Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

# Bakalářská práce

# Explicitní sémantická analýza

Plzeň 2017 Michal Tušl

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 25. dubna 2017

Michal Tušl

# Poděkování

Děkuji Ing. Tomáši Brychcínovi, Ph.D., za vedení mé bakalářské práce, za cenné rady a čas, který mi věnoval.

# Abstract

This bachelor thesis investigates semantic analysis of texts in natural language. It focuses on Explicit Semantic Analysis and Latent Semantic Analysis methods. These methods are based on unsupervised machine learning and use Wikipedia as a training data. Singular Value Decomposition is used to reduce the memory requirements and also to improve the results. Standard English and Czech datasets are used for testing purposes. These datasets contain word pairs and manually annotated semantic similarity. The quality of semantic representation is evaluated by Pearson and Spearman correlation. All tested methods provide very promising results on both languages.



# Abstrakt

Tato práce je zaměřena na sémantickou analýzu textů. Konkrétně na metody Explicitní sémantická analýza a Latentní sémantická analýza. Tyto metody jsou založené na trénování bez učitele a jako trénovací data využívají Wikipedii. Na výsledek metod je aplikován singulární rozklad matic, který redukuje paměťové nároky a zároveň vylepšuje výsledky metod. Testování výsledků je prováděno na standardních datasetech pro anglický a český jazyk. Tyto datasety obsahují páry slov a manuálně definovanou sémantickou podobnost. Kvalita sémantické reprezentace je měřena pomocí Pearsonovy a Spearmanovy korelace. Všechny testované metody dosahují na obou jazycích velmi dobrých výsledků.

# Obsah

# 1 Úvod

Zpracování přirozeného jazyka – NLP (*Natural Language Processing*) je rozvíjejícím se oborem, který je čím dál víc využíván. Používají ho velké společnosti jako jsou Google nebo Facebook ve svých vyhledávačích, zobrazování příspěvků a dalších funkcích. Lze předpokládat, že se obor bude dále rozšiřovat i do jiných oblastí informačních technologií. Jádrem tohoto zpracování je sémantika, která se zabývá významem výrazů z různých úrovní jazyka. V tomto případě se tedy jedná hlavně o zpracování a částečné porozumění textu počítačem. V současné době existuje již mnoho metod, které dokážou určit sémantický význam textu. Metody pro určování významu textu se dělí na dvě skupiny, podle postupu při vytváření sémantické interpretace, na metody s trénováním s učitelem (*supervised*) a na metody trénování bez učitele (*unsupervised*). Metody, které se trénují s učitelem musí mít k dispozici manuálně označkovaná data, slovníky, gramatická pravidla, atd., což je zdlouhavý a finančně náročný proces, který se navíc násobí pokud potřebujeme pracovat s dalším jazykem. Metody trénování bez učitele se učí samy na velkém korpusu dat. Tyto metody jsou založené na distribuční hypotéze. Jednou z metod, využívajících distribuční hypotézy, je i ESA (*Explicit Semantic Analysis*). Cílem práce je implementovat a pokusit se tuto metodu vylepšit.

ESA vytváří vícerozměrné vektory, kde každý prvek vektoru je jeden kontext. Tyto vektory reprezentují sématiku jednotlivých slov. Na základě vypočtených vektorů pak lze porovnávat podobnost významu slov nebo dokonce souvislých textů. K učení významu potřebuje metoda velké množství dat. Je k tomu použita Wikipedie, která obsahuje velké množství článků z různých oborů (kontexty). Výsledek této reprezentace lze využít pro další aplikace jako je například vyhledávání informací, strojový překlad, opravy pravopisu a gramatiky, jazykové modelování pro rozpoznávání řeči, dialogové systémy a další.

Tato práce se dále zabývá metodou LSA (*Latent Semantic Analysis*), která je velice podobná metodě ESA. Při vylepšení metody ESA jsem použil kategorie z Wikipedie jako kontexty. Zároveň jsem místo původního výpočtu TF-IDF, který využívá ESA i LSA, použil metodu PMI (*Pointwise Mutual Information*). Z tohoto původního vylepšení vznikla nová metoda pro sémantickou interpretaci. Celkem tak bylo v této práci implementováno několik metod na reprezentaci významu přirozeného jazyka, všechny postupy při výpočtu jsou detailně popsány v následujících kapitolách.

Všechny metody byly testovány na standardních datasetech *RG-65*, *WordSimilarity-353* a *Simlex999*. Tyto datasety obsahují páry slov a číselné hodnocení, které udává v jaké míře si jsou slova významově podobná. Číselné hodnocení v datasetech určili lidé. Výsledek metody se tedy porovnává s lidským vyhodnocením výrazu. Při testování se změří kosinová podobnost u všech párů testovaných slov. Série naměřených hodnot se porovná s dostupným hodnocením pomocí Pearsonovy a Spearmanovy korelace.

Testování jsem prováděl nejen na anglických datech, ale i pro čestinu, pro kterou jsem rovněž použil data z článků na Wikipedii. Díky nízké flexi jazyka, fungují metody pro sémantickou reprezentaci textu na angličtině většinou dobře. Oproti tomu je čeština daleko komplexnější a morfologicky složitější jazyk, což komplikuje správnou sémantickou interpretaci počítačem a činí ji obtížnější.

Čeština byla testována na datasetech *RG-65* a *WordSimilarity-353*, které byly přeloženy z anglické verze do češtiny.

Práce má následující strukturu. V kapitole 2 se věnuji základům jazykovědy a některým lingvistickým pojmům, neboť na této oblasti vědy je postaveno i zpracování přirozeného jazyka. Kapitola 3 popisuje Distribuční sémantiku, která se zabývá vyhodnocením a pochopením věcného významu jazyka. V kapitole 4 je popsána metody Explicitní sémantická analýza, která je právě na distribuční sémantice postavená. Následuje kapitola 5 o singulárním rozkladu matic, který některé metody využívají pro redukci dimenze matice, aby se snížily paměťové nároky a zároveň zlepšily výsledky metod. Singulární rozklad byl v této práci též hodně využíván, proto mu byla věnována samostatná kapitola. V kapitole 6 jsou popsány vylepšení původní metody a zároveň popis nové metody *Latent Semantic Categories*. V kapitole 7 se práce věnuje přípravě trénovacích dat pro tyto metody a evaluaci jejich výsledků. Ná závěr je pak uvedena výpočetní složitost úloh a dosažené výsledky.

# 2 Jazykověda

Jazykověda (lingvistika) je vědní obor, který se zabývá psanou a mluvenou formou jazyka [3]. Mezi její základní disciplíny patří:

- fonetika – zabývá se popisem zvuků a hlásek jazyka, jejich tvorbou a vnímáním
- fonologie – zkoumá funkci zvuků a hlásek jazyka
- morfologie (tvarosloví) – zabývá se tvorbou slov
- syntax – zabývá se tvorbou a stavbou vět
- lexikologie – popisuje slovní zásobu jazyka
- sémantika – zabývá se věcným významem, zejména slov

Existuje i řada disciplín, které stojí na pomezí lingvistiky a jiných vědních oborů. Jednou z nich je i počítačová (strojová) lingvistika, která se zabývá počítačovým zpracováním přirozeného jazyka. Mezi problémy, které počítačová lingvistika řeší je například porozumění mluvenému slovu, jeho zápis do psané podoby, strojový překlad nebo korekce gramatiky [3, 4].

## 2.1 Lingvistické pojmy

Pro potřeby této práce zmiňuji několik základních lingvistických pojmů, které se v této práci později vyskytnou:

- gramatika (mluvnice) – soubor pravidel, podle kterých se řídí tvorba vět, slov a jejich tvarů
- flexe – ohýbání slovních tvarů (např. skloňování, časování, stupňování)
- kořen – základ sloužící k tvorbě dalších slov nebo jejich tvarů
- kmen – základ slova skládající se ze samotného kořene, nebo kořene a jednoho či více afixů
- afix – část připojovaná ke slovu
  - prefix (předpona) – afix připojovaný před kmen slova, např. u<u>dělat</u>

  - sufix (přípona) – afix připojovaný za kmen slova, např. nebeský
  - infix (vpona) – afix připojovaný do kmene slova, např. černobílý
- korpus – soubor textů z daného jazyka sloužící jako zdroj dat k další analýze
- lemma – základní (slovníkový) tvar slova
- lemmatizace – přiřazení lemmat slovům
- fráze – ustálené slovní spojení
- synonymie – dvě slova mají stejný význam (např. hezký/pěkný)
- antonymie – dvě slova opačného významu (např. vysoký/nízký)
- homonymie – dvě slova mají stejnou zvukovou nebo psanou podobu, avšak mají jiný význam a jsou jiného původu (např. travička – zeleň / travička – žena, která otravuje)
- polysémie – jedno slovo má různé významy, jeden význam je však odvozen z druhého (např. lidské ucho, ucho hrnce)
- hyponymie – podřazený pojem (např. slovo pes je podřazený pojem slovu zvíře)
- hyperonymie – nadřazený pojem (např. slovo zvíře je nadřazený pojem slovu pes)

## 2.2 Typologie jazyků

Z lingvistického hlediska lze jazyky rozdělit do několika skupin. Kromě genetické klasifikace, která je klasifikuje podle původu a příbuznosti, lze jazyky rozdělit i typologicky na jazykové typy podle charakteristických společných rysů. Nejpropracovanější typologií je strukturní typologie, jež rozděluje jazyky podle vyjadřování gramatických funkcí a srovnává je z pohledu různých částí lingvistiky, včetně fonologie, morfologie nebo syntaxe. Tato typologie dělí jazyky na aglutinační (např. finština, korejština), flektivní (např. čeština, latina), analytické (např. angličtina, vietnamština) a polysyntetické (např. indiánské jazyky). Žádný jazyk nemá výhradně aglutinační nebo výhradně flektivní prvky, pouze u něj dané rysy převažují [3].

Aglutinační jazyky vyjadřují gramatické funkce pomocí afixů, tedy kumulováním předpon a přípon na kořen slova, přičemž každý afix má pouze

jednu funkci a nese jeden význam. Flektivní jazyky využívají flexe, která se uskutečňuje pomocí koncovek a afixů, často za přítomnosti alternací, tedy hláskových změn ve slově, které ale nemění jeho význam (např. ruka – ruce). Analytické jazyky vyjadřují gramatické funkce pomocnými slovy. Polysyntetické jazyky spojují plnovýznamová slova nebo kořeny v jediný celek [3].

Vzhledem k odlišnému zařazení češtiny a angličtiny v této typologii existují mezi těmito jazyky značné rozdíly. Skloňování je u angličtiny omezeno na dva pády, nominativ (český první pád) a genitiv (český druhý pád), přičemž tvorba genitivu a ostatně i množného čísla těchto pádů je značně zjednodušená (např. *house – houses*). Z hlediska sloves má angličtina šest časů, ve většině případů se však při jejich tvorbě využívá pomocných sloves. Dále se angličtina vyznačuje nerozlišováním rodu, přítomností členů a pevným slovosledem. Ke zpracování jazyka počítačem je tak angličtina přívětivější, neboť při předzpracování korpusu se všechny členy a pomocná slovesa snadno odstraní. Díky nižší flexi je i snažší proces lemmatizace, protože se snadněji naleznou správná lemmata ke slovům.

