Extrakcia pomenovaných entít z českých textov pomocou zoznamov entít

Alžbeta Strompová, 492945

Projekt na PA026

Rozpoznávanie pomenovaných entít (*angl. Named Entity Recognition, skrátene NER*) je podúlohou extrakcie informácií, ktorá sa snaží nájsť a klasifikovať pomenované entity uvedené v neštruktúrovanom texte do vopred definovaných kategórií, ako sú mená osôb, organizácií, miest, časové výrazy a iné.

V tomto projekte, ktorý bol vytvorený v rámci diplomové práce [10], sa zameriavam na využitie zoznamom pomenovaných entít(angl. *gazetteers*) ako prostriedku na zlepšenie modelov na extrakciu pomenovaných entít pre české texty, bez významného zvýšenia náročnosti tréningu.

1 Riešenia používajúce zoznamy pomenovaných entít

V tejto sekcii v stručnosti opisujem niektoré iné riešenia na rozpoznávanie pomenovaných entít založené na hlbokom učení, ktoré využívajú zoznamy pomenovaných entít.

2 Použitie vlastností zoznamov pomenovaných entít

V práci [9] predstavili spôsob použitia zoznamov pomenovaných entít ako dodatočnej informácie, ktorá vstupuje do BiLSTM.

Testovaná je na anglickom, čínskom a ruskom korpuse, kde ruský korpus predstavuje príklad jazyka s malým množstvom dostupných dát. Zároveň ruština rovnako ako čeština je morfologicky zložitý jazyk, ktorý má rôzne prípony a koncovky. Jednotlivé slová sú hľadané v zoznamoch pomenovaných entít a podľa toho vytvorené vektory reprezentujúce ich zhodu, ktoré sú následne spojené s výstupom z modelu BERT. Spojený vektor ďalej vstupuje do BiLSTM a následne do *Conditional random fields*.

Výsledky tohto prístupu na anglickom a čínskom korpuse ukazujú štatisticky významne zlepšenie, čo predstavuje zlepšenie F1 skóre o $0,52$ pre anglický a $0,3$ pre čínsky korpus. Zatiaľ, čo ruský korpus nepreukazoval štatisticky významné zlepšenie.

2.1 Rozšírenie tréningových dát

Ďalší spôsob na použitie zoznamov pomenovaných entít v práci [9] je generovanie doplnkových tréningových dát. Túto metódu testujem aj na českom korpuse a je bližšie vysvetlená v sekcii 6.2.

Rozšírenie tréningových dát im prinieslo zmiešané výsledky. V počiatočných experimentoch pozorovali zlepšenia vo výkonnosti modelov pri identifikácii osobných mien v ruskom súbore dát. Naproti tomu, použitie modelu BERT pri rovnakom prístupe neukázalo podobné zlepšenie. Medzi niektoré dôvody neúspechu, ktoré spomínajú v práci patrí malý pôvodný súbor dát a nekvalitné zoznamy pomenovaných entít(obsahujúce obskúrne entity). Záverom je, že aj keď rozšírenie tréningových dát pomocou pomenovaných entít nepreukázalo konzistentné zlepšenie, veria, že budúca práca na sofistikovanejšej a kontextovejšej schéme nahradenia bude prínosom pre jazyky s malými zdrojmi dát.

2.2 Fúzia zoznamu pomenovaných entít

V práci [11] testujú spôsoby fúzie zoznamu pomenovaných entít na medicínskej doméne. Vstup rovnako ako v skoršie spomínaných technikách prejde cez enkóder, ale na rozdiel od ostatných riešení pužili enkóder modelu RoBERTa. Predstavili dve verzie fúzie:

- skorú fúziu,
- neskorú fúziu.

Obe verzie spracujú zoznamy pomenovaných entít a vytvoria z nich embedding vektory pre každý vstupný token. Skoršia fúzia spojí výstup z kódovača a spracovania zoznamom pomenovaných entít a na výsledné rozširené vektory aplikuje *tagger*, ktorý pomocou lineárnej vrstvy a funkcie *softmax* priradí najpravdepodobnejší typ entity. *Tagger* je lineárna vrstva s následnou aplikáciou funkcie *softmax* a bližsie vysvetlená je v sekcii 6.1. Neskorá fúzia aplikuje *tagger* jednotlivo na výstup z kódovača a zoznamom pomenovaných entít a kombinuje výsledok z neho. Kombinácia výsledkov prebieha výberom vyššej pravdepodobnosti pre jednotlivé vstupné tokeny.

3 Český korpus pomenovaných entít

Český korpus pomenovaných entít(angl. *Czech Named Entity Corpus* skrátene CNEC) je prvým verejne dostupným korpusom, ktorý poskytuje veľký objem ručne anotovaných pomenovaných entít v českých vetách s detailnou klasifikáciou. Obsahuje 8993 viet, v ktorých je ručne anotovaných 35220 pomenovaných entít v 46 atomických typov entít [12]. Týchto 46 atomických entít sa spája do 8 kategórií, ktorými sú: čísla v adresách, geografické názvy, inštitúcie, mediálne názvy, číselné výrazy, názvy artefaktov, mená osôb a časové výrazy. Väčšina NER modelov identifikuje aspoň tieto kategórie:

- PER - mená osôb,
- ORG - názvy organizácií,
- LOC - geografické názvy.

Preto porovnávanie existujúcich riešení a tréning vlastného modelu som aplikovala na spomínané kategórie, ktoré mapujem na CNEC kategórie nasledovne:

- PER - začínajúce na „p“(nezávisle na veľkosti),
- ORG - začínajúce na „i“,

Tabuľka 1: Rozdelenie českého korpusu pomenovaných entít

	trénovacia časť		validačná časť		testovacia časť		spolu
	počet	percentá	počet	percentá	počet	percentá	počet
vety	7195	79,96%	901	10,01%	902	10,03%	8998
PER	4033	78,48%	542	10,55%	564	10,97%	5139
ORG	2467	81,56%	279	9,22%	427	14,12%	3025
LOC	3409	78,48%	279	6,42%	508	11,70%	4344

- LOC - začínajúce na „g“.

Pri vývoji modelov strojového učenia, ako sú tie na rozpoznávanie pomenovaných entít, je dôležité rozdeliť dostupný súbor dát do troch hlavných častí: trénovacej, validačnej a testovacej. Trénovacia časť je základom pre učenie modelu. Používa sa na prispôsobenie váh modelu, pričom model sa postupne učí identifikovať vzory a charakteristiky, ktoré sú dôležité pre úlohu rozpoznávania entít. Počas trénovania sa model testuje na validačnej sade, aby sa zistilo, ako dobre funguje na dátach, ktoré nevidel počas trénovania. To pomáha určiť, či sa model stáva univerzálnejším a schopným generalizovať na nové dáta. Validačná časť umožňuje monitorovať model, aby sa zabránilo jeho pretrénovaniu. Na základe výsledkov na validačnej sade sa na konci tréningu vyberá najlepšia verzia modelu. Testovacia časť slúži na konečné hodnotenie modelu, aby sa zistilo, ako bude pravdepodobne fungovať v reálnych aplikáciách. Korpus je už rozdelený na tréningovú, validačnú a testovaciu sadu. Rozdelenie zachovávam rovnaké aj z dôvodu, aby existujúce riešenia, v prípade, že boli natrénované na CNEC, nemali pri evaluácií neférovú výhodu. Tabuľka 1 zobrazuje rozloženie viet a jednotlivých entít medzi tréningovú, validačnú a testovaciu časť korpusu.

