Automatická morfologická disambiguace korpusů řady SYN: spolupráce lingvistické introspekce a strojového učení[1]

Vladimír PETKEVIČ | Ústav lingvistiky FF UK
Tomáš JELÍNEK | Ústav lingvistiky FF UK

Automatic morphological disambiguation of the SYN series corpora: combination of linguistic introspection and machine learning

The paper deals with the current method of automatic morphological tagging of corpora of contemporary Czech of the SYN series and other corpora within the Czech National Corpus. From SYN2020 onwards, the corpora are annotated on the basis of an improved concept. The paper starts with a brief description of newly introduced features concerning tokenization and lemmatization (introduction of sublemmata), and of the tagging of multiword tokens (i.e. compound forms like *abys*, *cos*); a new attribute, verbtag, is also presented. Then the successive steps of the entire annotation process are described. The core of the paper provides a detailed description of the procedure of automatic morphological disambiguation, namely the combination of two methodologically different approaches: the LanGr system of linguistically motivated disambiguation rules based on introspection, and the MorphoDiTa tool based on machine learning – we call this combination a hybrid approach. Particular emphasis is laid on the detailed characterization of the LanGr system, primarily on compiling specific lists of bigrams and trigrams of lemmas and forms labeled as global identifiers and on using these identifiers in disambiguation rules. The success rate of the hybrid system compared to the success rate of the stand-alone MorphoDiTa system is also presented and plans are briefly outlined for further development of our hybrid morphological tagging approach.

Keywords: annotation process, automatic morphological tagging, corpora of the SYN series, global identifiers, LanGr as a programming language, LanGr as a rule system, linguistically motivated rules, machine learning, MorphoDiTa, morphological disambiguation

Klíčová slova: automatické morfologické značkování, globální identifikátory, korpusy řady SYN, LanGr jako programovací jazyk, LanGr jako systém pravidel, lingvisticky motivovaná pravidla, morfologická disambiguace, MorphoDiTa, proces anotace, strojové učení

[1] Tento článek vznikl při realizaci projektu Český národní korpus (LM2023044) financovaného Ministerstvem školství, mládeže a tělovýchovy v rámci aktivity Projekty velkých infrastruktur pro VaVaI.

1 Úvod

Článek se zabývá aktuálně užívaným způsobem morfologické disambiguace korpusů současné češtiny řady SYN, počínaje žánrově reprezentativním korpusem SYN2020 (dalšími korpusy jsou například žánrově nereprezentativní korpusy SYN(v9 až v13), Online 2) v rámci projektu Český národní korpus (ČNK), s důrazem na disambiguaci prováděnou systémem lingvisticky motivovaných pravidel a na spolupráci tohoto systému s nástrojem MorphoDiTa založeným na strojovém učení. V oddílu 2 jsou velmi stručně popsány některé novinky v koncepci lingvistické anotace (značkování) korpusů a poté proces této anotace. Hlavní oddíl 3 se zaměřuje na klíčovou fázi v tomto procesu: morfologickou disambiguaci. Je v něm charakterizována tzv. hybridní metoda morfologické disambiguace: spolupráce (i) systému disambiguačních pravidel založených na introspekci s (ii) anotačním nástrojem MorphoDiTa, založeným na strojovém učení. Podrobně je přitom popsán zejména systém disambiguačních pravidel, zvaný LanGr. Je také uvedena aktuální úspěšnost anotace z různých hledisek včetně srovnání úspěšnosti metody hybridní s úspěšností samotného nástroje MorphoDiTa. V závěrečném oddílu 4 jsou představeny plány dalšího rozvoje systému LanGr a celého hybridního systému.

2 Lingvistická anotace korpusu SYN2020 a pozdějších korpusů současné češtiny

Lingvistická anotace korpusu SYN2020 (Křen et al., 2020)[2] a mladších korpusů řady SYN (SYN verze 9 až 13)[3] i dalších korpusů současné češtiny (např. Online 2)[4] se v některých ohledech liší od anotace starších korpusů současné češtiny (SYN2000, SYN2005, SYN2010, SYN2015; SYN v3 – SYN v8),[5] změny jsou přitom podrobně popsány v článku Křivana a Šindlerové (2022) a Jelínka et al. (2021).[6] V zájmu snazšího porozumění hlavnímu tématu článku – morfologické disambiguaci (odd. 3) a jejímu zapojení do procesu lingvistické anotace – se v tomto oddílu stručně zmíníme o těchto změnách: k základnímu morfologickému tagu s 15 pozicemi[7] přibyl další atribut: **verbtag** s 6 pozicemi (§ 2.1), v němž jsou obsaženy hodnoty morfologických kategorií slovesných tvarů. Dále nově pracujeme s tzv. **agregáty** (§ 2.2)

---

2 Srov. i <https://wiki.korpus.cz/doku.php/cnk:syn2020>.

3 Viz <https://wiki.korpus.cz/doku.php/cnk:syn:verze9>, <https://wiki.korpus.cz/doku.php/cnk:syn:verze13>.

4 Viz <https://wiki.korpus.cz/doku.php/cnk:online:gen2>.

5 Viz <https://wiki.korpus.cz/doku.php/cnk:syn2000>, <https://wiki.korpus.cz/doku.php/cnk:syn2005>, <https://wiki.korpus.cz/doku.php/cnk:syn2010>, <https://wiki.korpus.cz/doku.php/cnk:syn2015>, <https://wiki.korpus.cz/doku.php/cnk:syn:verze3>, <https://wiki.korpus.cz/doku.php/cnk:syn:verze8>.

6 Srov. i <https://wiki.korpus.cz/doku.php/cnk:syn2020>.

7 Viz <https://wiki.korpus.cz/doku.php/seznamy:tagy>.

a kromě lemmat používáme i **sublemmata** (§ 2.3). V závěru oddílu pak popisujeme jednotlivé kroky v procesu lingvistické anotace (§ 2.4).

2.1 Verbtag

Verbtag je značka o 6 pozicích obsahující morfologické informace o slovesném tvaru složeném (*pracovala jsem*) i jednoduchém (*pracuji*). Ve verbtagu

(i) se odlišuje pomocné sloveso (tak je anotováno pouze sloveso *být*[8]) od slovesa plnovýznamového.[9] U pomocného slovesa se ve verbtagu anotuje pouze fakt, že je to pomocné sloveso (verbtag=A-----); jeho morfologické vlastnosti (čas, číslo, jmenný rod, způsob) se anotují ve verbtagu s ním spjatého významového slovesa;

(ii) jsou u každého plnovýznamového tvaru slovesa uvedeny hodnoty pěti morfosyntaktických a morfologických kategorií: způsobu, slovesného rodu, osoby, čísla a času, platné pro celý jednoduchý i složený slovesný tvar.[10] Ve verbtagu se rovněž anotují deverbativní adjektiva utvořená z pasivních slovesných příčestí – jejich lemma je v základním tvaru zakončeno na *-ný*/*-tý*.[11]

Pro zavedení verbtagu jsme se rozhodli proto, abychom uživatelům poskytli více morfosyntaktických informací o slovních tvarech. Zatímco tag popisuje jednotlivé tvary, verbtag vychází z kontextu celé věty a tyto informace více interpretuje. U infinitivů, které jsou součástí složeného futura, nebo u minulého i trpného příčestí se například v tagu neuvádí osoba, složený tvar ale osobu vyjadřuje a ve verbtagu ji najdeme. V některých případech může být hodnota v tagu a ve verbtagu v rozporu, například ve větě „Chraň Bůh!“ je v tagu druhá osoba, zato ve verbtagu osoba třetí.

---

[8] Vzhledem ke značné obtížnosti disambiguace se u sloves *mít* a *dostat* nerozlišuje, zda jsou pomocná, nebo plnovýznamová.

[9] U slovesa *být* se tak rozliší, zda je pomocné (spolupodílí se s plnovýznamovým slovesem na vyjádření slovesného významu tak, že vyjadřuje hodnoty některých morfologických kategorií složeného slovesného tvaru), nebo plnovýznamové čili existenční a sponové v širokém smyslu. Toto rozlišení usnadňuje např. zkoumání vzájemného postavení pomocného *být* a plnovýznamové komponenty slovesného tvaru v klauzi a jeho popis v mluvnicích češtiny.

[10] Viz <https://wiki.korpus.cz/doku.php/seznamy:verbtagy>.

[11] Důvodem značkování těchto deverbativ (a nikoli např. substantiv či jiných deverbativních adjektiv) je jejich sémantická blízkost pasivním příčestím, která jsou ve verbtagu značkována. Navíc deverbativní adjektiva utvořená z přítomných a minulých přechodníků již jsou značkována v tagu (*konající*: tag=AG…, *přeživší*: tag=AM…). Deverbativní substantiva zakončená v základním tvaru na *-ní*, *-tí* jsou již příliš vzdálena slovesům, obecněji predikativnosti, některá z nich jsou nadto lexikalizována (*brnění*).

2.2 Agregáty

Nově jsou zavedena vícenásobná lemmata a značky pro zvláštní skupiny slov – *agregáty*. Jsou to slova psaná jako jedno ortografické slovo, ale ze syntaktického pohledu nebo z hlediska určování hodnot gramatických kategorií se chovají jako slova dvě (výjimečně tři). Patří mezi ně

(i) kondicionálové spojky (*aby*, *kdyby…*);

(ii) spojení slov, zvláště *l*-ového slovesného příčestí se slovesným příklonným *-s* (*připravovalas*, *viděls*, *komus*, *vždyť s*, *včeras*…);

(iii) spojení předložek s některými zájmeny osobními a tázacími (*naň*, *očpak*, *zač*…), případně kombinace s typem (ii) (*načs*).

U každého z takovýchto tvarů se uvádějí současně dvě/tři lemmata, sublemmata (viz § 2.3), tagy a verbtagy podle jejich jednotlivých komponent. Uživatel tak může v korpusu vyhledávat a zahrnovat do statistik jak agregované formy (*kdybyste*, *přišels*), tak jejich interpretace: lemmata komponent, z nichž se agregované formy skládají (*kdyby*, *být*; *přijít*, *být*), a jejich tagy a verbtagy.[12]

2.3 Sublemmata

V korpusu SYN2020 a mladších korpusech je zavedena tzv. dvojúrovňová lemmatizace: každý tvar má nyní kromě atributu lemma přiřazen i atribut *sublemma*. Lemma tak jako dříve sdružuje více variant jednoho slova (např. lemma *filozofie* reprezentuje všechny tvary s kořenem *filozof-* i *filosof-*), avšak nově zavedená sublemmata vymezují podskupiny tvarů podle této variantnosti, např. sublemma *filozofie* reprezentuje jen tvary s kořenem *filozof*, sublemma *filosofie* jen tvary s kořenem *filosof*.[13]

2.4 Proces lingvistické anotace

Lingvistická anotace korpusů se provádí plně automaticky, přičemž zásadním způsobem se na ní podílí nástroj MorphoDiTa manželů Strakových, vyvinutý v Ústavu formální a aplikované lingvistiky MFF UK.[14] Popíšeme nyní jednotlivé fáze lingvistické anotace vstupního textu počínaje tokenizací a konče morfologicky plně disambiguovaným textem. Anotace je rozdělena do těchto na sebe navazujících kroků (srov. Jelínek et al., 2021, s. 6–9).[15]

---

[12] Podrobnosti viz na adrese <https://wiki.korpus.cz/doku.php/cnk:syn2020:agregat>.
[13] Podrobnosti viz na adrese <https://wiki.korpus.cz/doku.php/cnk:syn2020:lemmatizace>.
[14] Viz <https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-43CD-0>.
[15] Viz i <https://wiki.korpus.cz/doku.php/cnk:syn2020:automaticka_anotace>.

2.4.1 Tokenizace a větná segmentace

Základní tokenizaci a větnou segmentaci provádí nástroj MorphoDiTa (Straková et al., 2014), založený na strojovém učení. Při tokenizaci jsou na samostatné tokeny rozdělena ortografická slova obsahující interpunkci, například *česko-německý*, *západo-východní*, *psycho-sociální*, *e-mail*, *O'Brien*, *S'-čchuan*[16] (v kroku 2.4.4 níže jsou však opět spojena v jeden token); kompozita neobsahující interpunkci (např. *černomořský*, *pozdněbarokní*) rozdělována nejsou a i agregáty (např. *abyste*, *odešels*) zůstávají nerozdělené.

