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

Diplomová práce

Vícejazyčné značkování sémantických rolí

Plzeň 2017 Ondřej Pražák

Prohlášení

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

V Plzni dne 17. května 2017

Ondřej Pražák

Abstrakt

Tato práce se zabývá automatickým značkováním sémantických rolí (SRL, z anglického Semantic Role Labeling) ve větách. V teoretické části práce jsou srovnané různé aktuálně používané metody. Zvláštní pozornost je věnována metodám nezávislým na jazyce. Tedy metodám projekce anotací, přenositelným modelům a učení bez učitele.
V práci jsou navržené experimenty ověřující použitelnost univerzální syntaxe - Universal Dependencies (UD) pro vícejazyčné značkování sémantických rolí. Na závěr je navržena, implementována a ověřena metoda vícejazyčného SRL využívající Universal Dependencies. Výsledky vypadají velice slibně a dokazují, že UD jsou pro metody vícejazyčného SRL velice dobré. Práce přináší spoustu možností dalšího výzkumu.

Abstract

This thesis is focused on Semantic Role Labeling (SRL). The theoretical part compares the most interesting known methods for SRL. Special attention is paid to language independent methods. Specifically annotation projection, model transfer and unsupervised methods.
We design experiments to verify whether the Universal Dependencies are suitable for cross-lingual SRL. Finally, the whole SRL system based upon Universal Dependencies is designed and implemented. The results are very promising. They prove that Universal Dependencies are suitable for cross-lingual SRL. The work opens new interesting research paths for the future.

Poděkování

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

Obsah

1 Úvod

Značkování sémantických rolí (Semantic role labeling, zkráceně SRL) se dnes úspěšně používá v mnoha úlohách zpracování přirozeného jazyka, například v systémech pro odpovídání otázek [SL07] nebo ve strojovém překladu [LG10]. Klasické metody učení s učitelem už dnes dosahují dobrých výsledků a nedají se příliš vylepšit. [Haj+09] Trénovací data pro SRL ale existují pouze pro několik málo jazyků[1] a ani pro ně není dat příliš mnoho. To dalo podnět snaze navrhnout metody, které trénovací data nepotřebují a metody přenositelné mezi jazyky, kde můžeme využít data v jednom jazyce pro natrénovaní systému i pro jiné jazyky.

V roce 2015 byla dokončena specifikace vícejazyčné syntaktické anotace *Universal Dependencies*. Na mnoha úlohách vypadají výsledky s jejím použitím velice slibně.

Tato práce se zabývá primárně vícejazyčným značkováním sémantických rolí. V práci je navržen systém pro vícejazyčné značkování sémantických rolí založený na *Universal Dependencies*. Hlavním cílem prováděných experimentů je ověřit vhodnost Universal Dependencies pro vícejazyčné značkování sémantických rolí.

1.1 Multilingual a Cross-lingual

Když hovoříme o vícejazyčných přístupech (modelech), v anglické literatuře se vyskytují dva termíny:

- **Multilingual** - Vícejazyčný model. Můžeme ho natrénovat na různých jazycích. Proces učení může být stejný pro všechny jazyky, ale k vytvoření modelu potřebujeme data v tom jazyce, pro který chceme vytvořit model.

[1] V *CoNLL 2009* jsou data pro 7 jazyků a byly vytvořené některé další korpusy pro experimenty s projekcí anotací (viz dále).

- **Cross-lingual** - Dalo by se přeložit jako model fungující napříč různými jazyky. Jedná se o model, který natrénujeme na jednom jazyce (nebo na libovolné podmnožině jazyků) a vytvořený model funguje i na ostatních jazycích (ne nutně všech existujících).

Tato práce se zabývá výhradně cross-linguálními modely, proto když nebude uvedeno jinak, pojmem vícejazyčný model myslíme právě cross-lingualní.

1.2 Reprezentace příznaků

1.2.1 Bag-of-words

Využívá se pro vektorovou reprezentaci textu. Z textů se vytvoří slovník a text je potom reprezentovaný vektorem o velikosti slovníku, kde každá složka obsahuje počet odpovídajících slov v textu. Reprezentace se nazývá bag-of-words, protože neuchovává informaci o uspořádání slov.

1.2.2 One-hot

One-hot reprezentuje příznak vektorem o velikosti počtu všech možných hodnot, kde je vždy právě jeden prvek vektoru nenulový. V případě reprezentace slov je one-hot v podstatě bag-of-words reprezentací jednoho slova.

2 Značkování sémantických rolí

Úkolem značkování sémantických rolí je v každé větě:

1. Najít predikáty. V původní definici úlohy podle [Gil02] byly predikáty pouze slovesa (činnosti). Později rozšířeno i na podstatná a přídavná jména.

2. Jednoznačně určit význam predikátů (word sense disambiguation) je potřeba pro namapování obecných argumentů na konkrétní, popřípadě k výběru použitého modelu, protože jedno slovo může mít více i naprosto odlišných významů a predikát pak může mít v závislosti na významu různé role argumentů.

3. Identifikovat argumenty každého predikátu (aktéry).

4. Argumentům přiřadit role. Sémantické role se v různých korpusech mírně liší, ale ty základní jsou podobné. Například v *Proposition bank* základní sémantické role jsou:

   A0 aktivní entita

   A1 pasivní entita (akcí ovlivněná)

   A2-A4 další entity

   AM-X modifikátory. Například modifikátor času AM-TMP nebo modifikátor místa AM-LOC

   R-X Omezující nebo vymezující argumenty (z anglického restriction)

   kompletní seznam modifikátorů uvádí [Gil02]

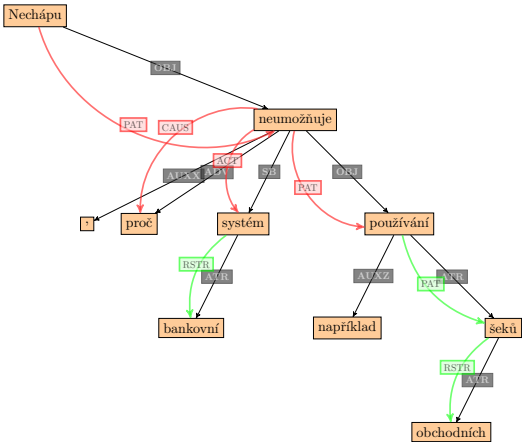
Obrázek 2.1 ukazuje příklad SRL anotace v angličtině.

(1) [He]$_{\text{AGENT|A0}}$ believes [in what he plays] $_{\text{THEME|A1}}$.

(2) Can [you] $_{\text{AGENT|A0}}$ cook [the dinner] $_{\text{PATIENT|A1}}$?

(3) [The nation‘s] $_{\text{AGENT|AM-LOC}}$ largest [pension]$_{\text{THEME|A1}}$ fund,

Obrázek 2.1: Tři příklady SRL anotace



Obrázek 2.2: Grafické znázornění SRL anotace (věta 30 z české testovací datové sady)

SRL anotace je na úrovni celých podstromů závislostního stromu[1]. To znamená, že když je slovo označeno jako argument, všichni jeho potomci v závislostním stromu jsou součástí tohoto argumentu. Příklad je na obrázku 2.2.

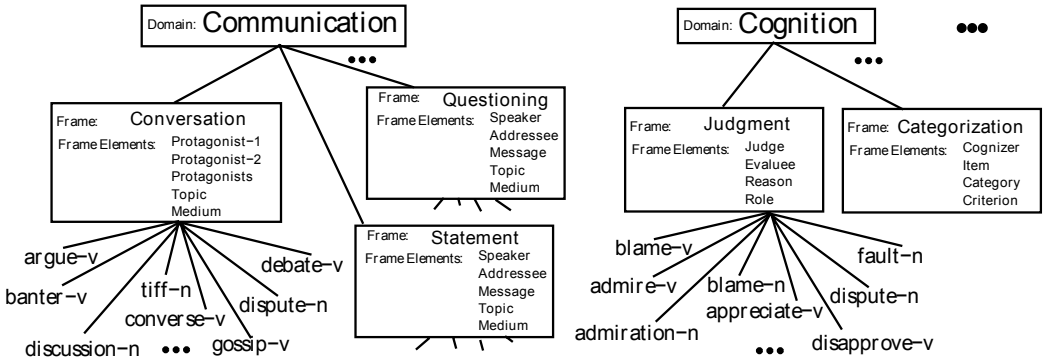
2.1 Lexikální databáze

První lexikální databází používanou pro SRL byl *FrameNet*. Dnes se pro SRL v angličtině používá spíše *Proposition bank* [PGK05]. Pokud se zabýváme SRL pro češtinu, je prakticky jediný dostupný korpus *PDT 3.0*. Ten obsahuje mimo jiné tektogramatickou vrstvu anotace, která je zobecněním SRL anotace.

2.1.1 FrameNet

FrameNet [BFL98] je lexikální databáze obsahující definice slov – predikátů a jejich argumentů. Predikáty jsou seskupované do rámců (frames), kde každý rámec obsahuje predikáty se stejnou množinou možných argumentů. Rámce jsou dále seskupované podle domény. Struktura FrameNetu je vidět na obrázku 2.3.

[1]Závislostní stromy jsou vysvětlené v kapitole 3.2.2.



Obrázek 2.3: Ukázka FrameNet rámců, převzato z [Gil02]

2.1.2 Proposition bank

Autoři [PGK05] vytvořili tuto lexikální databázi přidáním anotace predikátů a argumentů do nejznámějšího anglického korpusu se syntaktickou anotací *Penn Treebank*. Na rozdíl od FrameNetu, kde je více predikátů seskupováno do rámců, kde všechny predikáty v rámci mají stejné sémantické role argumentů, v Proposition bank má každý predikát - význam slova - vlastní význam hlavních argumentů (A_0 - A_n), pouze význam modifikujících a omezujících argumentů[2] je sdílený všemi predikáty.

Struktura Proposition bank je následující (příklad pro predikát *kick*):

```
Frameset kick.01 "drive or impel with the foot"
Arg0: Kicker
Arg1: Thing kicked
Arg2: Instrument (defaults to foot)
Ex1: [ArgM-DIS But] [Arg0 two big New York banks] seem [Arg0 *trace*]
to have kicked [Arg1 those chances] [ArgM-DIR away],
[ArgM-TMP for the moment], [Arg2 with the embarrassing failure
of Citicorp and Chase Manhattan Corp. to deliver 7.2 billion in bank
financing for a leveraged buy-out of United Airlines parent UAL Corp].
(wsj_1619)
Ex2: [Arg0 John] tried [Arg0 *trace*] to kick [Arg1 the football],
but Mary pulled it away at the last moment.
```

[2]Tedy argumentů *AM-x* a *R-x*.

2.1.3 VerbNet

VerbNet [Sch05] je rozsáhlá lexikální databáze anglických sloves se syntaktickou a sémantickou anotací. Slovesa jsou seskupována do třídy a třídy sloves jsou hierarchicky uspořádané. Každá třída může mít více syntaktických rámců. Každý syntaktický rámec má unikátní syntaktický strom. Rámce představují v podstatě syntaktické třídy.

2.1.4 CoNLL

CoNLL (Conference on Computational Natural Language Learning) je každoročně pořádaná konference o řešení společného úkolu. Každý rok se vypíše úkol nebo úkoly, k jejichž řešení se může přihlásit kdokoliv. *CoNLL* se v dřívějších letech zaměřovalo hlavně na syntaxi, v letech 2004, 2005, 2008 a 2009 se zabývalo SRL. Data vytvořená pro evaluaci na těchto konferencích se používají k evaluaci dodnes. Anglická data jsou extrahovaná z PropBank, česká data z *CoNLL 2009* [Haj+09] jsou extrahována z *PDT 3.0*.

2.2 Evaluace

Nejčastější evaluační metrika používaná pro SRL je F1 míra - harmonický průměr přesnosti a úplnosti (*precision*, *recall*). Někdy se používá *accuracy*. V rámci *CoNLL 2008* a 2009 byly vytvořeny evaluační skripty, které počítají přesnost úplnost a F1 míru (a řadu dalších statistik jako F1 přes jednotlivé role, přes syntaktické relace atd.).

Tabulka 2.1 uvádí možná chyby pří klasifikace do tříd (pro jednu konkrétní třídu) či při značkování. *TP* (z anglického *true positive*) uvádí, kolik prvků mělo být označeno a skutečně označeno bylo. FP (*false positive*) uvádí, kolik prvků bylo označeno, ačkoliv správně označené být neměly. *FN* (*false negative*) pak prvky které měly být správně označené a nebyly. TN (*true negative*) je počet prvků, které správně nebyly označeny.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{2.1}$$

	označený	neoznačený
pozitivní	TP	FN
negativní	FP	TN

Tabulka 2.1: Matice záměn

$$P = \frac{TP}{TP + FP} \tag{2.2}$$

$$R = \frac{TP}{TP + FN} \tag{2.3}$$

Další často používanou metrikou je *accuracy*:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

Ve většině publikací o SRL byly výsledky vyhodnocovány pouze přes hlavní slova (*head words* - nejvyšší uzly podstromu). SRL anotace byla navržena tak, že když je uzel označen jako argument, považuje se automaticky za argument celý jeho podstrom v závislostním stromě. Proto tento způsob vyhodnocování nezahrnuje chybu vytvoření závislostního stromu. Naměřených výsledků tedy systém reálně dosáhne pouze tehdy, když bude mít k dispozici bezchybné závislostní stromy.

