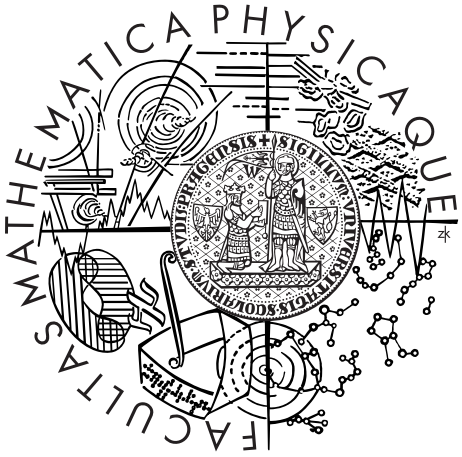
Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Marek Tlustý

Jazykové modelování pro němčinu

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Ondřej Bojar, Ph.D.
Studijní program: Informatika
Studijní obor: Obecná informatika

Praha 2013

Poděkování. Ondřejovi Bojarovi Rudolfovi Rosovi za identifikaci anglických klauzí. Daniel Zeman - nbestlisty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Jazykové modelování pro němčinu

Autor: Marek Tlustý

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Ondřej Bojar, Ph.D.

Abstrakt:

Klíčová slova: jazykové modelování, němčina,

Title: Language Modelling for German

Author: Marek Tlustý

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Ondřej Bojar, Ph.D.

Abstract:

Keywords: language modelling, German,

Obsah

1. Úvod

2. Jazykové modely

Jazykový model se snaží charakterizovat a zachytit zákonitosti v přirozeném jazyce. K tomu je možné přistupovat z pohledu statistiky nebo z pohledu hlubší lingvistické analýzy. Statistický přístup automaticky určuje všechny parametry z velkého množství textu v daném jazyce. Tento proces se nazývá *trénování modelu*. Modely opírající se hlavně o lingvistiku jsou tvořeny pravidly, která je potřeba naprogramovat ručně. Lze však využít i obou přístupů zároveň, a to například tak, že model nenecháme trénovat jenom na samotném textu, ale i na morfologických nebo jiných značkách či gramatických vztazích. Právě takovými modely se budeme zabývat.

2.1 Pohled z Bayesovy věty

Na přirozený jazyk lze nahlížet jako na množinu kontextuálních jednotek (např. vět, slov nebo jejich částí), které jsou náhodnými proměnnými s určitým rozdělením pravděpodobnosti. Například při překladu hledáme takové slovo B, které s největší pravděpodobností následuje po kontextu slov A. Hledáme tedy takové B, které maximalizuje podmíněnou pravděpodobnost $P(B|A)$. S využitím Bayesovy věty máme:

$$\arg\max_B P(B|A) = \arg\max_B \frac{P(A|B) \cdot P(B)}{P(A)} \tag{2.1}$$

Jmenovatel můžeme vynechat, neboť $P(A)$ je v tuto chvíli pouze konstanta, která hledání maxima nijak neovlivní. Dostáváme tedy:

$$\arg\max_B P(B|A) = \arg\max_B P(A|B) \cdot P(B) \tag{2.2}$$

2.2 N-gramové modely

N-gramové modely jsou založené na statistickém pozorování dat. Využívají například skutečnosti, že některá slova se často vyskytují v určitých dvojících (obecně n-ticích) - pro němčinu typicky třeba člen a podstatné jméno. Jistě častěji spatříme v trénovacích datech *der Hund* než *das Hund*. Stejně jako po slovese *fragen* uvidíme předložku *nach* nebo *um* spíše než *auf* nebo *an*.

Zajímá nás, jaká je pravděpodobnost výskytu takové posloupnosti slov $w_1, \ldots, w_m$. Tuto pravděpodobnost vypočítáme tak, že spočítáme výskyty všech těchto posloupností v datech a normalizujeme je velikostí dat. Trénovací data jsou ale obvykle řídká[1], a proto budeme chtít pozorované vlastnosti zobecnit.

Z Bayesovy věty víme, že

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{2.3}$$

[1]Řídkostí dat rozumíme počet různých kombinací slov v trénovacích datech vzhledem k celkovému počtu všech možných kombinací, kterých je nesrovnatelně více.

odtud vyjádříme $P(A, B)$ a dostaneme

$$P(A, B) = P(A|B) \cdot P(B) \tag{2.4}$$

nyní aplikujeme tento vztah na $P(w_1, \ldots, w_m)$ m-krát

$$P(w_1, \ldots, w_m) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \ldots \cdot P(w_m|w_1, \ldots, w_{m-1}) \tag{2.5}$$

Tento postup se nazývá **pravidlo zřetězení** a díky němu můžeme pravděpodobnost $P(w_1, \ldots, w_m)$ modelovat postupně člen po členu (např. slovo po slově).

Markovův předpoklad navíc říká, že každý člen posloupnosti $w_1, \ldots, w_m$ závisí jen na k předchozích. Potom tedy:

$$P(w_m|w_1 \ldots w_{m-1}) \simeq P(w_m|w_{m-k}, \ldots, w_{m-1}) \tag{2.6}$$

Toto tvrzení vede k zavedení pojmu n-gram a je vlastně předpokladem pro fungování n-gramových modelů.