2.4.2 Morfologická analýza

Morfologickou analýzu všech tokenů včetně interpunkce realizuje podle pojetí tzv. „pražské" morfologické analýzy (viz Hajič, 2004) rovněž MorphoDiTa (Straková et al., 2014). Využívá přitom verzi morfologického slovníku upravenou pro projekt ČNK. Ta vychází z pracovní verze slovníku MorfFlex[17] (Štěpánková et al., 2020) z července 2020 a zahrnuje některé změny v tagsetu (viz Hajič et al., 2020). Tato verze byla v projektu ČNK dále upravena, především v souvislosti se zavedením víceslovných tokenů/agregátů a rozlišením lemmat a sublemmat, dále byly přehodnoceny slovní druhy predikativ, zkratek a zvláštních jmenných tvarů adjektiv a druhy číslovek. Slovník byl také doplněn o velké množství lemmat, aby se zlepšilo pokrytí publikovaných korpusů. Oproti převzaté verzi bylo do vydání korpusu SYN v13 (prosinec 2024; Křen et al., 2024) doplněno 504 000 lemmat.

Výsledkem morfologické analýzy je přiřazení množiny dvojic (sublemma, tag) každému vstupnímu tokenu.

2.4.3 Oprava větné segmentace

V této fázi se opravuje větná segmentace provedená v kroku 2.4.1, přičemž se využívá nedisambiguovaná morfologická anotace z kroku 2.4.2. Opravy se týkají hlavně segmentace, kterou znesnadňuje homonymie teček, jimiž mohou být zakončeny jak zkratky, tak věty (*nebo využijte poštovní adresu Deník,* ***nám. Přemysla Otakara*** *II.* vs. *pak přišli přátelé* ***k nám. Přemysla*** *jsme dali do dětského pokoje*). V této fázi se opravuje přibližně třicet nejčastějších typů chyb v segmentaci, chybně rozdělené věty se spojují, chybně spojené věty se rozdělují. V průměru dojde v této fázi k úpravě segmentace u 1,8 % vět.

---

[16] Například řetězec *e-mail* je rozdělen na tři tokeny: *e* / - / *mail*; řetězec *S'-čchuan* je rozdělen dokonce na čtyři tokeny: *S* / *'* / - / *čchuan*.

[17] Jeho aktuální verzí je MorfFlex CZ 2.0, viz <https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3186>.

2.4.4 Úprava tokenizace

Tokeny vybraných typů, které byly rozděleny v první fázi tokenizace (§ 2.4.1), například *S* / *'* / - / *čchuan*, jsou opět spojeny, například *S'-čchuan* (jiné řetězce zůstávají rozdělené, například *Brožová* / - / *Polednová*). Dále jsou na základě seznamu slov (*abyste*, *nač*, *ses*, *včeras*…) a morfologické analýzy (*otevřels*…) identifikovány agregáty a rozděleny na jednotlivé tokeny, přičemž každému z nich je přiřazena množina dvojic (sublemma, tag). Během disambiguace se tedy s jednotlivými složkami agregátů pracuje samostatně, až na konci anotace jsou zpětně spojeny do jednoho tokenu. Díky tomuto postupu dosahujeme lepší úspěšnosti v přiřazení tagů a verbtagů, umožňuje to také syntaktickou anotaci jednotlivých syntaktických slov (například u agregátu *ses* ve větě *Neměl ses tomu smát* je tvar *se* součástí zvratného slovesa *smát se*, zatímco *s*, klitická forma slovesa *být*, spoluutváří tvar druhé osoby minulého času s tvarem *neměl*).

2.4.5 Nahrazení sublemmat lemmaty

V této fázi jsou sublemmata nahrazena lemmaty na základě příslušných seznamů. Ve většině případů stačí pro přiřazení lemmatu samotné sublemma, v jednotlivých případech se navíc rozhoduje podle morfologické značky (tagu), například sublemmatu *vylízat* se přiřadí lemma *vylézat* či *vylízat* podle vidu, po této fázi tedy bude mít např. tvar *vylízáme* dvě lemmata. Informace o sublemmatech zůstává uchována v datech uložených po předchozí fázi, později je využita pro zpětné doplnění sublemmat. Práce s lemmaty má výhodu zvláště pro tagger založený na hlubokém strojovém učení, díky nižší variantnosti lemmat oproti sublemmatům má takto vyšší úspěšnost značkování.

2.4.6 Přidání verbtagů

Na základě slovního druhu (1. pozice v tagu), u sloves i na základě slovního poddruhu (2. pozice v tagu) a dalších pozic jsou ke každému tokenu přidány jeho verbtagy. U deverbativních adjektiv utvořených z pasivních slovesných příčestí se verbtag doplňuje na základě seznamu. Verbtag není zařazen do slovníku, protože by zbytečně znásobil počty položek ve slovníku a zpomalil jeho využívání.

2.4.7 Morfologická disambiguace

Při morfologické disambiguaci se odstraňuje homonymie, cílem disambiguace je tedy u každého tokenu jednoznačně určit náležitou trojici (lemma, tag, verbtag) v daném kontextu (viz podrobně odd. 3).

2.4.8 Doplnění sublemmat

U každého tokenu se nyní již jednoznačnému lemmatu přiřadí sublemma podle příslušných seznamů a dat uložených po fázi 2.4.4; systém ověří, že určené dvojici (lemma, tag) odpovídá jedna dvojice (sublemma, tag) z nabídky po fázi 2.4.4. Tvaru *vylízá* se například přiřadí sublemma *vylízat* bez ohledu na určené lemma (*vylézat* či *vylízat*) a tag, tvaru *jeskyň* s lemmatem *jeskyně* se podle pádu přiřadí buď sublemma *jeskyně* (pro genitiv plurálu), nebo sublemma *jeskyň* (pro nominativ či akuzativ singuláru), například *pravěkou jeskyň* (velmi zřídka).

2.4.9 Zpracování agregátů

Rozdělené agregáty se zpětně spojí v jeden token (*a* + *byste* → *abyste*). Spojenému agregátu však zůstávají přiřazeny interpretace jeho složek v podobě (lemma, sublemma, tag, verbtag). Uživatel tak bude moci pracovat s celou agregovanou formou (*abyste*) i s interpretacemi jejích složek.

2.4.10 Závěrečná oprava disambiguace

Na základě namátkově zjištěných chyb se pomocí speciálního programu a pro něj vytvořeného seznamu opraví chyby v disambiguaci. Tato fáze je zařazena především proto, že například zpracování téměř šestimiliardového korpusu trvá týdny, znovu spouštět celý proces po objevení chyby tedy nelze.

Výsledkem celého procesu je text rozdělený na věty a tokeny, každému tokenu je přiřazena jediná čtveřice (lemma, sublemma, tag, verbtag), u agregátů je taková čtveřice přiřazena každé části.

2.4.11 Příklad zpracované věty

Celý proces zpracování věty předvedeme na větě (1), u níž předpokládáme, že je již výsledkem větné segmentace:

(1) *Chci-li, abyste napsal e-mail o filosofii, učiňte tak, prosím.*

2.4.11.1 Tokenizace a větná segmentace

Větu identifikuje větná segmentace a tokenizace ji nadrobno rozdělí na jednotlivé tokeny (oddělujeme je lomítkem):

„ / Chci / - / li / , / abyste / napsal / e / - / mail / o / filosofii / , / učiňte / tak / , / prosím / . / “

2.4.11.2 Morfologická analýza přiřadí každému tokenu všechny jeho morfologické interpretace, z nichž každá má podobu dvojice (sublemma, tag);[18] níže píšeme jednotlivou dvojici vždy na nový řádek:

```
„           subl="          tag=Z:-------------
Chci        subl=chtít      tag=VB-S---1P-AAI--
-           subl=-          tag=Z0-------------
li          subl=li         tag=BN-------------
,           subl=,          tag=Z:-------------
abyste      subl=aby        tag=J,-----------e-[19]
napsal      subl=napsat     tag=VpIS----R-AAP--
            subl=napsat     tag=VpMS----R-AAP--
e           subl=e          tag=BN-------------
            subl=e          tag=II-------------
            subl=e          tag=S2--------A----
-           subl=-          tag=Z0-------------
mail        subl=mail       tag=NNIS1-----A----
            subl=mail       tag=NNIS4-----A----
o           subl=o          tag=BA-------------
            subl=o          tag=F%-------------
            subl=o          tag=RR---4---------
            subl=o          tag=RR---6---------
filosofii   subl=filosofie  tag=NNFS3-----A----
            subl=filosofie  tag=NNFS4-----A----
            subl=filosofie  tag=NNFS6-----A----
,           subl=,          tag=Z:-------------
učiňte      subl=učinit     tag=Vi-P---2--A-P--
tak         subl=tak        tag=Db-------------
            subl=tak        tag=J^-------------
,           subl=,          tag=Z:-------------
prosím      subl=prosit     tag=VB-S---1P-AAI--
.           subl=.          tag=Z:-------------
“           subl="          tag=Z:-------------
```

2.4.11.3 Úprava tokenizace

V této fázi jsou opět spojeny některé tokeny (*e-mail*, *O’Brien*, *S’-čchuan*) a naopak agregáty rozděleny na jednotlivé tokeny (*abyste* → *a* / *byste*):

```
„           subl="          tag=Z:-------------
Chci        subl=chtít      tag=VB-S---1P-AAI--
```

[18] Morfologicky nehomonymním tokenům je přiřazena jediná taková dvojice.

[19] Agregát *abyste* je dosud nerozdělený. Hodnota *e* na 14. pozici vyjadřuje 2. osobu plurálu; takto se ve slovníku MorfFlex značkují agregáty, viz <https://ufal.mff.cuni.cz/pdt-c>.

```
-                       subl=-          tag=Z0-------------
li                      subl=li         tag=BN-------------
,                       subl=,          tag=Z:-------------
a       forma=a         subl=aby        tag=J,-------------
byste   forma=byste     subl=být        tag=Vc-P---2----I--
napsal                  subl=napsat     tag=VpIS----R-AAP--
                        subl=napsat     tag=VpMS----R-AAP--
e-mail                  subl=e-mail     tag=NNIS1-----A----
                        subl=e-mail     tag=NNIS4-----A----
o                       subl=o          tag=BA-------------
                        subl=o          tag=F%-------------
                        subl=o          tag=RR---4---------
                        subl=o          tag=RR---6---------
filosofii               subl=filosofie  tag=NNFS3-----A----
                        subl=filosofie  tag=NNFS4-----A----
                        subl=filosofie  tag=NNFS6-----A----
,                       subl=,          tag=Z:-------------
učiňte                  subl=učinit     tag=Vi-P---2--A-P--
tak                     subl=tak        tag=Db-------------
                        subl=tak        tag=J^-------------
,                       subl=,          tag=Z:-------------
prosím                  subl=prosit     tag=VB-S---1P-AAI--
.                       subl=.          tag=Z:-------------
“                       subl="          tag=Z:-------------
```

2.4.11.4 Nahrazení sublemmat lemmaty

V této fázi jsou sublemmata nahrazena lemmaty, např. token *filosofii* dostává lemma *filozofie*:

```
„           lem="           tag=Z:-------------
Chci        lem=chtít       tag=VB-S---1P-AAI--
-           lem=-           tag=Z0-------------
li          lem=li          tag=BN-------------
,           lem=,           tag=Z:-------------
a           lem=aby         tag=J,-------------
byste       lem=být         tag=Vc-P---2----I--
napsal      lem=napsat      tag=VpIS----R-AAP--
            lem=napsat      tag=VpMS----R-AAP--
e-mail      lem=e-mail      tag=NNIS1-----A----
            lem=e-mail      tag=NNIS4-----A----
o           lem=o           tag=BA-------------
            lem=o           tag=F%-------------
            lem=o           tag=RR---4---------
            lem=o           tag=RR---6---------
filosofii   lem=filozofie   tag=NNFS3-----A----
            lem=filozofie   tag=NNFS4-----A----
```

```
                lem=filozofie     tag=NNFS6-----A----
,               lem=,             tag=Z:-------------
učiňte          lem=učinit        tag=Vi-P---2--A-P--
tak             lem=tak           tag=Db-------------
                lem=tak           tag=J^-------------
,               lem=,             tag=Z:-------------
prosím          lem=prosit        tag=VB-S---1P-AAI--
.               lem=.             tag=Z:-------------
“               lem="             tag=Z:-------------
```