3 Předzpracování pro SRL

3.1 Předzpracování textu

Lemmatizace je převedení slova na základní tvar. Například u sloves na infinitiv a u podstatných jmen na nominativ singuláru.

part-of-speech tagging je zařazení slov do kategorií podle jejich syntaktických vlastností. Těmito kategoriemi mohou být slovní druhy nebo nebo může být členění do kategorií ještě jemnější. [RP06]

Určení významu slova (z anglického word sense disambiguation) je určení jednoznačného významu slova na základě kontextu. [IV98]

Pojmenované entity (named entities) jsou faktografické informace v textu. Například:

- vlastní jména (osoby, města státy firmy),
- číselné údaje (čísla, data),
- další důležité informace (např. čísla zákonů).

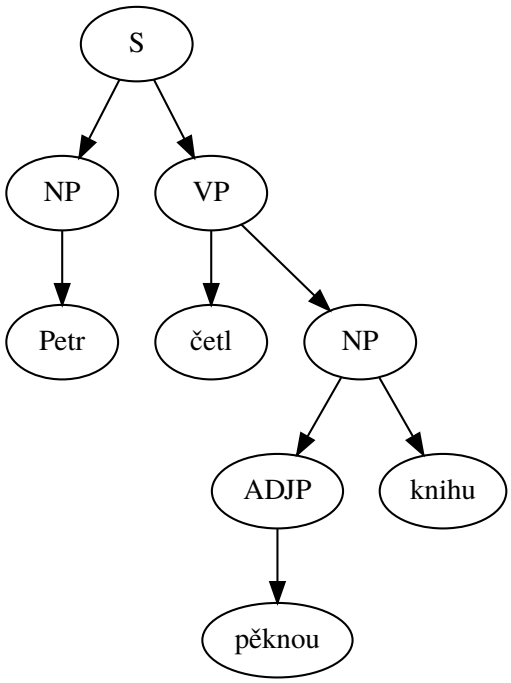
[SD02]

3.2 Syntaktické stromy

Syntaktické stromy se ve zpracování přirozeného jazyka používají k popisu syntaktické struktury věty. Nejvíce se používají dva druhy syntaktických stromů: Závislostní (*dependency*) a složkový (*constituency*).

3.2.1 Složkový strom

Složkový strom (*constituency tree*) zobrazuje větný rozbor a větné členy. Vnitřní uzly představují větné členy, listy stromu jsou jednotlivá slova.



Obrázek 3.1: Příklad složkového stromu

3.2.2 Závislostní strom

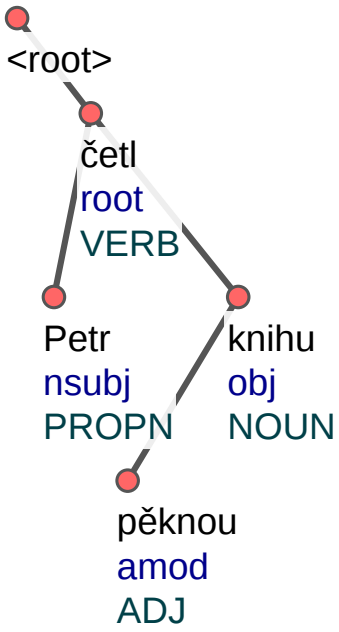
Závislostní strom (*dependency tree*) zobrazuje závislosti mezi slovy. Uzly stromu představují slova a hrany relace mezi nimi. Relace jsou pojmenované a mají směr závislosti.

3.3 Sémantické vektory

3.3.1 Sémantická podobnost slov

Základní sémantickou informací slova je jeho význam. Dvě slova jsou si sémanticky podobná, pokud mají podobný význam. Dále se zavádí ještě pojem sémantické souvislosti (*relatedness*), která udává, jak moc spolu slova souvisí. Například slova automobil a silnice spolu celkem úzce souvisí, ale rozhodně si nejsou svým významem podobná.

Metody distribuční sémantiky převádí slova na vektory reprezentující jejich



Obrázek 3.2: Příklad závislostního stromu

význam. Tyto metody jsou založené na distribuční hypotéze.

Distribuční hypotéza říká, že pokud se dvě slova vyskytují často ve stejném kontextu, mají pravděpodobně podobný význam.

Metody distribuční sémantiky lze rozdělit na dvě skupiny:

Metody využívající globální kontext uvažují jako kontext slova celý dokument. Výsledné vektory jsou si podobné, pokud se slova vyskytují často ve stejných dokumentech. Podobnost výsledných vektorů proto udává spíše souvislost než podobnost. Příkladem takových metod je LSA a LDA.

Metody využívající lokální kontext pracují s kontextem pevné, malé velikosti (například 10 slov na každou stranu). Pro malý kontext udávají skutečnou podobnost slov. Příkladem metod využívajících lokální kontext jsou HAL [LB96], *GloVe* [PSM14] a *Word2Vec* [Mik+13]

Detailní srovnání metod pro výpočet sémantické podobnosti uvádí [KP15].

State-of-the-art metodou pro určení sémantických vektorů je dnes stále *Skip-gram* z *Word2Vec*.

Shlukování je seskupování objektů do předem neznámých tříd na základě podobnosti. Metriky podobnosti mohou bít různé, pro vektory se nejčastěji používá kosinová podobnost.

Sémantické shluky jsou shluky slov, kde metrikou podobnosti je sémantická podobnost slov.

3.3.2 Word2Vec

V současné době state-of-the-art metoda na vytvoření sémantických reprezentací slov. Autoři [Mik+13] vytvořili dva modely: Continuous bag-of-words (CBOW) a Skip-gram. Základní myšlenka, objevená už výrazně dříve [Ben+03], je, že vektory vah skryté vrstvy neuronové sítě jako jazykového modelu dobře reprezentují význam slov. Tedy když se neuronová síť učí předpovídat slova na základě kontextu, naučí se jejich sémantickou reprezentaci (distribuční hypotéza viz výše). Základní dopředné sítě pro jazykové modelování mají čtyři vrstvy:

1. **vstupní** – Vstupují do ní slova ve one-hot reprezentaci[1].
2. **projekční** – Vstupní vektory kontextu se vynásobí společnou váhovou maticí a vzniknou interní reprezentace slov
3. **skrytá** – Vytváří reprezentaci celého kontextu
4. **výstupní** – Softmaxová vrstva odhaduje pravděpodobnosti aktuálního slova na základě kontextu

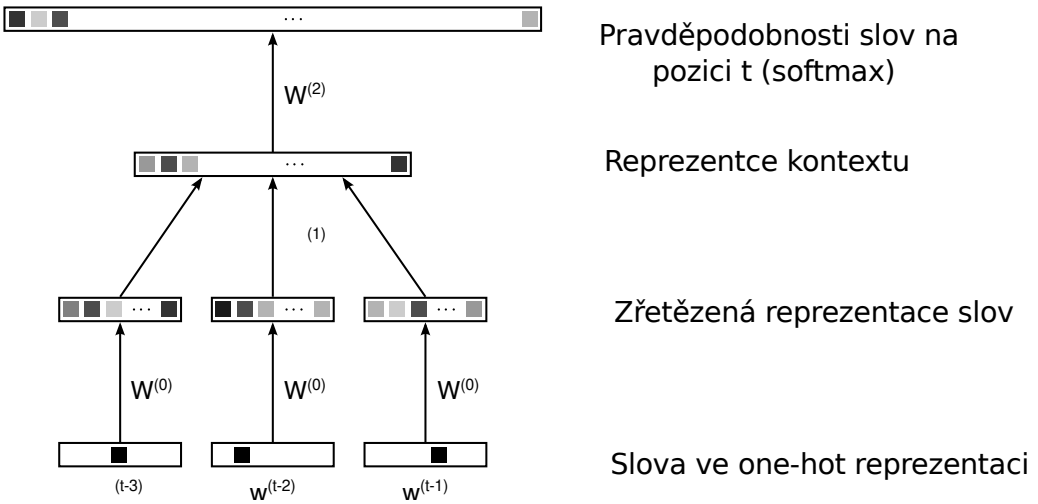
Softmax je zobecněním logistické funkce pro více tříd. Výstupem softmaxu je pravděpodobnost příslušnosti ke konkrétní třídě. Výstupem logistické regrese je pravděpodobnost pozitivní třídy při binární klasifikaci.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

[1]Slovo je reprezentované vektorem velikosti slovníku, kde pouze jedna složka odpovídající aktuálnímu slovu je nenulová.

$$P(x)_{softmax} = \frac{e^x}{\sum_{i=0}^{|C|} e^{x_i}} \tag{3.2}$$

kde C je množina všech tříd



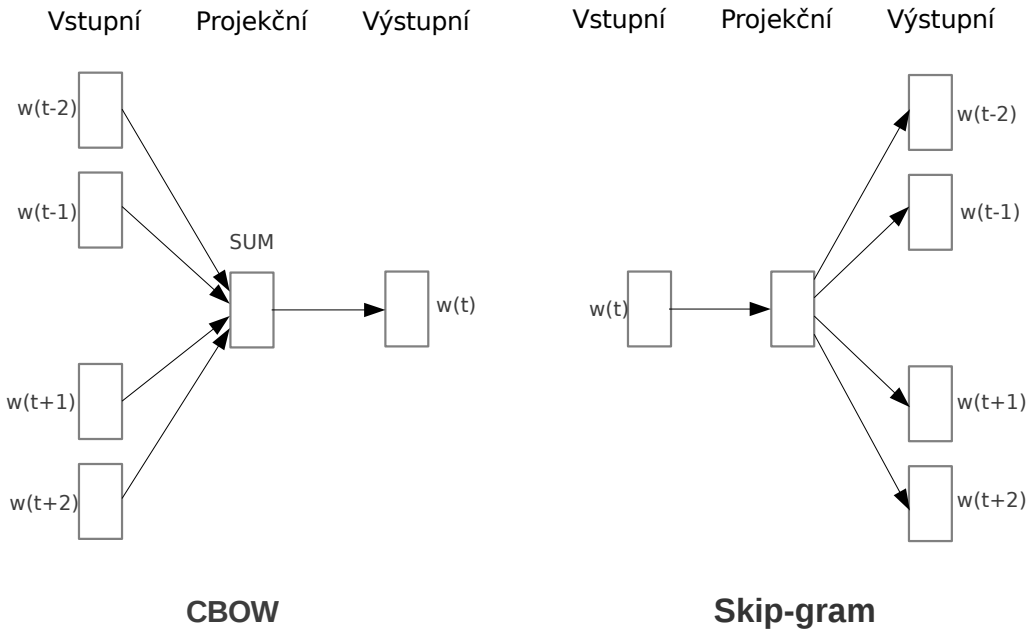
Obrázek 3.3: Architektura základní neuronové sítě pro jazykové modelování

[Mik+13] tento model výrazně výpočetně zjednodušil odstraněním skryté vrstvy a vektory kontextových slov pouze sčítá.

Podobnost vytvořených vektorů se pak nejčastěji počítá pomocí kosinové podobnosti.

Kosinová podobnost je kosinus úhlu mezi vektory. Tato míra podobnosti je nezávislá na velikosti obou vektorů.

$$\cos(\vec{u}, \vec{v}) = \frac{\vec{u}.\vec{v}}{|\vec{u}|.|\vec{v}|} \tag{3.3}$$



Obrázek 3.4: Architektura *Word2Vec* modelů (převzato z [Mik+13])

4 Klasické přístupy

Původní přístupy využívají učení s učitelem. Úlohu je možné rozdělit na několik klasifikačních podúloh a ty jsou vyřešené samostatnými klasifikátory. Úloha značkování sémantických rolí se dělí na tři podúlohy:

1. identifikace argumentů (argument identification),
2. přiřazení rolí (role labeling),
3. globální optimalizace (global optimization).

4.1 Používané příznaky

4.1.1 Syntaktické příznaky

Mezi základní syntaktické příznaky patří POS tag predikátu a argumentu, závislostní relace a další syntaktické příznaky jako například rod predikátu (činný/trpný).

Pravděpodobně nejdůležitější skupinou syntaktických příznaků jsou příznaky popisující cestu v závislostním stromu od predikátu k argumentu. Používá se orientovaná a neorientovaná cesta. Neorientovaná je pouze posloupností značek relací, orientovaná obsahuje ještě směr relace.

4.1.2 Lexikální příznaky

Důležitým příznakem v klasickém SRL je lemma nebo význam predikátu a argumentu. Zakódují se buď jako one-hot (viz kapitolu 1.2.2) reprezentace kořene podstromu, nebo bag-of-words (viz kapitolu 1.2.1) celého podstromu.

