Mendelova univerzita v Brně
Provozně ekonomická fakulta

Rozpoznání pojmenovaných entit v textu

Diplomová práce

Vedoucí práce:
doc. Ing. František Dařena, Ph.D.

Bc. Martin Süss

Brno 2019

Zde bude vloženo zadání práce.

Poděkování

Děkuji svému vedoucímu diplomové práce doc. Ing. Františku Dařenovi, Ph.D. za poskytnutí velmi zajímavého tématu a za cenné rady, návrhy a připomínky. Děkuji také své rodině, která mě po celou dobu mého studia podporovala a poskytovala komfortní prostředí pro tvorbu této práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Rozpoznání pojmenovaných entit v textu** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 12. května 2019 .....

Abstract

Süss, M. Recognizing named entities in text. Diploma thesis. Brno: Mendel University in Brno, 2019.

This thesis deals with the named entity recognition (NER) in text. It is realized by machine learning techniques. Recently, techniques for creating word embeddings models have been introduced. These word vectors can encode many useful relationships between words in text data, such as their syntactic or semantic similarity. Modern NER systems use these vector features for improving their quality. However, only few of them investigate in greater detail how much these vectors have impact on recognition and whether they can be optimized for even greater recognition quality. This thesis examines various factors that may affect the quality of word embeddings, and thus the resulting quality of the NER system. A series of experiments have been performed, which examine these factors, such as corpus quality and size, vector dimensions, text preprocessing techniques, and various algorithms (Word2Vec, GloVe and FastText) and their parameters. Their results bring useful findings that can be used within creation of word vectors and thus indirectly increase the resulting quality of NER systems.

Keywords

Text mining, natural language processing (NLP), information extraction, named entity recognition (NER), machine learning, neural network, word embedding, word vectors, Word2Vec, GloVe, FastText.

Abstrakt

Süss, M. Rozpoznání pojmenovaných entit v textu. Diplomová práce. Brno: Mendelova univerzita v Brně, 2019.

Tato práce se zabývá rozpoznáváním pojmenovaných entit v textu, které je realizované technikami strojového učení. V nedávné době byly představeny techniky vytváření modelů vektorových reprezentací slov, které dokáží do vektorů zakódovat mnoho užitečných vztahů mezi slovy v textových datech, jako např. jejich syntaktickou či sémantickou podobnost. Moderní systémy pro rozpoznávání pojmenovaných entit tyto vlastnosti vektorů využívají, čímž výrazněji zlepšují svoji kvalitu. Málo z nich však detailněji zkoumá, jak velký vliv tyto vektory na rozpoznávání mají a jestli je lze optimalizovat pro ještě větší nárůst kvality rozpoznávání. Tato práce zkoumá různé faktory, které mohou ovlivnit kvalitu modelů vektorových reprezentací slov, a tím i výslednou kvalitu rozpoznávání pojmenovaných entit. V práci je vykonána série experimentů, které tyto faktory, jako je kvalita a velikost korpusu, počet dimenzí vektorů, techniky předzpracování textu či různé algoritmy (Word2Vec, GloVe a FastText) a specifické nastavení jejich parametrů, zkoumají. Jejich výsledky přinášejí řadu poznatků, které lze využít při vytváření vektorových reprezentací slov, a tím i nepřímo navýšit výslednou kvalitu rozpoznávání pojmenovaných entit.

Klíčová slova

Dolování dat z textu, zpracování přirozeného jazyka (NLP), extrakce informací, rozpoznávání pojmenovaných entit (NER), strojové učení, neuronová síť, vnoření slov, vektorová reprezentace slov, Word2Vec, GloVe, FastText.

Obsah

Seznam tabulek

1 Úvod a cíl práce

Dolování znalostí z dat (tzv. „Data mining“) je v současnosti dynamicky se rozvíjejícím odvětvím, které působí napříč mnoha obory, jako je např. obor informačních technologií, matematiky, psychologie či marketingu. Klade si za cíl objevit užitečné informace z dat. S příchodem internetu a dostupných technologií celkové množství dat exponenciálně vzrostlo. Výzkumná centra i společnosti si uvědomovaly nesčetné možnosti využití těchto dat, a tak se započala nová éra datové analýzy. Se stále více dostupnějším výpočetním výkonem se postupem času přecházelo od tradičního programování pomocí pravidel a podmínek k více sofistikované umělé inteligenci (Voss, 2017).

Na světě vznikají dva typy dat – strukturovaná (např. databáze) a nestrukturovaná (volně psaný text určitého jazyka). Dle průzkumů je na celém internetu nestrukturovaných dat až 80 % (Breakthrough Analysis, 2008), což značí obrovské možnosti jejich využití (ať už jako konkurenční výhoda společností či pro obecné blaho). Díky tomuto rozložení vznikla vědecká disciplína, pojmenovaná jako dolování z textu (tzv. „Text mining“), která se zabývá zkoumáním a analýzou velkých objemů nestrukturovaných dat (Rouse, 2018).

Jednou z metodologií dolování z textu je zpracování přirozeného jazyka (anglicky „Natural Language Processing“ – NLP), která v sobě zahrnuje spoustu dílčích úloh, jež se mohou vzájemně doplňovat (Expert System, 2016). NLP je úzce spjato s počítačovou lingvistikou[1], která je oborem aplikované lingvistiky. NLP využívá řadu úloh, jako je klasifikace textu, shlukování, sumarizace, extrakce informací či strojový překlad.

Rozpoznávání pojmenovaných entit (anglicky „Named Entity Recognition“, zkráceně NER) spadá do kategorie extrakce informací z textu. Pojmenovaná entita je „*slovo nebo slovní spojení označující určitou instanci, typicky jména osob, institucí, míst, zboží, uměleckých děl, telefonní čísla, e-mailové adresy, zkratky, jednotky apod.*“ (Centrum zpracování přirozeného jazyka, c2019a). Rozpoznání pojmenovaných entit lze provádět různými způsoby – od založených na pravidlech (anglicky „rule-based“) až po způsoby zahrnující hluboké neuronové sítě (tzv. „deep learning“). Pro anglický jazyk již v dnešní době existuje spousta přístupů, které dosahují excelentních výsledků a jsou velmi spolehlivé. Hlavním důvodem je to, že většina výzkumníků se zabývá rozpoznáváním pojmenovaných entit právě pro tento jazyk, který je ve světě (i ve světe informačních technologií) nejvíce rozšířený. Dalším důvodem je relativní jednoduchost jazyka v porovnání s jinými z hlediska lingvistického úhlu pohledu.

Čeština je jedním z jazyků, které se řadí mezi složitější, zejména z důvodu její flektivnosti (Cambridge Dictionary, 2019) a bohaté morfologii. Pro tento a jemu podobné jazyky je obtížnější vyvinout model pro rozpoznávání pojmenovaných entit, který by byl spolehlivý a takřka bezchybný. Často je pro takový model nutné naleznout správné charakteristiky jazyka, které nezanedbatelně ovlivňují kvalitu

[1] Lingvistika je věda zkoumající přirozený jazyk a dělí se na obecnou lingvistiku, aplikovanou lingvistiku, gramatiku a jazykovědu (Wikipedie, 2018).

rozpoznávání. V modelech se ale využívají i charakteristiky či techniky, které nejsou závislé na použitém jazyce (anglicky tzv. „language-independent features“).

Jednou z takových velmi moderních technik je např. vnoření slov (anglicky „word embeddings“), což zajišťuje reprezentaci slov pomocí mnoharozměrných vektorů. V nedávné době vědečtí výzkumníci Mikolov a kol. (2013) představili způsob tvorby vnoření slov, jenž za pomoci neuronových sítí a obrovského množství textových dat v daném jazyce dokáže vyprodukovat vektorový prostor, ve kterém jsou zachyceny sémantické a syntaktické závislosti daného jazyka. Podobná slova (resp. slova využívaná v podobném kontextu) tak vytvářejí v tomto prostoru přirozené shluky. Tyto vlastnosti vektorů mohou být přínosné pro zvýšení kvality rozpoznávání pojmenovaných entit, ale i v dalších úlohách NLP. Navíc lze pomocí jednoduchých algebraických operací s vektory dospět k zajímavým poznatkům.

Velmi populárním přístupem pro tvorbu modelů NER se stala kombinace neuronových sítí a modelů vnoření slov. Tato práce se zaměřuje na nejmodernější (tzv. „state-of-the-art“[2]) techniky (zejména pak na zmíněnou kombinaci), které umožňují sestrojit funkční model pro rozpoznávání pojmenovaných entit v českém jazyce.

1.1 Cíl práce

Cílem této práce je navrhnout a implementovat systém rozpoznávající pojmenované entity (NER) z českého textu. Výsledný model bude vytvořen pomocí moderních metod strojového učení. Pro trénování modelů budou využity rozsáhlé externí zdroje znalostí v podobě textových dat, která se budou dělit na texty s vyznačenými entitami a texty otevřené. Na základě anotovaného textu se modely naučí rozpoznávat pojmenované entity a generalizovat rozpoznávání na další, ještě nespatřený text. Otevřený text poslouží pro vytvoření moderních vektorových reprezentací textových dat (modelu vnoření slov). Ty systému NER přinesou znalost sémantické a syntaktické podobnosti slov na základě podobného kontextu, což by mělo vést k nezanedbatelnému zlepšení rozpoznávání pojmenovaných entit.

S modely vnoření slov souvisí dílčí cíl práce, ve kterém je snaha zjistit míru přínosu těchto modelů na úlohu NER a objevit optimální nastavení algoritmů pro vytváření vektorových reprezentací, které přinesou pro NER nejvyšší navýšení kvality rozpoznávání. Naplnění cíle se uskuteční sadou navržených experimentů, z nichž se každý zaměřuje na určitou část tvorby vnoření slov a zkoumá její vliv na NER.

[2]State-of-the-art je v oblasti NLP celosvětově využívaným pojmem, který se většinou vztahuje k určitému modelu či technice a značí, že se v dané době jedná o nejmodernější přístup.

2 Současný stav

Následující kapitola přináší přehled o současném stavu ve zkoumané problematice, jehož pochopení je nezbytné pro další části práce. Nejdříve bude předložena obecná teorie ohledně dolování textových dat, dále pak budou vysvětleny některé často používané pojmy pro snadnější orientaci v textu. Následují základní informace o strojovém učení, s detailnějším důrazem na neuronové sítě, neboť ty jsou v oblasti dolování textových dat dnes velmi rozšířené a populární.

Zbylé části pojednávají o specifických tématech v oblasti zpracování přirozeného jazyka, konkrétně o technikách pro vytvoření vektorových reprezentací slov (vnoření slov) a rozpoznávání pojmenovaných entit (NER) v textu. Jejich součástmi budou také přehledy volně dostupných zdrojů či existujících řešení.

2.1 Dolování textových dat

Příchod informačních technologií započal éru dat. Na jejím počátku docházelo více k ukládání dat nežli k jejich manipulaci, zejména z důvodu omezených výpočetních prostředků (Ismail, 2018). Na světě tak přibývalo množství uložených strukturovaných i nestrukturovaných dat. S příchodem dostupnějšího výpočetního výkonu se začaly využívat techniky pro dolování dat („Data mining“) ze strukturovaného textu (Li, 2017). Avšak i nestrukturovaná data, speciálně texty, v sobě také mohou skrývat mnoho informací. Pokud jich je dostatek, mohou subjektu přinést cenné poznatky, např. společnostem mohou pomoci v rozhodovacích procesech.

Techniky dolování textových dat („Text mining“) se začaly rozvíjet koncem 20. století jako odnož či rozšíření Data miningu (Grimes, 2007). Umožňují odhalovat ukryté informace v textu. Pokud se navíc jedná o dolování dat z psaného textu v určitém jazyce, hovoří se o technikách zpracování přirozeného jazyka (NLP, anglicky „Natural Language Processing“). Často se pojmy „Text mining“ a „NLP“ zaměňují. Některé zdroje však uvádějí (Expert System, 2016), že množina NLP technik je spíše komponentou Text miningu, která provádí lingvistickou analýzu textu, a Text mining samotný je pak zodpovědný za konečné dolování užitečných informací z textu.

2.1.1 Pojmy

V této sekci jsou představeny pojmy, které jsou v práci dále využívány. Znalost jejich významu je nezbytná pro pochopení souvislostí.

Zpracování přirozeného jazyka je na pomezí informatiky, matematiky, ale také **lingvistiky** (Centrum zpracování přirozeného jazyka, c2019b), proto je zde představen význam některých pojmů, které se používají pro určení vlastností textu či jeho elementů (Wikipedie, 2018):

- **kořen, prefix, infix a sufix slova** – kořen je nedělitelná část slova nesoucí základní význam, prefix, infix a sufix se ve flektivních jazycích přidávají před, resp. do a za kořen slova k vyjádření mluvnických kategorií,

- **morfém** – nejmenší vydělitelná část slova, která je nositelem věcného nebo gramatického významu, je ním kořen, prefix, infix či sufix slova,
- **morfologie** – věda zabývající se ohýbáním a pravidelným odvozováním slov pomocí morfémů,
- **lexém** – základní jednotka slovní zásoby jazyka, jedná se o množinu všech tvarů určitého slova nebo slovního spojení,
- **lemma** – základní tvar lexému, který se uvádí jako reprezentativní ve slovnících (někdy též jako slovníkový tvar),
- **syntax** – disciplína zabývající se vztahy mezi slovy ve větě, správným tvořením větných konstrukcí a slovosledem,
- **sémantika** – nauka o významu výrazů (vztahu těchto výrazů ke skutečnosti) z různých strukturních úrovní jazyka – morfémů, slov, slovních spojení a vět, popř. i vyšších textových jednotek,
- **kontext** – jazykové okolí sledované jednotky textu (věty, odstavce, verše apod.).

Jednotky textu (typicky slova či slovní spojení) tvoří větší celky, jako jsou věty (sekvence), dokumenty či korpusy:

- **token** – základní element korpusu, výskyt jednoho slova či fráze,
- **dokument** – množina tokenů uspořádaných do sekvencí či vět,
- **korpus** – rozsáhlá množina (kolekce) dokumentů,
- **n-gram** – n jednotek, např. tokenů či znaků, tvořících celek (gram),
- **stop-slovo** – nejběžnější slova daného jazyka, která v daném kontextu nemají výrazný vliv na význam.

2.1.2 Aplikace

Techniky (úlohy) NLP lze rozdělit do několika základních skupin podle způsobu provádění a cíle (Davydova, 2017b):

- klasifikace, někdy též kategorizace („classification“),
- získávání informací („information retrieval“),
- extrakce informací („information extraction“),
- sumarizace („summarization“),
- shlukování („clustering“),
- odpovídání na dotazy („question answering“),
- strojový překlad („machine translation“).

Klasifikace je nejběžnější úloha NLP. Jejím úkolem je identifikovat, do jaké kategorie určitý vstup (token, věta, sekvence, dokumente či korpus) patří. Může se jednat např. o filtrování spamů, klasifikaci novinových článků apod. (Stubbs a Pustejovsky, 2012).

Získávání informací je množina technik, které se snaží získat vhodný text (typicky dokumenty z obvykle velké kolekce), který uspokojuje určité požadavky, definované např. uživatelským dotazem (Manning, Raghavan a Schütze, 2008, s. 1). Nejznámějším příkladem je webový prohlížeč snažící se nalézt nejrelevantnější webové stránky na základě dotazu.

Techniky extrakce informací dokáží automatizovaně získávat z nestrukturovaných dat (textu) strukturované informace (Krallinger, 2016). Může se jednat např. o rozpoznávání pojmenovaných entit v textu či o určování slovních druhů (anglicky „part-of-speech tagging“, zkráceně POS). Právě rozpoznávání pojmenovaných entit je předmětem této práce.

Techniky sumarizace se snaží zkrátit rozsáhlé texty do menších celků, které přitom stále zachovávají podstatné informace. V době obrovského kvanta informací je stále větší tlak na jejich zkracování a poskytování stručného souhrnu, který zkrátí dobu čtení a zefektivní hledání relevantních dokumentů, a to je hlavním důvodem příchodu těchto technik (Garbade, 2018).

Techniky shlukování mají široké uplatnění, a to nejen v NLP. Využívají se v případech, kdy je třeba seskupit určité objekty, v NLP např. textové dokumenty, ačkoliv nejsou známé přesné vlastnosti těchto skupin. Metody shlukování umožňují v datech nalézt podobnosti, na základě kterých se vytvoří daný počet skupin (Ghaffari, 2015).

Existují situace, kdy se lidé mohou obracet se svými dotazy na určitý subjekt (např. na společnost či instituci), který musí investovat prostředky do lidské síly, jež dokáže na tyto dotazy odpovídat (např. cestovní agentury apod.). Pokud se většina dotazů opakuje, je na místě využít techniky, které dokáží na základě obsahu dotazu automatizovaně odpovědět či uživatele nasměrovat správným směrem (Arbuzova, 2018). Tyto systémy automatického odpovídání na dotazy se tedy řadí mezi NLP úlohy, neboť vždy je potřeba provést analýzu textu, která přinese informace, na základě kterých lze relevantně odpovědět. Systémy často využívají kombinaci různých NLP technik.

Poslední kategorie ze zmíněných (nikoliv však poslední možná) sdružuje techniky, které dovedou provést překlad z jednoho přirozeného jazyka do jiného, to vše automatizovaně. Pro spolehlivý systém je nutné mít k dispozici rozsáhlé množství trénovacích dat – paralelních textů (ve smyslu naprosto stejných obsahů) v jazycích, jejichž překlad má systém podporovat (GALA, c2019). Nejmodernější systémy využívají neuronové sítě v kombinaci s vektorovými reprezentacemi slov (viz sekce 2.2 a 2.3), díky čemuž dokáží vnímat také samotný obsah, nikoliv jen výskyt jednotlivých slov, což vede k daleko přesnějším překladům.

2.1.3 Data

Při zpracovávání přirozeného jazyka jsou kromě použitých technik neméně podstatná data, která se v úlohách využívají. Může jít o trénovací data, která slouží k vybudování modelů pomocí technik strojového učení, ale také o data, na kterých se úlohy v konečném důsledku vykonávají.

V textu je oproti strukturovaným datům obvykle daleko více rušivých elementů, kterých je vhodné se za použití vhodných metod zbavit. Kromě toho je obvykle nutné textová data transformovat do požadovaného formátu. Všem těmto postupům se obecně říká „předzpracování textu“ (anglicky „text preprocessing“). Lze je rozdělit do jedné ze čtyř obecných kategorií dle jejich účelu (Fortney, 2017) – **čistící** (odstraňovací), **anotační** (přinášející dodatečnou informaci o tokenu), **normalizační** (transformující tokeny do jiného tvaru) a **analytické** (využívající statistiky textu). Konkrétní techniky předzpracování jsou např. (Vijayarani, 2015):

- tokenizace – získávání jednotlivých tokenů z textu,
- změna formy tokenů (např. převod písmen na malá, čísel na jednotnou reprezentaci apod.),
- transformace slov na jejich kořeny (tzv. „stemming“),
- transformace slov na jejich lemmata (tzv. „lemmatization“),
- získání slovního druhu slova (anglicky „part-of-speech tagging“),
- expanze zkratek (anglicky „abbreviation expansion“),
- nalezení frází a slovních spojení (anglicky „phrase detection“),
- odstranění stop-slov (anglicky „stopwords removal“),
- odstranění nežádoucích znaků (např. interpunkčních, speciálních apod.),
- odstranění nejfrekventovanějších slov (např. pomocí metod založených na Zipfovém zákoně – tzv. „Z-metody“).

Většina zmíněných technik však z textu odebírá určitou informaci. Některé techniky tak mohou nejen pozitivně, ale i negativně ovlivnit kvalitu NLP úlohy[3]. Záleží pak, zdali přínos z předzpracování textu je větší nežli množství odstraňované informace. Vždy je však vhodné tyto techniky využívat s rozmyšlením. Techniky lze kombinovat, avšak i jejich pořadí vykonání může ve kvalitě předzpracování hrát významnou roli.

[3]Například odstraněním negujících slov, která se mnohdy považují za stop-slova (např. sloveso „nebýt“ v různých tvarech), v analýze sentimentu pravděpodobně dojde k výraznému zhoršení kvality modelu.

2.2 Strojové učení

Strojové učení je neodmyslitelnou součástí dolování textových dat a většina NLP úloh je dnes založena na jeho technikách. Jedná se o sadu algoritmů a metod, které přinášejí systémům schopnost se učit provádět specifické úkony, aniž by byly dopředu explicitně známy jejich přesné kroky vedoucí k výsledku (Expert System, c2019). Vnímá se jako podmnožina umělé inteligence. Proces učení může být realizován několika způsoby, např. pomocí trénovacích dat či prostřednictvím přímé zkušenosti.

Zpracování přirozeného jazyka využívá techniky strojového učení pro automatizovanou extrakci znalostí a vhledů z textových dat (Redmore, 2019), které nejsou na první pohled zřejmé a člověkem velmi těžko odhalitelné (GATE, c1995–2011, s. 5). Některé techniky dovedou ve velmi krátkém čase najít skryté souvislosti v textu a provádět úsudky na novém, dosud nespatřeném textu.

2.2.1 Způsoby učení a trénování

Techniky strojového učení lze rozdělit do čtyř skupin podle způsobu učení (resp. trénování modelu):

- učení s učitelem (anglicky „supervised learning“),
- učení bez učitele (anglicky „unsupervised learning“),
- kombinace obou předešlých (anglicky „semi-supervised learning“),
- zpětnovazební učení (anglicky „reinforcement learning“).

Techniky učení s učitelem předpokládají trénovací data, na základě kterých se budou schopné naučit souvislosti, které poté mohou uplatnit na nových datech. Trénovací datová množina musí obsahovat výstupy, které jsou očekávány na základě vstupů. Vyjádřeno matematicky, cílem je najít funkci $y = f(x)$, která nejlépe popisuje mapování množiny vstupních dat x na množinu výstupních y, jedná se tedy o prediktivní modely (Fumo, 2017). Konkrétními technikami jsou např. rozhodovací stromy, logistická regrese či neuronové sítě. Většina NLP úloh využívá právě techniky učení s učitelem. Zmínit lze např. analýzu sentimentu (vstupem je textový element a výstupem jeho sentiment) či rozpoznávání pojmenovaných entit (vstupem je sekvence tokenů a výstupem množina nalezených entit).

Techniky učení bez učitele využívají vstupní data, ke kterým však nejsou známé správné výstupy. Mají za úkol najít skryté společné vlastnosti, které data společně sdílejí. S těmito informacemi pak lze nakládat různě, např. vytvořit shluky dat s podobnými vlastnostmi (shlukování). I na tomto principu několik NLP úloh funguje, jedná se např. o analýzu podobnosti dokumentů (Kessel, 2018) či o algoritmy pro vytvoření vektorových reprezentací slov, které jsou pro tuto práci velmi důležité (viz sekce 2.3). Konkrétními technikami jsou např. tzv. „K-means shlukování“, samoučící se neuronová sít či asociační analýza (Mrázová, 2018).

Kombinace obou předešlých skupin technik využívá malou trénovací množinu dat, ve které je ke každému vstupu znám správný výstup a k tomu využívá většinou rozsáhlá data bez přiřazených výstupů (Expert System, c2019). Cílem těchto technik je naučit se co nejlépe mapovat vstupy na výstupy z omezeného množství trénovacích dat a přitom se naučit souvislosti a vztahy z rozsáhlé datové množiny bez výstupů, které mohou pomoci vybudovat spolehlivější model (Fumo, 2017). V NLP se často využívá tato kombinace tam, kde je limitovaný počet trénovacích dat (Blitzer a Zhu, 2008). Nejdříve je provedeno učení bez učitele na rozsáhlých textových korpusech, které přinese užitečné informace o vztazích a souvislostech mezi tokeny či dokumenty a následně je aplikováno učení s učitelem, které tyto informace využívá společně s trénovacími daty pro vybudování prediktivního či klasifikačního modelu (Sienčnik, 2015, s. 1).

Poslední ze skupin se nazývá zpětnovazební učení a jeho využití je spíše v robotice, neboť učení spočívá v iterativním se zlepšování na základě interakce s okolím a zpětné vazby z provedených akcí (Fumo, 2017). V NLP se tento přístup příliš neuplatňuje, ačkoliv v poslední době bylo provedeno pár úspěšných pokusů s využitím těchto technik v některých úlohách (Elvis, 2018).

Modely lze těmito přístupy trénovat jednorázově, tj. na jeden souvislý běh algoritmu vytvořit funkční model. Někdy však trénovací data mohou být dostupná postupně a je žádoucí model trénovat inkrementálně s příchodem nových dat. Této technice se říká inkrementální trénování a umožňuje učit model přírůstkově s novými daty. V NLP se tato technika může hodit např. při optimalizaci již natrénovaného modelu (anglicky tzv. „fine-tuning“) na specifickou datovou doménu či jej naučit novým funkcím. Je však nutné dávat pozor na tzv. „katastrofický problém zapomínání“, který může nastat při nesprávném využití inkrementálního trénování (Honnibal, 2017). K tomu může dojít v případě, že novými trénovacími daty se pozmění rozhodovací mechanismy modelu natolik, že již nejsou schopny provádět svou činnost na předešlých datech v odpovídající kvalitě. Tomu lze předejít tak, že tato přírůstková data se využijí společně s částí předešlých trénovacích dat, díky čemuž rozhodovací mechanismy „nezapomenou“ již naučené postupy.

2.2.2 Role dat

Data jsou ve strojovém učení klíčová. Jejich robustnost a reprezentativnost rozhoduje o výsledné kvalitě modelu. Jestliže jsou v trénovacích datech přítomny chyby či nepřesnosti ve dvojicích vstup–výstup, nelze očekávat spolehlivý model (anglicky se tomuto říká „garbage in – garbage out“). V NLP má navíc velký vliv doména trénovacích (textových) dat na schopnost modelu činit kvalitní rozhodnutí. Pokud se jedná o velmi specifickou doménu, bude většinou model fungovat spolehlivě pouze v ní.

U technik založených na učení s učitelem je zvykem trénovací množinu dat rozdělit na tři různě velké části[4] – na trénování, validaci a testování modelu (Shah, 2017). Trénovací část dat slouží k samotnému naučení modelu vykonávat svou činnost. Validační část se využívá v průběhu trénování (většinou v neuronových sítích) pro zajištění dostatečné generalizace a zamezení problému přeučení modelu (Matematická biologie, c2019c). Ovšem v některých případech nemusí být využita a místo ní je zvětšena některá z jiných částí[5]. K přeučení (anglicky tzv. „overfitting problem“) může dojít tehdy, jestliže si model sestaví mechanismus rozhodování striktně na data obsažená v trénovací množině (Elite Data Science, 2017). Pozná se to většinou tak, že na trénovacích datech dosahuje takřka 100% výsledků, ačkoliv na validační části pohoří.

Testovací část by měla obsahovat data, která nejsou přítomna v části trénovací, neboť se využije na finální evaluaci naučeného modelu, ve které je žádoucí zjistit jeho dostatečnou generalizaci a aplikaci na dosud nespatřených datech. Také by testovací množina měla představovat reprezentativní ukázku dat, která se budou objevovat v prostředí, v němž bude model nasazen.

2.2.3 Neuronové sítě

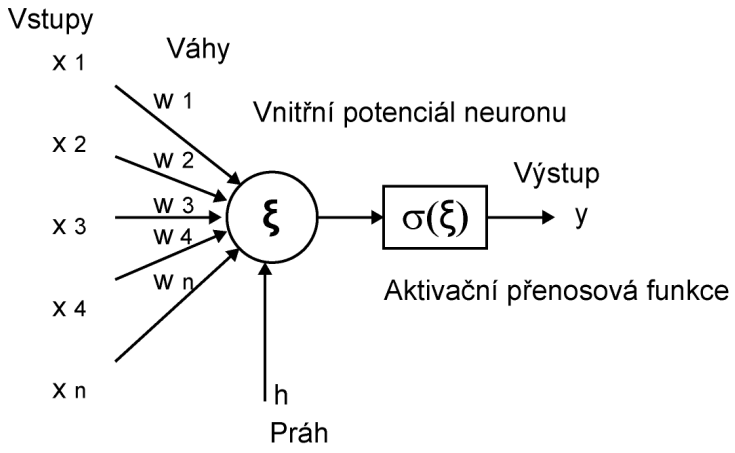
Neuronové sítě se poslední dobou těší velké popularitě, a to i v úlohách NLP, kde dlouhou dobu dominovaly tradiční přístupy strojového učení s využitím lineárních modelů, jako např. SVM či logistická regrese (Goldberg, 2016, s. 345). Ačkoliv se nejedná o nic nového (historie neuronových sítí sahá do 40. let minulého století), jejich častější využívání přišlo až se zpřístupněním výpočetního výkonu a vyřešením některých algoritmických problémů (Jiaconda, 2016).

Neuronová síť, jak sám název napovídá, je výpočetní systém inspirovaný lidským mozkem a jeho způsobem učení se (Dormehl, 2019). Základní výpočetní jednotkou sítě je neuron (viz obrázek 1), který přijímá libovolné množství vstupů a provede jejich vážený součet (vektor vah je v každém neuronu unikátní), nazývaný též jako „vnitřní potenciál neuronu“. Na výsledek součtu je aplikována nelineární aktivační přenosová funkce, která určí hodnotu výstupu neuronu (Bushaev, 2017). Neurony spolu mohou být propojeny (výstup jednoho neuronu je vstupem do jiného), čímž vznikají vážená spojení, která jsou v průběhu učení upravována dle trénovacích dat (Rusell, 1996).

Neurony jsou seskupovány do vrstev (viz obrázek 2), ve kterých sdílí základní vlastnosti, jako např. typ aktivační funkce. Každá síť má vstupní a výstupní vrstvy a dále libovolné množství vrstev mezi nimi, nazývané skryté (jestliže je jich více než jedna, hovoří se o hluboké neuronové síti, anglicky „deep neural network“). Počet

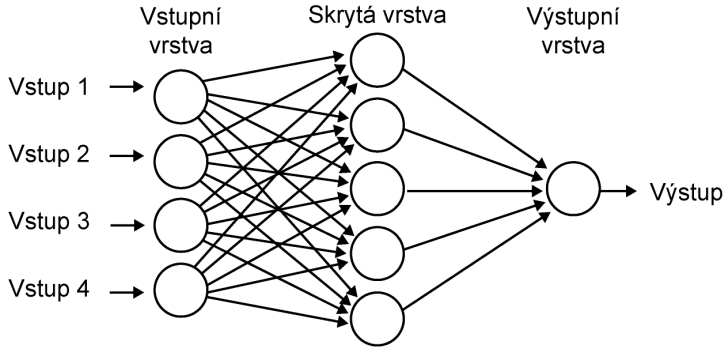
[4]Velikost jednotlivých částí se může lišit v závislosti na dostupných datech a typu modelu (Shah, 2017). Často se využívá poměr trénovacích, validačních a testovacích dat 60:20:20 až 80:10:10.