2.4.11.5 Přidání verbtagů

Ke každé trojici (lemma, sublemma, tag) se přidává verbtag. Verbtag má neprázdné hodnoty jen u tvarů sloves a pasivních deverbativních adjektiv, u ostatních tokenů je irelevantní (verbtag=------). Některé slovesné tvary obdrží větší množství hodnot verbtagu, např. *l*-ovým příčestím plnovýznamových sloves je přiřazeno 28 různých hodnot (např. *napsal* níže), z nichž náležitá hodnota se stanoví na základě kontextu procesem disambiguace (v příkladu níže je zvýrazněna tučně: **verbtag=VCA2vP**):

```
„               lem="             tag=Z:-------------     verbtag=------
Chci            lem=chtít         tag=VB-S---1P-AAI-      verbtag=VDA1SP
-               lem=-             tag=Z0-------------     verbtag=------
li              lem=li            tag=BN-------------     verbtag=------
,               lem=,             tag=Z:-------------     verbtag=------
a               forma=a           tag=J,-------------     verbtag=------
byste           forma=byste       tag=Vc-P---2----I--     verbtag=A-----[20]
napsal          lem=napsat        tag=VpIS----R-AAP--     verbtag=VCA1SP
……
                lem=napsat        tag=VpIS----R-AAP--     verbtag=VCA2vP
……
                lem=napsat        tag=VpIS----R-AAP--     verbtag=VDA3SR
e-mail          lem=e-mail        tag=NNIS1-----A----     verbtag=------
                lem=e-mail        tag=NNIS4-----A----     verbtag=------
o               lem=o             tag=BA-------------     verbtag=------
                lem=o             tag=F%-------------     verbtag=------
                lem=o             tag=RR---4---------     verbtag=------
                lem=o             tag=RR---6---------     verbtag=------
filosofii       lem=filozofie     tag=NNFS3-----A----     verbtag=------
                lem=filozofie     tag=NNFS4-----A----     verbtag=------
                lem=filozofie     tag=NNFS6-----A----     verbtag=------
```

[20] Připomínáme (viz začátek § 2.1), že u pomocného slovesa se ve verbtagu anotuje pouze fakt, že je to pomocné sloveso. Jeho morfologické vlastnosti (čas, číslo, jmenný rod, způsob) se anotují ve verbtagu s ním spjatého významového slovesa (zde tokenu *napsal*).

```
,           lem=,           tag=Z:-------------     verbtag=------
učiňte      lem=učinit      tag=Vi-P---2--A-P--     verbtag=VIA2P-
tak         lem=tak         tag=Db-------------     verbtag=------
            lem=tak         tag=J^-------------     verbtag=------
,           lem=,           tag=Z:-------------     verbtag=------
prosím      lem=prosit      tag=VB-S---1P-AAI--     verbtag=VDA1SP
.           lem=.           tag=Z:-------------     verbtag=------
“           lem="           tag=Z:-------------     verbtag=------
```

2.4.11.6 Morfologická disambiguace

Morfologická disambiguace pracuje pouze s lemmaty, tagy a verbtagy, nezabývá se sublemmaty. Jejím výsledkem je přiřazení jediné trojice (lemma, tag, verbtag) každému tokenu a každé složce agregátu:

```
„           lem="           tag=Z:-------------     verbtag=------
Chci        lem=chtít       tag=VB-S---1P-AAI--     verbtag=VDA1SP
-           lem=-           tag=Z0-------------     verbtag=------
li          lem=li          tag=BN-------------     verbtag=------
,           lem=,           tag=Z:-------------     verbtag=------
a           forma=a         tag=J,-------------     verbtag=------
byste       forma=byste     tag=Vc-P---2----I--     verbtag=A-----
napsal      lem=napsat      tag=VpMS----R-AAP--     verbtag=VCA2vP
e-mail      lem=e-mail      tag=NNIS4-----A----     verbtag=------
o           lem=o           tag=RR--6----------     verbtag=------
filosofii   lem=filozofie   tag=NNFS6-----A----     verbtag=------
,           lem=,           tag=Z:-------------     verbtag=------
učiňte      lem=učinit      tag=Vi-P---2--A-P--     verbtag=VIA2P-
tak         lem=tak         tag=Db-------------     verbtag=------
,           lem=,           tag=Z:-------------     verbtag=------
prosím      lem=prosit      tag=VB-S---1P-AAI--     verbtag=VDA1SP
.           lem=.           tag=Z:-------------     verbtag=------
“           lem=“           tag=Z:-------------     verbtag=------
```

2.4.11.7 Přidání sublemmat

Každému tokenu se zpětně přiřadí jeho sublemma, přičemž se tu využije výsledek fáze (2.4.11.2), jinak se nic nemění:

```
„       lem="        subl="       tag=Z:-------------   verbtag=------
Chci    lem=chtít    subl=chtít   tag=VB-S---1P-AAI–   verbtag=VDA1SP
-       lem=-        subl=-       tag=Z0-------------   verbtag=------
li      lem=li       subl=li      tag=BN-------------   verbtag=------
,       lem=,        subl=,       tag=Z:-------------   verbtag=------
a       forma=a      subl=aby     tag=J,-------------   verbtag=------
byste   forma=byste  subl=být     tag=Vc-P---2----I--   verbtag=A-----
napsal  lem=napsat   subl=napsat  tag=VpMS----R-AAP--   verbtag=VCA2vP
```

```
e-mail   lem=e-mail   subl=e-mail   tag=NNIS4-----A----  verbtag=------
o        lem=o        subl=o        tag=RR--6----------  verbtag=------
filosofii lem=filozofie subl=filosofie tag=NNFS6-----A----  verbtag=------
,        lem=,        subl=,        tag=Z:-------------  verbtag=------
učiňte   lem=učinit   subl=učinit   tag=Vi-P---2--A-P--  verbtag=VIA2P-
tak      lem=tak      subl=tak      tag=Db-------------  verbtag=------
,        lem=,        subl=,        tag=Z:-------------  verbtag=------
prosím   lem=prosit   subl=prosit   tag=VB-S---1P-AAI--  verbtag=VDA1SP
.        lem=.        subl=.        tag=Z:-------------  verbtag=------
“        lem="        subl="        tag=Z:-------------  verbtag=------
```

2.4.11.8 Zpracování agregátů

Rozdělené agregáty se opět spojí v jeden token (*a* + *byste* → *abyste*), ale zůstávají mu interpretace jeho složek.

```
„        lem="        subl="        tag=Z:-------------  verbtag=------
Chci     lem=chtít    subl=chtít    tag=VB-S---1P-AAI--  verbtag=VDA1SP
-        lem=-        subl=-        tag=Z0-------------  verbtag=------
li       lem=li       subl=li       tag=BN-------------  verbtag=------
,        lem=,        subl=,        tag=Z:-------------  verbtag=------
abyste   (lem=aby,    subl=aby,     tag=J,-------------, verbtag=------)
         (lem=být,    subl=být,     tag=Vc-P---2----I--, verbtag=A-----)
…
```

Výsledkem je, že každé složce agregátu jakožto jediného tokenu a každému tokenu v ostatních případech je přiřazena čtveřice údajů: lemma, sublemma, tag, verbtag.

3 Morfologická disambiguace: spolupráce introspekce vyjádřené pravidly s metodou strojového učení

V tomto hlavním oddílu se zabýváme morfologickou disambiguací, která je spolu s analýzou syntaktickou nejobtížnější fází lingvistické anotace korpusů. Nejprve stručně popíšeme obecné metody disambiguace, poté tzv. hybridní metodu, kterou jsme pro disambiguaci zvolili, včetně procesu disambiguace en gros (3.1). Pak se zaměříme na charakteristiku systému pravidel, zvaného LanGr (3.2). V 3.3 velmi stručně charakterizujeme nástroj MorphoDiTa, v 3.4 popíšeme spolupráci systému LanGr s nástrojem MorphoDiTa a předvedeme příklady zpracování dvou příkladových vět. Na závěr představíme v 3.5 úspěšnost disambiguace.

3.1 Metody disambiguace a hybridní disambiguace

Morfologická disambiguace spočívá v přiřazení náležité trojice (lemma, tag, verbtag) každému slovnímu tvaru podle kontextu, tj. podle jeho postavení ve větě. Trojice se vybírá z nabídky poskytnuté morfologickou analýzou, jejíž výsledek je vstupem

celého disambiguačního systému. Obecně existují tyto typy metod morfologické disambiguace:

A. metody založené na strojovém učení (včetně hlubokého strojového učení z trénovacích dat metodou neuronových sítí);
B. metody založené na lingvisticky motivovaných pravidlech;
C. metody hybridní: spolupráce metod typu A a B.[21]

Při značkování korpusů řady SYN se konkrétně uplatňuje typ C, kdy spolupracuje:

– systém lingvisticky motivovaných pravidel, zvaný LanGr (Květoň, 2006);
– nástroj MorphoDiTa založený na hlubokém strojovém učení (metoda neuronových sítí). Použitý nástroj je experimentální, nezveřejněnou verzí vyvíjeného softwaru (MorphoDiTa-research), jež byla poskytnuta ČNK k trénování na jeho vlastních datech a ke značkování jeho korpusů. Vlastnosti tohoto nástroje – s malými odlišnostmi – nejlépe vystihuje článek Straky et al. (2019).

Hybridní disambiguační systém obdrží na vstupu českou větu, která je výstupem výše popsané tokenizace, větné segmentace a morfologické analýzy. Disambiguace probíhá takto (podrobně viz § 3.4):

Nejprve se aplikuje systém pravidel LanGr spolu s frazémovým modulem Phras (Hnátková, 2002). Celý tento systém spočívá v odstraňování nesprávných (v daném kontextu) interpretací, tj. trojic (lemma, tag, verbtag) a

– buď úplně zpracuje vstupní větu, tj. každý token obdrží na výstupu jedinou výslednou interpretaci v podobě (lemma, tag, verbtag) a pak je disambiguace hotova;
– nebo provede pouze částečnou disambiguaci věty, tj. na výstupu zbude u některých tokenů více interpretací; v tomto případě pak disambiguaci dokončí nástroj MorphoDiTa, kterému systém LanGr předá řízení;
– nebo neodstraní z nabídky poskytnuté morfologickou analýzou ani jedinou trojici (lemma, tag, verbtag) u tokenů věty a celou disambiguaci provede výhradně nástroj MorphoDiTa, kterému systém LanGr předá řízení.

V každém případě bude mít po celém procesu disambiguace každý token přiřazenu právě jednu trojici (lemma, tag, verbtag), a to i v homonymních větách. Takovýmto způsobem jsou disambiguovány všechny věty ve zpracovávaném dokumentu/korpusu.

[21] Pokud víme, používá se této hybridní metody jen při značkování korpusů psané češtiny (zejména korpusů řady SYN) v projektu Český národní korpus.

Poznámka 1. Při zpracování věty systémem pravidel LanGr se **výjimečně** může stát, že systém odstraní všechny interpretace (lemma, tag, verbtag) u nějakého tokenu. Není to proto, že vstupní věta je gramaticky a/nebo pravopisně (např. absence čárky) chybná; je to vždy chyba některého pravidla, tj. chyba technického rázu, která nijak nesouvisí s vlastnostmi disambiguovaného textu. Pravidla jsou totiž konstruována tak, že interpretaci (lemma, tag, verbtag) odstraňují u každého tokenu jen za podmínky, že není jediná. Pokud k takovéto chybě dojde, na konci činnosti systému LanGr se u takového tokenu – a jen u něho – vrátí na výstup všechny interpretace, které poskytla morfologická analýza (ostatní tokeny ve větě systém LanGr standardně, tj. případně jen částečně disambiguoval), a disambiguaci pak dokončí nástroj MorphoDiTa.

3.2 LanGr – systém lingvisticky motivovaných pravidel

Systém LanGr je soustava cca 4 700 lingvisticky založených pravidel, která se použijí na vstupní větu jakožto posloupnost tokenů zpracovaných morfologickou analýzou. Postupně se na větu aplikují pravidla, která odstraňují (v daném kontextu) nesprávné trojice (lemma, tag, verbtag). Děje se tak kombinací dvou přístupů:

– odstraňováním co největšího počtu nesprávných lemmat a/nebo morfologických interpretací (tag a verbtag), dokud (v optimálním případě) nezbude u každého tokenu jediná náležitá trojice (lemma, tag, verbtag) – negativní redukční metoda;
– přímým určováním náležitých trojic (lemma, tag, verbtag) – pozitivní metoda.

3.2.1 Charakteristické vlastnosti systému LanGr

Systém LanGr má tyto vlastnosti (3.2.1.1–3.2.1.15):

3.2.1.1 Vystižení zákonitostí češtiny

Pravidla systému LanGr se snaží vystihnout zákonitosti češtiny platné především v pravopisně a gramaticky správně česky psaném textu, dokáže se však vypořádat i s odchylkami od kodifikace spisovného jazyka: zpracuje např. apokopované tvary minulých příčestí jako *řek*, *prokop*; dále nekodifikované tvary a konstrukce jako *bysme*, *kdybysme*, *by jsi*; tvar *si* jako nespisovnou variantu slovesného tvaru *jsi* a další. Obecně samozřejmě záleží na tom, které tvary obsahuje morfologický slovník použitý k anotaci. Systém LanGr zpracovává jazykové jevy na těchto rovinách jazykového popisu:

– fonetika
– morfologie
– syntax
– sémantika
– lexikologie (frazémy a obecně kolokace)

Níže uvádíme příklady jazykových zákonitostí v češtině v podobě tvrzení, která jsou pak v systému LanGr implementována jako pravidla mající formalizovanou podobu počítačových programů:

(a) Těsně před jménem v lokálu nemůže stát předložka s nelokálovou rekcí.

