, minimální výskyt slova 150x

Následují tabulky pro vyhledávaná slova pro neuronovou síť skip-gram a nastavení minimálního počtu slov 150 výskytů.

GUI	0.8313563466072083
toolkit	0.769024670124054
graphical	0.7663723230361938
graphical_interface	0.7509778738021851
user_interface	0.736338198184967
command_line_interface	0.711102306842804
toolkits	0.6962933540344238
command-line_interface	0.6930946111679077
windowing	0.6844749450683594
User_Interface	0.6833075881004333

Tabulka A.21: graphical user interface

AI	0.7854486107826233
cognitive_science	0.754462718963623
pattern_recognition	0.7279977202415466
natural_language_processing	0.7188439965248108
human-computer_interaction	0.7148535847663879
machine_learning	0.7141247987747192
artificial_life	0.6948615908622742
cybernetics	0.6910111308097839
theory	0.6765691637992859
robotics	0.6716815233230591

Tabulka A.22: artificial intelligence

machine_learning	0.7737637758255005
pattern_recognition	0.7557234168052673
information_retrieval	0.7465181946754456
data_mining	0.7227152585983276
artificial_intelligence	0.7188440561294556
cognitive_science	0.7004119753837585
machine_translation	0.6819087862968445
linguistics	0.6732376217842102
natural_language	0.6675568222999573
computational	0.6585089564323425

Tabulka A.23: natural language processing

OOP	0.8277782797813416
object_oriented	0.8179820775985718
object-oriented	0.7833099365234375
object-oriented_language	0.7627520561218262
programming_paradigm	0.727989912033081
OO	0.7247948050498962
declarative	0.6988118886947632
programming_language	0.6744760870933533
functional_programming	0.6690437197685242
functional_programming_language	0.6654204726219177

Tabulka A.24: object-oriented programming

wireless	0.8221161365509033
wifi	0.774468183517456
wireless_router	0.7696439623832703
wired	0.7525731325149536
LAN	0.7493525147438049
network	0.7415907979011536
WLAN	0.7415059208869934
connection	0.7341766953468323
wireless_LAN	0.7270090579986572
router	0.7269285917282104

Tabulka A.25: wireless network

microcomputer	0.7359427809715271
IBM_PC	0.6828758716583252
IBM-compatible	0.6815131902694702
low-end	0.6412234306335449
mass-market	0.6396372318267822
computer	0.6330452561378479
workstation	0.6305092573165894
mainframe	0.6269218921661377
compatibles	0.6266996264457703
Apple_II	0.6230087280273438

Tabulka A.26: personal computer

social_networking	0.8373622298240662
social_medium	0.8239201903343201
social_networking_site	0.7691444158554077
blogging	0.7054885029792786
Facebook	0.7049433588981628
Reddit	0.6950662732124329
Twitter	0.6904321312904358
bookmarking	0.6788766384124756
Tumblr	0.665260910987854
Instagram	0.6514384746551514

Tabulka A.27: social network

Linux_kernel	0.7285524010658264
user-space	0.6654354333877563
user-mode	0.6642922163009644
device_driver	0.6620129942893982
hypervisor	0.6483256816864014
monolithic	0.6393470764160156
FUSE	0.6125743389129639
NetBSD	0.6096698045730591
patching	0.6094051003456116
microkernel	0.6056873202323914

Tabulka A.28: kernel

machine	0.6883833408355713
personal_computer	0.6330452561378479
hardware	0.5833504796028137
remotely	0.5507045984268188
PC	0.5499809980392456
laptop	0.5473527908325195
scanner	0.5473410487174988
hooked	0.545274555683136
desktop	0.5384228825569153
modern	0.5310214757919312

Tabulka A.29: computer

pattern_recognition	0.8090521097183228
data_mining	0.7852498888969421
natural_language_processing	0.7737635374069214
artificial_neural_network	0.7422856688499451
information_retrieval	0.7323516011238098
neural_network	0.7240930199623108
artificial_intelligence	0.7141245603561401
computational	0.7092500329017639
probabilistic	0.6657412052154541
unsupervised	0.6645969748497009