V tomto případě je však velkou nevýhodou angličtiny její velká slovní zásoba, což je dáno hlavně historickým vývojem tohoto jazyka. I když se jedná o germánský jazyk, je zde velmi patrný vliv francouzštiny a latiny. Pro jeden význam existuje v angličtině řada synonym, přičemž každé z nich má jiný původ. Například slovo *svoboda* se dá do angličtiny přeložit jako *freedom*, které je germánského původu, nebo jako *liberty*, které je románského původu. Užití románských slov se oproti germánským častěji vyskytuje ve formálnějších výrazech, byť jsou významově stejné (např. *Statue of Liberty*). Při zpracování přirozeného jazyka počítačem je pak obtížnější zachytit všechna synonyma ve stejných kontextech.

# 3 Distribuční sémantika

Distribuční sémantika nabízí poměrně jednoduchý a praktický způsob, jak reprezentovat sémantiku jednotlivých slov z textu. Modely distribuční sémantiky jsou založeny na předpokladu, že význam slova lze odvodit z jeho použití (distribuce v textu). Tyto modely vytvářejí sémantickou reprezentaci ve formě vektorových prostorů z vysokou dimenzí, využitím statistické analýzy kontextu, ve kterém se slovo vyskytlo [8, 15].

Většina metod je založena na trénování bez učitele. K tomu je zapotřebí velkého množství dat, z kterých lze určit, v jakých kontextech se slova nacházejí. Distribuční sémantika vychází ze dvou základních hypotéz: *distribuční hypotéza* a *Bag-of-Words* hypotéza.

*Distribuční hypotéza*, na rozdíl od *Bag-of-Words*, bere v úvahu rozmístění slov ve větě. Zároveň zkoumá přímo význam jednotlivých slov, tedy jestli jsou dvě rozdílná slova použita ve stejném kontextu, pak by měli mít i stejný význam. Metody distribuční sématiky, ale často využívají i *Bag-of-Words* hypotézy.

*Bag-of-Words* hypotéza je založena na porovnávání množin, kde pořadí prvků nehraje roli. Například množina *a,a,b,b,c* a množina *c,a,b,a,b* jsou ekvivalentní. Praktické využití má především u porovnávání celých dokumentů, tedy pokud dva dokumenty obsahují stejná slova, jsou dokumenty podobné. Nevýhoda je u krátkých částí textu, jako jsou třeba věty. V *Bag-of-Words* hypotéze jsou věty *Kočka je větší než pes.* a *Pes je větší než kočka.* totožné, neboť pořadí slov zde nehraje roli.

Kontexty, pro které se určuje míra asociace se slovem, jsou dvojího typu: lokální kontexty a globální kontexty. Globální kontexty obvykle používají modely založené na *Bag-of-Words* hypotéze. Předpokládá, že slova mají podobný sémantický význam, pokud se vyskytují v podobných dokumentech. Dokumentem zde může být věta, odstavec, nebo i větší kus textu. Tyto modely jsou schopny zachytit větší rozsah závislostí mezi slovy. Například dokument o *autech* bude pravděpodobně obsahovat slova jako *motor* nebo *řidič*. Tato slova pak budou významově podobná. Modely využívající lokální kontexty nepotřebují k sémantické analýze text rozdělený do dokumentů nebo souvislých částí textu. Tyto modely berou v úvahu i uspořádání slov. Na rozdíl od modelů využívající globální kontexty jsou tyto modely schopné nalézt vhodné substituty pro slova v zadaných kontextech. Například ve větě *Pes je zvíře.* může být slovo *pes* nahrazeno za slovo *kočka* [1].

## 3.1 Sémantické prostory

Význam jednotlivých slov je popsán jako vektor v prostoru s vysokou dimenzí, kde každý prvek vektoru odpovídá jednomu kontextu, ve kterém se slovo vyskytuje. Slova, která mají podobný význam, jsou k sobě blízko ve vektorovém prostoru. Pro výpočet vzdálenosti mezi těmito vektory se nejčastěji používá *kosínová podobnost*. Například pro dva vektory $\boldsymbol{a}$ a $\boldsymbol{b}$ se spočte vzdálenost pomocí nasledujícího vzorce 3.1:

$$s_{cos}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|} = \frac{\sum_{i=0}^{k} a_i b_i}{\sqrt{\sum_{i=0}^{k} a_i^2 \sum_{i=0}^{k} b_i^2}}, \tag{3.1}$$

kde $k$ je dimenze vektorů. Výsledná podobnost mezi dvěma slovy je kosínus úhlu jejich vektorů.

Vektory jednotlivých slov jsou vytvořeny pomocí matice $\boldsymbol{M} = |\boldsymbol{D}| \times |\boldsymbol{W}|$, do které se ukládají četnosti jednotlivých slov $t \in \boldsymbol{W}$, kde každý řádek matice $\boldsymbol{M}$ odpovídá jednomu slovnímu vektoru. Sloupce matice pak odpovídají dokumentům $d \in \boldsymbol{D}$, kde $\boldsymbol{D}$ je množina dokumentů. Z této matice lze pak snadno získat vektor pro jednotlivá slova (řádek) nebo dokumenty (sloupce), kde každý index je jedna složka vektoru.

Do matice se ukládá hodnota TF-IDF, která se získá kombinací dvou hodnot TF (*term frequency*) a IDF (*inverse document frequency*).

TF udává důležitost slova pro daný dokument. Existují různé možnosti výpočtu, každá metoda využívající hodnotu TF může mít svůj vlastní vzorec. Například velice jednoduchý způsob by bylo pouhé určení poměru výskytu výrazu v celém dokumentu, tedy:

$$\mathrm{tf}(t_i, d_j) = \frac{\mathrm{N}(t_i)}{\sum_{t_a \in d_j} N(t_a)}, \tag{3.2}$$

kde $\mathrm{N}(t_i)$ je počet výskytů výrazu $t_i$ v dokumentu $d_j$.

IDF udává důležitost slova ve všech dokumentech dohromady. Pokud se tedy slovo $t_i$ vyskytuje ve všech dokumentech, nebude příliš důležité. Vzorec je zde jednotný a na rozdíl od TF se počítá pouze jednou pro každé slovo:

$$\mathrm{idf}(t_i) = \log \frac{n}{df_i}, \tag{3.3}$$

kde $t_i$ je výraz, pro který inverzní index počítáme, $n = |\boldsymbol{D}|$ je počet dokumentů a hodnota $df_i$ je počet, v kolika dokumentech se výraz vyskytoval.

## 3.2 Modely

Většina modelů používá pro výpočet sémantických vektorů jeden z následujících čtyř způsobů: Kookurenční matici, Model témat, náhodné indexování a neuronovou síť [1]. U metod založených na kookurenční matici se do matice ukládají frekvence výskytu slov v kontextu. Váha výskytu slova pro daný kontext bývá obvykle vypočtena nějakou funkcí (např. TF-IDF). U těchto modelů bývá často problémem vysoká dimenze matice. *Modely témat* jsou založeny na *Bag-of-Word* hypotéze, podle níž odhalují skrytá témata z textu. Význam textu je obvykle reprezentován jako vektor témat, ale lze jej použít i pro reprezentaci slov. Modely založené na náhodném indexování jsou založeny na distribuční hypotéze a používají lokální kontext pro reprezentaci významu slov. Výhodou je, že se zde předem určí dimenze vektorů (v řádu tisíců). Na začátku jsou do vektorů náhodně uloženy hodnoty -1 a +1. Vektory se pak při výskytu slov v kontextu sčítají. Nejnovějším přístupem jsou pak metody využívajících neuronové sítě. Metody založené na neuronových sítí se velmi liší ve způsobu, jakými jsou využívány.

### 3.2.1 Word2Vec

*Word2Vec* [20] je v současnosti neúspěšnější metoda založená na neuronových sítí. Metoda ze zadaného textového korpusu vytvoří seznam slov. Pro každé z těchto slov vypočte odpovídající vektor, který reprezentuje dané slovo. Pro vypočtení podobnosti jednotlivých slov se používá výpočet vzdálenosti mezi dvěma vektory. Metoda *Word2Vec* používá dva algoritmy pro výpočet vektorů, *continuous bag-of-words* a *skip-gram*. V *continuous bag-of-words* architektuře se předpovídá kontext slova z okolních slov. V *skip-gram* se ze slova předpovídá kontext okolních slov.

### 3.2.2 GloVe

GloVe (*Global Vectors*) [21] je model pro reprezentaci slov na základě globálních vektorů. Metoda zachycuje statistiky výskytu slova v celém korpusu a pozoruje, jaká slova se vyskytovala ve stejném kontextu. Kontextem se zde myslí okolní slova, která souvisí se zkoumaným slovem. Například slova *led* a *pára* by se mohla vyskytovat ve stejných kontextech (např. v okolí slova *voda*). Avšak zároveň *led* může být spojený se slovem *pevný*, zatímco slovo *pára* nikoliv. Jednotlivé složky vektoru jsou vypočteny z pravděpodobnosti výskytu ve stejných kontextech P*(led/voda)*. Následně je na ně aplikován log-bilineární regresní model pro odhad vektorů slov.

### 3.2.3 LSA

LSA (*Latent Semantic Analysis*) [14] je metoda pro získání a reprezentaci významu slova, statisticky vypočteného na velkém korpusu textu. Základní myšlenka je shromáždit všechny kontexty, ve kterých se slovo vyskytlo. Ty pak určují vzájemnou podobnost slov. Metoda LSA nejprve napočte hodnoty TF-IDF pro všechna slova ve všech kontextech, které uloží do matice, kde každý sloupec odpovídá jednomu kontextu a každý řádek představuje jednotlivá slova. V matici na pozici $i$ a $j$ se tedy nachází váha, s jakou slovo na indexu $i$ souvisí s kontextem uloženým na indexu $j$. Všechny řádky pak představují vektory v sémantickém prostoru pro daná slova. Pokud se pracuje na velkém korpusu dat, je výsledná matice obsahující *slovo – dokument* příliš velká na to, aby se vešla do paměti. Proto se zde používá aproximace matice metodou SVD (*Singular Value Decomposition*) pro redukci dimenze. Metoda SVD je blíže popsána v kapitole 5. Z této metody též vychází metoda ESA, která je blíže popsána v následující kapitole 4.

# 4 Explicitní sémantická analýza

ESA (Explicitní sémantická analýza – *Explicit Semantic Analysis*) [9] je metoda, která reprezentuje význam textu ve vícerozměrných prostorech na základě kontextů (článků) odvozených z Wikipedie. K posouzení podobnosti jednotlivých textů v těchto prostorech se používá hodnota porovnání dvou odpovídajících vektorů použitím kosínové podobnosti.

Metoda se tedy používá pro přiřazení sémantické interpretace slovům a textu. K dispozici je vektor kontextů, $C_1$,...,$C_n$, a každý kus textu $t$ je reprezentován vektorem vah, $\omega_1$,...,$\omega_n$, kde $\omega_i$ reprezentuje sílu asociace mezi textem $t$ a kontextem $C_i$. Tato množina kontextů může být brána jako n-rozměrný sémantický prostor. Sémantika každého textu odpovídá jednomu bodu v tomto prostoru. Vážený vektor pro tento text se nazývá *vektor sémantické interpretace*.