(b) Těsně po předložce nemůže stát slovesný tvar.[22]

(c) Těsně před jménem v lokálu nemůže stát slovesný tvar.[23]

(d) Ve větě po sobě nemohou bezprostředně následovat tři předložky. Tento syntaktický fakt se dá využít například při slovnědruhové disambiguaci věty:

(1) *Dorazili **na místo v** rekordním čase.*[24]

kde za předpokladu, že tvar *na* i tvar *v* jsou předložky, lze s jistotou tvrdit, že tvar *místo*, stojící mezi nimi, není předložka, nýbrž substantivum.

(e) Jestliže v kondicionálové klauzi je jediný tvar, který je potenciálním *l*-ovým příčestím, je to určitě *l*-ové příčestí.

(f) Jestliže se v klauzi vyskytuje záporné zájmeno jiné než lexém *nic*, není žádné finitní sloveso v této klauzi kladné. Tento gramatický fakt lze využít například při disambiguaci věty (2):

(2) *„**Žádný omyl**,“ přerušila ho rázně Joanna.*

kde tvar *omyl* je nutně substantivum, nikoli tvar slovesného lexému *omýt*.

3.2.1.2 Vytváření pravidel

Lingvistická pravidla v systému LanGr se na rozdíl od metod strojového učení vytvářejí takto (srov. Petkevič, 2014, s. 543n.):

– buď jsou formulována na základě lingvistické intuice autorů a ověřována na korpusových datech;

– nebo se naopak indukcí vyvozují z korpusových dat, při níž se přitom nepoužívá strojové učení, pouze vhodné dotazy do korpusů v aplikaci Kontext a nejrůznější kolokační míry;

– nebo kombinací obou přístupů.

---

[22] Až na frazémy typu *od nevidím do nevidím*, které se zpracovávají na počátku spuštění pravidel systému LanGr nebo se zpracovávají jako výjimky z příslušných obecných pravidel.

[23] Toto tvrzení neplatí stoprocentně, podobně dále. Pravidlo počítačově implementující toto tvrzení musí zahrnovat příslušné výjimky.

[24] Není-li řečeno jinak, jsou všechny příkladové věty převzaty z korpusu SYN2020 a dalších novějších korpusů současné češtiny (SYN v11, SYN v12, totalita…).

Obecně tak lze říci, že na základě parole (korpusová data) se odhalují systémové zákonitosti češtiny (langue) na výše uvedených rovinách popisu jazyka.

3.2.1.3 Jazyková závislost pravidel

Lingvistická pravidla v systému LanGr jsou jazykově závislá: vystihují vlastnosti gramatické a pravopisné zákonitosti/pravidla současné psané češtiny.

3.2.1.4 Pravidla a jejich skupiny

Systém LanGr se skládá z lingvisticky motivovaných pravidel a jejich skupin. Rozlišujeme přitom tyto pojmy:

– *pravidlo*: počítačový program provádějící disambiguaci; je psán ve speciálním jazyce, zvaném – stejně jako celý systém – LanGr.[25] Například následující pravidlo disambiguuje slovo po předložce jako neslovesný tvar:

```
Prep = ITEM IsSafe Preposition;
NonVerb = ITEM Possible not Verb;
LEAVE ONLY not Verb IN NonVerb.
```

První dva řádky popisují vlastnosti korpusových pozic (tokenů) nacházejících se ve větě těsně za sebou. Na první pozici označené *Prep* je specifikován tvar, který je jednoznačnou předložkou (*IsSafe Preposition*). Na druhé pozici je tvar, který má možnou neslovesnou interpretaci (*Possible not Verb*). Výkonný příkaz *LEAVE ONLY* stanoví, že na pozici označené jako *NonVerb* se odstraní všechny slovesné interpretace (*not Verb*), pokud tvar má nějakou neslovesnou interpretaci (*Possible not Verb*). Stručně řečeno, pravidlo formalizuje syntaktický fakt, že těsně po předložce se nenachází sloveso. Obecně: podmínka za predikátem *IsSafe* stanoví, že všechny interpretace daného tokenu jí musí vyhovovat; podmínka za predikátem *Possible* stanoví, že alespoň jedna interpretace daného tokenu jí musí vyhovovat. Pravidlo se může uplatnit, teprve když jsou splněny takovéto podmínky pro všechny zúčastněné tokeny.

– *Pravidlo*: posloupnost *pravidel*; pravidla jsou v *Pravidle* lineárně uspořádána a aplikují se v pořadí, v jakém jsou v *Pravidle* zapsána.
– *Skupina*: každá skupina je tvořena posloupností, jejímiž prvky jsou jiné skupiny (tzv. *podskupiny*) a/nebo *Pravidla*; *podskupiny* a *Pravidla* jsou ve skupině lineárně uspořádány a aplikují se v pořadí, v jakém jsou ve skupině zapsány.

[25] Označení LanGr tedy užíváme v obou významech; z kontextu by vždy mělo být zřejmé, o jaký význam právě jde.

Systém *skupin* a *podskupin* tvoří hierarchii, tj. *podskupina* může obsahovat zase své podskupiny a tak dále. Na konci tohoto rekurzívního systému jsou *Pravidla* se svými *pravidly*.

Z algebraického hlediska tedy skupiny, podskupiny a Pravidla tvoří stromový graf, kde

(i) nekoncové uzly představují podskupiny, přičemž na vrcholu hierarchie je kořenová skupina, pojmenovaná *root*; ta se skládá z podskupin *root0* a *root1* v tomto pořadí;

(ii) koncové uzly (listy stromu) představují Pravidla jakožto posloupnosti pravidel realizujících vlastní disambiguaci.

Uspořádání

a) pravidel v Pravidle

b) podskupin a Pravidel ve skupině

je relevantní, neboť systém je konstruován tak, že specifičtější *pravidla* v Pravidle předcházejí před obecnějšími. Týž princip je vyjádřen pořadím *podskupin* a Pravidel ve skupině.

3.2.1.5 Spolehlivost pravidel

Pravidla systému LanGr jsou podle spolehlivosti rozdělena do dvou skupin:

– stoprocentně bezpečná pravidla (shromážděná ve skupině *root0*);

– pravidla, která mohou výjimečně chybovat (shromážděná ve skupině *root1*).

Tímto rozdělením se také řídí proces disambiguace věty (viz § 3.4).

3.2.1.6 Typy pravidel

Kromě míry spolehlivosti rozlišujeme disambiguační pravidla podle následujících kritérií:

A. Podle počtu použití:

– **pravidla jednorázová:** použijí se pouze jednou a dostávají se ke slovu na samém začátku disambiguačního procesu. Odstraňují veškerý balast: nereálná lemmata jako *dostánýˊ* (homonymie se substantivním tvarem *dostání*), *býčit* (homonymie s adjektivními tvary *býčí*, *býčím*), *násilnit* (homonymie s adjektivním tvarem *násilní*)); dále morfologicky správně utvořené, ale nereálné morfologické interpretace tvarů, např. (téměř) nikdy nepoužívané přechodníky jako *baště*, *bohatě*, *Brně*, *Nedorost*, *Přerost* a další). Dále

zpracovávají jednoznačné gramatické kolokace: např. *k ničemu*, *kolem mne*, *pro tuhle*, *třídní boj*, *došlou poštu*, *prezidentu Masarykovi*…;

– **pravidla ostatní**, která se mohou aplikovat opakovaně, umožní-li to podmínky během zpracování (pravidla se aplikují na stále disambiguovanější větu): např. pravidla o shodě přísudku s podmětem, pravidla zpracovávající předložkovou rekci, pravidla pro valenci sloves, adjektiv, substantiv, adverbií a spojek.

B. Podle míry obecnosti:

– **pravidla obecná:** např. pravidlo o shodě přísudku s podmětem, o shodě přívlastků s řídícím substantivem, pravidla zpracovávající předložkovou rekci, valenci různého druhu, stupňování adjektiv a adverbií, různé typy klitik a další;

– **pravidla specifická:** zpracování konkrétních kolokací, pravidla vztahující se ke konkrétnímu lemmatu (např. *stát*, které je jak infinitivem sloves různého významu patřících k různým konjugačním paradigmatům, tak substantivem), pořadí příklonek…

C. Podle tematiky:

– podle slovních druhů a poddruhů: adjektiva, krátká adjektiva, slovesa reflexiva tantum, slovesa s určitou valencí…;

– podle homonymní konkurence slovních druhů a poddruhů: např. Adjektivum_Sloveso (token může být morfologicky buď adjektivum, nebo substantivum), Předložka_Substantivum, KrátkéAdjektivum_Sloveso…;

– podle morfologických kategorií: např. pád a pádová homonymie;

– podle rozdělení lexémů na apelativa / propria (při lemmatizaci);

– podle syntaktických jevů: shoda, rekce/valence, příklonky, slovosled, složené slovesné tvary…;

– podle syntaktických struktur: nominální skupiny, přístavkové skupiny, předložkové skupiny, číslovkové skupiny…

D. Podle specifického typu morfologické a lemmatické homonymie tvarů,[26] např.:

– typ substantivum maskulinum životné v nominativu singuláru / slovesné *l*-ové příčestí v singuláru maskulina životného nebo neživotného: *básnil*

[26] Systém LanGr pracuje primárně se slovními tvary, ovšem v disambiguačních pravidlech je možné specifikovat lemmata i slovní tvary pomocí jejich prefixů a sufixů.

(lemmata: *básnil* / *básnit*), *blb* (lemmata: *blb* / *blbnout*), *brblal* (lemmata: *brblal* / *brblat*), *fňukal* (lemmata: *fňukal* / *fňukat*), *žvatlal* (lemmata: *žvatlal* / *žvatlat*)…;

- typ substantivum femininum v dativu/lokálu singuláru / příslovce v pozitivu: *barevně* (lemmata: *barevna* / *barevně*; podobně dále), *divizně*, *léčebně*, *služebně*, *úhlopříčně*…;
- typ substantivum maskulinum životné v lokálu singuláru / substantivum maskulinum neživotné v nominativu či akuzativu singuláru / sloveso typu apokopované *l*-ové příčestí v singuláru maskulina životného nebo neživotného: *poslech* (lemma: *posel* / *poslech* / *poslechnout*);
- typ substantivum neutrum v instrumentálu singuláru či dativu plurálu / sloveso v indikativu prézentu 1. osoby singuláru: *hledím* (lemmata: *hledí* / *hledět*; podobně dále), *chrastím*, *klestím*, *pohořím*…;
- typ substantivum maskulinum neživotné v nominativu či akuzativu singuláru / sloveso v infinitivu: *drát* (lemmata: *drát* / *drát*; podobně dále), *kout*, *obrat*, *pažit*, *plát*, *růst*, *stát*…;
- vidová homonymie: *dát*, *hodit*, *povést* / *vést*…

3.2.1.7 Paradigmatika a syntagmatika češtiny – globální identifikátory

Paradigmatika češtiny (v širším smyslu) (viz 3.2.1.7.1 níže) a její syntagmatika (viz dále 3.2.1.7.2) jsou vyjádřeny tzv. *globálními identifikátory*. Jsou to označení množin nebo posloupností lemmat/tvarů spjatých nějakou společnou vlastností. Globální je nazýváme proto, že se dají použít v jakémkoli pravidle.

3.2.1.7.1 Vyjádření paradigmatiky češtiny globálními identifikátory

Globální identifikátory pro vyjádření paradigmatiky češtiny v širším smyslu jsou množiny lemmat či slovních tvarů sdílejících nějakou společnou vlastnost, která se dá vystihnout nějakým disambiguačním pravidlem. Níže uvádíme charakteristické příklady:

- globální identifikátor VerbInfValLemmas označuje seznam lemmat sloves majících obligatorní nebo fakultativní infinitivní valenci (*hodlat*, *chtít*, *snažit*…);
- globální identifikátor VerbOblGenValLemmas označuje seznam lemmat sloves majících obligatorní genitivní valenci (*týkat se*, *všímat si*…);
- globální identifikátor ImprobableLemmas označuje seznam lemmat, která jsou krajně nepravděpodobná; pravidla je v konkurenci s reálnějšími interpretacemi odstraňují (*lítostnit*, *loučeně*, *doleza*, *častě*, *pitomět*…);

– globální identifikátor AccToKillForms označuje seznam tvarů s chybnou nebo krajně nepravděpodobnou akuzativní interpretací (*bratří*);
– globální identifikátor AdjToKillForms označuje seznam tvarů s potenciální, ale téměř stoprocentně nesprávnou adjektivní interpretací (*kalí*, *klenou*, *nebílí*…).