4.1.3 Sémantické příznaky

Používají se sémantické shluky slov nebo jejich celé vektorové reprezentace. Dalším častým příznakem jsou pojmenované entity.

4.2 Identifikace argumentů

Z pohledu strojového učení se jedná o binární klasifikaci, tedy rozhodnutí, jestli je nebo není dané slovo argumentem daného predikátu. Tato úloha je dobře řešitelná na základě analýzy syntaxe. Nejdůležitější jsou tedy příznaky využívající syntaktický strom. V SRL se používá spíše závislostní než složkový syntaktický strom. Používají se tyto příznaky:

- pozice slova relativně k predikátu (nalevo, napravo, vzdálenost),
- relace argumentu,
- cesta stromem od predikátu k argumentu (včetně relací i bez relací),
- part-of-speech tag.

4.3 Přiřazení rolí

Jedná se o klasifikaci do předem známých tříd. Používají se stejné příznaky jako v případě identifikace argumentů a některé důležité příznaky přibývají:

- slovesný rod (činný / trpný).

Nastává často opomíjený problém, že argumenty různých predikátů, respektive predikátů různých rámců mají vlastní argumenty, které jsou naprosto nezávislé na argumentech ostatních rámců. Jednoduše řečeno argument A2 v jednom rámci může mít úplně jinou roli než argument A2 v rámci jiném.

Když tedy použijeme jeden model klasifikátoru pro všechny predikáty, dopouštíme se chyby. Druhou možností je natrénovat model pro každý predikát zvlášť.

4.4 Globální optimalizace

SRL anotace je navržena tak, že predikát nesmí obsahovat více argumentů se stejnou rolí (alespoň ve většině jazyků, neplatí například česká data v *CoNLL 2009*). Proto nepřiřazujeme argumentu jednoduše nejpravděpodobnější roli, ale přiřadíme argumentům jednoho predikátu takové role, abychom maximalizovali sdruženou pravděpodobnost rolí všech argumentů daného predikátu. Přímočarý algoritmus takové optimalizace má exponenciální složitost (všechny kombinace rolí). Proto se používají omezující podmínky, se kterými sice optimalizace nemusí najít nejlepší řešení, ale pravděpodobně najde dostatečně dobré.

5 Zdroje pro vícejazyčné modely

Většina současných metod, které podporují více jazyků, využívá speciální zdroje dat. Například je téměř vždy potřeba paralelní korpus. Paralelní korpus je text, ve kterém je každá věta přeložena do více jazyků (typicky dvou). V následujících kapitolách rozebereme různé zdroje a metody nezávislé na jazyce.

5.1 Universal Dependencies

Universal Dependencies (UD) vznikly ze tří různých anotací:

- univerzální relace Stanfordu [MM08],
- part-of-speech tagy Googlu [PDM12],
- morfologické značky [Zem08] ÚFALu Karlovy univerzity.

UD anotace se nijak zásadně neliší od klasických závislostních anotací využívaných v SRL. Nejzásadnější rozdíl z pohledu SRL je to, že UD označuje za hlavní uzel podstromu obsahová slova. Ve standardních anotacích používaných v SRL jsou hlavními slovy často předložky, spojky nebo pomocná slovesa.

Ačkoliv Universal Dependencies vznikly teprve nedávno, existuje několik frameworků, které umožňují vytvoření UD stromů. Například UDpipe [Str16], Stanford CoreNLP [Man+14], Malt parser [Niv+07] a další.

5.2 Paralelní data

Základem většiny vícejazyčných modelů je paralelní korpus. Jedná se o shodný text přeložený do několika jazyků. Paralelní data jsou vytvářená primárně pro strojový překlad. Nejrozšířenější paralelní korpus je Europarl. Jedná se o zápisy z evropského parlamentu. Korpusy jsou dostupné pro dvacet dvojic jazyků, respektive jsou k dispozici pouze data pro angličtinu s dvaceti různými jazyky. Nevýhody tohoto korpusu jsou:

1. Pro většinu jazyků obsahuje málo dat.
2. Je relativně úzce doménově zaměřen.

Pro češtinu existuje mnohem větší a méně doménově zaměřený paralelní korpus CZENG. [Boj+16]

Další korpus pro češtinu, který stojí za zmínku, je *Prague Czech-English Dependency Treebank 2.0*. Jedná se o paralelní data pro češtinu a angličtinu ručně označená podle stejných pravidel jako PDT. Je to tedy paralelní korpus, který je mimo jiné označen sémantickými rolemi.

5.2.1 Europarl

Europarl [Koe05] je nejrozšířenější paralelní korpus pro evrovské jazyky používaný zejména pro strojový překlad. Korpus je vytvořen ze zápisů z evropského parlamentu. Korpus obsahuje 21 jazyků, respektive skládá se z dvaceti dvojic, vždy angličtina a druhý jazyk.

5.2.2 Vyhledávač bilinguálních dat OPUS

OPUS [Tie12] slouží k vyhledávání bilinguálních korpusů. Data sbírá z různých zdrojů, převádí je do jednotného formátu a vytváří zarovnání slov pro strojový překlad.

5.3 Vícejazyčné sémantické shluky

Často používaným příznakem v SRL jsou sémantické shluky slov. Pokud chceme shluky slov použít v jazykově nezávislém modelu, je potřeba vytvořit alespoň bilinguální shluky.

5.3.1 Bivec

Autoři [LPM15] upravili skip-gram model ([Mik+13]). Jejich varianta předpovídá kontextová slova v jednom jazyce na základě centrálního slova v jazyce druhém a obráceně. Takto se síť naučí vektorové reprezentace slov společné pro oba jazyky. K trénování potřebujeme paralelní korpus a jeho zarovnání, aby bylo možné nahradit centrální slova v jednom jazyce slovy v jazyce druhém.

5.3.2 Modifikace Brown clusteringu

Autoři [TMU12] upravili pro vícejazyčnost sémantické shlukování z [UB08] které je modifikací Brownova shlukování s tím rozdílem, že pravděpodobnost třídy závisí na předchozím slově a ne na předchozí třídě jako u Brawnova shlukování.

Autoři navrhli dva algoritmy pro vytvoření dvojjazyčných shluků.

1. V jednom jazyce se vytvoří shluky pomocí obyčejného monolinguálního shlukování. Ve druhém jazyce je každému slovu přiřazen takový shluk, na jehož slova je nejčasněji zarovnáno.
2. Slova jsou iterativně monolinguálně shlukována a následně mapována na druhý jazyk střídavě pro oba jazyky.

Autoři [Upa+16] provedli důkladné srovnání metod pro dvojjazyčné sémantické reprezentace slov. Měření úspěšnosti provedli na několika úlohách zpracování přirozeného jazyka. Z jejich výsledků vychází nejlépe právě Bivec.

6 Přístupy podporující vícejazyčnost

Protože trénovací data pro SRL jsou k dispozici jen pro velice malou množinu jazyků, vznikla potřeba vymyslet metody, které buď trénovací data vůbec nepotřebují, nebo dokáží vytvořit model pro více jazyků na základě dat pro jeden jazyk. Vícejazyčné přístupy mají dvě zásadní výhody:

1. Umožňují vytvořit SRL systém i pro jazyky, pro které nejsou k dispozici žádná data.
2. Informace získaná učením v jednom jazyce může v některých případech pomoci modelu pro jiný jazyk.

Tyto přístupy lze rozdělit do tří hlavních kategorií:

1. Učení bez učitele,
2. projekce anotací,
3. přenositelné modely.

6.1 Učení bez učitele

Úlohu SRL s využitím učení bez učitele lze rozdělit na dvě podúlohy:

Identifikace argumentů je většinou provedena na základě lingvistických pravidel.

přiřazení rolí je vždy shlukování argumentů.

6.1.1 Identifikace argumentů

V úloze identifikace argumentů je nejpoužívanější sada lingvistických pravidel publikovaných [LL10] a později vylepšených v [LL11a]. Velkou nevýhodou je,

že pravidla jsou pouze pro angličtinu, takže systémy, které používají tuto sadu pravidel (což je většina), nemohou být použity pro velký počet jazyků[1].

Autoři [ARR09] navrhli metodu učení bez učitele, kde pro identifikaci argumentů využívající unsupervised závislostní stromy a POS tagy. Jejich metoda má bohužel velice nízkou úspěšnost (59% F1 míry).

6.1.2 Přiřazení rolí

Existují tři odlišné přístupy:

1. Grafové přístupy,
2. beyesovské metody,
3. neuronové sítě.

Grafové přístupy

Predikát je reprezentován grafem, kde uzly reprezentují potenciální argumenty (slova). Hrany jsou ohodnocené podle lexikální a syntaktické podobnosti argumentů. Dále se shlukují slova podle ohodnocení. Představiteli těchto přístupů jsou [LL11a] a [LL11b].

Autoři [LL11a] používají split-merge shlukování. Argumenty jsou na začátku v jednom shluku. Následuje rozdělovací fáze, v té je každý argument přiřazen do shluku na základě klíče, který je složením několika příznaků:

1. Slovesný rod (činný / trpný),
2. pozice od predikátu (nalevo / napravo),
3. závislostní relace argumentu,
4. předložka použitá v argumentu.

[1] Pro každý by bylo potřeba napsat vlastní pravidla, což je netriviální úloha.

Po rozdělovací fázi budou tedy v každém shluku slova se shodnými hodnotami výše zmíněných příznaků. Výsledné shluky mají vysokou *purity* (cca 90%), ale nízkou *collocation*.

Následuje několik slučovacích fází. Metrikou pro slučovací fázi je míra podobnosti argumentů. Používá se několik podobností:

- POS,
- lexikální podobnost,
- inverzní překrytí podstromů.

Tyto tři podobnosti se zkombinují a dva shluky s největší podobností jsou spojeny pokud podobnost je větší než určitá hranice

Beyesovské metody

Autoři [TK12] vytvořili dva generativní modely pro přiřazení shluků argumentům. První používá CRP (Chinese restaurant process) a shlukuje argumenty pro každý predikát zvlášť. Druhý používá dd-CRP(distance-dependent chinese restaurant process) a sdílí model mezi predikáty.

Neuronové sítě

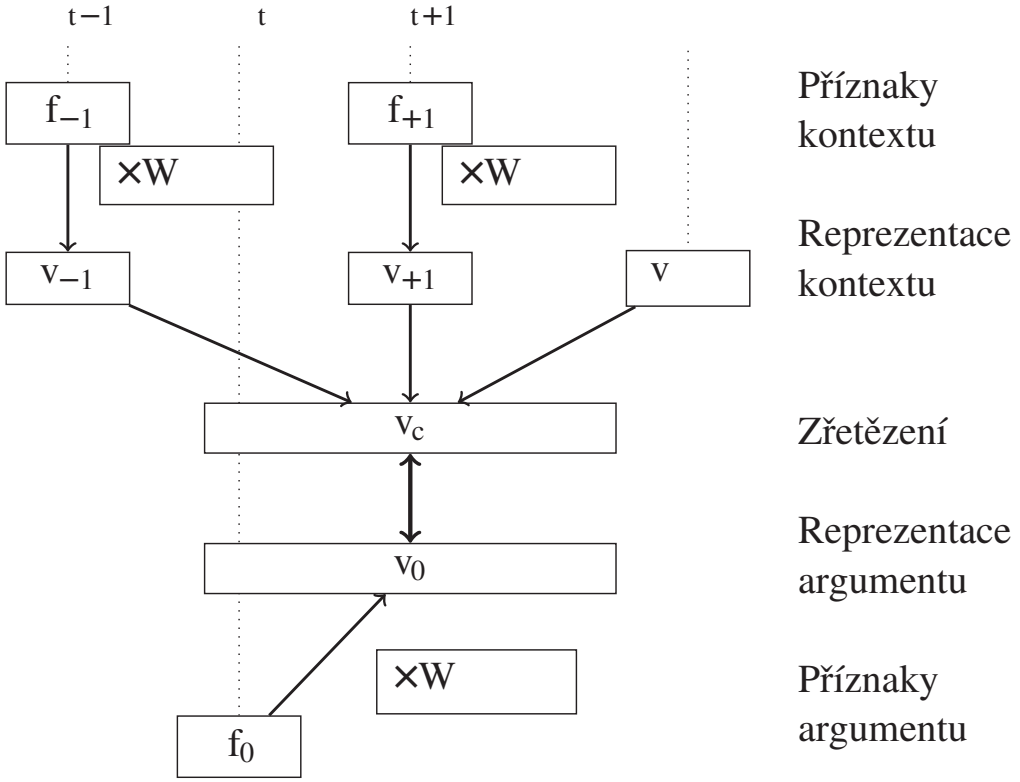
Autoři[WL15] využívají dopřednou neuronovou síť podobnou CBOW k určení vektorové reprezentace predikátů a argumentů.