Tabulka A.30: machine learning

Windows_XP	0.8464378118515015
OSX	0.8199148774147034
Windows_Vista	0.8147355914115906
Mac	0.8111541271209717
OS	0.8032218217849731
WinXP	0.7979260087013245
Vista	0.79343581199646
XP	0.7885239720344543
Mac_OS	0.7876792550086975
Windows_2000	0.7824292182922363

Tabulka A.31: Windows

iPhone	0.7904753088951111
App_Store	0.7724881172180176
iPad	0.7416660785675049
iPhone_iPod	0.7115271091461182
iPod_Touch	0.7076528072357178
Blackberry	0.7002319693565369
iPhone_iPad	0.700178861618042
iPod_touch	0.6795750260353088
iOS	0.6775909066200256
Apple_Inc	0.6701352000236511

Tabulka A.32: Apple

email	0.882024347782135
mail	0.8125163912773132
message	0.7601068615913391
inbox	0.7312226295471191
sending	0.7027531862258911
chat_room	0.6965113878250122
gmail	0.6911800503730774
Gmail	0.6828120946884155
SMS	0.6723442673683167
send	0.6717882752418518

Tabulka A.33: e-mail

Perl	0.8110734820365906
Java	0.7892314791679382
Python	0.787574291229248
VBScript	0.7784502506256104
JavaScript	0.7769742608070374
Javascript	0.7693170309066772
MySQL	0.7438035607337952
scripting	0.725967526435852
server-side	0.7171634435653687
scripting_language	0.7159431576728821

Tabulka A.34: PHP

cloud_computing	0.810369610786438
on-premises	0.7779325842857361
cloud-based	0.736765921158752
SaaS	0.7238690853118896
enterprise	0.6909788846969604
infrastructure	0.6553126573562622
platform	0.6501408815383911
virtualization	0.6393280029296875
provisioning	0.6387097239494324
Cloud	0.6376312971115112

Tabulka A.35: cloud

USB_port	0.8706784844398499
adapter	0.8636434674263
FireWire	0.7911050319671631
plug	0.7907059788703918
Firewire	0.789085328578949
usb	0.7798641920089722
OTG	0.7755909562110901
parallel_port	0.7714306116104126
plugged	0.7661305069923401
SATA	0.7645293474197388

Tabulka A.36: USB

Twitter	0.8738222122192383
MySpace	0.8279355764389038
Tumblr	0.7829769253730774
Instagram	0.7798489928245544
Myspace	0.772232174873352
social_networking_site	0.7474170327186584
YouTube	0.7422717213630676
Reddit	0.7368338108062744
Twitter_a|c|c|ount	0.721291720867157
Digg	0.7185707092285156

Tabulka A.37: Facebook

heuristic	0.8123724460601807
greedy_algorithm	0.7841375470161438
approximation_algorithm	0.7664929032325745
deterministic	0.7624087929725647
polynomial_time	0.7403062582015991
optimization_problem	0.7342524528503418
iterative	0.7295801043510437
approximation	0.7289083003997803
optimal_solution	0.7280164957046509
gradient_descent	0.727059543132782

Tabulka A.38: algorithm

extract	0.7360821962356567
extracted	0.7252177000045776
storing	0.6964079737663269
extraction	0.6938391327857971
manipulating	0.6559672355651855
retrieving	0.6546376347541809
unstructured	0.6363616585731506
raw	0.623889684677124
stored	0.6220522522926331
analyzing	0.6158974766731262

Tabulka A.39: extracting

processor	0.9098926186561584
GPU	0.8460447192192078
clock_speed	0.8193826675415039
cpu	0.7843027710914612
dual-core	0.779620349407196
Pentium	0.7648609280586243
overclocked	0.7525323629379272
Intel_Core_Duo	0.7514443397521973
Opteron	0.7463881373405457
clock_rate	0.7459242343902588

Tabulka A.40: CPU

A.3 Síť CBOW, minimální výskyt slova 50x

Následují tabulky pro vyhledávaná slova pro neuronovou síť CBOW a nastavení minimálního počtu slov 50 výskytů.

