

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Nástroje pro přípravu českého textu na strojové  
zpracování**  
Bakalářská práce

Autor: Vlasta Matějková  
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Ph.D., Jiří Haviger

Hradec Králové

květen 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 1.5.2024

Vlasta Matějková

## Poděkování

Ráda bych poděkovala Mgr. Ph.D. Jiřímu Havigerovi za odborné vedení mé práce, trpělivost, ochotu a cenné rady, které mi umožnily práci dokončit.

## **Abstrakt**

Práce vysvětluje principy přípravy textu pro strojové zpracování. Mezi hlavními principy patří normalizace, segmentace, tokenizace, lemmatizace, stematizace, desambiguace, part of speech tagování a další. Tyto principy fungují obecně na většinu jazyků, pro správnou a detailní přípravu textu je ovšem třeba postupy upravit na základě požadovaného jazyka (v tomto případě češtinu) a budoucímu účelu. Práce uvádí řešení problematiky z hlediska filologického a programátorského. Filologický přístup zdůrazňuje důležitost porozumění specifik jazyka, gramatickým prvkům a kontextu. Programátorský přístup pak představuje vyzkoušení veřejně dostupných nástrojů, které implementují tyto principy. Práce pojednává o výhodách a omezeních používání veřejně dostupných nástrojů. Jedním z faktorů je například přesnost, chybovost, rychlost, adaptace, zachování významu v konkrétních scénářích (dlouhé texty, komplikovaná souvětí, ...).

## **Klíčová slova**

NLP, tokenizace, lemmatizace, POS tagování, desambiguace

## **Abstract**

### **Title: HTools for preprocessing Czech text for machine processing**

The thesis explains principals of text preparation for machine processing. The main principals include normalization, segmentation, tokenization, lemmatization, stemmatization, disambiguation, part of speech tagging and more. These principals work in general for most languages, but for correct and detailed preparation of text, the procedures must be adjusted based on the wanted language (in this case Czech) and future purpose. The thesis presents a solution to the problem from a philological and programming point of view. The philological approach emphasizes the importance of understanding the specifics of language, grammatical elements and context. The programming approach the represents the testing of publicly available tools that implement these principles. The thesis discusses the advantages and limitations of using these tools. One of the factors is, for example, accuracy, error

rate, speed, customization, keeping the meaning in specific scenarios (long texts, complicated sentences, ...).

## **Keywords**

NLP, tokenization, lemmatization, POS tagging, disambiguation

## Obsah

1	Úvod.....	8
2	Cíl práce.....	9
3	Specifika českého textu vzhledem k angličtině, tokenizace, lemmatizace, desambiguace a part-of-speech tagování.....	9
3.1	Jednotlivé kroky zpracování.....	10
3.2	Zpracování pomocí LLM.....	16
3.3	Přehled dostupných nástrojů.....	17
4	Metodika zpracování.....	18
4.1	Analyzované texty .....	18
4.2	Metriky.....	19
4.3	Používané nástroje.....	21
5	Výsledky testování na reálných textech.....	22
5.1	Stop slova.....	22
5.2	Tokenizace a segmentace .....	25
5.3	Lemmatizace.....	26
5.4	Part of speech tagování .....	29
5.5	Named entity recognition.....	31
5.6	Další způsob přípravy textu.....	32
6	Shrnutí a diskuse výsledků.....	33
7	Závěry a doporučení .....	35
8	Seznam použité literatury.....	36
9	Zadání práce.....	38

## Seznam obrázků

Obrázek 1 Vizualní reprezentace nejčastějších slov s pomocí wordcloudu z lemmatizovaného textu Saturnin bez odstranění stop slov. Corpy. Zdroj: Vlastní zpracování. ....	19
Obrázek 2 Vizualní reprezentace nejčastějších slov s pomocí wordcloudu z lemmatizovaného textu Saturnin s odstraněním stop slov. Corpy. Zdroj: Vlastní zpracování. ....	20
Obrázek 3 Ověření, zda jsou tokeny známé pro nástroj. Simplemma. Zdroj: Vlastní zpracování. ....	20
Obrázek 4 Ukázka generování POS tagování s Chat GPT. Zdroj: OpenAI. (2023) [20]. ....	21

## Seznam tabulek

Tabulka 1 Porovnání lemmatizace a stematizace. Zdroj: Vlastní zpracování. ....	12
Tabulka 2 Over stemming. Zdroj: Vlastní zpracování. ....	12
Tabulka 3 Under stemming. Zdroj: Vlastní zpracování. ....	13
Tabulka 4 Seznam stop slov podle jejich frekvence z knihy Saturnin. Lemmatizace s CorPy. Zdroj: Vlastní zpracování. ....	23
Tabulka 5 Odstranění stop slov z knihy Saturnin – porovnání. Zdroj: Vlastní zpracování. ....	24
Tabulka 6 Odstranění stop slov z knihy Povídání o pejskovi a kočičce – porovnání. Zdroj: Vlastní zpracování. ....	25
Tabulka 7 Výsledky tokenizace s interpunkcí z „Saturnin“ a „Povídání o pejskovi a kočičce“. Zdroj: Vlastní zpracování. ....	26
Tabulka 8 Porovnání poměru slovních druhů v „Povídání o pejskovi a kočičce“ a „Saturnin“. Corpy, Stanza. Zdroj: Vlastní zpracování. ....	31

# 1 Úvod

Bakalářská práce je zaměřena na zkoumání technik přípravy českého textu pro strojové zpracování. Práce má posloužit k lepšímu porozumění principů zpracování jazyka a dostupným nástrojům pro přípravu textu v oblasti strojového zpracování přirozeného jazyka.

Pro detailní a správné strojové zpracování je klíčová oblast přípravy přirozeného jazyka. Kvalitní a vhodně připravená data mají přímý vliv na výkon NLP modelů.

Účelem práce je přiblížit čtenáři principy přípravu textu (se zaměřením na český text) pro další strojové zpracování, jak z hlediska filologického, tak i programátorského. V následujících kapitolách se budou zkoumat principy jednotlivých algoritmů. Dále obsahuje přehled dostupných nástrojů a jejich testování na reálných textech.

Práce se zaměřuje na důkladnou analýzu nástrojů určených pro přípravu textu před jeho podáním strojovým algoritmům. Cílem této práce je poskytnout ucelený přehled nad dostupnými nástroji, zhodnotit jejich klíčové vlastnosti a efektivitu v konkrétních scénářích. Přitom se snaží odpovědět na otázky týkající se optimálního výběru nástroje v závislosti na požadavcích a charakteristikách dané úlohy strojového zpracování textu.



## 2 Cíl práce

Účelem práce je přiblížit čtenáři principy přípravu textu (se zaměřením na český text) pro další strojové zpracování, jak z hlediska filologického, tak i programátorského – představuje pohled na proces přípravy textových dat pro další efektivní strojové zpracování a zároveň zvažuje i jiné přístupy (LLM).

Dále chce porovnat funkčnost existujících algoritmů, i jejich efektivnosti, se zaměřením na dlouhé texty a případné získání kontextu. Každý algoritmus bude otestován na dlouhém textu a na vybraných větách, které jsou pro individuální proces problematické.

Je kladen důraz na budoucí zpracování, některé nástroje jsou lepší pouze s ohledem na specifické odvětví.

Je klíčové porozumět struktuře českého textu a jeho problémům. Cíl analýzy je porovnat dostupné nástroje pro jednotlivé algoritmy podle několika kritérií, podle kterých bude možné odhalit jejich silné a slabé stránky. Testování probíhá na reálných textech, pro některé testy se testuje pouze na úryvcích.

Podle složitosti úkolu, který má nástroj splnit, je žádoucí otestovat více nástrojů se stejnou funkcí – bohužel pro složitější problémy existuje méně dostupných nástrojů.

## 3 Specifika českého textu vzhledem k angličtině, tokenizace, lemmatizace, desambiguace a part-of-speech tagování

Pro strojové zpracování jakéhokoliv jazyku se využívá NLP (natural language processing) neboli v češtině – zpracování přirozeného jazyka. Přirozeným jazykem se myslí jazyk vázaný na konkrétní skupinu lidí sloužící ke komunikaci <sup>[1]</sup> – čeština, angličtina, němčina apod, tedy jazyk, který byl vytvořen přirozeně. Nepřirozené nebo také umělé jazyky jsou např. fiktivní jazyky, které jsou do jisté míry založené na přirozeném jazyce, ale vytvořil je jeden člověk nebo malá skupina za nějakým účelem mimo prosté komunikace.

NLP má mnoho využití. Jedná se odnož informatiky se zaměřením na umělou inteligenci, která má za cíl naučit stroje porozumět textům i mluvenému slovu <sup>[1]</sup>.

S tím se pojí i další cíl – naučit je vnímat podtexty, náladu autora či jeho motivaci a získat vlastní postřehy.

Pro zpracování jazyka je nutné ho předem upravit, což zahrnuje několik procesů: *segmentace*, *sumarizace*, *normalizace*, *tokenizace*, *lemmatizace*, *stematizace*, *desambiguace*, *part of speech tagování* a mnohem více [1]. Na vytvoření efektivního algoritmu není třeba použít všechny tyto techniky, jejich užitečnost se může lišit jazykem, a především účelem vytvářeného algoritmu. Většina těchto technik má více přístupů, a ne vždy lze s přehledem určit, jestli se jedná o správný přístup.