Korpus je v štyroch rôznych formátoch, ktoré predstavujú rovnaké dáta. Pre spracovanie som vybrala `xml` formát korpusu, z ktorého sa mi najjednoduchšie získavajú potrebné informácie. Jednotlivé sady korpusu majú vlastný súbor s vetami, kde každá je na novom riadku. Začiatok každej entity je označený pomocou `<ne type=“gc”>`, kde znaky medzi úvodzovkami reprezentúju aký typ má daná entita, a koniec entity je označený pomocou `</ne>`. Príklad vety z korpusu:

„Až do porážky <ne type=“gc”>Japonska</ne> ve druhé světové válce nebyl taoismus na <ne type=“gc”>Tchaj-wanu</ne> obnoven.“.

Korpus zahŕňa vnorené entity, čo znamená, že niektoré slová majú pridelených viacero typov entít. Táto situácia sa často vyskytuje, keď je celé meno označené ako typ „meno osoby“ a súčasne sú krstné meno a priezvisko označené oddelene. Napríklad:

„<ne type=“P”><ne type=“pf”>Sigmund</ne> <ne type=“ps”>Freud</ne></ne>.“.

Niektoré prípady vnorenej entity predstavujú úplne odlišné typy. Napríklad, keď názov organizácie obsahuje aj lokáciu, ako je to ukázané tu:

„<ne type=“if”>Mlékárnu <ne type=“gu”>Klatovy</ne></ne>“.

V týchto prípadoch preferujem dlhšiu entitu, a teda vonkajšia entita je správna a kratšiu ignorujem. Takže v predchádzajúcom príklade je označená entita „Mlékárnu Klatovy“ ako ORG a iná entita vo vete nie je.

Existuje viacero spôsobov na označovanie entít, ale zvyčajne sa v úlohách NER používa BIO formát, ktorého ukážka je v tabuľke 2. Pozostáva z troch druhou označenia:

- B - začiatok(angl. *beginning*) entity,
- I - vnútro(angl. *inside*) entity,
- O - mimo(angl. *outside*) entity [5].

Tabuľka 2: Príklad označenia vety

veta	Masarykova	univerzita	se	nachází	v	Brně	.
označenie	B-ORG	I-ORG	O	O	O	B-LOC	O

4 Metriky na porovnanie úspešnosti riešení

Systémy pre rozpoznávanie pomenovaných entít sa hodnotia spustením na anotovaných dátach a porovnávaním ich výsledkov s anotáciami. V tomto projekte využívam knižnicu v pythonu `nervaluate`[1], ktorá definuje štyri rôzne metriky.

- Striktné zhodnotenie (Strict): Vyžaduje presnú zhodu hraníc aj typu entity.
- Exaktné zhodnotenie (Exact): Vyžaduje presnú zhodu hraníc, bez ohľadu na typ entity.
- Čiastočné zhodnotenie (Partial): Umožňuje čiastočné zhody hraníc, bez ohľadu na typ entity.
- Typové zhodnotenie (Type): Vyžaduje, aby sa predikovaná entita aspoň čiastočne prekrývala s anotáciou.

Pri validácii modelu počas tréningu sa používa práve jedna metrika. Z tohto dôvodu a pre prehľadnosť zavediem ešte jednu metriku, ktorá je aritmetickým priemerom spomínaných štyroch metrík, ktoré sme si zadefinovali v tejto sekcii.

$$AvgF1 = \frac{StrictF1 + ExactF1 + PartialF1 + TypeF1}{4}$$

Tabuľka 3: Príklad aplikovania metrík na predikciu

veta	Muni	je	v	Česku	v	městě	Brno	.
ozn.	B-ORG	O	O	B-LOC	O	O	B-LOC	O
pred.	B-LOC	O	B-LOC	I-LOC	O	O	B-PER	O

V tabuľke 3 je zobrazená veta, jej pravdivé označenie entít a umelo vytvorená predikcia. Predikcia našla vo vete tri entity. Hodnoty metrík sú:

$$\begin{aligned} \texttt{Type F1} &= 0,33, \\ \texttt{Partial F1} &= 0,83, \\ \texttt{Strict F1} &= 0, \\ \texttt{Exact F1} &= 0,67, \\ \texttt{Avg F1} &= 0,458. \end{aligned}$$

[1] `https://pypi.org/project/nervaluate/5`

Striktné vyhodnotenie nenašlo žiadnu správne určenú entitu, keďže žiadna nespĺňa správne označenie hraníc a typu. Typové vyhodnotenie nevyžaduje presné hranice, takže považuje entitu „Česku“
za správnu. Zatiaľ čo exaktné vyhodnotenie nevyžaduje správne určený typ, ale hranice entity musia byť presne určené. Takže za správne entity vyhodnotí „Muni“ a „Brno“. Nakoniec čiastočné vyhodnotenie považuje „Muni“ a „Brno“ za správne určené entity a „Česku“ za čiastočne správne určenú entitu.

5 Generovanie zoznamov pomenovaných entít

Súčasťou práce [9] sú aj vygenerované zoznamy pomenovaných entít pre anglický jazyk. Medzi týmito zoznamami sú napríklad názvy štátov, etnických skupín, chorôb, oblečenia, budov, parkov, filmov, povolaní a iné. Väčšina z týchto zoznamov by nebola užitočná pre český text, preto som vybrala iba mená osôb(851 337 entít), názvy organizácii(238 981 entít) a názvy miest v Európe(421 entít).

Dodatočne bolo potrebné obohatiť zoznamy pomenovaných entít ďalšími entitami prevažne z českého prostredia. V nasledujúcich sekciách opisujem získanie českých pomenovaných entít pre jednotlivé kategórie.

5.1 Mená osôb

V českom korpuse pomenovaných entít je pod entitou „mená osôb“ týchto sedem podkategórii:

1. národnosti,
2. akademické tituly,
3. krstné mená,
4. stredné mená,
5. priezviská,
6. mená náboženských alebo mýtických postáv,
7. nešpecifikované.

Dáta som získala z python knižnice *names-dataset*[2], ktorá pozostáva z najpopulárnejších mien rozdelených do 105 krajín. Knižnica obsahuje 18441 českých priezvisk, 5 502 mužských a 5 385 ženských českých krstných mien. Zároveň som rozšírila zoznam pomenovaných entít o zoznam akademických titulov získaný z webových stránok Masarykovej univerzity[3], Karlovej univerzity[4] a českých univerzít[5], čo predstavovalo 27 rôznych titulov. Ďalej som pridala 156 mien viažucich sa k miestu a 41 mien náboženských alebo mýtických postáv pomocou ChatGPT[6].

[2] https://pypi.org/project/names-dataset/
[3] https://www.muni.cz/en/students/addressing-academic-staff
[4] https://lfhk.cuni.cz/Faculty/Organization-structure/Titles-in-Czech/
[5] https://www.czechuniversities.com/article/list-of-academic-titles-and-their-correct-spelling
[6] https://chatgpt.com/

Tabuľka 4: Počet získaných pomenovaných entít pre jednotlivé kategórie

	článok	WikiANN cz	WikiANN sk	ostatné	celkovo
PER	851 109	32 563	8 793	19 605	895 168
ORG	238 976	26 001	6 869	9 975	278 354
LOC	417	29 640	9 205	37 262	68 749

5.2 Názvy organizácií

Zoznam pomenovaných entít názvov organizácii som rozšírila o zoznam najrýchlejšie rastúcich súkromných firiem v Európe[7] a zoznam firiem v Brne[8], čo spoločne predstavuje 10 005 nových entít.