## 4.1 Výběr dat pro vytvoření kontextů

K vytváření vektorů potřebuje metoda velké množství dat z různých oborů lidského vědění [10]. Požadavky na zdroj dat jsou tedy následující:

1. Měl by být dostatečně rozsáhlý, aby pokryl velké množství rozdílných témat.

2. Měl by být průběžně udržovaný a rozšiřovaný o nová témata.

3. Cílem metody je interpretovat věcný význam *přirozeného* jazyka, bylo by tedy nejlepší, pokud by i zdroj dat byl *přirozený*. Měl by být srozumitelný a použitelný pro běžné lidi.

4. Každý kontext $C_i$ by měl obsahovat text $d_i$, aby bylo možné určit sílu spojení významu každého výrazu v jazyce s kontextem $C_i$.

Pro udržování takového množství dat je potřeba práce mnoha lidí. Tyto podmínky nejlépe splňuje anglická verze Wikipedie, která má přes 126 000 dobrovolníků, kteří ji spravují a přidávají nové články. Celkem obsahuje přes 3,15 miliard slov ve více než 5,2 milionech článků a stále se rozšiřuje[1].

[1]Statistiky převzaty z Wikipedie

Dále je také snadno přístupná v XML formátu. Proto jsou data z Wikipedie velmi vhodná k trénovaní Explicitní sémantické analýzy. Rozsah Wikipedie je zároveň mnohem větší, než anglický rival Encyclopaedia Britannica, obsahujících 55 milionů slov v přibližně 120 000 článcích[2]. Lze tedy říci, že Wikipedie je největší souhrný dostupný zdroj lidského vědění. Hodnoty o velikostech zdrojů jsou aktuální k 16. listopadu 2016.

## 4.2 Vytvoření sémantické interpretace

Množině kontextů $C_i$,...,$C_n$, je přiřazena množina dokumentů $d_i$,...,$d_n$. Sestaví se matice $\boldsymbol{T}$, kde každý z $n$ sloupců odpovídá jednomu kontextu. Každý řádek pak odpovídá jednomu slovu, které se vyskytuje v některém z dokumentů korpusu. Počet výskytů výrazu $t_i$ v dokumentu $d_j$ označme jako $\mathrm{N}(t_i, d_j)$. Hodnota $\boldsymbol{T}_{i,j}$ v matici odpovídá hodnotě TF-IDF výrazu (slovu) $t_i$ v dokumentu $d_j$.

$$\boldsymbol{T}_{i,j} = \mathrm{tf}(t_i, d_j) \cdot \log \frac{n}{df_i} \tag{4.1}$$

Hodnota *tf* je definována jako:

$$\mathrm{tf}(t_i, d_j) = \begin{cases} 1 + \log \mathrm{N}(t_i, d_j), & \text{pokud } \mathrm{N}(t_i, d_j) > 0 \\ 0, & \text{jinak} \end{cases} \tag{4.2}$$

Hodnota $df_i = |\{d_k : t_i \in d_k\}|$ představuje počet dokumentů (článků), ve kterých se vyskytuje výraz (slovo) $t_i$.

Nakonec se na každý řádek matice $\boldsymbol{T}$ aplikuje kosínová normalizace, pro redukci rozdílů v délkách jednotlivých dokumentů

$$\boldsymbol{T}_{i,j} \leftarrow \frac{\boldsymbol{T}_{i,j}}{\sqrt{\sum_{l=1}^{n} \boldsymbol{T}_{l,j}^2}}, \tag{4.3}$$

kde $n$ je počet výrazů.

Sémantická interpretace slova $t_i$ je řádek $i$ z matice $\boldsymbol{T}$. Význam slova je dán vektorem kontextů spojeným s jejich TF-IDF hodnotami, které odráží spojitost mezi každým tématem a vybraným slovem [10].

Sémantická interpretace textu (posloupnosti slov), $\langle t_1, ..., t_k \rangle$, je těžiště vektorů reprezentujících jednotlivá slova. Tato definice dovoluje částečně určovat význam slova podle kontextu, v kterém bylo použito. Příkladem může být interpretace vektoru pro výraz *myš*. Slovo *myš* je polysémické slovo s dvěma rozdílnými významy: *myš* (hlodavec) a *myš* (počítačová). Podobně

[2]Statistiky převzaty z Wikipedie

interpretační vektor homonyma slova *rys* s významy *rys* (kočkovitá šelma) a *rys* (příznak). V textu *Rys snědl myš* po sečtení interpretačních vektorů se zvýší hodnota týkající se zvířat. Tímto způsobem se určí význam obou slov.

Matice $\boldsymbol{T}$ může být brána též jako invertovaný index a mapovat každé slovo do seznamu kontextů, v kterých se vyskytuje. Invertovaný index poskytuje velmi efektivní výpočet vzdálenosti dvou interpretačních vektorů.

Wikipedie obsahuje velké množství informací, proto je důležité kontrolovat množství šumu v textu, jako jsou nedostatečně popsané články nebo malá spojitost mezi slovem a článkem. Nevyhovující články, které mají méně než 100 slov, se vypustí a už se s nimi dále nepracuje. U spojistosti slov a článků, kde je malá vazba mezi nimi, se nastaví hodnota váhy na nula v matici $\boldsymbol{T}$.

## 4.3 Použití struktury odkazů

Wikipedie, jako většina elektronických encyklopedií, obsahuje reference mezi jednotlivými články ve formě hypertextových odkazů. Typický článek z Wikipedie má tak mnohem více odkazů, než jiné verze stejného článku ve vytištěných encyklopediích.

Každý odkaz je spojen s nějakým textem, který určuje, jaký článek je spojen s tímto odkazem. Tyto texty spojené s odkazy se nazývají *anchor text* (zvýrazněné fráze v textu, na které je možné kliknout). *Anchor text* není vždy identický se jménem článku na který odkazuje. Zároveň rozdílné texty se používají k odkazům na stejný článek v jiném kontextu. Například *anchor text*, odkazující na článek FEDERAL RESERVE, jsou *Fed*, *U.S. Federal Reserve Board*, *U.S. Federal Reserve System*, *Board of Governors of the Federal Reserve*, *Federal Reserve Bank*, *foreign reserves* a *Free Banking Era* [10]. *Anchor text* tedy mají pro jeden článek různá jména, hláskování a příbuzné fráze, které obohatí text článku k danému konceptu.

Odkazy mezi články též často odráží důležité vztahy mezi koncepty, které odpovídají připojeným článkům. Tyto odkazy též obsahují velké množství znalostí, které nelze najít v textu jednotlivých článků. Následně maximální využití těchto znalostí vede k lepšímu interpretačnímu modelu. Proto se model rozlišuje na model prvního řádu a model druhého řádu. Model prvního řádu používá pouze znalosti obsažené v článcích Wikipedie, tedy TFIDF hodnoty slov pro jednotlivé články. Model druhého řádu navíc zahrnuje znalosti obsažené v odkazech mezi jednotlivými články. Podobně se též nazývají informace získané z odkazů, informace druhého řádu.

Existence odkazu nemusí vždy znamenat, že dva články jsou silně související. Zároveň pokud chybí odkaz mezi dvěma články, nelze vyloučit, že neexistuje spojitost mezi nimi. Mnoho slov a frází v článcích Wikipedie odkazuje na jiné články jenom proto, že jsou zde záznamy pro odpovídající kontext. Například podkapitola Education (vzdělání) v článku UNITED STATES má bezdůvodné odkazy na články: HIGH SCHOOL, COLLEGE a LITERACY RATE. Pro využití odkazů Wikipedie na sémantickou interpretaci je důležité vyfiltrovat přiřazené kontexty podle jejich relevance k textu, který je interpretován [10].

Nejlepší cesta, jak zahrnout vztahy mezi jednotlivými kontexty, je vyzkoušet počet nejlépe hodnocených kontextů a zlepšit hodnocení kontexty, na které vede odkaz [10].

Hodnoty metod z prvního a druhého řádu jsou od sebe rozlišení horním indexem, ve kterém je uvedené číslo řádu. Interpretační vektor je značem $ESA^{(k)}$, kde $k$ je číslo řádu. Interpretační vektor výrazu $t$ je $ESA^{(1)}(t) = \langle \omega_1^{(1)}, ..., \omega_n^{(1)} \rangle$.
*Second-level* interpretace výrazu $t$ je:

$$ESA^{(2)}(t) = \langle \omega_1^{(2)}, ..., \omega_n^{(2)} \rangle, \tag{4.4}$$

kde

$$\omega_i^{(2)} = \omega_i^{(1)} + \alpha \cdot \sum_{j:(d_i,d_j)} \omega_j^{(1)}. \tag{4.5}$$

Hodnota $\mathrm{N}(d_i, d_j)$ udává počet odkazů vedoucí z článku $d_i$ na článek $d_j$. Hodnota $\alpha$ by měla být menší než 1, aby se zajistilo, že připojené kontexty jsou brány s menší vahou.

Podle článku [10] byla použita hodnota $\alpha = 0,5$. Pokud tedy existuje více odkazů mezi články, může se stát, že hodnota z přidružených článků přes linky, převýší hodnotu původního článku, v kterém se slovo vyskytlo, a výslednou interpretaci výrazu to pouze zhorší. Proto jsem pro model druhého řádu změnil vzorec výpočtu hodnoty z přidružených článků. Nejprve se sečte počet odkazů, které vedou z článku. Dále se uloží počet odkazů mezi zdrojovým článkem $d_i$ a článkem, na který se odkazuje $d_j$. Tento krok se provede pro všechny články, na které je odkazováno. Nyní, podobně jako u *inverse document frequency*, se vypočte procentuální podíl, který mají odkazy mezi články $d_i$ a $d_j$. Tímto podílem je dále vynásobena TF-IDF hodnota pro všechny výrazy ze článku $d_j$, které jsou dále vynásobeny hodnotou $\alpha$ a přičteny k hodnotám výrazů v článku $d_i$.

Výsledný vzorec pro interpretaci druhého řádu vypadá tedy následovně:

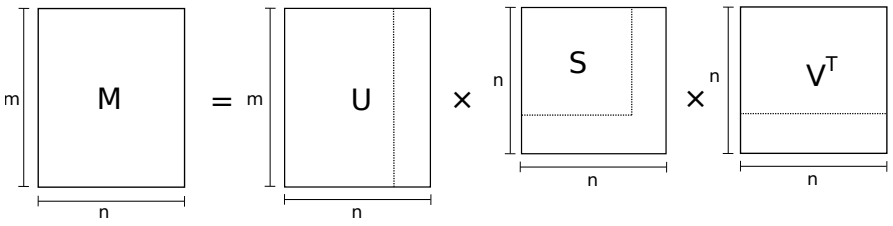
$$\omega_i^{(2)} = \omega_i^{(1)} + \alpha \cdot \sum_{j=0}^{n} \frac{\mathrm{N}(d_i, d_j)}{\mathrm{N}(d_i, *)} \cdot \omega_j^{(1)}, \tag{4.6}$$

kde $\mathrm{N}(d_i, *)$, představuje celkový počet odkazů z článku $d_i$.

# 5 Singulární rozklad

SVD (Singular Value Decomposition – singulární rozklad) [23] je rozklad matice na tři matice $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}$, viz obrázek 5.1. Rozklad matice $\boldsymbol{M}$, o rozměrech $m \times n$, lze provést přes singulární rozklad pokud je matice složena z reálných nebo případně komplexních čísel. Zároveň platí, že vynásobením rozložených matic vznikne zpět původní matice $\boldsymbol{M}$:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{5.1}$$




Obrázek 5.1: SVD rozklad matice na U, S, $V^T$.