Při práci je nutné zvážit problémy spojené s daným jazykem. Chyby v gramatice se pojí ke každému jazyku. Pro zpracování českého jazyka je specifické užití interpunkce a diakritiky, v závislosti na kontextu je důležité uvážit, zda ji autor textu bude využívat – zejména v rámci komunikaci přes internet (při vyhledávání, v recenzích a v komentářích) – psaní malých písmen pro názvy a jména či využití slangových slov je také velmi běžné (platí obecně i pro ostatní jazyky).

### **3.1 Jednotlivé kroky zpracování**

**Segmentace** neboli větné rozdělení pomocí triviálního algoritmu pokládá věty jednoduché a souvětí jako stejné struktury [2]. Netriviální algoritmus by rozdělil text na jednoduché věty, tedy souvětí by se rozdělila na jednoduché věty. Netriviální rozdělení může být obtížné. Ne každá tečka nebo čárka označuje konec věty (řadová čísla, zkratky) a konec věty neoznačuje vždy jen tečka (spojky, čárky, pomlčky), speciálním případem jsou nadpisy.

**Tokenizace** je proces rozdělování textu na menší části – může se jednat o slova, zkrácená slova (lemmatizací, stematizací), znaky [3], ... Složitějšími případy mohou být třeba zkratky (cz, a. s.), časové údaje apod. U některých algoritmů nastává problém při rozdělování interpunkce, v případech, kdy není rozdělena mezerou. Např. tři tečky za sebou je možné brát jako jeden token, protože společně vyjadřují něco lehce jiného než jediná tečka, naproti tomu přímá věta, která je zakončena tečkou a uvozovkami, je žádoucí rozdělit jako dva samostatné tokeny.

Zahrnutí interpunkce může a nemusí být žádoucí. To stejné platí o tzv. „stop slovech“. Stop slova jsou slova, která nenesou žádný dodatečný význam, nejčastěji se jedná o předložky, spojky, některá zájmena, pomocná slovesa a případně i slovní

spojení [4]. Často představují rušivý element. Další stop slova se odvíjejí od využití v NLP, jejich odstranění/ignorování stroji dovoluje se soustředit na důležitá slova. Například při zadání dotazu „co je to tokenizace“ do internetového vyhledávače, by byl dotaz rozdělen na tokeny „co“, „je“, „to“, „tokenizace“. Vyhledávač by se pokusil najít stránky, které obsahují tyto tokeny, našli bychom tedy i stránky, které nijak nesouvisí s naším dotazem. Běžně by tedy byla slova „co“, „je“ a „to“ na seznamu stop slov internetových vyhledávačů. Seznam stop slov může být velmi rozdílné délky. Nevýhodou je v některých případech odstranění i důležitých informací, pro překlady a sumarizace textu se odstranění stop slov nedoporučuje. Odstranění interpunkce a *stop slov* jsou jedny z technik normalizace textu.

Odstranění číslovek je zvláštní případ, dá se říct, že význam věty „Karel a Eva vychovali tři děti.“ je sdělit, že dva lidé mají děti a mimo kontext nezáleží, jestli jich mají deset nebo jedno; v širším kontextu záleží o jaký počet se jedná a z věty se může získat více informací, například že pár má zkušenost s výchovou dětí, nejspíš jsou to rodiči zmíněných dětí apod. Podobný případ by byla i vlastní jména, pokud se věta „Martin jede do Prahy.“ analyzuje samostatně, význam věty se nemění při záměně vlastních jmen „Tomáš jede do Olomouce.“, pokud stačí informace, že někdo někam jede.

Pro omezení náhodnosti, kterou jazyk má, je třeba text normalizovat, vytvořit normu – což povede k odstranění přebytečných/zbytečných informací a snadnější strojové zpracování [5]. Pro anglický jazyk je například adekvátní přepis staženého tvaru *we're* na *we are*. Dalším obecnějším příkladem je přepis časových údajů *12.12.* na *dvanáctého prosince* nebo *18:00* na *osmnáct/šest hodin*; přepis zkratk na plný tvar apod. Tento druh normalizace textu se běžně užívá například pro *text to speech* (text na řeč).

**Stematizace** je nalezení kořene slova viz tabulka 1, zatímco lemmatizace znamená zkrácení slova na jeho základní/slovníkový tvar. Obě techniky slouží k normalizaci textu, obě mají své výhody a nevýhody, liší se hlavně způsobem, jakým fungují a jejich výsledkem [8]. Algoritmus pro lemmatizaci se nazývá lemmatizátor (lemmatizer), pro stematizaci pak stematizátor (stemmer).

**Tabulka 1** Porovnání **lemmatizace** a **stematizace**. Zdroj: Vlastní zpracování.

Tokeny (slova)	Lemmatizace	Stematizace
tanečník	tanečník	tan
tancoval	tancovat	tan
v	v	v
tanečním	taneční	tan
sále	sál	sál

**Lemmatizace** je zpravidla přesnější, ale vyžaduje více dat/informací ohledně struktury jazyka, algoritmus pro lemmatizaci zabere více práce než algoritmus pro stematizaci [6]. Pro lemmatizaci potřebuje stroj znát slovní druhy v rámci daného jazyka, aby slovo mohl převést do základního tvaru platného pro slovní druh viz tabulka 1, i přestože slova „tanečník“, „tancovat“ a „taneční“ mají stejný původ, jejich základní tvar je rozdílný na základě jejich slovního druhu. V češtině se prakticky výhradně používají lemmatizátory, kvůli problémům s použitím stematizátorů.

Hlavní problémy stematizace je *over stemming* a *under stemming*. V případě *over stemmingu* je kořen slova shodný s kořenem jiného slova, které má jiným významem viz tabulka 2, což může vést k špatnému pochopení slov/považování je za synonyma, zatímco u *under stemmingu* se mění kořen díky rozdílnému tvaru při skloňování a změně slovnímu druhu, i přestože se jedná o významově stejná nebo příbuzná slova viz tabulka 3 [7]. Stematizátor je algoritmus, který postupně zkracuje slovo – *odřezává* písmena, dokud se nedostane ke kořenu, čeština má ovšem mnoho výjimek, které vyžadují složitější postup.

**Tabulka 2** Over stemming. Zdroj: Vlastní zpracování.

Slova s rozdílným významem		Společný kořen
písek	písmo	pís
píseň	pískat	pís
kolo	koleno	kol
léto	létat	lét
voda	vodit	vod

**Tabulka 3** Under stemming. Zdroj: Vlastní zpracování.

Slova	se	stejným	Kořen prvního slova	Kořen druhého slova
pták		ptačí	pták	ptač
střela		střílet	střel	stříl
dům		domov	dům	dom
okno		okenice	okn	oken
stolek		stůl	stol	stůl
kniha		knížka	knih	kníž

S ohledem na tyto informace, odvozování tvaru slova stematizací nepotřebuje kontext, zatímco lemmatizace ho vyžaduje, stematizace je založena na pravidlech, zatímco lemmatizace vychází spíše z lingvistiky.

V češtině může být odhalit význam slova, slovního spojení nebo i věty na základě kontextu výzva i pro člověka natož pro stroj.

**Desambiguace** pomáhá určit význam cílený autorem textu a odstraňuje mnohoznačnost v rámci daného kontextu [8]. Desambiguace je obtížný úkol. Komplikací jsou například metafory, homonyma (slova se stejným tvarem a rozdílným významem) použití více záporů (v českém jazyce), personifikace, sarkasmus, ironie, nejednoznačná zájmena, složité větové struktury (složitá souvětí), idiomy (např. „Šplouchá mu na maják.“, „Má se jako prase v žitě.“, ...), vtipy, slang, gramatické chyby apod. Homonyma představují nejčastější problém, který se desambiguace snaží řešit.

Existuje několik metod, ale testování zabere mnoho času, a i tak se nedá na proces naplno spolehnout [8].

**Part of speech (POS) tagging** je proces přiřazování slovních druhů (podstatná, přídavná jména, slovesa apod.) k jednotlivým slovům v textu [9]. Part of speech tagování pomáhá třídit slova do kategorií a je důležitý pro porozumění struktury vět (vztahy mezi slovy) a významu jednotlivých slov, což je klíčové například u překladů (zachování významu). Každému slovu nebo interpunkčnímu znaménku je přiřazen tag (sadová značka) – různé taggery POS mají různé způsoby tagování. Obecně tagy označují slovní druh a další detaily platné pro daný druh (rod,

jednotné/množné číslo, čas ...) [10]. Některé taggery obsahují mnohem víc informací než pouhé určení slovního druhu a díky pevné struktuře jsou dobře čitelné. Proces výběru mezi možnými tagy se nazývá syntaktická desambiguace – pokud slovo může být více slovních druhů [9]. Například ve větách „Lež má krátké nohy.“ a „Lež klidně!“ je slovo *lež* v prvním případě podstatné jméno, zatímco v druhé se jedná o sloveso. Taggery POS se učí na základě dat, existuje několik druhů: pravidlový, statistický (stochastický) a hybridní – každý přístup má své výhody a nevýhody [11]. S úkolem POS tagování z části souvisí úkol určení větných členů – pro určení vztahů jednotlivých větných členů ve větě: jak jsou na sobě závislé, jaká je jejich role a jakým způsobem jsou data strukturovaná.