GUI	0.8043637275695801
user_interface	0.7548195123672485
graphical_interface	0.731995701789856
UI	0.6855037808418274
windowing_system	0.6810895204544067
widget_toolkit	0.6682736873626709
front-end	0.6516410708427429
Graphical_User_Interface	0.6449233889579773
functionality	0.6446083188056946
GNOME_desktop_environment	0.6427207589149475

Tabulka A.41: graphical user interface

AI	0.7196123003959656
neuroscience	0.6899906992912292
natural_language_processing	0.6690598726272583
theory	0.6644940376281738
computer_science	0.6460037231445312
robotics	0.6429806351661682
theoretical_computer_science	0.6368385553359985
cognitive_science	0.6360152959823608
linguistics	0.6244781613349915
biology	0.621038556098938

Tabulka A.42: artificial intelligence

pattern_recognition	0.7389746308326721
data_mining	0.7228318452835083
machine_learning	0.7161930799484253
machine_translation	0.690359354019165
artificial_intelligence	0.669059693813324
computer_vision	0.6475123167037964
information_retrieval	0.637970507144928
bioinformatics	0.6320317387580872
knowledge_representation	0.6229563355445862
computational_linguistics	0.6217935681343079

Tabulka A.43: natural language processing

object-oriented	0.7095054388046265
OOP	0.6956813931465149
prototype-based	0.66167151927948
object_oriented	0.6584893465042114
functional_programming	0.6566161513328552
domain-specific	0.6419342756271362
inheritance	0.6262103915214539
object-oriented_language	0.622478187084198
high-level	0.6152750253677368
procedural	0.6081942915916443

Tabulka A.44: object-oriented programming

wifi	0.7431923151016235
wireless	0.7412391901016235
network	0.7391093969345093
wireless_internet	0.7169849872589111
wi-fi	0.6950522661209106
wireless_LAN	0.6801022887229919
wired	0.6611830592155457
LAN	0.6573325991630554
Wi-Fi	0.6408827900886536
broadband_Internet	0.6405928730964661

Tabulka A.45: wireless network

microcomputer	0.7960578799247742
mainframe_computer	0.6926971673965454
mainframe	0.6765276193618774
minicomputer	0.6695687174797058
PC	0.6564918160438538
computer	0.641882598400116
netbooks	0.6190574169158936
programmable_calculator	0.6068350672721863
workstation	0.5983648300170898
high-end	0.5929993987083435

Tabulka A.46: personal computer

social_networking	0.7139344811439514
social_medium	0.6739434599876404
social_networking_site	0.6278300881385803
photo_sharing	0.5918545722961426
e-commerce	0.5884159207344055
mobile_apps	0.5795571804046631
discussion_forum	0.5607890486717224
Facebook_Twitter	0.5566734671592712
portal	0.5523614883422852
blogging	0.5325117111206055

Tabulka A.47: social network

Linux_kernel	0.6722376942634583
kernel-mode	0.6465441584587097
hypervisor	0.6328706741333008
monolithic_kernel	0.62635338306427
device_driver	0.6167489290237427
BSD	0.6142944693565369
positive_definite	0.6113897562026978
kernel_module	0.6087340116500854
user-mode	0.5968297123908997
VM	0.5962842106819153

Tabulka A.48: kernel

PC	0.6702527403831482
personal_computer	0.6418827772140503
machine	0.6410247087478638
laptop	0.562408983707428
microcomputer	0.5280351042747498
my_laptop	0.5261626839637756
device	0.5225568413734436
my_pc	0.5198904275894165
window_vista	0.49010568857192993
thin_client	0.4813291132450104

Tabulka A.49: computer

computer_vision	0.7688296437263489
pattern_recognition	0.7610399723052979
natural_language_processing	0.7161931991577148
data_mining	0.6827767491340637
statistical	0.6804611086845398
bioinformatics	0.6759695410728455
computational_geometry	0.658515453338623
machine_translation	0.6568875312805176
deep_learning	0.6491469144821167
analytical	0.6467697620391846

Tabulka A.50: machine learning

OS	0.8324782848358154
Windows_XP	0.8113648295402527
Vista	0.795852780342102
Microsoft_Windows	0.7795665860176086
Windows_Vista	0.7743601202964783
XP	0.7737743854522705
Linux	0.7564771175384521
Mac_OS	0.7447975873947144
Mac	0.7337431311607361
window	0.7308934330940247

Tabulka A.51: Windows

iPhone	0.627647340297699
Apple_Inc	0.6177367568016052
1st_generation	0.6088740229606628
iPad	0.605204701423645
Microsoft	0.6008187532424927
iPhone_iPad	0.5755926966667175
Amazon	0.5740593075752258
Samsung	0.5719084143638611
app	0.5650652050971985
Applecom	0.5632712841033936