Matice $\boldsymbol{U}$ je unitární matice s rozměry $m \times n$, kde každý sloupec matice odpovídá jednomu vlastnímu vektoru matice, která vznikne vynásobením původní matice $\boldsymbol{M}$ transponovanou maticí $\boldsymbol{M}^T$.

Matice $\boldsymbol{S}$ je diagonální matice $n \times n$, kde na hlavní diagonále jsou nezáporná reálná čísla a zbytek matice obsahuje samé nuly. Na diagonále jsou umístěny vlastní čísla matice $\boldsymbol{M}$ seřazené podle velikosti od nejvyšší hodnoty po nejmenší, tedy platí že:

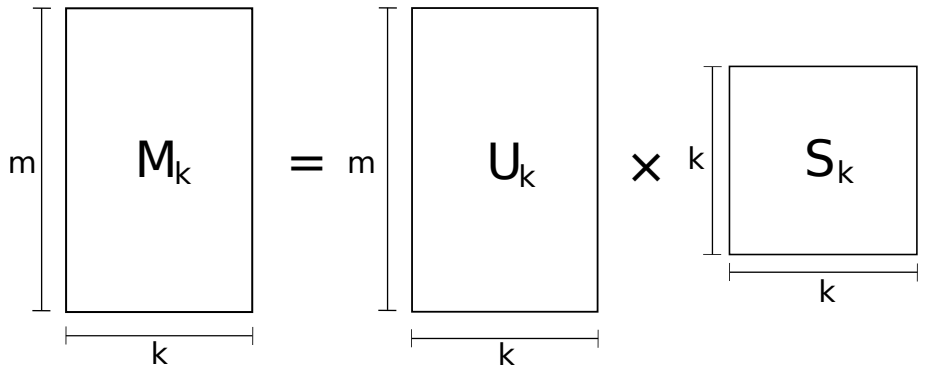
$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0. \tag{5.2}$$

Matice $\boldsymbol{V}^T$ je transponovaná unitární matice $\boldsymbol{V}$ o rozměrech $n \times n$, kde každý sloupec je vlastní vektor matice vzniklé vynásobením matic $\boldsymbol{M}^T$ a $\boldsymbol{M}$.

Pro redukci dimenze v metodě LSA je potřeba nejdříve redukovat matice $\boldsymbol{U}$ a $\boldsymbol{S}$. Matici $\boldsymbol{U}$ je potřeba redukovat na požadovanou cílovou dimenzi, označenou jako $k$, redukovaná matice má tedy rozměry $m \times k$ a je označena jako $\boldsymbol{U}_k$. Diagonální matici $\boldsymbol{S}$ je zapotřebí též redukovat, zde se kromě dimenze snižuje i počet řádků matice, které by po redukci dimenze byly stejně nulové. Rozměr redukované matice $\boldsymbol{S}_k$ je tedy $k \times k$.

Aproximace původní matice $\boldsymbol{M}$ s redukovanou dimenzí na $k$ se získá vynásobením matic $\boldsymbol{U}_k$ s maticí $\boldsymbol{S}_k$, součinem matic pak vznikne matice $\boldsymbol{M}_k$, viz 5.3.

$$\boldsymbol{M}_k = \boldsymbol{U}_k \times \boldsymbol{S}_k \tag{5.3}$$




Obrázek 5.2: Redukce matice na dimenzi $k$

Výsledná matice $\boldsymbol{M}_k$ pak obsahuje vektory pro jednotlivá slova, stejně jako původní matice $\boldsymbol{M}$, zde ovšem s redukovanou dimenzí, díky čemuž lze snížit paměťové nároky pro uložení a operace s maticí v paměti.

# 6 Kategorie jako kontexty

Základním předpokladem pro sémantickou analýzu u metod ESA a LSA je přítomnost dokumentů, které jsou brány jako kontext, a k nim přiřazený text. Tyto data musí obsahovat korpus, na kterém jsou metody testovány. Wikipedie tuto podmínku splňuje, neboť obsahuje články, které se dají nazvat jako dokumenty, což je dostatečné pro vytvoření sémantické interpretace výpočtem TF-IDF, které vyjadřuje souvislost slova k článku ve kterém se vyskytovalo. Wikipedie obsahuje daleko více informací, které nemusí být na první pohled vidět, a můžou mít význam pro lepší sémantickou interpretaci. Za dodatečnou informaci se považují například linky, tedy hypertextové odkazy mezi články Wikipedie. Těchto informací využívá metoda ESA pro zlepšení modelu. Kromě odkazů jsou u jednotlivých článků přiřazeny další informace, jako jsou kategorie. Na konci většiny dobře napsaných článků se nachází položka *Kategorie*, kde jsou všechny kategorie do kterých článek patří. Články mají většinou více kategorií než jednu, mezi články a kategoriemi je tedy relace *m*:*n*. Zároveň má Wikipedie poměrně složitý strom kategorií[1], který obsahuje hlavní kategorie a pak řadu podkategorií. Dalo by se předpokládat, že s takovým roztřídením článků do kategorií by se dala vylepšit metoda ESA. Na tomto předpokladu je tato kapitola založena. Jsou zde popsány způsoby, jakými bylo zkoušeno metodu ESA vylepšit. Následující metody nebyly ještě nikde představeny, jedná se tedy o zcela nové experimenty.

## 6.1 ESA s použitím kategorií

Pokud se budu držet definice, že množině kontextů $C_i$,...,$C_n$, je přiřazena množina dokumentů $d_i$,...,$d_m$ [10], pak kontexty budou jednotlivé kategorie a dokumenty budou články patřící do dané kategorie. Nyní se vytvoří matice $\boldsymbol{M} = |\boldsymbol{W}| \times |\boldsymbol{C}|$, kde jednotlivé sloupce jsou kategorie ($\boldsymbol{C}$), tedy jednotlivé položky vektoru pro slova ($\boldsymbol{W}$), které představují řádky matice.

Pro každé slovo se nejprve spočte jeho TF-IDF hodnota pro daný dokument (článek), stejně jako je tomu v původní implementaci ESA, která je popsána v kapitole 4. Hodnota se uloží do matice $\boldsymbol{T}$, která obsahuje TF-IDF hodnoty pro slova v dokumentech. Po výpočtu TF-IDF se hodnoty pro dokument $d_k$ přičtou do matice $\boldsymbol{M}$ pro všechny kategorie pro které platí: $d_k \in C_j$.

[1] `https://en.wikipedia.org/wiki/Portal:Contents/Categories`

Finální hodnota se tedy získá podle vzorce:

$$\boldsymbol{M}_{i,j} = \sum_{k=1}^{n} \mathrm{tf}(t_i, d_k) \cdot \mathrm{idf}(t_i). \tag{6.1}$$

Každý řádek matice $M$ je pak sémantický vektor pro výraz $t_i$.

Bohužel výsledky této úpravy metody ESA jsou velmi špatné, což lze vidět v kapitole 8. Při hledání případné chyby jsem zjistil, že na Wikipedii je mnoho kategorií, do kterých patří velké množství článků, například kategorie: *Living People, American films, The Football League players*, dále Wikipedie přidává vlastní kategorie, které nemají žádný sémantický význam a nejsou tedy vůbec vypovídající, například: *Year of birth missing (living people), Articles created via the Article Wizard, Articles containing video clips* a další. Kvůli tomu, že tyto kategorie jsou u velkého množství článků, tak je jim přiřazena vysoká váha pro spojení slova a kategorie. Slova jsou si tedy blízko sebe v sémantickém prostoru, i když významově spolu naprosto nesouvisí. Proto jsem se rozhodl pro změnu výpočtu váhy slova pro kategorii, aby kategorie, které jsou u většiny článků, měli malou váhu, neboť ve skutečnosti nejsou důležité.

## 6.2 PMI

Pro výpočet váhy, jakou má slovo ve spojení s kategorií, jsem tedy použil PMI (*Pointwise Mutual Information*). PMI určuje míru, jakou jsou na sobě dva jevy závislé [5]. PMI dvou náhodných veličin $t \in \boldsymbol{W}$ (množina slov) a $c \in \boldsymbol{C}$ (množina kategorií) se vypočte vzorcem:

$$\mathrm{pmi}(t, c) = \log \frac{\mathrm{p}(t, c)}{\mathrm{p}(t)\mathrm{p}(c)} = \log \frac{\mathrm{p}(t|c)}{\mathrm{p}(c)}. \tag{6.2}$$

Čitatel $\mathrm{p}(t|c)$ vyjadřuje míru závislosti výskytu slova na kategorii, jedná se o podmíněnou pravděpodobnost, která se vypočte:

$$\mathrm{p}(t|c) = \frac{\mathrm{N}(c, t)}{\mathrm{N}(t)}, \tag{6.3}$$

kde $\mathrm{N}(c, t)$ je počet výskytů slova $t$ v kategorii $c$.
Jmenovatel $\mathrm{p}(c)$ určuje pravděpodobnost výskytu kategorie, kde vzorec je následující:

$$\mathrm{p}(c) = \frac{\mathrm{N}(c)}{\mathrm{N}(*)}. \tag{6.4}$$

Čitatel $\mathrm{N}(c)$ jsou všechna slova z korpusu, která patří do kategorie $c$. Pokud tedy článek $d$ má kategorii $c$, přičte se do $\mathrm{N}(c)$ počet slov v dokumentu $d$. Ve jmenovateli $\mathrm{N}(*)$ je počet slov v celém korpusu dat.

Hodnota $\mathrm{pmi}(t, c)$ může nabývat velkého rozsahu hodnot. Proto je lepší PMI znormalizovat, výsledkem poté bude NPMI (*Normalized Pointwise Mutual Information*), které se vypočte vydělením hodnoty $\mathrm{pmi}(t, c)$ jeho entropií ($\mathrm{h}(t, c)$):

$$\mathrm{npmi}(t, c) = \frac{\mathrm{pmi}(t, c)}{\mathrm{h}(t, c)}. \tag{6.5}$$

Pro entropii je pak vzorec následující:

$$\mathrm{h}(t, c) = -\log_2 \frac{\mathrm{N}(t)}{\mathrm{N}(*)}. \tag{6.6}$$

Hodnoty npmi se pohybují v rozmezí [-1; 1]. Hodnota -1 nastává, pokud se slovo $t$ nikdy nevyskytuje v kategorii $c$. Hodnota 0 znamená, že slovo $t$ a kategorie $c$ jsou na sobě nezávislé. Pokud $\mathrm{nmpi}(t, c)$ má hodnotu 1, jsou na sobě maximálně závislé a vždy se objevují spolu. Slovo $t$ se tedy vyskytuje vždy a pouze v kategorii $c$.

Všechny hodnoty nmpi se uloží do matice $\boldsymbol{M} = |\boldsymbol{W}| \times |\boldsymbol{C}|$, každý řádek $i$ pak opět představuje vektor v sémantickém prostoru pro slovo $t$. Zde v porovnání s metodou ESA využívající kategorie jako kontexty, která byla popsána výše, dosahuje tato metoda mnohem lepších výsledků. Metoda PMI tedy efektivně snižuje váhu pro kategorie, které jsou velmi časté, a tedy bezvýznamné z hlediska sémantické interpretace.

