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

Diplomová práce

Vícejazyčné metody pro hledání sémantické podobnosti slov

Plzeň, 2018 Josef Strolený

Prohlášení

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

V Plzni dne 16. května 2018

Josef Strolený

Abstract

This thesis deals with multilingual semantic word similarity. At first, we present approaches to modeling multilingual semantic word similarity and we introduce several selected multilingual methods. In the next section, we deal with the creation of our own multilingual system. We test several existing multilingual methods that we combine to achieve better results. In the last part of the thesis, we compare the created system with state-of-the-art system, which won the SemEval-2017 international scientific competition. Our system achieved more than 6 percent better results on tested data.

Abstrakt

Tato práce se zabývá vícejazyčnou sémantickou podobností slov. Nejprve představujeme přístupy k modelování vícejazyčné sémantické podobnosti slov a dále uvádíme několik vybraných vícejazyčných metod. V další části práce se zabýváme vytvořením vlastního vícejazyčného systému. Nejdříve otestujeme několik existujících vícejazyčných metod, které v další části zkombinujeme k dosažení lepších výsledků. V poslední části práce pak porovnáváme vytvořený systém s nejlepším systémem, který zvítězil v mezinárodní vědecké soutěži SemEval-2017. Náš systém přitom na použitých testovacích datech dosáhl o více než 6 procent lepších výsledků.

Poděkování

Děkuji Ing. Miloslavu Konopíkovi, Ph.D. za vedení mé diplomové práce, za cenné rady a za čas, který mi věnoval.

Tato práce vznikla za podpory projektů CERIT Scientific Cloud (LM2015085) a CESNET (LM2015042) financovaných z programu MŠMT Projekty velkých infrastruktur pro VaVaI.

Obsah

1 Úvod

Schopnost určit sémantickou podobnosti slov nalézá své uplatnění v mnoha oblastech zpracování přirozeného jazyka, jako je například získávání informací, sémantická desambiguace, strojový překlad nebo sumarizace textu.

Měření sémantické podobnosti může být například použito k rozšíření schopností vyhledávacího systému. Pokud uživatel zadá do vyhledávače slovo „pes“ a v prohledávaných datech je přitom slovo „psisko“, dokáže systém naměřit mezi slovy vysokou sémantickou podobnost a může tato data nabídnout uživateli jako výsledek.

Vícejazyčná sémantická podobnost pak dokáže určit sémantickou podobnost i mezi slovy z různých jazyků. Takovýto systém pak k hledanému slovu „pes“ dokáže nalézt i dokumenty obsahující anglická slova „dog“ nebo „hound“.

1.1 Obsah práce

V této práci uvedeme přístupy k modelování vícejazyčné sémantické podobnosti slov a následně představíme vybrané vícejazyčné metody a modely. Druhou část práce pak věnujeme implementaci vlastního systému pro určování vícejazyčné sémantické podobnosti slov, jeho výsledky změříme a porovnáme je se state-of-the-art systémem.

V rámci tvorby tohoto systému se zaměříme na pět jazyků, kterými jsou angličtina, němčina, španělština, čeština a čínština. Vytvořený systém bude schopen určit sémantickou podobnost mezi dvojicí slov z těchto jazyků a zároveň bude schopen pro zadané slovo najít jemu sémanticky nejpodobnější slova.

2 Modelování sémantické podobnosti slov

Přístupy k modelování sémantické podobnosti lze shrnout do tří skupin.

První skupina používá k reprezentaci slov vektory reálných čísel, druhá skupina reprezentuje slova a vazby mezi slovy sémantickou sítí a třetí skupina shlukuje sémanticky podobná slova do slovních tříd.

2.1 Slovní vektory

První přístup je založen na reprezentaci slov pomocí vektorů. Každé slovo je reprezentováno krátkým vektorem reálných čísel, které určuje jeho pozici v sémantickém prostoru.

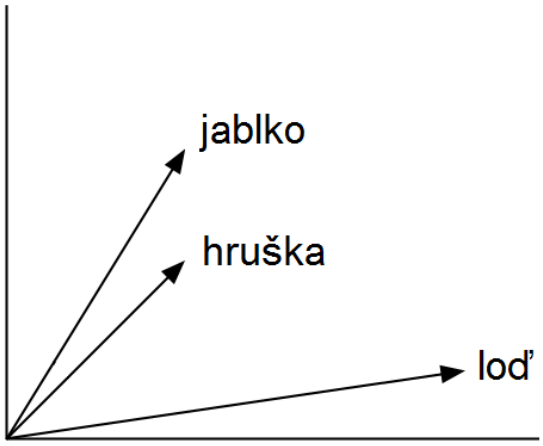
2.1.1 Sémantický prostor

Sémantický prostor je vektorový prostor, který umožňuje zachytit sémantické vazby mezi slovy. Slova se stejným významem budou v sémantickém prostoru ležet blízko u sebe, jak je vidět na obrázku 2.1. Tento způsob eliminuje některé nedostatky pravidlově založených přístupů, především pak závislost na ručně vytvářených pravidlech [37].

Starší metody, jako například *Hyperspace Analogue to Language* [24], vytvářely sémantický prostor o velkých dimenzích, novější metody používají menší dimenzi. Sémantické prostory vycházejí z myšlenky *distribuční sémantiky.*

2.1.2 Distribuční sémantika

Distribuční sémantika říká, že je možné určit význam slova podle toho, jak je distribuováno v textu. Jejím základem je *distribuční hypotéza* a *bag-of-word hypotéza.*



Obrázek 2.1: V sémantickém prostoru jsou slova reprezentována slovními vektory. Významově podobná slova se nachází blízko u sebe.

Distribuční hypotéza

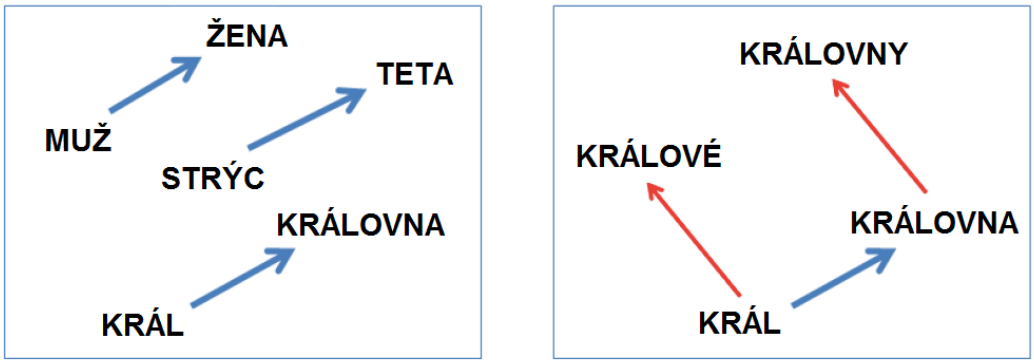
Distribuční hypotéza využívá pro určení významu slova okolního textu, přičemž slova s podobným významem se budou vyskytovat v podobném kontextu a slova s podobným kontextem budou mít podobný význam [36]. Tato hypotéza používá *lokální kontext*, tj. bezprostřední okolí slova.

Bag-of-word hypotéza

Bag-of-word hypotéza pracuje s *globálním kontextem*, tj. využívá namísto okolí slova celý dokument. Hypotéza říká, že slova budou mít podobný význam, pokud jsou podobně distribuována v dokumentech.

2.1.3 Slovní analogie

Vlastnosti slovních vektorů lze ověřit pomocí slovních analogií. Opět je využita skutečnost, že podobné vektory mají podobný tvar a zachycují sémantické vztahy mezi slovy. Například vztah mezi vektory muž-žena je podobný jako vztah mezi vektory strýc-teta nebo král-královna, čehož lze dále využít. To je znázorněno na obrázku 2.2.



Obrázek 2.2: Vektorová reprezentace slov dokáže zachovat vazby mezi slovy.

Pokud například od vektoru král odečteme vektor muž a přičteme vektor žena, vzniklý vektor bude odpovídat významu slova královna. Podobným způsobem se můžeme dotazovat na hlavní města států. Pokud od vektoru Francie odečteme vektor Paříž a přičteme vektor Řím, získáme vektor odpovídající Itálii.

2.1.4 Podobnost vektorů

Pro porovnání vektorů je důležitější orientace vektoru, spíše než jeho délka. Proto se pro porovnání dvojice vektorů používá nejčastěji *kosinová vzdálenost*. Ta definuje podobnost dvojice vektorů jako kosinus úhlu mezi vektory, jak je uvedeno ve vzorci 2.1.

$$sim(A, B) = cos(A, B) = \frac{\sum_{i=1}^{n}(A_i B_i)}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

2.1.5 Početně založené modely

Tyto modely používají k vytvoření sémantických vektorů *kookurenční matici* slov, vytvořenou podle *jazykového korpusu*. Tato matice zaznamenává jaká slova a v jakém počtu se nacházejí ve vymezeném kontextu slova. Početně založené metody pak s touto maticí dále pracují k vytvoření sémantického prostoru.

GloVe

Pennington et al. navrhli model využívající kookurenční matici slov, ze které pomocí *regrese* vytváří slovní vektory[30]. Vzniklý sémantický prostor je obvykle dimenze 50 až 300 a zachovává sémantické vazby mezi jednotlivými slovy.

2.1.6 Prediktivní modely

Prediktivní modely nepoužívají kookurenční matici, namísto toho pracují přímo s jazykovým korpusem a jsou trénovány k *předvídání* okolních slov.

Word2Vec

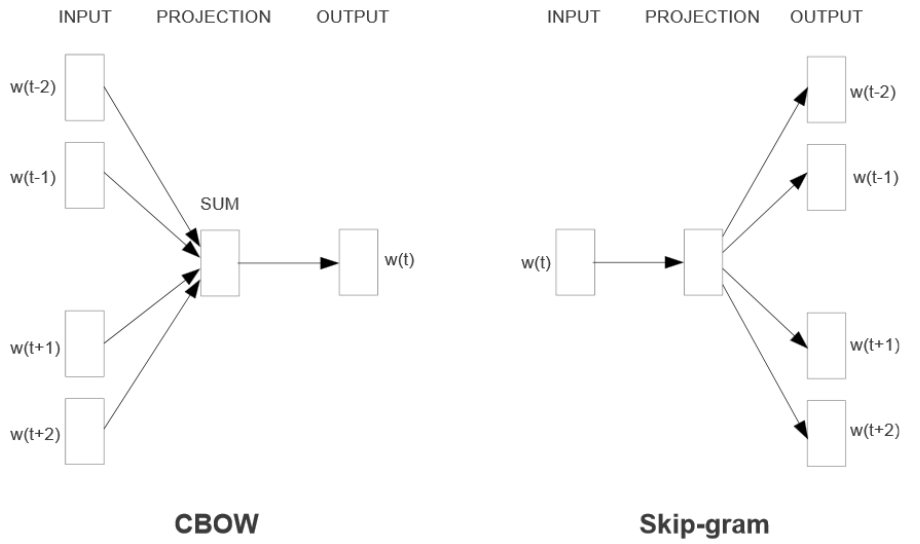
Mikolov et al. přišli s dvojici prediktivních modelů[26]. Oba modely používají *neuronovou síť* s jednou skrytou vrstvou.

Vstupní i výstupní vrstva obou modelů má dimenzi rovnou velikosti slovníku a pro reprezentaci slov se používá tzv. *one-hot kódování*, kde vektor zavedený na vstup má jeden element rovný jedné, a to na pozici, jež odpovídá indexu slova ve slovníku, zbylé prvky vektoru jsou nulové. Skrytá vrstva pak má dimenzi rovnou požadovanému rozměru vytvářeného vektorového prostoru.

Prvním z modelů je *Continuous Bag-of-Words* (*CBOW*), který používá několik kontextových slov k predikci jednoho slova cílového. Druhým modelem je *skip-gram*, který naopak používá jedno dané slovo k predikci slov v jeho kontextu. To je ilustrováno na obrázku 2.3.

Metoda skipgram funguje dobře i při trénování na malém jazykovém korpusu a dokáže dobře reprezentovat málo se vyskytující slova či fráze. Metoda CBOW se trénuje násobně rychleji a dosahuje mírně lepších výsledků na často se vyskytujících slovech a frázích.

Slovní vektory se po natrénování nacházejí ve váhových maticích. Tyto vektory mají, podobně jako vektory *GloVe*, obvykle dimenzi 50 až 300 a zaznamenávají sémantickou reprezentaci slova.



Obrázek 2.3: Ukázka modelu Contiuous Bag-of-Words a Skip-gram. Převzato z [26].

fastText

Bojanovski et al. doplnili původní modely Word2vec o několik funkcí, díky kterým vylepšili jejich výsledky [16]. Během procesu trénování jsou slova navíc rozdělována na písmenné n-gramy („podslova“) určité délky a tyto se pak také podílejí na trénování. Tento způsob umožňuje zachytit například skutečnost, že několik slov má stejný kořen, a proto mezi nimi existuje sémantická podobnost.

2.2 Sémantické sítě

Dalším způsobem reprezentace sémantické podobnosti slov jsou sémantické sítě.

2.2.1 Sémantická síť

Sémantická síť je *orientovaný graf*, který je tvořen vrcholy reprezentující objekty popisovaného světa a hranami, které vyjadřují vazby mezi objekty.

V našem případě reprezentují vrcholy slova či *synsety* a hrany vyjadřují sémantické vazby mezi nimi. Těchto hran v síti obvykle existuje více druhů, čímž lze zachytit různé druhy sémantických vazeb.

Tyto rozsáhlé sítě jsou obvykle automatizovaně vytvářeny a aktualizovány z dostupných internetových databází a ontologií, jako je *Wikipedia* nebo *DBPedia*.