Například skupina sloves majících obligatorní nebo fakultativní infinitivní valenci tvoří paradigma v tomto smyslu: je to skupina elementů (zde sloves) majících tuto společnou vlastnost – totiž vázat na sebe infinitiv (tato vlastnost sama je ovšem syntaktická). Této společné vlastnosti se dá využít při disambiguaci třeba takto: následuje-li po tvaru slovesa, jehož lemma je v seznamu VerbInfValLemmas, token *t*, který je morfologicky potenciálním infinitivem (může být totiž morfologicky homonymní, a mít tak i jinou morfologickou interpretaci), je velmi pravděpodobné, že token *t* bude opravdu infinitiv a disambiguací bude i takto určen.

V systému jsou stovky takovýchto globálních identifikátorů a pravidla jich využívají buď v pozitivním smyslu, tj. stanovují u příslušných lemmat/tvarů ve větě danou vlastnost, nebo v negativním, tj. danou vlastnost u příslušných lemmat/tvarů odstraňují.

3.2.1.7.2 Vyjádření syntagmatiky češtiny globálními identifikátory

Globálními identifikátory se vyjadřuje i syntagmatika češtiny: globální identifikátory označují n-tice (konkrétně dvojice či trojice) lemmat nebo tvarů majících společnou vlastnost, která se dá vystihnout nějakým disambiguačním pravidlem. Možnost vyjádřit lokální syntagmatiku ve zpracovávaných větách pomocí pozitivních bigramů či trigramů, tedy dvojic/trojic nějak kolokačně spjatých lemmat/tvarů, významně přispívá ke zpřesnění a zefektivnění činnosti pravidel. Pomocí seznamů označených globálními identifikátory lze ve větě identifikovat takové bigramy či trigramy, jež v ní nemusí být nutně v těsném sousedství, ale ve slovosledně volnějším vztahu – záleží na zpracovávajícím pravidle. V těchto seznamech nejsou pochopitelně všechny bigramy či trigramy s danou vlastností, ale hlavně ty, které se v korpusech vyskytují nejčastěji. Opět uveďme několik charakteristických příkladů:

1. Globální identifikátor NounAdjGenNounGenList označuje seznam mnoha tisíc nejfrekventovanějších trigramů lemmat ($lemma_1$, $lemma_2$, $lemma_3$), kde
   – $lemma_1$ je lemma potenciálního substantiva;
   – $lemma_2$ je lemma potenciálního adjektiva v genitivu;
   – $lemma_3$ je lemma potenciálního substantiva v genitivu.

Cílem je disambiguovat trigramy určitých korpusových pozic ve větě jako (Noun, Adj_{gen}, $Noun_{gen}$), tedy tak, že

– na první pozici bude tvar, jehož lemma = $lemma_1$, označkován jako tvar substantiva, je-li mezi jeho interpretacemi taková možnost;
– na druhé pozici bude tvar, jehož lemma = $lemma_2$, označkován jako tvar adjektiva v genitivu, je-li mezi jeho interpretacemi taková možnost;
– na třetí pozici bude tvar, jehož lemma = $lemma_3$, označkován jako tvar substantiva v genitivu, je-li mezi jeho interpretacemi taková možnost

a navíc chceme vyjádřit shodu tvarů na druhé a třetí pozici ve jmenném rodě a čísle, pokud to jejich interpretace umožňují. Výsledkem bude, že z potenciality se stane jistota. Korpusové pozice, jichž se uvedené vlastnosti týkají, přitom nemusí následovat těsně za sebou, ani nemusí být ve větě v tomto pořadí, záleží na formulaci pravidla, které s tímto globálním identifikátorem pracuje.

Mezi trigramy v seznamu označeném jako NounAdjGenNounGenList budou kupř. tyto trojice lemmat z korpusu totalita (Skoumalová et al., 2010):[27]

(*agregát*, *vlastní*, *konstrukce*)

(*akt*, *třídní*, *spravedlnost*)

Takovéto trigramy se rozpoznají například v těchto větách:

(3) *Zvláštní chloubou je oddělení montáže obráběcích, svářecích a jiných* ***agregátů vlastní konstrukce****.*

(4) *Současně nastolovala otázku odčinění všech revolučních* ***aktů*** *skutečné* ***třídní spravedlnosti*** *po Únoru 1948.*

Uveďme ještě pár dalších případů seznamů kolokací popsaných globálními identifikátory a využívaných v pravidlech.

2. Velmi významný je globální identifikátor NounNounGenList označující seznam obsahující bigramy lemmat, kde

– $lemma_1$ je lemma potenciálního substantiva;
– $lemma_2$ je lemma potenciálního substantiva v genitivu.

Cílem je nacházet v textech bigramy tvarů, jež mají – mimo jiné – tyto vlastnosti, a tyto vlastnosti jim přisuzovat. V seznamu budou například tyto bigramy lemmat:

(*absolvent*, *Princeton*)

(*možnost*, *pohyb*)

(*návštěva*, *restaurace*)

(*průběh*, *sklizeň*)

[27] Viz <https://wiki.korpus.cz/doku.php/cnk:totalita>.

Vhodné pravidlo tak umožní náležitě disambiguovat třeba dvojici tvarů *možnost pohybu* ve větě

(5) *Máme téměř neomezenou* ***možnost pohybu*** *po všech zákoutích chrámu.*

kde pád tvaru *pohybu* bude správně určen jako genitiv (nikoli jako dativ či lokál).

3. Globální identifikátor VerbPrepAccNounAccList označuje seznam trigramů lemmat

(lemma$_1$, lemma$_2$, lemma$_3$), kde

- lemma$_1$ je lemma potenciálního slovesa;
- lemma$_2$ je lemma potenciální předložky s akuzativní rekcí;
- lemma$_3$ je potenciálního substantiva s akuzativní rekcí.

Cílem je nacházet v textech trigramy tvarů s těmito vlastnostmi a přiřazovat jim uvedené vlastnosti, pokud tyto tvary mají ve svém morfologickém repertoáru uvedené interpretace. V seznamu jsou například tyto trigramy:

(*bojovat*, *o*, *dosažení*)
(*chodit*, *na*, *očkování*)
(*dělat*, *na*, *přání*)
(*považovat*, *za*, *neštěstí*)

4. Klíčový je globální identifikátor AdjNounList, který označuje seznam bigramů lemmat, kde

- lemma$_1$ je lemma potenciálního adjektiva;
- lemma$_2$ je lemma potenciálního substantiva.

Cílem je nacházet v textech bigramy tvarů s těmito vlastnostmi, a navíc chceme vyjádřit shodu tvarů na první a druhé pozici ve jmenném rodě, čísle a pádě, pokud to jejich interpretace dovolí. V seznamu budou například bigramy:

(*Adamův*, *jablko*)
(*bezprávný*, *stav*)
(*černý*, *hnát*)
(*křížový*, *palba*)
(*kulturní*, *září*)
(*kvalitní*, *kousek*)
(*tajný*, *poslech*)
(*třídní*, *vědomí*)
(*ženský*, *nemoc*)

Seznam například umožní náležitě disambiguovat bigram (*tajný*, *poslech*), kde

tajný může být

- adjektivum;
- nebo zpodstatnělé substantivum

a *poslech* může být

- buď nominativ singuláru, nebo akuzativ singuláru substantiva *poslech*;
- nebo lokál plurálu substantiva *posel*;
- nebo tvar maskulina singuláru apokopovaného *l*-ového příčestí slovesa *poslechnout*.

Takovýto bigram se uplatní například při zpracování věty:

(6) *V roce 1942 gestapo zatklo za **tajný poslech** zahraničního rozhlasu 59 osob z Obytic, Habartic, Kocourova a Mochtína.*

5. Globální identifikátor PrepAccAdjFormNounFormList označuje seznam trojic tvarů ($tvar_1$, $tvar_2$, $tvar_3$), kde

- $tvar_1$ je tvarem potenciální předložky s akuzativní rekcí;
- $tvar_2$ je tvarem potenciálního adjektiva v akuzativu;
- $tvar_3$ je tvarem potenciálního substantiva v akuzativu.

Vhodné disambiguační pravidlo zjistí přítomnost tohoto trigramu ve větě a všechny tvary tohoto trigramu náležitě disambiguuje, tj.

- $tvar_1$ je disambiguován jako předložka s akuzativní rekcí;
- $tvar_2$ je disambiguován jako adjektivum v akuzativu;
- $tvar_3$ je disambiguován jako substantivum v akuzativu.

Toto pravidlo navíc realizuje shodu $tvaru_2$ s $tvarem_3$ ve jmenném rodě a čísle, pokud to jejich interpretace umožňují. V seznamu těchto trigramů jsou například tyto trojice:

(*za*, *vysoké*, *půjčovné*)

(*za*, *zničené*, *sebevědomí*)

(*pro*, *návykové*, *nemoci*)

Seznam označený uvedeným globálním identifikátorem umožňuje například správně označkovat řetězec *za vysoké půjčovné* ve větě:

(7) *Po třech měsících obě pandy putovaly do San Francisca, také na tři měsíce, ale **za vysoké půjčovné**.*

Jak již bylo řečeno, že jednotlivé komponenty takovýchto trojic nemusí v disambiguovaných větách tvořit souvislou posloupnost, záleží na konstrukci příslušného pravidla, jak efektivně využije daného seznamu tak, aby rozpoznal co nejvíce uvedených trojic ve zpracovávaných větách.

6. V analogickém seznamu PrepLocAdjFormNounFormList lokálových trigramů tvarů ($tvar_1$, $tvar_2$, $tvar_3$), kde

- $tvar_1$ je tvarem potenciální předložky s lokálovou rekcí;
- $tvar_2$ je tvarem potenciálního adjektiva v lokálu;
- $tvar_3$ je tvarem potenciálního substantiva v lokálu.

nechybí například trigram (*na*, *taneční*, *ploše*), na jehož základě vhodné pravidlo stanoví lokál u všech slov a navíc určí, že *taneční* je tvar adjektivní (nikoli substantivní) a že *ploše* je tvar substantivní (nikoli adverbiální), a zajistí shodu ve jmenné skupině *taneční ploše*. Konkrétně ve větě

(8) *Po pár desítkách minut se **na** travnaté **taneční ploše** objevily první dvě tanečnice.*

umožňuje seznam označený globálním identifikátorem PrepLocAdjFormNounFormList disambiguovat trigram (*na*, *taneční*, *ploše*) takto:

- *na*: předložka s lokálovou rekcí (a nikoli například citoslovce);
- *taneční*: adjektivum feminina lokálu singuláru (a nikoli zpodstatnělé substantivum);
- *ploše*: substantivum *plocha* v lokálu singuláru (a nikoli například adverbium *ploše*).

Příslušné pravidlo navíc zajistí shodu ve jmenné skupině *taneční ploše* ve jmenném rodě a čísle. Dokáže úsek takto disambiguovat, i když slova netvoří slovosledně těsnou trojici, přičemž možný kontext (*travnaté*) musí být samozřejmě v takovém pravidle zohledněn.

7. Globální identifikátor VerbDatInsFormList označuje seznam trigramů tvarů ($tvar_1$, $tvar_2$, $tvar_3$), kde

- $tvar_1$ je tvarem potenciálního slovesa;
- $tvar_2$ je tvarem potenciálního dativu;
- $tvar_3$ je tvarem potenciálního instrumentálu.

Cílem je takto v textech disambiguovat příslušné trojice forem, tj. stanovit, že uvedené potenciální interpretace jsou interpretacemi jistými. Seznam obsahuje například tyto trojice slovních tvarů:

(*bude*, *mi*, *ctí*)

(*blesklo*, *jí*, *myslí*)

(*šel*, *lidem*, *příkladem*).

8. Je zjevné, že pomocí bigramů a trigramů lze zachytit nejrůznější syntaktické vztahy, přičemž zvlášť důležitá je povrchová valence. Například globální identifikátor VerbNounAccList označuje seznam nejfrekventovanějších bigramů lemmat ($lemma_1$, $lemma_2$), kde

– $lemma_1$ je lemma potenciálního tranzitivního slovesa;

– $lemma_2$ je lemma potenciálního substantiva v akuzativu.