## 6.3 Redukce dimenze

Vzhledem ke vztahům, jaké mají články a kategorie, že jeden článek má hned několik kategorií, se zvyšuje hustota a dimenze matice, což vede k větším požadavkům na paměť, jak pro uložení, tak pro operace nad maticí. Pro snížení nároků na paměť se tedy sníží dimenze matice pomocí rozkladu SVD, jak je tomu u metody LSA. Matice s redukovanou dimenzí dosahuje i lepších výsledků. Nejen tedy, že sníží nároky na paměť, ale aplikace SVD ještě výsledky vylepší.

Tuto metodu jsem nazval LSC (*Latent Semantic Categories*). *Semantic Categories* je v názvu proto, neboť pro sémantickou interpretaci jsou použity články z Wikipedie rozřazené do kategorií. Kategorie jsou zde jednotlivé kontexty. Na tuto matici je dále aplikována metoda SVD, podobně jako

u metody LSA, pro redukci dimenze a snížení paměťových nároků. Původní hodnoty jsou skryty, proto slovo *Latent* je i v názvu této meotdy.

# 7 Data a evaluace

Metody pro sémantickou analýzu jsou založené na trénování na velkých korpusech dat, které je potřeba připravit do zpracovatelné podoby. Všechny metody zpracování textu jsou popsány níže. Dále je potřeba výsledek metody změřit a ověřit, do jaké míry odpovídá lidskému vyhodnocení výrazu. Pro změření, jak jsou si výsledky lidí a algoritmu podobné, se používají korelace, které jsou popsány níže. Data, na kterých se tyto korelace měří, jsou převzaty ze standardních datasetů pro významovou podobnost slov.

## 7.1 Trénovací data

Pro trénování vlastní implementace jsem použil dump Wikipedie z data 1. 6. 2016. Články z Wikipedie a další užitečné informace je možné stáhnout z webové stránky: (dumps.wikimedia.org) v jednom souboru XML formátu. Velikost tohoto souboru po dekompresi je 56 GB a obsahuje 5 164 793 článků a 1 759 101 849 slov. Množství dat pro trénování je tedy obrovské, mnohem větší, než bylo použito pro původní implementaci metody ESA v listopadu 2005, kdy velikost Wikipedie byla 1,9 GB a obsahovala 910 989 článků. Pro použití dumpu je nejprve potřeba provést předzpracování korpusu a odstranit kódové značky, které se používají k formátování textu na Wikipedii. K tomuto předzpracování jsem použil již hotovou implementaci *Wikipedia Extractor*[1], která z XML souboru odstranila všechny přebytečné značky a zanechala pouze oddělené články, označené jejich id hodnotou, a čistý text.

Pro trénování implementace na českém korpusu jsem použil XML dump Wikipedie z data 1. 3. 2017. Česká verze Wikipedie je též dostupná na stránce (dumps.wikimedia.org). Velikost české Wikipedie je podstatně menší než její anglická verze, velikost XML souboru je pouze 2,5 GB a obsahuje 375 262 článků a 88 745 854 slov.

Data je nejprve nutné upravit do zpracovatelné podoby, tím se změní i velikost trénovacího korpusu. Je potřeba odstranit slova bez sémantického významu jako předložky, spojky, zájmena. Na Wikipedii se též nachází mnoho krátkých článků, které nemůžou být považovány za hodnotný dokument, proto jsou články obsahující méně než 100 slov z korpusu odstraněna. Pro trénování vlastní implementace na angličtině bylo použito pro výpočet vektorů 2 383 189 článků a 300 000 nejčetnějších slov. V případě výpočtu

[1] `http://medialab.di.unipi.it/wiki/Wikipedia_Extractor`

PMI s využitím kategorií jako jednoho kontextu, bylo využito 300 000 nejčetnějších kategorií. Tento počet byl zvolen, aby co nejblíže odpovídal původní implementaci *ESA-Wikipedia (March 26, 2006 version)*, kde po preprocessingu zbylo 241 393 článků použitých jako jednotlivé kontexty a 389 202 různých slov [9].

Pro trénování implementace na české Wikipedii bylo použito 207 031 článků a 300 000 nejčetnějších slov. V případě výpočtu PMI, kde kontext byla jedna celá kategorie a všechny články patřící do této kategorie, bylo použito 90 474 kategorií.

## 7.2 Předzpracování dat

Text, na kterém jsou metody reprezentace významu slov trénovány, je třeba upravit do zpracovatelné podoby. S tímto souvisí následující metody zpracování, jako je odstranění *stop-words*, tokenizace a lemmatizace textu, *Part-of-Speech tagging* (POS značkování) a nakonec i *stemming*.

Tokenizace je proces rozdělení většího množství textu, jako jsou například věty a odstavce na jednotlivá slova. Při tokenizaci se zároveň odstraňuje interpunkce. Výstupem tokenizace jsou pak tokeny (jednotlivá slova), která jsou dále zpracována na POS tagging nebo lemmatizaci.

*Stop-words* jsou slova, která nemají žádnou hodnotu v rámci reprezentace slov. Často se vyskytují ve všech kontextech, nejčastěji to jsou předložky, spojky, zájmena, sloveso „je“ a další [18]. Z těchto důvodů jsou z textu odstraňovány.

POS značkování přiřazuje slovní druhy (podstatná jména, přídavná jména, slovesa, atd.) k jednotlivým slovům. Při předzpracování se tak odstraní slovní druhy, které nemají žádnou vypovídající hodnotu vzhledem k sémantickému významu, například zájmena, předložky a spojky.

Lemmatizace je proces, při kterém se určí základní tvar slova neboli lemma [18]. Je potřeba zlemmatizovat celý text, aby se zde nenacházela stejná slova s různými tvary.

Pro tokenizaci, POS tagging a lemmatizaci textu jsem využil knihovnu *Stanford CoreNLP* [19] napsanou v jazyce *Java*. Knihovna má open source licenci a je používána ve velké míře ve výzkumu v oblasti zpracování přirozeného jazyka a též v aplikacích pro komerční účely.

POS značkování a lemmatizace byla používána pouze u anglického jazyka. Na češtině tyto metody nefungují příliš dobře, kvůli větší flexi jazyka.

### 7.2.1 Stemming

Pojem stemming (z angl. *stem* – kmen) znamená přibližné nalezení kmene slova pomocí odstranění jeho afixů. Nástroj, který stemming provádí, se nazývá stemmer. Například slovo „stůl“ může mít tvar „stolu“ nebo „stolem“, stemmer odstraní koncovku a zanechá pouze kmen slova, který ale nemusí dávat žádný věcný význam. V tomto případě stemmer určí jako kmen slova „stol“. Stemming je velmi podobný procesu lemmatizace, která určuje základní tvar slova, zatímco stemming jeho kmen.

Pro stemming se používají dva různé přístupy: pravidlový a statistický [2]. Stemmer založený na pravidlovém přístupu se pokouší převést slovo na stem použitím specifických pravidel pro daný jazyk. Tato pravidla jsou vytvořena manuálně lingvisty. Statistický stemmer je trénovaný na velkém množství dat bez přítomnosti učitele. Pravidlový stemmer bývá většinou kvalitativně lepší, zejména u komplexnějších jazyků. Vytvoření takového stemmeru ovšem vyžaduje přítomnost experta na lingvistiku pro daný jazyk, čímž se tento postup stává časově více náročným.

Výhodou statistického stemmeru je použití velkého množství textu, z kterého je možné zachytit i méně časté případy flexe. Přesto si statistický stemmer pravděpodobně neporadí se slovy jako „sing“ a „sang“, „foot“ a „feet“ a dalšími, ačkoliv je jejich výskyt v angličtině poměrně častý [2].

Použitý *High Precision Stemmer*[2] se skládá ze dvou fází. V první fázi se generují data rozdělením do skupin podle sémantického významu. V druhé fázi jsou pak tyto skupiny použity jako trénovací data pro klasifikátor. Klasifikátor vybere ty slova, která obsahují nejvyšší míru informace o užitečnosti afixů. Tato slova jsou poté použita jako soubor pravidel, podle kterých je pak určován stem slova u ostatních slov [2].

## 7.3 Testovací data

K otestování implementace se používají datasety, pro testování angličtiny jsou datasety: *WordSimilarity-353* [7], *RG-65* [22] a *Simlex999* [11], které jsou popsány níže. Tyto datasety jsem použil též k otestování implementace *esalib*[3], kde byly k dispozici již vypočtené vektory pro jednotlivá slova, jejímž autorem je Lukáš Žilka. Pro trénování metody zde byla použita Wikipedie z roku 2005.

Datasety obsahují několik párů slov a číslo, které odpovídá průměrnému hodnocení, jaké udělili lidé. Toto číslo představuje míru podobnosti tohoto

[2] `http://liks.fav.zcu.cz/HPS/`

[3] `https://github.com/ticcky/esalib`

páru slov. Tyto seznamy mohou být použity k trénování, nebo právě k testování algoritmů zabývajících se sémantickou podobností výrazů.

Jedním z datasetů je *WordSimilarity-353* [7], který obsahuje 353 párů podstatných jmen z různých oblastí témat. Každý pár slov nezávisle posoudilo 13 – 16 porotců, kteří měli vysokoškolský titul v oblasti mateřského (anglického) jazyka nebo velmi plynulou angličtinu. Každému páru přiřadili číslo od 0 (pár slov si není vůbec podobný) do 10 (jsou si velmi podobné nebo identické). U každého páru pak tedy bylo 13 – 16 hodnocení, které se zprůměrovalo do jediného skóre, které je přiřazeno páru slov.

Dalším datasetem je *RG-65* [22], který obsahuje 65 párů slov. Věcný význam každého páru slov je ohodnocen 51 lidmi na škále od 0 (nejméně podobné) až 4 (velmi podobné nebo identické).

Nejrozsáhlejší dataset je *Simlex999* [11], který používá vyšší skóre pro slova podobná a nízké pro slova příbuzná. Na rozdíl od datasetu *WordSim353* přísněji hodnotí páry slov, která jsou si navzájem příbuzná. Například pár slov *clothes – closet* má v datasetu *Simlex999* hodnocení 1,96 zatímco v datasetu *WordSim353* hodnocení 8,00. Celkově dataset obsahuje 999 párů slov, z toho 666 párů jsou podstatná jména, 222 slovesa a 111 přídavná jména. *Simlex999* tedy klade i důraz na podobnost různých slovních druhů.

Pro testování češtiny se používají datasety *RG-65* [13] a *WordSimilarity-353* [6], které jsou přeloženy z angličtiny a zkontrolovány anotátorama.

## 7.4 Evaluace

Naměřená data je potřeba vyhodnotit (evaluovat). Jelikož se jedná o dvě řady reálných čísel (podobnosti slov definované lidmi a odhadnuté automaticky), je vhodné použít korelaci. Korelace hodnotí vzájemný vztah mezi dvěma veličinami. Míru korelace pak vyjadřuje korelační koeficient na stupnici od -1 do 1. Čím blíže hodnotě -1 nebo +1, tím víc si jsou veličiny podobné. V hodnotě +1 jsou identické a v hodnotě -1 jsou si opačně podobné, naopak v hodnotě 0 nemají žádnou spojitost. Pro vyhodnocení výsledků změřených na datasetech se používají korelace: *Pearsonova* a *Spearmanova*.