Geografické názvy
Knižnice `CzechData`[9] v jazyku R umožňuje jednoduchší prístup k dátam z registru územnej identifikácie, adries a nehnuteľností Českej rebubliky. Pomocou nej som získala zoznam obcí v Českej republike a ku každej obci jej odpovedajúci kód. Pomocou daného kódu som pre každú obec získala zoznam názvov ulíc a častí obce. Súčasne som pridala zoznam kontinentov a zoznam krajín[10], čo dokopy predstavuje 37 262 entít.

5.3 WikiANN

Na záver som extrahovala pomenované entity z českej časti WikiANN [6], ktorý je viacjazyčný súbor dát zostavený na NER úlohy. Pozostáva z článkov Wikipédie anotovaných označeniami LOC, PER a ORG vo formáte BIO. Obsahuje viac ako sto jazykov vrátene češtiny [8].

Tabuľka 4 zobrazuje, koľko pomenovaných entít bolo celkovo nahromadených pred ich predspracovaním. Pri zjednocovaní jednotlivých zoznamov pomenovaných entít z rôznych zdrojov boli odstránené duplikáty.

6 Implementácia

V tejto časti popisujem základný model založený na kódovači modelu RobeCzech a Small-e-Czech a dve aplikácie zoznamu pomenovaných entít na zlepšenie úspešnosti základného modelu. Pričom jednotlivé spôsoby vyžadujú rozdielne predspracovanie zoznamov pomenovaných entít.

6.1 Základný model

Základný model pozostáva z kódovača modelu RobeCzech, ktorý zo vstupných tokenov vyprodukuje embedding vektory. Embedding vektor je vektorová reprezentácia tokenu zachytávajúca sé-

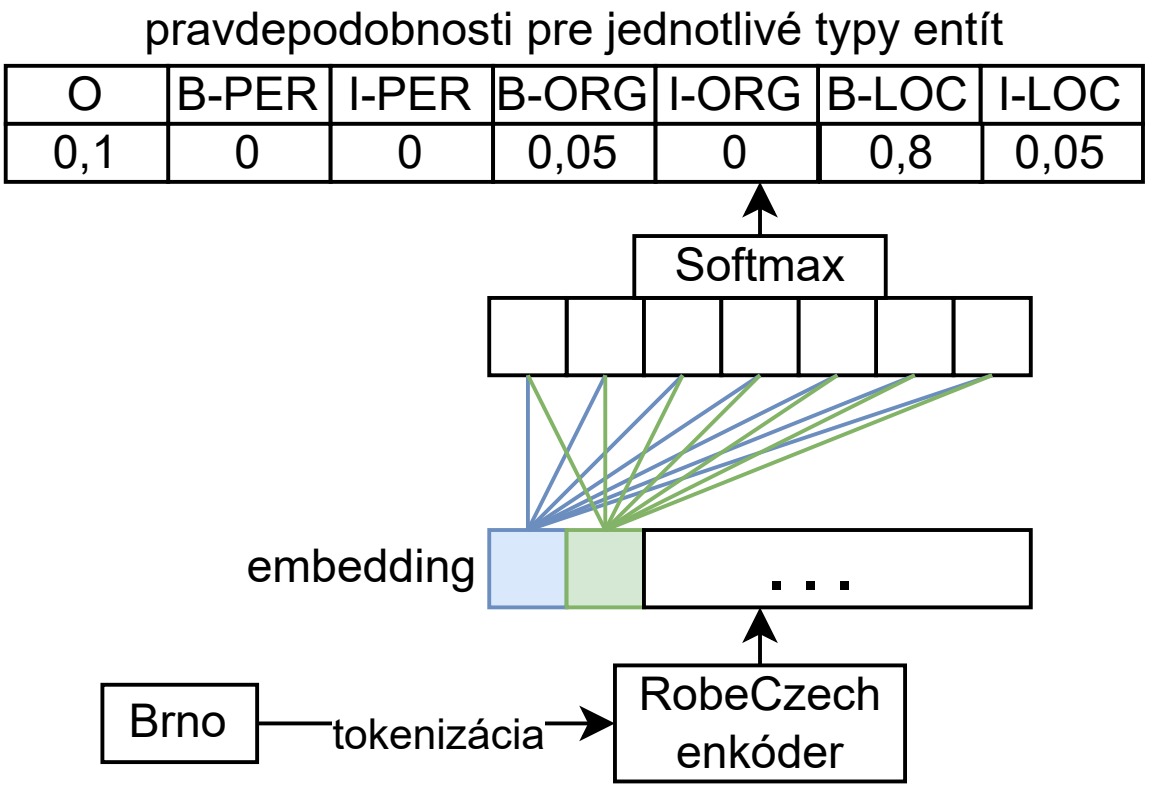
[7] `https://data.world/aurielle/inc-5000-europe-2016`
[8] `https://data.brno.cz/datasets/54e90ebb0fd5463bac48e975b799e583_0/explore`
[9] `https://jancaha.github.io/CzechData/`
[10] `https://www.kaggle.com/datasets/nelgiriyewithana/countries-of-the-world-2023`

mantickú podobnosť medzi slovami [4]. Následne každý vytvorený embedding vektor sa stane jednotlivým vstupom pre lineárnu vrstvu s počtom výstupných neurónov odpovedajúcim počtu rôznych typov entít. Na záver sa na výstup z lineárnej vrstvy aplikuje funkcia `softmax`, ktorá sa často používa vo finálnej vrstve neurónových sietí na klasifikačné úlohy. Funkcia prevádza vektor nespracovaných hodnôt na pravdepodobnosti pre jednotlivé typy entít. Pre vektor $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, kde n je počet typov entít, je funkcia `softmax` definovaná ako:

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad \text{pre } i = 1, 2, \ldots, n$$

kde $\text{softmax}(\mathbf{x})_i$ je pravdepodobnosť i-tého typu a e je základom prirodzeného logaritmu.

Alternatívou je použiť funkciu `argmax`, ktorá vráti pozíciu najvyššej hodnoty. Používa sa vtedy, ak nás nezaujíma konkrétne skóre a ide nám len o najpravdepodobnejší typ entity.



Obr. 1: Príklad fungovania modelu

Tento proces je konkrétnejšie zobrazený na obrázku 1, kde vstupom je slovo „Brno“. Po tokenizácii sa pomocou kódovača vytvorí 768 čísel dlhý embeddinng vektor, ktorý je vstupom do lineárnej vrstvy. Vrstva je plne prepojená. Následne sa na výstupný vektor aplikuje funkcia `softmax`, ktorá vypočíta pravdepodobnosti pre každý typ. Výsledným typom entity sa stáva ten, ktorý

ma najvyššiu pravdepodobnosť. V našom prípade sa ním stáva LOC-B.

6.2 Generovanie rozšírených tréningových dát

Prvou z možností ako využiť zoznamy pomenovaných entít v modeloch NER je zväčšenie korpusu, konkrétne generovanie ďalších tréningových dát. Entity z korpusu anotované človekom nahradíme entitami náhodne vybranými zo zoznamu pomenovaných entít rovnakého typu, ktoré ešte neboli použité. Pri nahradzovaní entít ignorujeme vety, ktoré neobsahujú žiadnu entitu, aby výsledné tréningové dáta neobsahovali rovnakú vetu viac ako raz. V tabuľke 5 je ukážka transformácie vety s dvomi entitami. Dĺžka entít sa nemusí zhodovať, podstatné je len aby pri nahradení bol zachovaný typ entity. S dostatočným množstvom entít v zozname pomenovaných entít vieme vytvoriť aj niekoľkonásobne väčší korpus.