[5]Někdy se místo validační části využívá technika nazývaná křížová validace (anglicky „Cross--validation“), která rozdělí trénovací data na několik podmnožin, kde jedna podmnožina je vždy validační a zbylé trénovací. Tento proces se několikrát opakuje, pokaždé s odlišným výběrem trénovací a validační části.



Obrázek 1: Základní výpočetní jednotka neuronové sítě (Matematická biologie, c2019b)

vrstev a neuronů, jejich vlastnosti a propojení tvoří architekturu sítě (Bushaev, 2017).



Obrázek 2: Uspořádání neuronů do vrstev (Matematická biologie, c2019a)

Vlastnostem sítě, kam spadá i její architektura, které lze před trénováním upravovat, se říká hyper-parametry (Prabhu, 2018). Jejich nastavení může značně ovlivnit výslednou kvalitu modelu, často je však nutné najít nejlepší nastavení metodou „pokus-omyl“[6]. Jedná se zejména o:

- **počet iterací** („epochs“) – iterace značí jeden průchod všemi trénovacími daty,
- **velikost dávky** („batch size“) – určuje, po jak velkých částech trénovacích dat budou upravovány parametry sítě v rámci jedné iterace,

[6]Metoda „pokus-omyl“ není jedinou, kterou lze zjistit optimální nastavení hyper-parametrů. Lze to provést sofistikovanějšími postupy, jako např. použitím genetického algoritmu, jehož jedinci reprezentují specifické nastavení hyper-parametrů sítě (Osipenko, 2018).

- **ztrátová funkce** („loss function“) – její způsob výpočtu určuje, jakým způsobem bude vypočtena chyba mezi správným výsledkem a výsledkem, jež dodala síť,
- **optimalizační algoritmus** („optimization algorithm“) – jeho typ určuje, jak bude prováděna změna parametrů sítě,
- **velikost parametru učení** („learning rate“) – konstanta určující dopad na velikost změny vah neuronů,
- **pravděpodobnost náhodného vypnutí neuronu** („dropout probability“) – pravděpodobnost, že bude neuron náhodně vypnut.

Před samotným trénováním sítě jsou váhy jednotlivých spojů většinou nastaveny náhodně[7]. Trénování následně probíhá tak, že je na vstupní vrstvu poslána část (dle definované velikosti dávky) trénovacích dat, která projde jednotlivými neurony prostřednictvím definovaných spojení. Průchod některými neurony může být záměrně zamezen (dle definované pravděpodobnosti náhodného vypnutí neuronu) pro dosažení lepší generalizace modelu (Goldberg, 2016, s. 379). Konečné hodnoty na výstupní vrstvě udávají výsledek neuronové sítě, na jehož základě je spočítána hodnota ztrátové funkce. Cílem sítě je minimalizovat chybu vyjádřenou ztrátovou funkcí, což v této fázi lze dokázat jedině změnou vah neuronů (neboli úpravou parametrů sítě). Podle definovaného optimalizačního algoritmu je provedena změna vah, což by mělo způsobit snížení ztrátové funkce při dalším běhu stejné části trénovacích dat (Goldberg, 2016, s. 369). U některých optimalizačních algoritmů lze nastavovat parametr učení, který mění velikost úprav vah, což může mít (v případě vhodně zvolené hodnoty) za následek lepší konvergenci ztrátové funkce (Zulkifli, 2018). V kostce lze tedy shrnout proces tvorby neuronové sítě (Brownlee, 2016a) postupně jako definici hyper-parametrů sítě, implementaci ve zvoleném programovacím jazyce, načtení trénovacích dat a trénování sítě, evaluaci modelu pomocí testovacích dat a jeho využívání (např. v podobě nasazení do reálného prostředí).

Existuje mnoho typů neuronových sítí, každá z nich je vhodná pro jiné typy úloh. Mezi ty nejznámější, a v NLP úlohách zároveň nejpoužívanější, patří dopředné (anglicky „feed-forward“) a rekurentní neuronové sítě. V dopředných sítích mohou informace získané z trénovacích dat (vstupů) putovat pouze dopředu k výstupní vrstvě (Dormehl, 2019). Oproti tomu rekurentní sítě umožňují i zpětný směr.

Dopředné sítě jsou většinou realizovány jako vícevrstvé („multilayer“) perceptrony (MLP), které mají tři a více vrstev (Davydova, 2017a). Pomocí nich je možné reprezentovat jakoukoliv funkci, mohlo by se tedy zdát zbytečné používat jiné typy. Kromě teoretických možností je třeba zvážit také naučitelnost takové funkce. V praxi bývá mnohdy k dispozici velmi omezené množství trénovacích dat

[7]Možností, jakým způsobem bude náhoda vypočítána, je mnoho. Rovněž lze přiřadit všem vahám nuly namísto náhodných veličin (Yadav, 2018). Právě rozdílné váhy na počátku trénování jsou jedním z činitelů stochastického chování neuronové sítě. Nelze tedy očekávat při stejném nastavení naprosto stejné výsledky (Brownlee, 2016b).

a je takřka nemožné vytrénovat kvalitní model na síti typu MLP, tudíž je vhodné hledat komplexnější neuronové sítě, které by to mohly dokázat (Goldberg, 2016, s. 358). I přesto mají tyto sítě v NLP úlohách své uplatnění, např. ve strojovém překladu či rozpoznávání řeči (Davydova, 2017a).

V NLP úlohách je často žádoucí mít na vstupu neuronové sítě celou sekvenci slov (např. celou větu) a ideálně také rozlišovat lokální pořadí slov. MLP síť si sice dokáže poradit s celou sekvencí, nedokáže však již zachytit pořadí. Tyto podmínky však umějí splnit konvoluční (další možná realizace dopředných sítí) či rekurentní neuronové sítě. Konvoluční oproti rekurentním bohužel dokáží využít informace obsažené v pořadí slov jen velmi omezeně (Goldberg, 2016, s. 389), a jsou tedy oblíbené spíše v rozpoznávání obrázků, ačkoliv i některé NLP úlohy mohou být poměrně úspěšně tímto typem realizovány (Yoon, 2014).

Rekurentní neuronové sítě jsou v poslední době v řešení NLP úloh velmi využívané. Důvodem je jejich schopnost predikce postupným průchodem jednotlivých elementů libovolně dlouhé sekvence. Má několik variant. Její základní podoba obsahuje spojení mezi neurony, které tvoří orientované cykly. To znamená, že výstup neuronové sítě závisí nejen na konkrétním vstupu, ale také na předešlých stavech sítě, což je jistá forma paměti (Davydova, 2017a). Matematicky lze rekurentní síť vyjádřit takto funkcí RNN (Goldberg, 2016, s. 390):

$$RNN(s_0, x_{1:n}) = s_{1:n}, y_{1:n}$$

$$\begin{aligned} s_i &= R(s_{i-1}, x_i) \\ y_i &= O(s_i) \end{aligned}$$

Proměnná s_i značí stav v kroku i, $x_{1:n}$ značí kompletní vstupní sekvenci o n elementech. Funkce R je definovaná rekursivně, jejími argumenty jsou předešlý stav s_{i-1} a vstupní element x_i, výstupem pak nový stav s_i ve stávajícím kroku i. Funkce O mapuje stav s_i na výstup y_i.

Tyto stavební bloky lze kombinovat a navzájem spojovat, čímž vzniknou různé varianty rekurentních sítí. Lze je stavět do vrstev, čímž vznikne vícevrstvá (hluboká) rekurentní síť (Goldberg, 2016, s. 395). Velmi populární variantou je obousměrná rekurentní síť, ve které je predikce v určitém kroku založena jak na předešlých vstupech, tak na budoucích. To je pro NLP úlohy velmi žádané, neboť v jazyce je často užitečné znát levý i pravý kontext slov.

Konkrétní definice funkcí R a O určují specifické architektury rekurentní sítě. Může se jednat např. o LSTM („Long Short-Term Memory“) síť, která pracuje se speciálními strukturami, nazývanými brány („gates“), které ovlivňují toky informací v síti, např. zapomínáním, či naopak pamatováním si určité informace (Skansi, 2018, s. 143). Výpočetně méně náročnou alternativou je architektura GRU („Gated Recurrent Unit“). Obsahuje sice méně brán, ale její výkonnost je přesto podobná (Goldberg, 2016, s. 401).

2.3 Vektorová reprezentace slov

S příchodem strojového učení a jeho nasazením na dolování znalostí z textových dat bylo již od počátku základní otázkou, jak zakódovat jednotlivé elementy textu – tokeny (slova či fráze) – do struktur, se kterými si umějí techniky strojového učení, potažmo výpočetní jednotky, poradit. Ty pracují pouze s čísly, nikoliv textem. Prvotní techniky přiřazovaly tokenům unikátní číselný identifikátor, což sice funguje, ale o daném slově si z této reprezentace systém nedokáže vzít žádnou další informaci, jako například použitý kontext (Banerjee, 2018). Tento nedostatek motivoval pro vytvoření technik, které by jej mohly alespoň z části odstraňovat – techniky vnoření slov (anglicky „word embeddings“).

Formálně řečeno, vnoření slov je souhrnné označení technik používaných v počítačovém zpracování přirozeného jazyka, které každému tokenu z množiny všech tokenů[8] daného slovníku vytváří vektor, jehož dimenze (či složky) mají přiřazenu hodnotu z množiny reálných čísel. Počet dimenzí vektoru udává v geometrické reprezentaci rozměr prostoru. Matematicky lze tuto definici vyjádřit jako zobrazení $\varphi : w \mapsto R^d$, kde w jsou unikátní tokeny korpusu a d je počet dimenzí vektoru.

První z technik byl tzv. „kód 1-z-n“ (anglicky „one-hot encoding“), který každému tokenu přiřazuje unikátní vektor, ve kterém jsou všechny dimenze nulové, kromě jedné, která odpovídá danému slovu (dimenze tedy zastupuje index tokenu). Počet dimenzí je tedy roven počtu unikátních tokenů (Brownlee, 2017b). Vektor je tudíž velmi řídký, rozsáhlý a tím i paměťově náročný, nicméně splňuje definici.

Z této techniky vychází metoda multimnožiny slov (anglicky „bag-of-words“). Ta dokáže zachytit společný výskyt jednotlivých slov v určitém dokumentu sečtením jejich „one-hot“ vektorů, čímž vznikne (binární) vektor, reprezentující výskyt slov v dokumentu ze všech slov ve slovníku. Modifikací pak může být místo pouhého (binárního) výskytu zachycovat také četnost slova v dokumentu.

Tyto techniky však zanedbávají význam jednotlivých slov v kontextu a jejich uspořádání, což znemožňuje zachytit hlubší informace o slovech z korpusu. Existuje však teze, jež se nazývá „distribuční hypotéza“, která říká, že *slova, která se nacházejí ve stejném kontextu, mají podobný význam*[9] (Harris, 1954). Techniky pro vytváření vektorových reprezentací slov, které chtějí tuto vlastnost splňovat, se tedy nutně musejí zabývat kontextem pro zachycení hlubších informací z korpusu.

Moderní modely vnoření slov se touto hypotézou inspirovaly a vytvořily se techniky, založené na strojovém učení bez učitele, jejichž vektorové reprezentace slov dokáží reflektovat informaci o používaném kontextu slova v daném trénovacím korpusu (za předpokladu, že je dostatečně rozsáhlý). Zjistilo se, že pozorováním podobnos-

[8] Tokeny, které nejsou přítomny v trénovacím korpusu, nemají v daném modelu přiřazenou reprezentaci. Anglicky se jim říká tzv. „out-of-vocabulary“ (OOV) words, neboli slova mimo slovník a lze s nimi nakládat různými způsoby (např. přiřadit jim náhodný či nulový vektor).

[9] Velmi populární věta distribuční hypotézy zní: „*význam slova je dán slovy, která se s ním často pojí*“, anglicky „*word is characterized by the company it keeps*“ (Firth, 1957).

tí či vzdáleností[10] vektorů v prostoru lze rozkrýt velmi komplikované sémantické či syntaktické vztahy[11] mezi slovy, jako např. antonyma, jednotná a množná čísla, gradace přídavných jmen a další (Svoboda a Brychcín, 2018).

Tyto zakódované vztahy mezi slovy mají široké uplatnění v různých úlohách NLP a v dnešní době se v nich považují za stavební kameny. Kvalitní vektorové reprezentace dokáží například při správném použití vylepšit analýzu sentimentu, sumarizaci dokumentů či systém doporučení (Gupta, 2019). Nedílnou součástí je již také v mnoha systémech NER, čehož se využívá i v této práci.

Kvalitu vektorů (ve smyslu kvality distribuční hypotézy) lze evaluovat několika způsoby. Jedním z nich je evaluace korpusem analogií, který obsahuje čtyři slova w_a, w_b, w_c a w_d. Ty tvoří proměnné v testované hypotéze „*Slovo* w_a *je podobné slovu* w_b *jako je slovo* w_c *podobné slovu* w_d“ (Svoboda a Brychcín, 2018, s. 4). Pravdivost této hypotézy lze ověřit algebraickou operací vektorů reprezentující tato slova:

$$vektor(w_a) + vektor(w_b) - vektor(w_c) = x$$

Pokud je výsledný vektor x dostatečně podobný vektoru slova w_d, je test úspěšný. Příkladem budiž tato analogie: „maximum“ je podobné slovu „minimum“ jako je slovo „export“ podobné ke slovu „import“. Podobnými sadami lze testovat různé sémantické a syntaktické zákonitosti slov (Mikolov a kol., 2013b).

Další využívanou metodou evaluace je podobnost dvou slov. V testovací sadě jsou vytvořeny různé dvojice slov a člověkem přiřazena vektorová podobnost, která se poté testuje na vytvořených vektorových reprezentacích. Tato metoda je však velmi náchylná na subjektivní vnímání slov (Faruqui a kol., 2016, s. 31).

2.3.1 Typy moderních přístupů

Každou z technik pro vytváření vektorových reprezentací slov lze zařadit do jedné ze dvou metod, které určují způsob vytváření vektorů (Banerjee, 2018):

- metoda založená na globálních statistikách korpusu,
- metoda predikce kontextu.

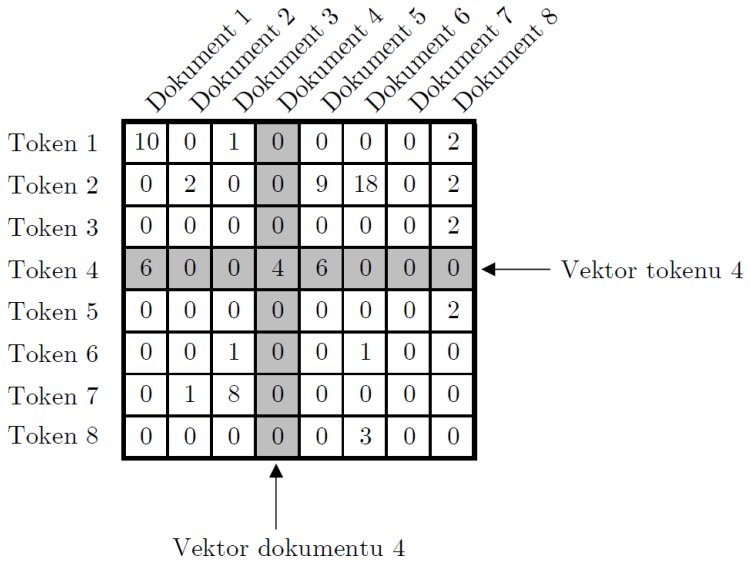
První ze zmíněných metod využívá globální statistiky korpusu jako jediný zdroj informací pro vytváření vektorového prostoru slov. Jednotlivé varianty metody pak se liší ve využití těchto informací. Jedna z nich využívá již zmíněnou techniku „bag-of-words“, která každému dokumentu z korpusu přiřadí dle výskytu slov unikátní

[10] Podobnost či vzdálenost vektorů lze v prostoru měřit několika technikami. Často používanou je kosinová vzdálenost, která určuje kosinus úhlu mezi dvěma vektory – čím podobnější z hlediska orientace jsou, tím menší úhel mezi nimi bude, a tedy kosinus nabude vyšší hodnoty (Perone, 2013).

[11] Sémantická podobnost zachycuje podobnost ve významech slov – např. vztah mezi státem a jeho hlavním městem. Oproti tomu syntaktická podobnost zachycuje vztah ve výstavbě slov či morfologickou strukturu slov.

vektor. Tyto vektory poté tvoří sloupce matice a vektorová reprezentace jednotlivých tokenů je zachycena v řádcích, jak ukazuje obrázek 3. Jistým vylepšením této techniky může být vážení dimenzí vektorů metrikou tf-idf (Analytics Vidhya, 2017).

Avšak nejvýznamnější variantou metody je technika, která je založená na výpočtu spolu-výskytu jednotlivých slov. Pro všechny možné páry slov je spočtena frekvence jejich společného výskytu ve stejném kontextu v rámci celého korpusu, čímž vzniká matice o velikosti $V \times V$, kde V je počet unikátních tokenů v korpusu. Vektory slov se poté získají zmenšením dimenzí matice, tzv. faktorizací, např. metodou SVD či PCA (Sharma, 2018).



Obrázek 3: Matice vektorových reprezentací dokumentů a tokenů (Analytics Vidhya, 2017)

Všechny varianty první metody jsou deterministické, tedy pro určitý korpus budou vektorové reprezentace vygenerovány vždy stejně. Bohužel žádné z nich se nedaří kvalitně zachytit výše popsanou myšlenku distribuční hypotézy. Tu však umějí zachytit varianty druhé metody. Ty predikují pravděpodobnosti výskytu slov v určitém kontextu, k čemuž využívají různé neuronové sítě (jednotlivé varianty se liší de facto jen ve zvolené architektuře sítě). Známým kandidátem této kategorie je technika zvaná Word2Vec, která bude detailněji popsána v sekci 2.3.3, kde bude také představen detailnější vhled do metody predikce kontextu. Budou také představeny další „state-of-the-art“ techniky pro tvorbu vektorových reprezentací slov.

U všech moderních přístupů lze nastavovat stejné parametry trénování, jako je dimenze vektorů, nastavení kontextu slov (velikost a směr), optimalizace velikosti slovníku (maximální počet unikátních slov, minimální počet výskytů slova pro zařazení) či počet průchodů (iterací či epoch) slovníkem. Tyto základní parametry mohou výrazně ovlivnit výslednou kvalitu modelů (Levy, Goldberg a Dagan, 2015, s. 217). U většiny technik lze pak navíc nastavovat jejich specifické parametry.

2.3.2 Problémy a omezení

Jakkoliv dokonale moderní techniky vnoření slov zní, mají také svá omezení a problémy, které mohou znesnadnit zachycení vztahů mezi slovy a jejich zakódování do vektorů. Většina problémů však nepramení z technik samotných, ale z použitých korpusů (neboli trénovacích dat) a také ze samotné podstaty jazyků.

Pro věrohodné zachycení vztahů a podobností slov je nutné použít robustní korpus nejen z hlediska velikosti, ale i kvality. Obecně zde platí pravidlo „čím více, tím lépe“ (nejméně však desítky milionů tokenů). Kromě toho je však také nutné vyhodnotit kvalitu textu, jako je například počet unikátních slov, množství překlepů, styl psaní apod. Může se stát, že menší, za to kvalitnější korpus, přinese větší přidanou hodnotu nežli rozsáhlý, ale nekvalitní. Některé techniky předzpracování textu mohou trénovacím datům výrazně pomoci, ale není to řešením na všechny problémy s tímto spojené. Pokud nejsou zachyceny určité vztahy slov v korpusu (např. hlavní město – stát), rovněž je nelze očekávat zakódované ve vektorech.

Nutno podotknout, že slova mají jiný význam v různých doménách. Např. slovo „střílet“ může být používáno ve zcela jiném kontextu ve sportu a v kriminální žurnalistice. Pokud je model vnoření slov využíván v rámci jiné NLP úlohy, je vhodné využít pro trénování vektorových reprezentací korpus, který je stejně doménově zaměřený, jako jsou data pro NLP úlohu. Např. v NER může být použitá doména klíčová a hrát významnou roli ve kvalitě rozpoznávání, neboť některá slova mohou být v různých doménách jinými typy entit (Kulkarni, Mehdad a Chevalier, 2016).

Omezení a problémy spojené s jazykem mohou také hrát nezanedbatelnou roli v kvalitě zachycení vztahů slov. Ve vektorovém prostoru má každé slovo pouze jednu unikátní reprezentaci. Tudíž např. pro slovo mající více významů (např. polysémie či homonymie[12]) nelze najít věrohodný vektor, zachycující všechny jeho významy. Existují sice techniky, které dokáží z vektorových reprezentací takových slov vyřadit jeden z významů (Schmidt, 2015), otázkou však je, do jaké míry jsou automatizovatelné a robustní.

Pro morfologicky bohaté jazyky může být obtížnější získat pro všechna jeho slova kvalitní vektory. Důvodem je vysoký počet různých morfologických obměn slov, čímž se rapidně zvyšuje unikátní počet slov. Tím se zároveň pro každé takové slovo snižuje počet jejich využití v korpusu, což může mít negativní dopad (zejména v menších korpusech) na kvalitu vektorů. Pomoci mohou techniky předzpracování, jako např. lemmatizace, ovšem ty vypouští z textu určité informace o slově, jako například pád či slovesný čas. Je pak otázkou, zdali přínos z těchto technik (tedy např. snížení počtu unikátních slov) předčí pokles informačních schopností slov v korpusu.

[12]Polysémie označuje slova, která stejně zní a mají také genetickou souvislost, homonymie oproti ní vypouští druhý z předpokladů (Hladká, c2012–2018).

2.3.3 Word2Vec

První ze zmíněných technik pro tvorbu vektorových reprezentací slov je v NLP světě velice známá technika Word2Vec. Jejím autorem je český vědec Tomáš Mikolov, který ji vynalezl za dobu svého působení v Google (Mikolov a kol., 2013b). Lze bez nadsázky tvrdit, že se jednalo o průlomový objev – nejenže natrénované modely pomocí Word2Vec dosahovaly nejlepších výsledků na testech podobností slov, ale také se enormně urychlila doba vytváření vektorových prostorů na obrovských korpusech z měsíců na hodiny. Jedním z dalších zásadních objevů s příchodem této techniky je také odhalení možnosti provádět intuitivní algebraické operace s naučenými vektory (Mikolov, Yih a Zweig, 2013, s. 1). Velmi známá rovnice spojená s touto problematikou vystihuje tento objev výstižně:

$$vektor(\text{„}král\text{“}) - vektor(\text{„}muž\text{“}) + vektor(\text{„}žena\text{“}) \approx vektor(\text{„}královna\text{“})$$

Po odečtení vektoru reprezentujícího slovo „muž“ od vektoru slova „král“ a následným přičtením vektoru slova „žena“ je výsledkem vektor velmi blízký vektoru slova „královna“. Lze také vypozorovat, že sémanticky či syntakticky podobná slova jsou ve vektorovém prostoru blíže sebe. Díky všem těmto vlastnostem lze vektorové prostory využívat v mnoha problémech spojených s dolováním znalostí z textu.

Předešlé pokusy pro vytvoření vektorové reprezentace využívaly neuronové sítě s více skrytými vrstvami pro odhalení nelineárních vazeb mezi slovy. Tyto architektury však znemožňovaly trénování vektorových reprezentací na rozsáhlých korpusech z důvodu jejich výpočetní složitosti. Word2Vec model se tento problém pokusil vyřešit tak, že zjednodušil architekturu neuronové sítě (Mikolov a kol., 2013b, s. 4). Ta má pouze tři vrstvy – vstupní, skrytou (někdy též nazývanou projekční) a výstupní. Počet neuronů ve vstupní a výstupní vrstvě je roven velikosti slovníku[13], ve skryté vrstvě je počet neuronů roven uživatelem definovanému počtu dimenzí vektorů.

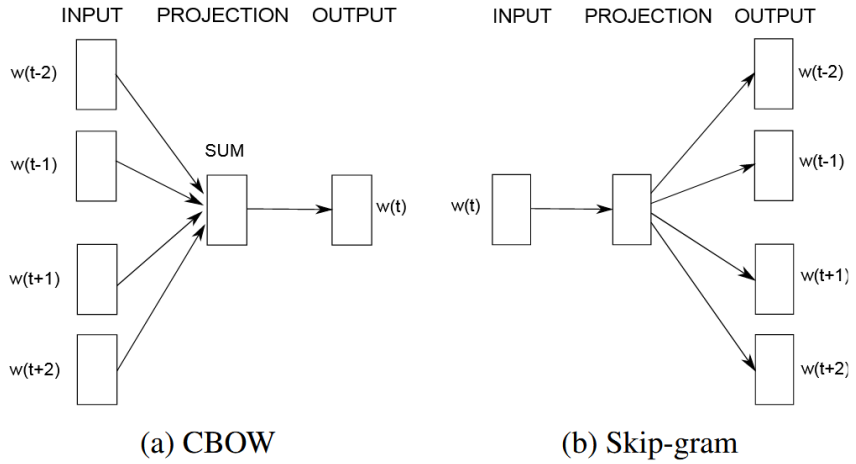
Neuronové sítě (resp. modely) lze trénovat dvěma různými způsoby – Continuous Bag-of-Words (CBOW) a Skip-Gram. Rozdíl v přístupech ilustruje obrázek 4. Oba mají za úkol predikovat na základě vstupních dat výstupní data dle dodaného kontextu[14], čímž se řadí do skupiny prediktivních metod pro trénování vnoření slov. Ovšem oproti typickému využití sítí, kde cílem je, aby dokázala plnit daný úkol i na neznámých datech a kde trénovací proces slouží pouze jako prostředek pro optimalizaci parametrů, zde je cílem samotné trénování sítě. V průběhu se postupně podle vstupních dat upravují váhy neuronů skryté vrstvy. Ty tvoří matici vah (tzv. „weight matrix“) $V \times N$, kde V je velikost slovníku a N je počet dimenzí ve vektorovém prostoru. Po skončení trénování vyjadřují jednotlivé řádky této matice[15]

[13] Každé slovo se zakóduje již zmíněným unikátním „one-hot“ vektorem.

[14] Velikost kontextového okna může uživatel ovlivnit v parametrech testování.

[15] Ve skutečnosti má skrytá vrstva dvě matice vah. Jednu s váhami mezi vstupní a skrytou vrstvou, druhou pak mezi skrytou vrstvou a výstupní. Slovo má pak dvě různé vektorové reprezentace – jednu pro slovo a jednu pro jeho kontext, přičemž pro finální reprezentaci se využívá vektor slova (Šůstek, 2017).

vektory daných slov (McCormick, 2016). Počet průchodů korpusem, a tedy počet úprav vah neuronů skryté vrstvy, může uživatel ovlivnit v parametrech trénování – počtem epoch.



Obrázek 4: Rozdílné přístupy CBOW a Skip-Gram (Mikolov a kol., 2013b, s. 5)

V přístupu CBOW dostává neuronová síť vstupní tokeny, které reprezentují malý kontext okolo daného slova z korpusu. Výsledkem neuronové sítě je predikce slova, které by se v daném kontextu mělo vyskytovat. Vzhledem k tomu, že neuronová síť kromě kontextu zná i predikované slovo, dokáže si spočítat ztrátovou funkci a dle toho patřičně upravit váhy neuronů.

Přístup Skip-Gram pracuje na podobném principu jako CBOW. Nicméně namísto predikce slova na základě dodaného kontextu se snaží odhadnout kontext na základě dodaného slova. Vstupem neuronové sítě je tedy dané slovo z trénovacího korpusu a neuronová síť se pro něj snaží predikovat slova, která tvoří kontext před i za vstupním slovem. Opět má síť k dispozici jak vstupní slovo, tak i kontext, dokáže tedy vypočítat ztrátovou funkci a upravovat ve fázi tréninku váhy neuronů.

Skip-gram oproti CBOW v sobě nezahrnuje složité maticové výpočty, což dělá trénování daleko rychlejším a efektivnějším (Mikolov a kol., 2013a). V praxi na anglickém textu bylo dokázáno, že Skip-Gram modely lépe zachycují sémantickou podobnost a CBOW naopak syntaktickou (Mikolov a kol., 2013b). Je tedy vhodné zvolit (ať už formou experimentů či znalostí) přístup, který vyhovuje dané NLP úloze více.

Pro Skip-Gram model existují techniky, které trénování ještě více zrychlují. Může se jednat o techniku „Sub-sampling“, která pro každé slovo z trénovací množiny vypočte pravděpodobnost výmazu (ta je vyšší u frekventovanějších slov). Podstatou této techniky je fakt, že velmi frekventovaná slova nemají v kontextu velký význam. Tato technika nejenže zrychluje proces trénování (neboť redukuje trénovací množinu), ale také zlepšuje přesnost naučených vektorů ojedinělých slov, protože pracuje

s kontextem, který obsahuje relevantnější slova, čímž mezi nimi může objevit nové vztahy (Levy, Goldberg a Dagan, 2015, s. 215).

Dalším zefektivněním trénování Skip-Gram modelu může být změna architektury poslední vrstvy neuronové sítě. Výchozí je založena na funkci „full-softmax“, která je výpočetně velmi náročná[16], neboť pro každý běh počítá sumu vektorů všech slov, kterých může být velmi mnoho (typicky 10^5–10^7). Modifikace této funkce na tzv. „hierarchický softmax“ může vést ke snížení časové složitosti trénování na $O(log_2 V)$ bez signifikantní ztráty kvality vektorů (Mikolov a kol., 2013a). Hlavní myšlenkou této modifikace je použití binárního stromu, jehož listy reprezentují pravděpodobnosti jednotlivých slov a uzly kumulativní sumu pravděpodobností slov na cestě (Olejnik, 2017).