### 7.4.1 Pearsonova korelace

Pearsonova korelace je míra lineární závislosti mezi veličinami. Používá se při měření podobnosti mezi dvěma sadama dat obsahujících $n$ prvků

$\{x_1,...,x_n\}$ a $\{y_1,...,y_n\}$. Míra korelace se značí $r$ a je dána vzorcem:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{7.1}$$

kde $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ a $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

### 7.4.2 Spearmanova korelace

Spearmanova korelace je neparametrický test, který se používá k měření asociace mezi dvěma veličinami. Jednotlivé prvky $x_i$ a $y_i$ obou veličin se uspořádají vzestupně podle hodnot, následně se přiřadí pořadová čísla $p_i$ a $q_i$. Hodnota korelace je pak dána vzorcem:

$$\rho = 1 - \frac{6\sum_{i=1}^{n}(p_i - q_i)^2}{n(n^2 - 1)}, \tag{7.2}$$

kde $n$ je počet hodnot ve veličinách.

Spearmanovu korelaci lze též spočítat jako Pearsonovu korelaci pořadových čísel $p_i$ a $q_i$.

# 8 Experimenty

V této kapitole jsou popsány prováděné experimenty a uvedeny jejich výsledky.

## 8.1 Výpočetní složitost

K vytvoření kvalitní a vypovídající sémantické interpretace je zapotřebí velké množství dat, což anglická verze Wikipedie s velikostí XML dumpu 56 GB rozhodně splňuje. Zpracování takového množství dat je téměř nereálné na většině osobních počítačů. Pouze uložení matice s TFIDF hodnotami pro celou Wikipedii, za předpokladu použití datového typu *float* pro uložení hodnot, bylo zapotřebí minimálně 120 TB paměti RAM. Podmínku k takovému množství volné paměti RAM příliš výpočetních strojů nesplňuje. Vzhledem k tomu, že matice je řídká a obsahuje pouze necelých 0,2 % hodnot z celé matice, které jsou nenulové, používá se pro reprezentaci matice v paměti *HashMap*, které jsou paměťově méně náročné. Dále nemá význam vést záznamy o slovech, které se vyskytují pouze vzácně, nebo o článcích, které mají pouze pár slov. Přes všechny tyto úpravy byly nároky na paměť, pro vytvoření vektorů a reprezentaci matice, 38 GB, což je stále hodně pro běžné osobní počítače, ale pro výpočetní servery to již není velké množství. Proto většina výpočetních úloh byla pouštěna na metacentru[1], kde je možné pouštět úlohy pro vědecké a výzkumné činnosti.

MetaCentrum je virtuální organizace sdružující všechny uživatele registrované v MetaCentru. Členem metacentra se můžou se stát zaměstnanci a studenti všech vědecko-výzkumných institucí v České republice. MetaCentrum sdružuje a poskytuje svým uživatelům výpočetní a úložné kapacity ze zapojených institucí, jako jsou například: CESNET, CERIT-SC, FI MU, Fyzikální ústav AV ČR, ZČU. Zároveň se stará o provoz a rozvoj výpočetních strojů, datových uložišť a aplikačních programů. Dále se pak stará o uživatelskou podporu a bezpečnost celé infrastruktury systému. Uživatelé zde nejčastěji provádí výpočetně náročné úlohy z oblastí: výpočetní chemie, materiálové a strukturní simulace, simulace proudění plynů a kapalin, rozpoznávání a generování řeči, fyzikální geodézie, ekologické modelování, zpracování videa, data mining, analýza lékařských obrazů a další.

Kromě vysokých paměťových nároků byly úlohy i časově náročné na vý-

[1] `https://metavo.metacentrum.cz/cs/`

počet, například zpracování celého textu, což obsahovalo procesy tokenizace, POS značkování a lemmatizace, zabralo 150 hodin na CPU, tedy téměř týden čistého času. Další paměťově a časově náročnou úlohou byla implementace SVD pro jazyk *Java* z knihovny *Commons Math*[2], která umí spočítat rozklad pouze celých matic, bylo tedy zapotřebí uložit do paměti celou matici včetně nulových hodnot. Aby bylo možné matici spočítat dokonce i na Metacentru, bylo zapotřebí zredukovat matici a použít pouze malou část dat. S maticí o 100 000 slovech v 50 000 článcích, si implementace neporadila za dobu dvou týdnů na CPU, kdy spotřebovávala 160 GB RAM. Implementaci z *Javy* jsem poté nahradil implementací v jazyce C[3], která dokázala počítat SVD na *sparse* (řídkých) maticích. Tato implementace kromě výpočtu rozkladu jednotlivých matic zároveň redukovala jejich dimenze na požadovanou hodnotu. Zde už byly časové i paměťové nároky podstatně nižší, podle velikosti dimenze, na kterou bylo potřeba redukovat původní matici, se paměťové nároky pohybovali mezi 3 – 12 GB RAM a délka výpočtu na CPU byla necelých 10 minut (pro výpočet cílové dimenze 100) až 9 hodin (pro výpočet matic s dimenzí 1000).

Další časově náročnou operací bylo vytvoření slovníků, tedy seznamu slov z korpusu a vybrání těch nejčetnějších. Tato operace zabrala na metacentru přibližně jeden den času na CPU. Zároveň jsem se snažil co nejvíce optimalizovat běh, aby trénování na korpusu bylo co nejrychlejší. První implementace na malých vzorcích dat (10 MB) trvala řádově 2 hodiny, ale čas sem postupně snížil až na řádově desítky vteřin. Běh standardního modelu ESA na celé Wikipedii, bez počítání odkazů mezi články a s dostupným slovníkem slov, trvá zhruba 40 minut. Časy běhu jednotlivých úloh se velmi často měnily z důvodu různě výkonných strojů, kde byly momentálně spuštěny.

Během celé implementace modelů a všech experimentů jsem na Metacentru spustil 607 úloh s celkovou spotřebou 164,6 dnů na CPU. Do těchto úloh ale není zahrnuto testování v průběhu implementace na menších vzorcích dat, které jsem pouštěl na svém osobním počítači, a zároveň také trénování na české verzi Wikipedie, která je podstatně menší a obsahuje tak méně článků i slov. Proto byl potřebný výpočetní výkon a paměťové nároky nižší.

## 8.2 Výsledky

K interpretaci výsledků jsou použity korelace, které jsou popsány v kapitole 7. V následujících tabulkách s výsledky korelací jsou použity některé

[2] http://commons.apache.org/proper/commons-math/

[3] `https://tedlab.mit.edu/~dr/SVDLIBC/`

zkratky: PC (Pearsonova korelace), SC (Spearmanova korelace), WS353 (dataset *WordSimilarity-353*) a SL999 (dataset *Simlex999*).

V tabulce 8.1 jsou uvedeny výsledné hodnoty korelací pro anglické datasety *RG-65*, *WordSimilarity-353* a *Simlex999*. Výsledky implementací *ESA-Wikipedia (November 11, 2005 version)* a *ESA-Wikipedia (March 26, 2006 version)* jsou převzaty z článku od E. Gabrilovich [10], neboť zmíněné implementace nejsou volně dostupné. Pro metody *Word2vec* (*Skip-gram* a *Continous Bag-of-Words*) a *GloVe* jsou hodnoty korelací převzaty z článku *GloVe: Global Vectors for Word Representation* [21]. Implementace *esalib*, která je zmíněna výše, je vypočtena na dostupných vektorech, které jsou výsledkem implementace trénované na datech z Wikipedie z roku 2005. Pro ostatní metody jsou pak výsledky převzaty z webové stránky aclweb.org, kde jsou rozděleny výsledky pro datasety *RG-65*, *WordSim-353* a *Simlex999*.

Výsledky uvedené v tabulce 8.1 se nedají příliš dobře porovnávat, jelikož každá implementace používala jinak velký korpus dat. Například implementace *GloVe* a *Word2Vec* jsou testovány na korpusu o velikosti 6 miliard slov, což je podstatně více, než má anglická verze Wikipedie s velikostí 1,76 milard slov. Některé implementace byly dokonce testovány i na menších korpusech, např. metody ESA z roku 2005. Nicméně výsledky v tabulce postačí pro přehled některých z metod na zpracování přirozeného jazyka, a jak jsou v tomto úkolu úspěšné.

| | RG-65 | | WS353 | | SL999 | |
|---|---|---|---|---|---|---|
| **Model** | **PC** | **SC** | **PC** | **SC** | **PC** | **SC** |
| ESA [9] | 0,716 | 0,749 | 0,50 | 0,75 | - | - |
| ESA [10] | - | - | 0,71 | 0,74 | - | - |
| ESA [10] | - | - | 0,72 | 0,75 | - | - |
| ESA [16] | - | - | - | - | 0,145 | 0,271 |
| esalib | 0,570 | 0,801 | 0,50 | 0,74 | - | - |
| LSA [14] | 0,644 | 0,609 | 0,56 | 0,58 | 0,233 | - |
| Word2vec (Skip-gram) [21] | - | 0,697 | - | 0,63 | - | - |
| Word2vec (CBOW) [21] | - | 0,682 | - | 0,57 | - | - |
| GloVe [21] | - | 0,778 | - | 0,66 | - | - |
| H& S [12] | 0,732 | 0,813 | 0,36 | 0,30 | - | - |
| Lin [17] | 0,834 | 0,788 | 0,36 | 0,35 | - | - |

Tabulka 8.1: Výsledky Pearsonovy a Spearmanovy korelace na datasetech *RG-65*, *WordSim353* a *Simlex999*. Výsledky získány z dostupných zdrojů.

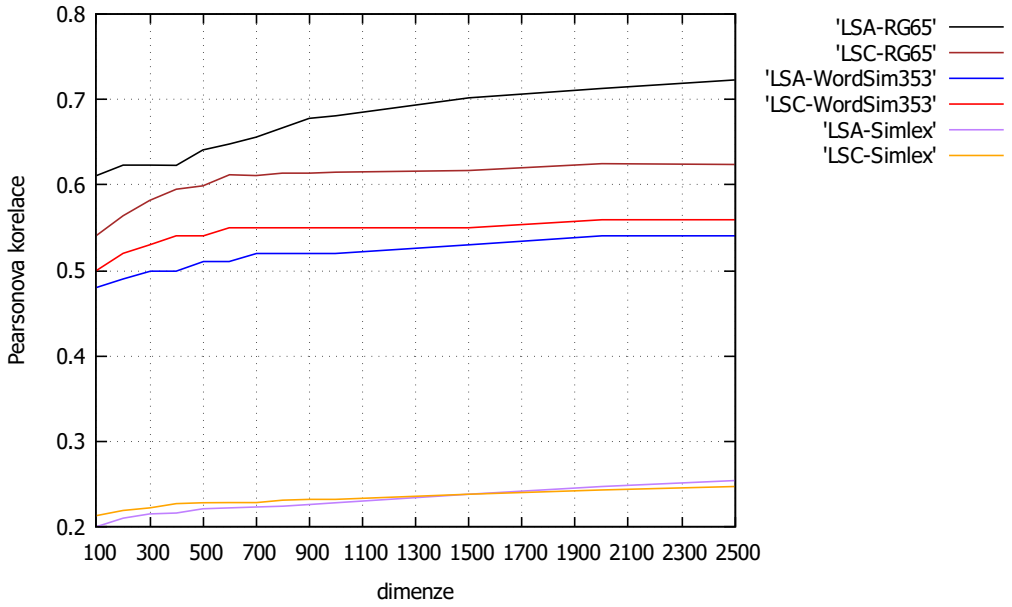
| | RG-65 | | WS353 | | SL999 | |
|---|---|---|---|---|---|---|
| **Model** | **PC** | **SC** | **PC** | **SC** | **PC** | **SC** |
| ESA | 0,569 | 0,758 | 0,41 | 0,51 | 0,135 | 0,170 |
| ESA (alpha = 0,5) | 0,556 | 0,758 | 0,39 | 0,49 | 0,111 | 0,172 |
| ESA (alpha = 0,25) | 0,559 | 0,761 | 0,39 | 0,49 | 0,111 | 0,172 |
| ESA kategorie | 0,181 | 0,207 | 0,09 | 0,11 | - | - |
| PMI | 0,507 | 0,579 | 0,49 | 0,48 | 0,195 | 0,216 |
| PMI (normalizované) | 0,508 | 0,579 | 0,49 | 0,50 | 0,195 | 0,216 |
| PMI (normalizované, alpha = 0,5) | 0,502 | 0,573 | 0,48 | 0,49 | 0,195 | 0,215 |
| LSA | 0,684 | 0,723 | 0,52 | 0,54 | 0,228 | 0,254 |
| LSC | 0,627 | 0,624 | 0,55 | 0,56 | 0,231 | 0,247 |

Tabulka 8.2: Výsledky Pearsonovi a Spearmanovi korelace na datasetech *RG-65*, *WordSim353* a *Simlex999* pro anglický jazyk.