Tag nástroje CorPy má patnáct míst, pokud není některé z míst využito, je označeno pomlčkou. Neexistuje slovo, které by mělo tag bez alespoň jedné pomlčky, nejvíc informací poskytuje tag o ohebných slovních druzích. Tag je sestaven tak, aby obsáhl informace specifické pro každý slovní druh a zároveň byl čitelný i ve chvíli kdy jsou hledány informace, které mají některé slovní druhy společné. Díky tomu mají tagy velký potenciál v otázce filtrování.

Kromě deseti českých slovních druhů rozlišuje i interpunkci a neznámé tokeny. Případné zpřesnění má rozsáhlý seznam specifik – pro podstatná jména, slovesa, zájmena, číslovky, přídavná jména a další.

První dvě místa slouží k určení slovního druhu a případnému zpřesnění. Všechna vlastní jména, která jsou ve tvaru přídavných jmen (Ivanský, Novotný, ...), jsou klasifikována jako podstatná jména. Třetí pozice označuje rod, čtvrtá číslo.

Rod a číslo poskytuje více informací hlavně o slovesech, podstatných jménech, základní tři rody (střední, ženský, mužský) jsou rozšířeny o další případy, ve větách „Kohoutek zakokrhal.“ a „Kohoutek mi kape.“ v prvním případě je určen pro kohoutek rod mužský životný, v druhém mužský neživotný. Kromě základních čísel (jednotné, množné) se určují i další např. pro podstatná jména pomnožná a slova s dvojným číslem.

Pátá pozice slouží k určení pádu, i některé předložky mají tuto informaci v tagu, např. s má sedmý pád, se kterým je běžně spojované, přesto jde o zvláštní případ, protože předložky mluvnický pád nemají.

Šestá a sedmá pozice se týká hlavně přídavných jmen a zájmen, šestá určuje rod (přivlastňovací rod) majitele (matčino – ženský, tátův – mužský, ...), sedmá číslo (přivlastňovací číslo) majitele (naše – my, vaše – vy, ...).

Osmá a devátá pozice u sloves popisuje osobu a čas. Desátá u přídavných jmen charakterizuje stupeň (velký, větší, největší). Jedenáctá popisuje, zda slovo je v negovaném tvaru, u sloves a přídavných jmen se často negované slovo při lemmatizaci převede na základní tvar, ne vždy se správně oddělí negace – v tagu je vidět, zda bylo původně slovo v negovaném tvaru.

Dvanáctá pozice označuje u sloves aktivní a trpný rod, třináctá určuje vid dokonavý, nedokonavý a obouvidový (slova, která mohou vyjadřovat oba vidy).

Případ předposlední pozice se objevuje zřídka, jde o spojení dvou nebo více slov (často rozdílných slovních druhů) – spojení je těžké zařadit do slovních druhů, abychom je spojené „aby“ a „bychom“, kdopak „kdo“ a „pak“ apod.

Poslední pozice označuje doplňkové informace, jde o jedinou pozici, kde pomlčka neznamená pouhé neurčení, v tomto případě označuje běžnou základní variantu slova v souladu s dobovou mluvou. Jinak se snaží označovat archaické, knižní výrazy (míru zastaralosti), hovorové výrazy (běžné, neobvyklé), zkráceniny apod.

Existují podobné způsoby značkování, obecně se způsob značkování průběžně mění a zahrnují se/vylučují se některé informace, některé POS taggery se snaží pouze o určení slovních druhů (Stanza). Podobný seznam morfologických značek poskytuje Český národní korpus, a i přestože má také patnáctimístný tag a podobnou strukturu, liší se v detailním rozpisu jednotlivých pozic <sup>[12]</sup>.

**Named entity recognition** (NER) neboli rozpoznávání pojmenovaných entit se soustředí na identifikaci pojmenovaných entit a jejich rozdělení do předem určených kategorií – typicky organizace (ORG), osoba (PER), místa (LOC) a různé (MISC) <sup>[13]</sup>. Existují i jiné typy rozdělení, od podobně obecných až po velmi detailně definovaných <sup>[14]</sup>. Tato úloha je dobře definovaná a rozsáhle zmapovaná především pro angličtinu, existují rozsáhlé výzkumy i v dalších jazycích – pro češtinu např. *Czech Named Entity Corpus* <sup>[14]</sup>. Existuje více způsobů řešení, pro kategorii vlastních jmen může stačit sada pravidel (velké písmeno/tvar), zatímco jiné kategorie jako místa mohou být definovány tím, že jsou na předem určeném seznamu –

nejznámější místa typu *Paříž, Eiffelova věž, Praha a Sněžka* tedy nejspíš budou správně označena jako místa, zatímco méně časté/známé ne.

NER se často využívá pro strojový překlad, vyhledávání informací, dále může sloužit i k desambiguaci – seznam kategorií může tedy obsahovat i časové úseky, názvy knih a filmů nebo kategorie specifické pro různé vědecké účely [14]. Příklad využití pro desambiguaci ve větě „Caesar byl římským vojevůdcem.“, kde je *Caesar* vlastním jménem, zatímco ve větě „Mám hlad, dal bych si salát caesar.“ se jedná o jídlo – existuje řada podobných příkladů, třeba lokace pojmenované podle známých osobností (stát Washington – George Washington), osobnosti a po nich pojmenované vynálezy (programovací jazyk pascal – Blaise Pascal; morseova abeceda – Samuel Morse) a další (apple – společnost Apple; společnost Tesla – Nicola Tesla). Využití NER je užitečné pro různé vědecké obory. Zaměření na medicínu je žádoucí, lze přizpůsobit kategorie pro snadnější uspořádání lékařského textu (nemoci, vitamíny, druhy léčeb, léky, orgány apod.) [14]. Stejně lze upravit kategorie i pro další odvětví – marketing, psychologie, právní věda, ekonomie, chemie a mnoho dalších.

Mezi techniky NER patří ruční sestavení pravidel, určení seznamu jednotlivých entit (vlastních jmen, organizací, míst, ...) – tyto metody jsou přesné, ale špatně zobecňují a pravidla/seznamy musí sestavit lingvista [14]. Zatím jsou nejúspěšnější statické modely, které se pomocí vybraných rysů (slovní druhy, koncovky, ...) učí z trénovacích dat – je potřeba velké množství ručně kategorizovaných dat [13]. Díky tomu, že je úloha dobře definovaná, lze vyvíjet modely pro češtinu

### **3.2 Zpracování pomocí LLM**

S vývojem umělé inteligence, výrazným pokrokem v odvětví strojového učení a velkých jazykových modelů je důležité a nevyhnutelné zamyslet se nad jejich využitím pro přípravu jazyka pro strojové zpracování.

**Large language models** (LLM) neboli velké jazykové modely jsou neuronové sítě, které se pomocí obrovských výpočetních zdrojů a testovacích dat snaží přesvědčivě odpovídat v přirozeném jazyce [15]. Jedná se o odvětví umělé inteligence, Důležitou otázkou zůstává možnosti využití, neboť bylo prokázáno, že v současnosti LLM umí napodobit lidské postoje, uvažování apod [15].



Modely při trénování analyzují a identifikují vzorce (slovní spojení, gramatická struktura a další NLP vlastnosti) v trénovacích datech, a ve chvíli kdy jsou zakomponovány do vnitřní struktury modelu, lze využívat tyto nabitě znalosti k různým účelům (generování textu, doplnění informací, odpovídání, překlady, přepis, ...) [15]. Nicméně některé spojení např. psychologické je pro LLM těžší vyhledat, problematický může být humor, sarkasmus a ironie [16].

Jedním z nejznámějších LLM je Chat GPT, který podobně jako ostatní LLM plně nezvládá všechny úlohy přípravy textu pro strojové zpracování.

Některé se LLM přímo zaměřují na úlohy spojené s přípravou jazyka pro strojové zpracování, nicméně i správné výsledky nejsou formátovány konsistentně, takže se s výstupy neparuje tak snadno. Příklad s Chat GPT ukazuje, že součástí rozsáhlosti a všestrannosti, kterou LLM mají, je nemožné zaměření na konkrétní složité problémy v oblasti NLP. Zároveň díky rozsáhlosti je těžké je kontrolovat a upravovat.

Po zdokonalení LLM bude žádoucí je využít i pro NLP.

### **3.3 Přehled dostupných nástrojů**

Všechny následující knihovny slouží k práci s NLP.

**Stanza** je knihovna Pythonu pod licencí Apache License. Obsahuje nástroje, které lze použít k převodu textu na seznamy vět a slov, lemmatizace, POS taggování, získání morfologických vlastností (rod, číslo, čas, ...), k analýze závislosti syntaktické struktury a rozpoznání entit. Nástroje fungují paralelně mezi více než 70 jazyky [17].

**NLTK** (natural language toolkit) je open source knihovna Pythonu. Poskytuje několik nástrojů, návodů a zdrojů (korpusy, lexikony, ...) pro analýzu, manipulaci a porozumění textu v *lidském* jazyce. Knihovna byla od roku 2009 průběžně aktualizována pro nové verze Pythonu. NLTK má mnoho funkcí, mezi hlavní patří segmentace, tokenizace, lemmatizace, stemming, part of speech tagging, odstranění stop slov a mnohem více. Ne všechny funkce jsou dostupné pro český jazyk.