Další alternativou změny architektury je nahrazení „softmax“ funkce za metodu zvanou „Negative Sampling“. Ta vytváří negativní trénovací vzorky (v počtu většinou mezi 2–20) pro každé vstupní slovo tak, že do výstupního kontextu vybere místo vstupního slova náhodně jiné, čímž se výstup považuje za negativní vzorek pro trénování. Díky této metodě se pro každá trénovací data neupravují váhy všech neuronů (kterých je typicky velmi mnoho), ale jen ta část, která odpovídá trénovacímu vstupu společně s negativními vzorky.

Jistou nevýhodou Word2Vec modelů je fakt, že pro fráze (tj. víceslovní spojení) neexistuje unikátní vektor, který by reprezentoval jejich začlenění do prostoru. Např. pro frázi „Česká republika“ je v základním modelu oběma jednotlivým slovům „Česká“ a „republika“ přiřazen unikátní vektor a žádná algebraická operace nepovede k adekvátnímu vektorovému vyjádření, které by pro frázi neslo přidanou informační hodnotu. Existují však techniky, které se s tímto problémem dokáží vypořádat. Jednou z nich je tzv. „model založený na frázích“ (Mikolov a kol., 2013a), který v textu nalezne frekventovaná slovní spojení. Ty následně převede na jeden token (v použitém příkladu by se jednalo o token „Česká_republika“), se kterým dokáže nakládat stejně tak, jako s jednotlivými slovy. Díky tomu se pro ně vytvoří vektorová reprezentace, která již dokáže nést užitečnou informační hodnotu. Každý běh tohoto nástroje vytvoří fráze o jedno slovo delší, je tedy nutné pro získání víceslovných frází nástroj pustit opakovaně (McCormick, 2017).

2.3.4 GloVe

GloVe, neboli „Global Vectors“, je akademickým, volně dostupným dílem vědců Jeffreyho Penningtona, Richarda Sochera a Christophera D. Manninga (2014). Většina existujících přístupů pro vytvoření vektorových reprezentací slov je založena na jedné ze dvou výše popsaných metod – globálních statistikách korpusu či predikci kontextu. Autoři se domnívají, že obě metody trpí určitými nedostatky.

Tvrdí, že metoda založená na globálních statistikách sice efektivně zachycuje statistické informace korpusu díky globální analýze společného výskytu jednotlivých

[16] Časová složitost „full-softmaxu“ je $O(V)$, kde V je velikost slovníku, tedy počet unikátních tokenů (Olejnik, 2017).

slov, ale již nedokáže věrohodně reprezentovat lokální závislosti slov v kontextu. Naopak metoda predikce kontextu umí kvalitně zachytit lokální závislosti slov (jako např. lineární vztah mezi určitými vektory slov), což ukazují vysoké výsledky evaluací pomocí korpusu analogií (Pennington, Soche a Manning, 2014, s. 6), ale globální statistiky datové množiny jsou zcela ignorovány, neboť metoda prohledává postupně jednotlivé kontexty napříč celým korpusem a nebere jej jako jeden celek.

GloVe se snaží minimalizovat zmíněné nedostatky tím, že využívá nejlepší vlastnosti obou zmíněných metod. Jádro GloVe spočívá v sestrojení tzv. „matice spolu-výskytu“ („co-occurrence matrix“) slov (Keitakurita, 2018). Její řádky tvoří slova a sloupce kontexty[17]. V jednotlivých buňkách je zachycena frekvence X_{ij} výskytu slova i v kontextu s jiným slovem j. Z nich lze vypočítat pravděpodobnosti $P(j|i) = X_{ij}/X_i$ (kde $X_i = \sum_k X_{ik}$ vyjadřuje frekvenci výskytu všech slov v kontextu se slovem i) výskytu určitého slova j v kontextu se slovem i v celém korpusu. U slov, která se vyskytují častěji společně, bude pravděpodobnost vyšší a naopak (Sciforce, 2018).

Otázkou každé techniky pak je, jak na základě těchto informací vytvoří vektorové reprezentace slov. V případě GloVe počátečním bodem pro výpočet vektorových reprezentací nejsou tyto pravděpodobnosti samotné, nýbrž poměry pravděpodobností $P(i|k)/P(j|k)$ výskytů dvou různých slov i a j ve stejném kontextu (neboli se slovem) k, které mohou být úzce spjaty s významem těchto slov (Keitakurita, 2018). Myšlenka je relativně jednoduchá – vytvořit funkci, která se bude snažit predikovat tyto již známé poměry pravděpodobností za pomocí vektorových reprezentací (Pennington, Soche a Manning, 2014, s. 3):

$$F(w_i, w_j, \tilde{w_k}) = \frac{P(i|k)}{P(j|k)}$$

Jak již bylo zmíněno, GloVe využívá to nejlepší z obou metod pro tvorbu vektorových reprezentací – přednosti první metody byly využívány doposud, přednosti druhé uplatňuje v konstrukci funkce F. V ní se snaží zachytit lineární závislosti mezi slovy tak, že se uplatní rozdíl vektorových reprezentací slov w_i a w_j (čímž je právě zachycena jednoduchá, přesto mocná, aritmetika vektorů) a následně se provede skalární součin vypočteného rozdílu a vektorové reprezentace slova $\tilde{w_k}$. Po několika matematických úpravách této formule (Pennington, Soche a Manning, 2014, s. 3–4) lze dospět k funkci J, založené na vážené metodě nejmenších čtverců. Tu se GloVe snaží minimalizovat, čímž po určitém počtu kroků (pomocí metody klesání podle gradientů) dospěje k optimálním vektorovým reprezentacím slov:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b_j} - logX_{ij})^2$$

Proměnná V značí velikost slovníku, konstanty b_i a $\tilde{b_j}$ pro zachování symetrie a $f(X_{ij})$ je vyvažovací funkce, která z výpočtu vyřazuje všechna slova i, která nejsou

[17]Kontext je zde definován jako slovo, které tvoří okolí zkoumaného slova. Okolím jsou slova ležící do vzdálenosti n zleva, zprava, či do obou směrů. Hodnotu n i směr okolí definuje uživatel.

v kontextu se slovem j a upřednostňuje méně frekventovaná slova před častými. Na počátku výpočtu jsou vektory slov zvoleny náhodně.

Autoři tvrdí, že GloVe je vhodnou technikou pro řadu NLP úkolů, jako je např. NER, což také dokazují v provedených experimentech (Pennington, Soche a Manning, 2014, s. 9). Jejich tvrzení navíc umocňují provedené evaluace sémantické a syntaktické podobnosti slov, ve kterých GloVe dosahoval zdaleka nejvyšších výsledků (Pennington, Soche a Manning, 2014, s. 6) v porovnání s jinými metodami (např. CBOW či Skip-Gram techniky Word2Vec), bez ohledu na použitém trénovacím korpusu či počtu dimenzí vektoru. V porovnání s Word2Vec by měl být GloVe také výpočetně rychlejší.

Jako u ostatních technik pro vytváření vektorové reprezentace slov, i u GloVe lze nastavovat několik parametrů trénování, které mohou ovlivnit výslednou kvalitu modelu. Kromě již zmíněných základních parametrů lze nově také optimalizovat zmíněnou vyvažovací funkci $f(X_{ij})$, jejíž hodnoty jsou ve výchozím stavu nastaveny dle heuristiky.

2.3.5 FastText

Technika FastText byla představena tři roky po Word2Vec vědci z výzkumné skupiny Facebook AI Research (Grave, 2016). Jejím spoluautorem je již jednou zmiňovaný Tomáš Mikolov. Není tedy divu, že FastText velmi úzce navazuje na Word2Vec přístup. Dalo by se tvrdit, že se jedná spíše o rozšíření nežli konkurenční produkt, ačkoliv obě techniky jsou pod záštitou jiných společností.

I v tomto případě se využívá mnohodimenzionální prostor, ve kterém je kromě každého slova přiřazena hodnota také každému n-gramu z trénovací množiny dat.

Jde de facto o změnu původní ztrátové funkce neuronové sítě Word2Vec modelu, která brala v potaz pouze celá slova. Nyní je každé slovo obohaceno o speciální hraniční symboly „<“ a „>“, díky kterým lze rozlišovat prefixy, infixy a sufixy slov. Např. pro slovo „místo“ se vytvoří speciální slovo „<místo>“, které lze rozdělit do množiny n-gramů, např. pro $n = 3$ (tedy trigramy) by taková množina vypadala následovně: {<mí, mís, íst, sto, to>}. Tato množina a celé speciální slovo „<místo>“ pak slouží jako vstup neuronové síti. Velikost n je uživatelsky volitelná a může být také zadána intervalem – v takovém případě jsou v množině pro dané slovo zahrnuty všechny možné n-gramy dle definovaného intervalu n.

Díky těmto hraničním znakům lze také rozlišovat různá slova, která mají stejnou část. Např. pro slovo „sto“ se vytvoří speciální slovo „<sto>“, které je jiné, než trigram „sto“ ve slově „místo“.

Vektorová reprezentace slova je pak počítána jako prostá suma vektorů jednotlivých n-gramů. Lze tedy předpokládat, že u morfologicky bohatých jazyků, kde prefixy, infixy a sufixy jsou často součástí slova, budou různé obměny slova mít velmi podobný vypočítaný vektor. V jiných vektorových reprezentacích by každá taková obměna slova měla svůj unikátní vektor, což komplikuje vytváření kvalitní vektorové reprezentace.

Obrovskou výhodou FastTextu je vypořádání se se slovy, která se neobjevila v trénovací množině (tzv.„ out-of-vocabulary“ slova), čímž by pro ně v předešlých přístupech nebyla vypočtena hodnota vektoru. To se řešilo přiřazením náhodného vektoru, ale v tomto přístupu lze prostým průměrem či součtem (dle implementace) n-gramů daného slova dospět k vektorové reprezentaci, která již může nést informaci o sémantickém i syntaktickém významu. Je však zřejmé, že takto nelze získat vektorovou reprezentaci pro jakékoliv slovo – vždy musí ve vektorovém prostoru existovat alespoň jeden n-gram tvořící dané neznámé slovo.

Dle výsledků práce autorů Bojanowského a kol. (2016, s. 5) lze vypozorovat, že tento přístup výrazně zlepšuje výsledky modelů vnoření slov u morfologicky bohatých jazyků. Nutno podotknout, že se zlepší především syntaktická podobnost slov, nikoliv sémantická.

Další velkou výhodou tohoto přístupu je fakt, že stačí řádově méně trénovacích dat pro dosažení velmi dobrých výsledků v porovnání s jinými technikami (Bojanowski a kol., 2016, s. 7). To může být výhodou např. u trénování modelů v datové doméně, ve které je omezené množství prostého textu.

2.3.6 Způsoby využití a tvorby modelů

Existuje několik způsobů, jak získat či vytvořit modely vnoření slov. Autoři každé představené techniky zveřejnili zdrojové kódy, kterými lze modely trénovat. Díky tomu pro vytvoření vlastního modelu stačí sehnat pouze zdrojová data (viz sekce 2.3.7) a využít tyto volně dostupné kódy. Velkou výhodou těchto modelů je fakt, že je lze opětovně využívat v různých úlohách NLP. Díky širokému využívání těchto technik v mnoha oblastech existuje spousta již natrénovaných (tzv. „pre-trained“) modelů pro různé jazyky, které lze stáhnout a ihned využívat. Ty jsou mnohdy trénované na obrovských korpusech (čítající miliardy tokenů), jejichž text bývá bez doménového zaměření.

Pro mnohé NLP úlohy mohou tyto „pre-trained“ modely dostačovat, nicméně pro maximální nárůst kvality je ve většině případů výhodnější vytvořit vlastní model vnoření slov dle dané úlohy (Brownlee, 2017a), zejména v případech, kdy jsou data velmi specifická. V praxi je však běžné, že není k dispozici dostatek těchto specifických dat, ze kterých by šlo natrénovat reprezentativní model vnoření slov. Tomuto problému může pomoci tzv. „Retrofitting“ technika, která namísto tvorby nového modelu vezme již existující model a pozmění jej dle dostupných specifických dat (Gupta, 2019). Výsledkem by měl být model, který lépe odráží význam slov v této specifické doméně.

2.3.7 Zdroje rozsáhlých textů pro trénování modelů

Tato sekce přináší výčet otevřených datových zdrojů, které se hojně využívají v různých pracích pro vytváření moderních modelů vnoření slov, zachycující sémantickou a syntaktickou podobnost slov v číselném mnohodimenzionálním prostoru. Nutno

podotknout, že se nejedná o úplný výčet všech dostupných dat, neboť datovým zdrojem může být teoreticky jakýkoliv smysluplný prostý text většího rozsahu. Přesto pro možnost vytváření užitečné vektorové reprezentace slov je vhodné, aby zdroje splňovaly některé vlastnosti. Jedná se zejména o kvalitu textu (minimum chyb a překlepů) a také robustnost, tj. dostatečné množství textu, ze kterého jdou vyvodit podobnosti slov.

Velmi užitečnými zdroji rozsáhlých textů v českém jazyku jsou Český národní korpus[18] (ČNK) a repositář LINDAT-Clarin[19]. Oba lze bez omezení využívat pro akademické účely. Obsahují spoustu otevřeného textu z různých datových domén, některé z nich obsahují dodatečné anotace, jako jsou slovní druhy či lemmata slov. Významné rozsáhlé texty, které mohou být užitečné pro vytvoření vektorových reprezentací, zachycuje tabulka 1. Obsahuje rovněž informace o celkovém počtu slov a unikátních tokenů v jednotlivých korpusech.

Tabulka 1: Vybrané korpusy otevřeného českého textu z LINDAT-Clarin repositáře

Korpus	**Počet slov**	**Počet unikátních tokenů**
CoNLL-2017 (česká část)	1,62 miliardy	21,5 milionů
CWC-2011 – články	628 milionů	1,8 milionů
CZES	497 milionů	3,5 milionů
EuroParl (česká část)	13 milionů	304 tisíc
Extrakce české Wikipedie	134 milionů	2,6 milionů
SYN-2015	121 milionů	1,4 milionů

Korpus CoNLL-2017 reprezentuje jeden ze zdrojů, které byly k dispozici účastníkům konference CoNLL v roce 2017. Jejím tématem byla vícejazyčná syntaktická analýza z textu velkého rozsahu. Ze zmíněného vyplývá, že má zdroj několik jazykových obměn, včetně češtiny. Kromě anotovaných textů byly pro účely konference vytvořeny také rozsáhlé zdroje psaného textu, ze kterých účastníci vytvářeli modely vnoření slov. Obsah textů byl získán z vícero zdrojů – z automatické extrakce Wikipedie a neziskové organizace Common Crawl, která extrahuje texty z mnoha rozličných zdrojů na internetu (Zeman a kol., 2014, s. 3). Česká verze obsahuje 1,62 miliardy slov, které tvoří přes 21 milionů unikátních tokenů, což je mnohokrát více než u všech ostatních zmíněných korpusů. Vysoký počet unikátních tokenů značí velké množství chyb či překlepů, které vytvářejí další unikání tokeny, které však mohou způsobit řadu komplikací. Nicméně díky velikosti korpusu by i přesto mělo být možné vytvořit kvalitní vektorové reprezentace pro slova bez překlepů.

CWC-2011 je český korpus čítající přes 2,5 miliardy slov. Text byl získán vytěžením vybraných webů z českého internetu a skládá se ze tří částí – novinových článků, blogů a diskusí (Spoustová a Spousta, 2012, s. 311). První část obsahuje přes

[18]Český národní korpus je dostupný na adrese `https://ucnk.ff.cuni.cz/cs/`.
[19]Repositář LINDAT-Clarin je dostupný na adrese `https://lindat.mff.cuni.cz/en/`.

600 miliónů slov a pro tvorbu kvalitní vektorové reprezentace se zdá být nejvhodnější, neboť by se ve většině případů mělo jednat o spisovnou češtinu s minimem chyb (to dokazuje poměrně nízký počet unikátních tokenů) a také s obecným zaměřením (Spoustová a Spousta, 2012, s. 314). Autoři se domnívají, že korpus dosahuje vyšších kvalit díky využití technik předzpracování textu a také z důvodu vysokého podílu lidského zásahu při vytváření (Spoustová a Spousta, 2012, s. 311).

CZES je dalším ryze českým korpusem. Data byla extrahována ze zpravodajských webových stránek z let 1995–1998 a 2002 (Sketch Engine, 2015), korpus tedy může být zastaralý a některá novější slova, např. názvy společností, v něm nemusí být obsažena. To by mohlo mít negativní vliv na kvalitu NER systému, který by využíval vektorovou reprezentaci slov tohoto korpusu. Oproti CWC-2011 obsahuje takřka dvakrát více unikátních tokenů, ačkoliv celkový počet slov je menší. Důvodem může být buďto vyšší chybovost slov či rozmanitější obsahy textů.

SYN-2015 je specifickým korpusem z množiny korpusů nesoucí souhrnný název SYN. Jedná se o tzv. „synchronní“ korpusy, což z hlediska terminologie ČNK značí snahu o dodržení jednoho jazykového úzu, tj. tvorby korpusu v jednom vymezeném období (Cvrček a Richterová, 2015). V jejich textech dominují publicistické články, beletrie a odborné články (Cvrček a Richterová, 2018). Souhrnný korpus SYN, obsahující všechny subkorpusy, je ročně aktualizovaný. Každá nová verze obsahuje vždy data všech předešlých verzí s přírůstkem v podobě aktuálních publicistických dat daného roku. Nejnovější sedmá verze obsahuje přes 4 miliardy slov. Subkorpus SYN-2015 je tedy pouhým zlomkem celé rodiny SYN korpusů, nicméně jeho předností je doménová vyváženost (všechny tři výše zmíněné domény jsou naprosto vyrovnané) a aktuálnost, neboť je ze všech vydaných SYN subkorpusů nejnovější (data pocházejí z let 2010 až 2014).

Extrakce české Wikipedie již byla zmíněna výše při popisu korpusu CoNLL-2017, neboť je jeho součástí. Výhodou tohoto datového zdroje je jeho snadná dostupnost a aktuálnost, protože existuje mnoho různých veřejných nástrojů pro stažení současných dat[20].

Posledním zdrojem ze zmíněných je EuroParl. Oproti všem ostatním korpusům má však dva velké rozdíly. Prvním je velmi specifické zaměření obsahu, neboť texty pocházejí z jednání Evropského parlamentu mezi lety 1996–2011 (Tiedemann, 2012). Druhým je velikost, která je mnohonásobně menší než všechny předešlé korpusy. Původní účel zdroje tkví ve vytvoření automatického strojového překladu politických textů díky přítomnosti stejného obsahu v mnoha jazycích. Korpus je ukázkou toho, že i pro velmi specifickou doménu, jako je např. politika, lze najít dostatečně rozsáhlé textové zdroje pro tvorbu vektorových reprezentací slov.

Pro anglický text je obecně více příležitostí získat rozsáhlé zdroje psaného textu pro takřka jakoukoliv datovou doménu[21]. Důvodem je rozšiřitelnost jazyka na světě,

[20]Jeden z takových nástrojů je dostupný na adrese `https://dumps.wikimedia.org/` a jeho nejaktuálnější extrakce české Wikipedie zde: `https://dumps.wikimedia.org/cswiki/latest/`.

[21]Příkladem budiž seznam volně dostupných zdrojů psaného textu převážně v anglickém jazyce zde: `https://github.com/niderhoff/nlp-datasets`.

zejména pak v internetovém prostoru. Rovněž díky hojnému využívání vektorových reprezentací slov v NLP a dostupnosti rozsáhlých zdrojů existuje mnoho natrénovaných modelů vnoření slov[22], nicméně většina z nich využívá základní nastavení učících algoritmů, což může snižovat jejich kvalitu.

2.4 Rozpoznání pojmenovaných entit

Rozpoznávání pojmenovaných entit (anglicky „Named Entity Recognition“, zkráceně NER) sahá až do 80. let 20. století, v té době začalo dolování dat z textu mít svou nezastupitelnou roli v odvětví umělé inteligence. Zpočátku nebyla tato úloha explicitně definována a spadala do obecnější úlohy, zvané „automatizované porozumění dokumentům“ (Nouvel, Ehrmann a Rosset, 2016, s. 11).

První z pokusů o identifikaci pojmenovaných entit byl započat na konferenci MUC (Message Understanding Conference), která si kladla za cíl vytvořit systém pro automatizovanou identifikaci vlastností dokumentů o teroristických útocích, jako např. typ události, zmíněné osoby, lokace apod. Ačkoliv výsledky konference nebyly příliš přívětivé, zejména z důvodu náročnosti na tehdy nedostupný výpočetní výkon, zasadily stavební kameny pro úlohy NLP, včetně rozpoznávání pojmenovaných entit.

Postupem času se konaly další série konferencí (další ročníky MUC, ale také i jiné, např. ACE – Automatic Content Extraction), které zasadily první definici NER: „*Rozpoznávání kusů informací, které jsou relevantní k diskutovanému subjektu a které hrají významnou roli v popisu události či faktu.*“ Do této definice spadaly nejen pojmenované entity jako osoby či lokace, ale také numerické výrazy jako např. datum či peněžní jednotky (Nouvel, Ehrmann a Rosset, 2016, s. 13).

Další obrovský rozmach NER započala konference CoNLL (Conference on Natural Language Learning) v roce 2003, která účastníkům poskytla robustní anotovaný korpus pojmenovaných entit v anglickém a německém jazyce. Systémy NER byly do té doby založeny výhradně na pravidlech a rozsáhlých slovnících pojmenovaných entit, tento korpus započal éru strojového učení v NER.

Novodobé definice NER se oproti zmíněné první zásadně neliší a v podstatě sdělují totéž: „*NER je systém pro automatizované rozpoznávání slov či slovních spojení (pojmenovaných entit) v textu a jejich klasifikaci do předem určených kategorií (typů), jako je např. osoba, lokace či organizace.*“ (Deep AI, c2017–2018). Úloha se obvykle skládá ze dvou částí: identifikace pojmenovaných entit (u každého slova správně určit, zdali se jedná o entitu či nikoliv) a následná klasifikace (zařazení slov, které byly v předešlém kroku vyhodnoceny jako entity, do předem stanovených kategorií). Tyto části bývají ve většině systémů brány jako jeden nedělitelný celek (Straková, Straka a Hajič, 2016, s. 7). Ačkoliv výše zmíněné kategorie bývají nejčastěji rozpoznávanými, typem pojmenovaných entit může být takřka cokoliv, co tvoří množinu slov či slovních spojení se společnými rysy (Straková, Straka a Hajič, 2016,

[22]Zmínit lze např. repositář dostupný na adrese `http://vectors.nlpl.eu/repository/`, ovšem podobných pramenů existuje mnohem více.

s. 8). Rovněž může být definována i víceúrovňová hierarchie kategorií pojmenovaných entit.

NER se řadí do NLP kategorie extrakce informací (Gupta, 2018a). Konkrétněji spadá do skupiny úloh nazývaných „sequence labelling“, neboli značení sekvence textu, ve které se každému elementu v sekvenci přiřadí určitý štítek. Ze zmíněného je patrné, že sekvence se bere jako jeden celek a její elementy jsou určitým způsobem provázané a nelze na ně nahlížet jako na samostatné jednotky. Tuto vlastnost využívá mnoho moderních systémů NER, které používají techniky strojového učení založené na zkoumání kontextu slov.

Jednotlivé kroky tvorby modelu NER lze popsat takto:

1. definice kategorií pojmenovaných entit,
2. zvolení přístupu rozpoznávání (viz sekce 2.4.2),
3. příprava zdrojů dle zvoleného přístupu,
4. implementace modelu,
5. extrakce entit z testovacího textu (tzv. „Golden standard“[23] text),
6. vyhodnocení kvality rozpoznávání (viz sekce 2.4.1),
7. přijetí implementace či její vylepšení a opakování vyhodnocení,
8. využívání modelu.

Vyhodnocení systémů probíhá většinou pouze v implementační fázi. Jakmile je NER systém dostatečně kvalitní (hranici určuje uživatel), daný model je přijat a lze jej využívat pro rozpoznávání nového textu. Často je nutné tento text předzpracovat před samotnou extrakcí entit. Jedná se např. o tokenizaci, určování hranic vět či hledání frází.

2.4.1 Vyhodnocení systému

Důležitou součástí implementace systému NER je vyhodnocení kvality rozpoznávání, které dá představu o tom, jak se systém přiblížil lidskému rozpoznávání[24]. Kvalitu lze měřit několika způsoby. Vždy je však nutné mít k dispozici testovou datovou množinu, ve které jsou ručně anotovány rozpoznávané typy entit pro porovnání skutečnosti oproti rozpoznání systémem. Navíc by tato data měla reprezentovat typické využití systému v praxi a mělo by se jednat o dosud nespatřená data systémem pro zjištění kvality generalizace.

[23] Termín „Golden standard“ se využívá v NLP pro označení korpusů, jejichž text je dle NLP úlohy ručně anotovaný lidskými experty, a tudíž se považuje za 100% správný. Využívají se zejména pro trénování či testování systémů NLP.

[24] Ani lidské rozpoznávání pojmenovaných entit však nemusí být vždy správné a existují situace, ve kterých se lidé nedokáží jednoznačně shodnout na správném výsledku.

Důležitými pojmy v evaluaci NER, které se využívají pro výpočet mnohých metrik, jsou:

- správně pozitivní výskyt T_p („true positive“) – systém **správně označil** token za entitu,
- správně negativní výskyt T_n („true negative“) – systém **správně neoznačil** token za entitu,
- falešně pozitivní výskyt F_p („false positive“) – systém **nesprávně označil** token za entitu,
- falešně negativní výskyt F_n („false negative“) – systém **neoznačil** token za entitu, **ačkoliv se o entitu jedná**.

Nejčastěji využívanými metrikami pro vyhodnocení kvality NER systému jsou přesnost („precision“), pokrytí („recall“) a F1-míra („F1-score“). Přesnost P udává, kolik procent tokenů označených jako entita je rozpoznaných správně, tedy:

$$P = T_p/(T_p + F_p)$$

Pokrytí R představuje procento rozpoznaných entit z celkového počtu v textu, tedy:

$$R = T_p/(T_p + F_n)$$

F1-míra je (vyváženým[25]) harmonickým průměrem obou těchto metrik, díky čemuž přináší dobrou představu o celkové kvalitě NER systému:

$$F1 = 2\times\frac{P\times R}{P + R}$$

Vzhledem k tomu, že typů rozpoznávaných entit je obvykle více, existují techniky pro průměrování metrik každého z nich, neboť obvykle je žádané znát kvalitu NER systému jako celku, nikoliv jen jednotlivých typů (Pahwa, 2017). První z technik se nazývá „makro-průměrování“ a provádí výpočet tak, že se nejprve spočítají metriky každého typu a poté se z nich vypočte aritmetický průměr. Tím tedy přiřazuje všem typům stejnou váhu. Vzhledem k tomu, že většinou jsou jednotlivé typy nevyvážené co do počtu výskytů v textu (tzv. „support“), uplatňuje se častěji technika, zvaná „mikro-průměrování“, která agreguje jednotlivé typy dohromady a až poté provede samotný výpočet metrik, díky čemuž má každý typ ve výpočtu svou váhu dle počtu výskytů jeho entit.

Při evaluaci systému NER je třeba také rozhodnout, kdy se entita považuje za správně rozpoznanou. Vzhledem k tomu, že pojmenované entity jsou často tvořeny vícero tokeny (anglicky tzv. „chunks“), je nutné rozlišovat úplné a částečné

[25]F1-míra je vyvážená tehdy, když je na obě složky výpočtu – přesnost a pokrytí – kladen stejný důraz. Lze ovšem za použití mírně odlišného vzorce docílit k výsledku, ve kterém je na jednu ze složek kladen větší důraz.

rozpoznání. K úplnému dochází tehdy, jestliže systém rozpozná všechny tokeny tvořící entitu. K částečnému, pokud byly rozpoznány jen některé z nich. Je pak na uživateli, jak s těmito informacemi naloží při evaluaci (Batista, 2018). Ovšem pro zajištění zachycení víceslovných pojmenovaných entit v testovací datové množině je nutné využít schémata, která to umožňují. Nejznámějšími z nich jsou tzv. „BIO“ a „BILUO“ schémata[26]. Ty přiřazují každému typu prefix B či I (resp. B, I, L či U) dle toho, zdali se jedná o první token entity (B), či o další v řadě (I, resp. L pro poslední). Značka O je přiřazena všem tokenům, které nejsou součástí žádné entity. Jednoslovným entitám se přiřazuje značka B (resp. U).

2.4.2 Přístupy

Systémy NER lze rozdělit do následujících přístupů dle způsobu rozpoznávání (Goyal, Kumar a Gupta, 2017, s. 1904):

- založené na pravidlech a slovnících,
- založené na strojovém učení:
  - statistické metody,
  - neuronové sítě,
- kombinace obou zmíněných (někdy nazývaný jako „hybridní“ systém).

Historicky první systémy byly založeny na pravidlech či slovnících, nebo na obojím v kombinaci. Pravidla se využívala u entit, které lze rozeznat regulárními výrazy (Ingersoll, Morton a Farris, 2013, s. 117). Jedná se například o datum, čas či entity s určitým výskytem textu (v češtině např. společnosti s koncovkami „s.r.o.“ apod.). Slovníkové přístupy (tzv. „gazetteers“) spočívají v hledání jednotlivých slov v předem definovaných seznamech pojmenovaných entit, jako je např. seznam všech jmen a příjmení či společností (GATE, c1995–2019). V některých systémech se využívá kombinace slovníků a pravidel tak, že jakmile je nalezeno slovo ve slovníku (např. křestní jméno) a zároveň splňuje určité podmínky (např. druhé slovo začíná velkým písmenem), je označeno jako pojmenovaná entita (GATE, c1995–2019).

Výhoda pravidel spočívá v jednoduchosti implementace. Na druhou stranu mnohdy může být náročné nalézt taková pravidla, která dostatečně přesně identifikují určitý typ entity a zároveň neidentifikují chybně jiné tokeny. Velkou nevýhodou pravidel je fakt, že jen několik málo typů entit jimi lze nalézat. Výhoda slovníků je ta, že stačí sehnat slovníky pojmenovaných entit a systém lze pak okamžitě jednoduše implementovat. Otázkou však je, zdali jsou seznamy již vytvořené či volně dostupné, anebo je třeba je ručně vytvořit. Druhý případ je časově o dost náročnější. Komplikací slovníkových metod jsou mnohoznačná slova a pak také entity, které jsou kombinací více slov ze seznamu (Ingersoll, Morton a Farris, 2013, s. 118).