Tabuľka 5: Príklad generovania rozšírených tréningových dát

Slovo	Typ
Vklínila	O
se	O
mezi	O
ně	O
pouze	O
Vondrová	B-PER
z	O
Prahy	B-LOC
.	O

→

Slovo	Typ
Vklínila	O
se	O
mezi	O
ně	O
pouze	O
Mgr.	B-PER
Miloslava	I-PER
Machová	I-PER
z	O
Brno	B-LOC
.	O

Skloňovanie pomenovaných entít
Jedným z problémov tohto nahradzovania je skloňovanie. Nová entita nemusí zdielať so starou rovnaký slovesný pád, čo zhoršuje čitateľnosť vety. Riešením je pomocou nástroja `MorphoDiTa`[11] zistiť slovesný pád a číslo nahradzovanej aj novej entity. Nástroj vyprodukuje informácie o každom slove zvlášť. Keďže entity môžu byť rôzne dlhé, tak do úvahy beriem posledné podstatné meno v entite a jeho slovesný pád a číslo. Následne pomocou nástroja `declension`[12] upravím novú entitu podľa vlastností nahradzovanej entity. Napríklad, ak je nahradzovaná entita „Prahy“, tak `MorphoDiTa` určí 2. slovesný pád a jednotné číslo. Novej entite „Brno“ určí 1. slovesný pád a jednotné číslo. Tieto informácie s novou entitou nástroj `declension` spracuje a vráti novú entitu v požadovanom tvare, a to „Brna“. Toto spracovanie entít síce zlepšuje zapojenie pomenovaných entít do tréningových dát, ale výrazne zvyšuje čas na vytvorenie rozšírených tréningových dát. Napriek spracovaniu jednotlivých viet paralelizovane na 32 jadrách, tak čas vytvorenia tréningových dát sa zvýšil z približne troch minút(bez paralelizácie) na viac ako hodinu a pol s aplikáciou skloňovania.

[11] `https://corpy.readthedocs.io/en/stable/guides/morphodita.html`
[12] `https://nlp.fi.muni.cz/projekty/declension/index.py`

Tabuľka 6: Porovnanie stemmingu a lemmatizácie

pôvodné slovo	stemming	lemmatizácia
Václavem	václavem	Václav
Brně	brně	Brno
univerzitou	univerzit	univerzita

Predspracovanie zoznamu pomenovaných entít
Predspracovanie zoznamu pomenovaných entít zahŕňa niekoľko dôležitých krokov, ktoré zabezpečia, že dáta budú vhodné pre použitie v NER modeloch. Na začiatok pre každý typ sa vytvorí množina obsahujúca entity, pre rýchlejšie vyhľadávanie a odstránenie nepotrebných duplikátov. Ďalším krokom je odstránenie čisto numerických entít a entít kratších ako tri znaky. Entity získane z WikiANN často obsahujú v zátvorkách bližší popis entity. Napríklad: „Václav II. (1271–1305)“ alebo „Národní osvobozenecká armáda (Makedonie)“. Tieto informácie majú často iný typ alebo sú zavádzajúce. Z tohto dôvodu ich počas predspracovania odstraňujem.

Zároveň je nutné pôvodne získané entity typu `PER` spojiť. Počas generovania zoznamov pomenovaných entít sme získali zoznamy obsahujúce tituly, krstné mená a priezviská. Keď nahrádzame entitu, ktorá reprezentuje osobu, tak ju nechceme vymeniť len za titul ale za celé meno. Preto je nutné vytvoriť komplexnejšie mená osôb, aby to odpovedalo skutočným dátam.

6.3 Pridanie dodatočných informácii k výstupu z kódovača

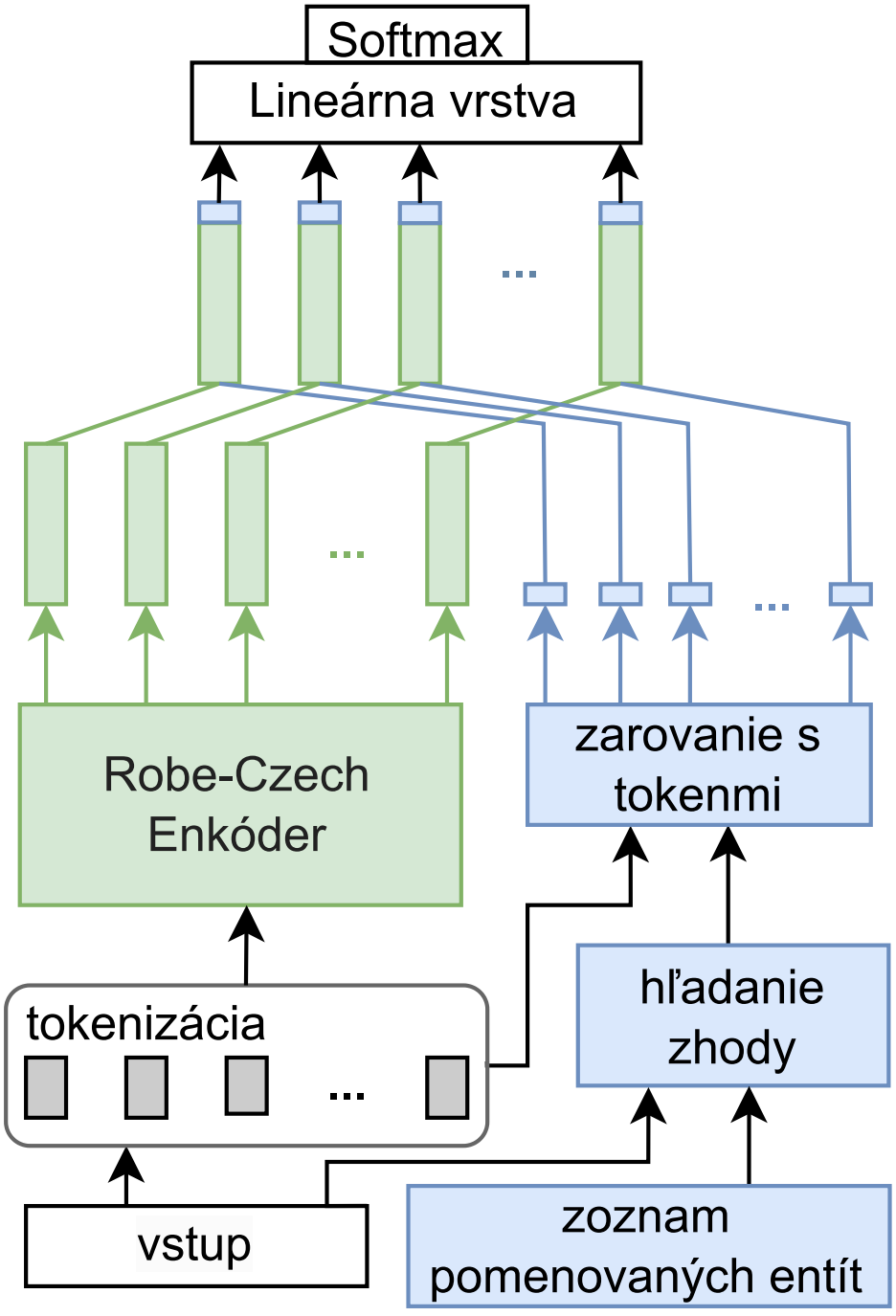
Ďalším spôsob na zapojenie zoznamu pomenovaných entít do NER modelov je rozšírenie embedding vektorov. Tento prístup vznikol upravením a kombináciou iných prístupov [9, 11]. Na obrázku 2 je zobrazená architektúra úpraveného modelu. Na vstup sa aplikuje tokenizácia a výsledné tokeny RobeCzech kódovač transformuje na embedding vektory, tak isto ako v základnom modeli.