Tabulka A.52: Apple

email	0.8313530683517456
mail	0.7702250480651855
instant_message	0.7358167171478271
message	0.7054136395454407
spam	0.6708726286888123
gmail	0.661406934261322
inbox	0.660676896572113
E-mail	0.6601711511611938
electronic_mail	0.642715334892273
Email	0.6280889511108398

Tabulka A.53: e-mail

Java	0.8201928734779358
Perl	0.8124675750732422
Python	0.8042388558387756
Javascript	0.7984417080879211
JavaScript	0.7889787554740906
Lua	0.7350451350212097
Tcl	0.7103492021560669
Visual_Basic	0.7044013738632202
MySQL	0.6966562271118164
Objective-C	0.6902841329574585

Tabulka A.54: PHP

cloud_computing	0.7443052530288696
cloud-based	0.6692754030227661
cloud_storage	0.61394202709198
PaaS	0.6037663221359253
enterprise	0.603637158870697
on-premises	0.5934213995933533
social_networking	0.5860669612884521
unified_communication	0.5741235017776489
SaaS	0.5711436867713928
high_availability	0.5497843027114868

Tabulka A.55: cloud

USB_20	0.7778988480567932
adapter	0.7640661001205444
SATA	0.7539756894111633
USB_port	0.733649492263794
usb	0.7188190221786499
adaptor	0.7144667506217957
SCSI	0.7136526703834534
serial_port	0.7136251330375671
IEEE_1394	0.7040562629699707
ethernet	0.704045295715332

Tabulka A.56: USB

Twitter	0.8160302639007568
Instagram	0.7480965256690979
MySpace	0.7380514740943909
Yahoo	0.734257698059082
Facebook_Twitter	0.7272824048995972
LinkedIn	0.7261114120483398
Reddit	0.7090898752212524
Tumblr	0.6997812390327454
Google	0.6920366287231445
YouTube	0.6837939023971558

Tabulka A.57: Facebook

heuristic	0.8074427843093872
sorting_algorithm	0.7742641568183899
method	0.7686731815338135
randomized_algorithm	0.7132996916770935
FFT_algorithm	0.7116833925247192
generalization	0.7116292119026184
greedy_algorithm	0.7063485980033875
depth-first_search	0.7028312683105469
decision_tree	0.6971718072891235
simulated_annealing	0.6930079460144043

Tabulka A.58: algorithm

storing	0.7154590487480164
compressing	0.6825777888298035
extracted	0.6819174289703369
extraction	0.6531233191490173
extract	0.648033857345581
retrieving	0.637912929058075
may_contain	0.6274407505989075
manipulating	0.6218680143356323
stored	0.6138348579406738
analyzing	0.6079496741294861

Tabulka A.59: extracting

processor	0.8415044546127319
GPU	0.7448861002922058
dual-core	0.6971346139907837
FPU	0.693338930606842
cpu	0.6823139786720276
chip	0.6778613328933716
chipset	0.6738382577896118
microprocessor	0.6646378636360168
graphic_card	0.6636857986450195
MMU	0.6635696291923523

Tabulka A.60: CPU

A.4 Síť skip-gram, minimální výskyt slova 50x

Následují tabulky pro vyhledávaná slova pro neuronovou síť CBOW a nastavení minimálního počtu slov 50 výskytů.

GUI	0.8221586346626282
graphical_interface	0.7954292893409729
graphical_user_interface_GUI	0.7875458598136902
user_interface	0.7653071284294128
graphical	0.7645277976989746
Graphical_User_Interface	0.723319947719574
windowing_system	0.7232717871665955
command-line_interface	0.713936448097229
widget_toolkit	0.7079626321792603
user-interface	0.7038522958755493

Tabulka A.61: graphical user interface

cognitive_science	0.751270592212677
AI	0.7413702011108398
robotics	0.7168495059013367
machine_learning	0.7082957029342651
natural_language_processing	0.7046661376953125
theory	0.6985126733779907
neuroscience	0.6947837471961975
computational	0.694147527217865
subfield	0.6903601288795471
human-computer_interaction	0.688478946685791