2.2.2 Synset

Některé sítě používají namísto slov *synsety* (*synonym set*). Ty vyjadřují jeden konkrétní význam slova. Synset se proto může sestávat z více slov a naopak jedno slovo může být přiřazeno více synsetům. Sítě pak mohou namísto slov použít synsety jako vrcholy grafu [38].

2.2.3 Jednojazyčné sítě

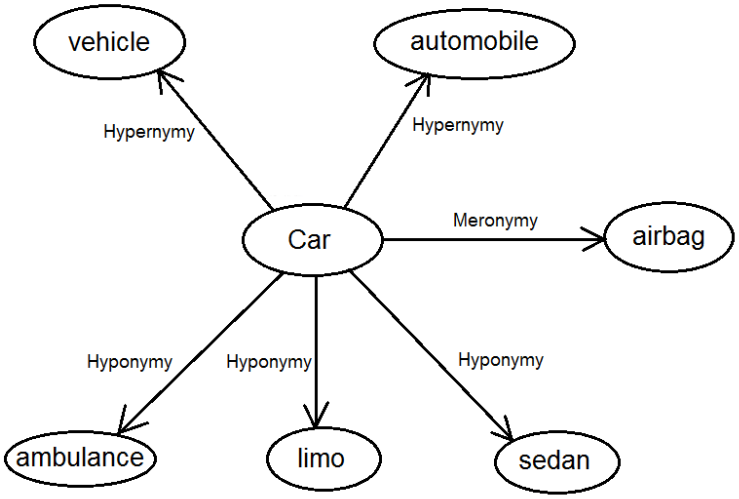
Existuje množství sítí. které zachycují pouze jeden jazyk, často angličtinu. Typickým představitelem této skupiny je síť *WordNet*

WordNet

WordNet je rozsáhlá lexikální databáze anglického jazyka, vyvinutá na Princetonské univerzitě. Jejím hlavním úkolem je napomáhat analýze textu a aplikacím spojeným s umělou inteligencí [38].

Obsahuje více než 117 tisíc synsetů, propojených šesti druhy vazeb. Síť navíc obsahuje definice synsetů a ukázku jejich použití. Část sítě je zachycena na obrázku 2.4.

WordNet byl inspirací pro vznik dalších, často neanglických, sítí, jako je *Czech WordNet* (*Český WordNet*). Také z něj vychází některé sítě vícejazyčné.



Obrázek 2.4: Sémantická síť WordNet.

2.3 Shlukování

Posledním přístupem je shlukování slov do slovních tříd.

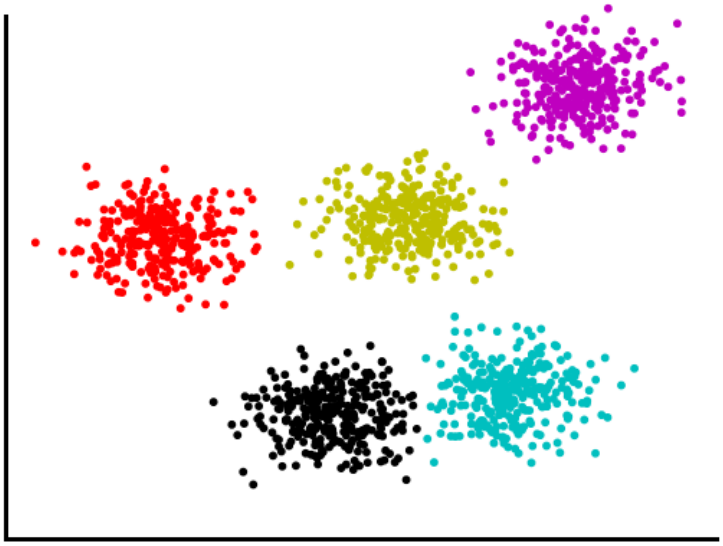
2.3.1 Shlukování

Shlukování je proces, během něhož jsou objekty tříděny do tříd podle své podobnosti. Objekt bude podobnější ostatním objektům v téže třídě, než ve třídě jiné. To je ilustrováno na obrázku 2.5.

Metody sémantické podobnosti slov založené na shlukování vycházejí, stejně jako metody slovních vektorů, z distribuční sémantiky a slova jsou řazena do tříd podle lokálního či globálního kontextu. Výstupem metod jsou slovní třídy.

2.3.2 Slovní třídy

Slovní třída je množina slov, které si jsou sémanticky podobná. Slovo je sémanticky podobnější ostatním slovům ze stejné slovní třídy, než z třídy jiné.



Obrázek 2.5: Cílem shlukování je seskupit objekty do tříd. Objekty stejné barvy patří do stejné třídy.

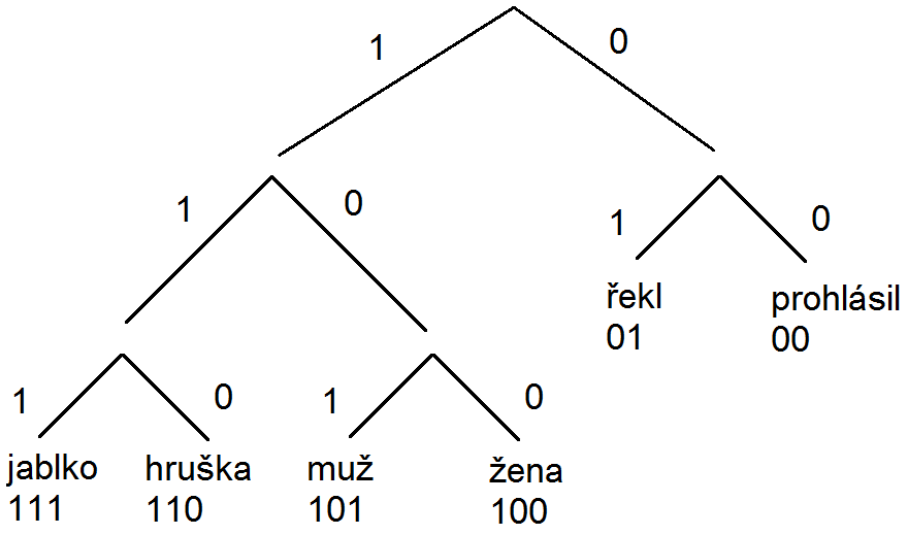
Některé metody dokáží slovní třídy hierarchicky seskupovat do stromů.

2.3.3 Brownovo shlukování

Brownovo shlukování (také *IBM shlukování*) je jednojazyčná metoda hierarchického shlukování slov dle lokálního kontextu[17]. Je základem některých dalších metod.

Metoda prochází jazykový korpus a slučuje podobná slova do hierarchických slovních tříd. Tím ze slov utváří binární strom, přičemž slova lze zapsat dlouhými binárními vektory. Výstup Brownova shlukování slov je znázorněn na obrázku 2.6.

Mnoho dalších metod založených na shlukování, včetně metod vícejazyčných, vychází právě z Brownova shlukování.



Obrázek 2.6: Princip Brownova shlukování. Slova jsou podle sémantické podobnosti hierarchicky řazena do slovních tříd.

3 Vícejazyčné metody sémantické podobnosti slov

Níže představíme způsoby, jak lze základní jednojazyčné přístupy rozšířit na vícejazyčné a uvedeme vybrané vícejazyčné metody sémantické podobnosti slov.

3.1 Vícejazyčné slovní vektory

Dosud uvedené modely slovních vektorů, které pracují s jednojazyčnými korpusy, lze rozšířit, aby fungovaly i na korpusech vícejazyčných. Tyto vícejazyčné modely pak lze dělit dvěma způsoby[31].

3.1.1 Rozdělení podle způsobu učení

Prvním způsobem je dělit modely podle toho, jakým způsobem se učí vícejazyčným vektorům.

Jednojazyčné mapování Tyto modely nejprve trénují nezávislé slovní vektory na velkých jednojazyčných korpusech, pro každý jazyk zvlášť. Ve druhé fázi pak nalézají mapování z jednoho systému do druhého. K tomu většinou využívají překlady ze slovníku, případně data ze *slovně zarovnaného* paralelního korpusu.

Pseudo-vícejazyčné Tyto modely vytvářejí vícejazyčný datový korpus z několika korpusů jednojazyčných. Většina přístupů identifikuje slova, která lze přeložit z jednoho jazyka do druhého a ve všech korpusech je nahradí zástupnými symboly. Tím je zajištěno, že tato slova budou mít stejnou vektorovou reprezentaci napříč jazyky.

Vícejazyčné Tyto modely se soustředí výhradně na trénování na paralelním korpusu. K tomu většinou využívají *větně zarovnaná* data, namísto použití slovníkových překladů.

Spojitě optimalizované Tyto modely se soustředí jak na optimalizaci vztahů v jednojazyčných korpusech, tak i na vazby vícejazyčné.

3.1.2 Rozdělení podle druhu paralelního korpusu

Druhý způsob dělení je podle druhu použitého paralelního korpusu, ze kterého se reprezentace získává.

Slovně zarovnaná data Modely používají paralelní korpus, který je zarovnán na slova. Tento korpus je obvykle získán automatizovaně z větně zarovnaných dat.

Větně zarovnaná data Modely používají korpus, který je zarovnaný na věty.

Překlady dokumentů Tyto modely používají jako paralelní korpus překlady dokumentů.

Slovníky Paralelní korpus je slovník, obsahující překlady slov.

Bez paralelních dat Nejsou dostupná žádná paralelní data. Učení vícejazyčných reprezentací probíhá pouze z jednojazyčných zdrojů.

3.2 Vícejazyčné metody slovních vektorů

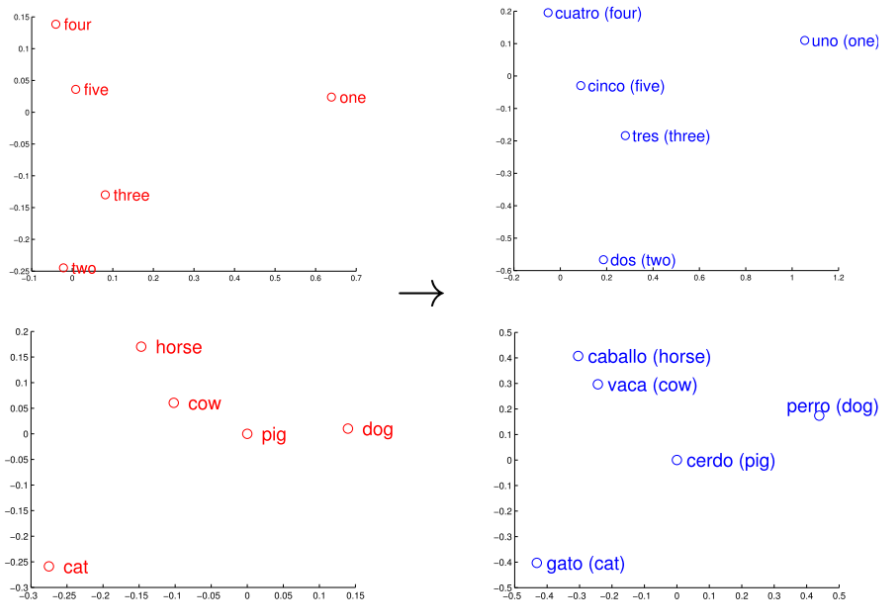
3.2.1 Lineární projekce

Vektorový prostor umožňuje zachytit i vztahy mezi jednotlivými slovy a tyto vztahy si jsou v různých jazycích velice podobné[27].

Z toho vychází myšlenka, že je možné natrénovat dvojici nezávislých vektorových prostorů na jednojazyčných datových korpusech a následně vytvořit *transformační matici* W, která dokáže transformovat jeden vektorový prostor do prostoru druhého.

Na obrázku 3.1 je promítnuta část vektorových prostorů získaných z anglických a španělských jednojazyčných korpusů. Na snímcích je vidět podobnost vztahů mezi slovy v různých jazycích.

Metoda lineární projekce využívá dvojjazyčné slovníky k překladu části slov z jednoho sémantického prostoru do jazyka druhého prostoru. Pomocí těchto slov pak nalézá transformační matici W, která minimalizuje eukleidovskou vzdálenost mezi reprezentací x_i slova w_i, transformovanou maticí W, a reprezentací překladu slova z_i ze slovníku. To je vyjádřeno vzorcem 3.1.



Obrázek 3.1: Vektorová reprezentace slov je v různých jazycích velice podobná. Pomocí transformační matice W se lze transformovat z jednoho prostoru do druhého. Převzato z [27].

$$\min_{W} \sum_{i=1}^{n} ||Wx_i - z_i||^2 \tag{3.1}$$

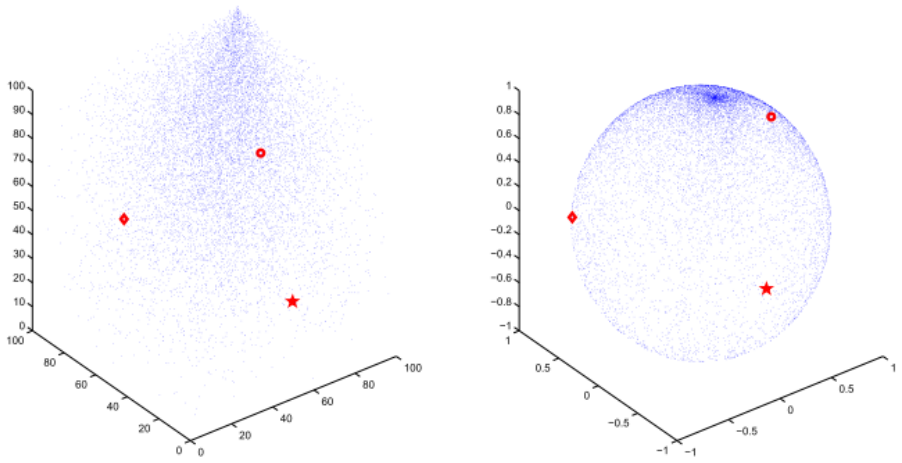
Tato metoda tak spadá do kategorie jednojazyčného mapování s využitím slovníku jako vícejazyčného korpusu.