N-gram je n po sobě jdoucích členů $w_1, \ldots, w_n$ z dané posloupnosti $w_1, \ldots, w_m$ (např. n po sobě jdoucích slov ve větě). Pro $n = 1, 2, 3$ používáme označení *unigram*, *bigram* a *trigram*.

Pravděpodobnost $P(w_m|w_{m-k}, \ldots, w_{m-1})$ z (2.6) přesně určit nelze, a proto se používá odhad maximální věrohodnosti (**MLE**):

$$P_{MLE}(w_m|w_{m-k}, \ldots, w_{m-1}) = \frac{count(w_{m-k}, \ldots w_m)}{\sum_l count(w_{m-k}, \ldots, w_{m-1}, w_l)} = \tag{2.7}$$

$$= \frac{count(w_{m-k}, \ldots w_m)}{count(w_{m-k}, \ldots, w_{m-1})}$$

Takto se rozdělí pravděpodobnost mezi všechny spatřené n-gramy v trénovacích datech a právě toto rozdělení pravděpodobnosti tvoří **n-gramový model**.

Problémem však stále zůstává skutečnost, že pro neviděné n-gramy v testovacích datech, dostaneme nulovou pravděpodobnost.

2.3 Good-Turing vyhlazování

Good-Turing vyhlazování se snaží vyhradit část rozdělení pravděpodobnosti od více frekventovaných n-gramů pro ty méně frekventované a neviděné. Používá k tomu frekvenci frekvencí n-gramů N_r, které se v trénovacích datech vyskytly r-krát. Tedy například pro $r = 3$ je N_3 rovno počtu n-gramů vyskytujících se v trénovacích datech právě třikrát.

Zajímavějším příkladem je ale N_0 tj. počet neviděných n-gramů. Ty nemůžeme spočítat přímo, ovšem výpočet také není nijak složitý. Stačí vzít počet všech možných n-gramů a odečíst počet n-gramů viděných. Pokud uvažujeme model slov, pak pro $n = 3$, velikost slovníku 100 a počet viděných n-gramů 350 000 je $N_0 = 100^3 - 350\,000 = 650\,000$.

Tato metoda bere n-gramy, které se vyskytly r-krát, jakoby se vyskytly r*-krát

$$r^* = (r+1) \cdot \frac{N_{r+1}}{N_r} \tag{2.8}$$

V jednodušší variantě se pro vhodně zvolenou konstantu k pravděpodobnost n-gramu vypočítá jako:

$$P_{GT}(w_1, \ldots, w_n) = \begin{cases} \dfrac{r^*}{\sum_r r \cdot N_r} & \text{je-li } r < k \\ \text{MLE} & \text{jinak} \end{cases} \tag{2.9}$$

Pokud bychom počítali pravděpodobnost pro všechny n-gramy podle prvního vzorce, nejen pro $r < k$, dostaly by ty nejvíce spatřené nulovou pravděpodobnost, neboť pro ně bude $N_{r+1} = 0$. Z tohoto důvodu je potřeba vhodně volit konstantu k a pro $r >= k$ počítat pravděpodobnost standardně odhadem maximální věrohodnosti (MLE), který dává dobré výsledky.

Ve složitější variantě se namísto konstanty k volí funkce $S(r)$ podle zjištěných hodnot r a N_r.

$$r^* = (r+1) \cdot \frac{S(r+1)}{S(r)} \tag{2.10}$$

Odhad pravděpodobnosti potom vypadá následovně:

$$P_{GT}(w_1, \ldots, w_n) = \begin{cases} \dfrac{N_1}{N_0 \cdot N} & \text{pro r} = 0 \\ \dfrac{r^*}{\sum_r r \cdot N_r} & \text{jinak} \end{cases} \tag{2.11}$$

Jedním ze způsobů určení funkce $S(r)$ je vykreslit $\log N_r$ proti $\log r$ a pomocí lineární regrese proložit přímku. Hodnoty $S(r)$ se potom určují podle této přímky. Spoustu hodnot N_r je ale nulových, proto se namísto $\log N_r$ používá $\log Z_r$:

$$Z_r = \frac{N_r}{0.5(t-q)} \tag{2.12}$$

kde q, r a t jsou po sobě jdoucí indexy mající N_q, N_r a N_t nenulové. Je-li N_r poslední nenulová frekvence n-gramů, dosadíme $t = 2r - q$. V případě, kdy $r = 1$, bereme $N_q = N_0$.

Good-Turing vyhlazování podává dobré výsledky pro málo frekventované n-gramy, a proto se v praxi často používá. Je také výchozím nastavením SRILM toolkitu[2] při trénování n-gramových modelů. Podrobněji o Good-Turing vyhlazování píše třeba .....[x] nebo ....[y].

[2]Sada nástrojů pro jazykové modelování. V tomto toolkitu budeme také trénovat všechny n-gramové modely. Více viz (odkaz na zdroj)

2.4 Katz back-off n-gramové modely

V trénovacích datech se nemusel objevit n-gram, který zrovna chceme a bez použití vyhlazování bychom dostali nulovou pravděpodobnost. V trénovacích datech ale mohl být podobný n-gram lišící se jen délkou historie. Pro získání informace od kratších n-gramů se proto využívá kombinace n-gramových modelů nižších řádů pomocí **lineární interpolace**.