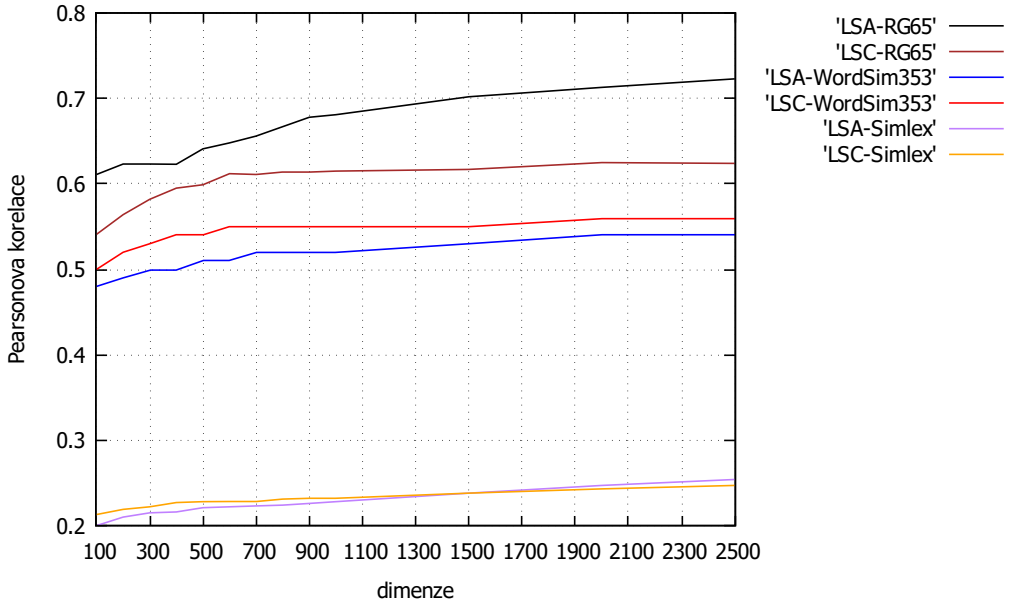
Výsledky mých vlastních implementací, na anglické verzi Wikipedie, jsou uvedené v tabulce 8.2. Všechny experimenty pro anglický jazyk jsem tak prováděl na stejných datech. Korpus byl nejprve ztokenizován a zlemmatizován, přičemž se také odstranily zájmena, předložky, spojky a slova nepatřící

do anglického jazyka. Po předzpracování se vytvořily slovníky. Ty se vytvářely z článků, které měly alespoň 100 slov, aby se odfiltrovaly krátké články, které nemají tak dobrou vypovídající hodnotu jako články, které jsou delší. Celkem tak bylo použito 2 383 189 článků. Z těchto článků se poté vytvořil seznam 300 000 nejčetnějších slov. Četnost se zde neporovnává podle celkových výskytů v korpusu, ale podle toho, v kolika článcích se slovo vyskytlo. Pro výpočet PMI z kategorií se ještě vytvořil slovník 300 000 nejčetnějších kategorií, které obsahují nejvíce článků. Pro LSA bylo potřeba matici s vektory zmenšit, jinak by byly paměťové nároky na SVD příliš velké. Vytvořil jsem tedy menší matici s dimenzí 300 000, kde bylo možné SVD spočítat. Na této matici jsou pak naměřeny metody LSA.

Pro dataset *RG-65* vyšla nejlépe metoda ESA se Spearmanovou korelací 0,758. Pro *WordSim353* dosáhla implementace ESA o něco horších výsledků, to může být dané jiným předzpracováním dat, než bylo v původní implementaci. Rovněž výsledky implementace se započtenými odkazy jsou o trochu horší. To může být zapříčeněno nesmyslnými odkazy mezi články, které by se daly ještě lépe odfiltrovat, aby informace z přidružených článků byly více relevantní. Pro *WordSim353* a *Simlex999* si vedla hodně dobře metoda LSC, kde se výsledky zlepšovaly s aproximací na vyšší dimenzi. Celkově původní metody ESA i PMI se zlepšily po aplikaci SVD pro redukci dimenze, nejen, že se snížily paměťové nároky pro práce s maticemi, ale zároveň zlepšily výsledky. SVD jsem aplikoval pro dimenze od 100 až 2500. Pro prvních 1000 sem zvyšoval dimenzi o 100, jakmile jsem dosáhl dimenze 1000 provedl jsem ještě SVD na dimenzi 1500, 2000 a 2500. Celkový přehled korelací u metod LSA a LSC, které jsou závislé na velikosti dimenze, je vidět na obrázku 8.1 a 8.2.




Obrázek 8.1: Výsledky Spearmanovy korelace implementací LSA a LSC na anglických datasetech (*RG-65*, *WordSim353*, *Simlex999*) pro různé velikosti dimenze.




Obrázek 8.2: Výsledky Pearsonovy korelace implementací LSA a LSC na anglických datasetech (*RG-65*, *WordSim353*, *Simlex999*) pro různé velikosti dimenze.

| | RG-65 | | WS353 | | WS-CZ | |
|---|---|---|---|---|---|---|
| **Model** | **PC** | **SC** | **PC** | **SC** | **PC** | **SC** |
| ESA | 0,704 | 0,746 | 0,32 | 0,51 | 0,34 | 0,58 |
| ESA (alpha = 0,5) | 0,712 | 0,742 | 0,32 | 0,51 | 0,34 | 0,58 |
| ESA (alpha = 0,25) | 0,704 | 0,746 | 0,32 | 0,51 | 0,34 | 0,58 |
| ESA Stemming | 0,381 | 0,828 | 0,38 | 0,49 | 0,40 | 0,54 |
| LSA | 0,650 | 0,665 | 0,46 | 0,44 | 0,50 | 0,50 |
| LSA Stemming | 0,656 | 0,701 | 0,49 | 0,51 | 0,52 | 0,55 |
| PMI | 0,583 | 0,619 | 0,41 | 0,45 | 0,44 | 0,51 |
| PMI (normalizované) | 0,583 | 0,620 | 0,41 | 0,45 | 0,44 | 0,51 |
| PMI Stemming | 0,474 | 0,695 | 0,43 | 0,44 | 0,46 | 0,49 |
| PMI (normalizované) Stemming | 0,474 | 0,695 | 0,43 | 0,44 | 0,46 | 0,49 |
| LSC | 0,567 | 0,618 | 0,43 | 0,42 | 0,49 | 0,48 |
| LSC Stemming | 0,592 | 0,592 | 0,48 | 0,48 | 0,51 | 0,52 |

Tabulka 8.3: Výsledky implementací otestovaných na českých datasetech *RG-65*, *WordSim353* a *WordSim*.

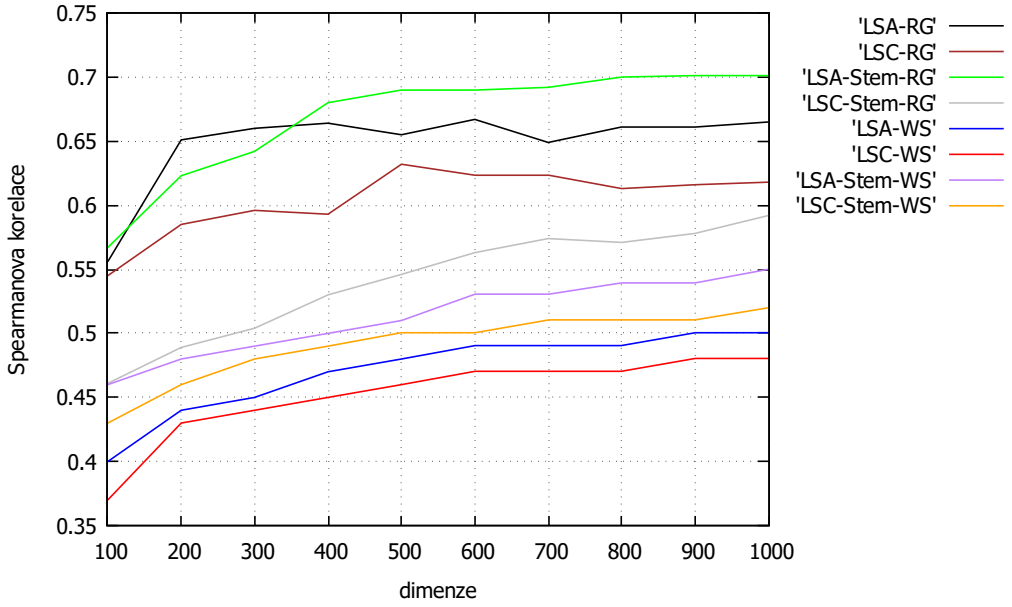
Výsledky experimentů na české Wikipedii jsou umístěny v tabulce 8.3. Podobně jako u anglických dat, i zde jsou všechny experimenty prováděny na jednotných datech. Na rozdíl od angličtiny nebyla provedena lemmatizace textu, neboť pro češtinu je lemmatizace těžší a není tak přesná jako u anglického jazyka. Z tohoto důvodu se zde při předzpracování provádí tokenizace textu a místo lemmatizace je zde použit stemming.

Stejně jako při anglických experimentech, i zde jsem nejprve odstranil články obsahující méně než 100 slov a poté vytvořil slovník 300 000 nejčetnějších slov. Ovšem vzhledem k tomu, že česká verze Wikipedie je mnohem menší než její anglická verze, není zde potřeba dále zmenšovat dimenzi kvůli aplikaci SVD, neboť zde bylo použito pouze 207 031 článků a 90 474 kategorií.