3.2.2 Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation

Na myšlenku lineární projekce navázali Xing et al. Ti si povšimli, že zatímco pro určení podobnosti mezi vektory se používá kosinová vzdálenost, lineární projekce optimalizuje vzdálenost eukleidovskou. Použití různých metrik pak může degradovat výsledky[39].

K vyřešení tohoto problému proto navrhli během trénování slovní vektory normalizovat na jednotkovou vzdálenost. Tím dojde k tomu, že vytvořené

vektory budou ležet v prostoru hyperkoule, jak je znázorněno na obrázku 3.2.



Obrázek 3.2: Po normalizace se slovní vektory nacházejí v prostoru hyperkoule, převzato z [39].

Jako druhý krok pak pozměnili cenovou funkci mapování do tvaru 3.2. Nakonec také stanovili, že vytvořená transformační matice W musí být ortogonální, čímž je zajištěno, že jednotkovou vzdálenost budou mít i promítnuté vektory Wx_i.

$$\max_{W} \sum_{i} (Wx_i)^T z_i \tag{3.2}$$

I tato metoda tedy patří do kategorie lineárního mapování s použitím slovníku jako vícejazyčného korpusu.

3.2.3 Word Translation Without Parallel Data

Myšlenku lineární projekce a ortogonálních vektorů použili také Conneau et al. Ti navrhli metodu, která dokáže vytvořit transformační matici W i bez použití dodatečných paralelních dat a slovníků [19].

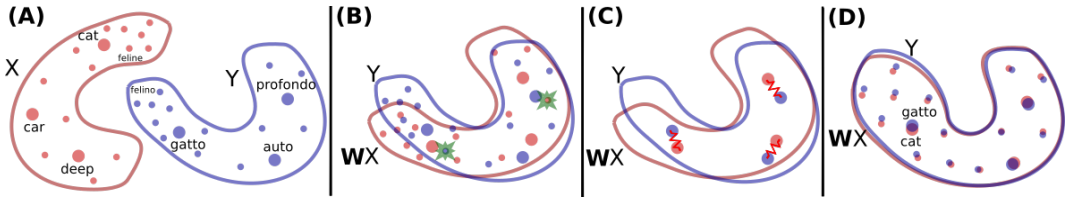
Vycházejí přitom z poznatků, že lineární projekce dosahuje lepších výsledků, pokud je transformační matice W ortogonální. Tím se jim podařilo

upravit původní vzorec do nového tvaru 3.3.

$$W^* = \underset{W \in O_d(\mathbb{R})}{argmin} ||WX - Y||_F = UV^T$$
$$U\Sigma V^T = SVD(YX^T) \tag{3.3}$$

Tento tvar pak vede na Prokrustovu analýzu [1]. Jedná se o matematickou metodu, která porovnává dvojici vícerozměrných prostorů a za pomoci škálování velikostí, transpozice a rotace se snaží minimalizovat sumu čtverců vzdáleností.

Metoda pak pro transformaci prostorů používá iterativní algoritmus Prokrustovy analýzy. To je znázorněno na obrázku 3.3.



Obrázek 3.3: Word translation Without Parallel Data používá pro transformaci prostorů Prokrustovu analýzu. Převzato z [19].

Metoda patří do kategorie lineárního mapování, a to buď to zcela bez použití paralelních dat, nebo s použitím slovníku jako vícejazyčného korpusu.

3.2.4 Bilingual Correlation Based Embeddings (BiCCA)

Na myšlenku lineární transformace navázali také Faruqui et al. Namísto lineárního mapování ale použili matematickou metodu zvanou *kanonická korelační analýza*[22].

Tato vícerozměrná metoda používá vektorový prostor Σ', který je vzorkem sémantického prostoru Σ jednoho z jazyků a jemu odpovídající prostor Ω' z prostoru Ω druhého jazyka. Prostor Ω' je nalezen pomocí vícejazyčných slovníků a jeho prvky odpovídají překladům slov z prostoru Σ'. Pomocí těchto

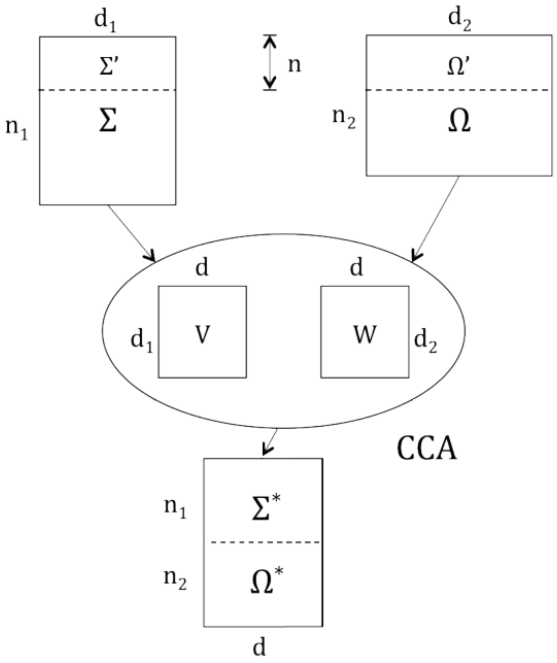
[1] Pojmenováno podle lupiče z řecké mytologie, který své oběti buď natahoval, nebo jim odsekával končetiny, aby se vešli na jeho lože.

vzorků z dvojice jazykových prostorů pak nalezne dvojici transformačních matic V a W. To lze zapsat vzorcem 3.4.

$$V, W = CCA(\Omega', \Sigma') \tag{3.4}$$

S těmito získanými maticemi V a W pak lze oba vektorové prostory Σ a Ω transformovat do jednoho společného prostoru. To je zapsáno vzorcem 3.5 a znázorněno na obrázku 3.4. Σ^* i Ω^* zde reprezentují jeden společný sémantický prostor.

$$\begin{aligned} \Sigma^* &= \Sigma \times V \\ \Omega^* &= \Omega \times W \end{aligned} \tag{3.5}$$



Obrázek 3.4: Pro transformaci do společného prostoru je použita dvojice matic nalezených kanonickou korelační analýzou. Převzato z [22].

I tato metoda spadá do kategorie jednojazyčného mapování s použitím slovníku jako paralelního korpusu.

3.2.5 MultilingualCCA

Ammar et al. pak na myšlenku použití kanonické korelační analýzy navázali a rozšířili její použití na libovolný počet prostorů [14].

Podobně jako v předchozím případě použijí vzorky sémantických prostorů Σ a Ω k získání transformačních matic V a W. V dalším kroku vypočtou inverzi matice W a tu použijí k transformaci prvního sémantického prostoru ze společného prostoru do prostoru druhého jazyka, jak je vyjádřeno ve vzorci 3.6. Tímto způsobem lze transformovat libovolný počet sémantických prostorů do prostoru vybraného jazyka.

$$\Omega = \Sigma \times V \times W^{-1} \tag{3.6}$$

3.2.6 Random translation Replacement

Gouws et al. přišli s metodou *Random Translation Replacement* (*RTR*), která vytváří pseudo-vícejazyčný jazykový korpus z korpusů jednojazyčných [23].

Metoda používá jednojazyčné jazykové korpusy a vícejazyčný slovník. V korpusech je pak část slov nahrazena za své překlady ze slovníku a všechny korpusy jsou propojeny. Tím je vytvořen pseudo-vícejazyčný jazykový korpus, na kterém lze natrénovat slovní vektory.

Guows et al. použili jako jazykové korpusy články z Wikipedie a překlady slov získali použitím *Google Translate*. Na tomto korpuse pak natrénovali *Continuous Bag Of Words*.

Tato metoda patří do kategorie pseudo-vícejazyčných metod s použitím slovníku jako vícejazyčného korpusu.

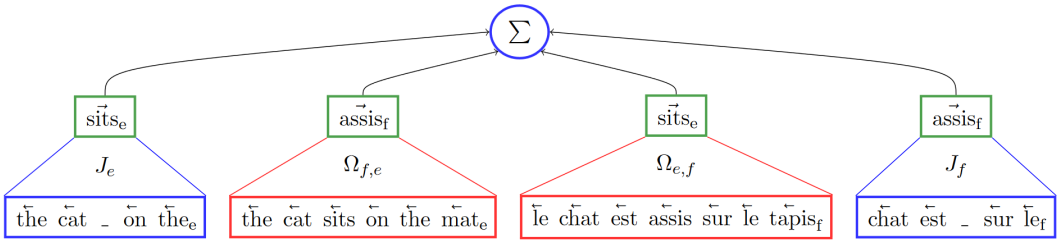
3.2.7 Bilingual Skipgram without Word Alignments

Coulmance et al. navrhli metodu *Bilingual Skipgram without Word Alignments* (*BSwWA*), která používá větně zarovnaný korpus k vytvoření vícejazyčných slovních vektorů[20]. Předpokládají, že každé slovo ve zdrojové větě je kontextem každého slova v cílové větě.

Pro tyto potřeby definovali optimalizační kritérium skip-gramu jako 3.7, kde s jsou věty, C je větně zarovnaný korpus, w jsou slova věty a c je kontext a $-log\sigma()$ je cenová funkce skip-gramu.

$$\Omega_{e,f} = \sum_{(s_{i_1}, s_{i_2}) \in C_{i_1,i_2}} \sum_{w_{i_1} \in s_{i_1}} \sum_{c_{i_2} \in s_{i_2}} -log\sigma(w_{i_1}, c_{i_2}) \tag{3.7}$$

Jelikož je toto kritérium asymetrické, používají jedno kritérium k trénování zdrojové věty na cílovou a druhé k trénování cílové věty na zdrojovou. Navíc trénují každé slovo věty na všechna ostatní slova ve větě, jak je znázorněno na obrázku 3.5. Existuje i další varianta metody, která namísto skip-gramu používá CBOW.



Obrázek 3.5: Metoda optimalizuje vazby mezi jazyky i uvnitř jazyka, převzato z [20].

Metoda patří do kategorie spojitě optimalizovaných metod a používá větně zarovnaný korpus.

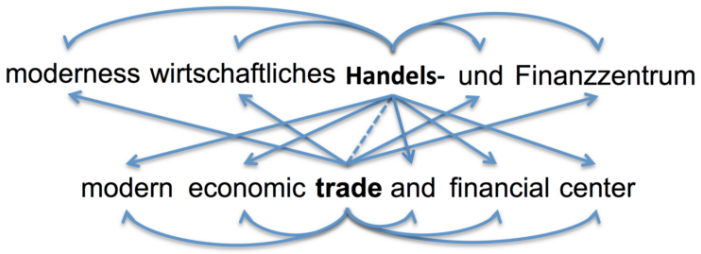
3.2.8 Bilingual Skip-Gram (BiSkip)

Luong et al. rozšířili klasický skip-gramový model, aby jej bylo možné použít na vícejazyčných datech [25].

S použitím slovně zarovnaného paralelního korpusu trénují systém k predikci nejen okolních slov zdrojového jazyka, ale i jazyka cílového, jak je zachyceno na obrázku 3.6.

Autoři navrhli i druhou verzi algoritmu, která používá větně zarovnaný korpus. Předpokládají, že slova ve zdrojové i cílové větě jsou monotónně zarovnána, přičemž každé slovo zdrojové věty na pozici i je zarovnáno na

slovo cílové věty na pozici $i\frac{T}{S}$, kde S je délka zdrojové věty a T je délka věty cílové.



Obrázek 3.6: Dvojjazyčný skip-gram trénovaný k predikci okolních slov zdrojového i cílového jazyka. Převzato z [25].

Tato metoda spadá do kategorie spojitě optimalizovaných metod se slovně, nebo větně zarovnaným paralelním korpusem.

3.3 Vícejazyčné sémantické sítě

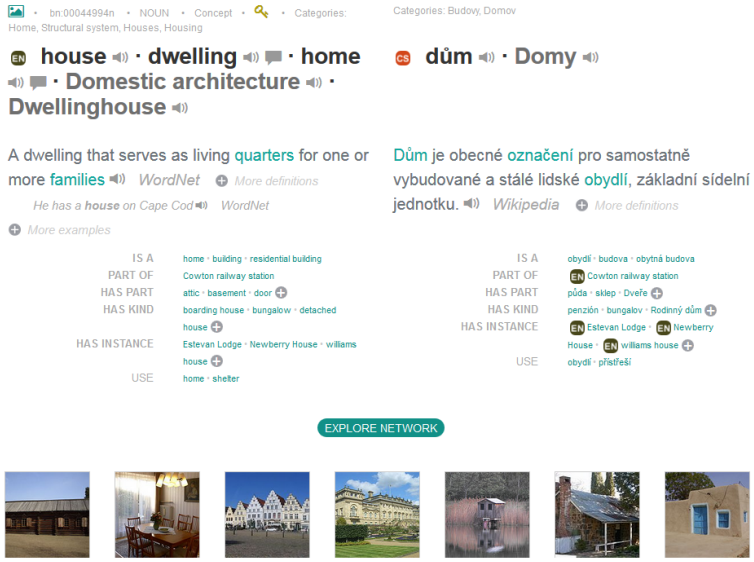
Vícejazyčné sítě rozšiřují původní sítě jednojazyčné a umožňují buďto přiřadit slovům z různých jazyků stejný synset, tj. bez ohledu na to, z jakého jazyka slovo pochází, vždy je mu přiřazen synset vyjadřující konkrétní význam slova napříč různými jazyky, nebo, pokud síť používá namísto synsetů slova, obsahuje síť sémantické vazby napříč slovy z různých jazyků.