**CorPy** je knihovna Pythonu, na rozdíl od NLTK je pod licencí GNU General Public License (copyleftová licence). Jedná se o práci Ústavu formální a aplikované lingvistiky Univerzity Karlovy v Praze. CorPy pomáhá přidat lingvistickou anotaci k zpracovávaným datům, což pomáhá k lepší analýze a interpretaci; další funkcí je

tokenizace, lemmatizace, POS taggování. Zajímavou možností představuje tvoření word cloudů (grafická reprezentace frekvence užívaných slov) a přenesení textu do fonetické podoby (grafem na foném – pouze v češtině). Knihovna je dostupná hlavně pro angličtinu a češtinu.

**Simplemma** je nástroj pod licencí MIT. K povrchovému vyhledávání používá lemmatizační seznamy odvozené z různých zdrojů. Slouží primárně k lemmatizaci. Rozšiřuje úlohu lemmatizace na s ní spojené úkoly a poskytuje nové přístupy k práci. Oproti jiným nástrojům lze snadno kombinovat lemmatizaci několika jazyků. V současnosti je simplemma dostupná plně nebo z části pro 49 jazyků, čeština je plně podporovaná a dle dokumentace je spolehlivá na 89 procent [18]. Některé rozšiřující funkce nejsou pro češtinu spolehlivé.

**Polygot** je knihovna Pythonu pod licencí GPLv3 a poskytuje některé nástroje pro NLP – tokenizace, detekce jazyka, NER, POS taggování a další. Většina funkcí je dostupná ve velkém počtu jazyků. Část funkcí lze otestovat v přímo v prohlížeči [19].

## 4 Metodika zpracování

Hlavní úkoly přípravy textu pro strojové zpracování je segmentace, tokenizace, normalizace, lemmatizace, POS tagování, NER. V práci se analyzují nástroje určené pro přirozené zpracování textu na jednotlivých úkolech.

### 4.1 Analyzované texty

Jako testovací texty poslouží knihy „Saturnin“ od Zdeňka Jirotky a „Povídání o pejskovi a kočičce“ od Josefa Čapka, které jsou používány s vědomím zákona č. 429/2022 Sb o právu autorském, § 39c Licence k rozmnožování díla pro účely automatizované analýzy textů nebo dat: (1) Do práva autorského nezasahuje ten, kdo zhotoví rozmnoženinu díla za účelem automatizované analýzy textů nebo dat v digitální podobě, prováděné za účelem získání informací, zahrnujících mimo jiné vzory, tendence a souvztažnosti; takto zhotovenou rozmnoženinu je oprávněn uchovat pouze po dobu nezbytnou pro účely této automatizované analýzy textů nebo dat.

„Povídání o pejskovi a kočičce“ je dětská kniha od Josefa Čapka o příhodách kočky a psa v lidském světě. Celkem má text 7058 slov.





**Obrázek 2** Vizuální reprezentace nejčastějších slov s pomocí wordcloudu z lematizovaného textu Saturnin s odstraněním stop slov. Corpy. Zdroj: Vlastní zpracování.

U **tokenizace** je hodnocena přesnost počtu tokenů v textu, správnost a problémové případy.

U **segmentace** se porovnává schopnost rozdělování vět, správnost, způsob rozdělování a problémové případy.

U **Lemmatizace** se posuzuje přesnost, obsáhlost viz obrázek 3, správnost, přístup k neznámým slovům, pokusy o desambiguace, práce s výstupem, zpracování zkratk, odstranění negace a další.

```

1  from simplemma import simple_tokenizer, is_known
2  tokeny = simple_tokenizer("Vyrázil jsem z křesla jako šílenec a skočil po něm.")
3  neznameTokeny = []
4  i, j = 0, 0
5  for token in tokeny:
6      if is_known(token, lang='cs'):
7          i += 1
8      else:
9          j += 1
10     neznameTokeny.append(token)
11 print("Známé: " + str(i))
12 print("Neznámé: " + str(j))
13 print(neznameTokeny)

```

**Obrázek 3** Ověření, zda jsou tokeny známé pro nástroj. Simplemma. Zdroj: Vlastní zpracování.

Pro **POS tagování** je důležitý způsob sestavení tagu, správnost, rychlost, manipulace s výstupem.

Podobně jsou hodnoceny i další procesy zpracování přirozeného jazyka.

Několik testů proběhlo s LLM Chat GPT. Při položení otázky „Jaký je případ slova ohebného slovního druhu, které na základě kontextu mění význam?“ chat bot odpovídal špatně viz obrázek 4, přičemž příklad byl několikrát replikován se stejně špatným výsledkem, i přestože čeština má mnoho platných odpovídajících případů a i příklad (slovo „želva“), který Chat GPT uvádí, by mohl být zpracován, tak aby byl validní – i kvůli ohebnosti češtiny. Díky chybovosti a povrchnosti LLM pro některé úkoly nejsou v práci porovnávány s knihovnami Pythonu.



#### ChatGPT

Jedním z příkladů slova, které může změnit svůj slovní druh na základě kontextu, je "želva."

Příklady:

1. **Podstatné jméno:** Můj bratr si koupil novou želvu.
2. **Sloveso:** Pokud želva, běž rychleji než včera!
3. **Přídavné jméno:** Viděl jsem želvu zelenou.

V každém z těchto případů má slovo "želva" jiný slovní druh v závislosti na kontextu věty.

**Obrázek 4** Ukázka generování POS tagování s Chat GPT. Zdroj: OpenAI. (2023) [20].

### 4.3 Používané nástroje

Veřejně dostupné nástroje jsou převážně k dispozici jako open source (otevřené) knihovny, většina testování probíhalo v Pythonu. Mezi tyto nástroje patří Stanza, CorPy, simplemma, Polygot, repositáře githubu stopwords-iso a python-stop-words.

- Python: Verze 3.12.1
- NLTK: Verze 3.8.1, poslední aktualizace 2. ledna 2023
- Stanza: Verze 1.8.1, poslední aktualizace 1. března 2024
- Corpy: Verze 0.6.1, poslední aktualizace 5. dubna 2023

- Simplemma: Verze 0.9.1, poslední aktualizace 20. ledna 2023
- Polygot: Verze 16.7.4, poslední aktualizace 22. září 2020
- Stopwords-iso: poslední aktualizace 17. září 2020
- Python-stop-words: poslední aktualizace 23. července 2018

## 5 Výsledky testování na reálných textech

### 5.1 Stop slova

NLTK v základu nenabízí seznam českých stop slov, ale existuje několik volně dostupných seznamů. Další možností je sestavit si vlastní.

Centrum zpracování přirozeného jazyka Masarykovy univerzity poskytuje vlastní seznam stop slov, slova jsou vybrána dle jejich frekvence objevování v korpusu **DESAM** (viz <https://nlp.fi.muni.cz/cs/StoplistZakladnichTvaru>), text prošel lemmatizací. Výběr stop slov dle četnosti se snadno implementuje, ale zdá se poměrně jednostranný, neboť i často se vyskytující slova mohou nést význam a slova s nízkou frekvencí naopak nemusí nést význam. Dále závisí na daném korpusu, častá slova v jednom korpusu nemusí být v jiném – seznam stop slov z korpusu DESAM například obsahuje i diskutabilní slova „léto“, „Praha“, „strana“ a další. Výběr závisí i na délce korpusu, obecně čím je korpus delší tím menší je potřebný poměr počtu výskytu slova na počet slov v korpusu.

Při praktické aplikaci **vlastního frekvenčního algoritmu** na knize „Saturnin“ se mezi stop slovy objevila i vlastní jména viz tabulka 4, při reálném použití by záleželo na typu analýzy, pro analýzu obecných vlastností textu by vlastní jména mohla být rušivá. Testovací text „Povídání o pejskovi a kočičce“ měl na seznamu i slova „pejsek“ a „kočička“, což by v tomto případě fungovalo jako substituce jejich jmen. Poměr výskytu slova a celkového počtu slov převedený na procenta měl vždy přesahovat 0,15 %, aby se slovo dostalo na seznam – na základě seznamu Masarykovy univerzity bylo zvoleno několik hranic, byla vybrána hranice, která nejlépe reprezentovala problematiku výběru stop slov podle frekvence. Pro text jiné povahy a délky by byla stavena jiná meze (pro kratší text by byla meze větší, pro delší menší).