Tabulka A.62: artificial intelligence

computational_linguistics	0.7585175037384033
machine_learning	0.7458779215812683
machine_translation	0.7439097762107849
pattern_recognition	0.7121661901473999
artificial_intelligence	0.7046661972999573
text_mining	0.701724112033844
information_retrieval	0.7016003131866455
computational_biology	0.6997258067131042
data_mining	0.6964676976203918
computational_chemistry	0.6885359883308411

Tabulka A.63: natural language processing

OOP	0.7693296074867249
object_oriented	0.7315754890441895
object-oriented_programming_language	0.7298269867897034
object-oriented	0.7229610085487366
prototype-based	0.7222098708152771
functional_programming	0.71977698802948
object-oriented_language	0.7186964154243469
object-based	0.7081734538078308
paradigm	0.7061777710914612
programming_paradigm	0.673315703868866

Tabulka A.64: object-oriented programming

wireless	0.8126227259635925
wifi	0.7950035333633423
wireless_router	0.7881866097450256
wireless_connection	0.7702813744544983
WLAN	0.7513231039047241
network	0.735107421875
LAN	0.7325102090835571
connection	0.7283474802970886
router	0.7282823324203491
wi-fi	0.7227232456207275

Tabulka A.65: wireless network

microcomputer	0.7418967485427856
PC-compatible	0.6793242692947388
mainframe	0.6716433763504028
programmable_calculator	0.6699073314666748
smartphones_tablet	0.6614247560501099
IBM-compatible	0.6606281995773315
low-end	0.657763659954071
computer	0.6567084193229675
mass-market	0.6531215906143188
mainframe_computer	0.6462290287017822

Tabulka A.66: personal computer

social_networking	0.818598210811615
social_networking_site	0.8130648732185364
social_medium	0.7901661396026611
Facebook_Twitter	0.7322298288345337
social_medium_site	0.7176599502563477
Twitter	0.7118661403656006
Facebook	0.7074253559112549
blogging	0.6988393068313599
social-networking	0.6874714493751526
social_bookmarking	0.6772880554199219

Tabulka A.67: social network

Linux_kernel	0.7328839898109436
monolithic_kernel	0.7266373038291931
kernel_module	0.7102507948875427
userland	0.6911308169364929
loadable_kernel_module	0.6893668174743652
device_driver	0.6758898496627808
initrd	0.6704739928245544
user-space	0.6547905206680298
kernel-based	0.6510433554649353
OSes	0.6398967504501343

Tabulka A.68: kernel

machine	0.6798543334007263
personal_computer	0.6567084193229675
laptop	0.6371073722839355
PC	0.6112921237945557
hardware	0.5959345698356628
window_vista	0.5936201810836792
desktop	0.5913107395172119
my_laptop	0.5868101119995117
ethernet_cable	0.5820971131324768
my_pc	0.5789662003517151

Tabulka A.69: computer

pattern_recognition	0.7921779155731201
data_mining	0.7773587703704834
natural_language_processing	0.7458781003952026
computational	0.7367974519729614
computer_vision	0.7286208868026733
computational_chemistry	0.726529061794281
subfield	0.7250680327415466
artificial_neural_network	0.7210837602615356
statistical	0.7180211544036865
information_retrieval	0.7158389687538147

Tabulka A.70: machine learning

OS	0.8435415625572205
Windows_XP	0.832556962966919
Mac	0.8286304473876953
XP_SP2	0.8115060329437256
Linux	0.8026548027992249
Mac_OSX	0.7987619638442993
Xp	0.7975661158561707
XP	0.7969266772270203
Mac_OS	0.7962579727172852
OSX	0.7935374975204468

Tabulka A.71: Windows

iPhone	0.7786290049552917
iPad	0.7361947894096375
iPod_touch	0.7148223519325256
Apple_Inc	0.7047652006149292
Apple_Watch	0.7020667195320129
iPhone_iPad	0.6978839039802551
App_Store	0.6839324235916138
iPod_Touch	0.6772884726524353
Apple_iPhone	0.6767436861991882
iPad_iPhone	0.6742246747016907

Tabulka A.72: Apple

email	0.9063448905944824
mail	0.8489903211593628
inbox	0.7491521239280701
instant_message	0.7442536354064941
message	0.7416598200798035
gmail	0.7319704294204712
emailing	0.7304559946060181
email_address	0.7224278450012207
spam	0.6989098191261292
spam_message	0.6912412643432617