BabelNet

Síť *BabelNet* je vícejazyčným encyklopedický slovníkem i sémantickou sítí spojující pojmy a pojmenované entity (tj. jména osob, názvy produktů, časové údaje apod.) do velké sítě se sémantickými vztahy[29]. Síť obsahuje přibližně 15 milionů synsetů, každý synset reprezentuje jeden konkrétní význam napříč 284 jazyky.

BabelNet vznikla sloučením sítě *WordNet* s daty z *Wikipedie*. V současnosti je dostupná verze *BabelNet live*, která se neustále rozrůstá díky automatickým denním aktualizacím z mnoha zdrojů, jako jsou *Wikipedie*, *Wikislovník*, *Wikidata*, *GeoNames*, *ImageNet* a mnoha dalších.

Ukázka části dostupných informací a vztahů k synsetu odpovídajícímu významu *„dům“* je zobrazena na obrázku 3.7.



Obrázek 3.7: Synset sítě BabelNet vyjadřuje jeden konkrétní význam napříč mnoha jazyky. Převzato z [29].

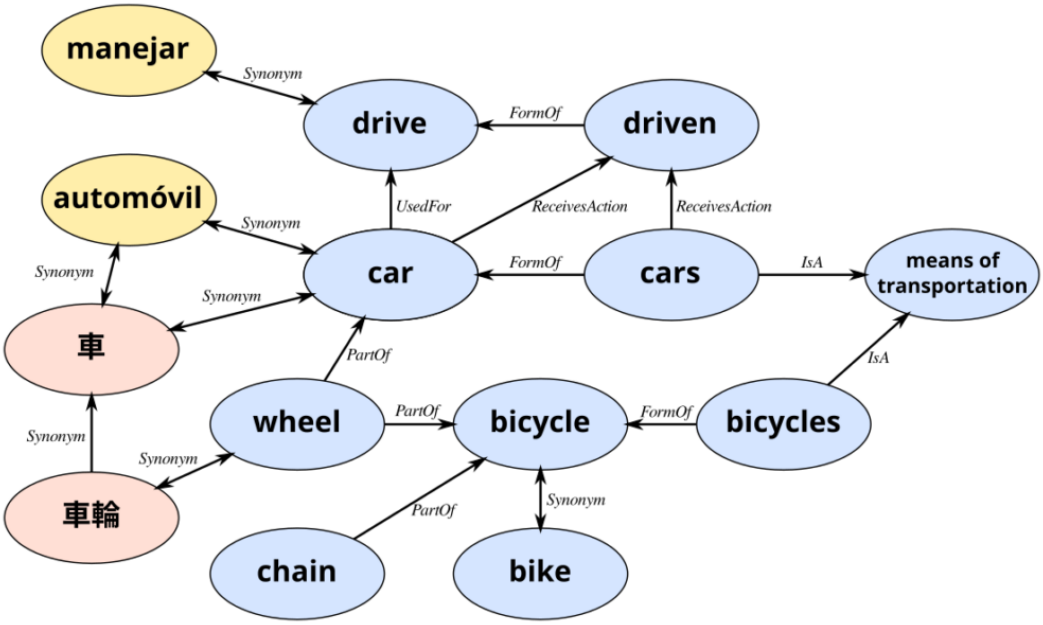
ConceptNet

ConceptNet je volně dostupná sémantická síť, vytvořená k podpoře počítačového zpracování přirozeného jazyka[32]. Její první verze byla spuštěna roku 1999 na *MIT*.

Původně byla síť ručně udržována, v současnosti využívá automaticky sbíraná data ze zdrojů jako je *DBPedia*, *Wikislovník* nebo *Open Multilingual Wordnet*. Obsahuje přibližně 28 milionů slov z celkem 304 jazyků, z nichž 87 má stálou podporu.

Na rozdíl od sítě *BabelNet* nevyužívá synsety, základním objektem jsou slova, u nichž jsou mezi synonymy uvedeny jejich možné překlady.

Na obrázku 3.8 jsou zachyceny různé druhy vztahů mezi slovy, včetně vazeb vícejazyčných.



Obrázek 3.8: Sémantické vztahy v síti ConceptNet. Převzato z [2].

3.4 Vybrané metriky podobnosti v sémantických sítích

Protože je sémantická síť grafem, používají se pro určení sémantické podobnosti slov většinou metriky, které procházejí graf. Níže vybíráme několik metrik.

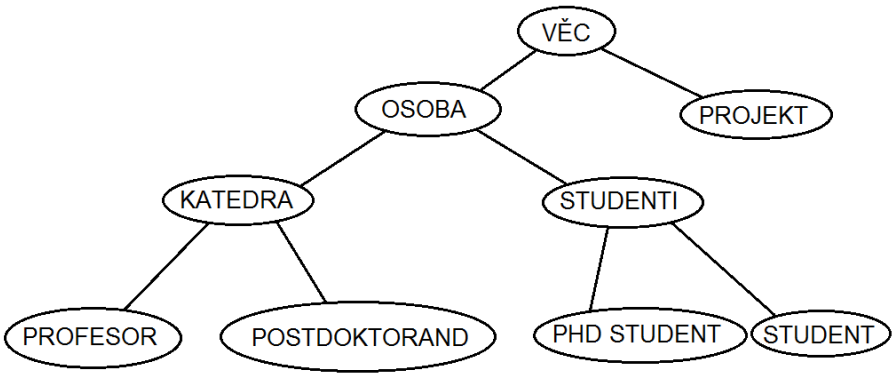
3.4.1 Metrika délky cesty

Základní metrika využívá pouze délku cesty mezi vrcholy.

Myšlenkou je, že sémanticky podobná slova se nacházejí v síti blízko u sebe a s rostoucí vzdáleností se podobnost slov zmenšuje, jak je uvedeno ve vzorci 3.8.

$$sim(A, B) = \frac{1}{dist(A, B)} \tag{3.8}$$

Na příkladu 3.9 si tedy profesor a postdoktorand budou sémanticky bližší, než profesor a student.



Obrázek 3.9: Metriky podobnosti v sémantických sítích používají metody prohledávání grafu.

3.4.2 Wu & Palmerův algoritmus

Wu & Palmerův algoritmus[15] využívá k určení podobnosti vrcholů hloubku obou vrcholů od *kořene* a rovněž hloubku nejbližšího společného předka (LCS) obou vrcholů od kořene, jak je uvedeno ve vzorci 3.9.

Myšlenkou je, že vrcholy, které budou mít blízkého společného předka si jsou podobnější, než vrcholy, u nichž je společný předek příliš obecný.

$$sim(A, B) = \frac{2 \cdot depth(LCS)}{depth(A) + depth(B)} \tag{3.9}$$

V příkladu 3.9 je nejbližším společným předkem profesora a postdoktoranda katedra. Výsledná podobnost, je tedy $^2/_3$. Podobnost profesora a studenta, jejichž nejbližším společným předkem je osoba, pak bude pouze $^1/_3$.

3.4.3 Leacock a Chodorowův algoritmus

Leacock a Chodorovův algoritmus[15] využívá informaci o vzdálenosti mezi vrcholy a informaci o maximální hloubce vrcholu. Tato informace je pak vyhlazena logaritmem, jak je uvedeno ve vzorci 3.10.

Zde je myšlenkou, že konkrétnější vrcholy, které se nacházejí níže v hierarchii, si jsou sémanticky podobnější, než vrcholy obecnější, které jsou v hierarchii postaveny výše.

$$sim(A, B) = -log(\frac{dist(A, B)}{2 \cdot depth}) \tag{3.10}$$

V našem příkladu 3.9 pak bude vzdálenost mezi profesorem a postdoktorandem rovna $-log(\frac{2}{6})$, což je přibližně $0,477$. V případě profesora a studenta je vzdálenost rovna $-log(\frac{4}{6})$, tedy asi $0,176$.

3.4.4 Resnikův algoritmus

Resnikův algoritmus se od předchozích liší, neboť používá *míru informace*. Ta je definována jako 3.11, kde LCS je nejbližší společný předek a $IC(LCS)$ je míra informace. Resnikův algoritmus je pak definován jako 3.12.

Myšlenkou je, že konkrétnější vrcholy budou dosahovat vyšší míry informace a budou proto podobnější než vrcholy s menší mírou informace.

$$IC(LCS) = -log(P(LCS)) \tag{3.11}$$

$$sim(A, B) = IC(LCS(A, B)) \tag{3.12}$$

3.5 Obohacení slovních vektorů

Existují metody, které dokáží již vytvořené slovní vektory dále upravovat, díky čemuž je lze obohatit o nové informace.

3.5.1 Retrofitting

Faruqui et al. navrhli metodu, která upravuje souřadnice vektorů podle nových informací [21]. Tímto způsobem lze slovní vektory obohatit o informace

ze sémantických sítí.

Autoři zde optimalizují eukleidovskou vzdálenost vektorů. Pro získání nových souřadnic vektoru q_i přitom berou v úvahu původní pozici vektoru $\hat{q}_i$ a nové informace získané ze sémantické sítě, kde jsou reprezentovány jako hrany mezi slovy i a j. Váhy α a β pak umožňují nastavit relativní síli asociací. To lze zapsat vzorcem 3.13.

$$\psi(Q) = \sum_{i=1}^{n}[\alpha_i||q_i - \hat{q}_i||^2 + \sum_{(i,j)\in E}\beta_{ij}||q_i - q_j||^2] \tag{3.13}$$

3.5.2 Rozšířený retrofitting

Rozšířený retrofitting (*Expanded retrofitting*) funguje jako nadstavba původního retrofittingu a přidává možnost do sémantického prostoru vkládat nové vektory. Ty utváří podle informací z již existujícího sémantického prostoru a nových informací, pocházejících typicky ze sémantické sítě.

3.6 Vícejazyčné shlukování

Shlukovací metody, včetně vícejazyčných, většinou vycházejí z Brownova shlukování. Vícejazyčných shlukovacích metod ale není mnoho.

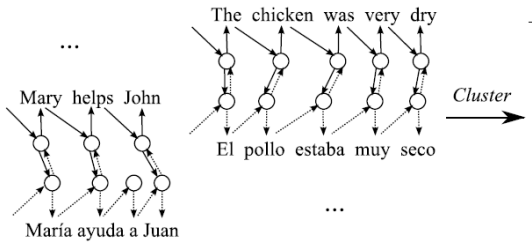
3.6.1 Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure

Täckström et al. navrhli dvojici shlukovacích metod, které rozšiřují jednojazyčné shlukování na shlukování vícejazyčné [34]. Vycházejí přitom z prediktivního modelu založeném na Brownově shlukování.

První navržená metoda vytváří shluky ze slov jednoho zdrojového jazyka a ty následně promítá do jazyka cílového za použití slovně zarovnaných paralelních dat.

Druhá metoda vytváří slovní shluky na zdrojovém i cílovém jazyku zároveň a slovně zarovnaný paralelní korpus je použit k zarovnávání shluků

z obou jazyků. Princip této metody je znázorněn na obrázku 3.10.



Cluster	Lang.	Sample words
60	EN	was, wasn't, was'nt, wasn"t, hasnít, doesn't, ...
60	ES	estaba, estaráen, estubo, fúe, quedaba, ...
101	EN	very, mildly, wholly, terribly, gloriously, ...
101	ES	muchomás, fuerte, fuertes, duro, duros, poco, ...
153	EN	chicken, bird, ostriches, beef, pork, burger, steak, ...
153	ES	pollo, achote, manzana, tortugas, marsupiales, ...
195	EN	The, ...
195	ES	El, La, Los, Las, LoS, ...
236	EN	dry, wet, moist, lifeless, dullish, squarish, limpid, ...
236	ES	seco, secos, semiseco, semisecos, mojado, humedo, ...

Obrázek 3.10: Princip vícejazyčného shlukování. Metoda využívá dvojice jednojazyčných korpusů a slovně zarovnaná data k získání vícejazyčných shluků. Převzato z [34].

4 Analýza úlohy

Naším dalším úkolem je tvorba vlastního systém k určování sémantické podobnosti slov. Systém přitom bude pracovat s angličtinou, němčinou, španělštinou, češtinou a čínštinou.

Svou pozornost jsme se rozhodli věnovat použití slovních vektorů, neboť se jedná o účinné řešení, ke kterému již existuje množství vícejazyčných metod. Zvažovali jsme také použití sémantických sítí, které bychom v další fázi použili k obohacení vytvořených slovních vektorů. Na této možnosti nám ale vadila značná závislost na použité vícejazyčné sémantické síti, která by musela všechny použité jazyky obsahovat, a to navíc v dostatečném počtu slov. Zřejmě bychom se tak nevyhnuli značné redukci slovníku. Použití shlukovacích metod jsme zavrhli úplně, neboť zde neexistuje dostatek vícejazyčných metod.

4.1 SemEval-2017

Užitečné informace nám poskytla mezinárodní vědecká soutěž SemEval-2017[4], která se zabývala vícejazyčnou sémantickou podobností slov, a které se blíže věnujeme v kapitole 5. Prostudování zúčastněných systémů ukázalo, že většina řešitelů použila právě slovní vektory, popřípadě jejich kombinaci se sémantickými sítěmi. Ze soutěže jsme získali cenné povědomí o tom, jakých výsledků dosahují aktuálně používané systémy.