K lineární interpolaci potřebujeme vektor vah λ, pro který platí:

$$\forall i : 0 \leq \lambda_i \leq 1 \quad \text{a} \quad \sum_i \lambda_i = 1 \tag{2.13}$$

Výsledná pravděpodobnost pro trigramový model pak vypadá takto:

$$P(w_3|w_1, w_2) = \lambda_3 P(w_3|w_1, w_2) + \lambda_2 P(w_3|w_2) + \lambda_1 P(w_3) \tag{2.14}$$

Vektor vah je však zatím neznámý, existují algoritmy pro jeho automatické určení - např. *EM algoritmus* (viz ZDROJ).

Na podobné myšlence kombinace n-gramových modelů s různou délkou historie jsou právě založeny **back-off n-gramové modely**. Ty ovšem neurčují pravděpodobnost vždy podle všech n-gramových modelů nižších řádů, ale využívají nižší řády pouze, pokud ty vyšší neposkytují dostatečnou informaci. Začíná se u modelů s nejvyšším řádem, pokud tento n-gram nebyl spatřen, proběhne tzv. **back-off** k nižšímu řádu a n-gramu se zkrátí historie o poslední člen (např. slovo). Pokud ani tento nižší řád n-gram se zkrácenou historií nikdy neviděl, pokračuje se v back-off operacích, dokud takový řád není nalezen.

Stejně jako v případě lineární interpolace se pravděpodobnosti jednotlivých modelů musely přenásobit vahami λ, aby se stále jednalo o validní rozdělení pravděpodobnosti, musíme najít takový způsob i u této metody. Zde musíme určit složitější normalizační faktor, neboť modelů nižších řádů nebudeme využívat vždy.

Katz back-off modely proto odhadují pravděpodobnost n-gramu následovně:

$$P_{BO}(w_n|w_1, \ldots, w_{n-1}) = \begin{cases} d_{w_1,\ldots,w_n} \cdot P_{MLE}(w_1, \ldots, w_n) & \text{pro } count(w_1, \ldots, w_n) > k \\ \alpha_{w_1,\ldots,w_{n-1}} \cdot P_{BO}(w_n|w_2, \ldots, w_{n-1}) & \text{jinak} \end{cases} \tag{2.15}$$

kde

- P_{MLE} označuje odhad maximální věrohodnosti zavedený ve vzorci (2.7)
- k je nejméně důležitý parametr a často je voleno $k = 0$
- d je snižující parametr, který zajišťuje vyhrazení určité části pravděpodobnosti pro odhady pravděpodobností s použitím back-off operací
- α je normalizační faktor přerozdělující zbývající část pravděpodobnosti

Parametr d je možné stanovit na základě popsaného Good-Turing vyhlazování následovně:

$$d_{w_1,\ldots,w_n} = \frac{count(w_1, \ldots, w_n)^*}{count(w_1, \ldots, w_n)} \tag{2.16}$$

přičemž $count(w_1, \ldots, w_n)^*$ se spočítá dle vzorce (2.8) nebo (2.10) z Good-Turing vyhlazování.

Výpočet normalizačního faktoru α je o něco složitější. Nejprve zavedeme β jako doplněk pravděpodobnosti součtu všech n-gramů s počtem výskytu (count) vyšším než k. β tak bude představovat zbývající vyhrazenou část pravděpodobnosti pro (n-1)-gramy.

$$\beta_{w_1,\cdots,w_{n-1}} = 1 - \sum_{\{\text{n-gram}|count(\text{n-gram})>k\}} d_{w_1,\cdots,w_n} P_{MLE}(w_1, \ldots, w_n) \tag{2.17}$$

Potom se normalizační faktor α vypočítá jako podíl zbývající pravděpodobnosti β a součtu pravděpodobností n-gramů vyskytujících se nejvýše k-krát. Tím se zajistí vždy ještě dostatek pravděpodobnosti pro další přechod k n-gramům nižších řádů back-off operacemi.

$$\alpha_{w_1,\cdots,w_{n-1}} = \frac{\beta_{w_1,\cdots,w_{n-1}}}{\sum_{\{\text{n-gram}|count(\text{n-gram})\leq k\}} P_{BO}(w_n|w_2 \cdots w_{n-1})} \tag{2.18}$$

Back-off n-gramové modely podávají dobré výsledky, a proto jsou v praxi často využívány. Tento typ modelů je výchozím nastavením nástroje `ngram-count` pro trénování modelu z již zmíněného SRILM toolkitu a právě takové modely budeme v této práci vyrábět.

2.5 Kneser-Ney vyhlazování

Kneser-Ney vyhlazování se snaží nahradit unigramovou pravděpodobnost, která závisí pouze na frekvenci výskytu slova v trénovacím korpusu, chytřejší pravděpodobností, která bude zohledňovat, v kolika různých kontextech se toto slovo vyskytuje. Tato metoda předpokládá, že slovo vyskytující se ve více kontextech je pravděpodobnější i pro výskyt v kontextu novém.