Tabulka A.73: e-mail

Java	0.8279722929000854
Perl	0.8160688877105713
Python	0.8117995858192444
JavaScript	0.8080374598503113
Javascript	0.7698615789413452
MySQL	0.7672544121742249
VBScript	0.7661694288253784
scripting_language	0.7577295303344727
VB	0.741542637348175
Ruby	0.719584047794342

Tabulka A.74: PHP

cloud_computing	0.8070127964019775
on-premises	0.7665688991546631
private_cloud	0.7582270503044128
unified_communication	0.7528784871101379
cloud-based	0.7444546222686768
enterprise	0.7099321484565735
IaaS	0.7034497857093811
PaaS	0.7026900053024292
SaaS	0.7009006142616272
infrastructure	0.6863337159156799

Tabulka A.75: cloud

USB_port	0.847751259803772
adapter	0.8389794230461121
USB_20	0.8358200192451477
Firewire	0.7989856004714966
USB_cable	0.7987616062164307
FireWire	0.7940806746482849
firewire	0.7939668297767639
USB_device	0.7933222651481628
usb	0.7776947021484375
external_hard_drive	0.7755358815193176

Tabulka A.76: USB

Twitter	0.899215579032898
Instagram	0.8163809776306152
MySpace	0.8028464317321777
Tumblr	0.7628253698348999
social_networking_site	0.7627602219581604
Facebook_Twitter	0.7494291067123413
Bebo	0.7468041181564331
Reddit	0.7411428689956665
LinkedIn	0.7338110208511353
YouTube	0.7327550053596497

Tabulka A.77: Facebook

dynamic_programming	0.8156738877296448
heuristic	0.8022686839103699
simulated_annealing	0.7875794172286987
belief_propagation	0.7857147455215454
simplex_algorithm	0.7739542126655579
approximation_algorithm	0.769854724407196
greedy_algorithm	0.7667945623397827
iterative_method	0.765742301940918
hill_climbing	0.7577208280563354
decision_tree	0.7551628947257996

Tabulka A.78: algorithm

extract	0.724399209022522
extracted	0.708008348941803
storing	0.6861444115638733
extraction	0.6840733885765076
retrieving	0.6645886301994324
compressing	0.6328518390655518
meta-data	0.6287441253662109
annotating	0.616972029209137
metadata	0.6154733300209045
analyzing	0.6084109544754028

Tabulka A.79: extracting

processor	0.9152953624725342
GPU	0.8203068971633911
dual_core	0.8194321393966675
single-core	0.811764657497406
CPU_core	0.8025991916656494
cpu	0.792268693447113
clock_speed	0.7920529246330261
overclocked	0.7819507122039795
dual-core	0.7750965356826782
microprocessor	0.7718425393104553

Tabulka A.80: CPU

Příloha B

Příklad spuštění skriptu *findNearest*

Zde je rozebrán postup a jednotlové kroky při spouštění a obsluze skriptu findNearest a zjištění odpovědi na 10 nejlbižších slov pro sousloví „graphical user interface“.

```
./findNearest.py resources/wikipedia_IT_truecased_CBOW_150.model resources/wikipedia_IT_truecased
```

Prvním argumentem je cesta pro předem natrénovaný model. V tomto případě se jedná o model natrénovaný s použitím neuronové sítě CBOW, který zahrnuje pouze slova, která se v korpusu objeví více než 150x.

```
2016-05-10 16:40:21,254 : INFO : loading Word2Vec object from resources/wikipedia_IT_truecased_CBOW_150.model
2016-05-10 16:40:21,764 : INFO : setting ignored attribute syn0norm to None
2016-05-10 16:40:21,764 : INFO : setting ignored attribute cum_table to None
2016-05-10 16:40:21,765 : INFO : precomputing L2-norms of word weight vectors
Model sucesfully loaded
Do you want to load multiword models? It will take around 2 minutes but its recommended. y/n
```