Ze soutěže také budeme moci použít testovací data. Tato data byla vytvořena k testování jednojazyčné i vícejazyčné sémantické podobnosti slov pro angličtinu, němčinu, španělštinu, italštinu a perštinu, tři z těchto jazyků se tedy shodují s našimi. Vítězný systém ze soutěže jsme se pak rozhodli použít pro porovnání výsledků s naším systémem.

4.2 Jazykové korpusy

Dalším nezbytným krokem bude získat dostatečný počet kvalitních jazykových korpusů, na nichž budeme schopni sémantický model natrénovat. To-

muto kroku se blíže věnujeme v kapitole 6. Jak vyplynulo z prostudovaných metod, většina jich vyžaduje použití buď'to větně zarovnaných korpusů, nebo kombinaci korpusů jednojazyčných se slovníky. Na tyto druhy korpusů se proto zaměříme. Opět nám zde pomůže soutěž SemEval-2017, která některé jazykové korpusy přímo doporučuje.

U metod používajících mapování budeme také potřebovat získat jednojazyčné slovní vektory, které budeme transformovat do společného sémantického prostoru. Jednou z možností by bylo tyto vektory ručně vytvořit, ale vzhledem k tomu, že se jedná o časově náročný proces, a protože pro požadované jazyky kvalitní modely již existují, rozhodli jsme se použít již existující jednojazyčné sémantické modely.

4.3 Testování metod

Před tvorbou vlastního systému jsme se rozhodli nejprve otestovat některé z vícejazyčných sémantických metod. To nám umožní lépe pochopit jak metody fungují a také můžeme vyzkoušet jejich účinnost. Při výběru testovaných metod jsme se přitom snažili, aby byly zastoupeny metody používající různé přístupy ke tvorbě slovních vektorů. Vlastní systém pak budeme moci postavit na kombinaci některých z těchto otestovaných metod v závislosti na tom, jakých výsledků metody dosáhnou.

Při testování metod se vždy budeme snažit nejprve najít vhodné parametry metody s použitím malého jazykového korpusu a s malou velikostí dimenze sémantického prostoru. To nám umožní rychle vyzkoušet větší množství parametrů. Po nalezení vhodných parametrů pak model natrénujeme na větším množství dat, abychom získali porovnatelné výsledky. Tomuto kroku se blíže věnujeme v kapitole 7.

4.4 Vlastní systém

Obdobným způsobem budeme postupovat i při tvorbě vektorů vlastního systému. Nejprve použijeme menší jazykový korpus k nalezení vhodných parametrů metody a konečný systém pak natrénujeme na velkém korpusu s použitím *gridových výpočtů*. K tomu využijeme služby *MetaCentra Cesnet*. Metacentrum je virtuální organizace otevřená akademickým pracovníkům, zaměst-

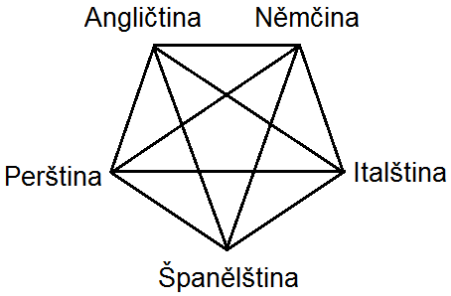
nanců, a studentům vědeckovýzkumných institucí v České republice. Organizace zajišťuje a koordinuje provoz distribuované výpočetní infrastruktury. Prostředí MetaCentra umožňuje využití zapojených výpočetních a datových zdrojů pro řešení velmi náročných výpočetních úloh [18].

Pro vlastní tvorbu jazykových vektorů z textových korpusů jsme se rozhodli použít jednu z dostupných a ověřených knihoven. V úvahu přicházely knihovny *Word2vec*, *GloVe* a *fastText*. Jelikož je knihovna fastText vylepšením knihovny Word2vec, vybírali jsme pouze mezi knihovnami GloVe a fastText. Nakonec jsme se přiklonili k použití knihovny fastText, neboť se jedná o novější a také rychlejší knihovnu, což nám dále umožní snadno a rychle ladit parametry vznikajícího systému.

5 SemEval-2017

V roce 2017 se konala mezinárodní vědecká soutěž SemEval-2017, jejíž druhý úkol se zabýval sémantickou podobností slov [4]. Tento úkol (v originále *Multilingual and Cross-lingual Semantic Word Similarity*) se skládal ze dvou podúkolů.

První podúkol se zabýval jednojazyčnou sémantickou podobností (tj. testovala se sémantická podobnost dvojice slov či frází stejného jazyka), druhý podobností vícejazyčnou, jak ze zobrazeno na obrázku 5.1. V obou případech bylo použito pětice jazyků, konkrétně angličtiny, němčiny, italštiny, španělštiny a perštiny.



Obrázek 5.1: Druhý podkúkol soutěže se zabýval vícejazyčnou sémantickou podobností slov a frází angličtiny, němčiny, italštiny, španělštiny a perštiny.

5.1 Zúčastněné systémy

Druhého podúkolu se zúčastnilo přes deset různých systémů. V tabulce 5.1 jsou zobrazeny výsledky několika systémů, které dosáhly nejlepších výsledků. Výsledek je vypočten jako harmonický průměr Pearsonovi a Spearmanovi korelace z šesti nejlepších výsledků z celkových deseti vícejazyčných testů.

Tato tabulka také ilustruje, jakých výsledků v současnosti dosahují systémy vícejazyčné sémantické podobnosti slov. Jak je vidět, na prvním místě se s výrazným odstupem umístil systém *Luminoso*.

Tabulka 5.1: Ukázka výsledků systémů zúčastněných v soutěži SemEval-2017.

Pořadí	**Systém**	**Výsledek**
1	Luminoso	0,754
2	NASARI	0,598
3	OoO	0,567
4	SEW	0,558
5	HCCL	0,464
6	RUFINO	0,336

5.1.1 Luminoso

Systém Luminoso od autorů *Speer* a *Lowry-Duda* [33] kombinuje několik metod sémantické podobnosti slov. Využívá předtrénovaných jednojazyčných modelů *GloVe* a *Word2vec*, kombinovaných se sémantickou sítí *ConceptNet* pomocí *rozšířeného retrofittingu*.

Autoři nejprve vytvořili podgraf sémantické sítě ConceptNet, který obsahuje vhodné sémantické vazby mezi slovy v určitém minimálním počtu. Poté aplikovali rozšířený retrofitting na oba jednojazyčné modely zvlášť. Tím oba tyto modely obohatili o znalosti ze sémantické sítě a navíc byl rozšířen jejich slovník, a to i o neanglická slova.

Následně autoři zvolili konečný slovník, přičemž brali v úvahu především četnost výskytů jednotlivých slov v textu. Zde využili faktu, že oba modely již byly seřazeny sestupně podle četnosti slov. Oba modely pak propojili tím způsobem, že spojili odpovídající slovní vektory. Nakonec pomocí rozkladu na singulární hodnoty (*SVD*) zredukovali dimenzi vzniklého sémantického prostoru. Tím získali první verzi vícejazyčného modelu.

Autoři dále experimentovali a pokusili se tento model obohatit o další informace. V prvním případě použili slovní vektory *Polyglot*, ve druhém případě větně zarovnaný paralelní korpus *OpenSubtitles2016*. Právě tato druhá varianta přinesla další zlepšení.

ConcepNet Numberbatch

Vytvořené sémantické vektory a jejich další aktualizace autoři zveřejnili pod názvem *ConceptNet Numberbatch* [7]. V současnosti (2018) se jedná o *state-of-the-art* sémantické vektory a zřejmě nejlepší dostupné vícejazyčné séman-

tické vektory.

Nevýhodou těchto slovních vektorů je ale malá velikost slovníku, která je dána především závislostí na velikosti sémantické sítě. Celý model obsahuje méně než dva miliony slov a kolokací pro celkem 78 jazyků, přičemž z jazyků, na které je tato práce zaměřena (tj. angličtina, němčina, španělština, čeština a čínština) je zastoupeno necelých 650 tisíc slov a kolokací.

Tyto sémantické vektory jsme se rozhodli použít pro porovnání úspěšnosti s naším systémem.

5.2 Testovací data

V soutěži byla použita dvojice testovacích dat, pro každý podúkol jedna. První testovací data jsou určena pro sémantickou podobnost uvnitř jednoho jazyka, druhá pro podobnost vícejazyčnou.

Autoři soutěže nejprve vytvořili jednojazyčná testovací data pro angličtinu s využitím ručního ohodnocení od několika anotátorů. Tato data pak ručně přeložili do dalších čtyř jazyků. Vícejazyčná testovací data byla vytvořena automaticky podle dat jednojazyčných. Všechna testovací data následně ještě ověřena lidskými rozhodčími.

Vzhledem k tomu, že tři z pěti jazyků, pro která jsou tato data určena, bude používat i vytvářený systém, a protože se jedná o nová data určená přímo pro hodnocení sémantické podobnosti slov, rozhodli jsme se je použít pro testování vytvářeného systému.

6 Příprava jazykových korpusů

Pro otestování úspěšnosti některých metod a vytvoření konečného systému jsme museli nejprve získat vhodná trénovací data. Protože vzniklý systém bude používat jazyky angličtinu, němčinu, španělštinu, češtinu a čínštinu, vyhledávali jsme korpusy s těmito jazyky.

6.1 Jazykové korpusy

Pro další práci jsme získali jazykové korpusy dvojího druhu. Jednak jednojazyčné korpusy, které obsahují texty z jednoho jazyka a nejsou nijak vícejazyčně zarovnány, a pak také větně zarovnané korpusy, které se skládají z překladů vět.

6.1.1 Jednojazyčné korpusy

Prvním zdrojem jednojazyčných korpusů byla internetová encyklopedie *Wikipedie*. Vycházeli jsme přitom z informací soutěže SemEval-2017, která doporučuje využít výpisy článků z této encyklopedie jako jednojazyčných trénovacích dat.

Jedná se o korpus se staršími články, které byly původně určeny k trénování slovních vektorů *Polyglot* [13]. Pro požadovaných pět jazyků je zde k dispozici téměř 17 GB dat v textovém formátu. Různé jazyky jsou ale zastoupeny v různé míře, přičemž největší korpus je anglický a nejmenší český a čínský.

I proto jsme se rozhodli použít i další korpusy. Druhým důvodem bylo, že jsme chtěl mít v trénovacích datech zastoupeno více typů textu, než jen encyklopedické články. Využili jsme proto automaticky shromažďované články z internetových novin (*News Crawl*) z let 2008 a 2016 a novinové komentáře (*News Commentary*) [5]. Tím jsme získali dalších přibližně 5 GB textových dat.

Posledním korpusem byly další výpisy článků z Wikipedie, tentokrát ale pouze pro čínštinu [6], pro kterou jsme stále neměli dostatek dat. Tento kor-

pus přidal další téměř 2 GB dat.

Tabulka použitých korpusů

Tabulka 6.1 udává počty slov v použitých jednojazyčných korpusech pro různé jazyky. Počty jsou uvedeny v miliónech.

Tabulka 6.1: Počty slov z různých jazyků v použitých jednojazyčných korpusech v miliónech.

	cs	en	de	es	zh
Wikipedia	82	1 683	583	394	314
News Crawl 2008	69	304	120	48	0
News Crawl 2016	92	0	0	0	0
News Commentary	0	10	8	0	0

6.1.2 Větně zarovnané korpusy

Pro větně zarovnané korpusy jsme se opět drželi doporučení SemEval-2017 a jako první zdroj jsme použili korpus *Europarl* [9]. Jedná se o zápisy z Evropského parlamentu, které se překládají do úředních jazyků Evropské unie. Tím jsme získali asi 2,5 GB textových dat, ale bez zastoupení čínštiny.

Jako další zdroj trénovacích dat jsme použili paralelní titulky k filmům *OpenSubtitles2018* [11]. Tím jsme získali dalších téměř 14 GB dat, stále však bez zastoupení čínštiny.

Pro čínštinu se nám podařilo získat několik menších jazykových korpusů [1]. Jednalo se o překlady anglických článků a knih v celkové velikosti přes 1 GB.

Nakonec jsme použili ještě česko-anglický paralelní korpus *CzEng 1.7* (*Czech-English Parallel Corpus*), vytvořený na *Ústavu formální a aplikované lingvistiky Matematicko-fyzikální fakulty Univerzity Karlovy* [8]. Jedná se o kvalitní, udržovaný korpus, vytvořený z mnoha zdrojů, jako jsou překlady novinových článků, překlady knih a překlady technických dokumentů, o celkové velikosti 1,5 GB.

Tabulka použitých korpusů

Tabulka 6.2 udává počty slov v použitých větně zarovnaných korpusech pro různé jazyky. Počty jsou uvedeny v miliónech.

Tabulka 6.2: Počty slov z různých jazyků v použitých větně zarovnaných korpusech v miliónech.

	cs-en	de-en	es-en	zh-en	cs-de	cs-es	de-es
Europarl	30	102	111	0	26	28	99
OpenSubtitles 2018	436	260	732	0	164	384	209
Casia 2015	0	0	0	45	0	0	0
Casict 2015	0	0	0	47	0	0	0
NEU	0	0	0	60	0	0	0
zhBook	0	0	0	46	0	0	0
CzEng 1.7	245	0	0	0	0	0	0

6.2 Předzpracování dat

Všechny jazykové korpusy bylo nezbytné před dalším použitím nejprve předzpracovat.

6.2.1 Formátování dat

Jako první krok jsme museli z dat odebrat nepotřebné informace a upravit je do formátu vhodného pro další zpracování. Jednalo se například o odebrání číselných označení článků, které se nacházely v korpusech Wikipedie, odebrání očíslování řádků v některých korpusech apod. Dále bylo potřeba rozdělit některé větně zarovnané korpusy, které byly uloženy v jednom souboru, do dvojic paralelních souborů, jak je to obvyklé. V tomto případě věta na daném řádku v jednom souboru odpovídá překladu věty ve druhém souboru na stejném řádku.