Pro příklad se často uvádí věta se San Franciscem a brýlemi:

- Mějme část věty: *Nemohu najít své čtecí*
- Naším úkolem je uhádnout slovo, které bude následovat.
- Předpokládejme, že unigramový model by nabídnul slovo Francisco. Proč? Protože se v trénovacím textu vyskytovalo nejčastěji.
- Kneser-Ney vyhlazování zavádí pravděpodobnost zohledňující počet kontextů, kde se dané slovo vyskytlo. Tato pravděpodobnost proto odhalí, že ačkoliv se Francisco objevovalo často, pak jenom po slovu San. Naproti tomu brýle se vyskytovaly v o mnoho více kontextech, a proto jim bude přidělena vyšší pravděpodobnost.

Pravděpodobnost zohledňující počet kontextů je definována jako:

$$P_{CONTINUATION}(w_i) = \frac{|\{w_{i-1} : count(w_{i-1}, w_i) > 0\}|}{\sum_{w_j} |\{w_{i-1} : count(w_{i-1}, w_j) > 0\}|} \tag{2.19}$$

Čitatel představuje počet slov, které se v trénovacím textu objevily před slovem w_i. Jmenovatel pak celkový počet slov objevujících se před všemi možnými slovy.

$P_{CONTINUATION}$ lze využít jak u interpolace, tak u back-off modelů jako náhrada unigramového modelu. Podrobnější informace se lze dočíst ve (ZDROJ: http://www.ee.columbia.edu/ stanchen/papers/h015a-techreport.pdf).

2.6 Modely maximální entropie

Entropie je minimální průměrný počet bitů potřebný k zakódování popisu výstupu nějaké náhodné veličiny. Pro náhodnou veličinu X a její distribuci P_X je dána entropie vztahem:

$$H(P_X) = -\sum_x P_X(x) \cdot log_2 P_X(x) \tag{2.20}$$

Ideou modelů **maximální entropie** je najít podmíněné rozdělení pravděpodobnosti, které má za daných podmínek maximální entropii. Jinými slovy se snažíme najít co nejjednodušší popis na základě toho, co známe - *princip Occamovy břitvy*. Díky tomu se popis co nejvíce blíží rovnoměrnému rozdělení a má tak co nejvyšší entropii.

Z trénovacího textu se budeme snažit napozorovat jen některé důležité vlastnosti, které jsou reprezentovány pomocí binárních funkcí a nazývají se **rysy** (features). Tyto funkce mohou být např. použity pro reprezentování nám již známých n-gramů. Pro trigram w_1, w_2, w_3 a historii h může funkce vypadat následovně:

$$f_{w_1,w_2,w_2}(h,w) = \begin{cases} 1 & \text{pokud } h \text{ končí } w_1, w_2 \text{ a } w = w_3 \\ \\ 0 & \text{jinak} \end{cases} \tag{2.21}$$

Díky takovému popisu nejsme omezeni jen na n-gramy. Rysy mohou představovat jakoukoliv skutečnost z historie, ať už se jedná třeba o začáteční písmeno prvního slova věty nebo morfologickou třídu předchozího slova. Na takové rysy můžeme pohlížet jako na jednotlivé modely a budeme hledat jejich vhodné kombinace. Modely maximální entropie ale nestaví modely samostatně, nýbrž vytváří hned jediný kombinovaný model.

Na základě toho nebudeme používat při určování pravděpodobnosti jen posloupnosti slov, ale zavedeme obecnější pojmy. **Kontextem** budeme rozumět jakousi historii tj. data, která máme k dispozici v době predikce. **Výsledkem** pak výstup, jež chceme predikovat. Dvojice kontext a výsledek je označována jako **událost**. V případě modelů čistě se slovy může být událostí n-gram $w_1, \ldots, w_n$, kde predikujeme slovo w_n na základě historie slov $w_1, \ldots, w_{n-1}$.

Výsledný model má následující podobu:

$$P(x|h) = \frac{e^{\sum_i \lambda_i f_i(x,h)}}{Z(h)}, \tag{2.22}$$

kde

- x je predikovaný výsledek
- h je kontext představující historii
- λ_i jsou váhy
- $f_i(x, h)$ jsou funkce reprezentující rysy
- $Z(h)$ je normalizační faktor definovaný takto:

$$Z(h) = \sum_{x_i \in V} e^{\sum_j \lambda_j f_j(x_i, h)} \tag{2.23}$$

- V je množina všech možných výsledků (např. slov)

Během trénování modelu maximální entropie se snažíme naučit optimální váhy λ_i korespondující s funkcemi rysů f_i. To je ekvivalentní hledání odhadu maximální věrohodnosti vah Λ s využitím logaritmu věrohodnostní funkce $\mathcal{L}(X|\Lambda)$ trénovacích dat X. Váhy jsou určovány speciálními metodami, nejčastěji *GIS - Generalized Iterative Scaling* (Darroch, Ratcliff [ZDROJ]) nebo *LBFGS - Limited Memory BFGS* (Liu, Nocedal [ZDROJ]). *BFGS* jsou počáteční písmena příjmení autorů původní metody pro řešení neomezených nelineárních optimalizačních problémů - Broyden-Fletcher-Goldfarb-Shanno.