[26] Jednotlivá písmena schémat jsou zkratkami anglických slov popisující jejich využití: B – Beginning, I – Inside, O – Outside, L – Last, U – Unit-length.

Velkou nevýhodou slovníků je také fakt, že velmi rychle zastarají. Ty entity, které nejsou obsaženy v seznamech, nebudou nikdy v textu nalezeny. Jak pravidla, tak slovníky, mají charakteristickou vysokou přesnost, ale nízké pokrytí, právě z důvodu rychlého zastarání slovníků.

Modely založené na strojovém učení vyžadují trénovací data v podobě ručně (lidsky) anotovaných entit v textu, na základě kterých se naučí zákonitosti v rozpoznávání (Ingersoll, Morton a Farris, 2013, s. 119), jedná se tedy o techniky strojového učení s učitelem. Nejedná se však o prostý výpis či seznam entit, ale o jejich využití v různých kontextech, které se očekávají v reálných datech. Díky tomu je při zvolení vhodných technik strojového učení systém schopen rozpoznat entity na základě okolního kontextu. Je však nutné, aby jednotlivé typy entit byly v textu dostatečně zastoupené (typicky tisíce výskytů a více), jinak se nebude systém schopen dostatečně kvalitně naučit rozpoznávat určité typy. To je hlavní nevýhodou přístupu. Naopak obrovská výhoda spočívá v tom, že je rozpoznávání pojmenovaných entit založené právě na zmíněném kontextu, neboť některé techniky umějí díky tomu rozpoznat dosud nespatřené entity.

Systémy založené na strojovém učení se dále dají rozdělit do dvou podkategorií dle použitých technik učení. Do první z nich lze zařadit „tradičnější“ statistické metody, které se běžně využívají pro klasifikaci sekvencí a rozpoznávání vzorů. Jedná se např. o metody „Podmíněná náhodná pole“ (CRF), „Skrytý Markovův model“ (HMM) či „Metoda podpůrných vektorů“ (SVM), často v kombinaci s multinomiální logistickou regresí (někdy též označovanou jako logistická regrese pro více tříd či anglicky „Maximum Entropy“), která zaručí finální klasifikaci tokenu do definovaných typů entit (Athavale a kol., 2016, s. 1). Do druhé lze zařadit jakýkoliv systém NER, který je založen na neuronových sítích libovolné architektury. Velmi populární jsou rekurentní či konvoluční hluboké neuronové sítě.

Techniky obou zmíněných podkategorií mají společné to, že přijímají určitý vstup s jeho rysy (anglicky „features“), které mohou být libovolné a jejich výběr mívá značný dopad na kvalitu rozpoznávání. Rysy se dělí na jazykově závislé a nezávislé (Král, 2011, s. 3). Mezi jazykově závislé patří všechny ty, které se dají využít pouze pro určitý jazyk. Jedná se např. o slovní druhy, lemmata a kořeny slov či vektorovou reprezentaci tokenu (viz sekce 2.3), která se v dnešní době stává v NER systémech velmi populární. Mezi nezávislé lze zařadit všechny ty, které lze použít univerzálně pro jakýkoliv jazyk bez jakéhokoliv zásahu. Jedná se např. o délku tokenu či o různé binární informace o splnění určité vlastnosti – např. zdali je počáteční písmeno velké či zdali je token číslo. Ačkoliv jazykově závislé rysy obvykle přinášejí vysoký nárůst kvality rozpoznání, činí ale systém NER nerobustním z hlediska reprodukovatelnosti na další jazyk. To je jeden z hlavních důvodů, proč se v dnešní době moderní systémy omezují pouze na jazykově nezávislé rysy, které sice neslibu-

jí nejvyšší možnou přesnost a pokrytí, přesto ale dokáží uživateli nabídnout velmi kvalitní rozpoznávání, které navíc může být jazykově univerzální[27].

Modely postavené na strojovém učení většinou nedosahují takové přesnosti, jako ty, které jsou založené na pravidlech a slovnících, nicméně mají většinou lepší pokrytí, neboť si dokáží poradit s dosud nespatřenými entitami díky využívání kontextu při rozpoznávání. Některé moderní systémy jsou navrženy tak, aby si z obou přístupů vzaly to nejlepší a dosahovaly tím co nejvyšší kvality. Ty často využívají statistické metody jako svou hlavní součást a zvyšují kvalitu rozpoznávání doplňkovými pravidly či vyhledáváním entit v předem připravených seznamech. Pokud je žádoucí rozpoznávání mnoha typů entit, je většinou nutností využít oba přístupy, neboť pro některý typ entit, jako např. datum, nemá smysl využívat statistické metody.

2.4.3 Praktické využití

Rozpoznávání pojmenovaných entit má řadu využití. Často bývá systém NER součástí větších NLP řešení, které jej využívají jako jeden z mnoha dílčích kroků k naplnění zamýšleného účelu (Athavale a kol., 2016, s. 1). Může se jednat např. o (Gupta, 2018b):

- **klasifikaci obsahu** – např. vyhledání zmíněných osob či lokací v novinových článcích, což může vést k jejich jednodušší kategorizaci,
- **optimalizaci vyhledávání** – např. zlepšení hledání relevantních dokumentů na základě uživatelského dotazu,
- **relevantnější doporučování obsahu** – např. na základě uživatelských preferencí lze vyhledat podobné dokumenty, obsahující stejné pojmenované entity,
- **zlepšení uživatelské podpory** – např. na základě zmíněné pobočky v e-mailu uživatele lze efektivněji alokovat případ zaměstnanci.

Existuje také řada již funkčních NER systémů, které lze využívat prostřednictvím různých rozhraní či knihoven, které stačí pouze zahrnout do programovacího kódu, čímž lze ihned rozpoznávat určité pojmenované entity v textu. Bohužel však tyto systémy, určené k okamžitému používání, umějí rozpoznávat pouze pár typů entit (většinou tři základní – osoby, lokace a organizace). Navíc jsou mnohdy trénované na obecných datech, čímž jejich využití ve specifické doméně může dosahovat špatných výsledků. V takovém případě je stejně nutné implementovat vlastní systém[28].

[27] „Jazykově univerzální“ je zde nutno brát s velkou rezervou. NER systém postavený na strojovém učení a využívající jazykově nezávislé rysy podporuje i tak pouze ty jazyky, pro které má dostupnou trénovací datovou množinu v podobě anotovaných pojmenovaných entit v textu.

[28] Některé NER systémy určené k okamžitému používání nabízí možnost natrénování vlastních typů entit či vylepšení rozpoznávání existujících typů, např. pomocí inkrementálního trénování.

2.4.4 Dostupné datové zdroje pro tvorbu systému

Tato sekce popisuje datové zdroje, jejichž využití je nutné pro trénování systému NER metodami strojového učení. Jedná se tedy o textová data, která jsou anotovaná definovanými typy entit. Není zde však předložen úplný výčet všech dostupných dat, nýbrž těch, která jsou často používána v jiných prací.

V roce 1995 proběhla konference MUC, jejíž obsahem byla implementace NER systému pro anglický jazyk. Výsledky byly obstojné, bohužel však všechny implementace využívaly jazykově závislé prostředky (jako např. seznamy entit apod.). Na tento nedostatek se snažila navázat konference CoNLL, konaná v roce 2002, jejímž tématem byla implementace jazykově nezávislého NER systému (Tjong Kim Sang, 2002). Účastníkům poskytla anotované texty ve španělském a holandském jazyce, obsahující čtyři typy pojmenovaných entit – osoby, lokace, organizace a „různé“ (anglicky „miscellaneous“), které nepatří ani do jednoho ze zmíněných. Těmito datovými zdroji tak odstartovala éru strojového učení v NER systémech.

Data byla rozdělena na trénovací, validační a testovací část, dle konvence ve strojovém učení. Textové soubory obsahují na každém řádku jedno slovo s jeho anotací a jednotlivé věty jsou odděleny prázdným řádkem. Oddělení jednotlivých dokumentů je realizováno speciálními značkami. Pro rozlišování víceslovných pojmenovaných entit se využilo IOB schéma.

Stejná konference, pořádaná o rok později, přinesla účastníkům další anotované texty se stejnými typy entit jako v roce 2002 v dalších dvou jazycích – angličtině a němčině (Tjong Kim Sang a De Meulder, 2003). Právě až tento anglický korpus (ve světě známý jako CoNLL-2003) umožnil obrovský rozmach implementací NER pomocí různých technik strojového učení. Jedná se totiž o zdroj v nejrozšířenějším mezinárodním jazyce, jehož použitím v NER systémech se mohly začít věnovat mnohé vědecké skupiny. Obsah anglické části se skládá z novinových článků společnosti Reuters a obsahuje celkem zhruba 20 000 vět, 300 000 tokenů a 35 000 pojmenovaných entit (Tjong Kim Sang a De Meulder, 2003).

Další rozsáhlý anotovaný korpus vznikl v roce 2006 za podpory programu GALE (Global Autonomous Language Exploitation) pod vedením Agentury amerického ministerstva obrany (DARPA). Nese název OntoNotes a v průběhu let 2005–2013 se rozvíjel a vylepšoval až do poslední verze 5.0 (Weischedel a kol., 2013). Cílem bylo mimo jiné poskytnout anotovaná data z méně specifické datové domény, proto text pochází z různých zdrojů, jako jsou novinové články, rozhlasy, rozhovory, blogy či telefonní hovory (Weischedel a kol., 2013, s. 4). Korpus je ve třech jazycích – angličtině (1,5 mil. tokenů), čínštině (800 tis. tokenů) a arabštině (300 tis. tokenů) a obsahuje různé anotace, jednou z nichž je zaznačení 18 typů pojmenovaných entit (11 pro názvy a jména, 7 pro různé hodnoty). Nechybí základní typy entit s rozšířením o např. produkty, události či jazyky (Weischedel a kol., 2013, s. 21–22). Tento datový zdroj se díky své velikosti a obecnosti využil v implementaci mnoha systémů NER, komerčních nevyjímaje.

Speciálním případem anotovaného zdroje je Wikipedie. Její primární účel netkví v poskytování zdrojových dat pro NLP, nicméně v jejích textech využívá interní značkování význačných slov, jako např. jmen, míst či událostí, čehož lze využít pro automatické vygenerování velmi rozsáhlého a obecného korpusu pojmenovaných entit. Způsobů extrakce těchto dat z Wikipedie je několik a každý může být více či méně účinný a kvalitní. Pokud se ale nalezne vhodný, lze získat velmi mocný korpus pro NER, který obsahuje miliardy tokenů s desítkami milionů pojmenovaných entit a velmi obecným zaměřením, navíc v mnoha různých jazycích (Ghaddar a Langlais, 2017, s. 1).

Existuje spousta dalších rozsáhlých zdrojů pro NER, které však již nejsou tolik využívané a oblíbené jako doposud zmíněné. Některé z nich jsou pak úzce specializované např. na vyhledávání speciálních pojmenovaných entit v medicíně či na rozpoznání entit ve tweetech.

Situace je ale jiná pro češtinu. Na rozdíl od výzkumu NER pro anglický jazyk, pro češtinu až do roku 2007 neexistoval korpus obsahující označkované pojmenované entity, který by se dal použít jako datová množina pro strojové učení. Tuto mezeru vyplnili autoři Ševčíková, Žabokrtský a Krůza (2007), kteří vytvořili korpus pro úlohu NER v českém jazyce, nesoucí název CNEC 1.0 (Czech Named Entity Corpus). Ten obsahuje dva tisíce náhodně zvolených vět[29] z Českého národního korpusu (ČNK), ve kterých je označeno přes 11 tisíc pojmenovaných entit. Oproti anglickým korpusům, jako např. zmíněný CoNLL-2003 korpus, obsahuje CNEC dvouúrovňovou anotaci. První úroveň rozděluje entity do 10 typů (tzv. „supertypy“), druhá úroveň rozšiřuje typy z první na 62 typů, a přináší tak detailnější klasifikaci. Dalším rozdílem oproti anglickým korpusům je využívání vnořených entit, tedy anotace textu, který je již součástí jiné pojmenované entity.

Postupem času byly zveřejněny další dvě verze, konkrétně 1.1, která se liší spíše technickými změnami a verze 2.0, ve které jsou změny větší – z původních 64 typů vzniklo sjednocením 46 a dále přibyl nový text, ve kterém se objevují entity typů, které v předchozích verzích neměly reprezentativní zastoupení.

Kromě přibývajících verzí vznikl také nový korpus, který všechny verze CNEC převedl do formátu velmi podobnému korpusu CoNLL, díky čemu došlo k značnému zjednodušení anotace a zamezení vnořených entit (Konkol a Konopík, 2013, s. 2–3). Proběhlo také omezení typů na 7 základních. Korpus se nazývá Extended CNEC.

2.4.5 Existující systémy

Rozpoznání pojmenovaných entit, zejména v anglickém textu, je ve světě již velmi dobře řešeným tématem. Dřívější implementace byly postaveny převážně na ručně vytvářených pravidlech a slovnících. Na přelomu tisíciletí, konkrétně po publikacích anotovaných korpusů CoNLL-2002 a CoNLL-2003, začaly tyto systémy vytlačovat modernější implementace, založené na strojovém učení.

[29] Vzhledem k náhodnému výběru vět z Českého národního korpusu, obsahující texty z různých oblastí, nelze jednoznačně určit zaměření (doménu) dat.

Dalším milníkem bylo zveřejnění techniky Word2Vec, které započalo široké využívání moderních vektorových reprezentací v systémech NER. Nejlepší ze systémů v dnešní době využívají různé architektury neuronových sítí v kombinaci s těmito vektorovými reprezentacemi slov, díky čemuž dosahují přes 90 % F1-míry v rozpoznávání pojmenovaných entit na anglickém textu (Lample a kol., 2016, s. 267).

V této sekci jsou představeny práce jiných autorů, které se zabývají implementací NER systémů různými přístupy a způsoby založenými na technikách strojového učení. Jsou zde však zahrnuty jen ty práce, které ve své dokumentaci uvádějí výkonnost na některém ze zmíněných dat v sekci 2.4.4, a to v podobě představených metrik v sekci 2.4.1. Vzhledem ke stanovenému cíli práce jsou nejdříve předloženy významné práce, které realizují systém NER pro češtinu. Následují některé zahraniční práce, jež implementují moderní vektorové reprezentace slov ve svých systémech NER. Na to navazují práce, které využívají tyto techniky na český jazyk. Poslední část této sekce pak přináší shrnutí všech prací, které implementují NER systém pro češtinu.

Téma NER pro český jazyk začíná být mezi vědeckou komunitou populárnější až při zveřejnění první verze anotovaného korpusu CNEC (Ševčíková, Žabokrtský a Krůza, 2007). V té samé publikaci je rovněž představen první systém NER, který tento korpus využívá pro vytvoření rozhodovacích stromů (Ševčíková, Žabokrtský a Krůza, 2007, s. 191). Dosahuje poměrně slibných výsledků (Ševčíková, Žabokrtský a Krůza, 2007, s. 193), nicméně většina parametrů stromu je tvořena silně jazykově závislými vlastnostmi.

Všechny další zmíněné práce, implementující NER pro český jazyk, používají ve svých technikách strojového učení jako trénovací korpus CNEC či jeho alternativu Extended CNEC.

Autoři Kravalová a Žabokrtský (2009) využívají klasifikační techniku SVM v kombinaci s jazykově závislými pravidly a slovníky. Dosahují o něco lepších výsledků než předešlá práce, zejména pak v rozpoznávání víceslovných entit (Kravalová a Žabokrtský, 2009, s. 199).

Král (2011) experimentuje se statistickou metodou CRF, ve které využívá jak jazykově závislé vlastnosti slov (jejich lemmata či slovní druhy, slovníky atp.), tak i nezávislé (ortografické vlastnosti slov atp.). Dokazuje, že jejich výběr a kombinace hraje významnou roli na kvalitě rozpoznávání, přičemž jazykově nezávislé vlastnosti jsou pro NER přínosnější (Král, 2011, s. 5). Oproti předešlým pracím však dosahuje značně nižších výsledků (Král, 2011, s. 4), jeho přínos však tkví právě v exploraci vlastností. Práce má také praktické využití – implementovaný systém NER je využit pro vylepšení vyhledávače České tiskové kanceláře (Král, 2011, s. 1).

Ve stejný rok je zveřejněna publikace Konkola a Konopíka (2011). Ta předkládá NER systém založený na principu multinomiální logistické regrese („Maximum Entropy“), který vyžaduje přesně definované vlastnosti slov, jež budou využity při rozhodování v rozpoznávání entit. Práce jako první pro český jazyk experimentuje s vektorovými reprezentacemi slov. Využívá techniku COALS, která je založena na globálních statistikách korpusu a faktorizaci matice spolu-výskytu metodou SVD

(Konkol a Konopík, 2011, s. 206). Vzhledem k tomu, že práce nevyužívá pro rozpoznávání entit neuronovou síť, bylo nutné sestrojit techniky, které zužitkují vytvořené vektorové reprezentace. Bohužel se však ukázalo, že žádná z nich nevede ke zlepšení kvality NER systému (Konkol a Konopík, 2011, s. 207). Přesto práce dosahuje v té době nejlepších výsledků pro NER v českém jazyce. Autoři se správně domnívají (jak dokážou další předložené práce níže), že i přes jejich negativní výsledky má smysl použít vektorové reprezentace slov v kombinaci s NER systémy, ovšem je nutné využít modernější techniky jejich tvorby v kombinaci s vhodnějšími technikami strojového učení (např. neuronovými sítěmi).

Kromě transformace korpusu CNEC do formátu podobnému CoNLL-2003 (viz sekce 2.4.4) předkládají autoři Konkol a Konopík (2013) systém NER, který stejně jako práce Krále (2011) využívá statistickou metodu CRF, ovšem oproti ní si dosahuje o několik procent F1-míry více (Konkol a Konopík, 2013, s. 157). Důvodem je zřejmě lepší zvolení vlastností slov a zdrojů (např. slovníků pojmenovaných entit), které metoda CRF využívá při svém rozhodování. Práce porovnává své výsledky se všemi dosavadními systémy NER pro český jazyk a všechny překonává, jedná se tedy v té době o nejlepší dostupné řešení.

V ten samý rok autoři Straková, Straka a Hajič (2013) přinášejí další „state-of-the-art“ systém NER, který se stává podkladem pro vytvoření volně dostupného software pro rozpoznávání pojmenovaných entit zvaného „NameTag“[30]. Využívá princip zvaný „Maximum Entropy Markov Model“, který kombinuje vlastnosti již zmíněného principu Maximum Entropy a nově také Skrytého Markovova modelu (Batista, 2017).

S příchodem moderních technik pro vektorové reprezentace slov (Mikolov a kol., 2013b) začalo mnoho zahraničních autorů s jejich využitím v NER. Jednalo se však především o práce, které se zaměřovaly na rozpoznávání pojmenovaných entit v anglickém textu. Většina tyto reprezentace využívá v kombinaci s různými architekturami neuronových sítí, jako je např. rekurentní LSTM (Lample a kol., 2016) či konvoluční LSTM (Chiu, 2016; Rudra Murthy, 2018). Tato kombinace se stala preferovanou technikou při vytváření nejlepších systémů NER. Existují však i práce, které vnoření slov využívají v kombinaci s tradičními statistickými metodami, jako je CRF (Seok a kol., 2016), a přesto dosahují dobrých výsledků.

Práce autorů Al-Rfou a kol. (2015) přináší mnohojazyčný systém pro rozpoznávání pojmenovaných entit, který podporuje 40 jazyků. Toho je docíleno díky využití automatizované extrakce Wikipedie, která pro všechny tyto jazyky automatizovaně generuje jak anotovaný korpus pojmenovaných entit (viz sekce 2.4.4), tak datové zdroje pro vytváření vektorových reprezentací slov (viz sekce 2.3.7). Samotný NER systém je pak realizován neuronovou sítí, která jako vstup přijímá takto vytvořené vektory slov.

Jak lze vypozorovat z výše zmíněných prací, zapojení vektorových reprezentací do systémů NER přináší nezanedbatelný nárůst kvality v podobě několika procent

[30]Nejnovější verze NameTagu je dostupná na adrese `http://ufal.mff.cuni.cz/nametag`. Ta využívá implementaci Strakové, Straky a Hajiče (2016).

F1-míry. Navíc pro dosažení vynikajících výsledků již není nutné ručně definovat zkoumané vlastnosti slov („features“), využívat ručně vytvářená pravidla či vytvářet zdroje znalostí (jako např. seznamy). Tohle vše totiž dokáží nahradit vektory slov ve svých skrytých vztazích a podobnostech (Rudra Murthy, 2018, s. 428). Autoři však v implementacích často využívají volně dostupné, již natrénované, modely vnoření slov, nebo si je vytvoří pomocí algoritmů ve výchozím nastavení. Nezkoumají tak detailněji **dopad kvality těchto vektorových reprezentací na kvalitu systému NER**, který by však mohl být nezanedbatelný.

První pokus o využití moderních technik vnoření slov v systému NER **pro český jazyk** provádějí Demir a Özgür (2014). Zkoumají použití moderních vektorových reprezentací v NER systémech pro morfologicky bohaté jazyky, konkrétně turečtinu a češtinu. Vektory slov jsou získány technikou Word2Vec Skip-Gram, autoři však již dále nezkoumají jejich různá nastavení (Demir a Özgür, 2014, s. 4). Pro trénování neuronové sítě je zvolen datový zdroj Extended CNEC 1.1. Většina systémů, které v dané době byly představeny pro morfologicky bohaté jazyky, využívala nástroje pro morfologickou analýzu slov, jako je lemmatizace či stemming a další specifické vlastnosti pro daný jazyk. V této práci je však upuštěno od jakýchkoliv jazykově závislých vlastností slov. Díky tomu práce získává hned dvě prvenství v systémech NER pro češtinu – využívá jak moderní vektorové reprezentace slov, tak pouze jazykově nezávislé vlastnosti slov. Její implementace překonává obě české práce z roku 2013, čímž se stává v té době nejlepším řešením. Práce dále demonstruje, že modely vnoření slov přispívají ke zlepšení rozpoznávání entit nejvíce (přes 10 % F1-míry) ze všech dalších vlastností vstupů (Demir a Özgür, 2014, s. 6).

Na všechny tyto poznatky navazuje publikace Strakové, Straky a Hajiče (2016), která přináší implementaci NER systému, jež překonává všechny dosavadní systémy pro český jazyk, nejlepší z nich (Demir a Özgür, 2014) pak o více než 5 % (Straková, Straka a Hajič, 2016, s. 6). Tím se implementace stává v dané době nejlepším zveřejněným NER systémem pro český jazyk. Publikace představuje dvě sady experimentů. První z nich dodává na vstup neuronové sítě s buňkami GRU slova trénovací množiny v původní, nezměněné podobě společně s jejich vektorovými reprezentacemi. Druhá pak navíc dodává lemmata, opět včetně jejich vektorových reprezentací, a slovní druhy jednotlivých slov (Straková, Straka a Hajič, 2016, s. 6). Na vstup neuronové sítě jsou následně postupně přidávány další vlastnosti, jako např. sufixy a prefixy slov či vektorové reprezentace znaků.

Práce využívá pro trénování vektorových reprezentací slov techniku Word2Vec Skip-Gram společně s množinou korpusů SYN (viz sekce 2.3.7), nicméně opět detailněji nezkoumá dopad dalších dostupných technik učení vektorů včetně jejich konfigurace na kvalitu výsledného NER systému (Straková, Straka a Hajič, 2016, s. 5).

Následovaly další dvě akademické práce (Nguyen, 2017; Matas, 2018), které implementují NER systém obdobnými přístupy. Navíc experimentují se zajímavými technikami v systémech NER, jako je např. 1D konvoluce slov (Nguyen, 2017, s. 29). Žádné z nich se však již nedaří překonat systém Strakové, Straky a Hajiče

(2016). Obě práce využívají vektorové reprezentace slov, nicméně opět žádná z nich neexperimentuje detailněji s technikami učení vektorů a jejich nastaveními.

Nedávno zveřejněná práce autorů Konopíka a Pražáka (2018) předkládá systém NER, který překonává doposud nejlepší, autorů Strakové, Straky a Hajiče (2016), ačkoliv jen o necelé jedno procento. Systém využívá několik postupných vrstev neuronové sítě, z nichž každá má svou specifickou činnost. Jedna z nich využívá volně dostupné vektorové reprezentace slov (v kombinaci s obousměrnými LSTM buňkami), trénované technikami GloVe a FastText. Opět tedy autoři dále nezkoumají kvalitu těchto vektorů, nýbrž jen využívají již vytrénované modely.

Tabulka 2 přináší shrnutí F1-mír všech zmíněných prací (resp. jejich nejlepších dosažených výsledků), které implementují NER systém pro český jazyk (tabulka je seřazena dle roku publikování práce). Jednotlivé práce však využívají různé verze korpusů CNEC či Extended CNEC, proto tabulka obsahuje zhodnocení obou z nich.

Tabulka 2: Srovnání F1-měr NER systémů pro český jazyk jednotlivých prací

Práce	**CNEC** [%]	**Extended CNEC** [%]
Ševčíková, Žabokrtský a Krůza (2007)	68,00	
Kravalová and Žabokrtský (2009)	71,00	
Konkol a Konopík (2011)	72,94	
Král (2011)	58,40[31]	
Konkol a Konopík (2013)	79,00	74,08
Straková, Straka a Hajič (2013)	82,82	
Demir a Özgür (2014)		75,61
Straková, Straka a Hajič (2016)	84,68	80,88
Nguyen (2017)	66,15	
Matas (2018)	77,16	
Konopík a Pražák (2018)		81,77

Důležitým výstupem této sekce je také fakt, že žádná ze zmíněných prací, ať už zahraniční či tuzemská, se detailněji nezaobírá kvalitou použitých modelů vnoření slov. Některé práce využívají již natrénované, volně dostupné modely, jiné si modely samy natrénují, avšak jednou konkrétní technikou, většinou ve výchozím nastavení. Jak bylo však dokázáno (Demir a Özgür, 2014, s. 6), vektorové reprezentace dokáží zvýšit kvalitu rozpoznávání entit o několik procent. Je tedy vhodné nejen využívat tyto vektorové reprezentace v NER systému, ale také se zabývat vhodností jejich jednotlivých technik včetně konkrétních nastavení.

[31] Výsledná F1-míra je zde značně zkreslená, neboť autor využil jen podmnožinu všech typů entit a využil rozdílný postup evaluace.

3 Metodika

Existující práce jiných autorů (viz sekce 2.4.5) dokázaly implementovat velmi kvalitní systémy NER pro český jazyk (podrobnější srovnání nabízí tabulka 2). Je tak velmi těžké překonat dané systémy, zejména bez použití jazykově závislých zdrojů a vstupních informací, které jsou užitečné pouze pro daný zdroj trénovacích dat. Někteří autoři se o překonání stále pokoušejí, někdy i úspěšně. Ovšem nárůsty kvality v podobě F1-měr bývají maximálně v desetinách procent.

Moderní vektorové reprezentace slov zaujímají právem podstatnou roli v dnešních NER systémech. Jejich použití přináší nárůst F1-míry o několik (až desítek) procent. Většina autorů však již detailněji nezkoumá, jestli jsou použité modely vnoření slov maximálně kvalitní. Je však velmi pravděpodobné, že zkvalitnění modelů povede také ke zkvalitnění systému NER.

Tato kapitola popisuje systém pro rozpoznávání pojmenovaných entit, ve kterém je kladen důraz právě na **kvalitu využitých modelů vektorových reprezentací slov**. Namísto pokusu o překonání dosud nejlepších systémů NER pro český jazyk se tento systém snaží odhalit dopad kvality vytvořených vektorů na kvalitu NER systému.

Zjišťování kvality vektorových reprezentací slov však není tak snadné. Existují sice techniky pro její měření, jako např. korpus analogií (viz sekce 2.3), nicméně jejich dobré výsledky nemusejí automaticky zaručovat také dobré výsledky systému NER. Na takové zjištění je třeba použít jiný přístup, který testuje přímý dopad kvality modelů vnoření slov na kvalitu rozpoznávání systému NER.

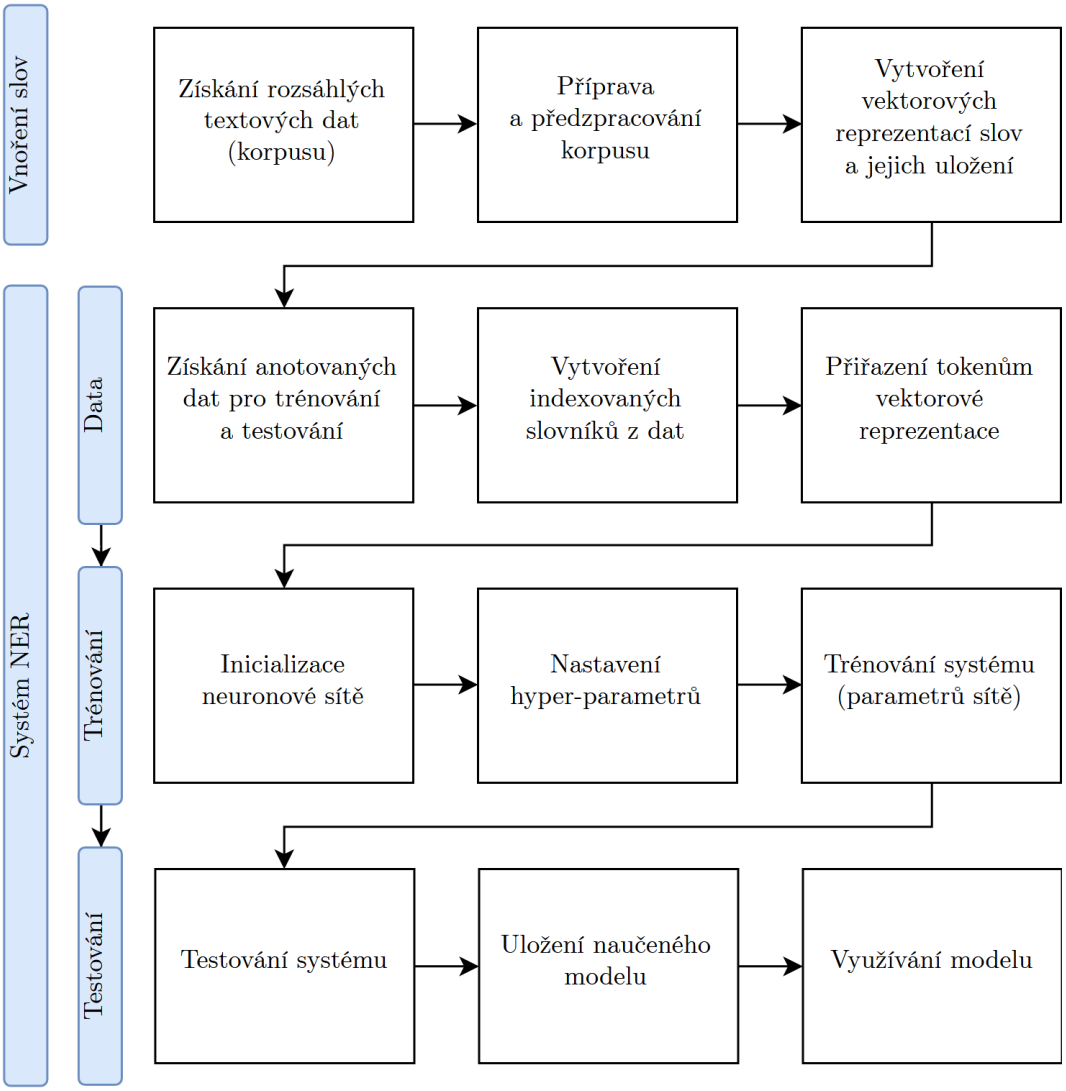
Zkvalitnění modelů vnoření slov může být docíleno několika způsoby. Může se jednat např. o využití jiného trénovacího korpusu, jiných technik předzpracování textu či o změnu trénovacích algoritmů a jejich nastavení. Pro zjištění závislostí těchto (a dalších) faktorů na kvalitě modelu vnoření slov v NER systému je vykonána řada experimentů, které jsou detailněji popsány v sekci 3.2. Experimenty by měly poskytnout dostatečnou představu o tom, jakým způsobem vytvořit model vektorových reprezentací slov, který přinese největší nárůst kvality NER modelu.