**Tabulka 4** Seznam stop slov podle jejich frekvence z knihy Saturnin. Lemmatizace s CorPy. Zdroj: Vlastní zpracování.

Slovo	Procento výskytu	Slovo	Procento výskytu	Slovo	Procento výskytu	Slovo	Procento výskytu
být	6,628	svůj	0,512	Milouš	0,285	něco	0,190
se	4,233	Saturnin	0,499	pravít	0,284	chvíle	0,180
a	4,140	jeho	0,485	Kateřina	0,269	potom	0,176
že	3,145	k	0,481	jak	0,262	celý	0,174
ten	3,104	tak	0,470	pan	0,260	muset	0,172
on	1,641	teta	0,467	takový	0,259	jít	0,169
v	1,557	moci	0,454	vědět	0,244	li	0,169
na	1,553	jako	0,434	už	0,242	věc	0,169
já	1,113	když	0,434	velmi	0,233	ptát	0,167
mít	0,849	všechn	0,429	pak	0,232	náš	0,165
s	0,795	slečna	0,406	říkat	0,226	i	0,163
do	0,724	my	0,386	ještě	0,208	než	0,156
který	0,700	Barbora	0,381	myslit	0,206	člověk	0,154
aby	0,664	co	0,375	vy	0,206	den	0,153
ale	0,625	za	0,375	kdyby	0,201	lidé	0,153
o	0,592	doktor	0,343	pro	0,201	pokoj	0,153
dědeček	0,589	můj	0,321	chtít	0,199		
z	0,585	po	0,321	protože	0,199		
řící	0,540	Vlach	0,309	tento	0,197		

Repositář **stopwords-iso** (dostupný na githubu) poskytuje seznam stop slov 58 jazyků (viz <https://github.com/stopwords-iso>), lze implementovat i jen pro jednotlivé požadované jazyky. Český seznam je poměrně rozsáhlý (423 slov), obsahuje předložky, spojky, pomocná slovesa, zájmena, některé citoslovce, několik podstatných jmen a číslovky. Seznam není konzistentní, co se týče diakritiky (slova jsou napsána s diakritikou, některá jsou napsána i bez) a obsahuje několik chyb. Jeden z důvodů obsáhlosti je univerzálnost, text musí předem projít jen tokenizací, některá častá slova jsou ve všech tvarech – např. slovesa „být“ a „dělat“ jsou vypsána

ve všech časech a osobách, seznam ale obsahuje jejich negace, ale pokud je po implementaci požadováno zanechání smyslu vět, není dobré negace v stop listu ponechávat – např. pro účely spojené se získáním informací, analýzy sentimentu, překladu apod. Implementace je snadná, seznam je dostupný na bower a npm (správci balíčků).

Repozitář **python-stop-words** (dostupný na githubu) obsahuje seznam stop slov 22 jazyků (viz <https://github.com/Alir3z4/python-stop-words>). V českém seznamu (256) chybí některá velmi běžná stop slova (a, aby, ani, až, by, být, i, k, o, s, u, v, z, ...). Seznam se snadno implementuje přes pip (správce balíčků), nelze zapracovat pouze pro jeden jazyk.

Slova na seznamu stop slov z korpusu DESAM, který je sestaven podle frekvence slov v korpusu, činí celkem 25 procent z celého textu, takže při použití na jiném textu viz tabulka 5 a 6 zůstává efektivní a poměrně konsistentní – první text zredukoval o 41 procent a „Saturnin“ o 37 procent. Seznam dle frekvence je nejspíš zbytečně přísný, ale byl sestaven podle určených parametrů a odstraňuje odpovídající počet slov („Povídání o pejskovi a kočičce“ – 59 procent, „Saturnin“ – 46 procent). V porovnání stop listů na jiné texty než, ze kterých byly vytvořeny, text z „Povídání o pejskovi a kočičce“ byl zredukován o 47 procent, zatímco text z „Saturnin“ o 41 procent, takže seznam frekvence „Saturnin“ se blížil více původnímu výsledku. Podle těchto dat lze určit, že seznam dle frekvence je mnohem přesnější pro delší text, čemuž odpovídá i seznam DESAM.

**Tabulka 5** Odstranění stop slov z knihy Saturnin – porovnání. Zdroj: Vlastní zpracování.

<i>Celkem slov:</i> 55795		<i>Jedinečná slova:</i> 7989		
Nástroje	Lemmatizace*	Počet zbývajících slov		Délka seznamu
		S lemmatizací	Bez lemmatizace	
<i>Desam</i>	ANO	35121 (63 %)	44059 (79 %)	75
<i>Podle frekvence</i>	ANO	30183/32966 (54 %/59 %)	41408/42853 (74 %/77 %)	74/73
<i>Stopwords-iso</i>	NE	30320 (54 %)	31892 (57 %)	423



<i>Python-stop-words</i>	NE	40651 (73 %)	39901 (72 %)	256
--------------------------	----	--------------	--------------	-----

*\*Lemmatizace – je nutné předem text lematizovat*

**Tabulka 6** Odstranění stop slov z knihy Povídání o pejskovi a kočičce – porovnání.  
Zdroj: Vlastní zpracování.

<i>Celkem slov:</i>		7056	<i>Jedinečná slova:</i>	1318
Nástroje	Lemmatizace*	Počet zbývajících slov		Délka seznamu
		S lematizací	Bez lematizace	
<i>Desam</i>	ANO	4132 (59 %)	5319 (75 %)	75
<i>Podle frekvence</i>	ANO	2903/3571 (41 %/53 %)	4705/5200 (67 %/74 %)	74/73
<i>Stopwords-iso</i>	NE	3571 (51 %)	3703 (52 %)	423
<i>Python-stop-words</i>	NE	4838 (59 %)	4868 (59 %)	256

*\*Lemmatizace – je nutné předem text lematizovat*

*Seznamy stopwords-iso a python-stop-words měly zanedbatelný rozdíl ve výsledku, ať už text prošel lematizací nebo ne. Python-stop-words ztrácel na efektivitě, pokud byl testovací text delší, zatímco pro stopwords-iso výsledky zůstávaly podobné, i přestože mírný pokles v účinnosti byl zaznamenán.*

Seznam podle frekvence a DESAM zůstaly efektivní i přes jejich délku a výsledky pro odstranění slov, když text neprošel lematizací, odpovídají poměru slov neohebných slovních druhů (hlavně spojky a předložky).

## 5.2 Tokenizace a segmentace

NLTK poskytuje několik možností pro tokenizaci – velmi efektivní bylo využít typ tokenizátoru, který z výsledných tokenů odstraní interpunkci.

CorPy a v menší míře i **simplemma** zahrnovala do tokenů například i odstavce (nové řádky) v textu. Tokenizace nebyla problematická, práce každého zmíněného

tokenizátoru byla v praxi dostačující, ale interpunkce se špatně odstraňuje viz tabulka 7.

**Tabulka 7** Výsledky tokenizace s interpunkcí z „Saturnin“ a „Povídání o pejskovi a kočičce“. Zdroj: Vlastní zpracování.

Nástroje	Povídání o pejskovi a kočičce	Saturnin
Celkem slov	7058	55699
NLTK <i>bez interpunkce</i>	7056	55795
NLTK	8928	65698
CorPy	9032	66015
Stanza	9054	66407
Simplemma	8737	65527

Segmentace vět **CorPy** rozdělí pouze věty pouze pomocí koncové interpunkce (tečka, otazník, vykřičník) – tzn. nerozděluje dále souvětí na věty. Při testování na reálném textu zvládá dobře ignorovat některé případy koncové interpunkce – přímá řeč, řadové číslice, data, iniciály a zkratky. Nicméně často problém nastával v případech, kdy po koncové interpunkci bezprostředně následovalo velké písmeno. **Stanza** rozdělila věty lépe, pokud šlo o interpunkci uprostřed věty – naopak občas nastával problém u některých vět, které byly zakončeny tečkou a následovalo velké písmeno, jednalo se o výjimečný případ, věty z knihy Saturnin: „Vyrazil jsem z křesla jako šílenec a skočil po něm.“ a „Bohužel, nebylo mi dopřáno, abych mu ublížil.“ se nerozdělily jako dvě věty ale jako jedna. Oproti CorPy zvládla Stanza rozdělit i nadpisy.

### 5.3 Lemmatizace

Při zpracování s **CorPy** se defaultně zpracovává detailně celá věta, token po tokenu, výstupem po zpracování jednoho slova je lemmatizace a jeho tag. Při lemmatizaci ovšem do jisté míry dochází i k desambiguaci např. pro případy, které jsou specifické pro češtinu např. „se“ je klasifikované jako zvrtné zájmeno nebo částice, což dodává doprovodnou informaci, i přestože z tagu se dá dočíst, že se jedná o reflexivní zájmeno. Dále některá archaická slova jsou vysvětlena: jenž\_^(který\_[ve\_vedl.větě]).

Podobně pomáhá lépe klasifikovat některá homogenní slova a odstraňuje nejednoznačnost, ve větách z knihy „Saturnin“: „Zjistil si, do kterých evropských států je ten časopis posílán...“, „Mohlo se mu stát ...“ a „Obvykle prováděl Saturnin věci, nad kterými zůstával rozum stát.“ při lemmatizaci označí nejen základní tvar, ale dále i odliší případy podle významu nebo kontextu, takže je snadnější identifikovat jejich význam i bez kontextu: stát-1^(státní\_útvár), stát-2^(stanu\_staneš) a stát-3^(stojím\_stojíš). Ne vždy je dále popsán smysl, ale je číselně naznačeno, že má slovo více významů, v praxi může být ovšem těžké zjistit jeho další význam – ve větách z „Povídání o pejskovi a kočičce“: „... která se ji do kopřiv *hodila* ...“ a z „Saturnin“: „V tom případě by se spíš *hodilo* přísloví.“ je rozlišen význam hodit-1, hodit-2^(bude\_se\_hodit), kde se tag liší pouze osobou a číslem. Běžně se snaží zamezit morfologické desambiguaci. Dělá i chyby, ve větě z „Povídání o pejskovi a kočičce“: „Podlaha byla stále mokrá a moc čistá také nebyla.“ bylo slovo *moc* v tagu označené jako podstatné jméno, z popisku moc-1^(nad\_někým;\_politická,\_vojenská;\_plná,...) lze vyčíst, podle jakého významu bylo slovo zpracováno.