Stanovení optimálních vah je náročná a složitá operace, která může trvat dlouhou dobu, pokud se k ní přistupuje zcela přímočaře. V každé iteraci algoritmu se musí spočítat normalizační faktor $Z(h)$ pro všechny spatřené kontexty v trénovacích datech. Pro každý kontext je zapotřebí projít přes všechna slova ze slovníku, tedy i přes ta, která se neobjevila v daném kontextu.

Jednou z technik jak snížit složitost počítání normalizačního faktoru jsou vnořené nepřekrývající se rysy - tedy např. n-gramové rysy. Pro ně totiž můžeme normalizační faktor spočítat takto - mějme historii w_{i-1}, w_{i-2}, pak

$$\begin{aligned} Z(w_{i-1}, w_{i-2}) = & \sum_{w_i \in V} e^{f w_i} + \\ & + \sum_{w_i \in V_{w_{i-1}}} (e^{f w_{i-1} w_i} - 1) \cdot e^{f w_i} + \\ & + \sum_{w_i \in V_{w_{i-2} w_{i-1}}} (e^{f w_{i-2} w_{i-1} w_i} - 1) \cdot e^{f w_{i-1} w_i}, \end{aligned} \tag{2.24}$$

kde

- V je slovník
- $V_{w_{i-1}}$ je množina slov pozorovaných po kontextu w_{i-1}
- $V_{w_{i-2} w_{i-1}}$ je množina slov pozorovaných po kontextu $w_{i-2} w_{i-1}$

První suma nezávisí na kontextu a může být předpočítána. Druhá je stejná pro všechny kontexty končící na w_{i-1} a její hodnotu proto můžeme mezi nimi sdílet. Poslední suma vyžaduje součet přes všechna slova spatřená po kontextu $w_{i-2} w_{i-1}$, takových je ale pro většinu kontextů málo.

2.7 Vyhlazování modelů maximální entropie

Stejně jako u n-gramových modelů se u modelů maximální entropie (maxentových) používá vyhlazování. Technice vyhlazování se zde často říká **regularizace**.

Jednou z nejčastějších metod je **Gaussian priors**, která přidává ke všem vahám rysů apriorní pravděpodobnost s nulovou střední hodnotou a daným rozptylem σ. Optimalizační kritérium modelu se tak změní na:

$$\mathcal{L}'(X|\Lambda) = \mathcal{L}(X|\Lambda) - \sum_i \frac{\lambda_i^2}{2\sigma_i^2} \tag{2.25}$$

Typicky se používá $\sigma_i = \sigma$ pro všechny parametry. Optimální rozptyl je obvykle stanoven z vývojových dat.

Vyhlazování Gaussian Prior je implementováno i v *MaxEnt Toolkitu* od Le Zhanga [ZDROJ], který také budeme využívat pro trénování maxentových modelů s vlastní množinou rysů.

Složitější technikou vyhlazování je $\ell_1 + \ell_2^2$ **regularizace**. Zde má optimalizační kritérium následující podobu:

$$\mathcal{L}_{\ell_1+\ell_2^2}(X|\Lambda) = \mathcal{L}(X|\Lambda) - \frac{\alpha}{D}\sum_i |\lambda_i| - \sum_i \frac{\lambda_i^2}{2\sigma_i^2 D}, \tag{2.26}$$

kde

- D je počet trénovacích pozorování
- α a σ jsou regularizační parametry

Parametry α a σ byly empiricky stanoveny na optimální hodnoty $\alpha = 0.5$ a $\sigma^2 = 6$ - viz Chen [ZDROJ]. ([4] z Tanela)

$\ell_1 + \ell_2^2$ regularizaci využívá rozšíření *SRILM Toolkitu* od Tanela Alumäe a Mikko Kurima [ZDROJ]. Toto rozšíření slouží pro trénování maxentových modelů s n-gramovými rysy. Pomocí tohoto rozšíření budeme vyrábět i naše maxentové n-gramové modely.

2.8 Hodnocení modelů

Abychom mohli vyhodnotit a porovnat kvalitu jazykových modelů, potřebujeme zavést taková kritéria, která budou dostatečně vypovídající a vzájemně porovnatelná i při použití různých druhů modelů a metod trénování.

2.8.1 Křížová perplexita

Jedním z hlavních měřítek pro kvalitu jazykového modelu je **křížová perplexita**. Udává, jak moc jsme překvapeni z následujícího slova a je dána vztahem:

$$PPL = 2^{H(P_E, P_{LM})}, \tag{2.27}$$

kde $H(P_E, P_{LM})$ je křížová entropie, P_E distribuce pravděpodobnosti trénovacích dat a P_{LM} distribuce pravděpodobnosti jazykového modelu.

Křížová entropie je obdobou entropie ze vzorce (2.20). Křížová ale udává vztah mezi dvěma distribucemi pravděpodobnosti namísto jedné a vypočítá se jako:

$$H(P_E, P_{LM}) = -\sum_x P_E \cdot log_2 P_{LM}(x), \tag{2.28}$$

Distribuce testovacích dat bývá nejčastěji stanovena jako $P_E(x) = \frac{n}{N}$, pokud se x vyskytlo n-krát v testovacích datech velikosti N.