3.1 Návrh systému

Návrh systému NER reflektuje existující nejmodernější („state-of-the-art“) řešení. Většina z nich je postavena na rekurentních neuronových sítích s jazykově nezávislými vlastnostmi vstupů. Přesná architektura sítí se pak již v jednotlivých implementacích může lišit, zejména v typech neuronů a jejich hyper-parametrech.

Vzhledem k tomu, že cílem předkládaného systému NER není překonání výsledků dosavadních prací, nýbrž zjištění míry dopadu kvality vnoření slov na kvalitu NER systému, je při jeho učení rozpoznávání spolu s anotovanými daty k dispozici pouze vytvořená vektorová reprezentace slov. Tím je zajištěno, že výsledky navržených experimentů nejsou ovlivňovány žádnými jinými vstupy. Jejich rozdíly ve kvalitě jsou tedy zapříčiněny pouze využitými vektorovými reprezentacemi slov.

Vývojový diagram na obrázku 5 demonstruje dílčí kroky realizace systému NER. První část se zabývá generováním vektorových reprezentací slov z určitého zdroje rozsáhlých textových dat, druhá trénováním NER systému, včetně jeho testování.



Obrázek 5: Jednotlivé kroky realizace NER systému

Prvním krokem realizace systému NER je získání rozsáhlých textových dat (trénovacího korpusu), která se využijí pro vytvoření vektorových reprezentací slov. Následně je nezbytné daný korpus očistit, neboť většinou obsahuje nadbytečné informace. Jednotlivé kroky transformace se mohou u jednotlivých korpusů lišit, vždy je však žádoucí zbavit se nepotřebných informací, jako jsou metadata, různé značky (např. HTML či XML tagy) či speciální (nealfanumerické) znaky.

Jakmile jsou textová data očištěna, je nutné provést jejich transformaci do požadovaného formátu trénovacího algoritmu. Většina implementací předpokládá formát, ve kterém jsou vymezeny hranice vět včetně jednotlivých tokenů. V této

fázi lze využít techniky předzpracování, jako je změna velkých písmen na malá, lemmatizace slov či odstranění stop-slov.

Po dokončení úprav dat se vytvoří vektorové reprezentace slov. K dispozici jsou tři učící algoritmy Word2Vec, GloVe a FastText, popsané v sekci 2.3. U všech tří lze nastavovat počet dimenzí vektorů, velikost kontextového okna, počet epoch a minimální počet výskytů tokenu pro zařazení do slovníku.

U algoritmů Word2Vec a FastText lze navíc měnit způsoby trénování – Skip-Gram či CBOW. V případě využití způsobu CBOW lze nastavovat způsob výpočtu vektorů kontextových slov – buďto součtem, či průměrem. Dále lze zvolit funkci poslední vrstvy neuronové sítě – hierarchický softmax, či Negative Sampling, v druhém případě je nutné navíc určit počet negativních vzorků. Rovněž lze nastavovat práh techniky Sub-sampling, který je však ve většině experimentů neměnný (má nastavenou hodnotu 0,001).

Ve FastTextu lze navíc ovlivňovat rozsah počtů n-gramů, který je ve výchozím stavu nastaven na 3 až 6. U GloVe je zamýšleno v rámci experimentů měnit navíc pouze koeficient ovlivňující vážící funkci, jehož výchozí hodnota je rovna 0,75.

Detailní popis všech těchto zmíněných specifických nastavení je rovněž uveden v sekci 2.3, v částech věnujícím se jednotlivým algoritmům. Zbylé parametry jednotlivých algoritmů jsou ponechány ve výchozích nastaveních od jejich tvůrců.

Vytvořené vektorové reprezentace je nutno uložit. Využívají se dva způsoby ukládání – v textové a v binární podobě. V první je na každém řádku token a jeho vektor, ve druhém lidsky nečitelná data, která umožňují natrénovaný model daným algoritmem znovu načíst. Druhý bývá paměťově náročnější, avšak rychlejší.

Následuje fáze tvorby systému NER. Vzhledem k tomu, že je systém realizován technikou strojového učení, konkrétněji neuronovou sítí, je zapotřebí mít anotovanou datovou množinu, která se rozdělí na trénovací a testovací část. Síť potřebuje znát pro každý vstup jeho výstup (typ entity), tudíž je třeba datový zdroj transformovat do požadovaného formátu konkrétní implementace.

Neuronová síť dokáže pracovat s čísly a nikoliv s textem, proto je nezbytné po načtení těchto dat vytvořit indexované slovníky jejich unikátních tokenů a typů entit. Není nutné definovat rozpoznávané typy entit, neboť ty určují trénovací data, resp. indexovaný slovník typů entit.

V další fázi se každému unikátnímu tokenu z anotovaných dat (resp. každému záznamu z indexovaného slovníku tokenů) přiřadí unikátní vektor z modelu vnoření slov, vytvořeného na začátku. Ovšem je pravděpodobné, že některé tokeny z množiny anotovaných dat nebyly obsaženy v trénovací množině pro model vnoření slov, a proto nemají svoji vektorovou reprezentaci. S takovou situací je třeba se vypořádat. V tomto systému je takovým tokenům přiřazen nulový vektor.

Nyní již nic nebrání začít trénovat systém NER. Nejprve se inicializuje neuronová síť, tj. vytvoří se jednotlivé vrstvy, definují se jejich vlastnosti a zvolí se ztrátová a optimalizační funkce. Tento systém využívá tři vrstvy. První přijímá vektorové reprezentace slov a data předává další, skryté vrstvě. Ta využívá obousměrné LSTM buňky, které se prokázaly být v systémech NER v kombinaci s modely vnoření slov

vhodným typem neuronů (viz sekce 2.4.5). Tím se z neuronové sítě stává rekurentní[32]. Počet neuronů v této skryté vrstvě je zjištěn experimentem, který je detailněji popsán v sekci 3.1.2. Poslední vrstva přijímá data ze skryté a pomocí funkce softmax provede predikci typu entity pro daný vstup. Za optimalizační funkci je zvoleno rozšíření stochastického gradientního sestupu (anglicky „Stochastic Gradient Descent“) nazývané „Adam“. Ztrátová funkce se počítá pomocí metody křížové entropie (anglicky „Cross entropy“). Všechny tyto funkce byly zvoleny empiricky na základě zkušeností z jiných prací (Gregorič, Bachrach a Coope, 2018; Raaj, 2018).

Po inicializaci se síť začíná trénovat. Průběh trénování určují hyper-parametry, které jsou definovány uživatelem. V tomto systému lze nastavovat počet epoch, velikost dávky vstupních dat, hodnotu parametru učení a pravděpodobnost náhodného vypnutí neuronu. Jejich přesné nastavení je zjištěno tím stejným experimentem, ve kterém je zjišťován počet neuronů skryté vrstvy. Účel všech těchto hyper-parametrů popisuje detailněji sekce 2.2.3.

Po úspěšném trénování následuje finální fáze. Nejprve je provedeno testování za použití testovací části anotovaných dat, získané v předešlých krocích. Evaluace správnosti rozpoznávání využívá všechny popsané metriky ze sekce 2.4.1. Za správně rozpoznanou **víceslovní** entitu se považuje rozpoznání všech jejich tokenů. Po testování lze volitelně natrénovaný model uložit pro jeho budoucí využití, jako např. součást jiné NLP úlohy.

3.1.1 Data

Pro trénování popisovaného NER systému je zvolena datová množina Extended CNEC ve verzi 2.0 (její detailnější popis viz sekce 2.4.4). Jedinou alternativou se pro český jazyk jeví pouze originální CNEC, nicméně důvodů pro zvolení rozšíření je více. Hlavním z nich je jednoduchost tohoto zdroje, která spočívá ve zrušení víceúrovňové hierarchie typů entit a zamezení vnořených entit. Díky tomu se značně zjednodušuje jak načítání těchto dat, tak i další kroky implementace a samotný proces učení. I přes svou jednoduchost korpus přesto obsahuje základní typy entit, které většina jiných systémů umí rozpoznávat. Dalším důvodem zvolení korpusu je jeho využívání v mnoha publikacích a systémech, tudíž je zajištěna možnost porovnávání.

Pro vytváření vektorových reprezentací jsou zvoleny všechny korpusy, které byly zmíněny v sekci 2.3.7. Ty jsou postupně využívány v experimentech, vždy však odděleně (nedochází tedy ke spojování). Všechny jsou více či méně předzpracovány, neboť byly vytvořeny primárně za účelem využití v NLP úlohách. Většina z nich má tak např. již určené hranice vět či oddělené tokeny. Přesto obsahují nadbytečné informace, kterých je vhodné se zbavit, jak již bylo zmíněno výše. K tomu lze využít mnoho technik. Jednou z nich je Linuxový terminál (tzv. „Bash“), který je využíván v této práci. Specifické příkazy jednotlivých korpusů lze nalézt v elektronické příloze.

[32]Použitím obousměrných LSTM buněk vzniknou v podstatě dvě oddělené vrstvy – jedna pro slova nalevo od vstupního, druhá pro slova napravo. Poté dochází k opětovnému spojení výsledků obou těchto vrstev (Lample a kol., 2016, s. 262).

Jak anotovaná data pro NER, tak trénovací korpusy pro vytváření vektorových reprezentací lze bezplatně stáhnout z repositáře LINDAT-Clarin. Některé korpusy však vyžadují ověření akademické licence, většinou prostřednictvím přihlášení přes vybrané univerzitní informační systémy.

3.1.2 Nastavení hyper-parametrů

Nastavení jednotlivých hyper-parametrů neuronové sítě může významně ovlivnit kvalitu dosahovaných výsledků rozpoznávání. Proto je vyhledáno uspokojivé (nikoliv však optimální) nastavení pomocí experimentu. V něm je učiněno několik pozorování, která provádějí změny hodnot postupně všech hyper-parametrů a vždy je uskutečněna nová kompletní realizace systému NER (kterou demonstruje obrázek 5). Pro vektorovou reprezentaci slov je využit volně dostupný, již natrénovaný model české části rozsáhlého korpusu CoNLL-2017 technikou Word2Vec Skip-Gram[33] a pro trénování systému NER korpus Extended CNEC 2.0. Na kvalitě modelu vnoření slov v této fázi nesejde. Důležité jsou pouze vzájemné rozdíly jednotlivých pozorování ve kvalitě NER systému.

Před započetím experimentu jsou nejprve zvoleny výchozí hodnoty hyper-parametrů dle empirických poznatků z jiných prací. Následně dochází ke kladné i záporné změně těchto hodnot. Tato změna je prováděna postupně pro každý hyper-parametr a pokaždé je vyhodnocena kvalita daného NER systému v podobě F1-míry. Pokud některá ze změn dosahuje lepších výsledků než výchozí hodnota, je u ní provedena ještě větší změna pro ověření dalšího zlepšování. To se opakuje do té doby, než k dalšímu zlepšení nedochází. Poté je hodnota přijata za nejlepší a následují změny hodnot dalších hyper-parametrů. Jediným fixním hyper-parametrem je počet epoch (roven 25), jehož nejlepší hodnota bude určena až na závěr po provedení všech pozorování. U každého experimentu je měřena kvalita rozpoznávání po 25 epochách pomocí F1-míry na testovací části zdrojových dat Extended CNEC 2.0. Rovněž je v každé epoše vyhodnocována kvalita rozpoznávání na validační části dat.

Kompletní výsledky všech pozorování po jednotlivých epochách, včetně rozsáhlejších grafů, lze nalézt v elektronické příloze práce. Graf 6 demonstruje učící křivku nejlepšího nastavení jednotlivých hyper-parametrů. Z tabulky 3 vyplývá, že nejlepším nastavením, které dosahuje 68,16 % F1-míry, ze všech testovaných je:

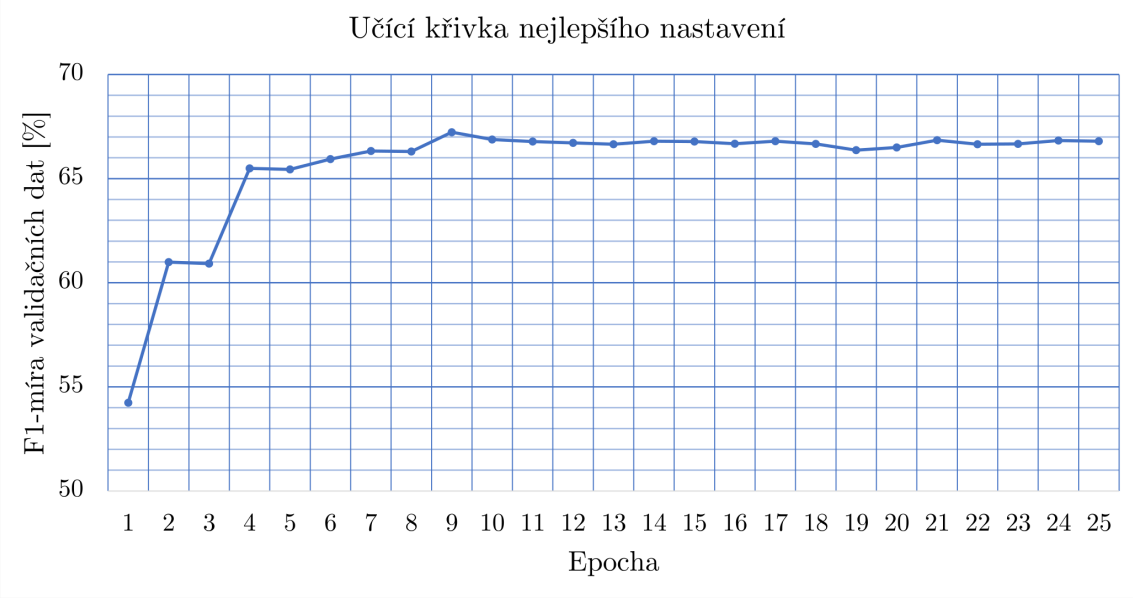
- velikost dávky = 32,
- velikost parametru učení = 0,02,
- pravděpodobnost náhodného vypnutí neuronu = 0,7,
- počet neuronů ve skryté vrstvě = 200,
- počet epoch = 12.

[33]Model lze stáhnout na adrese `http://vectors.nlpl.eu/repository` po vyhledání modelů pro český jazyk. Jeho vektory mají 100 dimenzí a kontextové okno je velikosti 10.

Zvolení tohoto počtu epoch vyplývá ze sledování postupných nárůstů kvality na validační části anotovaných dat v jednotlivých epochách (to lze vypozorovat na kompletních výsledcích v elektronické příloze práce). Pokud již s dalšími epochami nedochází k výraznému zlepšení, nemá smysl nadále model trénovat (navíc by mohlo začít docházet k přetrénování sítě). Takový počet epoch je tedy dostačující. Nejlepších výsledků je v průměru všech jednotlivých pozorování navíc dosaženo právě v této 12. epoše (viz kompletní výsledky v příloze).

Tabulka 3: Výsledky F1-měr NER systémů v jednotlivých nastaveních hyper-parametrů

Velikost dávky	**Velikost parametru učení**	**Pravděpodobnost náhodného vypnutí neuronu**	**Počet neuronů ve skryté vrstvě**	**F1-míra [%]**
32	0,01	**0,9**	200	65,41
32	0,01	**0,7**	200	**66,40**
32	0,01	**0,5**	200	64,44
32	**0,005**	0,7	200	64,95
32	**0,02**	0,7	200	**68,16**
32	**0,03**	0,7	200	65,80
16	0,02	0,7	200	66,16
128	0,02	0,7	200	62,92
32	0,02	0,7	**100**	64,95
32	0,02	0,7	**300**	66,75



Obrázek 6: Učící křivka nejlepšího nastavení v jednotlivých epochách

3.2 Návrh experimentů

Tato sekce popisuje jednotlivé experimenty, kterými je zjištěn dopad kvality vektorových reprezentací slov na kvalitu NER systému. Experimenty jsou vykonány sekvenčně tak, jak jsou zde postupně představeny v jednotlivých podsekcích. Každý z nich navazuje na předešlé experimenty tak, že se snaží využít zjištěné poznatky z trénovacích algoritmů a jejich nastavení. Zároveň však nejlepší výsledek z předešlého experimentu nemusí nutně znamenat jisté využití v dalších, neboť je také brána v potaz časová složitost daného nastavení, která by mohla značně ovlivnit výslednou dobu vykonávání všech dalších experimentů. Jestliže dojde k takovému uplatnění nastavení nejlepšího výsledku předešlého experimentu na další, lze pak snadno porovnávat rozdíly ve kvalitě NER systému mezi těmito experimenty. Anglicky se takovému výsledku z předešlého experimentu říká „baseline“ a ve výsledcích bude vždy vyznačen.

Experimenty mají za cíl zkoumat různé faktory, které by mohly mít vliv na výsledný NER systém. Každý z nich obsahuje řadu pozorování, z nichž v každém proběhne trénování modelu vnoření slov dle určitého nastavení a následně se s tímto modelem realizuje trénování systému NER (se zjištěným nejlepším nastavením hyper-parametrů neuronové sítě v sekci 3.1.2). Výstupem pozorování je pak F1-míra kvality rozpoznávání tohoto NER systému.

Konečným výsledkem těchto experimentů je jak konkrétní nastavení, jehož model vnoření slov dosahuje nejlepších výsledků v systému NER, tak i řada poznatků z jednotlivých experimentů, které mohou být užitečné pro pochopení důležitosti různých faktorů při tvorbě vektorových reprezentací slov. Právě druhý ze zmíněných je velmi důležitým výstupem této práce.

3.2.1 Trénovací datové zdroje

V prvním experimentu je cílem zjistit, zdali volba trénovacího korpusu, ze kterého se vytváří vektorové reprezentace slov, ovlivňuje kvalitu NER systému a případně v jaké míře. Postupně se testují všechny korpusy, které byly představeny v sekci 2.3.7. Ty jsou záměrně zvoleny tak, aby byla pokryta různě rozsáhlá a jinak kvalitní data. Například korpus CoNLL-2017 je sice zdaleka nejrozsáhlejší, nicméně podle počtu unikátních tokenů obsahuje řadu chyb, což by mohlo ovlivnit výslednou kvalitu z důvodu snížení počtu výskytů určitých slov v různých kontextech. Korpusy CWC--2011, CZES a SYN-2015 obsahují naopak realističtější počet unikátních tokenů, díky čemuž by unikátní tokeny měly mít více svých výskytů v datech. Tyto tři korpusy se liší zejména svou velikostí, což by se mělo promítnout ve výsledné kvalitě vektorů, a tedy i NER systému (Levy, Goldberg a Dagan, 2015, s. 219).

Vedle těchto korpusů je navíc také testován textový obsah anotovaného zdroje Extended CNEC 2.0, který primárně slouží pro trénování NER systému. Důvodem je snaha zjistit, zdali korpus velmi malého rozsahu může i tak pomoci systému NER, například v situacích, kdy jiná textová data než trénovací pro NER systém nejsou k dispozici.

Korpus EuroParl obsahuje data z úplně odlišné domény, což by se mělo negativně odrazit na kvalitě NER systému, a navíc není příliš rozsáhlý. U něj a u textových dat anotovaného zdroje Extended CNEC 2.0 má neuronová síť systému NER dovoleno měnit při učení jejich vektory (vektory jsou tzv. „trainable“) dle jejího uvážení z důvodu jejich malého rozsahu, u všech ostatních to má však zakázané. Ve všech použitých korpusech je před vytvářením vektorů provedena změna velkých písmen na malá a odstranění nealfanumerických znaků.

Pro trénování vnořených slov je zvolen algoritmus Word2Vec s technikou Skip-Gram, 100 dimenzemi, velikostí kontextového okna rovné 10, s využitím hierarchického softmaxu a počtu epoch a minimálního počtu výskytů tokenu pro zařazení do slovníku rovných 5. Toto nastavení volí rovněž mnozí autoři jiných NER systémů. Důvodem zvolení techniky Skip-Gram je fakt, že by měla být vhodná pro jakoukoliv NLP úlohu. Není sice zaručeno, že je vždy tou nejlepší, ale měla by poskytovat dostatečně kvalitní a užitečné vektorové reprezentace. Navíc má oproti jiným algoritmům a technikám nejmenší časovou a prostorovou složitost (Levy, Goldberg a Dagan, 2015, s. 222). Nicméně dopad těchto a dalších nastavení modelu vnořených slov na kvalitu NER systému je obsahem jiného experimentu (viz sekce 3.2.4).

3.2.2 Dimenze vektorů a velikost korpusu

V tomto experimentu je zkoumán dopad počtu dimenzí vektorů slov na výslednou kvalitu NER systému. Je empiricky dokázáno, že větší počet dimenzí do určité hodnoty dokáže lépe zachycovat vztahy mezi slovy, neboť je zde větší prostor k jejich zakódování (Pennington, Soche a Manning, 2014, s. 7). Ovšem experimenty, které tohle dokazují, byly prováděny na korpusu analogií a nikoliv na NER systému, na což navazuje tento experiment.

Předpokládá se, že správná velikost dimenze je o to důležitější u rozsáhlejších korpusů, neboť je v nich obsaženo daleko více skrytých vztahů mezi slovy než v méně rozsáhlých. Proto je experimentováno nejen s počty dimenzí, ale také s různě velkými korpusy, aby byl tento předpoklad potvrzen. Vybraný korpus je tedy rozdělen na jeho různě velké podmnožiny, konkrétně na 50, 10 a 1 % jeho velikosti. Využije se korpus, který dosahuje nejlepších výsledků z prvního experimentu. Opět na něj jsou uplatněny stejné techniky předzpracování, včetně učícího algoritmu, z důvodu umožnění porovnávat výsledky těchto dvou experimentů.

Počet dimenzí se ve většině prací pohybuje v intervalu <100;300>, nejčastěji jsou pak voleny právě hraniční hodnoty 100 či 300. Důvodem zvolení takového počtu je zřejmě i fakt, že počet dimenzí většiny volně dostupných již natrénovaných modelů vnoření slov se rovněž pohybuje v těchto hodnotách. V experimentu se tedy využívají velikosti dimenzí 50, 100, 200, 300 a 400, což pokrývá jak zmíněný interval, tak i jeho blízké okolní hodnoty.

Pokud by se potvrdilo, že obecně vyšší počet dimenzí je lepší, pak tato informace může velmi pomoci pro navýšení kvality NER systému. Přitom by se jednalo o doporučení, které je aplikovatelné velmi jednoduše, neboť zvýšení počtu dimenzí

lze uskutečnit ve všech algoritmech pro vytváření modelů vnoření slov. Jedinou nevýhodou vyššího počtu dimenzí je výsledná velikost souboru s vektory a vyšší časová a paměťová náročnost trénování modelu.

3.2.3 Techniky předzpracování textu

V tomto experimentu je zkoumán vliv technik předzpracování na výslednou kvalitu NER systému, které jsou uplatněny na trénovací korpus ještě před samostatným vytvářením vektorových reprezentací slov. Experimentuje se s lemmatizací, změnou velkých písmen na malá, odstraněním stop-slov a se všemi jejich možnými kombinacemi. Při využití lemmatizace či změny velkých písmen na malá je nutné obě tyto techniky rovněž uplatnit na anotovaná data NER systému. Jinak by nebylo možné pro některá slova dohledat vektorové reprezentace, čímž by se celá myšlenka uplatnění těchto technik předzpracování vytratila.

Dalším faktorem při zjišťováním nejlepších technik předzpracování může být velikost korpusu. Pro různě rozsáhlé korpusy mohou být vhodné různé techniky předzpracování. Každý unikátní token totiž potřebuje pro svou kvalitní vektorovou reprezentaci několik svých využití v různých kontextech. Vzhledem k tomu, že je čeština morfologicky bohatým jazykem, může se u menších korpusů stát, že v nich nebude dostatečný počet využití některých tvarů slov, neboť každý tvar slova vytváří unikátní token, pro který bude vytvořen unikátní vektor. Techniky lemmatizace a změna velkých písmen na malá by měly snížit počet unikátních tokenů, čímž navýší počet výskytů slov. Obě zároveň odstraňují některé informace o slovech (např. velká písmena u jmen či použitý čas slovesa), což naopak u rozsáhlých korpusů, ve kterých by i méně časté tvary slov mohly mít dostatečný počet využití, může působit negativně. Experimenty tedy prokáží, zdali užitek ze snížení počtu unikátních slov převýší množství odebíraných informací na různě velkých korpusech.

Některá velmi frekventovaná slova nemusí být příliš užitečná při tvorbě vektorů slov. Jejich odstraněním lze do kontextového okna zahrnout nové tokeny, které v něm jinak nejsou obsaženy. Na to je využita technika odstranění stop-slov, která tato častá slova odstraní. Nicméně i to může mít negativní vliv na výslednou kvalitu NER systému, neboť se tím ztrácejí vektorové reprezentace těchto stop-slov, které by však při učení neuronové sítě rozpoznávat entity mohly být užitečné.

Je také experimentováno se všemi různými kombinacemi zmíněných technik předzpracování. Je možné, že některé techniky se mohou doplňovat a tvořit tak synergický efekt.

Ve všech pozorováních je využit stejný korpus a učící algoritmus jako v předešlém experimentu. Pouze se změní počet dimenzí u vektorových reprezentací slov, který je nastaven podle výsledků předchozího experimentu. Korpus je tentokrát rozdělen pouze na tři podmnožiny, konkrétně na 50, 10 a 1 % jeho velikosti, což by mělo být dostatečné na odhalení vztahu technik předzpracování k rozsáhlosti textových dat.

3.2.4 Algoritmy a jejich nastavení

V tomto experimentu se nově využívají také algoritmy GloVe a FastText a technika CBOW algoritmu Word2Vec. Experiment si klade za cíl prozkoumat všechny tři algoritmy a jejich různá nastavení, která jsou zmíněná v sekci 3.1. Zjistí se tedy jak nejlepší nastavení každého z nich pro navržený NER systém, tak také jejich vzájemné porovnání. Je zde využit stejný trénovací korpus jako v minulém experimentu, včetně počtu dimenzí. Nově jsou na něj aplikovány kromě odstranění nealfanumerických znaků i ty techniky předzpracování, které z předešlého experimentu zvítězí.

Autoři GloVe ve své publikaci zmiňují (a i experimentem dokazují), že je algoritmus nejlepší volbou pro několik NLP úloh, jednou z nichž je rozpoznávání pojmenovaných entit (Pennington, Soche a Manning, 2014, s. 9). Avšak jiný experiment dokazuje, že v testování na korpusu analogií a v podobnosti dvou slov dosahuje lepších výsledků algoritmus Word2Vec (Levy, Goldberg a Dagan, 2015, s. 220). Tento experiment ověří, zdali je GloVe opravdu nejlepší volbou pro NER systém.

Technika Skip-Gram algoritmů Word2Vec a FastText lépe zachycuje sémantické podobnosti slov, CBOW naopak syntaktické, jak bylo dokázáno testováním modelů na korpusu analogií (Mikolov a kol., 2013b, s. 7). Otázkou však je, která z podobností slov má větší význam na kvalitnější rozpoznávání pojmenovaných entit. Odpověď na tuto otázku by měly přinést výsledky tohoto experimentu. Obě techniky se pro účely experimentu berou jako odlišné trénovací algoritmy, tudíž se obě nezávisle na sobě zúčastní stejných změn nastavení.

Algoritmy Word2Vec a GloVe neposkytují vektory pro slova, která nejsou obsažena v trénovacím korpusu. V navrženém NER systému se těmto slovům přidělí nulový vektor. Relativně nový algoritmus FastText si s tím umí do značné míry poradit sám. Díky znalosti všech možných n-gramů slov dokáže přiřadit vektor i neznámým slovům. To je značná výhoda, která může mít pozitivní vliv na výslednou kvalitu NER systému, neboť právě nulové vektory neznámých slov mohou značně ztížit trénování NER systému.

Nejdříve se u všech algoritmů experimentuje s velikostí kontextového okna hodnot 5, 10 či 15, při neměnném počtu epoch, který je stejně jako v předchozích experimentech roven 5. Následně se zjistí, zdali jiný počet epoch, konkrétně 1, 10 či 15, má vliv na konečné výsledky NER systému. Zde je naopak neměnná velikost kontextového okna, která je rovna 10.

Pro nejlepší nastavení algoritmů Word2Vec a FastText z těchto pozorování je dále provedena změna funkce poslední vrstvy neuronové sítě z hierarchického softmaxu na Negative Sampling, včetně experimentování s jejím počtem negativních vzorků, konkrétně 5 a 10. U techniky CBOW je navíc vyzkoušen postup, který namísto průměrování vektorů vstupních slov sčítá jejich hodnoty.