K tomuto účelu jsme v jazyce Java vyrobili jednoduchý program, který zpracoval zadané korpusy a odstranil nepotřebné informace, nebo korpus převedl do vhodného formátu.

6.2.2 Tokenizace

Dalším krokem bylo provést tokenizaci slov. Jedná se o proces, při němž je text rozdělen na *tokeny*, základní informační celky. Součástí tohoto procesu je obvykle i nahrazení velkých písmen malými.

K tomu jsme se rozhodli využít Stanfordský parser [12].

Stanford Parser

Stanford Parser je program vyvinutý Stanfordskou univerzitou určený pro zpracování gramatické struktury vět, jeho součástí je i *tokenizer*.

Nástroj *Stanford Tokenizer* je vytvořen především k tokenizaci anglických textů, z dostupné dokumentace ale vyplývá, že jej lze s úspěchem použít i pro další jazyky psané latinkou. *Stanford Words Segmenter* je pak určen pro zpracování arabštiny a čínštiny.

Tyto nástroje jsme použili pro tokenizaci jazykových korpusů a nahrazení velkých písmen malými.

6.2.3 Kolokace a jazykové prefixy

Dalším krokem bylo najít *kolokace*, což jsou často se vyskytující slovní spojení. Tento krok není nezbytný, ale protože testovací data kolokace obsahují, rozhodli jsme se jej zahrnout.

K tomuto účelu jsme v jazyce Java vytvořili jednoduchý program, který projde zadaný text, kolokace v něm nalezne a patřičné tokeny spojí. Při implementaci jsme přitom použili vzorec použitý v knihovně Word2vec [28]. Program prochází text a ukládá si počty slov a slovních bigramů a podle vzorce 6.1, kde δ je minimální počet výskytů bigramu, $sum(w_i w_j)$ je počet výskytů bigramu slov w_i a w_j a $sum(w_i)$ je počet výskytů slova w_i, pak určí skóre bigramu. Pokud skóre překročí daný práh, je bigram kolokací.

$$score(w_i, w_j) = \frac{sum(w_i w_j) - \delta}{sum(w_i) \times sum(w_j)} \tag{6.1}$$

Patřičné tokeny jsme pak spojili podtržítkem. Z tokenů „sluneční“ a „soustava“ se tak stal jeden token „sluneční_soustava“.

Nakonec jsme všem tokenům přidali prefix jazyka. K tomuto kroku jsme se rozhodli především proto, že jsme chtěli být schopni odlišit stejně psaná slova, která ale v různých jazycích budou mít různý význam. Prefix se skládá z dvojpísmenné zkratky jazyka a dvojtečky. Token „planeta“ se tak změnil na token „cs:planeta“.

Tím byly jazykové korpusy připraveny.

7 Testování vícejazyčných metod

Před tvorbou vlastního systému jsme otestovali úspěšnost několika vybraných vícejazyčných metod na menší množině dat. Záměrně jsme se přitom zaměřili na metody pracující se slovními vektory. Metody jsme vybírali tím způsobem, aby byly zastoupeny různé přístupy ke tvorbě vícejazyčných slovních vektorů.

7.1 MultilingualCCA

Nejprve jsme vyzkoušeli metodu *MultilingualCCA* (*Vícejazyčná kanonická korelační analýza*). Jak již bylo řečeno, tato metoda spadá do kategorie jednojazyčného mapování s využitím slovníků jako paralelního korpusu.

Metoda použije vzorky z již existujících jednojazyčných sémantických modelů a provede na nich kanonickou korelační analýzu, jejímž výstupem je dvojice transformačních matic. Tyto matice jsou pak použity k transformování jednoho sémantického prostoru do druhého.

7.1.1 Jednojazyčné slovní vektory

Prvním krokem bylo získat vhodné jednojazyčné slovní vektory. Jak již bylo řečeno v kapitole 4, rozhodli jsme se použít existující předtrénované modely.

Jedná se o modely s dimenzí 300 vytvořené knihovnou *fastText* na článcích z Wikipedie pomocí skip-gramu [3]. Tyto jednojazyčné modely jsou k dispozici pro 294 jazyků.

7.1.2 Slovníky

Z pěti použitých sémantických prostorů jsme dále museli získat vzorky, kde vektory z jednoho jazyka budou odpovídat překladům vektorů do druhého jazyka. Jako cílový jazyk, do kterého budeme ostatní jazyky mapovat, jsme zvolili angličtinu. Bylo proto potřeba získat čtveřici slovníků, které by šlo použít pro překlad zbylých čtyř jazyků do angličtiny.

Původně jsme tyto slovníky vytvořili tím způsobem, že jsme z jazyků kromě angličtiny vybrali vzorek 30 tisíc slov a ty jsme pomocí internetového překladače *Bing Microsoft Translator* přeložili do angličtiny. Zkoušeli jsme použít i překladač *Google Translate*, ale ten nedokázal takové množství slov najednou přeložit. Od těchto slovníků jsme ale vzápětí upustili, neboť se nám podařilo získat slovníky z knihovny *MUSE* (viz kapitola 7.2.2), které jsou obsáhlejší a zřejmě i přesnější.

S použitím těchto čtyř slovníků jsme pak ze sémantických modelů získali vzorky dat. Jednalo se o čtyři páry matic, kdy vždy jedna matice obsahovala slovní vektory z anglického modelu a druhá slovní vektory z jiného jazyka, které odpovídaly překladům vektorů z první matice.

7.1.3 Kanonická korelační analýza

Dalším krokem bylo použít kanonickou korelační analýzu. K tomu jsme použili program *Matlab*.

Postupně jsme načetli všechny páry matic a spustili jsme na nich kanonickou korelační analýzu. Výstupem byla dvojice matic V a W. Vynásobením matice V s inverzí matice W jsme pak získali transformační matici, kterou lze použít pro namapování zdrojového sémantického prostoru do prostoru cílového, kterým je angličtina.

Následuje ukázka použitého zdrojového kódu z programu *Matlab*, kde *source* je matice zdrojového jazyka, *target* matice jazyka cílového a *mapping* je výsledná transformační matice.

```
[V, W] = canoncorr(target, source);
mapping = V * inv(W);
```

Tímto způsobem jsme získali transformační matice pro všechny čtyři jazykové páry. Ty jsme pak použili k transformaci původních jazykových modelů do prostoru angličtiny.

7.1.4 Výsledky

Z každého z pěti jednojazyčných sémantických prostorů jsme pro mapování použili 300 tisíc nejčastějších slov. Vzniklý vícejazyčný vektorový model tak

obsahuje 1,5 miliónu slov a má velikost 4,7 GB. Při testování úspěšnosti jsme použili Pearsonovu korelaci a již zmíněná testovací data ze soutěže SemEval-2017.

Pokud se v datech vyskytla kolokace (tj. spojení několika slov), která se nevyskytovala ve slovníku, použili jsme přístup *Bag-of-Words*. V tom případě je kolokace rozdělena na jednotlivá slova a v modelu jsou nalezeny slovní vektory těchto slov. Tyto vektory jsou pak sečteny a kolokaci reprezentuje nově vzniklý vektor. Pokud nelze pro zadanou dvojici slov či kolokací určit sémantickou podobnost, protože slovní vektor se v modelu nevyskytuje, je použita průměrná hodnota z dosud vypočtených sémantických podobností pro daný test.

V tabulce 7.1 můžeme vidět dosažené výsledky. V prvních třech sloupcích lze vidět výsledky při testování sémantické podobnosti uvnitř jednoho jazyka, v dalších třech pak podobnost vícejazyčnou. Nakonec je uvedena průměrná hodnota z těchto šesti výsledků.

Tabulka 7.1: Výsledky metody MultilingualCCA

en	**de**	**es**	**de-es**	**en-de**	**en-es**	**Průměr**
0,661	0,638	0,673	0,632	0,646	0,653	**0,651**

Jak je vidět metoda dosahuje poměrně dobrých výsledků, přičemž podmínkou je použití kvalitních jednojazyčných modelů. Výhodou je také rychlost metody, neboť výpočet kanonické korelační analýzy a vytvoření transformační matice trvá řádově desítky vteřin.

7.2 Word Translation Without Parallel Data

I další testovaná metoda používá transformaci existujících sémantických prostorů. Používá k tomu iterativní Prokrustovu analýzu.

7.2.1 Předtrénované modely

Autoři metody uveřejnili vícejazyčné vektory pro celkem 30 jazyků, které získali namapováním jednojazyčných vektorů do prostoru anglického jazyka.

Využili přitom vektorové modely fastText, se kterými jsme pracovali v kapitole 7.1.1. Z každého jazyka namapovali 200 tisíc nejčetnějších slov z původních vektorů.

V předtrénovaných vícejazyčných modelech ale není zastoupena čínština, tu jsme proto museli s použitím knihovny *MUSE* domapovat.

7.2.2 Knihovna MUSE

K metodě autoři vytvořili knihovnu *MUSE* (*Multilingual Unsupervised or Supervised word Embeddings*) [10]. Ta je napsána v jazyce *Python* a dokáže transformovat zdrojový sémantický prostor do prostoru cílového, a to bud'to bez použití paralelních dat, nebo s použitím slovníků. Autoři pro namapování použili model se slovníky.

Tuto knihovnu jsme použili k namapování čínštiny do společného jazykového prostoru. Stejně jako autoři metody jsme k tomu použili model se slovníky.

7.2.3 Výsledky

Při testování metody jsme postupovali obdobně jaky v minulém případě. Opět jsme použili testovací data SemEval-2017 a Pearsonovu korelaci. U kolokací, které se nenacházely ve slovníku, jsme použili *Bag-of-Words*, a pokud se nepodařilo nalézt slovní vektor, použili jsme průměrnou hodnotu z dosud vypočtených sémantických podobností.

V tabulce 7.2 můžeme vidět dosažené výsledky v rámci jednoho jazyka, výsledky vícejazyčné a průměr z těchto hodnot.

Tabulka 7.2: Výsledky metody Word Translation Without Parallel Data.

en	**de**	**es**	**de-es**	**en-de**	**en-es**	**Průměr**
0,663	0,638	0,685	0,637	0,643	0,656	**0,654**

Metoda dosáhla mírně lepších výsledků než předchozí *MultilingualCCA*. Podmínkou opět bylo použití kvalitních jednojazyčných korpusů. Možnou nevýhodou metody je její časová náročnost, zejména není-li výpočet proveden

na grafickém akcelerátoru. Metoda totiž intenzivně využívá algoritmy, které lze s použitím grafického akcelerátoru významně urychlit.

7.3 Random Translation Replacement

Další testovanou metodou je *Random Translation Replacement* (*RTR*). Tato metoda patří do kategorie pseudo-vícejazyčných metod s použitím slovníků jako paralelního korpusu.

Metoda používá jednojazyčné jazykové korpusy, v nichž nahradí část slov za jejich překlady a tyto korpusy pak spojí. Na vzniklém psudo-vícejazyčném korpusu lze následně trénovat vícejazyčný sémantický model.

7.3.1 Jazykový korpus

Jednojazyčné jazykové korpusy jsme měli z kapitoly 6 již předzpracovány, dalším krokem proto bylo nahradit část slov za jejich překlady.

Opět jsme vytvořili jednoduchý program v jazyce Java, který slovo z korpusu, které se nacházelo ve slovníku, s určitou pravděpodobností nahradil za jeho překlad do jiného jazyka. Znovu jsme přitom použili slovníky z knihovny MUSE.

Ukázka korpusu

Následuje ukázka z vytvořeného korpusu.

cs:základem cs:nebeské_mechaniky en:are en:work cs:keplera cs:a cs:newtona cs:.
en:is es:iraq de:die en:next zh:阿富汗en:?
de:wäre en:n't es:vos de:weiß en:it en:?
es:eso es:genera es:agravios en:that es:abonan es:el es:apoyo es:a en:the en:pirates es:.
de:in de:besserer en:constitution de:, en:but de:nicht de:allein de:.
en:gold zh:物價en:even es:golpeó en:a en:record-high zh:$ en:1,300

en:recently en:.
en:this en:strategy en:has cs:již cs:vyrobené en:results en:.

7.3.2 Knihovna fastText

Pro trénování vícejazyčného modelu jsme se rozhodli použít knihovnu *fastText*. Jedná se o nástupce knihovny *Word2vec*, která během trénování používá i písmenné n-gramy slov.

Stejně jako u knihovny *Word2vec* lze při trénování použít modely skip-gram, nebo CBOW a negativní, nebo hierarchické vzorkování.

7.3.3 Výsledky

Při testování této metody jsme nejprve ladili parametry na malém jazykovém korpusu. Jednalo se o 10 miliónů tokenů z každého jazyka, celkem tedy 50 milionů tokenů.

Zkoušeli jsme vliv dimenze, vliv použitého modelu, pravděpodobnosti překladu slova a vliv velikosti podslov. Výsledky můžeme vidět v tabulce 7.3. Konečný výsledek je přitom vypočten jako průměrná hodnota ze všech šesti dostupných testů.

Na první pohled je patrné, že model skip-gram je na tomto malém korpusu výrazně lepší než CBOW. Je také vidět určitý vliv podslov (písmenných n-gramů), které pomohly ke zlepšení. Z výsledků pak vyplynulo, že metoda dosahuje nejlepších výsledků, pokud je pravděpodobnost překladu slova nastavena na 50 %, což koresponduje s hodnotou, kterou použili i autoři metody.