Predspracovanie zoznamy pomenovaných entít
Pred hľadaním zhody je nutné pripraviť zoznamy pomenovaných entít pre zjednodušenie a vyššiu úspešnosť hľadania zhody. Fúzne jazyky (jazyky, v ktorých sa slová skloňujú) sú obzvlášť zložité pre NER, pretože jedna entita sa v texte môže vyskytovať v rôznych formách. Na ich prekonanie môžu systémy NER použiť dve lingvistické techniky na konverziu slov do ich základných foriem:

- Stemming je jednoduchá, ale agresívna technika, ktorá zahŕňa redukciu slova na kmeň slova odstránením jeho prípon alebo predpôn.
- Lemmatizácia produkuje slovné jednotky, ktoré sú stále platnými jazykovými formami [2].

Pomocou triedy `PorterStemmer`[13] z python knižnice `nltk` a python knižnice `simplemma`[14] som vyskúšala obe metódy na zopár náhodne vybraných slovách z korpusu. V tabuľke 6 vidno, že lepšie výsledky dosahuje lemmatizácia. Pre zlepšenie hľadania zhody pridávam možnosť aplikovať na všetky entity v zozname pomenovaných entít lemmatizáciu. Počas porovnávania lemmatizujem aj slovo zo vstupu.

[13] `https://www.nltk.org/_modules/nltk/stem/porter.html`
[14] `https://pypi.org/project/simplemma/`



Obr. 2: Architektúra modelu používajúceho rozšírené embedding vektory

Hľadanie zhody
Na rozdiel od základného modelu vstupný text prejde procesom hľadania zhody v zozname pomenovaných entít, ktorý je zobrazený na obrázku 2 zelenou farbou. Vo všeobecnosti existujú dve metódy hľadania zhody, a to jednoslovná a viacslovná.

- Jednoslovná zhoda je vyhľadávanie, aby sa slovo zo vstupu zhodovalo s ľubovoľným slovom zo zoznamu pomenovaných entít. Pri tejto technike hľadania zhody je nutné všetky entity v zoznamoch pomenovaných entít rozdeliť na jednotlivé slová pred predspracovaním.
- Viacslovná zhoda je hľadanie najdlhšieho segmentu vety, ktorý sa nachádza v zozname pomenovaných entít. Okrem typu PER, kde sa naďalej aplikuje jednoslovná zhoda. Je to z dôvodu, že v texte zvyknú stáť samostatne krstné mena aj priezviska bez prítomnosti druhého. Na rozdiel od názvov organizácii, ktoré sú často nerozdeliteľné a nedávajú samostatne zmysel, ako napríklad: „Strana zelených“.

Napríklad, ak máme vetu „Moravské muzeum se nachází v Brně.“ a zoznamy pomenovaných entít vyzerajú takto:

```
{
  "LOC": ["Brno", "muzeum hudby"],
}.
```

Jednoslovná zhoda identifikuje iba slovo „Brně“ za predpokladu, že je použitá lemmatizácia. Viacslovná zhoda navyše identifikuje „Moravské“ a „muzeum“ ako LOC. Konkrétne v tomto príklade vyzerá, že jednoslovná zhoda je lepšia, ale keby sme mali vetu „Mám ráda poslech hudby.“, tak by nesprávne označila slovo „hudby“ ako LOC.

Spojenie do rozšíreného embedding vektoru
Výstupom z hľadania zhody je trojprvkový vektor pre každé slovo zo vstupu. Pred spojením s embedding vektormi z kódovača modelu RobeCzech je potrebné zarovnať vektory s tokenmi. Potom nasleduje spojenie embedding vektorov a vektorov obsahujúcimi informácie o zozname pomenovaných entít do rozšíreného 771 prvkového embedding vektoru, ktorý pokračuje ako vstup do lineárnej vrstvy rovnako ako v základnom modeli.

7 Experimenty

V tejto časti prezentujem výsledky evaluácie modelov so zaokrúhlením na tri desatinné miesta. Najprv porovnávam existujúce riešenia a následne porovnávam mnou implementované riešenia. Všetky mnou trénované modely majú rovnaké parametre použité pri tréningu.

7.1 Porovnanie existujúcich NER systémoch na českom jazyku

Porovnávam rôzne nájdené modely na testovacej časti CNEC. Pri hľadaní modelov som sa obmedzovala hlavne na novšie technológie. Narazila som len na malé množstvo českých modelov a tie sú väčšinou zároveň viacjazyčné. Aj z tohto dôvodu som vyskúšala aj rôzne iné viacjazyčné modely, ktoré neboli konkrétne určené na češtinu. Zároveň som vyskúšala aj *GPT3.5*.

xlm-roberta-large-finetuned-conll03-english
xlm-roberta-large-finetuned-conll03-english[15]je predstavená v [1]. Tento viacjazyčný model je založený na modely RoBERTa a následne trénovaný na korpuse `conll2003`[16].

WikiNEuRal
WikiNEuRal[17] je viacjazyčný model, ktorý je založený na viacjazyčnej verzii modelu BERT natrénovanovanej na korpuse *WikiNEuRal*[18] na úlohu rozpoznávania pomenovaných entít. Korpus obsahuje deväť jazykov, medzi ktorými nie je český jazyk. Napriek tomu dosahuje dobré výsledky naprieč jazykmi. Namiesto toho, aby reprezentovali slovo prvou kontextualizovanou reprezentáciou podslov, ako to poskytuje viacjazyčný BERT, berú strednú hodnotu jeho podslov. Výsledné vektory prechádzajú cez viacvrstvovú sieť BiLSTM na úrovni viet, ktorej výstup sa potom pošle do modelu CRF[13].

CNEC_xlm-roberta-large
Radek Štulc vytvoril český model *CNEC_xlm-roberta-large*[19], ktorý je založený na modeli *xlm-roberta-large*, ktorý vyvinula spoločnosť Facebook. Tento model je následne špecificky trénovaný na českom korpuse CNEC, čo mu umožňuje efektívne rozpoznávať pomenované entity v češtine. Model *xlm-roberta-large* je variantou modelu RoBERTa, ktorý je optimalizovaný pre širokú škálu jazykov.

Tento model je trénovaný na rozpoznávanie ôsmich rôznych typov entít, ktoré sú v CNEC. Na stránke modelu je zverejnené F1 skóre `0,88` F1 na validačnej časti dát CNEC, ktoré odpovedá metrike striktné F1 skóre popísane v tejto práci. Výsledky sa líšia od prezentovaných výsledkov v tejto kapitole kvôli evaluácii modelu len na troch typoch entít na testovacej časti dát.

SlovakBert-ner
Slovenský a český jazyk sú veľmi podobné a preto som zahrnula slovenský model[20] od autora Ivana Agarského na rozpoznávanie pomenovaných entít medzi porovnávane modely. Založený je na slovenskej verzie modelu BERT a trénovaný na súbore dát z WikiANN, kde dosahuje F1 skóre až `0,94`.