Známější zkratky se lemmatizují do jejich rozepsaného tvaru, ať už se jedná o iniciálovou (akronym), rámcovou, skeletovou, lineální apod., ať už je rozdělena tečkami nebo ne, nebo jsou dokonce v jiném jazyce. Zatímco zkratky jako „VIP“, „tzn.“ a „prof.“ se v závorce vypíší v plném tvaru – nicméně při tvorbě tagu se plný tvar nerozdělí na jednotlivé slova a v tagu se celé spojení označí jako zkratka. Naopak „s.r.o.“ se rozdělí na šest tokenů, ale rozepíše písmena na zamýšlené tvary „společnost“, „ručení“, „omezený“ a tag je již v klasickém tvaru.

**Simplemma** při porovnání s CorPy se nesnaží o desambiguaci, je tedy snazší pracovat s výstupním formátem a obecně je univerzálnější. Pro porovnání byl použit úryvek z knihy „Povídání o pejskovi a kočičce“, simplemma mělo trochu větší odchylku. Má mnoho možností, adaptuje se a manipuluje se s ní lépe než s CorPy. Velkou nabídku jazyků lze kombinovat, pokud je známo, jaké jazyky text obsahuje (v současnosti je běžné zahrnutí anglických slov), lze specifikovat více jazyků a nástroj je schopný lemmatizovat celý text.

Dále lze se simplemma testovat, zda se slovo vyskytuje v seznamu známých slov lemmatizace pro daný jazyk. V praxi tedy lze porovnat obsáhlost nástroje a případně

najít snadněji jeho slabé stránky. Na testovacích textech šlo o cca 20–21 procent a při manuálním procházení případů se jednalo o archaismy, zdrobněliny, přivlastňovací zájmena, jména, i o běžná slova a velkou část nepřekvapivě sestává ze slova neohebných slovních druhů, která by se stejně nelemmatizovala. Po odstranění stop slov šlo o 12–17 procent, přičemž nižší procenta neúspěšné lemmatizace byla ze „Saturnina“, ze stránek dokumentace simplemma se dá dočíst neúspěšná lemmatizace 11 procent.

Další funkce simplemma je zjištění jazyka, tedy kolik procent textu se nachází v dostupném lemmatickém seznamu pro zkoumaný jazyk, podobně jako u předchozí metody, seznam neobsahuje některá slova, která se dál nelemmatizují, reálný poměr by byl tedy trochu jiný. Tokeny v textu „Saturninu“ jsou z 85 procent v českém lemmatickém listu, 35 procent ve slovenském a 15 procent je neznámo, v druhém textu 76 procent v českém, 32 procent v slovenském a 24 je neznámo. Procenta dohromady nedávají 100 i kvůli porovnání na více jazycích, některá slova jsou v českém i slovenském lemmatickém listu, zároveň jde o individuální porovnání – oba testovací texty jsou čistě české, prakticky neobsahují cizí slova, proto součet procent poměru českých a neznámých slov je 100 procent. Výsledky poměru zkoumaného jazyka ovlivňují i faktory jako využití archaické češtiny, slang a celková délka textu.

Pokud jde o chyby, oba nástroje pracují jinak – chyby, které nastaly u jednoho z nich, nenastaly u druhého a naopak. CorPy také odstraňuje negaci sloves, zatímco simplemma ji ponechává. Informaci o negaci je s CorPy zapsaná v tagu.

Podobně jako se seznamy stop slov i pro lemmatizaci existují seznamy lemmatizovaných slov. Práce s nimi není tak snadná, ale minimálně podle nich lze postavit vlastní algoritmus. Oproti simplemma, která obsahuje celkem kolem 187 000 tvarů českých slov, existující seznamy jsou mnohem kratší, proto nemá smysl je testovat v porovnání s komplexnějšími seznamy jako má zmíněná simplemma a CorPy.

**Stanza** oproti CorPy a simplemma zpracovávala text velmi pomalu, práce s ní ovšem působila jako práce s jedním celkem, a zároveň byl každý aspekt snadno oddělitelný. Stanza podobně jako CorPy odstraňuje negaci, dále uchovává informaci o tom, že proběhla v rozšířeném tagu. Byla velmi přesná. Některé pokusy o lemmatizaci

vypadaly jako stematizace (under stemming), docházelo tedy k pouhému odstranění koncovek u některých přivlastňovacích jmen, nebo přímo ke špatnému výslednému tvaru slova. Poměr chyb na celý text byl zanedbatelný.

#### 5.4 Part of speech tagování

Pro porovnání tagů s CorPy slova *natáhnout* ve větách z „Saturnin“: „...*natáhl* jsem se pohodlně v hlubokém křesle.“ a „...šofér ležel *natažen* na pryčně v budce vrátného a spal.“ je pro první příklad 'VpYS----R-AAP-1' a pro druhý 'VsYS----X-APP-1', přičemž první písmeno identifikuje slova jako slovesa, druhé u *natáhl* určuje, že se jedná o příčestí minulé, aktivní (činný rod), u *natažen* příčestí minulé, pasivní (trpný rod), obě dle třetí a čtvrté pozice mají rod mužský, číslo jednotné. Na deváté pozici lze rozpoznat čas, kde *natáhl* je minulém čase a *natažen* může být v jakémkoliv – v tomto případě nelze určit čas jen ze slovesa *natažen* – ale ze sponového slovesa *být*.

Při testování se často určilo pouze základní číslo.

Ve větě z „Saturnin“: „Jen maska bolesti na tváři *mužově* byla pravá.“ je tag slova *mužově* 'AUFs6M-----', kde je na třetí pozici určen rod (ženský) podle koncovky a na šesté pozici rod (přivlastňovací rod) *majitele* (mužský).

Ve větách „Bylo to nesmyslné, ale někam jsem musel jít.“ a „Propustil mne v náladě velmi nepřívětivé.“ z knihy „Saturnin“ je slovo *nesmyslné* v základním tvaru podle CorPy *nesmyslný*, zatímco slovo *nepřívětivé* je v základním tvaru *přívětivý*, takže i přestože se jedná o podobný příklad, v prvním příkladu není v tagu slovo označeno jako negované a v druhém je negované. I přes tyto nesrovnalosti nedochází k přímé ztrátě informací, protože v prvním případě zůstává slovo v negované formě, i přestože si to nástroj neuvědomuje, zatímco v druhém, který se správně převedl, lze opakovaně negovat pro zachování významu.

Na poslední pozici tagu se nejčastěji objevovala pomlčka (značí běžnou variantu), zřídka kategorie pro méně časté pojmenování, většinou u slov, které mají známější synonymum – kapat, drhnout, kleknout, nejdříve, štípat a další. Ostatní případy byly velmi výjimečné. Zkráceninami je myšleno např. jednotky času, vzdálenosti apod. Záleželo také na přesném tvaru slova. Slova, která byla ve dvojném čísle (a měla i množné číslo), se považovala za hovorová – *s dětma* (místo *s dětmi*), *s kolenama*

(místo *s koleny*), *s okama* (místo *s oky*) apod, v těchto případech je většinou dvojné číslo chybné, přesto stále zůstává součástí hovorové češtiny, CorPy tyto případy správně převede a v tagu zanechá informaci o původním tvaru. Obecně jsou tagy vytvořené s CorPy správné.

Tagger **Stanza** určuje pouze slovní druhy, oproti CorPy sice tedy poskytuje méně informací, zároveň jsou získané informace snazší pro další zpracování. Tagger má sedmnáct možností, mezi obvyklé české slovní druhy jsou zařazena pomocná slovesa, vedlejší spojky, vlastní jména, determinátory (v angličtině typicky *the, a, an*), symboly a interpunkce. Determinátory v češtině se dají popsat jako slova, která většinou předcházejí jiné slovo a popisují ho nebo mu dodávají nějaký význam, což zahrnuje některá zájmena (*váš, toho, každý, svůj, ...*) [21]. I slova, která v rámci kontextu mění svůj slovní druh nebo i význam, byla správně zařazena. V úryvku z „Povídání o pejskovi a kočičce“: „Nás, pejsky, zas tahají za uši, a *to* vám bolí, že se *to* ani vypovědět nedá.“ v druhé větě slovo *to* Stanza označila jako částici, a i přestože by se obvykle jednalo o ukazovací zájmeno, v rámci kontextu věty by se dalo považovat za částici – v třetí větě, kde mělo zjevnou funkci zájmena, bylo správně zařazeno. Obecně nebyly zaznamenané větší chyby, mimo komplikovanější případy, kdy správnost obou stran (nástroje a lidské), nebyla jednoznačná – slovo *například* bylo CorPy i Stanza zařazeno do kategorie částic, přestože je v češtině zařazeno do příslovčí jako příslovečná spřežka, která vznikla spojením podstatného jména a předložky. Nastávaly podobné problémy se stejnými slovy jako u lemmatizace, některá přivlastňovací přídavná jména byla Stanza určena jako podstatná, a pokud bylo *vlastnictví* vyjádřeno vlastním jménem, slovo bylo většinou zařazeno do skupiny názvů/vlastních jmen.