Čím je perplexita nižší, tím lépe umí jazykový model předpovídat následující slovo a tím je samozřejmě lepší.

2.8.2 Adekvátnost a plynulost překladu

Kvalita jazykového modelu při překladu bývá hodnocena i ručně lidmi, a to především dvěma kritérii - adekvátností a plynulostí.

- **Adekvátnost** (adequacy) udává, zda překlad zachovává význam, či zda je změněn nebo nekompletní.
- **Plynulost** (fluency) hodnotí, jak je překlad plynulý, zda má přirozený slovosled apod.

Obě metriky nabývají hodnot $1, 2, \ldots, 5$ a nesou následující význam:

Hodnota	Adekvátnost	Plynulost
1	žádný význam	nesrozumitelný
2	málo z původního významu	neplynulý jazyk
3	dostatečně významu	nepřirozený
4	většina významu	dobrý jazyk
5	veškerý význam	bezchybný jazyk

Ruční hodnocení má ale nevýhodu v tom, že je pomalé, drahé a subjektivní. Mezianotátorská shoda ukazuje, že se lidé shodnou více na plynulosti než na adekvátnosti. [PÍŠE O TOM BAISA V UČEBNÍM TEXTU, ALE CHYBÍ ZDROJ]

V našich experimentech se zkusíme podívat, jak spolu koreluje právě automatické hodnocení (perplexita) s ručním hodnocením plynulosti.

2.9 Aplikace jazykových modelů

Jazykové modely mají široké využití. Používají se především ve strojovém překladu, kde se z nabízených překladových hypotéz snaží vybrat tu, co vypadá jako nejhezčí věta. Stejnou úlohu mají i v rozpoznávání mluvené řeči nebo tištěného textu. Mezi další patří např. obnovení diakritiky, korekce pravopisu nebo třeba prediktivní psaní SMS zpráv.

3. Problémy s němčinou

Němčina patří do skupiny flektivních jazyků tj. takových, které gramatické funkce vyjadřují pomocí flexe - ohýbání. Němčina používá mimo časování a skloňování složitý slovosled. Díky tomu mají tradiční n-gramy s němčinou problémy. V trénovacích datech se nám nemohou objevit všechny gramatické kombinace - např. spojení přídavného a podstatného jména ve všech pádech a kontextech. Techniky vyhlazování modelů gramatice nerozumí a nemohou určit v takovém případě za přídavné jméno daného tvaru správné podstatné jméno vhodného rodu, pádu a čísla.

3.1 Skloňování jmen

Německá gramatika zná 4 pády - *nominativ*, *genitiv*, *dativ* a *akuzativ*. Skloňování probíhá pomocí členů a koncovek.

- **Podstatná jména**
  Podstatná jména jsou skloňována především za pomocí členů, koncovka *-(e)s* se přidává ve druhém pádě rodu mužského a středního čísla jednotného a koncovka *-(e)n* ve třetím pádě čísla množného. *Např. der Hund, des Hundes.* Takto se skloňuje většina podstatných jmen.

  Mimo pravidelného (silného) skloňování existuje ještě skloňování slabé. Slabé skloňování přijímá koncovku *-en* ve všech pádech kromě prvního. *Např. der Student, des Studenten.* Tímto způsobem se obvykle skloňují podstatná jména rodu mužského označujících živé bytosti, příslušníky národností nebo slova cizího původu.

- **Přídavná jména**
  U přídavných jmen je situace ještě složitější. Mimo členu se v naprosté většině případů mění i koncovka. Ta je závislá mimojiné i na tom, zda předcházel člen určitý nebo neurčitý. Jednoduše se dá však říci, že koncovka má za úkol vyjádřit rod, pokud není zřejmý ze členu. *Např. ein schönes Haus x das schöne Haus.*

3.2 Pořádek slov

V němčině se rozlišují dva pořádky slov, asice pořádek přímý a pořádek nepřímý. Speciálním případem je pak ještě pořádek slov ve vedlejší větě.

- **Pořádek přímý**
  Pořádek přímý se používá hlavně v oznamovacích větách. Musí být dodrženo pořadí podmět, přísudek na začátku věty.
  Např. Jsem doma. - Ich bin zu Hause.

- **Pořádek nepřímý**
  Pořádek nepřímý se používá především v tázacích větách. Často se ale používá i ve větách oznamovacích, kde se předsune větný člen na začátek věty pro zdůraznění. Pořadí podmětu a přísudku se pak mění a podmět následuje hned za přísudkem.
  Např. Znáš ji? - Kennst du sie? Dnes jsem doma. - Heute bin ich zu Hause.

- **Pořádek ve vedlejší větě**
  Vedlejší věty mají speciální pořádek slov. Po podřadící spojce následuje hned podmět a sloveso jde až na konec věty.
  Např. Nevím, jestli ho zná. - Ich weiß nicht, ob sie ihn kennt.

3.3 Větný rámec

Němčina dává ve spoustě případů nějaké slovo na konec věty - nejčastěji se jedná o sloveso nebo odlučitelnou předponu. Tomuto jevu se říká větný rámec a k jeho tvorbě dochází v několika případech:

- **Způsobová slovesa**
  Po způsobovém slovesu jde sloveso plnovýznamové vždy na konec věty ve formě infinitivu.
  Např: Neumíme to říct. - Wir können es nicht sagen.