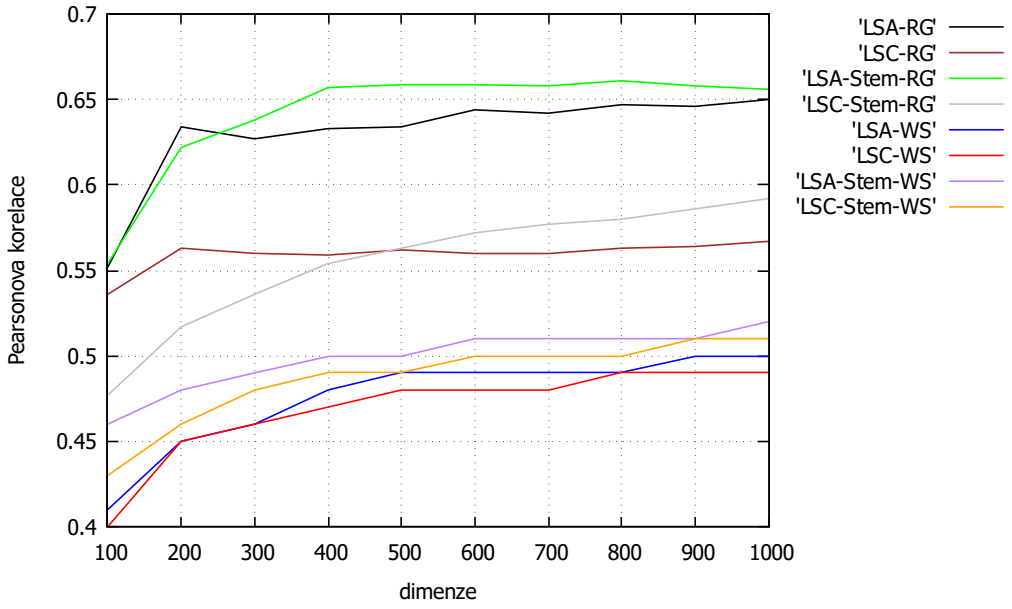
Dataset *WordSim353* po přeložení do českého jazyka obsahoval několik slov, která se v korpusu nevyskytovala, proto jsem pro zpřesnění výsledků tato slova z datasetu odstranil. Korelace pro tento dataset jsou poté uvedeny v pravém sloupci tabulky pod zkratkou *WS-CZ*.

Ve výsledcích si pak nejlépe vedla metoda ESA, která dosahovala Spearmanovy korelace 0,746 a Pearsonovy korelace 0,704 pro dataset *RG-65*. Pro *WordSim353* byla pak nejlepší metoda LSA s oběma korelacemi ko-

lem 0,45. Ovšem i metody založené na kategoriích si zde vedly velmi dobře a výsledky byly jenom o pár setin horší než klasické metody ESA a LSA. Po otestování čistě ztokenizovaného textu jsem také zkusil vliv stemmingu na tyto metody. Zde se nedá s jistotou říci, zda byly výsledky se stemmingem lepší. Například v metodě ESA se zvýšila Spearmanova korelace na hodnotu 0,828, což je už velmi vysoké číslo, ale Pearsonova korelace se snížila na 0,381. Obdobně to bylo i na jiných experimentech. Stejně jako na anglickém jazyce, i zde jsem aplikoval SVD pro metody ESA a PMI. Zde už nebyl nárust výsledků tak velký jako u angličtiny, ale mírné zlepšení zde bylo. Stejně tak snížení paměťové náročnosti metod. Celkový přehled metod LSA a LSC, pro různé velikosti dimenze, je vidět na obrázku 8.3 a 8.4.




Obrázek 8.3: Výsledky Spearmanovy korelace implementací LSA a LSC testovaných na češtině pro různé velikosti dimenze.




Obrázek 8.4: Výsledky Pearsonovy korelace implementací LSA a LSC testovaných na češtině pro různé velikosti dimenze.

# 9 Závěr

V práci jsem se seznámil s metodami pro reprezentaci významu přirozeného jazyka, konkrétněji pak s metodou Explicitní sémantická analýza, kterou jsem implementoval. Metoda využívá dat z Wikipedie, konkrétně obsahu článků, pro sémantickou interpretaci slov, která je rezprezentována vektorem s vysokou dimenzí. Kromě této metody jsem implementoval i metodu LSA. Dále jsem vytvořil metodu, která používala pro výpočet PMI místo TF-IDF. Na tuto metodu jsem zároveň aplikoval SVD a vznikla tak nová metoda LSC. Všechny tyto metody jsou založené na trénování bez učitele, stačí jim tedy pro trénování velký korpus dat. V tomto případě Wikipedie.

Výsledek metody byl měřen na datasetech *RG-65*, *WordSimilarity-353* a *Simlex999*. Výsledky metody pro *RG-65* jsou na úrovni nejlepších známých metod. Výsledky pro datasety *WordSimilarity-353* a *Simlex999*, pak byly trošku horší.

Po implementaci jsem začal provádět různé experimenty. Rozšíření původní implementace o interpretaci druhého řádu, což zahrnovalo přidání hodnot z přidružených článků přes odkazy mezi články Wikipedie, které výsledky příliš nevylepšilo. Dále pak použití aplikace SVD, na matici vektorů, pro redukci dimenze. Tato metoda se pak nazývá LSA. Pro tuto metodu byly výsledky na datasetu *RG-65* lepší (Pearsonova 0,65 a Spearmanova 0,68), než původní implementace (Pearsonova 0,64 a Spearmanova 0,61).

Dalším velkým experimentem bylo zahrnutí kategorií z Wikipedie do sémantické interpretace. Výpočet byl napřed prováděn přes hodnoty TFIDF, ale když se tento způsob výpočtu příliš neosvědčil, tak jsem se rozhodl pro výpočet pomocí PMI, která určuje míru závilosti slova na kategorii, a která už byla úspěšnější. S touto metodou jsem dosáhl výsledků Pearsonovy korelace 0,51 a Spearmanovy korelace 0,58 pro dataset *RG-65*. Při výpočtu byly paměťové nároky na metodu poměrně vysoké z důvodu vyšší hustoty matice. Aplikoval jsem tedy rozklad matice pomocí SVD pro redukci dimenze, čímž se nejen zmenšily nároky na paměť, ale zároveň se zlepšily výsledky pro všechny datasety. Tyto výsledky považuji za hlavní přínos práce, jelikož se jedná o zcela novou metodu. Této metodě jsem přiřadil název *Latent Semantic Categories* (LSC).

Po otestování výsledků na anglickém jazyce jsem zkusil ověřit funkčnost i na češtině. Metody fungovaly na českém jazyce velmi dobře, některé metody byly dokonce lepší, než pro anglický jazyk. Stejně jako na angličtině i na češtině byly provedeny všechny experimenty.

Všechny metody je jistě možné ještě vylepšit, ať už lepším předzpracováním dat nebo změnou výpočtu spojitosti mezi slovem a kontextem. Dále by se práce dala rozšířit o porovnávání celých textů, místo jednotlivých slov. Implementaci lze také použít i na jiné jazyky kromě angličtiny a češtiny, v budoucnu by tedy bylo dobré provést tyto experimenty znovu i na dalších jazycích.

# Literatura


[1] Brychcín, T.: *Distributional Semantics in Language Modeling*. PhD thesis, University of West Bohemia, 2015.

[2] Brychcín, T.; Konopík, M.: HPS: High precision stemmer. *Information Processing & Management*, rok 51, 1, 2015: s. 68 – 91, ISSN 0306-4573, doi:http://dx.doi.org/10.1016/j.ipm.2014.08.006.
URL `http://www.sciencedirect.com/science/article/pii/S0306457314000843`

[3] Čermák, F.: *Jazyk a jazykověda*. Praha: Karolinum, 2009, ISBN 978-80-246-0154-0.

[4] Černý, J.: *Úvod do studia jazyka*. Rubico, 1998.
URL `https://books.google.cz/books?id=d7LlAAAAMAAJ`

[5] Church, K. W.; Hanks, P.: Word Association Norms, Mutual Information, and Lexicography. *Comput. Linguist.*, rok 16, 1, Bezen 1990: s. 22–29, ISSN 0891-2017.
URL `http://dl.acm.org/citation.cfm?id=89086.89095`

[6] Cinková, S.: WordSim353 for Czech. In *Text, Speech and Dialogue, Proceedings of the 19th International Conference TSD 2016*, Lecture Notes in Artificial Intelligence, Berlin-Heidelberg, Germany: Springer, 2016.

[7] Finkelstein, L.; Gabrilovich, E.; Matias, Y.; aj.: Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, New York, NY, USA: ACM, 2001, ISBN 1-58113-348-0, s. 406–414, doi:10.1145/371920.372094.

[8] Firth, J. R.: A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, 1957: s. 1–32.

[9] Gabrilovich, E.; Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, s. 1606–1611.

[10] Gabrilovich, E.; Markovitch, S.: Wikipedia-based Semantic Interpretation for Natural Language Processing. *J. Artif. Int. Res.*, rok 34, 1, Bezen 2009: s. 443–498, ISSN 1076-9757.

[11] Hill, F.; Reichart, R.; Korhonen, A.: Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2015.

[12] Hirst, G.; St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: An electronic lexical database.*, editace Christiane Fellbaum, Cambridge, MA: MIT Press, 1998, ISBN 978-0262061971, s. 305–332.

[13] Krčmář, L.; Konopík, M.; Ježek, K.: Exploration of Semantic Spaces Obtained from Czech Corpora. In *Proceedings of the Dateso 2011: Annual International Workshop on DAtabases, TExts, Specifications and Objects, Pisek, Czech Republic, April 20, 2011*, 2011, s. 97–107.
URL `http://ceur-ws.org/Vol-706/paper24.pdf`

[14] Landauer, T. K.; Foltz, P. W.; Laham, D.: An Introduction to Latent Semantic Analysis. *Discourse Processes*, rok 25, 1998: s. 259–284.
URL `http://lsa.colorado.edu/papers/dp1.LSAintro.pdf`

[15] Lapesa, G.; Evert, S.; im Walde, S. S.: Contrasting Syntagmatic and Paradigmatic Relations: Insights from Distributional Semantic Models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics, *SEM@COLING 2014, August 23-24, 2014, Dublin, Ireland.*, 2014, s. 160–170.

[16] Lastra-Díaz, J. J.; García-Serrano, A.: A New Family of Information Content Models with an Experimental Survey on WordNet. *Know.-Based Syst.*, rok 89, C, Listopad 2015: s. 509–526, ISSN 0950-7051, doi:10.1016/j.knosys.2015.08.019.
URL `https://doi.org/10.1016/j.knosys.2015.08.019`

[17] Lin, D.: An Information-Theoretic Definition of Similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann, 1998, s. 296–304.

[18] Manning, C. D.; Raghavan, P.; Schütze, H.: *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008, ISBN 0521865719, 9780521865715.

[19] Manning, C. D.; Surdeanu, M.; Bauer, J.; aj.: The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, s. 55–60.
URL `http://www.aclweb.org/anthology/P/P14/P14-5010`

[20] Mikolov, T.; Chen, K.; Corrado, G.; aj.: Efficient Estimation of Word Representations in Vector Space. *CoRR*, rok abs/1301.3781, 2013.

[21] Pennington, J.; Socher, R.; Manning, C. D.: GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, s. 1532–1543.
URL http://www.aclweb.org/anthology/D14-1162

[22] Rubenstein, H.; Goodenough, J. B.: Contextual correlates of synonymy. *Communications of the ACM*, rok 8, 10, jen 1965: s. 627–633, ISSN 0001-0782.

[23] Wall, M. E.; Rechtsteiner, A.; Rocha, L. M.: *Singular Value Decomposition and Principal Component Analysis*. Boston, MA: Springer US, 2003, ISBN 978-0-306-47815-4, s. 91–109, doi:10.1007/0-306-47815-3_5.
URL http://dx.doi.org/10.1007/0-306-47815-3_5