Rozšířený tag je také patnáctimístný a má podobné vlastnosti jako CorPy. V porovnání na testovacích textech byl tagger CorPy přesnější (poskytoval více detailů), pád a další doplňkové informace určoval lépe než tagger Stanza. Mimo tagger lze získat rozepsané obecné morfologické údaje o jednotlivých výrazech a snadno získat pouze žádané informace (rod, číslo, pád apod.).

Porovnání výsledků CorPy a Stanza viz tabulka 8 ukazuje rozdílný přístup k rozdělení. Některé vzory se propsaly v případech obou testovacích textů – např.

podobné rozdíly v poměrech. Stejný počet slovního druhu a stejný procentní poměr je třeba rozlišovat.

**Tabulka 8** Porovnání poměru slovních druhů v „Povídání o pejskovi a kočičce“ a „Saturnin“. Corpy, Stanza. Zdroj: Vlastní zpracování.

Slovní druh	Povídání o pejskovi a kočičce		Saturnin	
	CorPy	Stanza	CorPy	Stanza
Podstatné jméno	1374 (15,2 %)	1455 (16,1 %)	12695 (19,2 %)	12794 (19,3 %)
Přídavné jméno	286 (3,2 %)	322 (3,6 %)	4363 (6,6 %)	4925 (7,4 %)
Zájmeno	1346 (14,9 %)	1334 (14,7 %)	8692 (13,2 %)	8783 (13,2 %)
Číslovka	57 (0,6 %)	39 (0,4 %)	533 (0,8 %)	313 (0,5 %)
Sloveso	1645 (18,2 %)	1665 (18,4 %)	13392 (20,3 %)	13414 (20,2 %)
Příslovce	764 (8,5 %)	817 (9 %)	4375 (6,6 %)	4526 (6,8 %)
Předložka	495 (5,5 %)	495 (5,5 %)	4627 (7 %)	4661 (7 %)
Spojka	852 (9,4 %)	863 (9,5 %)	6437 (9,8 %)	6404 (9,6 %)
Částice	115 (1,3 %)	75 (0,8 %)	582 (0,9 %)	432 (0,7 %)
Citoslovce	31 (0,3 %)	11 (0,1 %)	18 (0,03 %)	19 (0,03 %)
Interpunkce	1976 (21,9 %)	1966 (21,7 %)	10132 (15,35 %)	10132 (15,3 %)
Jiné	91 (1 %)	12 (0,1 %)	169 (0,3 %)	4 (0,01 %)
Celkem	9032	9054	66015	66407

## 5.5 Named entity recognition

**CorPy** se snaží o NER – pojmenované entity a některé odborné termíny se řadí do kategorií, bohužel pojmenování kategorií není příliš intuitivní. Nejčastější kategorie, které se objevovaly v testovacích textech, byla vlastní jména (Y), příjmení (S), místa (G) a příslušníci kmenů a národů (E). Celkem je kategorií 20 a každé slovo může být v několika kategoriích zároveň.

V textu „Saturnina“ se například objevuje postava doktora Vlacha, který se zařadil do kategorie vlastních jmen a národa – zařazení do vlastních jmen není nutně špatně, při existenci kategorie příjmeních by mělo toto jméno spadat spíše tam – při testování docházelo často k této záměně. Zároveň „Vlaši“ je staročeské označení pro románský národ [22], což je informace, kterou by lidský subjekt z textu přirozeně nezískal, i přestože v tomto kontextu má jméno pravděpodobně fungovat pouze jako příjmení.

Pro určení vlastního jména nebo příjmení bylo pro program klíčové, zda má slovo velké písmeno, proto bez rozdílu na kontextu *mlynář* na začátku věty byl vždy vlastním jménem.

**Polygot** má tři základní kategorie osoba (PER), místo (LOC) a organizace (ORG), na reálných NER pracoval dobře a i organizace, které byly napsány s malým písmenem zařadil do správné kategorie. Pokud nespádala jmenná entita do kategorie nebyla označena. Polygot častěji zahrnul více slov do některé z kategorií, než by nějakou jmennou entitu vynechal. Ve větách „Miluji Německo.“ a „Miluji tě.“ se v prvním případě se slovo *miluji* označilo jako lokace, v druhém jako osoba – pravděpodobně nastal problém kvůli velkému písmenu, naproti tomu ve větě „Vlastním s.r.o. firmu.“ se písmeno *r* ve zkratce *s.r.o.* označilo jako osoba. Pro triviální rozšíření byl Polygot dostačující, zároveň demo nástroje pro NER šlo vyzkoušet i v prohlížeči.

**Stanza** při tagování zařazuje slova do kategorie názvů/vlastních jmen (proper noun), rozsahem je s Corpy neporovnatelná, ale pro základní rozřídění v rámci jiného úkolu může posloužit dobře. Zároveň podobně jako CorPy zařazuje do kategorie názvů i názvy společností a značek, nicméně slovo musí začínat velkým písmenem. Nicméně čeština je jedním z mála jazyků, která nemá svůj NER model, nejznámější entity, křesní jména a příjmení dokáže rozeznat např. NER model pro angličtinu.

## **5.6 Další způsob přípravy textu**

**Stanza** neposkytuje analýzu sentimentu ani sumarizaci textu pro češtinu, naproti tomu ale vytváří vhodné podmínky pro vytvoření algoritmu pro sumarizaci textu. Lze například filtrovat věty podle délky, nebo je seřadit, což přispívá k získání



klíčových vět pro sumarizaci. Svými funkcemi představuje základ pro sestavení sumarizačního algoritmu.

Filtrování podle délky vět pro „Povídání o pejskovi a kočičce“ napovídá o struktuře – text obsahuje několik složitých dlouhých souvětí. U „Saturnin“ by mohlo filtrování sloužit ke zkrácení textu.

## 6 Shrnutí a diskuse výsledků

Při odstraňování stop slov frekvenčními seznamy, bylo předem potřebné text tokenizovat a lemmatizovat. Odborně sestavený seznam z korpusu DESAM si zachovával podobnou úspěšnost, zatímco algoritmem sestavený seznam bylo třeba protřídit, aby se neodstranily i podstatné informace. Oba seznamy byly krátké a snadno se s nimi manipulovalo.

Seznamy zaměřující se také na jiné jazyky byly mnohem delší, vybraná slova neměla soudržnou gramatiku, některé byla chybná a jiná ke dalšímu zvážení. Odstranily konsistentní poměr slov, přesto nebylo jasné, jaký je cílený poměr odstraňovaných slov – jednak v rámci konkrétních seznamů, ale i v obecně českém jazyce. Implementace byla snazší, přístup k seznamům v jiných jazycích může být pro některé aplikace užitečný. Nebylo nutné předem text lemmatizovat, při porovnání výsledků si zachovaly podobný poměr odstraněných slov. Repozitář *python-stop-words* neodstraňoval jedny z nejčastějších stop slov.

V tématu seznamů stop slov je nutné si uvědomit, že nejde o to, aby procento odstraněných slov bylo co největší, záleží hlavně na udržení stálých výsledků. U listů, které předem musely projít lemmatizací, bylo nutné počítat se špatnou lemmatizací, což mohlo lehce zkreslit získaná data.

Nástroj CorPy měl obecně dobré výsledky, ale oproti Stanza mu chyběla ucelenost. V rámci testování bylo zobrazeno zároveň původní slovo, lemmatizované slova a jejich tag a v porovnání byla jasná občasná nedostatečnost propojenosti – některé informace získané v lemmatizaci se nepropsaly do tagu.

Práce na lemmatizaci s CorPy je dvojsečná, záleží na využití – pro analýzu sentimentu, překlad, získání informací a další podobné úlohy jde o vhodný nástroj, ale pro statistiky, analýzy frekvence je odstranění desambiguace nepotřebné. Obecně nástroj není přesný, ale má téměř zanedbatelnou odchylku.

Snahy o odstranění desambiguace vypadaly velmi slibně, odstranění syntaktické i morfologické desambiguace bylo převážně správné – odlišení významu slov se stejným slovním druhem zásadně pomáhalo k přípravě textu pro další zpracování – např. v úloze strojového překladu.

Některé informace získané v rámci lemmatizace se nepřenesly na NER.

Rozdělování na tokeny bylo problematické, bylo částečně tokenizované i neviditelné formátování (odsazení, konec stránky) a nltk nabízelo snazší odstranění interpunkce. Segmentace vět byla až na pár případů obstojná.

Zároveň se špatně manipulovalo s výsledky: výstup lemmatizace neobsahoval pouhou lemmatizaci, ale i snahu o desambiguaci a případné poznámky vztahující se např. k NER rozdělení. Podobně tag obsahoval veškeré dostupné morfologické informace, což bylo přínosné, ale pro rychlou manipulaci bylo třeba parsovat jednotlivé části tagu.

Dále poskytuje rozsáhlou podporu pro morfologickou desambiguaci, díky obsáhlým informacím z tagu (rod, číslo, pád, čas, ...) a i při lemmatizaci odlišuje číselně homonymní slova s jiným významem a někdy v poznámce vysvětluje speciální varianty. CorPy je dostupná primárně v češtině a v angličtině.