GPT-3.5 Turbo
GPT-3.5 Turbo je variantom modelu *GPT-3* od OpenAI[21]. Tento model som testovala pomocou API volania. Testovala som rôzne prompty(vstupy do modelu) v českom aj anglickom jazyku. Český prompt dosahoval lepšej úspešnosti pravdepodobne z dôvodu, že veta, z ktorej chceme extrahovať pomenované entity, je v českom jazyku. Systémový prompt, ktorý sa nemení v priebehu testovania, je „Jsi český model rozpoznávání pojmenovaných entit. Tvým úkolem je extrahovat pojmenované entity z poskytnuté věty a vrátit je ve formátu JSON. Měli byste identifikovat a vrátit následující typy entit: lokace, osoba a organizace. U každé entity určete její typ a rozsah textu. Ujistěte se, že výstupem je správně vytvořený objekt JSON.“. Prompt, ktorý je iný pre každú testovaciu vetu, je „Věta: “ a veta, ktorú som chcela spracovať.

[15] `https://huggingface.co/FacebookAI/xlm-roberta-large-finetuned-conll03-english`
[16] `https://huggingface.co/datasets/conll2003`
[17] `https://huggingface.co/Babelscape/wikineural-multilingual-ner`
[18] `https://huggingface.co/datasets/Babelscape/wikineural`
[19] `https://huggingface.co/stulcrad/CNEC_extended_xlm-roberta-large`
[20] `https://huggingface.co/crabz/slovakbert-ner`
[21] `https://openai.com/`

Výsledky
Spomínané modely sú porovnané v tabuľke 7 a zvýraznený je najlepší výsledok v jednotlivých stĺpcoch. Najlepším nájdeným modelom je *CNEC_xlm-roberta-large*.

Tabuľka 7: Porovnanie hodnôt F1 skóre existujúcich riešení

Model	Type	Partial	Exact	Strict	Avg
xlm-roberta-large	0,811	0,858	0,819	0,749	0,809
WikiNEuRal	0,789	0,843	0,812	0,742	0,796
CNEC_xlm-roberta-large	**0,889**	**0,922**	**0,898**	**0,846**	**0,889**
SlovakBert-ner	0,644	0,683	0,632	0,741	0,632
GPT-3.5	0,57	0,58	0,525	0,485	0,54

Porovnávala som aj ďalšie modely, medzi ktoré napríklad patrí *SpanMarkerNER*[22], ale žiaden iný český model nezískal priemerné F1 skóre väčšiu ako `0,55`. Pomedzi viacjazyčných modelov dosahoval ešte jeden model hodnotu priemerného F1 skóre vyššiu ako `0,65`, konkrétne anglicko-španielsky model[23] s hodnotou `0,693`.

7.2 Výber hyperparametrov modelu

Hyperparametre sú nastavenia, ktoré ovplyvňujú proces učenia modelu. Medzi ne patrí aj optimalizačný algoritmus, ktorý je používaný na úpravu váh modelu počas tréningového procesu tak, aby sa minimalizovala chybová funkcia. Pri tréningu je použitý *AdamW*, ktorý je vylepšenou verziou optimalizačného algoritmu *Adam*. Koriguje implementovanú regularizáciu váh v pôvodnom algoritme *Adam*. Tradičný *Adam* a mnoho iných optimalizačných algoritmov pridávajú penalizáciu váh priamo do gradientov, čo môže viesť k suboptimálnemu správaniu. *AdamW* tento problém rieši tým, že oddeluje krok regularizácie váh od kroku optimalizácie, čo znamená, že regularizácia váh je aplikovaná separátne, čo umožňuje presnejšie a efektívnejšie prispôsobenie váh modelu bez ovplyvnenia samotného procesu učenia. Tento prístup vedie k lepšej generalizácii a stabilnejšiemu tréningu modelov [3].

Pri výbere hyperparametrov som sa zamerala na tri parametre. Ostané sú ponechané na predvolených hodnotách.

- *Learning rate* je parameter, ktorý určuje veľkosť krokov, ktoré sa vykonávajú pri aktualizácii váh modelu počas tréningu. Vyjadruje, ako rýchlo alebo pomaly sa model učí z tréningových dát a priamo ovplyvňuje konvergenciu algoritmu. Vysoká umožňuje rýchlu konvergenciu, ale môže spôsobiť, že model preskočí optimálne riešenie. Naopak nízka vedie k pomalšej konvergencii, čo môže byť užitočné na dosiahnutie presnejšieho minima, ale zvyšuje čas potrebný na tréning. Zároveň môže spôsobiť, že sa model zasekne v miestnych minimách a nedosiahne globálne optimum.

- *Batch size* je počet tréningových príkladov použitých v jednom tréningovom kroku [7]. Pri trénovaní je nastavený na hodnotu jedna a namiesto toho som upravovala parameter *gradient accumulation steps*, ktorý umožňuje efektívne využitie väčšieho *batch size* bez potreby

[22] `https://github.com/tomaarsen/SpanMarkerNER`
[23] `https://github.com/sagorbrur/codeswitch`

zvýšenia pamäťových nárokov na GPU. *Gradient accumulation steps* je parameter, ktorý určuje počet krokov, počas ktorých sa bude gradient hromadiť predtým, než sa aktualizujú váhy modelu. Pri nastavení *batch size* na 1 a zvýšení hodnoty *gradient accumulation steps* napríklad na 16, sa dosiahne *batch size* 16. Tento prístup umožňuje simulovať väčší *batch size* bez nutnosti mať dostatočne veľkú pamäť na GPU.

- *Weight decay* je regularizačný parameter, ktorý pomáha predchádzať pretrénovaniu modelu počas tréningu. Ide o techniku, ktorá pridáva penalizáciu za veľkosť váh modelu, čím povzbudzuje menšie váhy a zlepšuje jeho generalizačné schopnosti.

Následne som aplikovala rôzne kombinácie týchto hyperparametrov na trénovanie základného modelu a vyhodnotila výsledky. z vyskúšaních hodnôt sa prejavila najlepšie zvýraznená kombinácia.

```
{
  "learning rate": [1 × 10^−4, 5 × 10^−5,  1 × 10^−5],
  "batch size": [8, 16, 32, 64],
  "weight decay": [0,01, 0,005, 0,001],
}
```

Ďalej som zvolila ako metriku na porovnávanie modelov priemernú hodnotu F1 skóre definovanú v sekcii 4 a nastavila ukladanie najlepšieho modelu na záver tréningu. Ostatné parametre som nechala nastavené na ich predvolené hodnoty.

7.3 Porovnanie modelov trénovaných na rozšírených trénovaniach dátach

Zväčšenie trénovacieho korpusu som aplikovala na NER modeloch založených na modeloch Small-e-czech a RobeCzech. Pričom som menila parameter uvádzajú koľko iterácií zväčšenia korpusu je prevedených. Nula reprezentuje základný model, jedna reprezentuje model trénovaný okrem základného korpusu aj na kópii so zmenenými dátami. Pri vytváraní kópie sa ignorujú vety bez pomenovaných entít, aby trénovacia časť dát neobsahovala rovnaké vety. Pri modely *RobeCzech* som aplikovala aj možnosť skloňovania pomenovaných entít. Výsledky je možno vidieť v tabuľke 8. Rozšírenie tréningových dát nepreukázalo zlepšenie modelov a mierne zvýšilo čas tréningu oproti základnému modelu. Ani pridanie skloňovania neprinieslo zlepšenie, ale výrazne zvýšilo čas na predspracovanie dát.