Podle těchto výsledků jsme pak natrénovali slovní vektory na větší množině dat. Jednalo se o 50 milionů tokenů z každého jazyka, dohromady tedy 250 milionů tokenů. Použili jsme přitom model skip-gram, dimenzi vektorového prostoru 300, podslova délka 6 až 9 a pravděpodobnost překladu 50 %.

Výsledek vidíme v tabulce 7.4. Tyto výsledky již odpovídají výsledkům předchozích metod, a to i přesto, že použitý jazykový korpus je stále poměrně malý. Výsledky v jednojazyčných testech dopaly lépe než v testech vícejazyčných.

Tabulka 7.3: Ladění metody RTR na malé množině dat.

Dimenze	**Model**	**Překlad**	**Podslova**	**Výsledek**
300	SG	20 %	6-9	0,247
300	SG	20 %	0-0	0,236
300	CBOW	20 %	0-0	0,075
100	SG	20 %	0-0	0,243
100	SG	40 %	0-0	0,244
100	SG	50 %	0-0	0,262
100	SG	70 %	0-0	0,148

Tabulka 7.4: Výsledky metody RTR na větší množině dat.

en	**de**	**es**	**de-es**	**en-de**	**en-es**	**Průměr**
0,645	0,631	0,662	0,616	0,644	0,647	**0,641**

7.4 Bilingual Skip-gram without Word Alignments

Poslední z vyzkoušených metod je *Bilingual Skip-gram without Word Alignments* (*BSwWA*). Tato metoda patří do kategorie spojitě optimalizovaných metod, které používají větně zarovnaný korpus.

Metoda při trénování používá model skip-gram, přičemž předpokládá, že slova jsou v obou větách rovnoměrně zarovnána vzhledem ke všem ostatním slovům z obou vět. Metoda tedy předpokládá, že v kontextu každého slova z obou vět se nacházejí všechna ostatní slova z obou vět.

7.4.1 Modifikovaná verze

Tuto metodu jsme se rozhodli mírně pozměnit. Obě věty zkombinujeme do jediné, přičemž bereme v úvahu pořadí slov a poměr délek vět. Věty „Co budeš dělat ty?“ a „What are you gonna do?“ jsou pak zkombinovány do jediné věty „What are Co you budeš gonna dělat do? ty?“. Takto zkombinované věty pak půjde snadno natrénovat knihovnou fastText, aniž bychom museli měnit její optimalizační kritéria, jak to navrhli autoři metody.

Uvedeným způsobem jsme pak zkombinovali věty z předzpracovaných větně zarovnaných korpusů z minulé kapitoly a tyto nově vzniklé korpusy

jsme smíchali. Na tomto korpusu jsme následně trénovali vícejazyčný sémantický model knihovnou fastText.

Ukázka korpusu

Následuje ukázku vytvořeného korpusu.

> cs:a en:and cs:bud' en:please cs:prosím en:be cs:upřímný en:honest cs:. en:.
> en:would en:n't es:¿ en:you es:y en:know es:saben en:it es:qué en:? es:?
> zh:2008年en:what zh:败en:failed zh:在en:in zh:何处en:2008 zh:？ en:?
> de:warum en:why de:auf en:wait de:den en:for de:euro en:the de:warten en:euro de:? en:?
> de:vorlage de:von cs:předložení de:dokumenten cs:dokumentů de:: cs:: de:siehe cs:viz de:protokoll cs:zápis
> en:how en:did cs:jak en:you cs:jste en:get cs:se en:up cs:sem en:here cs:dostal en:? cs:?

7.4.2 Výsledky

Stejně jako v minulém případě, i nyní jsme nejprve ladili parametry metody na malé množině slov. Opět jsme přitom použili 10 milionů tokenů z každého jazyka, dohromady tedy 50 miliónů tokenů.

V tabulce 7.5 můžeme vidět výsledky. Ve všech případech jsme použili model skip-gram a dimenzi prostoru 100 bez použití podslov. Ladili jsme maximální poměr mezi délkami vět a minimální délku každé z vět. Z výsledků je patrné, že nejlépe vychází maximální poměr mezi délkami vět 1,5 a minimální délka každé z vět 5 tokenů.

S těmito parametry jsme v dalším kroku tuto metodu natrénovali na větší množině 50 milionů tokenů z každého jazyka, celkem tedy 250 milionů tokenů. Výsledky můžeme vidět v tabulce 7.6. Metoda dosáhla horších výsledků než předchozí metoda RTR. U této metody dosáhli vícejazyčné testy lepších výsledků, než testy jednojazyčné.

Tabulka 7.5: Ladění metody BSwWA na malé množině dat.

Poměr vět	**Minimální délka**	**Výsledek**
1,3	5	0,197
1,5	5	0,202
2,0	5	0,182
1,5	3	0,185
1,5	6	0,198
1,5	7	0,197
1,5	9	0,195
1,5	10	0,199

Tabulka 7.6: Výsledky metody BSwWA na větší množině dat.

en	**de**	**es**	**de-es**	**en-de**	**en-es**	**Průměr**
0,528	0,609	0,578	0,606	0,588	0,573	**0,580**

8 Vlastní systém

Vlastní systém jsme se rozhodli postavit na kombinaci několika z otestovaných metod. Předpokládali jsme přitom, že by se metody mohly vzájemně doplňovat.

8.1 Kombinace metod RTR a BSwWA

Nejprve jsme se rozhodli zkombinovat metody *Random Translation Replacement* s naší verzí metody *Bilingual Skipgram without Word Replacement*.

Vycházeli jsme z testů metod, kde metoda RTR dosáhla mírně lepších výsledků v jednojazyčných testech, než v testech vícejazyčných, a může proto pomoci především při vytváření jednojazyčných vazeb. Metoda BSwWA pak ve výsledcích dosáhla lepší úspěšnosti u vícejazyčných testů, než u testů jednojazyčných, a mohla by proto přispět právě při tvorbě vazeb vícejazyčných.

8.1.1 Textový korpus

Při výrobě jazykových korpusů, na kterých budeme model trénovat, jsme postupoval stejným způsobem jako u původních metod.

První korpus se skládal z jednojazyčných, předzpracovaných korpusů, u kterých jsme s pravděpodobností 50 % nahradili slovo za překlad ze slovníku.

Druhý korpus byl vytvořen z promíšených vět z větně zarovnaného korpusu, způsobem uvedeným u metody BSwWA. Použili jsme omezení, že maximální poměr mezi délkami vět je 1,5 a minimální délka každé věty je 5 tokenů.

Navíc jsme experimentovali i s použitím třetího korpusu, který se skládal z předzpracovaných, nijak dále neupravených jednojazyčných korpusů. Vycházeli jsme z předpokladu, že by tento korpus mohl dále pomoci při trénování jednojazyčných vazeb.

Všechny tyto korpusy jsme pak zkoušeli během testování v různém poměru kombinovat k dosažení co nejlepších výsledků. Sémantický model jsme opět trénovali za použití knihovny fastText, přičemž jsme jako v předchozích případech nejprve trénovali model na menší množině dat a optimalizovali jsme zastoupení jednotlivých jazykových korpusů. Ve druhé fázi jsme trénovali sémantický model na velké množině dat prostřednictvím *gridových výpočtů* a optimalizovali jsme parametry knihovny fastText.

8.1.2 Výsledky

Vliv zastoupení různých druhů korpusů

Nejprve jsme zkoušeli kombinovat trojici zmíněných korpusů v různém poměru k dosažení co nejlepších výsledků. Vždy jsme přitom použili 50 miliónů tokenů z každého jazyka, tj. 250 miliónů tokenů celkem.

Výsledky můžeme vidět v tabulce 8.4. První tři sloupce udávají zastoupení jednotlivých druhů korpusů, poslední sloupec udává průměrnou hodnotu z šesti dostupných testů. Z výsledků je patrné, že nejlepších výsledků bylo dosaženo při kombinaci korpusů z metody RTR a BSwWA v poměru 3 : 2. Naopak použití jednojazyčných korpusů výsledky zhoršovalo.

Tabulka 8.1: Výsledky při kombinování různých druhů korpusu.

RTR	**BSwWA**	**Jednojazyčné**	**Výsledek**
33 %	33 %	33 %	0,622
50 %	50 %	0 %	0,631
0 %	50 %	50 %	0,597
50 %	0 %	50 %	0,612
66 %	33 %	0 %	**0,647**
33 %	66 %	0 %	0,627
80 %	20 %	0 %	0,635
60 %	30 %	10 %	0,637

V tabulce 8.5 pak můžeme vidět porovnání obou základních metod s jejich kombinací. Zkombinování obou metod přineslo lepší výsledky, než když byly metody použity samostatně.

Tabulka 8.2: Porovnání metod.

Metoda	**en**	**de**	**es**	**de-es**	**en-de**	**en-es**	**Průměr**
RTR	0,645	0,631	0,662	0,616	0,644	0,647	0,641
BSwWA	0,528	0,609	0,578	0,606	0,588	0,573	0,580
Kombinace	0,638	0,635	0,664	0,641	0,652	0,652	**0,647**

Optimalizace parametrů knihovny fastText

Ve druhém kroku jsme trénovali sémantický model na velké množině dat. Vycházeli jsme z dosavadních výsledků a zkombinovali jsme korpusy metod RTR a BSwWA v poměru 3 : 2. Celkem bylo z každého jazyka zastoupeno 300 milionů tokenů, dohromady tedy 1,5 miliardy tokenů.

Vzniklý jazykový korpus jsme umístili na datový server MetaCentra, stejně tak jako knihovnu fastText. Pak jsme vytvářeli a spouštěl dávkové úlohy, kterými jsme na korpusu trénovali sémantický model, pokaždé s různými parametry knihovny.

V tabulce 8.3 můžeme vidět výsledky. První sloupec značí velikost kontextu, druhý velikost podslov a třetí velikost hashovací tabulky pro podslova. Výsledek je opět vypočten jako průměrná hodnota z šesti použitých testů. Jak můžeme vidět, nejlepších výsledků bylo dosaženo při použití kontextu 5, podslovech velikosti 6 až 9 a velikosti hashovací tabulky pro 5 miliónů podslov. Vzniklé vektory jsou dimenze 300 a velikost slovníku je téměř 3 milióny slov.

Tabulka 8.3: Trénování parametrů knihovny fastText na velkých datech.

Kontext	**Podslova**	**Hash**	**Výsledek**
5	6-9	2 M	0,699
5	6-9	5 M	**0,700**
5	3-6	2 M	0,686
5	3-6	5 M	0,654
5	3-6	10 M	0,670
8	3-6	2 M	0,582

8.2 Kombinace vektorů

V dalším kroku jsme se pokusili vytvořené vícejazyčné vektory dále vylepšit jejich zkombinováním s jinými modely.

Postupovali jsme tím způsobem, že jsme na použitých slovních vektorech provedli normalizaci průměru a vektory jsme propojili za sebe. Nezbytnou podmínkou přitom bylo, že se slovní vektor musí nacházet v obou sémantických prostorech, aby mohl být použit. Dochází tedy k redukci velikosti slovníku.

Omezením tohoto postupu je nárůst dimenze vektorů. Velká dimenze přitom není pro zachycení sémantických vztahů nezbytná a zachycená informace bude zřejmě do značné míry redundantní (tj. data budou korelovat). Navíc u souboru s vektory dojde k nárůstu velikosti.

8.2.1 Analýza hlavních komponent

Pro snížení dimenze vektorů jsme proto použili *analýzu hlavních komponent* (*PCA*) [35]. Jedná se o metodu sloužící k odstranění korelace dat, která je často používána ke snížení dimenzionality.

Metoda funguje tím způsobem, že transformuje zadanou matici do jiné souřadné soustavy, jak lze vidět ve vzorci 8.1, kde X je vstupní matice a Y je matice výstupní. Matice P je tvořena vlastními vektory a tvoří novou souřadnou soustavu. Tyto vektory matice P jsou přitom seřazeny sestupně podle rozptylu a pro redukci dimenze na n rozměrů pak stačí vybrat prvních n z nich. Metoda pracuje s co nejmenší ztrátou informace.

$$Y = XP \tag{8.1}$$

K výpočtu PCA nad novými vektory jsme používali program Matlab. Následuje ukázka použitého kódu, kde *matrix* je původní matice s propojenými modely a *newMatrix* je nová matice se sémantickými vektory redukovanými na dimenzi 300.

```
A = pca(matrix);
newMatrix = (matrix - mean(matrix)) * A(:, 1:300);
```

8.2.2 Výsledky

Nejprve jsme zkusili zkombinovat sémantické vektory z metod MultilingualCCA a Word Translation Without Parallel Data, získané v kapitole 7, s vektory natrénovanými na menší množině 50 miliónů tokenů z každého jazyka na kombinovaném korpusu metod RTR a BSwWA.

V tabulce 8.4 můžeme vidět výsledky před a po zkombinování vektorů. Použití vektorů z metody MultilingualCCA zde vedlo ke zlepšení výsledků, použití vektorů z metody Word Translation Without Parallel Data naopak ke zhoršení.

Tabulka 8.4: Kombinace vektorů na malé množině dat.

Metoda	**Výsledek**
RTR+BSwWA	0,647
PCA(MultilingualCCA, RTR+BSwWA)	**0,678**
PCA(WTWPD, RTR+BSwWA)	0,605

Vektory z metody MultilingualCCA jsme proto následně zkombinovali s vektory natrénovanými na velkém jazykovém korpusu. Výsledek před a po zkombinování vektorů můžeme vidět v tabulce 8.5. Tentokrát zkombinování vektorů vedlo ke zhoršení původních vektorů a navíc došlo k redukci slovníku z přibližně 3 miliónů slov na necelé 2 milióny slov.