Cílem je takto disambiguovat bigramy lemmat čili disambiguovat $lemma_2$ jako substantivum v akuzativu na základě tranzitivního slovesa $lemma_1$, přičemž toto $lemma_1$ je zároveň disambiguováno jako sloveso. V seznamu VerbNounAccList jsou například tyto dvojice lemmat:

(*brát*, *zřetel*)

(*sklízet*, *pšenice*)

(*zajistit*, *průběh*)

(*podat*, *výkon*)

Pokud se ve zpracovávané větě objeví příslušný bigram lemmat, pravidlo využívající globálního identifikátoru VerbNounAccList náležitě disambiguuje tvary ve větě odpovídající tomuto bigramu. Například ve větě

(9) *V Olomouci jsme **podali** velmi dobrý **výkon** a byli domácím přinejmenším rovnocenným soupeřem.*

takové pravidlo nalezne nespojitý bigram (*podali*, *výkon*), protože obě lemmata se rovnají lemmatům ve dvojici (*podat*, *výkon*) v seznamu VerbNounAccList, a stanoví pád tvaru *výkon* jako akuzativ. V pravidle se navíc bere zřetel na to, jaké tvary mohou být mezi členy bigramu, zde je to adverbium *velmi* a adjektivum *dobrý*.[28]

V systému LanGr je určitý seznam označený globálním identifikátorem typicky užíván v jednom pravidle. Mohou jej však užívat i jiná pravidla, například taková, která využívají daný n-gram v různých slovosledných variantách: různá pravidla využívají týž seznam pro disambiguaci té či oné slovosledné varianty. Obecně platí, že globální identifikátory a pravidla jsou navzájem nezávislé: libovolné pravidlo může užívat libovolného globálního n-gramu.

Vhodné n-gramy (tedy zvláště bigramy a trigramy), které je třeba disambiguovat, se – jakožto projev a realizace introspekce autorů pravidel – získávají vhodnými

[28] Systém obsahuje obdobné seznamy i pro ostatní pády.

dotazy do korpusů v aplikaci Kontext a různými kolokačními mírami, nikoli na základě strojového učení. Pro seznamy systému LanGr se vyhledávají zvláště n-gramy vykazující nějaký typ morfologické (včetně slovnědruhové) homonymie, který je zapotřebí disambiguovat: např. bigram forem předložkové skupiny (*při*, *výzkumu*) je pro disambiguaci zajímavý, neboť obsahuje slovnědruhový homonymní tvar *při* (nachází se v paradigmatech lexémů: substantivum *pře* / sloveso *přít* / předložka *při*), kdežto bigram jednoznačných forem předložkové skupiny (*vůči*, *odlišnostem*) nikoli.

Uvedeme nyní další podstatné vlastnosti systému LanGr (3.2.1.8–3.2.1.15).

3.2.1.8 Pravidla a kontext

Pravidla v systému LanGr jsou konstruována tak, že mohou využívat jen kontextu v rámci věty (stejně jako nástroj MorphoDiTa), kromě toho však kontext není ničím omezen. Například pravidla dokážou zpracovat vztah reflexiva (*se*/*si*) k libovolně vzdálenému tvaru jeho základového slovesa. S rozšířením kontextu nad rámec věty se nepočítá, bylo by to totiž velmi komplikované a dosud provedené experimenty ukázaly, že se úspěšnost značkování příliš nezvýšila.

3.2.1.9 Lingvistické zdroje pravidel

Autoři pravidel využívají dostupných elektronických i tradičních papírových zdrojů lingvistických informací: kromě korpusů samých jsou to třídy homonym (zvláště Petkevič, 2014), frazeologické slovníky, valenční slovníky, gramatiky obsahující různé seznamy lexémů a forem, databáze víceslovných lexikálních jednotek LEMUR[29] a další.

3.2.1.10 Ladění pravidel

Systém LanGr se dá snadno ladit. Systém je koncipován tak, že v disambiguačních pravidlech lze lehce lokalizovat chybu, tj. zjistit, které pravidlo chybně odstranilo správnou interpretaci u toho či onoho slovního tvaru, a chybu opravit.

3.2.1.11 Počet pravidel

Čeština je gramaticky velmi složitý jazyk: (i) má volný slovosled, a tedy bohatou slovoslednou kombinatoriku forem, (ii) jeho flexe je morfologicky bohatá zvláště

[29] Srov. Petkevič et al. (2020).

v nominálním podsystému (pády, jmenné rody), kde vykazuje jak homonymii zvláště **díky pádovému synkretismu**, tak synonymii (alomorfie), a navíc (iii) obsahuje velké množství homonymních forem. Není tedy divu, že takový jazyk lze postihnout jen počtem pravidel v řádu tisíců – od obecných po speciální týkající se jednotlivých slovních tvarů. V současnosti je jich přibližně 4 700.

3.2.1.12 Stabilita pravidel

Pravidla mají víceméně trvalý ráz – zvláště pravidla s obecnou platností lze valnou většinou považovat za platná nejméně po několik desetiletí (pravidla o shodě, předložkové rekci…), neboť jazyk se zvláště v plánu syntaktickém mění jen pozvolna.

3.2.1.13 Problémy disambiguace pomocí pravidel

Pravidla v systému LanGr mohou být samozřejmě chybná. Ty se dnes, kdy je systém již poměrně dost vyladěný, týkají již takřka výhradně složitějších syntaktických jevů, jen výjimečně nastane chyba při anotaci jednoduše disambiguovatelných struktur. Největší problémy pro disambiguaci představují tyto jevy (srov. i Petkevič, 2014, s. 545):

- převážně volný slovosled české věty;
- vysoká míra vlastní i nevlastní homonymie slovních tvarů v češtině (Těšitelová, 1966);
- bohatá flexe a zároveň vysoký stupeň morfologické homonymie, zvláště pádového synkretismu v nominálních (substantivních, adjektivních, zájmenných, číslovkových) paradigmatech – disambiguace pádů představuje patrně největší disambiguační problém;
- relativně malý počet syntakticky pevných bodů, tj. jednoznačně analyzovaných slov, o něž se může disambiguace opřít;
- stanovení hranic klauzí v souvětí, např. rozlišení koordinace klauzí a koordinace členské, což má pak vliv na anotaci např. samostatně stojících pasivních příčestí (verbtag=VO…), jež jsou poměrně často anotována chybně, dále pádů a řady dalších jevů;
- souvětí, která obsahují klauze vnořené do klauzí, které jsou jim nadřazené;
- cizí slova, která jsou homonymní s českými slovními tvary (něm. *den*, ang. *to*…) v disambiguačně složitějších kontextech;
- přejímaná cizí slova, která v češtině postupně zdomácňují: není dostatečně jasné, zda je anotovat jako cizí, nebo již jako některý standardní slovní druh (substantivum apod.).

3.2.1.14 Neanotované jevy

Některé jevy dosud nejsou anotovány, někdy proto, že jejich disambiguace by byla velice chybová vzhledem k široké „šedé zóně“ přechodových případů a k lexikalizaci. Takovými jevy jsou například:

- anotace reflexivních a kvazireflexivních sloves podtypu *se* a *si* (např. *snažit* [*se*], *posadit* [*se*], *všimnout* [*si*], *zamilovat* [*se*], *zamilovat* [*si*]), deverbativních adjektiv (*snažící* [*se*], *připravující* [*se*], *rozmýšlející* [*si*]) a verbálních substantiv (*prosazování* [*se*], *osvojení* [*si*], *přizpůsobení* [*si*]);
- anotace vidu deverbativních adjektiv (*přejímaný*, *přejatý*, *vytvářející*, *vytvořený*…) a verbálních substantiv (*sušení*, *usušení*…), případně adverbií (*nepřejícně*) – vid je tu dán motivujícím/fundujícím slovesem;
- anotace zkratek homonymních s běžnými slovními druhy.

3.2.1.15 Programovací jazyk LanGr

Označení LanGr není pouze označením systému a správy disambiguačních pravidel, ale i označením speciálního programovacího jazyka sloužícího k jejich implementaci. Tento jazyk i stejnojmenný správní systém nenaplněný daty a pravidly jsou jazykově nezávislé, a lze jich tedy využít k disambiguaci jakéhokoli přirozeného jazyka užívajícího grafémy.

3.3 Nástroj MorphoDiTa

Pro dokončení disambiguace v případech, kdy systém LanGr nedokáže větu plně disambiguovat, používáme nástroj MorphoDiTa-research, založený na hlubokém strojovém učení. Tento nástroj navazuje na starší úspěšný systém MorphoDiTa,[30] používaný k větné segmentaci, tokenizaci, morfologické anotaci i disambiguaci. Neuronová verze systému MorphoDiTa (využívá knihovnu Tensorflow) nebyla zveřejněna, byla ale poskytnuta ČNK ke značkování korpusů. MorphoDiTa byla natrénována na manuálně značkovaném korpusu Etalon o velikosti 2,27 milionu tokenů[31] (Skoumalová, 2021) s upraveným značkováním podle standardu SYN2020. MorphoDiTa využívá k disambiguaci vnoření slov (word embeddings, tj. vektorovou reprezentaci slov) vytvořené na základě korpusu SYN v7 (Křen et al., 2018) a využívá i exportu dat z české Wikipedie.

[30] Viz <https://ufal.mff.cuni.cz/MorphoDiTa>.
[31] Viz <http://hdl.handle.net/11234/1-3698>.

3.4 Spolupráce systému LanGr s nástrojem MorphoDiTa

V procesu disambiguace se nástroj MorphoDiTa dostává ke slovu až poté, co pravidla systému LanGr (spolu s modulem Phras) už nedokážou zpracovávanou větu dále disambiguovat a zároveň je ve větě aspoň jedno slovo s více než jednou interpretací (lemma, tag, verbtag), a disambiguaci dokončí.

3.4.1 Proces morfologické disambiguace věty

Vstupem pro morfologickou disambiguaci je věta jakožto posloupnost tokenů, kterou zpracovala morfologická analýza, jež každému tokenu (formě) identifikovanému tokenizací přiřadila množinu trojic (lemma, tag, verbtag). Celý proces disambiguace, při němž spolupracují systém LanGr, modul Phras a nástroj MorphoDiTa, probíhá v následujících fázích.

Fáze 1. Systém LanGr. Větu po morfologické analýze nejprve zpracovávají pravidla ze skupiny *root0*, která ve svých podskupinách a Pravidlech obsahuje stoprocentně platná pravidla. Ta jsou rekurzívně aplikována, tj. postupně odstraňují nesprávné trojice (lemma, tag, verbtag) u jednotlivých tokenů. Jakmile se ve větě odstraní nějaká trojice, změní se celková situace a v ní se mohou úspěšně uplatnit pravidla, která se dosud uplatnit nemohla, protože nebyly splněny náležité podmínky. Pravidla si pomáhají, neboť se rekurzívně uplatňují na stále disambiguovanější větu. Děje se tak do té doby, kdy už nejsou schopna nic disambiguovat, tj. odstraňovat chybné trojice (lemma, tag, verbtag). Nastane-li situace, že u všech tokenů ve větě zbude po aplikaci všech pravidel jediná trojice, je disambiguační proces dokončen. V opačném případě se nyní uplatní:

Fáze 2. Modul Phras, který se pokusí disambiguovat tokeny ve frazémech/kolokacích, jež rozpozná. Jakmile už nedokáže žádný token dále disambiguovat a věta dosud není úplně disambiguována, vrátí řízení pravidlům ze skupiny *root0* – následuje fáze 3. V opačném případě je disambiguační proces dokončen.

Fáze 3. Systém LanGr. Větu nyní opět zpracovávají pravidla ze skupiny *root0* jako ve fázi 1. Jakmile pravidla ze skupiny *root0* vyčerpají svůj disambiguační potenciál, tj. nedokážou už disambiguovat žádný token, a věta dosud není úplně disambiguována, aktivují se pravidla v Pravidlech a podskupinách skupiny *root1* (modul *Phras* se již podruhé neaktivuje) – následuje fáze 4.

Fáze 4. Systém LanGr. Aktivují se pravidla ze skupiny *root1* (obsahují méně bezpečná pravidla než pravidla ze skupiny *root0*). Jakmile už nedokážou žádný token dále disambiguovat a věta dosud není úplně disambiguována, vrátí řízení pravidlům ze skupiny *root0* a proces se opakuje. Takto spolu spolupracují pravidla obsažená ve skupinách *root0* a *root1* (bez modulu *Phras* mezi nimi), dokud jsou schopna přispět k disambiguaci tokenů ve větě. Nastane-li situace, že u všech tokenů ve větě zbude po této spolupráci jediná trojice (lemma, tag, verbtag), je disambiguační proces dokončen. V opačném případě systém LanGr svou činnost dokončil a disambiguace se nyní ujme nástroj MorphoDiTa – následuje fáze 5.