V algoritmu FastText lze navíc nastavovat minimální a maximální počty n-gramů. V experimentu jsou tedy zkoumány jejich různé kombinace, konkrétně minimální počet n-gramů roven 2 či 3 a maximální počet n-gramů roven 4 či 6, jako to podobně provádí autoři algoritmu Bojanowski a kol. (2016, s. 7–8).

Poslední z pozorování zkoumá různé hodnoty exponentu ve vážící funkci algoritmu GloVe. Při tom jsou využity poznatky z předešlých pozorování, které určí velikost kontextového okna a počet epoch. Ostatní jeho parametry trénování jsou nastaveny ve výchozích hodnotách.

3.2.5 Srovnání nejlepších nastavení jednotlivých algoritmů

Po zjištění nejlepších nastavení jednotlivých algoritmů je lze začít srovnávat navzájem na různě velkých trénovacích korpusech. Každý z algoritmů pracuje na jiném principu trénování, tudíž je možné, že se jejich vzájemná kvalita může lišit na různě rozsáhlých datech. Výstupem tohoto experimentu je tedy zjištění nejlepších algoritmů ze všech tří využívaných, včetně jejich konkrétních nastaveních na různě rozsáhlých trénovacích korpusech. Je využit stejný korpus jako v předešlých experimentech a je opět rozdělen na jeho různě velké podmnožiny, konkrétně na 50, 10 a 1 % jeho velikosti. Jednotlivé algoritmy využívají ve všech svých pozorováních ta nastavení, která se prokázala být nejlepšími v předešlém experimentu.

Autoři FastTextu tvrdí, že si algoritmus počíná oproti jiným daleko lépe na malých trénovacích datech a navyšování velikosti korpusu od určité úrovně již nemusí nutně vést ke zkvalitňování vektorových reprezentací (Bojanowski a kol., 2016, s. 6 až 7). Navíc s klesající velikostí korpusu se zvyšuje počet slov anotovaného zdroje pro NER, která nemají vektorovou reprezentaci. Jak již bylo zmíněno, FastText si s tím umí poradit. Je tedy pravděpodobné, že si povede o dost lépe na velmi malých korpusech právě díky oběma těmto vlastnostem. To však prověří experiment.

3.2.6 Aplikace Retrofitting metody

V tomto experimentu jsou všechny natrénované modely vnoření slov algoritmů[34] Word2Vec a FastText z předešlého experimentu dodatečně dotrénovány textem z anotovaného zdroje dat Extended CNEC 2.0 systému NER. Tomuto přístupu se říká tzv. „Retrofitting“ metoda, kterou již zmínila sekce 2.3.6. Hlavní výhodou, plynoucí z využití této metody, je eliminace nulových vektorů slov z anotovaného zdroje, která nejsou obsažena v trénovacím korpusu modelu vnoření slov. Anotovaný zdroj je však v poměru k využívaným trénovacím korpusům opravdu velmi malý (obsahuje 175 tisíc slov a 51 tisíc unikátních tokenů). Je tedy otázkou, zdali z tohoto důvodu bude mít metoda vůbec nějaký účinek. Dá se však předpokládat, že je vždy lepší alespoň nějaký vektor, ač nepříliš kvalitní, nežli nulový.

Vzhledem k tomu, že si FastText s neznámými slovy dokáže poradit, je otázkou, zdali má metoda v jeho případě smysl. Navíc metoda upraví i vektory n-gramů, čímž může potenciálně znehodnotit vytváření vektorů pro neznámá slova. Je tedy možné, že metoda bude mít dokonce i negativní vliv na FastText modely, a tedy i na výslednou kvalitu NER systému.

[34] Algoritmus GloVe bohužel neumožňuje aktualizovat jeho již jednou natrénované modely novými daty, tudíž je z experimentu vyřazen.

3.3 Implementace

Navržený systém NER (viz sekce 3.1) lze implementovat různými programovacími jazyky a knihovnami. Může se však stát, že některé knihovny programovacích jazyků mohou využité části v návrhu (např. tvorbu vektorových reprezentací slov či komponenty neuronových sítí) implementovat odlišným způsobem nebo i špatně. Proto je doporučeno pro identickou reprodukci výsledků využít stejné prostředky, které jsou využity v této práci, ale také dodržet naprosto stejnou architekturu neuronové sítě včetně hyper-parametrů.

Kompletní zdrojový kód, využívaný pro realizaci navrženého NER systému, je obsažen v elektronické příloze práce. Je obohacen komentáři pro snazší orientaci a pochopení funkcí. Soubory se zdrojovými kódy jsou *word_embeddings.py*, *corpus_utils.py*, *evaluation.py* a *NERBiLSTM.py*. Skript *word_embeddings.py* se využívá pro generování modelů vnoření slov pomocí algoritmů Word2Vec a FastText. Využívá řadu funkcí ze skriptu *corpus_utils.py*, které nabízejí techniky předzpracování a další práci s korpusy. Skript *NERBiLSTM.py* realizuje kompletní druhou část vývojového diagramu 5, tj. samotný systém NER od načtení dat, trénování, až po testování. V jeho kódu je využíván skript *evaluation.py*, který obsahuje řadu funkcí pro evaluaci kvality rozpoznávání.

Celý systém pro rozpoznávání entit je implementován v jazyce *Python 3.6.8*, který je v dnešní době pro realizaci úloh NLP velmi rozšířeným, díky čemuž nabízí spoustu užitečných balíčků. Je využita jedna z jeho volně dostupných distribucí, nazývaná *Anaconda*[35], jejíž součástí je mnoho předinstalovaných balíčků pro úlohy datové analýzy či strojového učení. Nabízí také nástroje pro snadnou instalaci a správu dodatečných balíčků (knihoven), včetně vytváření virtuálních prostředí.

3.3.1 Použité balíčky

Pro realizaci zmíněných technik předzpracování se využívá několik Python balíčků. Jedním z nich je balíček *stopwords*, který přináší seznamy stop-slov v mnoha jazycích, včetně češtiny. Dalším je *regex*, který se využívá pro odstranění nealfanumerických znaků. Lemmatizace je jazykově závislý úkon, pro jehož kvalitní práci je zapotřebí mít natrénovaný jazykový model, jenž umí českým slovům přiřazovat lemmata. V práci se využívá velmi kvalitní model *czech-morfflex-pdt-161115*, který je součástí balíčku MorphoDita[36].

Pro účely vytváření vektorových reprezentací slov pomocí algoritmů Word2Vec a FastText je využit Python balíček *Gensim 3.7.2*. V něm jsou tyto algoritmy implementovány v jazyce *Cython*[37], díky čemuž jsou časově velmi efektivní. Je tedy

[35] Více na `https://www.anaconda.com/`.

[36] Více na `http://ufal.mff.cuni.cz/morphodita`.

[37] Cython je programovací jazyk, který je v podstatě nadmnožinou jazyka Python. Lze v něm psát kód, který se kompiluje přímo v jazyce C, díky čemuž může být provádění algoritmu velmi rychlé. Zároveň však uživatelům ze svého kódu generuje balíček, který lze využívat v Pythonu. Uživatel tedy ani nemusí poznat, že je jádro algoritmu napsáno v jazyce Cython.

nutné mít rovněž nainstalovaný programovací jazyk Cython, ovšem některé Python distribuce, např. zmíněná Anaconda, jej již mají předinstalovaný. Obrovským benefitem tohoto balíčku je také fakt, že implementace algoritmu Word2Vec a FastText reflektují naprosto přesně původní implementace autorů. Gensim rovněž nabízí řadu různých funkcí pro práci s korpusy, načítání dat apod.

Pro GloVe je však využívána originální implementace[38] autorů v jazyce C, protože doposud neexistuje spolehlivá implementace v Pythonu, která by stoprocentně reflektovala tuto původní. Je tedy nutné mít nainstalovaný překladač programovacího jazyka C, kterým se originální implementace přeloží, čímž je připravena k použití.

Neuronová síť systému NER je implementována v komplexním Python balíčku *TensorFlow 1.12.0* od společnosti Google. Jedná se o velmi komplexní balíček, který obsahuje řadu nástrojů pro strojové učení, mezi něž patří rovněž komponenty neuronových sítí. Lze zde tedy najít např. již implementované architektury neuronových sítí, různé funkce vrstev, optimalizační algoritmy, ztrátové funkce apod. Lze volit mezi balíčkem, který své funkce počítá výhradně v procesoru a balíčkem, který je implementován pro využití v GPU. Některé architektury neuronových sítí mohou totiž být rychleji trénovány pomocí grafických karet. Při použití druhého z balíčků je však nutné splnit některé prerekvizity[39], jako je využití grafické karty výhradně od společnosti NVIDIA či instalace architektury CUDA.

Kompletní seznam použitých balíčků lze najít v souboru *requirements.txt* v elektronické příloze práce.

3.3.2 Použité technické vybavení

NER systém této práce je realizovaný na tomto technickém vybavení:

- operační systém Windows 10 Pro (Build 17134) a Ubuntu 15.10,
- procesor Intel Core i5-6500,
- grafická karta NVIDIA GeForce GTX 950,
- RAM paměť 32 GB DDR4,
- vývojové prostředí PyCharm 2019.1.1 (Professional Edition).

Operační systém Windows je využíván pro kompletní realizaci NER systému. Ubuntu se využívá pouze na trénování modelů vnoření slov pomocí algoritmu GloVe a prvotního očištění korpusů.

Doba vytváření vektorových reprezentací na tomto technickém vybavení je závislá na využitém algoritmu (Word2Vec<GloVe<FastText) a jeho nastavení (zejména na počtu epoch), technik předzpracování a také velikosti korpusu. V rámci navržených experimentů se pohybuje v rozmezí několika minut až hodin. Trénování NER systému bývá zpravidla dokončeno do jedné hodiny.

[38] Více na `https://nlp.stanford.edu/projects/glove/`.
[39] Více na `https://www.tensorflow.org/install/gpu`.

4 Výsledky

V této kapitole jsou představeny dosažené výsledky systému NER, o kterém pojednává kapitola 3. Nejdříve jsou popsány výsledky jednotlivých experimentů, které jsou navrženy v sekci 3.2. Poté je detailněji popsán nejlepší dosažený výsledek ze všech provedených realizací NER systému v experimentech, tedy konkrétní algoritmus pro vytváření vektorových reprezentací slov s jeho specifickým nastavením. Na závěr dojde k srovnání tohoto výsledku s výsledky existujících NER systémů jiných autorů.

4.1 Výsledky navržených experimentů

Následující sekce popisuje výsledky experimentů, které jsou navrženy v sekci 3.2. Výsledky jsou představovány postupně ve stejné posloupnosti jako návrhy, neboť většina experimentů navazuje na výsledky předešlých. U každého experimentu je zachycena tabulka s navrženými pozorováními. Hodnoty tabulky reprezentují kvalitu rozpoznávání systému NER, vyjádřenou F1-mírou, na testovací datové množině po 12. epoše trénování neuronové sítě.

Detaily, které ještě nebyly v době navrhování experimentů známy z důvodu jejich návaznosti, jako např. použitý korpus či hodnota některého nastavení, jsou zde u jednotlivých experimentů zmíněny. Rovněž je poskytnut detailnější popis důležitých zjištění a souvislostí, plynoucích z předkládaných výsledků. Pokud se v tabulce s výsledky objevuje výsledek z předešlého experimentu („baseline“), je v ní vyznačen buďto tučným zvýrazněním hodnoty, nebo doslovně.

Kompletní výsledky všech experimentů, včetně dílčích výsledků (testovaných na validační části trénovací množiny anotovaných dat) po jednotlivých epochách, se nacházejí v elektronické příloze práce. Tyto záznamy z experimentů rovněž obsahují detailní výsledky rozpoznávání jednotlivých typů entit po poslední epoše.

4.1.1 Trénovací datové zdroje

Výsledky tohoto experimentu demonstrují kvalitu systému NER při použití vektorových reprezentací slov, získaných z různých trénovacích korpusů. Experiment nemá jako jediný žádné návaznosti na jiné a všechny potřebné informace pro jeho realizaci zazněly již při návrhu.

Tabulka 4 demonstruje dosažené výsledky experimentu. Baseline výsledkem je zde neuronová síť bez vektorových reprezentací. V tomto případě systému nejsou poskytnuty žádné informace o vztazích mezi slovy, které by mohly pomoci systému v jeho rozhodování. Jediným zdrojem informací jsou tedy anotovaná trénovací data.

S korpusem EuroParl a extrahovaným textem z anotovaných dat Extended CNEC 2.0 dochází pouze k mírnému zlepšení oproti baseline. Problém zřejmě spočívá v poskytování nedostatku relevantních informací ve vektorech, které systému NER nijak zásadně nepomáhají. Důvodem neúčinných vektorových reprezentací slov

Tabulka 4: Výsledky F1-měr NER systémů při použití různých korpusů v modelu vnoření slov

Korpus	**F1-míra** [%]
Žádný (baseline)	49,83
EuroParl (česká část)	52,09
Text z Extended CNEC 2.0	52,54
Extrakce české Wikipedie	56,54
SYN-2015	61,31
CoNLL-2017 (česká část)	64,60
CZES	65,28
CWC-2011 – články	66,31

je u obou korpusů pravděpodobně velmi nízký celkový počet slov, kvůli čemuž se algoritmy modelů vnoření slov nedokáží nic podstatného naučit. V EuroParlu je sice několikanásobně více slov než v anotovaném zdroji, nicméně jedná se o odlišnou datovou doménu, která zachycuje pro NER systém bezcenné vztahy mezi slovy.

Překvapivé výsledky vykazuje extrakce české Wikipedie. Ačkoliv je podobně rozsáhlá jako korpus SYN-2015, dosahuje oproti němu pouze poloviničního nárůstu F1-míry od baseline výsledku. Možným vysvětlením rozdílných výsledků by mohlo být různorodé zaměření textů Wikipedie. Dále také fakt, že anotovaná data NER systému mohla být zčásti vytvořena z korpusu SYN-2015 (detailněji o tvorbě těchto anotovaných dat pojednává sekce 2.4.4).

Korpusy CoNLL-2017, CZES a CWC-2011 dosahují oproti ostatním poměrně vysokých výsledků. I bez jakékoliv snahy o přizpůsobení algoritmů vektorových reprezentací slov zde dochází k nárůstu kvality v porovnání s baseline o více než 15 %. Z těchto tří si však nejhůře stojí CoNLL-2017 i přes jeho dominanci v celkovém počtu slov. Pravděpodobným důvodem je několikanásobný počet unikátních tokenů oproti zbylým dvěma korpusům. To značí, že důležitá je nejen kvantita (celkové množství textu), ale také kvalita (minimum chyb). Právě kvalita může být od určitého rozsahu korpusu tím, co určuje rozdíl ve výsledcích.

Ani jeden z pozorovaných korpusů nevykazuje vyšší kvalitu NER systému než již natrénovaný korpus, který je využitý při hledání nejlepšího nastavení hyper-parametrů (viz sekce 3.1.2). To poukazuje na skutečnost, že i existující modely mohou NER systému přinášet vysokou přidanou hodnotu. Další experimenty však ukáží, zdali se může zkoumáním a úpravou některých faktorů dospět k výsledkům, které tento již natrénovaný model překonají.

Vítězem tohoto experimentu je korpus CWC-2011, díky čemuž bude využíván ve všech dalších experimentech. V některých z nich se budou využívat i jeho poměrné části, na což je korpus velmi dobře připraven. Jednou z technik předzpracování korpusu přímo od autorů totiž bylo náhodné promíchání vět, čímž by měla být zaručena jeho maximální reprezentativnost v rámci těchto podmnožin.

4.1.2 Dimenze vektorů a velikost korpusu

Výsledky předešlého experimentu dokazují, že výběr textového zdroje pro vytváření vektorových reprezentací slov může značně ovlivnit výslednou kvalitu systému NER. Vítězný korpus CWC-2011 se svou velikostí řadí mezi rozsáhlé textové zdroje. Je tak možné, že použitá dimenze 100 nemusí být dostačující pro zakódování důležitých vztahů mezi slovy. Tento experiment s korpusem CWC-2011 dále pracuje a zkoumá na něm (a jeho různě velkých částech), do jaké míry změna dimenzí ovlivňuje konečné výsledky systému NER.

Tabulka 5: Výsledky F1-měr NER systémů [%] při použití různých dimenzí a velikostí korpusu CWC-2011 v modelu vnoření slov (baseline výsledek je označen tučně)

Počet dimenzí	**Poměrná velikost korpusu**			
	1 %	**10 %**	**50 %**	**100 %**
50	53,84	59,14	63,98	62,55
100	54,56	61,14	64,83	**66,31**
200	55,17	62,79	66,35	67,29
300	56,63	63,78	67,93	67,95
400	55,70	65,51	67,68	68,33

Kompletní výsledky experimentu přináší tabulka 5. Rozdíly ve výsledcích mezi nejmenší a největší dimenzí jsou znatelnější u větších korpusů, ve kterých je zřejmě množství skrytých informací rozsáhlejší, a malé vektory tak nemusejí stačit k jejich kompletnímu zakódování. V nejmenší části není rozdíl již tak markantní, ale přesto každé navýšení dimenzí přináší zlepšení. Z výsledků je tedy patrné, že zvyšování počtu dimenzí většinou přináší zlepšení, nicméně mezi velikostmi 300 a 400 již nárůst stagnuje (dochází i naopak ke zhoršení). Dá se tedy konstatovat, že počet dimenzí z intervalu <300;400> je pro jakkoliv rozsáhlé korpusy dostatečný.

Zajímavé je i pozorování nárůstu F1-mír při zvětšování korpusu. Rozdíly v nárůstech postupně klesají s přibývající velikostí. Mezi velikostmi 1 % a 10 % dochází až k 10% nárůstu, zatímco mezi velikostmi 50 % a 100 % se výsledky již zásadně neliší. Lze tedy konstatovat, že velikost trénovacího korpusu modelu vnoření slov značně ovlivňuje kvalitu NER systému. Má smysl shánět co nejrozsáhlejší trénovací korpus, nicméně účinek ze zvětšování s přibývající velikostí logaritmicky degraduje. I přes vysokou flektivnost českého jazyka by tedy mohly být dostačující korpusy, které obsahují stovky milionů tokenů.

Nejlepší výsledky vykazuje nejrozsáhlejší korpus s největším počtem dimenzí. Oproti baseline (66,31 %) tedy může dojít k nárůstu i o více než 2 procenta (68,33 %). Není to sice mnoho, ale na druhou stranu, pokud není problém mít rozsáhlejší vektorový prostor, je škoda nevyužít této možnosti zlepšení NER systému.

Ve všech dalších experimentech je využíván počet dimenzí roven 300, protože dosahuje vynikajících výsledků napříč všemi velikostmi korpusu. V těch experi-

mentech, ve kterých se nemění velikost korpusu, se využívá pouze 50% část zdroje CWC-2011. Rozdíly ve výsledcích jsou totiž oproti 100% části takřka zanedbatelné. Navíc je doba trénování vektorových reprezentací slov poloviční. Tento počet dimenzí a velikost korpusu se tedy zdá být nejlepším kompromisem mezi dosaženými výsledky, časovou a prostorovou složitostí algoritmu pro trénování modelu vnoření slov a výslednou velikostí souboru.

4.1.3 Techniky předzpracování textu

V předešlém experimentu byly využívány různě velké části korpusu CWC-2011. Ty obsahují jiný počet unikátních tokenů, což má za následek rozdílný počet slov mimo slovník v NER systému. U menších korpusů mohou být následky tohoto problému značně redukovány technikami předzpracování textu. Doposud se využívala kromě odstraňování nealfanumerických znaků rovněž i změna velkých písmen na malá. Ta může v extrémním případě přinést až 50% redukci počtu slov bez vektorové reprezentace. V tomto experimentu jsou využity také další představené techniky předzpracování na různě velkých částech korpusu. Výsledky ukáží, jestli mají techniky (případně i jejich kombinace) vliv na výslednou kvalitu NER systému.

Tabulka 6: Výsledky F1-měr NER systémů [%] a počet slov mimo slovník (OOV) z celkových 51 092 při použití různých velikostí korpusu CWC-2011 a jejich technik předzpracování v modelu vnoření slov (LC = změna velkých písmen na malá, LM = lemmatizace, SW = odstranění stop-slov)

Technika předzpracování textu	**Poměrná velikost korpusu**					
	50 %	**10 %**	**1 %**	**50 %**	**10 %**	**1 %**
	F1-míra [%]			**Počet OOV slov**		
Žádná	68,70	65,94	56,96	4923	8982	21214
LC (baseline)	67,93	63,78	56,63	4173	7652	18665
LM	67,33	66,35	61,58	2744	4576	10699
SW	67,22	62,56	53,81	5367	9426	21645
LC + LM	66,81	66,22	60,80	2792	4628	10624
LC + SW	65,43	62,61	53,96	4617	8096	19107
LM + SW	68,27	64,38	58,25	2811	4662	10827
LC + LM + SW	66,85	63,23	56,39	2839	4701	10744

První část tabulky 6 zachycuje výsledky experimentu. Za baseline výsledky jsou označena ta pozorování, ve kterých je provedena pouze změna velkých písmen na malá. Už na první pohled je zřejmé, že mezi technikami v rámci jedné velikosti korpusu panují velké rozdíly (i více jak 7 % u 1% velikosti). To dokazuje, že zejména u menších korpusů mají techniky předzpracování textu významný dopad na výslednou kvalitu NER systému. Nejlepších výsledků v 1% a 10% části dosahuje technika

lemmatizace. Ta sice odebírá určité informace o slovech, jako je např. slovesný čas, nicméně užitek ze snížení počtu unikátních slov zdaleka převyšuje množství odebíraných informací.

Baseline výsledek, tedy změna velkých písmen na malá, si vede průměrně ve všech velikostech korpusu. Pokud se technika využije v kombinaci s jinými technikami, tak ve všech případech se výsledky zhorší oproti samostatnému využití těchto jiných technik. Z výsledků tedy plyne, že tato technika nemá hlubší opodstatnění ve využití. Avšak ani kombinace ostatních technik nepřinášejí větší rozdíly ve výsledcích. K synergickému efektu v kombinacích tedy vůbec nedochází.

Tímto experimentem však nelze tvrdit, že kromě lemmatizace jsou zbylé techniky zbytečné. Mohou nastat situace, ve kterých by se mohly hodit. Může se jednat o nekvalitní korpusy, ve kterých je snaha získat co největší počet výskytů unikátních tokenů i za cenu odebrání určitých informací z korpusu. To však není potřeba v datovém zdroji CWC-2011, neboť se jedná o velmi kvalitní textová data.

Druhá část tabulky 6 blíže ukazuje, jaký vliv na počet slov mimo slovník mají jednotlivé techniky předzpracování či jejich kombinace. Nejméně těchto slov po sobě zanechává technika lemmatizace (popř. i s kombinací změny velkých písmen na malá). Ta oproti některým jiným technikám dokáže zredukovat množství slov bez vektorové reprezentace na polovinu.

Odebírání stop-slov naopak množství slov bez vektoru oproti ostatním technikám navyšuje. Zároveň dosahuje nejhorších výsledků. Využití této techniky začíná mít smysl až s přibývající velikostí korpusu. Obstojných výsledků dosahuje v kombinaci s lemmatizací, avšak v menších korpusech samotnou lemmatizaci přesto nepřekoná.

Tyto rozdíly mezi počtem slov mimo slovník jsou zřejmě i odpovědí na tak velké rozdíly ve výsledcích. V největší části korpusu však již tento efekt minimalizace slov mimo slovník vyprchává a množství odebírané informace jednotlivými technikami jej začíná převyšovat. Proto v ní zřejmě dosahuje nejlepšího výsledku pozorování, ve kterém není uplatněna žádná z technik. To zřejmě proto, že se z korpusu nevytrácí žádné informace a zároveň je korpus již dostatečně robustní pro zachycení reprezentativních vektorů většiny slov (včetně jejich různých tvarů). V dalších experimentech, ve kterých se využívá pouze jedna velikost korpusu, nebude proto uplatňována žádná technika předzpracování.

4.1.4 Algoritmy a jejich nastavení

Doposud bylo v proběhlých experimentech využíváno jen jedno nastavení algoritmu Word2Vec. Je však pravděpodobné, že i různá nastavení tohoto algoritmu, ale také zcela jiné algoritmy, mohou mít vliv na výslednou kvalitu NER systému. Tento experiment všem třem pozorovaným algoritmům Word2Vec, GloVe a FastText nalezne nejlepší (nikoliv však nutně optimální) nastavení pomocí změn jejich některých parametrů učení.

Tabulka 7: Výsledky F1-měr NER systémů [%] při použití různých algoritmů pro vytváření modelů vnoření slov a různých hodnot kontextového okna a počtu epoch (baseline výsledek je označen tučně)

Algoritmus	**Velikost okna (počet epoch = 5)**			**Počet epoch (vel. okna = 10)**		
	5	**10**	**15**	**1**	**10**	**15**
Word2Vec (CBOW)	64,58	62,69	60,53	61,46	60,65	61,18
Word2Vec (Skip-Gram)	69,06	**68,70**	67,99	68,25	68,60	69,05
GloVe	63,90	64,46	64,54	60,43	63,04	65,00
FastText (CBOW)	66,01	63,91	64,39	64,33	63,62	63,62
FastText (Skip-Gram)	72,47	71,50	72,44	71,72	69,36	70,48

Tabulka 7 ukazuje výsledky experimentování s dvěma různými parametry učení, které lze měnit u všech tří zkoumaných algoritmů. První půlka tabulky zkoumá dopad změny velikosti kontextového okna na kvalitu NER systému při neměnném počtu epoch, druhá počet epoch při neměnné velikosti kontextového okna. Jak již výsledky napovídají, algoritmus FastText dominuje ve všech provedených pozorováních, ovšem tato informace není jediným výstupem tohoto experimentu.

Z výsledků vyplývá, že technika CBOW algoritmů Word2Vec a FastText je oproti technice Skip-Gram ve všech pozorováních o poznání horší. To značí, že je důležitější zaměřit se na kvalitnější sémantickou nežli syntaktickou podobnost slov[40].

Změna velikosti kontextového okna má na kvalitě NER systému větší vliv u techniky CBOW něž u Skip-Gram. Přesto u techniky Skip-Gram lze i touto změnou dospět k lepším výsledkům, avšak většinou s rozdílem do 1 procenta (oproti baseline je výsledek zlepšen z 68,70 % na 69,06 %).

Výsledky experimentování s různým počtem epoch u těchto dvou algoritmů nepřinesly markantní rozdíly. Ve většině případů se jedná o změny kvality do desetin procent. To může být také navíc zapříčiněno stochastickou povahou neuronové sítě v NER. Nejlepších výsledků algoritmy dosahují ve většině případů s nastavením epoch ve výchozím stavu, tj. 5. Nejlepším nastavením pro oba algoritmy Word2Vec i FastText je tedy využívání techniky Skip-Gram s velikostí kontextového okna a počtu epoch rovných 5.

Algoritmus GloVe oproti těmto dvěma algoritmům (v jejich variantě Skip-Gram) zaostává o více než 4 (Word2Vec), resp. 7 procent (FastText). Algoritmu sice svědčí vícero průchodů trénovacím korpusem, nicméně ani to mu nepomůže překonat některý z algoritmů (překonává pouze techniku CBOW). Rovněž ani experimentování s exponentem ve vážící funkci GloVe nepřineslo žádný nárůst F1-míry.

[40]To dává smysl, protože pro rozpoznávání entit je např. daleko přínosnější kvalitnější znalost sémantické podobnosti jmen (vektor slova „Martin“ je podobný vektoru slova „Tomáš“) než syntaktické podobnosti stupňování slov (vektory slova „hezký“ je podobný vektorům slov „hezčí“ a „nejhezčí“).

Změna z výchozí hodnoty 0,75 na 0,5 přinesla zhoršení takřka o 1 procento (na 64,04 %) a změna na 0,25 ještě více prohloubila zhoršování (na 62,98 %). Tvrzení autorů, že je GloVe vhodným algoritmem pro rozpoznávání pojmenovaných entit (Pennington, Soche a Manning, 2014, s. 9) se tímto experimentem tedy nepotvrdilo.

Tabulka 8: Výsledky F1-měr NER systémů [%] při změně funkce poslední vrstvy neuronové sítě algoritmů Word2Vec a FastText a výpočtu vektorů v CBOW technice (oba algoritmy mají ve všech pozorováních nastavenou hodnotu velikosti kontextového okna a počtu epoch na 5)

Algoritmus	**Negative Sampling s počtem neg. vzorků**		**CBOW součtem vektorů**
	5	**10**	
Word2Vec (CBOW)	62,64	63,93	66,62
Word2Vec (Skip-Gram)	70,37	69,51	–
FastText (CBOW)	62,30	61,49	64,19
FastText (Skip-Gram)	72,32	70,82	–

První část tabulky 8 popisuje experimentování se změnou funkce poslední vrstvy z hierarchického softmaxu na Negative Sampling nejlepších získaných nastavení algoritmů Word2Vec a FastText z předchozího kroku (baseline výsledek se tedy nachází v prvním sloupci výsledků tabulky 7). V případě Word2Vec Skip-Gram tato změna vede k lepšímu výsledku (70,37 % oproti původních 69,06 %). Tím se toto nastavení stává pro tento algoritmus nejlepším. U FastTextu nedochází ani v jednom z pozorování ke zlepšení, tudíž nejlepším nastavením zůstává výsledek baseline.

Druhá část tabulky experimentuje v technice CBOW se změnou způsobu výpočtu vstupních vektorů, a to z průměru na součet. U algoritmu Word2Vec dochází k nárůstu z 64,58 % na 66,62 %, přesto však ani tento výsledek nepřekoná techniku Skip-Gram v jakémkoliv nastavení. U FastTextu naopak oproti Word2Vec dochází ke zhoršení, zřejmě z důvodu jiného principu učení, založeném na n-gramech slov.