Tabulka 8.5: Kombinace vektorů na velké množině dat.

Metoda	**Výsledek**
RTR+BSwWA	0,700
PCA(MultilingualCCA, RTR+BSwWA)	0,663

9 Diskuse výsledků

Nejlepších výsledků jsme dosáhli s použitím slovních vektorů z kapitoly 8.1. Jedná se o sémantický model s dimenzí 300 natrénovaný knihovnou fastText pomocí skip-gramu na kombinovaném korpusu z metod Random Translation Replacement a Bilingual Skip-gram without Word Aligment. Slovní vektory mají velikost slovníku 2 989 505 slov a kolokací, zastoupeno je přitom pět jazyků – angličtina, němčina, španělština, čeština a čínština.

Tyto vektory jsme porovnali se slovními vektory ConceptNet Numberbatch, které zvítězili v mezinárodní vědecké soutěži SemEval-2017. Tyto vektory mají ve zmíněných pěti jazycích zastoupeno 644 167 slov a kolokací. Jejich velikost je tedy znatelně menší.

V tabulce 9.1 vidíme porovnání obou slovních vektorů. Výsledky jsou vypočteny stejně jako v předchozích případech, tj. jedná se o Pearsonovu korelaci s testovacími daty ze soutěže SemEval-2017. V případě kolokace, která se nenachází ve slovníku, je použit přístup Bag-of-Words, a pokud nelze podobnost určit, je použita průměrná hodnota z dosud vypočtených podobností v daném testu. Z výsledků je patrné, že v testovaných jazycích a při tomto způsobu testování námi vytvořené sémantické vektory systém ConceptNet Numberbatch překonaly.

Na výsledcích také můžeme vidět, že vytvořené slovní vektory dosahují v jednotlivých testech stabilních výsledků okolo hodnoty $0,7$, zatímco slovní vektory systému Numberbatch v testech oscilují mezi hodnotami $0,55$ a $0,7$.

Tabulka 9.1: Porovnání vytvořených vektorů s vítěznými vektory ze soutěže SemEval-2017.

Systém	**en**	**de**	**es**	**de-es**	**en-de**	**en-es**	**Průměr**
RTR+BSwWA	0,695	0,707	0,689	0,697	0,713	0,698	0,700
Numberbatch	0,542	0,694	0,695	0,698	0,594	0,596	0,636

9.0.1 Nejpodobnější slova

V tabulkách 9.2 a 9.3 vidíme porovnání obou systémů v úkolu najít nejpodobnější slova k několika vybraným slovům. Zobrazena je i vypočtená podobnost, určená jako kosinus úhlu mezi slovními vektory. Přestože oba systémy pracovaly se stejnými slovy, nalezená nejpodobnější slova se značně lišila.

Oba systémy nacházely slova, která odpovídají synonymům, či překladům vybraného slova. Výrazný rozdíl je vidět v dotazu na slovo sůl (*en:salt*). Systém Numberbatch nabídl spíše chemické termíny, zatímco náš systém nalezl překlady.

Zajímavostí může být také hodnota vypočtené podobnosti slov. Podobnost slov vypočtená systémem Numberbatch nabývala mnohem vyšších hodnot než u našeho systému, což svědčí o tom, že slovní vektory leží velmi blízko u sebe. To je zřejmě dáno použitím metody retrofittingu během vytváření slovních vektorů.

Tabulka 9.2: Nejpodobnější slova u vytvořeného systému.

cs:voda		en:salt		zh:食品	
cs:vodu	0.853	en:salts	0.775	zh:食物	0.820
cs:vody	0.824	cs:sůl	0.752	en:food	0.749
es:agua	0.753	de:salz	0.744	en:foods	0.729
de:wasser	0.751	cs:soli	0.721	en:food/beverage	0.703
en:water	0.742	en:saltine	0.698	zh:糧食	0.700

Tabulka 9.3: Nejpodobnější slova u systému Numberbatch.

cs:voda		en:salt		zh:食品	
cs:vodička	0.972	en:water_in_ocean	0.991	cs:potravina	0.932
cs:voděnka	0.963	en:fluoroboride	0.986	zh:食物	0.927
en:rewater	0.947	en:dibasic_salt	0.986	zh:可食用的	0.915
en:waterward	0.947	en:calcium_stearate	0.982	zh:吃的	0.894
en:branch_water	0.940	zh:盐	0.969	zh:綠豆	0.894

10 Závěr

V této práci jsme se zabývali přístupy k modelování vícejazyčné sémantické podobnosti slov, prostudovali a popsali jsme množství vícejazyčných metod a vytvořili jsme vlastní systém pro měření vícejazyčné sémantické podobnosti slov z angličtiny, němčiny, španělštiny, češtiny a čínštiny. Zaměřili jsme se především na použití slovních vektorů, neboť se jedná o dobře fungující řešení, ke kterému již existuje mnoho vícejazyčných metod.

Nejprve jsme některé z existujících vícejazyčných metod implementovali a otestovali, abychom lépe pochopili jak fungují a abychom zjistili jakých dosahují výsledků. Pro tvorbu vlastního systému jsme pak otestované metody *Random Translation Replacement* a *Bilingual Skip-gram without Word Alignments* spojili, čímž jsme dosáhli lepších výsledků, než když byly metody použity samostatně. Natrénované slovní vektory našeho systému jsme pak ještě zkusili zkombinovat s vektory vytvořenými pomocí jiných metod, ale výsledky se nepodařilo dále vylepšit.

Vytvořený systém jsme nakonec porovnali se systémem *Numberbatch*, jenž zvítězil v mezinárodní vědecké soutěži *SemEval-2017*, která byla zaměřena na vícejazyčnou sémantickou podobnost slov. Náš systém přitom na použitých testovacích datech dosáhl o více než šest procent lepších výsledků.

V rámci pokračování práce se nabízí otestovat další metody a vyzkoušet některé jejich kombinace. Další možností je vytvořit slovní vektory pomocí několika různých modelů a tyto pak zkusit zkombinovat. Funkčnost systému by také mohla být v budoucnu rozšířena o další jazyky.

Literatura

[1] China workshop on machine translation. http://nlp.nju.edu.cn/cwmt-wmt/, 2017. Online; naposledy navštíveno 10. 5. 2018.

[2] Conceptnet 5.5 and conceptnet.io. https://old.opendatascience.com/blog/conceptnet-5-5-and-conceptnet-io/, 2017. Online; naposledy navštíveno 12. 5. 2018.

[3] Pre-trained word vectors. https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md, 2017. Online; naposledy navštíveno 10. 5. 2016.

[4] Semeval-2017 task 2. http://alt.qcri.org/semeval2017/task2/, 2017. Online; naposledy navštíveno 10. 5. 2018.

[5] Translation task - acl 2017 second conference on machine translation. http://www.statmt.org/wmt17/translation-task.html, 2017. Online; naposledy navštíveno 10. 5. 2018.

[6] Wikipedia monolingual corpora - linguatools. http://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/, 2017. Online; naposledy navštíveno 10. 5. 2018.

[7] Conceptnet numberbatch. https://github.com/commonsense/conceptnet-numberbatch, 2018. Online; naposledy navštíveno 10. 5. 2018.

[8] Czeng 1.7 | Úfal. http://ufal.mff.cuni.cz/czeng/czeng17, 2018. Online; naposledy navštíveno 10. 5. 2018.

[9] Europarl. http://opus.nlpl.eu/Europarl.php, 2018. Online; naposledy navštíveno 10. 5. 2018.

[10] Muse: Multilingual unsupervised and supervised embeddings. https://github.com/facebookresearch/MUSE, 2018. Online; naposledy navštíveno 10. 5. 2018.

[11] Opensubtitles. http://opus.nlpl.eu/OpenSubtitles.php, 2018. Online; naposledy navštíveno 10. 5. 2018.

[12] The stanford parser: A statistical parser. https://nlp.stanford.edu/software/lex-parser.shtml, 2018. Online; naposledy navštíveno 13. 5. 2018.

[13] R. Al-Rfou. Polyglot. https://sites.google.com/site/rmyeid/projects/polyglot, 2015. Online; naposledy navštíveno 10. 5. 2018.

[14] W. Ammar, G. Mulcaire, Y. Tsvetkov, Lample G., Ch. Dyer, and N. A. Smith. Massively multilingual word embeddings. https://arxiv.org/pdf/1602.01925.pdf, 2016. Online; naposledy navštíveno 26. 4. 2018.

[15] M. Biniz, R. Ayachi, and M. Fakir. Ontology matching using babelnet dictionary and word sense disambiguation algorithms. http://www.iaesjournal.com/online/index.php/IJEECS/article/view/13241, 2017. Online; naposledy navštíveno 10. 1. 2018.

[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. https://arxiv.org/pdf/1607.04606.pdf, 2017. Online; naposledy navštíveno 3. 5. 2018.

[17] P. F. Brown, P. V. deSouza, R. L. Mercer, Della Pietra V. J., and J. C. Lai. Class-based n-gram models of natural language. https://dl.acm.org/citation.cfm?id=176316, 1992. Online; naposledy navštíveno 25. 1. 2018.

[18] CESNET. Cesnet | náročné výpočty (metacentrum). https://www.cesnet.cz/sluzby/metacentrum/, 2018. Online; naposledy navštíveno 4. 5. 2018.

[19] A. Conneau, G. Lample, M. A. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. https://arxiv.org/pdf/1710.04087.pdf, 2018. Online; naposledy navštíveno 3. 5. 2018.

[20] J. Coulmance, J. Marty, G. Wenzek, and A. Benhalloum. Trans-gram, fast cross-lingual word-embeddings. https://arxiv.org/pdf/1601.02502.pdf, 2016. Online; naposledy navštíveno 1. 5. 2018.

[21] M. Faruqui, J Dodge, S. Jauhar, Ch. Dyer, E. Hovy, and N. A. Smith. Retrofitting word vectors to semantic lexions. https://www.cs.cmu.edu/~hovy/papers/15HLT-retrofitting-word-vectors.pdf, 2015. Online; naposledy navštíveno 4. 5. 2018.

[22] M. Faruqui and Ch. Dyer. Improving vector space word representations using multilingual correlation. http://repository.cmu.edu/cgi/viewcontent.cgi?article=1031&context=lti, 2014. Online; naposledy navštíveno 21. 1. 2018.

[23] S. Gouws and A. Søgaard. Simple task-specific bilingual word embeddings. http://www.aclweb.org/anthology/N15-1157, 2015. Online; naposledy navštíveno 26. 4. 2018.

[24] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. https://link.springer.com/article/10.3758%2FBF03204766, 1996. Online; naposledy navštíveno 12. 5. 2018.

[25] M. Luong, H. Pham, and Ch. Manning. Bilingual word representations with monolingual quality in mind. http://www.aclweb.org/anthology/W15-1521, 2015. Online; naposledy navštíveno 17. 1. 2018.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. https://arxiv.org/pdf/1301.3781.pdf, 2013. Online; naposledy navštíveno 22. 1. 2018.

[27] T. Mikolov, Q. L. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. https://arxiv.org/pdf/1309.4168.pdf, 2013. Online; naposledy navštíveno 10. 1. 2018.

[28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. https://arxiv.org/pdf/1310.4546.pdf, 2013. Online; naposledy navštíveno 4. 5. 2018.

[29] R. Navigli. Babelnet | the largest multilingual encyclopedic dictionary and semantic network. http://babelnet.org/, 2007. Online; naposledy navštíveno 10. 1. 2018.

[30] J. Pennington, R. Socher, and Ch. D. Manning. Glove: Global vectors for word representation. https://nlp.stanford.edu/pubs/glove.pdf, 2014. Online; naposledy navštíveno 20. 1. 2018.

[31] S. Ruder. A survey of cross-lingual embedding models. http://ruder.io/cross-lingual-embeddings/index.html, 2016. Online; naposledy navštíveno 20. 1. 2018.

[32] R. Speer and C. Havasi. Representing general relational knowledge in conceptnet 5. http://lrec-conf.org/proceedings/lrec2012/pdf/1072_Paper.pdf, 2012. Online; naposledy navštíveno 17. 1. 2018.

[33] R. Speer and J. Lowry-Duda. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. https://arxiv.org/pdf/1704.03560.pdf, 2017. Online; naposledy navštíveno 4. 5. 2018.

[34] O. Täckström, R. McDonald, and J. Uszkoreit. Cross-lingualword clusters for direct transfer of linguistic structure. http://www.aclweb.org/anthology/N12-1052, 2012. Online; naposledy navštíveno 27. 1. 2018.

[35] Wikipedia. Analýza hlavních komponent – wikipedie. https://cs.wikipedia.org/wiki/Anal%C3%BDza_hlavn%C3%ADch_komponent, 2018. Online; naposledy navštíveno 7. 5. 2018.

[36] Wikipedia. Distributional semantics - wikipedia. https://en.wikipedia.org/wiki/Distributional_semantics, 2018. Online; naposledy navštíveno 10. 5. 2018.

[37] Wikipedia. Semantic space - wikipedia. https://en.wikipedia.org/wiki/Semantic_space, 2018. Online; naposledy navštíveno 10. 5. 2018.

[38] WordNet. Wordnet | a lexical database for english. https://wordnet.princeton.edu/, 2018. Online; naposledy navštíveno 10. 5. 2018.

[39] Ch. Xing, D. Wang, Ch. Liu, and Y. Lin. Normalized word embedding and orthogonal transform for bilingual word translation. https://pdfs.semanticscholar.org/77e5/76c02792d7df5b102bb81d49df4b5382e1cc.pdf, 2015. Online; naposledy navštíveno 3. 5. 2018.