Fáze 5. Nástroj MorphoDiTa disambiguaci dokončí, tj. každému tokenu bude přiřazena právě jedna trojice (lemma, tag, verbtag).

Fáze 6. Automatická oprava výsledků nástroje MorphoDiTa identifikuje případy, kdy MorphoDiTa zvolila řešení, které se nenacházelo v nabídce po pravidlové disambiguaci, a z původní nabídky zvolí trojici (lemma, tag, verbtag), která je nejblíže řešení navrženému nástrojem MorphoDiTa.

3.4.2 Příklady procesu disambiguace

Proces disambiguace předvedeme – počínaje morfologickou analýzou – na větách (10) a (11). Kromě slovních tvarů (tokenů) uvádíme v příkladech níže pouze lemmata, tagy a verbtagy (nikoli sublemmata). Začněme větou

(10) *Láska ho stojí jmění.*

A. Morfologická analýza

Výstupem morfologické analýzy je tato podoba zpracovávané věty, v níž po tokenu následuje vždy seznam lomítkem oddělených dvojic (lemma, tag), případně těch trojic (lemma, tag, verbtag) tam, kde je hodnota verbtagu relevantní:

```
Láska: (Láska, NNMS1-----A----) / (láska, NNFS1-----A----)
ho:    (ho, II------------) / (on, PHIS2-3--------) /
       (on, PHIS4-3--------) / (on, PHMS2-3--------) /
       (on, PHMS4-3--------) / (on, PHNS2-3--------) /
       (on, PHNS4-3--------)
stojí: (stát, VB-P---3P-AAI--, VDA3PP) / (stát, VB-S---3P-AAI--, VDA3SP)
jmění: (jmění, NNNP1-----A----) / (jmění, NNNP2-----A----) /
       (jmění, NNNP4-----A----) / (jmění, NNNP5-----A----) /
       (jmění, NNNS1-----A----) / (jmění, NNNS2-----A----) /
       (jmění, NNNS3-----A----) / (jmění, NNNS4-----A----) /
       (jmění, NNNS5-----A----) / (jmění, NNNS6-----A----)
.:     (., Z:-------------)
```

B. Disambiguace – fáze 1. Systém LanGr

```
Láska: (Láska, NNMS1-----A----) / (láska, NNFS1-----A----)
ho:    ~~(ho, II------------) /~~ (on, PHIS2-3--------) /
       (on, PHIS4-3--------) / (on, PHMS2-3--------) /
       (on, PHMS4-3--------) / (on, PHNS2-3--------) /
       (on, PHNS4-3--------)
stojí: ~~(stát, VB-P---3P-AAI--, VDA3PP) /~~ (stát, VB-S---3P-AAI--, VDA3SP)
jmění: ~~(jmění, NNNP1-----A----) /~~ (jmění, NNNP2-----A----) /
       (jmění, NNNP4-----A----) / ~~(jmění, NNNP5-----A----) /~~
       ~~(jmění, NNNS1-----A----) /~~ (jmění, NNNS2-----A----) /
       (jmění, NNNS3-----A----) / (jmění, NNNS4-----A----) ~~/~~
       ~~(jmění, NNNS5-----A----) / (jmění, NNNS6-----A----)~~
.:     (., Z:-------------)
```

Škrtnuté jsou dvojice/trojice, které byly v této fázi disambiguací odstraněny.

C. Disambiguace – fáze 2. Modul Phras
V této fázi se odstranily další dvojice/trojice z těch, jež zbyly z předchozí fáze.

```
Láska:    (Láska, NNMS1-----A----) / (láska, NNFS1-----A----)
ho:       (on, PHIS2-3--------) / (on, PHIS4-3--------) /
          (on, PHMS2-3--------) / (on, PHMS4-3--------) /
          (on, PHNS2-3--------) / (on, PHNS4-3--------)
stojí:    (stát, VB-S---3P-AAI--, VDA3SP)
jmění:    (jmění, NNNS4-----A----)
.:        (., Z:-------------)
```

Modul Phras tedy plně disambiguoval substantivum *jmění*.

D. Disambiguace – fáze 3 a 4 systému LanGr
Systém LanGr v těchto fázích už k disambiguaci nijak nepřispěje a předá řízení nástroji MorphoDiTa.

E. Disambiguace – fáze 5. Nástroj MorphoDiTa odstraní zbývající chybné dvojice/trojice a disambiguaci optimálně dokončí s tímto výsledkem:

```
Láska:    (Láska, NNMS1-----A----) / (láska, NNFS1-----A----)
ho:       (on, PHIS2-3--------) / (on, PHIS4-3--------) /
          (on, PHMS2-3--------) / (on, PHMS4-3--------) /
          (on, PHNS2-3--------) / (on, PHNS4-3--------)
stojí:    (stát, VB-S---3P-AAI--, VDA3SP)
jmění:    (jmění, NNNS4-----A----)
.:        (., Z:-------------)
```

tedy přehledněji bez škrtů:

```
Láska:    (láska, NNFS1-----A----)
ho:       (on, PHMS4-3--------)
stojí:    (stát, VB-S---3P-AAI--, VDA3SP)
jmění:    (jmění, NNNS4-----A----)
.:        (., Z:-------------)
```

Další příkladová věta

(11) *Robert je člověk, který vždy pracoval od nevidím do nevidím a stará se příkladně o svou rodinu.*

je zajímavá z těchto hledisek:

(a) systém LanGr ji zcela disambiguuje;

(b) užívá k tomu zpracování negramatických kolokací. Níže uvádíme jednotlivé fáze zpracování.

A. Morfologická analýza
Jen stručně uveďme, že analýza přiřadí jednotlivým tokenům ve větě v průměru 4,58 tagu.

B. Systém LanGr větu plně disambiguuje s tímto výsledkem:

```
Robert:     (Robert, NNMS1-----A----)
je:         (být, VB-S---3P-AAI--, VDA3SP)
člověk:     (člověk, NNMS1-----A----)
,:          (, , Z:-------------)
který:      (který, P4MS1----------)
vždy:       (vždy, Db-------------)
pracoval:   (pracovat, VpMS----R-AAI--, VDA3SR)
od:         (od, RR--2----------)
nevidím:    (vidět, VB-S---1P-NAI--, VDA1SP)
do:         (do, RR--2----------)
nevidím:    (vidět, VB-S---1P-NAI--, VDA1SP)
a:          (a, J^-------------)
stará:      (starat, VB-S---3P-AAI--, VDA3SP)
se:         (se, P7--4----------)
příkladně:  (příkladně, Db-------------)
o:          (o, RR--4----------)
svou:       (svůj, P8FS4---------1)
rodinu:     (rodina, NNFS4-----A----)
.:          (. , Z:-------------)
```

Při zpracování vstupní věty se hned po spuštění disambiguace ještě před aktivací pravidel uplatní program, který „zaslepí“ negramatické kolokace, zde kolokaci *od nevidím do nevidím*, tj. nahradí původní tagy přiřazené morfologickou analýzou (*od*, *do* jako předložky, *nevidím* jako sloveso) adverbiálním tagem D?----------, který pak pravidlům systému LanGr nekomplikuje zpracování. LanGr tedy na vstupu dostane čtyři slova této kolokace jako adverbia. Jakmile LanGr provede disambiguaci, vrátí se zvláštním programem původní tagy jejich tokenům a ke slovu přijde MorphoDiTa, které negramatičnost kolokace nevadí.

3.5 Úspěšnost disambiguace

Níže uvádíme v tabulkách úspěšnost disambiguace a snížení ambiguity připadající na jednotlivé kroky procesu na základě výpočtů provedených v listopadu 2024 a za použití nejnovější verze systému LanGr a modulu Phras (LanGr i Phras – stav k 30. říjnu 2024) a nově přetrénovaného systému MorphoDiTa-research. Úspěšnost byla měřena postupem 10násobné křížové validace (při deseti cyklech trénování systému MorphoDiTa a testování hybridního systému bylo vždy 90 % dat využito k trénování a 10 % k testování tak, aby bylo postupně provedeno testování na celých datech).

V tabulkách 1–3 uvádíme údaje jako výsledky po různých fázích zpracování; jejich zkratky znamenají:

– LanGr = systém LanGr včetně modulu Phras;

– LanGr + MorphoDiTa = systém LanGr včetně modulu Phras + MorphoDiTa;

– Final = LanGr + MorphoDita + závěrečné automatické opravy pomocí speciálních velmi lokálních pravidel. Opravy zahrnují i opravy zvláštních chyb, jichž se MorphoDiTa dopouští. Ta totiž někdy nebere zřetel na morfologickou nabídku a přiřadí nějakému tokenu tag či verbtag, který v ní nebyl. Oprava se provádí s přihlédnutím k tomu, jak token (příp. jen částečně) disambiguovala pravidla;

– MorphoDiTa = zpracování pouze nástrojem MorphoDiTa, tj. bez použití pravidel systému LanGr.

3.5.1 Průměrný počet interpretací na token

Tabulka 1 ukazuje, jak se během jednotlivých kroků disambiguace snižuje ambiguita v textu, tj. průměrný počet trojic (lemma, tag, verbtag) připadající na jeden token.

krok zpracování	počet trojic (lemma, tag, verbtag) na token	lemmata	tagy	verbtagy	pády
Morf. analýza	**5,60**	**1,19**	**5,56**	**7,73**	**2,88**
LanGr	**1,51**	**1,03**	**1,50**	**1,12**	**1,38**
LanGr + MorphoDiTa	**1**	**1**	**1**	**1**	**1**

Tabulka 1: Počet trojic (lemma, tag, verbtag) na token po jednotlivých fázích zpracování. Po morfologické analýze připadá na každý token v průměru 5,60 trojice (lemma, tag, verbtag). Činnost systému LanGr (včetně Phras) sníží toto číslo na 1,51 čili cca o 73 %. Spoluprací systému LanGr a nástroje MorphoDiTa se nakonec každému tokenu přiřadí jediná trojice (lemma, tag, verbtag). Podobně je třeba chápat údaje v dalších sloupcích, přičemž údaje ve sloupci nadepsaném *verbtagy* se týkají jen sloves, údaje ve sloupci *pády* se týkají pouze jmen (substantiv, adjektiv, zájmen, číslovek) a předložkové pádové rekce.

3.5.2 Podíl jednoznačných tokenů

V tabulce 2 je uveden podíl jednoznačných tokenů, tj. tokenů s jedinou trojicí (lemma, tag, verbtag), a vět v procentech po jednotlivých krocích disambiguace z hlediska parametrů v záhlaví sloupců (lemma, tagy, verbtagy a pády).

krok zpracování	jednoznačné tokeny v %	lemmata	tagy	verbtagy	pády	jednoznačné věty v %
Morf. analýza	**35,27**	**85,04**	**35,49**	**31,26**	**24,17**	**1,81**
LanGr	**79,44**	**97,86**	**79,70**	**89,60**	**76,96**	**16,57**
LanGr + MorphoDiTa	**100**	**100**	**100**	**100**	**100**	**100**

Tabulka 2: Podíl jednoznačných tokenů a jednoznačných vět v procentech po jednotlivých fázích zpracování. Po morfologické analýze je jednoznačných tokenů, tedy těch, které mají po morfologické analýze přiřazenu jedinou trojici (lemma, tag, verbtag), celkem 35,27 %. Jednoznačných vět v tomto smyslu, tj. vět skládajících se výhradně z jednoznačných tokenů, je pouze 1,81 %. Po zpracování systémem LanGr (včetně Phras) se počet jednoznačných tokenů zvýší na 79,44 % čili původní počet se více než zdvojnásobí. Počet vět s výhradně jednoznačnými tokeny se zvýší na 16,57 % čili více než devětkrát. Podobně je tomu u dalších parametrů. Spoluprací systému LanGr a nástroje MorphoDiTa se nakonec každému tokenu přiřadí jediná trojice (lemma, tag, verbtag), jednoznačnost bude tedy stoprocentní.