Simplemma byla rychlá a výsledek lemmatizace byl pouze nevýrazně horší než CorPy, zvládala tokenizaci, lemmatizaci a několik doplňkových funkcí. Velkou výhodou bylo kombinování jazyků i jejich velká nabídka. Strojově užitečná funkce bylo např. odhadnutí poměrů jazyků v textu, a v rámci testování se snadno dalo oddělit slova, která nebyla obsažena v lemmatickém listu, takže při manuálním procházení nepřekážela slova, o které se nelemmatizovala se zahrnutím neohebných slovních druhů, které se již nacházela v základním tvaru.

Tokenizátor Stanza fungoval správně. Segmentace nebyla dokonalá, ale o něco lepší než s CorPy, oproti které některé problematické případy zvládala. Lemmatizace probíhala jinak než s předešlými nástroji, občasná špatná lemmatizace často znamenala under stemming. Při porovnání na textech Stanza byla z nástrojů v lemmatizaci nejpřesnější.

Základní tagger získal pouze slovní druh, což bylo poměrně přesné, rozšířený se strukturou podobal CorPy, ale některé informace byly částečně chybné. Částečně

odstraňoval morfologickou desambiguaci. Podle potřeby šlo (snadněji než z CorPy) získat požadovaný výstup a zobrazovat pouze požadované informace.

Polygot měl tři kategorie pro NER, obecně rozdělil reálné jmenné entity do správných kategorií. Zatímco většina nástrojů má sklon být přísnější a odmítat i validní slova, tento má naopak sklon být benevolentní a nemá tedy tak velkou kontrolu nad výběrem *pouze* správných slov.

## 7 Závěry a doporučení

V práci byly popsány základní principy nutné pro přípravu jazyka pro strojové zpracování – tokenizaci, segmentaci, lemmatizaci, stematizaci, part of speech tagování, named entity recognition, normalizace textu a další. Výčet neobsahoval všechny techniky – slouží jako úvod pro seznámení se s NLP.

Praktická část se zaměřila na aplikaci teoretických poznatků (z hlediska programátorského a filologického) a srovnání konkrétních veřejně dostupných nástrojů pro zpracování přirozeného českého jazyka. Na testovacích textech byl prováděn kvantitativní výzkum, na krátkých textech/větech kvalitativní výzkum, který pomohl rozlišit výhody a nevýhody jednotlivých nástrojů a jejich možnosti využití v praxi. Přestože žádný nástroj nebyl bez chyby, výsledky byly uspokojivé a výchyly nikdy hrubě neporušovaly pravidla jazyka – špatný výsledek nevycházel ze špatného postupu, ale spíš z neošetření výjimek.

Úlohy, které v nástrojích nebyly dostupné pro češtinu, byly dobře zmapované pro jiné jazyky, některé představovaly dobrý podklad pro vyvinutí algoritmu a nebyly postihnuté komplikovaností českého jazyka.

Osobně pro tokenizaci doporučuji NLTK, segmentaci Stanza, odstranění stop slov DESAM, lemmatizaci pro analýzu dlouhého textu simplemma (jinak CorPy), pro desambiguaci CorPy, POS tagování CorPy, NER CorPy (pro základ algoritmu Polygot).

V budoucnosti se dá počítat s dalším vývinem nástrojů a jejich pokrytí i pro český jazyk. Podobně se i LLM budou zdokonalovat a možná nahradí v některých úkolech existující modely.

## 8 Seznam použité literatury

- 1 - Mansi, Agarwal. (2019). An Overview of Natural Language Processing. *International Journal for Research in Applied Science and Engineering Technology*, 7(5):2811-2813. doi: 10.22214/IJRASET.2019.5462
- 2 - Vladimír Petkevič (2017): *VĚTNÁ SEGMENTACE*. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), *CzechEncy - Nový encyklopedický slovník češtiny*. [https://www.czechency.org/slovník/VĚTNÁ\\_SEGMENTACE](https://www.czechency.org/slovník/VĚTNÁ_SEGMENTACE)
- 3 - Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2020). Fast wordpiece tokenization. arXiv
- 4 - Ganesan, K. (2023, March 16). *What are stop words?*. Kavita Ganesan, PhD. <https://kavita-ganesan.com/what-are-stop-words/>
- 5 - Aliero, A., Bashir, S., Aliyu, H., Tafida, A., Kangiwa, B., & Dankolo, N. (2023). Systematic Review on Text Normalization Techniques and its Approach to Non-Standard Words. *International Journal of Computer Applications*, 185, 975-8887.
- 6 - Khyani, D., Siddhartha, B. S., Niveditha, N. M., & Divya, B. M. (2021). An interpretation of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*, 22(10), 350-357.
- 7 - Zhao, Zhen. (2022). XSTEM: An exemplar-based stemming algorithm. doi: 10.48550/arxiv.2205.04355
- 8 - Bevilacqua, M., Pasini, T., Raganato, A., & Navigli, R. (2021, August). Recent trends in word sense disambiguation: A survey. In *International Joint Conference on Artificial Intelligence* (pp. 4330-4338). International Joint Conference on Artificial Intelligence, Inc.
- 9 - Lv, C., Liu, H., Dong, Y. et al. (2016, July 21). *Corpus based part-of-speech tagging*. *Int J Speech Technol* 19, 647–654. <https://doi.org/10.1007/s10772-016-9356-2>
- 10 - Michal Křen (2017): TAG. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), *CzechEncy - Nový encyklopedický slovník češtiny*. <https://www.czechency.org/slovník/TAG>
- 11 - Vladimír Petkevič (2017): TAGGER. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), *CzechEncy - Nový encyklopedický slovník češtiny*. <https://www.czechency.org/slovník/TAGGER>
- 12 - Cvrček, Václav - Richterová, Olga (eds) (2022, April 18). seznamy:tagy. Příručka ČNK. <http://wiki.korpus.cz/doku.php?id=seznamy:tagy&rev=1650299545>.

- 13 - Simeonova, L., Simov, K.I., Osenova, P.N., & Nakov, P. (2019). A Morpho-Syntactically Informed LSTM-CRF Model for Named Entity Recognition. *Recent Advances in Natural Language Processing*.
- 14 - Jana Straková (2017): ROZPOZNÁVÁNÍ POJMENOVANÝCH ENTIT. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), *CzechEncy - Nový encyklopedický slovník češtiny*. [https://www.czechency.org/slovník/ROZPOZNÁVÁNÍ POJMENOVANÝCH ENTIT](https://www.czechency.org/slovník/ROZPOZNÁVÁNÍ_POJMENOVANÝCH_ENTIT)
- 15 - Buttrick, N. (2024). Studying large language models as compression algorithms for human culture. *Trends in Cognitive Sciences*, 28(3), 187-189. ISSN 1364-6613. <https://www.sciencedirect.com/science/article/pii/S1364661324000019>
- 16 - Kjell, O.N.E., Kjell, K., Schwartz, H.A. (2024). Beyond rating scales: With targeted evaluation, large language models are poised for psychological assessment. *Psychiatry Research*, 333, 115667. ISSN 0165-1781. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0165178123006170>
- 17 - Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Association for Computational Linguistics (ACL) System Demonstrations*. 2020.
- 18 - Barbaresi A. (2021). Simplemma: a simple multilingual lemmatizer for Python [Computer software] (v1.0.5). Berlin, Germany: Berlin-Brandenburg Academy of Sciences. Dostupné z: <https://github.com/adbar/simplemma> DOI: 10.5281/zenodo.4673264 (Aktualizováno: 20. ledna 2023)
- 19 - Al-Rfou, R., Kulkarni, V., Perozzi, B., & Skiena, S. (2015). Polyglot-NER: Massive Multilingual Named Entity Recognition. *Proceedings of the 2015 SIAM International Conference on Data Mining*, Vancouver, British Columbia, Canada, April 30 - May 2, 2015. SIAM.
- 20 - OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. <https://chat.openai.com/chat>
- 21 - Petr Karlík (2017): DETERMINACE. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), *CzechEncy - Nový encyklopedický slovník češtiny*. <https://www.czechency.org/slovník/DETERMINACE>
- 22 - Friedman, V. A. (1998, July 20). VLACH | History, Language & Culture. *Encyclopedia Britannica*. <https://www.britannica.com/topic/Vlach>

## 9 Zadání práce



Univerzita Hradec Králové  
Fakulta informatiky a managementu

### Zadání bakalářské práce

**Autor:** Vlasta Matějková

**Studium:** I2100243

**Studijní program:** B1802 Aplikovaná informatika

**Studijní obor:** Aplikovaná informatika

**Název bakalářské práce:** **Nástroje pro přípravu českého textu na strojové zpracování**

**Název bakalářské práce AJ:** Tools for preprocessing Czech text for machine processing

#### **Cíl, metody, literatura, předpoklady:**

- 1/ Specifika českého textu vzhledem k angličtině, tokenizace, lemmatizace, desambiguace a part-of-speech tagování
- 2/ Přehled dostupných nástrojů pro uvedené činnosti
- 3/ testování jejich schopností + porovnání na reálných textech

<https://www.korpus.cz/>

<https://lindat.mff.cuni.cz/services/translation/>

<http://utkl.ff.cuni.cz/cz/linglink.html>

<https://oltk.upol.cz/en/>

<https://stanfordlp.github.io/>

<https://nltk.org/>

**Zadávací pracoviště:** Katedra informatiky a kvantitativních metod,  
Fakulta informatiky a managementu

**Vedoucí práce:** Mgr. Jiří Haviger, Ph.D.

**Datum zadání závěrečné práce:** 26.1.2021