Tabulka 9: Výsledky F1-měr NER systémů [%] při změně rozsahu n-gramů algoritmu FastText (baseline výsledek je označen tučně)

Min. počet n-gramů	**Max. počet n-gramů**	
	4	6
2	71,79	72,05
3	71,56	**72,47**

Tabulka 9 popisuje výsledky při změnách rozsahu n-gramů v algoritmu FastText. Baseline výsledek zde má nastavenou hodnotu minimálního počtu n-gramů na 3 a maximálního na 6, což se doposud využívalo ve všech pozorováních tohoto algoritmu. Žádná ze změn rozsahu n-gramů však nevede ke konečnému zlepšení NER systému, tudíž výchozí nastavení rozsahu zůstává nadále tím nejlepším.

4.1.5 Srovnání nejlepších nastavení jednotlivých algoritmů

Tento experiment shrnuje poznatky ze všech předešlých. Z prvního využívá korpus CWC-2011, který dosahuje nejlepších výsledků. Ze druhého počet dimenzí, který je roven 300. Na 1% a 10% část korpusu je uplatněna rovněž i lemmatizace, což dle výsledků třetího experimentu vychází jako nejlepší varianta. Ve čtvrtém experimentu bylo zjištěno pro každý ze tří pozorovaných algoritmů nejlepší nastavení. Jednotlivá pozorování tohoto experimentu aplikují zmíněné poznatky a zkoumají tyto tři algoritmy v různě rozsáhlých korpusech.

Tabulka 10: Výsledky F1-měr NER systémů [%] nejlepších nastavení algoritmů při použití různých velikostí korpusu CWC-2011 a lemmatizace (baseline výsledek je označen tučně)

Algoritmus	**Bez lemmatizace**				**S lemmatizací**	
	Poměrná velikost korpusu					
	100 %	**50 %**	**10 %**	**1 %**	**10 %**	**1 %**
Word2Vec	70,66	**70,37**	66,02	53,78	66,89	59,11
GloVe	63,54	**65,00**	60,12	52,62	58,54	53,44
FastText	71,69	**72,47**	71,12	66,51	70,34	66,42

V první části tabulky 10 jsou výsledky algoritmů bez aplikace lemmatizace na trénovací korpus vektorových reprezentací slov, v druhé části pak s její aplikací. Z tabulky lze vyčíst, že lemmatizace má opravdu velký vliv na výslednou kvalitu NER systému, avšak pouze v 1% části korpusu na algoritmy Word2Vec a GloVe a v 10% části na algoritmus Word2Vec. FastText si díky svému principu trénování dokáže poradit i s menším počtem výskytů slov v kontextu, neboť se učí z jejich n-gramů. Dokonce při použití lemmatizace u něj dochází ke zhoršení, protože se odebírá část informace (prefixy, infixy či postfixy slov), se kterou umí efektivně nakládat.

Rozdílné velikosti trénovacího korpusu nejvíce ovlivňují Word2Vec (se zhruba 17% rozdílem mezi nejlepším a nejhorším výsledkem), FastText naopak nejméně (necelým 6% rozdílem). Experiment tedy potvrzuje tezi autorů FastTextu (Bojanowski a kol., 2016, s. 6–7), že si algoritmus oproti jiným počíná daleko lépe na malých trénovacích datech. Rovněž se prokazuje jejich další teze, že navyšování velikosti korpusu od určité úrovně nutně nemusí vést ke zkvalitňování vektorových reprezentací (Bojanowski a kol., 2016, s. 7). Algoritmu v podstatě pro kvalitní vektorové reprezentace dostačuje 10% podmnožina korpusu CWC-2011. Mezi velikostmi 50 % a 100 % dokonce dochází k mírnému poklesu. Přesto však dosahuje oproti ostatním algoritmům nejlepších výsledků ve všech velikostech korpusu. Bez nadsázky lze tedy konstatovat, že se jedná o nejlepší algoritmus, jehož vektorové reprezentace slov dosahují nejvyšších výsledků v NER systému pro český jazyk. Navíc v případě nedostupnosti rozsáhlých korpusů dokáže algoritmus přesto zaručit vysokou kvalitu naučených vektorů oproti zbylým dvěma algoritmům.

Zajímavé je rovněž pozorovat výsledky algoritmů v rámci stejně velkého korpusu. Ve všech velikostech sice dominuje FastText, nejvíce je to však znatelné při menších korpusech. U 1% podmnožiny korpusu s využitím lemmatizace je rozdíl více než 7 %, bez využití lemmatizace pak o více než 13 %. Word2Vec začíná konkurovat algoritmu FastText až při 50% velikosti korpusu. GloVe je ve srovnání s nejlepšími výsledky těchto dvou algoritmů v propadu o více než 5 %.

4.1.6 Aplikace Retrofitting metody

V tomto experimentu je zkoumán dopad aplikace metody „Retrofitting“ na konečné výsledky systému NER. Vytvořené modely vektorových reprezentací slov pomocí algoritmů Word2Vec a FastText z předešlého experimentu jsou dodatečně natrénovány extrahovaným textem z anotovaných dat pro NER systém. U obou trénovacích algoritmů se zakáže technika Sub-sampling a nastaví počet výskytů tokenu na 0, díky čemuž jsou všechna slova z extrahovaného textu zahrnuta do trénování. Tím se ve všech velikostech korpusu odstraní slova mimo slovník.

Tabulka 11: Výsledky F1-měr NER systémů [%] po aplikaci Retrofitting metody na nejlepší nastavení algoritmů při použití různých velikostí korpusu CWC-2011

Algoritmus	**Bez lemmatizace**				**S lemmatizací**	
	Poměrná velikost korpusu					
	100 %	**50 %**	**10 %**	**1 %**	**10 %**	**1 %**
Word2Vec	69,11	69,94	64,20	54,41	66,04	60,69
FastText	65,46	66,60	66,17	65,81	68,20	68,52

Baseline výsledky pro všechna pozorování jsou výsledky algoritmů Word2Vec a FastText v tabulce 10. Z tabulky 11, zachycující výsledky tohoto experimentu, vyplývá, že tuto metodu má smysl aplikovat pouze v případě využití velmi omezeného trénovacího korpusu pro vektorové reprezentace slov (zde u 1% části), a to společně s lemmatizací. Bez lemmatizace dochází ke zlepšení pouze u algoritmu Word2Vec na nejmenší části korpusu, u FastTextu již dochází ke zhoršení.

FastText, jak již bylo dokázáno v předešlém experimentu, si dokáže velmi dobře poradit i s malými korpusy. Retrofitting metoda zhoršuje jeho výsledky, o to více pak ve větších korpusech (o více než 5 procent). Možným vysvětlením může být zmíněné tvrzení v podsekci 3.2.6. To říká, že dodatečné trénování novými daty mění vektory většině n-gramů, které se tak přizpůsobí na tato nová data. Těchto nových dat však není mnoho (pouze 175 tisíc slov), dochází tak naopak k znehodnocení vektorů n-gramů, které byly trénovány na velmi rozsáhlých datech (čítající stovky milionů slov). Tím se zhorší kvalita vektorů pro slova mimo slovník. Právě možnost vytvoření kvalitních vektorových reprezentací pro tato slova je největší předností algoritmu FastText a jejím znehodnocením proto pravděpodobně dochází ke zhoršení.

4.2 Zhodnocení nejlepšího výsledku

Absolutním vítězem ze všech tří algoritmů je FastText. Algoritmus vyhrává ve všech velikostech testovaného korpusu CWC-2011, avšak jeho dominance je nejvíce zřetelná v menších částech korpusu. Nejlepšího výsledku dokonce nedosahuje v jeho největší části, nýbrž v poloviční, s celkovou F1-mírou 72,47 % na testovací části anotovaných dat. Detailní metriky rozpoznávání jednotlivých typů entit jsou demonstrovány v následující tabulce:

Tabulka 12: Kvalita rozpoznávání jednotlivých typů entit (v korpusu Extended CNEC 2.0) NER systému s nejlepším dosaženým výsledkem v experimentech, vyjádřená přesností („precision“), pokrytím („recall“) a F1-mírou, včetně počtu systémem predikovaných entit a celkového počtu entit v korpusu

Typ entity	**Přesnost** [%]	**Pokrytí** [%]	**F1-míra** [%]	**Predik. entit**	**Celkem entit**
Čísla a adresy	90,38	85,45	87,85	52	55
Geografická označení	73,48	76,98	75,19	396	378
Názvy institucí	59,83	65,74	62,65	356	324
Názvy médií	66,67	58,33	62,22	42	48
Názvy artefaktů	47,78	47,91	47,84	383	382
Jména osob	77,49	86,04	81,54	533	480
Časová vyjádření	89,63	91,58	90,59	376	368
Celkem	**70,72**	**74,30**	**72,47**	**2138**	**2035**

Nejlépe systém rozpoznává časová vyjádření. Často se jedná o názvy měsíců či dnů, jejichž vektorové reprezentace jsou si velmi podobné (viz tabulka 13). Naopak nejhůře systém rozpoznává názvy artefaktů (např. měrné jednotky, jednotky měn, normy a direktivy či produkty). Tento typ entit je však v textu někdy velice těžko rozpoznatelný i člověkem, neboť pokrývá velmi širokou oblast. Často je také entita složena z několika tokenů a je možné, že neuronová síť nerozpozná všechny její části.

Ze standardních typů entit si systém nejlépe vede v rozpoznávání osob a geografických označení. Většinou se jedná o entitu skládající se obvykle z 1 až 3 tokenů. Rozpoznávání není tedy tak problematické jako u jiných typů, ve kterých mají entity hned několik tokenů. Navíc vektorové reprezentace slov těchto typů, jako např. měst či jmen, si bývají velmi podobné (viz tabulka 13).

Naopak nejhůře si systém počíná v rozpoznávání institucí. Jejich unikátních názvů je velmi mnoho a v trénovacím korpusu pro trénování vektorových reprezentací nejsou většinou zmíněny tolikrát (jako např. jména či města), aby se začaly tvořit užitečné podobnosti ve vektorech (jak naznačuje tabulka 13). Pomoci v rozpoznávání by mohly např. jazykově závislé informace o slovech.

Oproti nejlepšímu výsledku algoritmu Word2Vec (jeho detailní vyhodnocení rozpoznávání lze nalézt v příloze, tabulka 15) dokáže FastText daleko lépe rozpo-

Tabulka 13: Ukázka pěti nejbližších slov ve vytvořeném vektorovém prostoru ke zvoleným slovům (blízkost je měřena kosinovou podobností, jejíž hodnota je uvedena v závorkách)

leden	**pondělí**	**Tomáš**	**Brno**	**ČKD**
březen (0,90)	úterý (0,96)	Marek (0,70)	Opava (0,79)	Strojírny (0,72)
duben (0,88)	pátek (0,92)	Jakub (0,66)	Ostrava (0,78)	Strojírna (0,70)
říjen (0,87)	neděli (0,84)	Michal (0,62)	Olomouc (0,75)	strojírny (0,69)
únor (0,85)	sobotu (0,83)	Pavel (0,61)	Prostějov (0,74)	strojírně (0,68)
červen (0,85)	čtvrtek (0,68)	Lukáš (0,59)	Kroměříž (0,73)	strojírna (0,67)

znávat čísla a adresy (87,85 % F1-míry oproti 70,91 %). Většinu entit tohoto typu zastupují numerická vyjádření, jako např. telefonní čísla, pro která se velmi těžko hledají kvalitní vektorové reprezentace (pro spoustu z nich ani nemusejí existovat). FastText však čísla v trénovacím korpusu rozdělí na n-gramy, díky čemuž dokáže poskytnout většině čísel vektor, který bude velmi podobný vektorům ostatních čísel. Tato informace může být pro zlepšení rozpoznávání numerických vyjádření klíčová.

FastText si oproti Word2Vec také dokáže lépe poradit s názvy médií (62,22 % F1-míry oproti 57,47 %). Pod tímto typem entity se vyskytuje také mnoho e-mailových adres. Většina z nich však v případě Word2Vec nemá přidělenu vektorovou reprezentaci, neboť je velmi nepravděpodobné, že se adresa objeví rovněž i v trénovacím korpusu modelu vnoření slov. FastText však těmto adresám dokáže poskytnout vektor, který může pomoci neuronové síti v rozpoznávání tohoto typu entit. Potenciálním řešením pro algoritmus Word2Vec by mohla být technika předzpracování, která by všem e-mailovým adresám přiřadila v trénovacích korpusech jednotný speciální token (např. „<MAIL>“), díky čemuž by vznikla jediná vektorová reprezentace pro všechny adresy. Přesto by však samotné zlepšení rozpoznávání adres o moc nezvýšilo konečnou kvalitu NER systému, neboť celkový počet entit typu „Názvy médií“ je v testovací datové množině velmi nízký (48 výskytů).

Pro některé typy entit by se však místo neuronové sítě hodil více přístup, který je založený na pravidlech. Ten by mohl jednoduše pomocí regulárních výrazů nalézt např. zmíněná telefonní čísla a další numerická vyjádření či e-mailové adresy. Tento typ implementace však není předmětem této práce.

4.3 Srovnání výsledků s existujícími NER systémy

Tato sekce srovnává dosažené výsledky s existujícími systémy NER jiných autorů, které byly představeny v sekci 2.4.5. Do porovnávání jsou však zahrnuty jen ty systémy, které využívají neuronové sítě v kombinaci s modely vnoření slov. Systémy musejí navíc poskytovat ve svých publikacích výsledky na korpusu Extended CNEC 2.0, kterých dosahují pouze s využitím těchto modelů vnoření slov a žádných dalších vstupů, jinak by srovnání postrádalo smysl. Předkládané výsledky tedy nejsou těmi nejlepšími, kterých systémy dokáží dosahovat (ty jsou zachyceny v tabulce 2). Těch dosahují až při poskytnutí dalších informací na vstup jejich neuronových sítí.

Tabulka 14 srovnává tři NER systémy této práce. První z nich používá pouze korpus anotovaných dat bez využití modelu vnoření slov. Představuje baseline výsledek v prvním experimentu (viz sekce 4.1.1). Druhý systém reprezentuje nejlepší dosažený výsledek experimentu s hyper-parametry (viz sekce 3.1.2). Využívá již natrénovaný model české části rozsáhlého korpusu CoNLL-2017 technikou Word2Vec Skip-Gram. Třetí z nich vznikl v experimentu 4.1.4. Využívá model vnoření slov vytvořený algoritmem FastText a jedná se o nejlepší dosažený výsledek této práce.

Tabulka 14: Srovnání F1-měr (testovací části korpusu Extended CNEC 2.0) výsledků této práce s existujícími NER systémy, které využívají neuronové sítě jen s modely vnoření slov

NER systémy této práce	**F1-míra** [%]
Bez modelu vnoření slov	49,83
Výsledek experimentu s hyper-parametry	68,16
Nejlepší nastavení algoritmu FastText	72,47
NER systémy jiných autorů	**F1-míra** [%]
Demir a Özgür (2014)	64,72
Straková, Straka a Hajič (2016)	63,91

Začleněním modelu vnoření slov se navýší kvalita ve druhém systému této práce o více než 18 %. Vektory slov tedy dokáží značně ovlivnit výslednou kvalitu rozpoznávání. Přitom se však jedná o již natrénovaný model, který je volně dostupný a ihned připraven k použití. Ovšem pro ještě větší navýšení kvality NER systému je nutné vytvořit vlastní model vnoření slov s využitím vhodného korpusu, technik předzpracování a algoritmu. Zkoumáním všech těchto faktorů a jejich následnou optimalizací lze dospět k dalšímu navýšení o více než 4 % (z 68,16 % na 72,47 %).

Systém autorů Demira a Özgüra (2014) využívá pro vytváření vektorových reprezentací slov rozsáhlý český korpus, čítající 636 milionů slov a 906 tisíc unikátních tokenů, a dále algoritmus Word2Vec Skip-Gram s počtem dimenzí 200 a velikostí kontextového okna 5. Jejich korpus je velmi podobný využívanému korpusu CWC-2011 (alespoň co do počtu slov). Přesto však nejlepší výsledek této práce překonává systém o více než 7,5 %. Kromě rozdílných vektorů slov může konečné výsledky NER systému ovlivňovat i použitá architektura sítě. Je však nepravděpodobné, že by sama dokázala vytvořit tak markantní rozdíl.

Druhý systém Strakové, Straky a Hajiče (2016) využívá pro vytvoření vektorových reprezentací slov stejný algoritmus i nastavení jako předešlý. Rozdíl je pouze ve využitém korpusu. Tento systém využívá velmi rozsáhlý korpus SYN, čítající miliardy slov. Ovšem ani využití takto rozsáhlého korpusu nepomáhá k navýšení kvality rozpoznávání. Systém je překonán nejlepším výsledkem této práce o zhruba 8,5 %.

Výsledky tohoto srovnání naznačují, že by měl být prostor pro zlepšení celkové kvality rozpoznávání těchto existujících NER systémů pouhým využitím kvalitnějších vektorových reprezentací slov. Tím by pro zlepšování výsledků odpadla nutnost měnit samotnou architekturu systémů.

5 Shrnutí práce

Tato práce se zabývala rozpoznáváním pojmenovaných entit (NER) v českém textu, o čemž pojednává sekce 2.4. Jedná se o esenciální úlohu v oblasti zpracování přirozeného jazyka (NLP), která si klade za cíl z textu extrahovat význačná slova či slovní spojení (pojmenované entity) a klasifikovat je do předem určených kategorií (typů). Úloha má mnoho praktických uplatnění (viz sekce 2.4.3) a lze ji realizovat různými přístupy, které byly popsány v sekci 2.4.2.

Cílem práce bylo implementovat systém NER za použití technik strojového učení, o kterých pojednává sekce 2.2. V tomto systému, jehož návrh je popsán v sekci 3.1, se využívá rekurentní neuronová síť s buňkami LSTM (o neuronových sítích pojednává sekce 2.2.3). Přesné nastavení jejích hyper-parametrů bylo zjištěno experimentem, jehož výsledky jsou popsány v sekci 3.1.2. Implementace systému je popsána v sekci 3.3 a zdrojové kódy jsou součástí elektronické přílohy práce.

V roce 2013 byla zveřejněna publikace, která představuje nový způsob tvorby vektorových reprezentací slov z korpusu (Mikolov a kol., 2013b). Tyto vektory v sobě mohou mít zakódované skryté vztahy mezi slovy, která sdílejí společný kontext, jako například syntaktické či sémantické podobnosti slov. Tato publikace podnítila tvorbu dalších algoritmů pro vytváření těchto vektorů. Nejznámějšími a zároveň nejpoužívanějšími jsou moderní algoritmy Word2Vec, GloVe a FastText. O této problematice, včetně detailního popisu všech tří algoritmů, pojednává sekce 2.3.

Vektorové reprezentace slov, vytvořené moderními algoritmy, se od té doby staly nedílnou součástí NLP úloh, včetně nejlepších NER systémů. Výjimkou není ani implementovaný NER systém této práce, který rovněž využívá tyto reprezentace jako svůj hlavní zdroj znalostí o vstupech. Data využívaná pro vytváření vektorových reprezentací slov a pro trénování systému NER jsou popsána v sekci 3.1.1.

Dílčím, avšak neméně důležitým, cílem práce bylo zjistit míru přínosů těchto vektorů slov při jejich využití NER systémem a dále prozkoumat možné dopady na kvalitu rozpoznávání při použití různých modelů těchto vektorů. Pro naplnění cílů práce tedy byla navržena sada experimentů, které detailněji popisuje sekce 3.2. Jednotlivé experimenty se postupně snažily prozkoumávat různé faktory, které by mohly mít vliv na kvalitu vytvořených vektorových reprezentací slov, a tedy nepřímo i na výslednou kvalitu NER systému. Jejich výsledky byly interpretovány v sekci 4.1.

5.1 Shrnutí experimentů

První z experimentů (viz sekce 3.2.1) zkoumal vliv výběru trénovacích datových zdrojů na modely vektorových reprezentací. Bylo zkoumáno několik různě rozsáhlých a jinak kvalitních korpusů, které jsou detailněji popsány v sekci 2.3.7. Ty se využily pro vytvoření vektorových reprezentací slov pomocí algoritmu Word2Vec Skip-gram. Výsledky dokázaly (viz sekce 4.1.1), že volba korpusu značně ovlivňuje kvalitu vektorů, a tím nepřímo i výslednou kvalitu NER systému až o několik procent. Z dosažených výsledků vyplývá, že pro maximální nárůst kvality NER systému

je nutno hledět nejen na kvantitu korpusu (celkové množství textu), ale také na jeho kvalitu (minimum chyb) a doménové zaměření dat.

Druhý experiment (viz sekce 3.2.2) využil poznatky z předešlého a korpus s nejlepším výsledkem dále podrobil řadě pozorování. Zkoumal vliv změny dimenzí vektorů na jeho různě velké části. Z výsledků (viz sekce 4.1.2) plyne doporučení, že je obecně výhodnější využívat větší počet dimenzí, a to v jakýchkoliv velikostech korpusu. Nicméně tento benefit z nárůstu dimenzí přestává platit při počtu vyšším než 300. Výsledky dále značí, že zvolený počet dimenzí je o to důležitější v rozsáhlejších korpusech, kde jsou rozdíly markantnější. Vedlejším zjištěním tohoto experimentu je fakt, že pro vytvoření dostatečně kvalitních vektorů slov českého jazyka postačuje korpus čítající několik málo stovek milionů slov.

Ve třetím experimentu (viz sekce 3.2.3) byl ten samý korpus ještě před jeho využitím předzpracován několika technikami. Jejich cílem je upravit jeho text tak, aby výsledné vektorové reprezentace přinesly systému NER užitečnější informace. Výsledky dokázaly (viz sekce 4.1.3), že většinu technik je vhodné využít spíše v méně rozsáhlých korpusech, některé pak nemá smysl využívat vůbec. Nejužitečnější technikou v menších korpusech je lemmatizace, která zároveň nejvíce ze všech redukuje množství slov bez vektorové reprezentace v NER systému. Z výsledků lze dále vyčíst, že vyšší počet slov bez vektoru má většinou negativní vliv na výslednou kvalitu NER systému, a je tedy vhodné jej do určité míry snižovat.

Všechny předešlé experimenty využívaly algoritmus Word2Vec. Ve čtvrtém experimentu (viz sekce 3.2.4) byly prozkoumány další představené algoritmy GloVe a FastText. U algoritmů Word2Vec a FastText má podle výsledků (viz sekce 4.1.4) smysl využívat pouze techniku Skip-Gram, která NER systému vytváří vektory s kvalitněji zakódovanou sémantickou podobností slov. Pro všechny tři algoritmy byl dále zkoumán vliv změn jejich specifických parametrů. Výstupem experimentu bylo tedy nalezení nejlepších nastavení všech tří algoritmů.

V předposledním experimentu (viz sekce 3.2.5) se všechny získané poznatky z předešlých využily na zjištění, jak si jednotlivé algoritmy se svými nejlepšími parametry vedou v různě rozsáhlém korpusu. Z výsledků (viz sekce 4.1.5) je patrné, že algoritmus FastText je vhodný pro jakkoliv rozsáhlá data. Jeho dominance je však více viditelná v méně rozsáhlých korpusech. Algoritmu rovněž pro vytvoření kvalitních vektorů stačí daleko menší korpus než u zbylých algoritmů. Word2Vec je velmi závislý na velikosti korpusu a algoritmu FastText začíná konkurovat až ve velmi rozsáhlých korpusech. Algoritmus GloVe se v porovnání s nimi neprojevil jako příliš užitečný pro tvorbu vektorových reprezentací slov, využívaných v systému NER.

V posledním experimentu (viz sekce 3.2.6) se modely natrénované algoritmy Word2Vec a FastText dodatečně dotrénovaly daty ze zdroje anotovaných dat pro NER systém. Tato metoda se nazývá „Retrofitting“ a byla popsána v sekci 2.3.6. Jejím hlavním benefitem je obohacení modelu o slova bez vektorové reprezentace. Výsledky (viz sekce 4.1.6) však ukazují, že metoda nemá příliš velké uplatnění. Jediné nárůsty kvality NER systému byly zaznamenány v nejmenší části korpusu s využitím lemmatizace. Algoritmům pak ve větších korpusech metoda spíše škodí.

5.2 Diskuse

V této práci byla provedena série experimentů, jejichž výsledky mohou být do jisté míry závislé na využitých postupech a technikách. Ve všech experimentech se používal jeden konkrétní přístup systému NER, využívající specifikou architekturu neuronové sítě, konkrétně rekurentní neuronovou síť s jednou skrytou vrstvou, která využívá LSTM buňky. Otázkou však je, do jaké míry se mohou získané poznatky zobecnit i na jiné architektury. Nicméně srovnání dosažených výsledků v sekci 4.3 představuje dva existující systémy jiných autorů, ve kterých každý z nich využívá jinou architekturu neuronové sítě. Přesto jsou si jejich výsledky velmi podobné. Z toho je patrné, že zvolená architektura sítě sice může ovlivnit celkové výsledky rozpoznávání, nicméně konečné poznatky z experimentů by neměla ovlivnit. Měly by tedy být zobecnitelné na jakýkoliv typ neuronové sítě.

V prvních čtyřech experimentech se využívaly modely vektorových reprezentací slov, které byly vytvořeny algoritmem Word2Vec s technikou Skip-Gram a specifickým nastavením. Je však možné, že by některé výsledky mohly vyjít rozdílně při zopakování těchto experimentů s využitím zbylých dvou algoritmů. Tyto algoritmy sice pracují na odlišných principech (viz sekce 2.3.1), nicméně přesto sdílí většinu svých parametrů. Výsledky by tedy mohly být do jisté míry odlišné, nicméně ne natolik, aby se změnily konečné poznatky.

Pro potvrzení těchto zmíněných tezí by však bylo nutné zopakovat některé experimenty této práce s využitím jiných prostředků (jinou architekturou neuronové sítě či využitím všech algoritmů v prvních čtyřech experimentech). To se však v této práci neuskutečnilo z důvodu časové náročnosti.

Jeden z experimentů zkoumal různé parametry nastavení jednotlivých algoritmů. Nutno podotknout, že se nejedná o úplné pokrytí všech parametrů, které algoritmy dovolují nastavovat. Kvůli časovému limitu byly vybrány ty parametry, které by měly mít největší vliv na výslednou kvalitu modelů. Zbylé parametry byly nastaveny v doporučených hodnotách (samotnými autory či jinými pracemi). Je ovšem možné, že existuje určitá kombinace konfigurací, která by dosáhla ještě vyšších výsledků či přinesla další nové poznatky.

Provedené experimenty byly mezi sebou provázány, a tím i do určité míry na sobě závislé. Změna výsledků některého z nich by pak mohla vyvolat řetězovou reakci změn dalších. Bez těchto návazností by však nebylo časově možné realizovat všechny kombinace provedených pozorování, neboť by jich vzniklo velmi mnoho (s těmito návaznostmi byla provedena stovka experimentů a bez nich by byl počet ještě mnohonásobně vyšší).

Tato práce zkoumala dopad kvality vektorových reprezentací slov na kvalitu NER systému, což je velmi specifická úloha NLP. Nelze tak generalizovat výsledky na jiné úlohy NLP, které rovněž využívají vektory slov. Je možné, že pro jejich navýšení kvality bude potřeba zvolit jiný přístup k vytváření vektorů. Jisté však je, že má smysl se zabývat způsoby vytváření vektorových reprezentací slov, neboť ty mohou v konečném důsledku ovlivnit nepřímo i kvalitu úloh NLP.

Výsledky Retrofitting metody poměrně zklamaly. Až na velmi malé trénovací korpusy působí metoda spíše negativně. Mohla by se však aplikovat na modely vektorových reprezentací i jiným způsobem. Namísto dodatečného trénování modelů na nových datech by se tato data zahrnula do trénovacího korpusu ještě před samotným vytvořením modelů. Otázkou však je, zdali by tento přístup měl nějaký efekt, neboť např. využívaný korpus Extended CNEC 2.0 je v porovnání s jinými korpusy mnohonásobně menší, čímž by jeho text nemusel mít žádný vliv na vytváření modelů. Tento přístup však již nebylo možné z časových důvodů realizovat.

5.3 Závěr

Všechny stanové cíle práce byly splněny. Byl implementován systém NER pro český jazyk s technikami strojového učení, který využívá moderní vektorové reprezentace slov pro své kvalitnější rozpoznávání. Rovněž byla zjištěna míra přínosu těchto reprezentací v systému NER a dále, prostřednictvím experimentů, prozkoumány některé faktory, které ovlivňují jejich výslednou kvalitu a množství zakódovaných informací.

Práce však nepředkládá univerzální návod na vytváření vektorových reprezentací slov, jejichž využití automaticky navýší kvalitu systému NER. Spíše poukazuje na fakt, že využití kvalitních vektorových reprezentací slov dokáže významně zlepšit rozpoznávání pojmenovaných entit. Má tedy smysl se zabývat jejich zdokonalováním. Provedené experimenty v této práci postupně zkoumaly možnosti zkvalitňování těchto modelů. Přinesly užitečné poznatky, které lze zobecnit a aplikovat na jakékoliv jiné modely. Po uplatnění poznatků by tyto modely mohly poskytovat užitečnější informace již existujícím NER systémům, což může zlepšit jejich konečnou kvalitu[41] bez změny principu rozpoznávání. To by mohlo být motivací pro budoucí práci, ve které by se poznatky aplikovaly na dosavadní nejlepší systémy NER pro český jazyk.

Poznatky této práce vycházejí ze systému NER pro **český jazyk**. Ty se však dají využít i v dalších jazycích. Otázkou však je, jestli by zkoumané faktory v daném jazyce (např. v méně flektivní angličtině) dosahovaly podobných výsledků, ze kterých vycházejí stejné poznatky. Nalezení odpovědi může být obsahem dalšího výzkumu. K tomu by bylo třeba získat trénovací korpusy pro modely vnoření slov a zdroje anotovaných dat v daném jazyce. Dále by se jednotlivé faktory musely prozkoumat obdobným způsobem, a zjistit tak jejich dopad na výslednou kvalitu NER systému.

Největším překvapením (v pozitivním slova smyslu) byly dosažené výsledky algoritmu FastText. Ten se zdá být ideální volbou pro vytváření vektorových reprezentací slov, využívaných v systémech NER pro český jazyk. Jeho modely však ještě stále nejsou hojně využívané v systémech NER, zřejmě kvůli jeho teprve nedávnému zveřejnění, což by mohlo být podnětem pro další výzkum.

[41] Je zřejmé, že by nedocházelo k nárůstu kvality o několik procent, jako ve vykonaných experimentech. Dosavadně nejlepší systémy pro český jazyk využívají kromě modelu vektorových reprezentací také další vstupní informace, jako např. slovní druhy slov či různé příznaky slov. Je tedy možné, že v těchto vstupech bude obsažena již spousta zakódovaných informací ve vektorech. Přesto by však k nárůstu mělo dojít.