- **Minulý čas - perfektum**
  Perfektum se v němčině tvoří pomocí pomocného slovesa a příčestí minulého. Příčestí minulé patří na konec věty.
  Např. Neřekl jsem to. - Ich habe es nicht gesagt.

- **Budoucí čas**
  Budoucí čas se tvoří pomocným slovesem werden a infinitivem, který jde na konec věty.
  Např. Řeknu mu to. - Ich werde es ihm sagen.

- **Odlučitelné předpony sloves**
  Spousta německých sloves má odlučitelnou předponu, která se v určitých tvarech od zbytku slovesa odlučuje a patří opět na konec věty.
  Např. Zítra odjedu domů. - Morgen fahre ich nach Hause ab. (sloveso abfahren)

- **Vedlejší věty**
  Jak už bylo zmíněno, vedlejší věty mají speciální pořádek slov a určité sloveso v nich patří na konec věty.
  Např. Ptám se, jestli jsi doma. - Ich frage, ob du zu Hause bist.

K tvorbě větného rámce dochází i v dalších případech, jako je třeba trpný rod nebo čas předminulý (*plusquamperfektum*). Platí však stejná pravidla, tj. sloveso plnovýznamové nebo příčestí minulé patří na konec věty.

Vzdálenost mezi pomocným slovesem a slovesem plnovýznamovým nebo příčeštím minulým může být poměrně velká a běžné n-gramy o několika slovech nemohou tuto závislost zachytit.

3.4 Pozorování na větách

Na základě výše uvedeného popisu gramatických jevů, u kterých jsme předpokládali, že budou činit největší problémy, jsme pozorovali překladové hypotézy pro desítku vět. Pro každou větu jsme měli k dispozici 100 hypotéz. Zkoumali jsme především, v čem se jednotlivé návrhy liší, neboť to pomyslně znamená právě ty oblasti, kde si není jazykový model příliš jistý.

Z pozorování vyplynulo, že hypotézy se často liší jen ve tvarech přídavných jmen nebo členů. Modely si tak nebyly schopné poradit se skloňováním. Větný rámec se nepovedl takřka nikde, ba co víc, v některých případech plnovýznamové sloveso či příčestí minulé ve větě úplně chybělo.

3.4.1 Příklady konkrétních hypotéz

Zde uvedeme několik příkladů hypotéz, ve kterých budou červeně zvýrazněné některé gramatické chyby. Pokud se daný jev v některé další hypotéze povedl, bude označen zeleně. Zaměříme se vždy na nějaký gramatický jev, nikoliv na samotný špatný překlad některých slov.

- *Barack Obama erhält als vierte US - Präsident den Friedensnobelpreis*
- *Barack Obama wird als vierte US - Präsident den Friedensnobelpreis*
- *Barack Obama bekommt als vierte US - Präsident den Friedensnobelpreis*
- *Barack Obama erhält als vierter US - Präsident den Friedensnobelpreis*

Jak vidíme v první, druhé i třetí hypotéze, číslovka *vierte* je špatně vyskloňovaná. Má špatnou koncovku, neboť před ní nestojí určitý člen a patří k podstatnému jménu rodu mužského. Z tohoto důvodu je správně tvar označený zeleně ve čtvrté větě. Druhá hypotéza ještě navíc obsahuje sloveso *werden*, které ale pravděpodobně bylo pokusem o budoucí čas. Plnovýznamové sloveso však chybí.

- *US - Präsident Barack Obama přiletí des norwegischen in Oslo auf 26 Stunden , um sich hier als vierte US - Präsident in der Geschichte übernahm den Friedensnobelpreis .*

V této větě jsme zvýraznili dvojici *um* a *übernahm*. Pokud byla vedlejší věta uvozena spojkou *um*, pak se zřejmě mělo jednat o zkrácenou vedlejší větu a měl následovat infinitiv s *zu* na konci věty. Mimo tohoto jevu se ve větě vyskytují samozřejmě další gramatické chyby, jako třeba již zmíněné špatné skloňování.

- *Diplom , Medaille und Scheck auf 1,4 Millionen US - Dollar erhalten hat , unter anderem für die außergewöhnliche Anstrengungen zur Stärkung der Diplomatie und Zusammenarbeit zwischen den Völkern .*
- *Diplom , Medaille und Scheck auf 1,4 Millionen Dollar wird unter anderem für die außergewöhnliche Anstrengungen zur Stärkung der Diplomatie und Zusammenarbeit zwischen den Völkern .*

Zde se naopak v první hypotéze povedl větný rámec utvořením pořádku slov vedlejší věty, ačkoliv zde být neměl, neboť se o vedlejší větu nejedná. Druhá hypotéza obsahuje sloveso *werden*, které má zřejmě funkci slovesa pomocného. Infinitiv nebo příčestí minulé už ale chybí. V obou hypotézách je určité sloveso ve třetí osobě čísla jednotného, přestože takový podmět, s nímž by měl přísudek utvořit shodu, se ve větě nenachází.