Nejprve se pomocí dříve zmíněných pravidel určí argumenty. Potom se neuronová síť učí předpovídat aktuální argument na základě predikátu a okolních argumentů do velikosti okna. Predikát na vstupu je určen unikátním vektorem, argumenty na vstupu a výstupu syntaktickými příznaky (relace, POS tag) a sémantickým vektorem slova v kořeni stromu argumentu. Kontextové vektory se vynásobí stejnou váhovou maticí, poskládají se za sebe a vznikne vektor kontextu v_c. Příznaky aktuálního slova se vynásobí druhou váhovou maticí, vznikne vektor v_0. Pravděpodobnost centrálního slova je pak:

$$P(a_c|a_{context}, b) = v_c v_0 \tag{6.1}$$

Cenová funkce sítě je suma záporných logaritmů pravděpodobností jednotlivých argumentů v korpusu (negative log probability).

$$E = -\frac{1}{T}\sum_{i=0}^{T} P(a_i|a_{context}, b_i) \tag{6.2}$$



Obrázek 6.1: Architektura sítě (převzato z [WL15])

Vytvořené vektorové reprezentace se potom shlukují pomocí ILP (Integer linear programming). ILP se skládá ze sady pevných pravidel, která musí být splněna, a optimalizačního kritéria, v tomto konkrétním případě se maximalizuje vzdálenost jednotlivých shluků.

6.1.3 Evaluace

Ačkoliv metody využívají různé modely, téměř vždy se jedná o shlukování na základě ručně zvolených příznaků. Všechny tyto metody najdou pouze shluky. V publikovaných pracích se přiřazení rolí jednotlivým shlukům neřeší a výsledky jsou vyhodnocovány metrikami pro měření kvality shlukování, konkrétně *purity*, *collocation* a jejich harmonický průměr - F1.

Purity vyjadřuje, kolik argumentů ve stejném shluku (procentuálně) má stejnou roli:

$$Pu = \frac{1}{N} \times \sum_i \max_j G_j \cap C_i \tag{6.3}$$

kde G_j je množina argumentů se stejnou rolí, C_i je shluk argumentů a $|G| = |C| = N$ je počet různých rolí.

Collocation naopak vyjadřuje, jak často jsou argumenty se stejnou rolí obsaženy ve stejném shluku:

$$Coll = \frac{1}{N} \times \sum_j \max_i G_j \cap C_i \tag{6.4}$$

Z *purity* a *collocation* se dále počítá F1 míra stejně jako v případě přesnosti a úplnosti, tedy jako harmonický průměr.

$$F1 = \frac{2 \times Pu \times Coll}{Pu + Coll} \tag{6.5}$$

6.2 Projekce anotací

Projekce anotací využívají paralelní korpus. Ten je v jednom jazyce označen sémantickými rolemi, v experimentech ručně, aby bylo možné vyhodnotit

čistě kvalitu projekce, ale principiálně mohou být data označena i automaticky. Anotace paralelního korpusu je poté automaticky přenesena do dalších jazyků.

První aplikace projekce anotací na SRL byla vytvořena v [PL09]. Vstupem je paralelní korpus ve dvou jazycích (testováno na angličtině a němčině). Jeden korpus je označený sémantickými rolemi, cílem je přenést anotaci do druhého jazyka. Základem všech představených metod je zarovnání strojového překladu (IBM model). Autoři experimentují s gold-standard zarovnáním i s automatickým zarovnáním získaným pomocí *GIZA++* [ON03]. Obecné optimalizační kritérium uvádí vzorec 6.6.

$$\hat{A} = \arg\min_{A \in \mathcal{A}} \sum_{(u_s, u_t) \in A} weight(u_s, u_t) \tag{6.6}$$

Kde $\hat{A}$ je nejlepší zarovnání, $\mathcal{A}$ je množina všech zarovnání a u_s respektive u_t je zdrojový respektive cílový uzel v zarovnání.

$$weight(u_s, u_t) = -logsim(u_s, u_t) \tag{6.7}$$

Baseline algoritmus přenese role ze slova na slovo, když jsou slova na sebe zarovnaná. Složitější algoritmy neoperují na úrovni slov, ale na úrovni konstituentů čili podstromů syntaktického stromu. Využívají formalismu bipartitních grafů a algoritmů s nimi spojených. Používají perfektní párování (zobrazení 1:1), pokrytí hran a úplné zarovnání. Dále aplikují několik filtrů (postprocessing):

1. vyplnění mezer – role údajně musí být spojitý řetězec slov, takže přidají všechny slova mezi prvním a posledním;
2. word filter – odstranění některých slov (jednoduchá pravidla).

Autoři [AB10] používají skryté markovské modely na přenesení anotací z angličtiny do italštiny.

6.3 Přenositelné modely

Tyto metody trénují model klasickým strojovým učením s učitelem, viz kapitola 3.3.2. Rozdíl je pouze v tom, že zvolíme takové příznaky, které jsou nezávislé na jazyce (přenositelné mezi jazyky). Potom můžeme model natrénovat na jednom jazyce a používat ho pro označení dat v jazyce jiném. Jako příznaky se v SRL používají:

- vícejazyčné part-of-speech tagy [PDM12]
- jazykově nezávislé mapování slov (slova se na sebe namapují přes slovník získaný z paralelního korpusu)
- vícejazyčné sémantické shluky slov
- využití jazykově nezávislých modelů větné skladby

Přenositelné modely jsou vytvářené pro celou řadu úloh. Například autoři [MPH11] vytvořili přenositelný model pro generování závislostních stromů. Na SRL byl poprvé použit autory [KT13]. Autoři používají POS tagy z [PDM12], slovní mapování a sémantické shluky. O rok později v [KT14] využívají paralelní data v mapování jazykově závislých příznaků.

7 SRL využívající Universal Dependencies

Hlavním cílem této práce je navrhnout a implementovat SRL systém nezávislý na jazyce, který bude využívat Universal Dependencies. K tomu jsou v rámci předzpracování trénovacích a testovacích dat potřeba dva kroky:

1. UD syntaktická anotace – vytvoření UD závislostního stromu. Může být buď manuální (možné pouze v případě trénovacích dat) nebo automatické.
2. Sémantická anotace – v současné době neexistuje korpus označený SRL anotací a UD stromy. Je proto nutné takový korpus vytvořit. V případě trénovacích dat opět možné i manuálně.

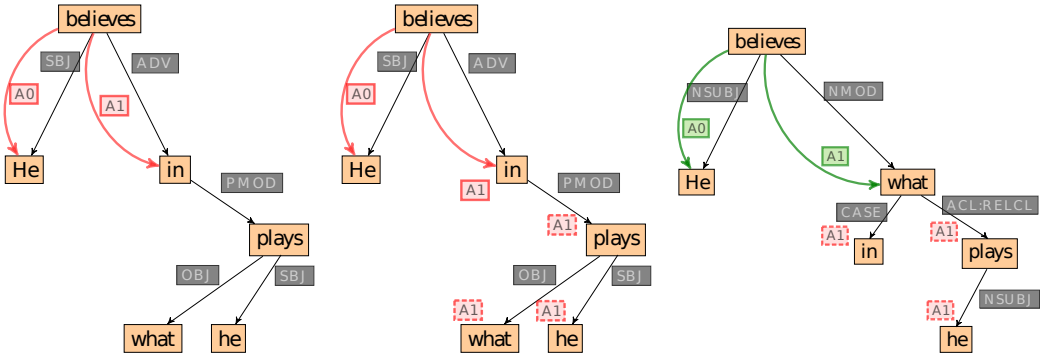
7.1 Vytvoření UD závislostních stromů

Pro vytvoření UD anotace nad trénovacími daty je teoreticky možné využít dva přístupy:

- Automatická pravidlová konverze ručně označeného korpusu v SD.
- Označení UD parserem

7.1.1 Konverze z SD

Pro mnoho jazyků musí teoreticky existovat konverzní skripty z SD do UD. V popisu UD dat je často uvedeno, že byla vytvořena automatickou konverzí z UD. Skripty bohužel nejsou nikde veřejně dostupné a nepodařilo se nám najít ani detailní popis UD pro jejich vlastní implementaci. Proto byl tento postup zavržen a k vytvoření SRL anotací používáme parsery.



Obrázek 7.1: Ukázka SRL anotace: Na obrázku a) je anotace na standardních stromech používaných pro SRL, obrázek (b) ukazuje význam takové anotace a obrázek (c) ukazuje stejnou anotaci na UD stromu. Na obrázku (a) a (b) jsou reálné příklady z CoNLL 2008 (věta 57 v train.closed.conll08 – all), obrázek (c) ukazuje výsledek naší konverzní metody.

7.1.2 UD parsery

Ačkoliv je specifikace Universal Dependencies pořád relativně čerstvá (2016), existuje několik parserů, pro které existují UD modely pro několik jazyků. Jako základní systém pro vytvoření UD stromů používáme *UDPipe* [Str16]. Protože *UDPipe* nemá model pro čínštinu, používáme ještě Stanford parser, který je součástí CoreNLP [Man+14].

7.2 Konverze anotací

SRL anotace je na úrovni podstromů syntaktického stromu a je tedy spjatá s konkrétní syntaktickou anotací. Přenesení anotace na jiné syntaktické stromy proto není triviální. Pro konverzi anotací jsme navrhli 3 metody.

První metoda slouží pouze jako baseline. Sémantické role jsou jednoduše přeneseny do UD na stejná slova která jsou označena v SD.

V druhé konverzní metodě hledáme takovou množinu podstromů, aby množina argumentů v UD byla co nejblíže množině argumentů v původní anotaci. Pro každý argument hledáme takový podstrom, který obsahuje co nejvíce původních argumentů a zároveň obsahuje co nejméně slov, které v původní

anotaci nebyly označené jako argumenty. Tato metoda označí jako argument pouze jeden podstrom a může být tedy použita globální optimalizace. Formální předpis optimalizačního kritéria uvádí vzorec 7.1

$$N^{\star}_{UD} = \underset{N_{UD} \in T_{UD}}{\arg\max} \, | \, \mathtt{C}(N_{UD}) \cap \mathtt{C}(N_{SD}) | - | \, \mathtt{C}(N_{UD}) \setminus \mathtt{C}(N_{SD}) | : \forall N_{SD} \in \mathtt{arg}(T_{SD}), \tag{7.1}$$

kde N_{SD} a N_{UD} jsou uzly původního stromu T_{SD} respektive UD stromu T_{UD}. `C` je operátor pokrytí, který vrací všechny uzly patřící k argumentům (celý podstrom), `arg` vrací všechny argumenty stromu.

Třetí metoda funguje podobně jako druhá, ale bez omezení pouze jednoho podstromu na argument. Algoritmus funguje následovně:

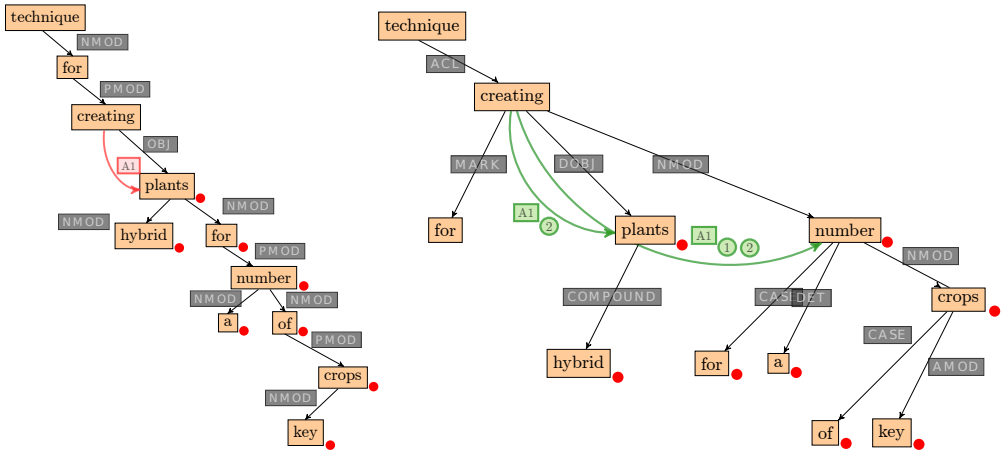
1. Nejprve označíme jako argumenty všechny uzly v podstromu argumentu původního závislostního stromu.
2. Najdeme společné předky argumentů.
3. Odstraníme argumenty, které mají předka, který je stejným argumentem.

Optimalizační kritérium uvádí vzorec 7.2.

$$\{N^{\star}_{UD}\} = \underset{\{N_{UD}\} \in T_{UD}}{\arg\max} \, | \, \mathtt{C}(\{N_{UD}\}) \cap \mathtt{C}(N_{SD}) | - | \, \mathtt{C}(\{N_{UD}\}) \setminus \mathtt{C}(N_{SD}) | : \forall N_{SD} \in \mathtt{arg}(T_{SD}), \tag{7.2}$$

kde $\{N_{UD}\}$ je množina uzlů z T_{UD} a $\{N_{UD}\} \in T_{UD}$ je množina všech možných disjunktních podstromů T_{UD}.