Skript nejprve načítá naučený model z udané cesty. Po jeho načtení je provedena normalizace vektorů a nenormalizované jsou zahozeny. Tím se ušetří velká část potřebné paměti. Nyní je na uživateli, zda chce načíst modely, automaticky vyhledávající ustálená slovní spojení ve vstupních dotazech. Při zvolení možnosti „n“ budou vyhledávána zadaná slova zvlášť a nikoliv jako sousloví, což značně negativně ovlivní výsledky. V tomto příkladě tedy volíme možnost „y“.

```
Now loading multiword models, this will take some time
2016-05-10 16:46:47,384 : INFO : loading Phrases object from resources/wikipedia_IT_truecased_CBOW_150_bigram.model
2016-05-10 16:47:17,015 : INFO : loading Phrases object from resources/wikipedia_IT_truecased_CBOW_150_trigram.model
2016-05-10 16:47:47,547 : INFO : loading Phrases object from resources/wikipedia_IT_truecased_CBOW_150_fourgram.model
Multiword models sucessfully loaded
Do you want to print cosine distance between input word and founded words? You can turn it off or on lately by typing *cosine* as word. y/n
```

Po načtení modelů vyhledávajích ustálená slovní spojení musí uživatel zvolit, zda chce k vyhledaným výsledkům zobrazovat kosinovou míru podobnosti vyhledaných slov ke slovu vyhledávanému. Pro tento příklad volíme možnost „y“.

```
Do you want to try find most similar articles for results? It will also allow to process shorts. You can turn off or on lately by typing *articles* as word. y/n
```

Dále je na uživateli, zda chce k vyhledaným výsledkům provést analýzu zkratek a vyhledat odpovídající články na Wikipedii. Pro tento příklad volíme možnost „y“.

```
Do you want to return nouns only? You can turn off or on lately by typing *nouns* as word. y/n
```

Další možností nastavení skriptu je možnost potlačení jiných slovních druhů ve výsledku než podstatných jmen. Opět volíme možnost „y“.

```
For quit write *quit* as word and press enter

Enter a word or words:graphical user interface
Enter a number of desired words:10
```

Uživatel vyplní požadované slovo či sousloví a počet nejbližších slov, které si přeje znát a vyčká na výsledek skriptu.

```
Your request was changed by multiword model to this shape
graphical_user_interface
```

Nejdříve skript vypíše výsledek vyhledávání ustálených slovních spojení v zadaném výrazu. Je vidět, že slova „graphical user interface“ se objevují blízko sebe často a proto je model pro vyhledávání spojil v jedno sousloví.

```
Searching for shorts:
graphical_user_interface (GUI)
```

Dále bylo provedeno vyhledávání zkratek u vyhledávaného termínu a byla nalezena shoda.

```
Terms related to graphical user interface:
('toolkits', 0.7354674935340881)
('GUI', 0.7315068244934082)
('widget', 0.6922416090965271)
('command_line_interface', 0.67087721824646)
('user_interface', 0.6708413362503052)
('graphical_interface', 0.6483340263366699)
('frontend', 0.6443015933036804)
('toolkit', 0.6379028558731079)
('window_manager', 0.6355262994766235)
('UI', 0.630246639251709)
```

Zde můžeme vidět výsledky vyhledávání významově nejbližších slov a hodnotu kosinové míry podobnosti.

```
Searching for articles, most similar to founded words...
Most similar article for word "toolkits"is: "BASIC extension"and keywords are common, type, program, home, computer, s., Generally, third-party, extensions
GUI is Graphical user interface and keywords are: Longman, Pronunciation, Dictionary, edition, Pearson, Education, Ltd., Harlow, page
Most similar article for word "widget"is: "Widget (GUI)"and keywords are widgets, frame, frame, bottom
I cant find similar article for word "command_line_interface".
Article for word "user_interface"(UI) is User interface and keywords are: space
I cant find similar article for word "graphical_interface".
I cant find similar article for word "frontend".
Most similar article for word "toolkit"is: "Fox toolkit"and keywords are open, source, cross-platform, widget, toolkit
Article for word "window_manager"is: "Window manager"and keywords are distinct, programs, Wayland, function
UI is User interface and keywords are: space
```

Na závěr se skript pokusí pro výsledky vyhledat konkrétní články z Wikipedie a z nich klíčová slova. Na řádku tři si můžeme povšimnou, že správně identifikoval zkratku a vypsal k ní plný název, shodný s názvem článku. Na řádku 6 naopak zprávně identifikoval, že sousloví „user_interface“ má zkratku UI. Nyní je skript připraven pro další slovo k vyhledání.