3.5.3 Úspěšnost disambiguace

V tabulce 3 je uvedena úspěšnost disambiguace, tj. podíl tokenů, u nichž množina možných interpretací v jednotlivých fázích disambiguace obsahuje správnou (tedy manuálně určenou) interpretaci. Údaje v procentech uvádíme podrobněji pro tokeny a celkovou úspěšnost pro celé věty. Ve sloupci *vše* nalevo se zobrazuje podíl tokenů se správnou trojicí (lemma, tag, verbtag); sloupce *lemmata*, *tagy* ukazují podíl tokenů se správně určeným lemmatem či tagem; *verbtagy* představují podíl sloves se správně určeným verbtagem; *pády* ukazují podíl tokenů vyjadřujících pád se správně určeným pádem. Ve sloupci *věty* je uveden podíl vět, u nichž všechny tokeny mají přiřazenu správnou trojici (lemma, tag, verbtag).

krok zpracování	tokeny					věty
	vše	lemmata	tagy	verbtagy	pády	vše
Morf. analýza	99,91	99,96	99,94	100	99,99	98,66
LanGr	99,48	99,92	99,56	99,67	99,63	93,63
LanGr+Mdita	97,55	99,74	97,74	99,05	97,96	75,95
Final	**97,62**	**99,79**	**97,76**	**99,08**	**97,96**	**76,59**
MorphoDiTa	**97,47**	**99,71**	**97,69**	**98,84**	**97,79**	**75,54**

Tabulka 3: Úspěšnost disambiguace v procentech v kategorii *vše*, *lemmata*, *tagy*, *verbtagy*, a to *pády* v *tokenech* a ve *větách* po jednotlivých fázích zpracování.

Upozorněme krátce na nejvýznamnější údaje v prvním sloupci:

– systém LanGr chybuje celkem v 0,43 % případů (99,91–99,48);
– systém LanGr v součinnosti se systémem MorphoDiTa chybuje v 2,45 %

případů (100–97,55), samotný systém MorphoDiTa se tedy dopustí 1,93 % chyb (99,48–97,55);

– výsledná úspěšnost (Final = 97,62 %) je o 0,15 % vyšší než úspěšnost samotného systému MorphoDiTa (97,62–97,47 = 0,15), tj. hybridní anotace vykazuje o 6 % méně chyb.

Údaje v dalších sloupcích se týkají pouze lemmat, tagů, verbtagů, pádů a celých vět.

U verbtagu se počítají pouze relevantní tokeny, tj. jen ty, které mají verbtag netriviální (verbtag není roven „------“).

I u pádů se počítají pouze relevantní tokeny, tj. jen ty, které vyjadřují pád (jména a předložky, u nichž se vyjadřuje pádová rekce).

Srovnáme-li úspěšnost disambiguace tagů obecně a pouze disambiguace pádů, vidíme nevelký rozdíl, což dokládá, že chybová a obtížná je právě pádová disambiguace.

Údaje v posledním sloupci vyjadřují úspěšnost se zřetelem k větám:

– celkem 98,66 % vět obsahuje bezvadnou morfologickou analýzu;

– systém LanGr se při disambiguaci dopustí 5,03 % chyb (98,66–93,63);

– spolupráce systémů LanGr a MorphoDiTa toto číslo sníží o dalších cca 18 % (93,63–75,95);

– následné korektury disambiguaci poněkud vylepší na výsledných 76,59 %.

Ve všech sledovaných kategoriích vidíme, že spolupráce systému pravidel LanGr a nástroje MorphoDiTa je nepatrně lepší než výkon samotného nástroje MorphoDiTa, jen u vět je rozdíl výraznější (cca o 1 %: 76,59–75,54 = 1,05), a že je tedy rozumné ve spolupráci pokračovat.

Doplňme ještě, že systém LanGr:

– sice chybuje, ale jen výjimečně ve větách jednoduchých a v jednodušších syntaktických konstrukcích;

– má problémy zvláště s vymezením hranic klauzí, s disambiguací pádů, s koordinací, nominativem jmenovacím, a zejména s větami, které nejsou gramatické a/nebo nejsou psány podle kodifikovaného pravopisu (zejména absence interpunkce);

– sice chybuje minimálně, zato přenechává systému MorphoDiTa k řešení řadu složitých konstrukcí, s nimiž si (zatím) sám neví rady. Jedním z příkladů takových konstrukcí je například krátká otázka skládající se z předložky a zájmena, kde z dané věty není jasné, v jakém pádě zájmeno je (je to jasné ze širšího kontextu přesahujícího rámec zpracovávané věty). V každém případě je ale rekce předložky stejná jako pád zájmenného tvaru:

  (12) *Na jaké?*

Předložka *na* má ve větě (12) akuzativní nebo lokálovou rekci, zájmeno *jaké* může vyjadřovat mnoho kombinací jmenného rodu, čísla a pádu, přičemž pádem je tu vzhledem k předložce *na* nepochybně akuzativ nebo lokál. LanGr o pádu vzhledem k pádové homonymii obou slov alibisticky nerozhodne, ale jeho pravidla zaručují, že rekční pádový požadavek předložky *na* a pád zájmenného tvaru *jaké* se shodují, jenže se neví, v jakém pádě je shoda. MorphoDiTa však někdy (vzhledem k velikosti zpracovávaných korpusů jde o minimální počet případů!) označkuje konstrukci chybně tak, že předložce přisoudí jinou rekci než pád tvaru *jaké*: *na* (akuzativ), *jaké* (lokál).

4 Závěr

V článku jsme popsali způsob, jakým se morfologicky značkují nejnovější korpusy současné češtiny řady SYN, tj. korpus SYN2020 a korpusy mladší: SYN v10, SYN v11, SYN v12, korpus Online 2 a další, přičemž jsme se zaměřili hlavně na morfologickou disambiguaci a v jejím rámci na anotaci pomocí pravidel. Anotační systém hodláme – zvláště v jeho disambiguační složce – dále rozvíjet v těchto směrech:

- stále doplňovat a opravovat morfologický slovník, který se užívá k morfologické analýze;
- stále doplňovat, opravovat a ladit disambiguační pravidla systému LanGr se zvláštním zaměřením na nejobtížnější oblast morfologické disambiguace: disambiguaci pádů. Přitom zatím neplánujeme zapojit strojové učení při vyhledávání dalších n-gramů; vystačíme s různými kolokačními mírami a vhodně syntagmaticky formulovanými dotazy v aplikaci Kontext;
- v systému LanGr dále opravovat a doplňovat seznamy lemmat, forem a jejich kombinací označené v systému globálními identifikátory a vytvářet seznamy nové;
- v systému LanGr znemožňovat zvláště chyby, jichž se dosud dopouští MorphoDiTa, tj. zaměřovat se v pravidlech systému LanGr zvláště na oblasti, v nichž MorphoDiTa chybuje, a v nich formulovat vhodná pravidla;
- nadále co nejvíce využívat víceslovné jednotky z databáze LEMUR;
- trénovat systém MorphoDiTa na nových datech a ustavičně ladit hybridní systém značkování.[32]

[32] Autoři děkují recenzentům a redaktorům časopisu za upozornění na chyby, nedostatky a nepřesnosti a za cenná doporučení, doplnění a upřesnění. Za pomoc při přípravě článku děkují rovněž Mileně Hnátkové, Janu Křivanovi a Haně Skoumalové.

LITERATURA

Hajič, Jan (2004): *Disambiguation of Rich Inflection: Computational Morphology of Czech*. Praha: Karolinum.

Hajič, Jan – Bejček, Eduard – Hlaváčová, Jaroslava – Mikulová, Marie – Straka, Milan – Štěpánek, Jan – Štěpánková, Barbora (2020): Prague Dependency Treebank – Consolidated 1.0. In: Nicoletta Calzolari et al. (eds.), *LREC 2020, Twelfth International Conference on Language Resources and Evaluation: Conference Proceedings*. Marseille: European Language Resources Association, s. 5208–5218.

Hnátková, Milena (2002): Značkování frazémů a idiomů v Českém národním korpusu s pomocí Slovníku české frazeologie a idiomatiky. *Slovo a slovesnost*, 63(2), s. 117–126.

Jelínek, Tomáš – Křivan, Jan – Petkevič, Vladimír – Skoumalová, Hana – Šindlerová, Jana (2021): SYN2020: A new corpus of Czech with an innovated annotation. In: Kamil Ekštein – František Pártl – Miloslav Konopík (eds.), *Text, Speech, and Dialogue: 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6–9, 2021, Proceedings*. Cham: Springer, s. 48–59. <https://doi.org/10.1007/978-3-030-83527-9>.

Křen, Michal – Cvrček, Václav – Čapka, Tomáš – Čermáková, Anna – Hnátková, Milena – Chlumská, Lucie – Jelínek, Tomáš – Kováříková, Dominika – Petkevič, Vladimír – Procházka, Pavel – Skoumalová, Hana – Škrabal, Michal – Truneček, Petr – Vondřička, Pavel – Zasina, Adrian (2018): *Korpus SYN, verze 7 z 29. 11. 2018* [online]. Praha: Ústav Českého národního korpusu FF UK. <http://www.korpus.cz>.

Křen, Michal – Cvrček, Václav – Henyš, Jan – Hnátková, Milena – Jelínek, Tomáš – Kocek, Jan – Kováříková, Dominika – Křivan, Jan – Milička, Jiří – Petkevič, Vladimír – Procházka, Pavel – Skoumalová, Hana – Šindlerová, Jana – Škrabal, Michal (2020): *SYN2020: Reprezentativní korpus psané češtiny* [online]. Praha: Ústav Českého národního korpusu FF UK. <http://www.korpus.cz>.

Křen, Michal – Cvrček, Václav – Čapka, Tomáš – Hnátková, Milena – Jelínek, Tomáš – Kocek, Jan – Kováříková, Dominika – Křivan, Jan – Milička, Jiří – Petkevič, Vladimír – Skoumalová, Hana – Šindlerová, Jana – Škrabal, Michal (2024): *Korpus SYN, verze 13 z 27. 12. 2024* [online]. Praha: Ústav Českého národního korpusu FF UK. <http://www.korpus.cz>.

Křivan, Jan – Šindlerová, Jana (2022): Změny v morfologické anotaci korpusů řady SYN: nové možnosti zkoumání české gramatiky a lexikonu. *Slovo a slovesnost*, 83(2), s. 122–145.

Květoň, Pavel (2006): *Rule-based Morphological Disambiguation* [nepublikovaná disertační práce]. Praha: MFF UK.

Petkevič, Vladimír (2014): *Morfologická homonymie v současné češtině*. Praha: NLN – Ústav Českého národního korpusu.

Petkevič, Vladimír – Kopřivová, Marie – Hnátková, Milena – Jelínek, Tomáš – Kopřiva, Pavel – Rosen, Alexandr – Skoumalová, Hana – Vondřička, Pavel (2020): Typologie víceslovných jednotek v češtině a frekvenční zastoupení jejich hlavních vlastností v žánrově vyváženém korpusu. *Studie z aplikované lingvistiky*, 2020(2), s. 37–62.

Skoumalová, Hana – Bartoň, Tomáš – Hnátková, Milena – Kocek, Jan (2010): *Totalita: Korpus jazyka komunistické totality* [online]. Praha: Ústav Českého národního korpusu FF UK. <http://www.korpus.cz>.

Skoumalová, Hana (2021): *Etalon 1.0, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University* [online]. Cit. 14. 1. 2025. <http://hdl.handle.net/11234/1-3698>.

Straka, Milan – Straková, Jana – Hajič, Jan (2019): Czech text processing with contextual embeddings: POS tagging, lemmatization, parsing and NER. In: Kamil Ekštein (ed.), *Text, Speech, and Dialogue: 22nd International Conference, TSD 2019, Ljubljana, Slovenia, September 11–13, 2019, Proceedings*. Cham – Heidelberg – New York, NY – Dordrecht – London: Springer, s. 137–150. <https://doi.org/10.1007/978-3-030-27947-9_12>.

Straková, Jana – Straka, Milan – Hajič, Jan (2014): Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, MD: Association for Computational Linguistics, s. 13–18.

Štěpánková, Barbora – Mikulová, Marie – Hajič, Jan (2020): The MorfFlex Dictionary of Czech as a source of linguistic data. In: Zoe Gavriilidou – Maria Mitsiaki – Asimakis Fliatouras (eds.), *Proceedings of the XIX EURALEX Congress: Lexicography for Inclusion, Vol. I.* Alexandroupolis: EURALEX, s. 387–391.

Těšitelová, Marie (1966): *O morfologické homonymii v češtině* [Rozpravy Československé akademie věd: Řada společenských věd, roč. 76, sešit 8]. Praha: Academia.

Ústav lingvistiky FF UK
Panská 7, 110 00 Praha 1
tomas.jelinek@ff.cuni.cz
vladimir.petkevic@ff.cuni.cz

Tento článek je volně přístupný na základě licence *Creative Commons Uveďte původ 4.0 Mezinárodní veřejná licence*, která umožňuje používání, šíření a reprodukci článku za podmínky, že původní článek bude řádně citován. Licenční podmínky jsou dostupné zde: https://creativecommons.org/licenses/by/4.0/legalcode.cs.

This is an open access article under the terms of the *Creative Commons Attribution 4.0 International Public License*, which permits use, distribution and reproduction in any medium, provided the original article is properly cited. The license terms are available here: https://creativecommons.org/licenses/by/4.0/legalcode.