Vyskúšala som modely aj na súbore dát WikiANN. Generované entity z WikiANN zapojené do rozširovania tréningových dát boli súčasťou trénovacej časti. Takže evaluácia na testovacej časti nemala nespravodlivú výhodu. Predpokladala som, že model trénovaný na rozšírených trénovacích dátach bude mať výrazne lepšie výsledky oproti základnému modelu, keďže v procese trénovania sa už stretol s podobnými entitami. Napriek tomu zlepšenie bolo nevýrazné.

7.4 Porovnanie modelov s rozšírenými embedding vektormi

Model založený na RobeCzech vyšiel diametrálne lepšie, tak rozšírené embedding vektory aplikujem len na neho. Pričom mením techniku na hľadanie zhody a aplikáciu lemmatizácie. Zároveň

Tabuľka 8: Porovnanie hodnôt F1 skóre modelov trénovaných na rozšírených trénovaniach dátach

Model	poč.	skl.	Type	Partial	Exact	Strict	Avg
Small-e-czech	0	—	**0,862**	**0,864**	**0,825**	**0,804**	**0,839**
Small-e-czech	1	nie	0,856	0,861	0,82	0,796	0,833
Small-e-czech	3	nie	0,854	0,858	0,82	0,8	0,833
RobeCzech	0	—	0,928	0,933	**0,913**	**0,899**	**0,918**
RobeCzech	1	nie	0,928	**0,934**	0,912	0,892	**0,918**
RobeCzech	3	nie	0,926	0,93	0,908	0,892	0,914
RobeCzech	1	áno	**0,931**	0,931	0,906	0,893	0,916
RobeCzech	3	áno	0,925	0,928	0,901	0,881	0,909

pridávam možnosť rozšírenia zoznamov pomenovaných entít o dáta získané z českého a slovesnkého WikiANN. Pridanie dodatočnej informácie k embedding vektorom nemalo výrazný efekt na čas potrebný na predspracovanie a tréning.

Tabuľka 9 zobrazuje porovnanie modelov s použitím jednoslovnej zhody a tabuľka 10 zobrazuje porovnanie modelov s použitím viacslovnej zhody. Použitie lemmatizácie navzdory predpokladom nezlepšilo výrazne model. Naopak rozšírenie zoznamom pomenovaných entít malo priaznivé účinky pri aplikácií na nelemmatizovanú aplikáciu jednoslovnej zhody. Najlepším modelom, ktorý dosiahol najvyššiu hodnotu vo všetkých metrikách naprieč týmito dvomi tabuľkami, je nelemmatizovaná aplikácia jednoslovnej zhody pomocou zoznamov pomenovaných entít z českého aj slovenského súboru dát WikiANN.

Tabuľka 9: Porovnanie modelov s rozšírenými embedding vektormi pri aplikácií jednoslovnej zhody

lem	cz	sk	Type	Partial	Exact	Strict	Avg
nie	nie	nie	0,94	0,936	0,914	0,905	0,924
nie	áno	nie	0,938	0,938	**0,918**	0,907	0,925
nie	áno	áno	**0,941**	**0,94**	**0,918**	**0,909**	**0,927**
áno	nie	nie	0,938	0,934	0,912	0,901	0,921
áno	áno	nie	0,933	0,934	0,912	0,899	0,919
áno	áno	áno	0,932	0,933	0,91	0,896	0,919

Tabuľka 10: Porovnanie modelov s rozšírenými embedding vektormi pri aplikácií viacslovnej zhody

lem	cz	sk	Type	Partial	Exact	Strict	Avg
nie	nie	nie	0,927	0,935	**0,915**	0,899	0,919
nie	áno	nie	0,935	0,934	0,911	0,9	0,919
nie	áno	áno	0,932	**0,936**	**0,915**	**0,902**	0,921
áno	nie	nie	**0,938**	**0,936**	0,913	**0,902**	**0,922**
áno	áno	nie	0,935	0,934	0,911	0,9	0,92
áno	áno	áno	0,932	**0,936**	**0,915**	**0,902**	0,921

Tabuľka 11: Porovnanie základného modelu a modelu s rozšírenými embedding vektormi na konktrétnom príklade

	Peter	Greenpeace	Amazonka
anotácia	PER	ORG	LOC
základný model	PER	LOC	LOC
rozšírené embedding vektory	PER	ORG	LOC

Základný model aj model s rozšírenými embedding vektormi našli vo vete „Peter z Greenpeace pracoval na projekte na ochranu rieky Amazonka.“ tri entity. Pri hľadaní zhody so zoznamom pomenovaných entitít algoritmus našiel tieto zhody na daných entitách:

- PER: „Peter“,
- ORG: „Peter“, „Greenpeace“,
- LOC: „Amazonka“.

Výsledky oboch modelov, zobrazené v tabuľke 11 naznačujú, že zoznamy pomenovaných entít prispeli k správnemu určeniu typu entity „Greenpeace“.

7.5 Porovnanie kombinovaného modelu

Na záver som vyskúšala, či bude mať pozitívny efekt rozšírenie tréningových dát na modely s rozšírenými embedding vektormi. Porovnanie najlepšieho modelu s rozšírenými embedding vektormi a daný model s rozšírenou trénovacou časťou korpusu je v tabuľke 12. Experiment neukázal zlepšenie pri kombinácií prístupov.

Tabuľka 12: Porovnanie kombinovaného modelu

rozšírené tréningové dáta	Type	Partial	Exact	Strict	Avg
nie	**0,941**	**0,94**	**0,918**	**0,909**	**0,927**
áno	0,943	0,939	0,914	0,903	0,926

8 Spustenie

V rámci tohto projektu bola vyvinutá webová aplikácia[24], ktorej primárnym účelom je demonštrovať a testovať najlepšie nájdené riešenie na rozpoznávanie pomenovaných entít. Aplikácia poskytuje intuitívne rozhranie pre užívateľov, umožňujúce im overiť efektivitu a presnosť modelu na rozpoznávanie pomenovaných entít v reálnom čase.

Nahrala som model na `HuggingFace`[25] a pomocou `Hugging Face Spaces`[26] nasadila model na vyskúšanie. Pre vizuálnu komponentu je použitá python knižnica `Gradio`[27].