Obrázek 7.2b ukazuje rozdíly mezi konverzními metodami. Druhá metoda označí vždy jen jeden uzel jako argumentu jednoho predikátu jednoho typu, v tomto případě slovo "number". Druhá metoda může označit libovolný počet argumentů, takže označí slova "plants"a "number".



(a) (b)

Obrázek 7.2: Ukázka rozdílu druhé a třetí metody. Tečky ukazují pokrytí argumentu *A1*: na obrázku (a) je původní anotace (věta 117 z train.closed.conll08) a (b) zobrazuje výsledky konverzních metod. – (1) značuje argumenty označené druhou metodou a (2) argumenty označené třetí metodou.

7.3 Univerzální příznaky

Implementovaný systém obsahuje syntaktické a sémantické příznaky příznaky. Syntaktické příznaky se používají následující:

- *Vzdálenost predikátu a argumentu* – Vzdálenost ve smyslu počtu mezilehlých slov.
- *POS* – POS tag predikátu, argumentu a jejich rodičů v závislostním stromu.
- *Závislostní relace* – relace v závislostním stromu predikátu, argumentu a jejich rodičů.
- *Orientovaná cesta* – Cesta v závislostním stromu mezi predikátem a argumentem včetně směrů relací (ke kořeni nebo od kořene).
- *Neorientovaná cesta* – Pouze sekvence relací na cestě mezi predikátem a argumentem.
- *Rod predikátu* – Příznak slovesného rodu (činný / trpný).

- *Další syntaktické příznaky* – `feats` sloupec v *CoNLLu* formátu, který obsahuje další syntaktické příznaky.
- *Bigramové příznaky* – POS tag a závislostní relace predikátu a argumentu jako bigram.

Sémantické příznaky:

- Lemma – Základní tvar kořene podstromu argumentu.
- Sémantické shluky – Příslušnost argumentu k sémantické třídě.

Další příznaky:

- *Stromové příznaky* – Relace, POS tagy, lemmata a sémantické shluky v celém podstromu argumentu, či s omezením jen do určité hloubky. Zakódováno jako bag-of-words.

8 Vytvořený SRL systém

V rámci práce byl vytvořen kompletní systém pro značkování sémantických rolí. V této kapitole jsou popsány základní principy implementovaného systému.

8.1 Datový model

Model úlohy je inspirován formátem CoNLL. formátů CoNLL je několik, mají však společné základní charakteristiky. Věta je modelována jako posloupnost slov a všechny syntaktické i sémantické anotace jsou na úrovni slov. V datovém modelu je tedy korpus reprezentovaný seznamem vět. Datový model jedné věty je znázorněn na obrázku 8.1.

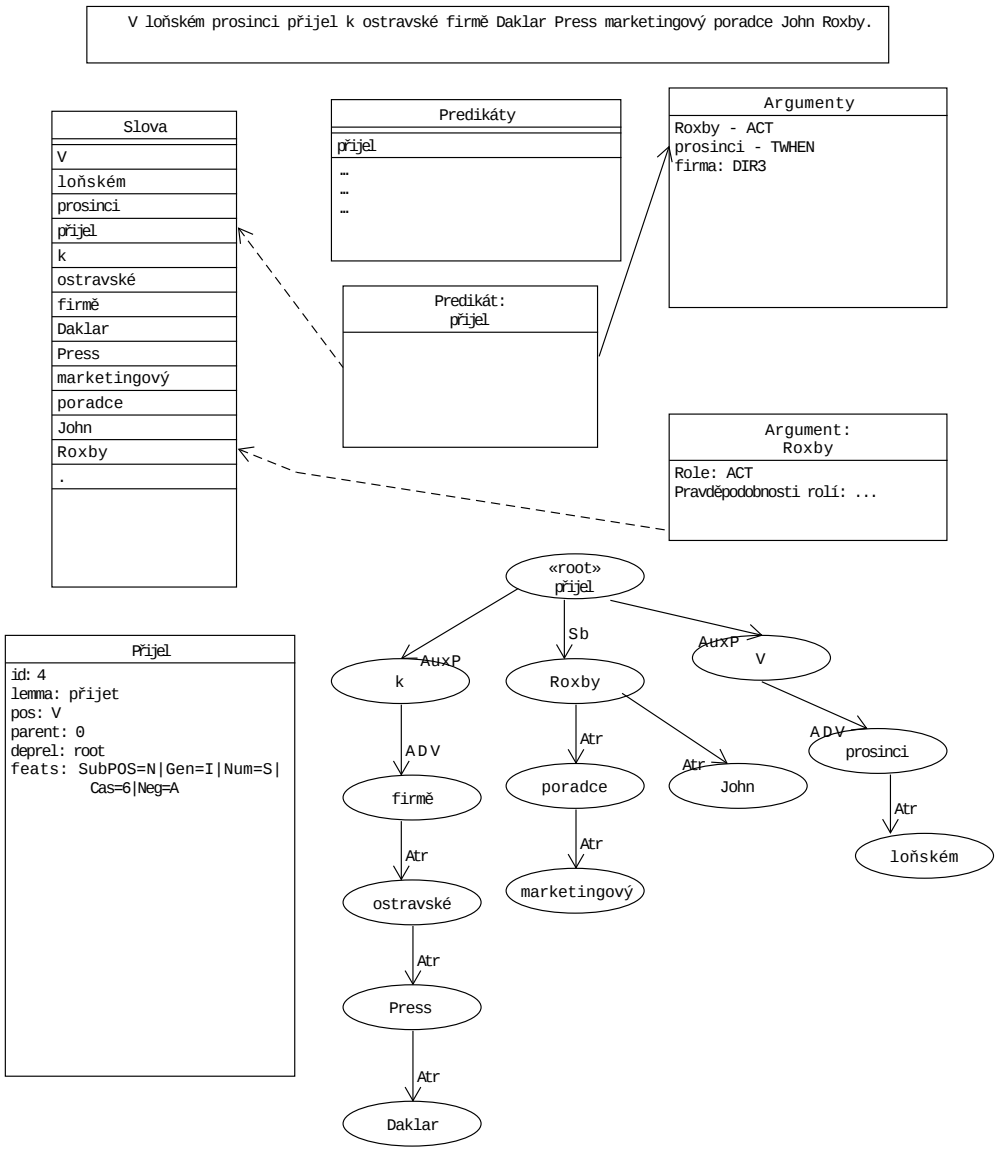
Každá věta obsahuje seznam slov a seznam predikátů. Každý predikát obsahuje seznam svých argumentů a každý argument obsahuje číslo a značku své role.

Každé slovo obsahuje minimálně následující anotace:

- Původní tvar (form),
- základní tvar (lemma),
- part-of-speech tag (POS),
- ID předka v závislostním stromu (headId),
- odkaz na předka v závislostním stromu (head),
- závislostní relaci s předkem (deprel).

Slovo může obsahovat další anotace uvnitř dynamické mapy anotací (annotations).

Dále slovo obsahuje základní metody pro procházení závislostního stromu jako:



Obrázek 8.1: Grafické znázornění modelu úlohy

- nalezení kořene,
- nalezení cesty ke kořeni,
- nalezení cesty mezi dvěma uzly.

Cesta je reprezentovaná seznamem uzlů, které obsahují slovo, název relace a

směr relace (ke kořeni nebo od kořene).

8.2 Zpracování vstupů a výstupů

Jak již bylo řečeno dříve, existuje několik formátů CoNLL a různé aplikace používají různé formáty. Ve všech CoNLL formátech je každé slovo na samostatném řádku, věty jsou pak oddělené prázdným řádkem. Jednotlivé formáty CoNLL se liší pouze ve způsobu popisu jednotlivých slov. Aplikace obsahuje parsery pro následující formáty:

- **CoNLL 2008**,
- **CoNLL 2009**,
- **CoNLLu**,
- **TSV formát** používaný Stanford taggerem,
- **Čistý text** – je potřeba nastavit předzpracování textu, viz kapitolu 8.2.1.

Pro jednoduchost zpracování obsahuje aplikace ještě univerzální reader a writer, které podle přípony souboru automaticky určí formát a vyberou odpovídající reader respektive writer.

8.2.1 Předzpracování

Aplikace umožňuje několik typů předzpracování:

- **Oddělení vět (sentence split)** - vstupem je čistý text, výstupem je model korpusu (seznam vět)
- **Tokenizace** - rozdělení věty na seznam tokenů (slov)
- **Předzpracování vět** - vstupem je celá věta. Jedná se o nejběžnější typ předzpracování. Používá se pro metody vyžadující kontext (lemmatizace, POS tagging, parsing).

- **Předzpracování slov** - Bezkontextové zpracování slov. Není příliš časté, může jít například o filtr, který odstraní určitá slova.
- **Předzpracování predikátů** - Stejně jako předchozí není příliš běžné. V aplikaci se používá například filtr pro odstranění neslovesných predikátů.
- **Předzpracování argumentů** - Bezkontextové zpracování argumentů. Používá se například pro mapování rolí mezi různými datasety.

Metody předzpracování, které se nastaví v konfiguraci (viz kapitolu 8.3), se aplikují po načtení dat.

8.3 Konfigurace

Pro účely snadného testování je aplikace konfigurovatelná pomocí jednoho globálního konfiguračního souboru *config.properties*. Aby byla aplikace dostatečně univerzální a její části použitelné formou knihovny, ke globální konfiguraci přistupují pouze spustitelné třídy a celá aplikace je konfigurovatelná programově. Také je snadno možné vytvořit alternativní spouštěče, které budou konfiguraci načítat například z argumentů příkazové řádky. Globální konfiguraci využívá i správa experimentů, která je popsaná v kapitole 8.4.

8.4 Správa experimentů

Kvůli častému spouštění různých experimentů s velkým množstvím parametrů vznikla potřeba detailní kontroly těchto experimentů. Proto byl vytvořen systém pro správu experimentů. Hlavní výhody tohoto systému jsou:

- **Opakovatelnost experimentů** – konfigurace, se kterou byl experiment spuštěn, se automaticky archivuje.
- **Snadné dohledání přesných výsledků starého experimentu** – archivovány jsou také záznamy výsledků, včetně logů všech částí aplikace.

- **Automatická archivace mezivýsledků jednotlivých kroků experimentu**
- **Záznam přesného stavu aplikace v době spuštění experimentu** – záznam experimentu mimo jiné obsahuje poslední revizi Git repozitáře, takže je možné dohledat přesný stav aplikace v době spuštění experimentu.
- **Inteligentní plánování kroků** – zatím není implementováno. Při návrhu experimentů jsem si všiml, že velké množství kroků je spouštěno v různých experimentech s naprosto stejnou konfigurací. Do budoucna tedy chci vytvořit plánovač, který projde jednotlivé kroky všech spuštěných experimentů, najde duplicity, odstraní je a vytvoří efektivní plán spuštění jednotlivých kroků.

Systém modeluje experimenty jako posloupnost jednoduchých kroků, kde každý jednoduchý krok představuje spuštění jedné samostatné aplikace. Krok má svůj konfigurační soubor (globální konfigurace, viz kapitolu 8.3), který mimo jiné obsahuje název třídy, která bude spuštěna. Tato třída musí obsahovat metodu *main()*.

Experiment může dále obsahovat složené kroky (advancedStep), které jsou opět posloupností jednoduchých kroků, ale mohou navíc obsahovat proměnné. Tyto proměnné jsou sdílené mezi všemi kroky. Příkladem složeného kroku v aplikaci je krok *evalAll*, který spouští všechny evaluace. Vstupem každé evaluace je správně označený soubor a soubor označený systémem, to jsou ony společné proměnné. Díky složeným krokům nemusíme v každém experimentu konfigurovat všechny možné evaluace, ale stačí nekonfigurovat krok *evalAll*.

8.5 Strojové učení

Základní klasifikační část úlohy je řešena dvěma klasifikátory. První je binární klasifikátor, jehož úkolem je pro každou dvojici predikát slovo věty rozhodnout, jestli jestli je slovo argumentem daného predikátu

Druhá úloha je klasifikace argumentů do tříd podle sémantické značky.

Pro strojové učení byla použita knihovna *Brainy* [Kon14].

Klasifikovanou entitou v SRL je dvojice predikát-argument. Klasifikátor pro spojování argumentů používaný pro globální optimalizaci klasifikuje dvojici argumentů. Protože klasifikátory mají některé příznaky společné, obě tyto dvojice implementují rozhraní dvojice slov, nad kterými jsou implementované společné příznaky (respektive extrakce příznaků).