6 Reference

AL-RFOU, Rami a kol. POLYGLOT-NER: Massive Multilingual Named Entity Recognition. In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2015, s. 586–594. ISBN 978-1-61197-401-0.

An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec. *Analytics Vidhya: Learn everything about analytics* [online]. 2017 [cit. 2019-04-18]. Dostupné z: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/.

ARBUZOVA, Yana. Automatic Question Answering. *Towards Data Science* [online]. 2018 [cit. 2019-04-18]. Dostupné z: https://towardsdatascience.com/automatic-question-answering-ac7593432842.

ATHAVALE, Vinayak a kol. Towards Deep Learning in Hindi NER: An approach to tackle the Labelled Data Sparsity. *Computing Research Repository (CoRR)* [online]. 2016 [cit. 2019-04-17]. Dostupné z: https://arxiv.org/pdf/1610.09756.pdf.

BANERJEE, Suvro. Word2Vec—a baby step in Deep Learning but a giant leap towards Natural Language Processing. *Medium* [online]. 2018 [cit. 2019-04-18]. Dostupné z: https://medium.com/explore-artificial-intelligence/word2vec-a-baby-step-in-deep-learning-but-a-giant-leap-towards-natural-language-processing-40fe4e8602ba.

BATISTA, David S. *Maximum Entropy Markov Models and Logistic Regression* [online]. 2017 [cit. 2019-04-17]. Dostupné z: http://www.davidsbatista.net/blog/2017/11/12/Maximum_Entropy_Markov_Model/.

BATISTA, David S. *Named-Entity evaluation metrics based on entity-level* [online]. 2018 [cit. 2019-04-17]. Dostupné z: http://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/.

BLITZER, John a Xiaojin Jerry ZHU. Semi-supervised Learning for Natural Language Processing. In: *Tutorial Abstracts of ACL-08* [online]. HLT, 2008. Dostupné také z: https://pdfs.semanticscholar.org/03a5/991eeb5ba61592eb6df4b017563c200df2ca.pdf.

BOJANOWSKI, Piotr a kol. Enriching Word Vectors with Subword Information. *Computing Research Repository (CoRR)* [online]. 2016, (abs/1607.04606) [cit. 2019-04-18]. Dostupné z: https://arxiv.org/pdf/1607.04606.pdf.

BROWNLEE, Jason. Develop Your First Neural Network in Python With Keras Step-By-Step. *Machine Learning Mastery* [online]. 2016a [cit. 2019-04-17]. Dostupné z: https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/.

BROWNLEE, Jason. Embrace Randomness in Machine Learning. *Machine Learning Mastery* [online]. 2016b [cit. 2019-04-17]. Dostupné z: https://machinelearningmastery.com/randomness-in-machine-learning/.

BROWNLEE, Jason. What Are Word Embeddings for Text. *Machine Learning Mastery* [online]. 2017a [cit. 2019-04-18]. Dostupné z: https://machinelearningmastery.com/what-are-word-embeddings/.

BROWNLEE, Jason. Why One-Hot Encode Data in Machine Learning. *Machine Learning Mastery* [online]. 2017b [cit. 2019-04-18]. Dostupné z: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

BUSHAEV, Vitaly. How do we 'train' neural networks. *Towards Data Science* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73.

CVRČEK, Václav a Olga RICHTEROVÁ. *Korpus SYN* [online]. Příručka ČNK, 2018 [cit. 2019-04-20]. Dostupné z: https://wiki.korpus.cz/doku.php/cnk:syn.

CVRČEK, Václav a Olga RICHTEROVÁ. *Synchronie, synchronní korpus* [online]. Příručka ČNK, 2015 [cit. 2019-04-20]. Dostupné z: https://wiki.korpus.cz/doku.php?id=pojmy:synchronni.

Czes corpus. *Sketch Engine* [online]. 2015 [cit. 2019-04-20]. Dostupné z: https://www.sketchengine.eu/czes-corpus/.

DAVYDOVA, Olga. 7 types of Artificial Neural Networks for Natural Language Processing. *Medium* [online]. 2017a [cit. 2019-04-17]. Dostupné z: https://medium.com/@datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2.

DAVYDOVA, Olga. 10 Applications of Artificial Neural Networks in Natural Language Processing. *Medium* [online]. 2017b [cit. 2019-04-17]. Dostupné z: https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a.

DEMIR, Hakan a Arzucan ÖZGÜR. Improving Named Entity Recognition for Morphologically Rich Languages Using Word Embeddings. In: *2014 13th*

International Conference on Machine Learning and Applications [online]. IEEE, 2014, s. 117–122 [cit. 2019-04-22]. DOI: 10.1109/ICMLA.2014.24. ISBN 978-1-4799-7415-3. Dostupné z: http://ieeexplore.ieee.org/document/7033101/.

DORMEHL, Luke. What is an artificial neural network? Here's everything you need to know. *Digital Trends* [online]. 2019 [cit. 2019-04-17]. Dostupné z: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/.

ELVIS. Deep Learning for NLP: An Overview of Recent Trends. *Medium* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d.

FARUQUI, Manaal a kol. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* [online]. Berlin, Germany: Association for Computational Linguistics, 2016, s. 30–35 [cit. 2019-04-18]. DOI: 10.18653/v1/W16-2506. Dostupné z: https://www.aclweb.org/anthology/W16-2506.

FIRTH, J.R. (1957). A synopsis of linguistic theory 1930-1955. In: *Studies in Linguistic Analysis*, s. 1–32. Oxford: Philological Society. Reprinted in F.R. Palmer (ed.), *Selected Papers of J.R. Firth 1952-1959*, London: Longman (1968).

FORTNEY, Kendall. Pre-Processing in Natural Language Machine Learning Go to the profile of Kendall Fortney Kendall Fortney. *Towards Data Science* [online]. 2017 [cit. 2019-04-18]. Dostupné z: https://towardsdatascience.com/pre-processing-in-natural-language-machine-learning-898a84b8bd47.

FUMO, David. Types of Machine Learning Algorithms You Should Know. *Towards Data Science* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861.

GARBADE, Michael J. A Quick Introduction to Text Summarization in Machine Learning. *Towards Data Science* [online]. 2018 [cit. 2019-04-18]. Dostupné z: https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f.

GATE's ANNIE System. *GATE: general architecture for text engineering* [online]. The University of Sheffield, c1995–2019 [cit. 2019-04-17]. Dostupné z: https://gate.ac.uk/ie/annie.html.

GHADDAR, Abbas a Philippe LANGLAIS. WiNER: A Wikipedia Annotated Corpus for Named Entity Recognition. *Proceedings of the Eighth International Joint Conference on Natural Language Processing* [online]. Taipei, Taiwan, 2017, (Volume 1: Long Papers), s. 413–422 [cit. 2019-04-19]. Dostupné z: https://www.aclweb.org/anthology/I17-1042.

GHAFFARI, Parsa. NLP and Text Analytics Simplified: Document Clustering. *LinkedIn* [online]. 2015 [cit. 2019-04-18]. Dostupné z: https://www.linkedin.com/pulse/nlp-text-analytics-simplified-document-clustering-parsa-ghaffari/.

GOLDBERG, Y. A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research*. 2016. sv. 57, s. 345–420. ISSN 1076-9757.

GOYAL, Archana, Manish KUMAR a Vishal GUPTA. *Named Entity Recognition: Applications, Approaches and Challenges* [online]. International Journal of Advance Research in Science and Engineering (IJARSE), 2017, s. 1902–1916 [cit. 2019-04-17]. ISSN 2319-8354. Dostupné z: https://pdfs.semanticscholar.org/2060/5fdae23f6e8d945deb22e09c46cecebb4f35.pdf.

GRAVE, Edouard. Releasing fastText. *fastText: Library for efficient text classification and representation learning* [online]. 2016 [cit. 2019-04-18]. Dostupné z: https://fasttext.cc/blog/2016/08/18/blog-post.html.

GREGORIČ, Andrej, Yoram BACHRACH a Sam COOPE. Named Entity Recognition With Parallel Recurrent Neural Networks. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Melbourne, Australia: Association for Computational Linguistics, 2018, s. 69–74.

GRIMES, Seth. A Brief History of Text Analytics. *BeyeNETWORK* [online]. 2007 [cit. 2019-04-17]. Dostupné z: http://www.b-eye-network.com/view/6311.

GUPTA, Mohan. A Review of Named Entity Recognition (NER) Using Automatic Summarization of Resumes. *Towards Data Science* [online]. 2018a [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/a-review-of-named-entity-recognition-ner-using-automatic-summarization-of-resumes-5248a75de175.

GUPTA, Shashank. Named Entity Recognition: Applications and Use Cases. *Towards Data Science* [online]. 2018b [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/named-entity-recognition-applications-and-use-cases-acdbf57d595e.

GUPTA, Shashank. Word Embeddings in NLP and its Applications. *Hackernoon* [online]. 2019 [cit. 2019-04-18]. Dostupné z: https://hackernoon.com/word-embeddings-in-nlp-and-its-applications-fab15eaf7430.

HARRIS, Zellig S. Distributional Structure. *Word*. Routledge, 1954, 10(2-3), s. 146–162. DOI: 10.1080/00437956.1954.11659520.

HLADKÁ, Zdeňka. Polysémie. *CzechEncy: Nový encyklopedický slovník češtiny* [online]. c2012–2018 [cit. 2019-04-18]. Dostupné z: https://www.czechency.org/slovnik/POLYS%C3%89MIE.

HONNIBAL, Matthew. Pseudo-rehearsal: A simple solution to catastrophic forgetting for NLP. In: *Explosion AI* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://explosion.ai/blog/pseudo-rehearsal-catastrophic-forgetting.

CHIU, Jason P.C. a Eric NICHOLS. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* [online]. 2016, s. 357–370 [cit. 2019-04-22]. DOI: 10.1162/tacl_a_00104. Dostupné z: https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00104.

INGERSOLL, Grant S., Thomas S. MORTON a Andrew L. FARRIS. *Taming text: how to find, organize, and manipulate it.* Shelter Island: Manning, [2013]. ISBN 9781933988382.

ISMAIL, Nick. The data era is shifting: From creation to storage to readiness. *Information Age* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://www.information-age.com/data-era-shifting-123476838/.

JIACONDA. A Concise History of Neural Networks. *Towards Data Science* [online]. 2016 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec.

KEITAKURITA. Paper Dissected: „Glove: Global Vectors for Word Representation“ Explained. *Machine Learning Explained: Deep learning, python, data wrangling and other machine learning related topics explained for practitioners* [online]. 2018 [cit. 2019-04-18]. Dostupné z: http://mlexplained.com/2018/04/29/paper-dissected-glove-global-vectors-for-word-representation-explained/.

KESSEL, Patrick. An intro to topic models for text analysis. *Medium* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://medium.com/pew-research-center-decoded/an-intro-to-topic-models-for-text-analysis-de5aa3e72bdb.

Koncept umělé neuronové sítě. *Matematická biologie: e-learningová učebnice* [online]. c2019a [cit. 2019-04-17]. Dostupné z: http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--uvod-do-neuronovych-siti--koncept-umele-neuronove-site.

KONKOL, Michal a Miloslav KONOPÍK. CRF-Based Czech Named Entity Recognizer and Consolidation of Czech NER Research. *Text, Speech, and Dialogue* [online]. Berlin, Heidelberg: Springer, 2013, s. 153–160 [cit. 2019-04-21]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-40585-3_20. ISBN 978-3-642-40584-6. Dostupné z: http://link.springer.com/10.1007/978-3-642-40585-3_20.

KONKOL, Michal a Miloslav KONOPÍK. Maximum Entropy Named Entity Recognition for Czech Language. *Text, Speech and Dialogue* [online]. Berlin, Heidelberg: Springer, 2011, s. 203–210 [cit. 2019-04-21]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-23538-2_26. ISBN 978-3-642-23537-5. Dostupné z: http://link.springer.com/10.1007/978-3-642-23538-2_26.

KONOPÍK, Miloslav a Ondřej PRAŽÁK. LDA in Character-LSTM-CRF Named Entity Recognition. *Text, Speech, and Dialogue* [online]. Cham: Springer International Publishing, 2018, s. 58–66 [cit. 2019-04-22]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-00794-2_6. ISBN 978-3-030-00793-5. Dostupné z: http://link.springer.com/10.1007/978-3-030-00794-2_6.

KRALLINGER, Martin. *Information extraction and other NLP tasks* [online]. 2016 [cit. 2019-04-17]. Dostupné z: https://www.fosteropenscience.eu/sites/default/files/original/2958.pdf.

KRAVALOVÁ, Jana a Zdeněk ŽABOKRTSKÝ. Czech Named Entity Corpus and SVM-based Recognizer. *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)* [online]. Suntec, Singapore: Association for Computational Linguistics, 2009, s. 194–201 [cit. 2019-04-21]. Dostupné z: https://www.aclweb.org/anthology/W09-3538.

KRÁL, Pavel. Features for named entity recognition in Czech language. *International Conference on Knowledge Discovery and Information Retrieval (KDIR)* [online]. Paris, France, 2011 [cit. 2019-04-17]. Dostupné z: https://home.zcu.cz/ pkral/papers/kral_keod11.pdf.

KULKARNI, Vivek, Yashar MEHDAD a Troy CHEVALIER. Domain Adaptation for Named Entity Recognition in Online Media with Word

Embeddings. *Computing Research Repository (CoRR)* [online]. 2016, (abs/1612.00148) [cit. 2019-04-18]. Dostupné z: http://arxiv.org/abs/1612.00148.

LAMPLE, Guillaume a kol. Neural Architectures for Named Entity Recognition. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [online]. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, s. 260–270 [cit. 2019-04-21]. DOI: 10.18653/v1/N16-1030. Dostupné z: http://aclweb.org/anthology/N16-1030.

LEVY, Omer, Yoav GOLDBERG a Ido DAGAN. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics* [online]. 2015, 3, s. 211–225 [cit. 2019-04-18]. DOI: 10.1162/tacl_a_00134. ISSN 2307-387X. Dostupné z: https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00134.

LI, Ray. History of data mining. *Hackerbits* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://hackerbits.com/data/history-of-data-mining/.

Lingvistika. *Wikipedie: otevřená encyklopedie* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://cs.wikipedia.org/wiki/Lingvistika.

Machine Translation. *GALA: Globalization and Localization Association* [online]. c2019 [cit. 2019-04-18]. Dostupné z: https://www.gala-global.org/what-machine-translation.

MANNING, Christopher D., Prabhakar RAGHAVAN a Hinrich SCHÜTZE. *Introduction to information retrieval*. New York: Cambridge University Press, 2008. ISBN 0521865719.

MATAS, Martin. *Rozpoznávání pojmenovaných entit pomocí neuronových sítí* [online]. Plzeň, 2018 [cit. 2019-04-22]. Dostupné z: https://dspace5.zcu.cz/handle/11025/32278. Bakalářská práce. Západočeská univerzita v Plzni. Vedoucí práce Ing. Michal Konkol, Ph.D.

Matematický model a aktivní dynamika neuronu. *Matematická biologie: e-learningová učebnice* [online]. c2019b [cit. 2019-04-17]. Dostupné z: http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--jednotlivy-neuron--matematicky-model-a-aktivni-dynamika-neuronu.

MCCORMICK, Chris. *Word2Vec Tutorial - The Skip-Gram Model* [online]. 2016 [cit. 2019-04-18]. Dostupné z: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/.

MCCORMICK, Chris. *Word2Vec Tutorial Part 2 - Negative Sampling* [online]. 2017 [cit. 2019-04-18]. Dostupné z: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/.

Meaning of inflected language in English. *Cambridge Dictionary* [online]. 2019 [cit. 2019-04-17]. Dostupné z: https://dictionary.cambridge.org/dictionary/english/inflected-language.

MIKOLOV, Tomas a kol. Distributed Representations of Words and Phrases and their Compositionality. *Computing Research Repository (CoRR)* [online]. 2013a, (abs/1310.4546) [cit. 2019-04-18]. Dostupné z: https://arxiv.org/pdf/1310.4546.pdf.

MIKOLOV, Tomas a kol. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR* [online]. 2013b [cit. 2019-04-17]. Dostupné z: https://arxiv.org/pdf/1301.3781.pdf.

MIKOLOV, Tomas, Wen-tau YIH a Geoffrey ZWEIG. Linguistic Regularities in Continuous Space Word Representations. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* [online]. Atlanta, Georgia: Association for Computational Linguistics, 2013, s. 746–751 [cit. 2019-04-18]. Dostupné z: https://www.aclweb.org/anthology/N13-1090.

Module 11: Machine Learning. *GATE: general architecture for text engineering* [online]. The University of Sheffield, c1995–2011 [cit. 2019-04-17]. Dostupné z: https://gate.ac.uk/sale/talks/gate-course-may11/track-3/module-11-machine-learning/module-11.pdf.

MRÁZOVÁ, Iveta. *Dobývání znalostí* (přednáška) [online]. Katedra teoretické informatiky: Matematicko-fyzikální fakulta Univerzity Karlovy v Praze [cit. 2019-04-17]. Dostupné z: https://ksvi.mff.cuni.cz/ mraz/datamining/lecture/ Dobyvani_Znalosti_Prednaska_Asociacni_pravidla.pdf.

Named-entity recognition: What are Named Entities. *Deep AI* [online]. c2017–2018 [cit. 2019-04-17]. Dostupné z: https://deepai.org/machine-learning-glossary-and-terms/named-entity-recognition.

Natural language processing and text mining. *Expert System* [online]. 2016 [cit. 2019-04-17]. Dostupné z: https://www.expertsystem.com/natural-language-processing-and-text-mining/.

NGUYEN, Long Hoang. *Rozpoznávání pojmenovaných entit s pomocí rekurentních neuronových sítí* [online]. Praha, 2017 [cit. 2019-04-22]. Dostupné z: https://dspace.cvut.cz/handle/10467/69368. Bakalářská práce. ČVUT. Vedoucí práce Jan Pichl.

NOUVEL, Damien, Maud EHRMANN a Sophie ROSSET. *Named entities for computational linguistics*. Hoboken, NJ, USA: Wiley, [2016]. Focus series (London, England). ISBN 9781848218383.

OLEJNIK, Galina. Hierarchical softmax and negative sampling: short notes worth telling. *Towards Data Science* [online]. 2017 [cit. 2019-04-18]. Dostupné z: https://towardsdatascience.com/hierarchical-softmax-and-negative-sampling-short-notes-worth-telling-2672010dbe08.

OSIPENKO, Alexander. Genetic algorithms and hyperparameters—Weekend of a Data Scientist. *Medium* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://medium.com/cindicator/genetic-algorithms-and-hyperparameters-weekend-of-a-data-scientist-8f069669015e.

Overfitting in Machine Learning: What It Is and How to Prevent It. *Elite Data Science* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://elitedatascience.com/overfitting-in-machine-learning.

PAHWA, Ramit. Micro-Macro Precision, Recall and F-Score. *Medium* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://medium.com/@ramit.singh.pahwa/micro-macro-precision-recall-and-f-score-44439de1a044.

PENNINGTON, Jeffrey, Richard SOCHE a Christopher D. MANNING. *GloVe: Global Vectors for Word Representation* [online]. 2014 [cit. 2019-04-18]. Dostupné z: https://nlp.stanford.edu/pubs/glove.pdf.

PERONE, Christian S. Machine Learning: Cosine Similarity for Vector Space Models (Part III). *Terra Incognita* [online]. 2013 [cit. 2019-04-18]. Dostupné z: http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/.

PRABHU. Understanding Hyperparameters and its Optimisation techniques. *Towards Data Science* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568.

RAAJ, Vivan. Named Entity Recognition using Bi-directional Long Short-Term Memory (Bi-LSTM). *Medium* [online]. 2018 [cit. 2019-05-05]. Dostupné z: https://medium.com/@vivanraaj/named-entity-recognition-using-bi-directional-long-short-term-memory-bi-lstm-ab54bbf7cc76.

REDMORE, Seth. Machine Learning for Natural Language Processing. *Lexalytics* [online]. 2019 [cit. 2019-04-17]. Dostupné z: https://www.lexalytics.com/lexablog/machine-learning-vs-natural-language-processing-part-1.

ROUSE, Margaret. Text mining (text analytics). *Search Business Analytics: Definition* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://searchbusinessanalytics.techtarget.com/definition/text-mining.

RUDRA MURTHY, V. a Pushpak BHATTACHARYYA. A Deep Learning Solution to Named Entity Recognition. GELBUKH, Alexander, ed. *Computational Linguistics and Intelligent Text Processing* [online]. Cham: Springer International Publishing, 2018, s. 427–438 [cit. 2019-04-22]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-75477-2_30. ISBN 978-3-319-75476-5. Dostupné z: http://link.springer.com/10.1007/978-3-319-75477-2_30.

RUSELL, Ingrid. Definition of a Neural Network. *Neural Networks Module* [online]. University of Hartford, 1996 [cit. 2019-04-17]. Dostupné z: http://uhaweb.hartford.edu/compsci/neural-networks-definition.html.

SCIFORCE. Word Vectors in Natural Language Processing: Global Vectors (GloVe). *Medium* [online]. 2018 [cit. 2019-04-18]. Dostupné z: https://medium.com/sciforce/word-vectors-in-natural-language-processing-global-vectors-glove-51339db89639.

SEOK, Miran a kol. Named Entity Recognition using Word Embedding as a Feature. *International Journal of Software Engineering and Its Applications* [online]. 2016, s. 93–104 [cit. 2019-04-22]. DOI: 10.14257/ijseia.2016.10.2.08. ISSN 17389984. Dostupné z: http://www.sersc.org/journals/IJSEIA/vol10_no2_2016/8.pdf.

SHAH, Tarang. About Train, Validation and Test Sets in Machine Learning. *Towards Data Science* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7.

SHARMA, Pulkit. The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes). *Analytics Vidhya: Learn everything about analytics* [online]. 2018 [cit. 2019-04-18]. Dostupné z: https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/.

SCHMIDT, Ben. Vector Space Models for the Digital Humanities. *Ben's Bookworm Blog* [online]. 2015 [cit. 2019-04-18]. Dostupné z:

http://bookworm.benschmidt.org/posts/2015-10-25-Word-Embeddings.html.

SIENČNIK, Scharolta Katharina. Adapting word2vec to Named Entity Recognition. In: *Proceedings of the 20th Nordic Conference of Computational Linguistics* [online]. 2015 [cit. 2019-04-17]. Dostupné z: https://www.aclweb.org/anthology/W15-1830.

SKANSI, Sandro. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*. Berlin: Springer, 2018. ISBN 978-3-319-73003-5.

SPOUSTOVÁ, Johanka a Miroslav SPOUSTA. A High-Quality Web Corpus of Czech. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)* [online]. Istanbul, Turkey: European Language Resources Association, 2012, s. 311–315 [cit. 2019-04-20]. Dostupné z: http://www.lrec-conf.org/proceedings/lrec2012/pdf/120_Paper.pdf.

STRAKOVÁ, Jana, Milan STRAKA a Jan HAJIČ. Neural Networks for Featureless Named Entity Recognition in Czech. *Text, Speech, and Dialogue* [online]. Cham: Springer International Publishing, 2016, 2016-09-03, s. 173–181 [cit. 2019-04-22]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-45510-5_20. ISBN 978-3-319-45509-9. Dostupné z: http://link.springer.com/10.1007/978-3-319-45510-5_20.

Stručný terminologický slovník. *Centrum zpracování přirozeného jazyka* [online]. c2019a [cit. 2019-04-17]. Dostupné z: https://nlp.fi.muni.cz/cs/Terminologie.

STUBBS, Amber a James PUSTEJOVSKY. *Natural Language Annotation for Machine Learning* [online]. O'Reilly Media, 2012 [cit. 2019-04-17]. ISBN 9781449332693. Dostupné z: https://www.oreilly.com/library/view/natural-language-annotation/9781449332693/#toc-start.

SVOBODA, Lukáš a Tomáš BRYCHCÍN. *New Word Analogy Corpus for Exploring Embeddings of Czech Words* [online]. 2018, s. 103–114 [cit. 2019-04-18]. DOI: 10.1007/978-3-319-75477-2_6. ISSN 978-3-319-75476-5. Dostupné z: https://arxiv.org/pdf/1608.00789.pdf.

ŠEVČÍKOVÁ, Magda, Zdeněk ŽABOKRTSKÝ a Oldřich KRŮZA. Named Entities in Czech: Annotating Data and Developing NE Tagger. MATOUŠEK, Václav a Pavel MAUTNER, ed. *Text, Speech and Dialogue* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, s. 188–195 [cit. 2019-04-19]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-540-74628-7_26. ISBN 978-3-540-74627-0. Dostupné z: https://ufal.mff.cuni.cz/žabokrtsky/publications/papers/tsd07-namedent.pdf.

ŠŮSTEK, Martin. *Word2vec modely s přidanou kontextovou informací* [online]. Brno, 2017 [cit. 2019-04-18]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=159497. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. František Zbořil, CSc.

TIEDEMANN, Jörg. Parallel Data, Tools and Interfaces in OPUS. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* [online]. Istanbul, Turkey: European Language Resources Association (ELRA), 2012 [cit. 2019-04-20]. Dostupné z: http://opus.nlpl.eu/Europarl.php.

TJONG KIM SANG, Erik F. a Fien DE MEULDER. Introduction to the CoNLL-2003 shared task. In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* [online]. Morristown, NJ, USA: Association for Computational Linguistics, 2003, s. 142–147 [cit. 2019-04-19]. DOI: 10.3115/1119176.1119195. Dostupné z: http://portal.acm.org/citation.cfm?doid=1119176.1119195.

TJONG KIM SANG, Erik F. Introduction to the CoNLL-2002 shared task. In: *Proceeding of the 6th conference on Natural language learning - COLING-02* [online]. Morristown, NJ, USA: Association for Computational Linguistics, 2002, s. 1–4 [cit. 2019-04-19]. DOI: 10.3115/1118853.1118877. Dostupné z: http://portal.acm.org/citation.cfm?doid=1118853.1118877.

Unstructured Data and the 80 Percent Rule. *Breakthrough Analysis: Seth Grimes on NLP, text analytics, sentiment analysis, BI, visualization and more* [online]. 2008 [cit. 2019-04-17]. Dostupné z: http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/.

VIJAYARANI, S., V. ILAMATHI a NITHYA. Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks* [online]. 2015, Vol 5(1), s. 7–16 [cit. 2019-04-18]. ISSN 2249-5789. Dostupné z: https://www.ijcscn.com/Documents/Volumes/vol5issue1/ijcscn2015050102.pdf.

Vícevrstvý perceptron a syndrom přeučení. *Matematická biologie: e-learningová učebnice* [online]. c2019 [cit. 2019-04-17]. Dostupné z: http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-perceptrony--vicevrstvy-perceptron--vicevrstvy-perceptron-a-syndrom-preuceni.

VOSS, Peter. *Becominghuman.ai: The Third Wave of AI* [online]. 2017 [cit. 2019-04-17]. Dostupné z: https://becominghuman.ai/the-third-wave-of-ai-1579ea97210b.

WEISCHEDEL, Ralph a kol. *OntoNotes Release 5.0*. Philadelphia: Linguistic Data Consortium, 2013. ISBN 1-58563-659-2.

What is Machine Learning: A definition. *Expert System* [online]. c2019 [cit. 2019-04-17]. Dostupné z: https://www.expertsystem.com/machine-learning-definition/.

YADAV, Saurabh. Weight Initialization Techniques in Neural Networks. *Towards Data Science* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/weight-initialization-techniques-in-neural-networks-26c649eb3b78.

YOON, Kim. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistic, 2014, s. 1746–1751.

ZEMAN, Daniel a kol. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Stroudsburg, USA: Association for Computational Linguistics, 2017, s. 1–19.

Zpracování přirozeného jazyka aneb NLP. *Centrum zpracování přirozeného jazyka* [online]. c2019b [cit. 2019-04-17]. Dostupné z: https://nlp.fi.muni.cz/cs/ZpracovaniPrirozenehoJazyka.

ZULKIFLI, Hafidz. Understanding Learning Rates and How It Improves Performance in Deep Learning. *Towards Data Science* [online]. 2018 [cit. 2019-04-17]. Dostupné z: https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10.

Přílohy

A Elektronická příloha

Obsahem elektronické přílohy je:

- složka *data*:
  - složka *data_NER* – anotovaný datový zdroj Extended CNEC 2.0,
  - složka *korpus_analogii* – korpus na evaluaci modelů vnoření slov v češtině,
  - složka *korpusy_vnoreni_slov* – složky všech použitých korpusů s pokyny pro stažení a předzpracování,
  - složka *lemmatizer* – jazykový model *MorphoDita* pro lemmatizaci,
- složka *zaznamy_experimentu*:
  - složka se záznamy všech provedených experimentů s podrobnými výsledky,
  - mapování záznamů na jednotlivá pozorování experimentů (*mapovani.xlsx*),
- složka *zdrojovy_kod*:
  - *Python* skripty,
  - složka *instalace* – instalační pokyny a soubory se seznamy potřebných balíčků,
- podrobné výsledky experimentu pro hledání nejlepších hyper-parametrů neuronové sítě, včetně grafů (*hyper_parametry_vysledky_experimentu.xlsx*).

B Detailní vyhodnocení nejlepšího výsledku Word2Vec

Tabulka 15: Kvalita rozpoznávání jednotlivých typů entit (v korpusu Extended CNEC 2.0) NER systému s nejlepším dosaženým výsledkem v experimentech algoritmu Word2Vec, vyjádřená přesností („precision“), pokrytím („recall“) a F1-mírou, včetně počtu systémem predikovaných entit a celkového počtu entit v korpusu

Typ entity	**Přesnost** [%]	**Pokrytí** [%]	**F1-míra** [%]	**Predik. entit**	**Celkem entit**
Čísla a adresy	70,91	70,91	70,91	55	55
Geografická označení	71,32	76,98	74,05	408	378
Názvy institucí	61,47	66,98	64,11	353	324
Názvy médií	64,10	52,08	57,47	39	48
Názvy artefaktů	45,34	45,81	45,57	386	382
Jména osob	77,15	80,21	78,65	499	480
Časová vyjádření	90,44	89,95	90,19	366	368
Celkem	**69,47**	**71,89**	**70,66**	**2138**	**2035**