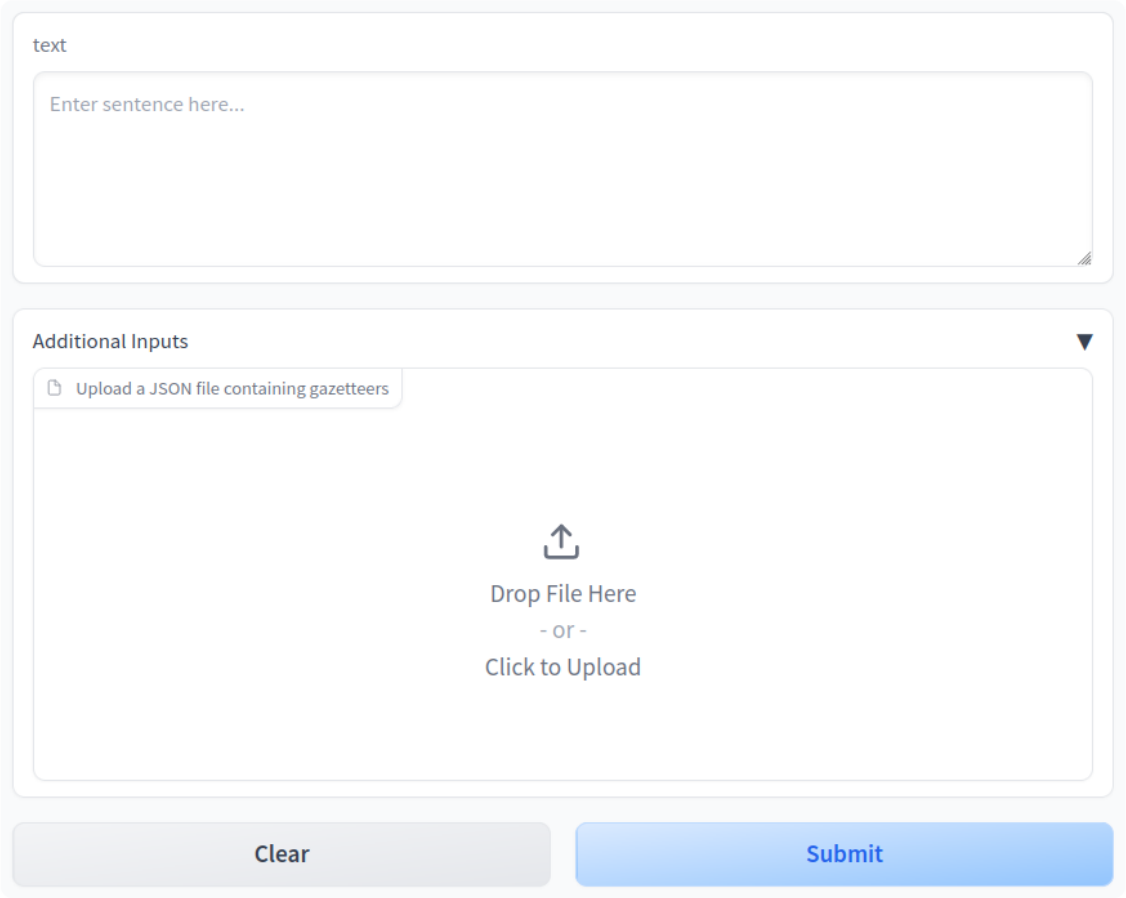
[24] `https://nlp.fi.muni.cz/projekty/gazetteer_ner/index.html`
[25] `https://huggingface.co/bettystr/NerRoB-czech`
[26] `https://huggingface.co/docs/hub/spaces`
[27] `https://www.gradio.app/docs`

V aplikácií je možné okrem zadania českého textu aj nahrať zoznamy vlastných pomenovaných entít. Možné je nahrať viacero súborov vo formáte `json`. Príklad obsahu súboru:

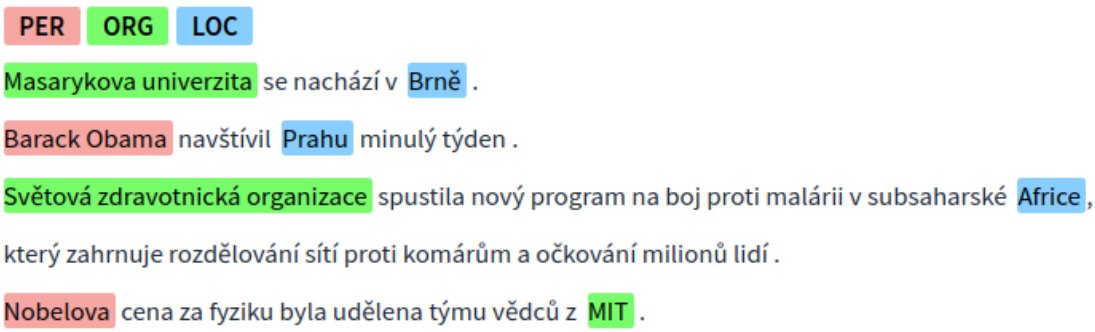
```
{
  "PER": ["John Doe"],
  "ORG": ["Masarykova univerzita"],
  "LOC": ["Brno", "Praha"],
}.
```

Riešenie je možné vyskúskúšať na spomínanej webovej stránke a príípadne v priloženom súbore s implementáciou je možne pomocou `train_scipt.py` natrénovať vlastný model. Informácie su v súbore `README.md`.

Ukážka webovej stránky:



Obr. 3: Ukážka vstupu webovej stránky



Obr. 4: Ukážka výstupu webovej stránky

Referencie

[1] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: 1911.02116 [cs.CL].

[2] Divya Khyani et al. „An interpretation of lemmatization and stemming in natural language processing“. In: *Journal of University of Shanghai for Science and Technology* 22.10 (2021), s. 350–357.

[3] Ilya Loshchilov a Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[4] Bernardo Magnini, Alberto Lavelli a Simone Magnolini. „Comparing Machine Learning and Deep Learning Approaches on NLP Tasks for the Italian Language“. English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, máj 2020, s. 2110–2119. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.259.

[5] Muskaan Maurya. *Name entity recognition and various tagging schemes*. Feb. 2023. URL: https://medium.com/@muskaan.maurya06/name-entity-recognition-and-various-tagging-schemes-533f2ac99f52 (cit. 20. 05. 2024).

[6] Xiaoman Pan et al. „Cross-lingual Name Tagging and Linking for 282 Languages“. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. Regina Barzilay a Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, júl 2017, s. 1946–1958. DOI: 10.18653/v1/P17-1178. URL: https://aclanthology.org/P17-1178.

[7] Martin Popel a Ondřej Bojar. „Training Tips for the Transformer Model“. In: *The Prague Bulletin of Mathematical Linguistics* 110.1 (apr. 2018), s. 43–70. ISSN: 1804-0462. DOI: 10.2478/pralin-2018-0002. URL: http://dx.doi.org/10.2478/pralin-2018-0002.

[8] Afshin Rahimi, Yuan Li a Trevor Cohn. „Massively Multilingual Transfer for NER“. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, júl 2019, s. 151–164. URL: https://www.aclweb.org/anthology/P19-1015.

[9] Chan Hee Song et al. *Improving Neural Named Entity Recognition with Gazetteers*. 2020. arXiv: 2003.03072 [cs.CL].

[10] Alžbeta Strompová. „Extrakcia pomenovaných entít z českých textov pomocou zoznamov entít [online]“. SUPERVISOR : Aleš Horák. Diplomová práce. Masarykova univerzita, Fakulta informatiky, Brno, [cit. 2024-05-27]. URL: https://is.muni.cz/th/f7viq/.

[11] Qing Sun a Parminder Bhatia. „Neural Entity Recognition with Gazetteer based Fusion“. In: (2021). arXiv: 2105.13225 [cs.CL].

[12] Magda Ševčíková, Zdeněk Žabokrtský a Oldřich Krůza. „Named Entities in Czech: Annotating Data and Developing NE Tagger“. In: *Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue*. Ed. Václav Matoušek a Pavel Mautner. Zv. 4629. Lecture Notes in Computer Science XVII. Berlin / Heidelberg: Springer, 2007, s. 188–195. ISBN: 978-3-540-74627-0.

[13] Simone Tedeschi et al. „WikiNEuRal: Combined Neural and Knowledge-based Silver Data Creation for Multilingual NER“. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. Marie-Francine Moens et al. Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. 2021, s. 2521–2533. DOI: 10.18653/v1/2021.findings-emnlp.215. URL: https://aclanthology.org/2021.findings-emnlp.215.