8.5.1 Reprezentace příznaků

Brainy nenabízí metody pro transformaci příznaků. Proto jsem základní metody pro transformaci příznaků doprogramoval:

- **Bag-of-words reprezentace** – Vstupem může být buď slovo, nebo seznam slov. Výstupem je vektor o velikosti slovníku, kde každá složka vektoru udává počet odpovídajících slov ve vstupním textu. Umožňuje nastavit omezení, kolikrát se musí slovo minimálně vyskytovat v trénovacích datech, aby bylo uvažováno.
- **Rovnoměrné rozdělení intervalu** – Číselné hodnoty jsou rozdělené do stejně velkých skupin (s metrikou počtu prvků) a každé číslo je převedeno na one-hot reprezentaci příslušnosti ke skupině.
- **Ekvidistantní rozdělení intervalu** – Rozdělí číselné hodnoty do stejně velkých skupin (s metrikou rozsahu intervalů) a každé číslo je převedeno na one-hot reprezentaci příslušnosti ke skupině.
- **Pravděpodobnosti příslušnosti ke třídám podmíněné původními příznaky** – Pravděpodobnost příslušnosti ke třídě podmíněná příznaky - globální příznak, který sdílí váhy pro všechny hodnoty příznaků.

8.6 SRL

8.6.1 Identifikace argumentů

V systému jsou implementované tři metody pro identifikaci argumentů.

1. Standardní přístup učení s učitelem

2. Pravidlový přístup navržený v [LL11a].

3. Kombinace předchozích dvou - Nejprve se identifikují argumenty pomocí lingvistických pravidel. Na výsledku se poté natrénuje klasifikátor a ten je znovu spuštěn nad stejnými daty. Hypotéza je taková, že statistické zpracování klasifikátoru může odstranit některé chyby pravidel.

8.6.2 Určení rolí

Určení rolí je v systému zatím řešeno pouze učením s učitelem. V systému jsou implementované dvě metody pro určení rolí.

1. S jedním klasifikátorem pro všechny predikáty.

2. Se samostatnými klasifikátory pro často se opakující predikáty. Tedy predikát, který se vyskytuje v datech více než stokrát (nastavitelná mez), má argumenty a jejich role klasifikované klasifikátorem natrénovaným pouze na onom predikátu. Samozřejmě, když se v trénovacích datech vyskytuje daný predikát pouze několikrát, nemá smysl pro něj používat samostatný klasifikátor a v takovém případě se použije klasifikátor globální (netrénovaný na všech datech).

8.6.3 Globální optimalizace

Aplikace obsahuje heuristický algoritmus pro globální optimalizaci. jedná se o jednoduché zkoušení všech kombinací rolí s nastaveným omezením, že zkouší pouze N nejpravděpodobnějších rolí pro každý argument.

Globální optimalizace v základní podobě není aplikovatelná na data upravená třetí konverzní metodou, kde jeden argument může být složen z více podstromů závislostního stromu.

Tento problém řešíme tak, že přidáváme další klasifikátor, jehož úkolem je rozhodnout, jestli dva podstromy jsou jedním argumentem nebo ne. V případě pozitivní klasifikace argumenty spojíme do jednoho a nové pravděpodobnosti rolí spočteme jako průměr pravděpodobností spojovaných argumentů.

8.6.4 Podpora více jazyků

V aplikaci jsou v současné době zaintegrované dva UD parsery. Stanford parser jako součást CoreNLP a UDPipe. Parsery jsou samozřejmě závislé na jazyce[1], proto je potřeba pro každý jazyk definovat příslušné modely. Aplikace proto musí obsahovat definice všech podporovaných jazyků. Kromě modelů parserů definuje ještě mapování rolí pro dvojjazyčné experimenty.

8.7 Evaluace

Aplikace specifikuje dva druhy evaluací:

1. Evaluace v paměti (in-memory evaluation) – Vstupem evaluace je seznam vět, během evaluace je celý dataset uložený v RAM, evaluace je rychlejší, ale při práci s velkými daty je velice paměťově náročná
2. Evaluace souborů – Vstupem evaluace jsou jména souborů a evaluace tedy může data zpracovávat proudově (ale práce se soubory je pomalejší a nelze efektivně paralelizovat).

Výsledek evaluace se vypisuje na standardní výstup, který je v případě spuštění experimentu přesměrován do souboru.

[1] To je největší jazykové omezení našeho systému - dokáže zpracovávat pouze jazyky, pro které existuje UD parser.

9 Experimenty

Navrhli jsme sadu experimentů, jejichž cílem je:

- Ověřit správnost implementace SRL systému (experiment 9.4.1).
- Zhodnotit a srovnat úspěšnost našich konverzních metod (experiment 9.3).
- Ověřit vhodnost Universal Dependencies pro značkování sémantických rolí (experiment 9.4.3).
- Zhodnotit úspěšnost navrženého vícejazyčného SRL systému (experiment 9.5).

Jelikož primárním cílem experimentů je ověřit vhodnost UD pro vícejazyčné značkování sémantických rolí, obsahují všechny experimenty, kromě těch zaměřených na lexikální příznaky, pouze syntaktické příznaky. Používají se všechny syntaktické příznaky uvedené v kapitole 7.3.

Všechny experimenty jsou vyhodnocovány na pěti jazycích. Konkrétně na angličtině (EN), češtině (CZ), němčině (DE), španělštině (ES) a čínštině (ZH). Výsledky jsou vyhodnocovány zvlášť pro slovesné predikáty (řádek V) a ostatní predikáty (řádek O). Řádek A udává výsledky pro všechny predikáty dohromady.

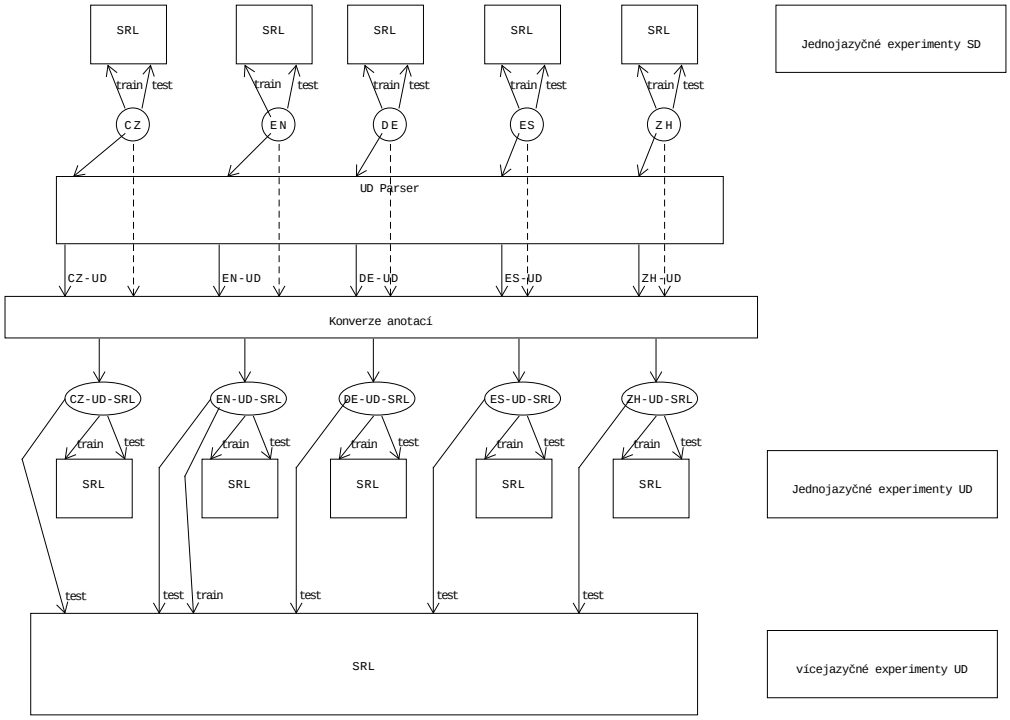
Princip experimentů je schématicky znázorněn na obrázku 9.1.

9.1 Datové kolekce

Všechny jednojazyčné experimenty používají data z CoNLL 2009. Pro evaluaci přenosu modelu z angličtiny do češtiny byla použita data z PCEDT pro trénování a česká data z CoNLL 2009 pro evaluaci. Tabulka 9.1 uvádí statistiky jednotlivých datových kolekcí.

train / test	CZ	EN	DE
počet vět	38 727 / 4 213	39 279 / 2 399	36 020 / 2 000
počet predikátů	414 237 / 44 585	179 014 / 10 498	17 400 / 550
verb	65 331 / 7 322	89 193 / 5 217	17 400 / 550
nonverb	348 906 / 37 263	89 821 / 5 281	0 / 0
prům. argumentů	0.882 / 0.88	2.2 / 2.22	1.97 / 1.95
max. argumentů	26 / 12	19 / 7	6 / 5
nejčastější role	RSTR(30.13%)	A1(37.22%)	A0(40.39%)
	PAT(18.36%)	A0(25.24%)	A1(39.46%)
	ACT(16.72%)	A2(11.87%)	A2(11.84%)
	APP(6.38%)	AM-TMP(5.93%)	A3(5.83%)
	LOC(4.43%)	AM-MNR(3%)	A4(1.28%)
train / test	ES	PCEDT	ZH
počet vět	14 329 / 1 725	30 149	22 277 / 2 556
počet predikátů	43 824 / 5 175	81 645	102 813 / 12 282
verb	40 887 / 4 831	81 645	102 809 / 12 282
nonverb	2 937 / 344	0	4 / 0
prům. argumentů	2.26 / 2.28	2.07	2.26 / 2.26
max. argumentů	8 / 7	18	11 / 9
nejčastější role	arg1-pat(20.46%)	ACT(31.8%)	A1(30.43%)
	arg0-agt(18.88%)	PAT(31.49%)	A0(26.56%)
	arg1-tem(14.72%)	TWHEN(6.22%)	ADV(19.84%)
	arg2-atr(8.20%)	EFF(5.88%)	TMP(6.61%)
	argM-tmp(8.14%)	LOC(3.65%)	DIS(4.44%)

Tabulka 9.1: Statistiky datových kolekcí



Obrázek 9.1: Scématické znázornění experimentů

9.2 Evaluační metriky

Abychom mohli naše výsledky srovnat se zástupci všech relevantních zdrojů, potřebujeme několik evaluačních metrik:

- labeled a unlabeled F1 míra vypočítaná oficiálním evaluačním skriptem z CoNLL 2009[1] (*ls* a *us*).
- F1 míra pro evaluaci fáze identifikace argumentů. Standardní F1 míru uvádí vzorec 2.1 na straně 6. Tato naše míra není počítaná pouze na kořenech podstromů argumentů jako v případě oficiální evaluační metriky, ale vyhodnocují se všechny uzly podstromů argumentů. Takto je ve výsledku započítaná i chyba závislostního stromu. Metriku počítáme tímto způsobem proto, že nemáme k dispozici ručně označená data SRL pro UD (*u*).

[1] stažen z: http://ufal.mff.cuni.cz/conll2009-st/scorer.html

- Accuracy pro evaluaci fáze přiřazení rolí (viz vzorec 2.4 na straně 7). Vyhodnocuje se pouze pro správně identifikované argumenty. Míra je opět počítána na celých podstromech (l).
- F1 míra pro evaluaci shlukování tak, jak byla definovaná v kapitole 6.1.3 na straně 24. Počítaná na celých podstromech pouze na správně identifikovaných argumentech (F_1^c).

9.3 Konverze anotací

První experiment spočívá v aplikaci všech navržených konverzních metod na data ve všech testovaných jazycích a vypočítání všech dále používaných evaluačních metrik

Cílem experimentu je srovnat úspěšnost navržených konverzních metod a určit horní hranici úspěšnosti dalších experimentů.

Výsledky

Tabulka 9.2 ukazuje výsledky jednotlivých konverzních metod.

Diskuze

Třetí konverzní metoda dosahuje značně lepších výsledků než metoda druhá a baseline. To je očekávaný výsledek, protože anotace vytvořená třetí metodou je obecnější (může označit více podstromů jako jeden argument). Na druhou stranu například anotace anglického korpusu nedovoluje, aby se argument skládal z více podstromů a také to znesnadňuje globální optimalizaci (viz kapitolu 8.6.3 na straně 38). Ta ale na naše experimenty neměla příliš velký vliv, proto jsme pro další experimenty zvolili právě třetí metodu. Tabulka nám také udává teoretickou horní hranici úspěšnosti SRL systému využívajícího Universal Dependencies.

			CZ	EN	DE	ES	ZH
1	u	V	86.70	70.31	68.99	72.78	83.17
		O	87.86	69.33	—	—	-
		A	87.19	69.98	68.99	72.78	83.17
	l	V	99.00	87.86	96.25	93.31	97.42
		O	99.41	85.18	—	—	—
		A	99.17	86.97	96.25	93.31	97.42
2	u	V	87.10	84.19	81.16	83.79	85.63
		O	85.29	81.52	—	—	-
		A	86.32	83.32	81.16	83.79	85.63
	l	V	98.78	92.75	97.28	96.40	97.81
		O	99.01	89.75	—	—	—
		A	98.88	91.79	97.28	96.40	97.81
3	u	V	89.10	95.06	93.43	91.08	89.87
		O	87.80	89.94	—	—	—
		A	88.55	93.38	93.43	91.08	89.87
	l	V	100.00	100.00	100.00	100.00	100.00
		O	100.00	100.00	—	—	—
		A	100.00	100.00	100.00	100.00	100.00

Tabulka 9.2: Úspěšnost konverze anotací.

9.4 Jednojazyčné experimenty

Pro zhodnocení obecných vlastností SRL systému a evaluaci vhodnosti UD pro SRL jsme navrhli sadu jednojazyčných experimentů.

9.4.1 Gold-standard SD

V prvním jednojazyčném experimentu testujeme náš SRL systém ve stejných podmínkách, jaké byly na CoNLL 2009 (SRL-only task). V experimentu používáme ručně vytvořené stromy, lemmata i POS tagy z CoNLL datasetů. Nejsou použité žádné lexikální příznaky. Používají se naprosto stejné příznaky jako v pozdějších vícejazyčných experimentech. Účelem tohoto experimentu je srovnat implementovaný SRL systém se state-of-the-art a tak ověřit, jestli je vhodný pro další experimenty. Oficiální evaluační metrika slouží ke srovnání s ostatními systémy. Ostatní metriky jsou naměřené pro srovnání s výsledky dalších experimentů a pro představu jaký je vztah mezi oficiální

metrikou a ostatními použitými metrikami, protože v dalších experimentech už není oficiální metrika použitelná.

Výsledky

Tabulka 9.3 ukazuje výsledky jednojazyčných experimentů s použitím ručně označených lemmat POS tagů a stromů. Sloupec *ls Zhao* uvádí výsledky nejlepšího systému z *CoNLL 2009*.

		CZ	EN	DE	ES	ZH
us	A	96.12	91.75	90.42	100.00	92.08
ls	A	87.02	80.52	72.48	75.62	81.73
ls Zhao	A	85.19	85.44	75.99	80.46	78.15
u	V	94.06	95.82	86.52	100.00	87.81
	O	91.41	78.52	—	—	—
	A	92.97	91.07	86.52	100.00	87.81
l	V	74.16	86.02	65.95	69.02	82.99
	O	76.51	79.11	—	—	—
	A	75.11	83.28	65.95	69.02	82.99
F_1^c	V	93.05	93.64	95.09	93.00	92.98
	O	95.38	88.56	—	—	—
	A	94.83	91.10	95.09	93.00	92.99

Tabulka 9.3: Výsledky s manuálně vytvořenými SD stromy

Diskuze

Z výsledků je vidět, že náš systém dosahuje téměř srovnatelných výsledků s nejlepšími systémy na CoNLL 2009. Trochu horší výsledky jsou pravděpodobně způsobené opuštěním lexikálních příznaků. Výsledky jsou ale dostatečně dobré pro další experimenty.

9.4.2 System SD

V tomto experimentu měříme úspěšnost systému na systémem označených datech (nemáme dopředu označené stromy, lemmata ani POS tagy). Výsledky měříme na datech z jiné domény, abychom se podmínkami co nejvíce přiblížili

finálním vícejazyčným experimentům, kde nemáme k dispozici data ze stejné domény.

Výsledky

Tabulka 9.4 uvádí výsledky na SD bez ručně označených dat.

		CZ	EN	DE	ES	ZH
us	A	91.94	86.27	86.24	93.49	81.25
ls	A	83.14	70.59	67.80	69.81	72.20
F_1^c	V	77.76	79.48	72.97	82.42	66.53
	O	84.26	66.45	90.00	82.23	75.54
	A	83.07	72.74	73.00	82.42	66.58
u	V	76.41	90.26	79.03	90.52	66.33
	O	86.09	69.46	90.00	68.26	58.60
	A	81.88	84.54	79.08	90.46	65.84
l	V	66.37	75.04	59.17	62.91	77.30
	O	74.31	62.07	00.00	-	65.28
	A	69.60	72.11	58.85	62.91	76.62

Tabulka 9.4: Výsledky s automatickými SD stromy

Diskuze

Výsledky na automaticky vytvořených SD stromech jsou výrazně horší, než výsledky na manuálně vytvořených. Z toho plyne, že hodně chyb SRL je způsobeno chybami ve vytvořených stromech. Věty v datech CoNLL jsou hodně složité a parser v nich udělá hodně chyb. Dalším důležitým faktorem je vypuštění lexikálních příznaků. Náš model, využívající pouze syntaktické příznaky, je přirozeně mnohem více závislý na syntaktickém stromu. Přirozeným závěrem je, že by bylo potřeba mít k dispozici ručně vytvořené UD stromy v SRL anotovaném korpusu (abychom mohli natrénovat systém na manuálně vytvořených datech, jeho úspěšnost bude pravděpodobně výrazně vyšší).

9.4.3 System UD

Experiment se zaměřuje na jednojazyčné SRL využívající Universal Dependencies. Pro konverzi SRL anotace na UD se používá konverzní metoda, která

dopadne nejlépe v prvním experimentu. Účelem experimentu je zjistit, jestli jsou UD použitelné pro SRL a dále nabídnout další srovnání k dvojjazyčným experimentům.

Výsledky

Tabulka 9.5 obsahuje výsledky s použitím UD stromů a nejlepší konverzní metody (tedy té třetí).

		CZ	EN	DE	ES	ZH
u	V	81.11	83.29	75.19	82.58	67.44
	O	72.12	59.01	—	—	—
	A	77.38	76.51	75.19	82.58	67.44
l	V	63.13	68.70	53.79	53.07	77.71
	O	69.51	64.05	—	—	—
	A	65.60	67.69	53.97	53.07	77.71
F_1^c	V	86.76	83.33	85.75	83.87	89.18
	O	93.22	85.33	—	—	—
	A	91.62	83.72	85.75	83.87	89.18

Tabulka 9.5: Výsledky s automatickými UD stromy

Diskuze

Experimenty na UD stromech (tabulka 9.5) ukazují, že Universal Dependencies jsou použitelné pro SRL. Systém na UD dosahuje srovnatelných výsledků s SD (s případem použití automatických stromů).

9.5 Dvojjazyčné experimenty

Hlavními experimenty této práce jsou experimenty dvojjazyčné, kde natrénujeme model na angličtině (protože anglická trénovací data jsou největší) a pomocí vytvořeného modelu označujeme data ve všech ostatních jazycích. Pokud datasety mají stejnou sadu rolí (angličtina-čínština, angličtina-čeština PCEDT), vyhodnocujeme určení rolí pomocí *accuracy*, jinak pouze pomocí F1 míry pro evaluaci shlukování. Ta pro každý unikátní predikát zvlášť najde optimální zarovnání rolí.

Výsledky

Výsledky uvádí tabulka 9.6 Pro všechny jazyky byl použitý stejný model natrénovaný na trénovací části CoNLL 2009 pro angličtinu. Řádek *K* uvádí odpovídající výsledky z [KT13]. Sloupce *EN-CZ*, *EN-ES* a *EN-DE* nemají změřenou *accuracy*, protože datasety neobsahují stejnou množinu sémantických rolí. Proto je v těchto případech počítaná pouze F1 míra pro identifikaci argumentů a shlukování.

		EN-CZ	EN-ES	EN-ZH	EN-DE	PCEDT
u	V	76.28	75.67	64.08	71.66	78.97
	K	—	—	51.70	—	63.90
l	V	—	—	75.57	—	55.09
	K	—	—	71.70	—	59.00
F_1^c	V	84.90	81.78	87.94	82.37	83.03
	K	—	—	84.50	—	74.10

Tabulka 9.6: Výsledky dvojjazyčných experimentů s UD stromy

Diskuze

Výsledky nejdůležitějšího experimentu této práce (tabulka 9.6), výsledky dvoujazyčných experimentů ukazují jednak, že úloha identifikace argumentů je dobře přenositelná mezi jazyky, výsledky fáze identifikace argumentů jsou u všech jazyků velice dobré. V obou srovnatelných experimentech dosahují výrazně lepších výsledků než [KT13]. Výsledky sice nejsou přesně porovnatelné, protože v [KT13] vyhodnocovali výsledky pouze na kořenech podstromů a my vyhodnocujeme metriky na celých podstromech argumentů[2]. Z tabulky 9.3 je však vidět, že výsledky obou metrik vycházejí podobně. Ve fázi přiřazení rolí dosahuje náš systém a systém prezentovaný v [KT13] srovnatelných výsledků. Oba systémy fungují mnohem lépe na čínštině než na češtině.

Výsledky experimentů měřené evaluační mírou pro shlukování ukazují, že přenos sémantických rolí z angličtiny do ostatních jazyků je možný. Dobré výsledky této míry říkají, že existují mapování (specifická pro predikáty) mezi sémantickými rolemi ve všech testovaných jazycích.

[2]Vyhodnocování na kořenech nedává smysl, protože máme jiné stromy než jsou v evaluačních datech.

9.6 Lexikální příznaky

Pro zhodnocení důležitosti lexikálních příznaků provádíme experimenty využívající slovní příznaky, sémantické shluky a samostatné klasifikátory pro často se opakující predikáty.

Výsledky

Tabulka 9.7 ukazuje výsledky na ručně vytvořených stromech s použitím lexikálních příznaků. Tabulka 9.8 pak výsledky jednojazyčných experimentů s využitím Universal Dependencies a lexikálních příznaků.

		CZ	EN	DE	ES	ZH
us	A	96.39	92.09	92.66	100.00	92.26
ls	A	89.52	83.93	74.99	83.83	84.24
ls Zhao	A	85.19	85.44	75.99	80.46	78.15
u	V	93.95	95.66	91.21	100.00	88.09
	O	91.48	78.97	—	—	—
	A	92.93	91.03	91.21	100.00	88.09
l	V	80.26	88.76	69.32	79.01	87.13
	O	81.33	81.09	—	—	—
	A	80.70	86.91	69.32	79.01	87.13
F_1^c	V	94.18	94.53	94.20	95.21	93.95
	O	95.57	92.54	—	—	—
	A	95.24	93.39	94.20	95.17	93.95

Tabulka 9.7: Výsledky s ručně vytvořenými SD stromy s lexikálními příznaky.

Diskuze

Z výsledků je vidět, že lexikální příznaky mají nemalý význam a bylo by dobré využít je i u vícejazyčných experimentů (na UD jsou výsledky přiřazení rolí lepší až o 9%). V současnosti s nimi experimentujeme, ale bohužel se je z časových důvodů zatím nepodařilo zapojit.

		CZ	EN	DE	ES	ZH
u	V	81.11	83.31	76.60	82.17	67.12
	O	72.12	61.11	—	60.34	—
	A	77.38	76.95	76.60	81.26	67.12
l	V	72.35	73.61	57.08	62.38	82.77
	O	78.33	71.70	—	43.50	—
	A	74.66	73.17	57.08	61.90	82.77
F_1^c	V	88.43	85.25	86.16	85.05	90.79
	O	94.47	88.88	—	95.74	—
	A	92.97	86.38	86.16	85.58	90.79

Tabulka 9.8: Výsledky s UD stromy a lexikálními příznaky.

10 Závěr

Hlavním cílem práce bylo navrhnout, implementovat a otestovat metodu pro vícejazyčné značkování sémantických rolí založenou na Universal Dependencies. Byl vytvořen základní SRL systém, ten se ukázal jako srovnatelný s jinými existujícími SRL systémy. Dále byly v práci navržené metody pro konverzi SRL anotace do Universal Dependencies. Nakonec byl navržen a implementován systém vícejazyčného SRL (přenositelný model) postavený na UD. Výsledky vypadají velice slibně. Navržená metoda dosahuje srovnatelných výsledků s vícejazyčným state-of-the-art a přináší několik výhod:

1. Možnost vytvořit model pro všechny jazyky, pro které existuje UD parser.
2. Model je možné natrénovat na libovolné podmnožině jazyků současně. Předchozí metody jsou pouze dvojjazyčné.

Ve studované oblasti je stále velký prostor pro zlepšení. Některá možná vylepšení uvádí následující kapitola.

10.1 Možná vylepšení

V rámci práce byl vytvořen komplexní systém pro značkování sémantických rolí. Byla provedena řada experimentů ověřující použitelnost Universal Dependencies pro SRL. Universal Dependencies se ukázaly jako velice dobré pro vícejazyčné SRL a nabízí spoustu možností dalšího vylepšení, například:

- Použít pro vícejazyčné SRL lexikální příznaky zmíněné v teoretické části práce.
- Vyzkoušet modely natrénované na jiném jazyce než na angličtině.
- Vyzkoušet modely natrénované na více jazycích současně.
- Vytvořit online demo. V podstatě připraveno, chybí pouze uživatelské rozhraní

- Sémantické vektory se v dřívějších pracích ukázaly jako významné příznaky. Jejich použití ve vícejazyčných experimentech pravděpodobně zvýší úspěšnost systému.
- Vyzkoušet experimenty s ručně vytvořenou UD syntaktickou anotací (žádný SRL UD korpus neexistuje, bude potřeba ho vytvořit).
- Nashlukovat predikáty a natrénovat modely zvlášť pro každý shluk. To vyřeší problém samostatných klasifikátorů pro každý predikát, kde pro některé predikáty je málo trénovacích dat a některé predikáty nejsou v trénovacích datech vůbec.

Seznam zkratek

SRL Semantic role labeling (Značkování sémantických rolí)

NLP Natural language processing (Zpracování přirozeného jazyka)

UD Universal Dependencies

SD Standard dependencies - v textu myšleno jako závislostní stromu běžně používané v SRL

CoNLL Conference on Computational Natural Language Learning

LSA Latent semantic analysis

LDA Latent Dirichlet allocation

HAL Hyperspace analogue to language

Seznam tabulek

Seznam obrázků

Seznam vzorců

Bibliografie

[AB10] Paolo Annesi a Roberto Basili. Cross-lingual alignment of FrameNet annotations through Hidden Markov Models. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Sv. 6008 LNCS. CICLing'10. Berlin, Heidelberg: Springer-Verlag, 2010, s. 12–25. ISBN: 3642121152.

[ARR09] Omri Abend, Roi Reichart a Ari Rappoport. Unsupervised Argument Identification for Semantic Role Labeling. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. August. Association for Computational Linguistics. 2009, s. 28–36. ISBN: 9781932432459.

[Ben+03] Yoshua Bengio et al. A Neural Probabilistic Language Model. In: *The Journal of Machine Learning Research* 3 (2003), s. 1137–1155. ISSN: 15324435. DOI: 10.1162/153244303322533223. arXiv: arXiv:1301.3781v3.

[BFL98] Collin F. Cf Baker, Charles J. Fillmore a John B. Lowe. The Berkeley FrameNet Project. In: *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -. Sv. 1. 1998, s. 86. DOI: 10.3115/980845.980860.

[Boj+16] Ondřej Bojar et al. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In: *Text, Speech, and Dialogue: 19th International Conference, {TSD} 2016*. Ed. Petr Sojka et al. Lecture Notes in Artificial Intelligence 9924. Masaryk University. Cham / Heidelberg / New York / Dordrecht / London: Springer International Publishing, 2016, s. 231–238. ISBN: 978-3-319-45509-9.

[Gil02] Daniel Gildea. Automatic labeling of semantic roles. In: *Computational Linguistics* 28.3 (2002), s. 245–288. ISSN: 08912017. DOI: 10.1.1.137.1060. arXiv: arXiv:1011.1669v3. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.1060.

[Haj+09] Jan Hajič et al. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In: *Computational Linguistics*. June. Association for Computational Linguistics. 2009, s. 1–18. ISBN: 9781932432299. DOI: 10.3115/1596276.1596305.

[IV98] Nancy Ide a Jean Véronis. Word Sense Disambiguation: The State of the Art. In: *Computational Linguistics* 24 (1998), s. 1–40. ISSN: 03600300. DOI: 10.1145/1459352.1459355.

[Koe05] Philipp Koehn. Europarl : A Parallel Corpus for Statistical Machine Translation. In: *MT Summit* 11 (2005), s. 79–86. DOI: 10.3115/1626355.1626380. URL: http://mt-archive.info/MTS-2005-Koehn.pdf.

[Kon14] Michal Konkol. Brainy: A machine learning library. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Ed. Leszek Rutkowski et al. Sv. 8468 LNAI. Lecture Notes in Computer Science PART 2. Springer International Publishing, 2014, s. 490–499. ISBN: 9783319071756. DOI: 10.1007/978-3-319-07176-3_43.

[KP15] Miloslav Konopik a Ondřej Pražák. Information sources of word semantics methods. Sv. 9319. 2015. ISBN: 9783319231310. DOI: 10.1007/978-3-319-23132-7_30.

[KT13] Mikhail Kozhevnikov a Ivan Titov. Cross-lingual Transfer of Semantic Role Labeling Models. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013, s. 1190–1200. ISBN: 9781937284503. URL: http://www.aclweb.org/anthology/P13-1117.

[KT14] Mikhail Kozhevnikov a Ivan Titov. Cross-lingual Model Transfer Using Feature Representation Projection. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014, s. 579–585. ISBN: 9781937284732.

[LB96] Kevin Lund a Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. In: *Behavior Research Methods, Instruments, & Computers* 28.2 (1996), s. 203–208. ISSN: 0743-3808. DOI: 10.3758/BF03204766.

[LG10] Ding Liu a Daniel Gildea. Semantic role features for machine translation. In: *Coling-2010*. August. Association for Computational Linguistics. 2010, s. 716–724. URL: http://dl.acm.org/citation.cfm?id=1873862.

[LL10] Joel Lang a Mirella Lapata. Unsupervised Induction of Semantic Roles. In: *In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, California, June 2010*. HLT '10 June. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, s. 939–947. ISBN: 1932432655.

[LL11a] Joel Lang a Mirella Lapata. Unsupervised semantic role induction via split-merge clustering. In: *Proceedings of the 49th Annual Meeting of the ...* Association for Computational Linguistics. 2011, s. 1117–1126. ISBN: 9781932432879. URL: http://www.aclweb.org/anthology/P11-1112{\%}5Cnhttp://dl.acm.org/citation.cfm?id=2002614.

[LL11b] Joel Lang a Mirella Lapata. Unsupervised semantic role induction with graph partitioning. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2011), s. 1320–1331. ISSN: 15309312. DOI: 10.1162/COLI_a_00195. URL: http://dl.acm.org/citation.cfm?id=2145432.2145571.

[LPM15] Minh-Thang Luong, Hieu Pham a Christopher D. Manning. Bilingual Word Representations with Monolingual Quality in Mind. In: *Workshop on Vector Modeling for NLP*. 2015, s. 151–159.

[Man+14] Christopher D Manning et al. The Stanford CoreNLP Natural Language Processing Toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2014, s. 55–60. ISBN: 9781941643006. DOI: 10.3115/v1/P14-5010. arXiv: arXiv:1011.1669v3. URL: http://aclweb.org/anthology/P14-5010.

[Mik+13] Tomas Mikolov et al. Efficient Estimation of Word Representations in Vector Space. In: *ArXiv e-prints* abs/1301.3 (2013). arXiv: 1301.3781 [cs.CL]. URL: http://arxiv.org/abs/1301.3781.

[MM08] Marie-Catherine de Marneffe a Christopher D. Manning. The Stanford typed dependencies representation. In: *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*. CrossParser '08 August. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, s. 1–8. ISBN: 978-1-905593-50-7. DOI: 10.3115/1608858.1608859. URL: http://dl.acm.org/citation.cfm?id=1608858.1608859.

[MPH11] Ryan McDonald, Slav Petrov a Keith Hall. Multi-source transfer of delexicalized dependency parsers. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 2007 (2011), s. 62–72. URL: http://www.aclweb.org/anthology/D11-1006.

[Niv+07] Joakim Nivre et al. MaltParser: A language-independent system for data-driven dependency parsing. In: *Natural Language Engineering*. Sv. 13. 2. 2007, s. 95–135. ISBN: 1351324906004. DOI: 10.1017/S1351324906004505.

[ON03] Franz Josef Och a Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. In: *Computational Linguistics* 29.1 (2003), s. 19–51. ISSN: 0891-2017. DOI: 10.1162/089120103321337421. URL: http://dl.acm.org/citation.cfm?id=778822.778824.

[PDM12] Slav Petrov, Dipanjan Das a Ryan McDonald. A Universal Part-of-Speech Tagset. In: *Proceedings of the International Conference on Language Resources and Evaluation*. Istanbul, Turkey: European Language Resources Association (ELRA), 2012. ISBN: 978-2-9517408-7-7. arXiv: 1104.2086.

[PGK05] Martha Palmer, Daniel Gildea a Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. In: *Computational Linguistics* 31.1 (2005), s. 71–106. ISSN: 0891-2017. DOI: 10.1162/0891201053630264.

[PL09] Sebastian Padó a Mirella Lapata. Cross-lingual annotation projection of semantic roles. In: *Journal of Artificial Intelligence Research* 36 (2009), s. 307–340. ISSN: 10769757. DOI: 10.1613/jair.2863.

[PSM14] Jeffrey Pennington, Richard Socher a Christopher D Manning. GloVe: Global Vectors for Word Representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014,

s. 1532–1543. ISBN: 9781937284961. DOI: 10.3115/v1/D14-1162. arXiv: 1504.06654. URL: http://aclweb.org/anthology/D14-1162.

[RP06] Uwe D Reichel a Hartmut R Pfitzinger. Text preprocessing for speech synthesis. In: *Proceedings of the TC-STAR Speech to Speech Translation Workshop* (2006), s. 207–212. URL: http://www.phonetik.uni-muenchen.de/forschung/publikationen/ReichelPfitzingerTCS06.pdf.

[Sch05] Karin Kipper Schuler. VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. In: *Dissertation Abstracts International, B: Sciences and Engineering* 66.6 (2005). ISSN: 04194217.

[SD02] Erik F. Tjong Kim Sang a Fien De Meulder. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* 4 (2002), s. 142–147. DOI: 10.3115/1119176.1119195. arXiv: 0306050 [cs]. URL: http://arxiv.org/abs/cs/0306050.

[SL07] Dan Shen a Mirella Lapata. Using Semantic Roles to Improve Question Answering. In: *Computational Linguistics*. June. 2007, s. 12–21. URL: http://acl.ldc.upenn.edu/D/D07/D07-1002.pdf.

[Str16] Milan Straka. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Paris, France: European Language Resources Association (ELRA), 2016, s. 4290–4297. ISBN: 978-2-9517408-9-1.

[Tie12] J Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In: *Lrec*. Sv. 2012. 2012, s. 2214–2218. ISBN: 978-2-9517408-7-7. URL: http://lrec.elra.info/proceedings/lrec2012/pdf/463{_}Paper.pdf.

[TK12] Ivan Titov a A Klementiev. A Bayesian approach to unsupervised semantic role induction. In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics* c (2012), s. 12–22. URL: http://dl.acm.org/citation.cfm?id=2380821.

[TMU12] Oscar Täckström, Ryan Mcdonald a Jakob Uszkoreit. Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure. In: *2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2012, s. 477–487. ISBN: 978-1-937284-20-6.

[UB08] Jakob Uszkoreit a Thorsten Brants. Distributed Word Clustering for Large Scale Class-Based Language Modeling in Machine Translation. In: *ACL*. 2008, s. 755–762.

[Upa+16] Shyam Upadhyay et al. Cross-lingual Models of Word Embeddings: An Empirical Comparison. In: *Acl 2016* (2016), s. 1661–1670. arXiv: 1604.00425. URL: http://arxiv.org/abs/1604.00425.

[WL15] Kristian Woodsend a Mirella Lapata. Distributed Representations for Unsupervised Semantic Role Labeling. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, {EMNLP} 2015, Lisbon, Portugal, September 17-21, 2015*. 2015, s. 2482–2491. URL: http://aclweb.org/anthology/D/D15/D15-1295.pdf.

[Zem08] Daniel Zeman. Reusable Tagset Conversion Using Tagset Drivers. In: *Lrec*. Marrakech, Morocco: European Language Resources Association (ELRA), 2008, s. 213–218. ISBN: 2-9517408-4-0.

A Uživatelská dokumentace

A.1 Přeložení a spuštění

Aplikace využívá Apache Maven. Má nakonfiguravaný execution plugin, takže ji je možné spustit příkazem:

```
mvn exec:java
```

Jar archiv je možné vytvořit příkazem `mvn install`, ten ale neobsahuje závislosti a jeho spuštění není triviální. Proto doporučuji použití spouštěcího cíle nebo spuštění z některého vývojového prostředí (testováno jen na IntelliJ Idea).

Hlavní aplikací je spuštění experimentu. V *config.properties* se očekává regulární výraz vyhovující spouštěným experimentům (experimentsRegex). Aplikace spustí požadované experimenty (jejich konfigurace se nachází ve složce *Experiments/configs*) a po dokončení uloží jednotlivé výstupy do složky *Experiments/outputs* pod jménem a pořadovým číslem experimentu. Aplikace obsahuje konfigurace všech experimentů prováděných v rámci práce.

Aplikace obsahuje celou řadu dalších spustitelných souborů (různá předzpracování a evaluace), které jsou používány jako jednotlivé kroky experimentů, ale lze je samozřejmě spouštět i samostatně. Každá spustitelná třída má v dokumentaci Javadoc uvedeno, co musí mít nastaveno v *config.properties* pro správný běh. Příklady konfigurace lze nalézt v jednotlivých krocích experimentů.

B Obsah doprovodného DVD

Struktura doprovodného DVD je následující:

- *doc* – Obsahuje text práce a diagramy.
- *app* – Obsahuje zdrojové kódy aplikace a data potřebná k jejímu spuštění.
  - *doc* – Obsahuje Javadoc dokumentaci aplikace.
  - *Experiments* – Obsahuje konfigurace prováděných experimentů.
  - *resources* – Obsahuje data potřebná pro běh aplikace.
  - *config* – Obsahuje konfigurační soubory.
  - *pom.xml* – Maven build skript.