- *der Präsident hat sich diesem Thema vermeiden will , weil sie erkennt , dass die Kosten übernimmt wie ein Präsident , der derzeit ein Krieg in zwei Ländern .*

V první klauzi se vyskytují dvě určitá slovesa, v poslední naopak sloveso úplně chybí. Třetí klauze je vedlejší větou a špatný pořádek slov zde staví *die Kosten* do role podmětu neshodujícího se s přísudkem, neboť *die Kosten* je pomnožné podstatné jméno a *übernimmt* je v čísle jednotném.

4. Modely s morfologickými značkami

Jak jsme popsali v předchozí kapitole, německá gramatika je díky shodě jmen, pořádku slov a tvorbě větného rámce složitá. Běžné n-gramové modely, které sledují jen posloupnosti po sobě jdoucích slov nezachycují gramatiku jako takovou. Vyzkoušíme proto, zda dopadnou lépe modely, které budeme trénovat a testovat na datech, v nichž nahradíme slova za různé morfologické značky. Budeme na nich zkoumat, jak spolu souvisí perplexita a ručně hodnocená plynulost.

Předpokladem je, že pokud nahradíme slova jejich morfologickými značkami, dojde ke zhuštění dat a model se bude snažit dosadit slovo v patřičném tvaru vycházejícím z morfologické analýzy.

4.1 Zdrojová data

Modely budeme trénovat na německých datech z WMT[1] 2012 - News Commentary (ZDROJ).

Pro otestování použijeme data z výstupů překladových systémů z WMT 2006, které se účastnily překladu z angličtiny do němčiny. Některé z překladových hypotéz obsahovaly ručně ohodnocenou plynulost překladu. Právě takové hypotézy použijeme pro zkoumání korelace perplexity a plynulosti. Celkem jich je k dispozici 2028, z toho 58 je hodnoceno dvakrát a 2 třikrát, celkem tedy 2090 hodnocení. Následující tabulka ukazuje přesné počty hypotéz:

Plynulost	**Počet hodnocení**
1	150
2	445
3	932
4	387
5	176

Počty jednotlivých plynulostí nejsou vyvážené. Díky malému vzorku ohodnocených hypotéz, především pak hodnocených plynulostí 1 a 5, nemusí mít výsledky vždy úplně vypovídající charakter. TAHLE VĚTA ZNÍ JAKO, ŽE TO BUDE K NIČEMU - JAK TO FORMULOVAT LÉPE?

Hypotézy hodnotí 400 různých vět překládaných osmi systémy. Plynulost hodnotili 4 hodnotitelé, kteří se u 58 hypotéz hodnocených dvěma hodnotiteli shodli následovně:

[1] WORKSHOP ON STATISTICAL MACHINE TRANSLATION

Shoda	Počet hypotéz	V procentech
shodli se	34	58.6 %
lišili se o 1	19	32.8 %
lišili se o 2	4	6.9 %
lišili se o 3	1	1.7 %

Dvě hypotézy hodnocené třikrát byly taktéž hodnoceny dvěma hodnotiteli, třetí hodnocení bylo vždy vykonáno jedním z nich a slouží pouze jako kontrola - ta dopadla v obou případech úspěšně, tedy udělením stejného hodnocení. U jedné hypotézy se hodnotitelé shodli, u druhé se lišili o 1.

Ačkoliv nevýhodou ručního hodnocení je vždy určitá míra subjektivity, výše uvedená tabulka uvádí nadpoloviční shodu a budeme-li brát jako uspokojivé i případy, kdy se hodnocení lišilo o 1, pak jsme dokonce lehce nad 90 %.

4.2 Princip experimentů

Princip experimentů bude následující

- na trénovacích datech provedeme morfologickou analýzu
- slova trénovacího textu nahradíme odpovídajícími morfologickými značkami
- natrénujeme standardní 6-gramový model a maxentový 6-gramový model
- stejně jako trénovací data, připravíme i data testovací
- změříme perplexitu na testovacích datech a provedeme vyhodnocení

Pro morfologickou analýzu použijeme parser ParZu[2]. Jedná se o nástroj, který kombinuje tagger Tree-Tagger (ZDROJ) a morfologický analyzátor Morphisto (ZDROJ). Pro Morphisto použijeme předkompilovaný model `morphisto-02022011.a`[3]. S využitím nástroje Morphisto uvádí ParZu přesnost 86.5 % (ZDROJ).

ParZu spouští nejprve vlastní tokenizér. Vzhledem k tomu, že data z WMT 06, která používáme, jsou již tokenizovaná, tento tokenizér vynecháme a pouze upravíme formát - jeden token na řádku, věty oddělené prázdným řádkem. Data z WMT 12 tokenizovaná nejsou, proto u nich tokenizér necháme běžet.

Příklad výstupu ParZu:

```
1  Der        der     ART   ART   Def|Masc|Nom|Sg      3  det   _  _
2  schönste   schön   ADJA  ADJA  Sup|Masc|Nom|Sg|Sw|  3  attr  _  _
3  Satz       Satz    N     NN    Masc|_|Sg            0  root  _  _
4  auf        auf     PREP  APPR  _                    3  pp    _  _
5  aller      aller   ART   PIAT  Fem|_|Sg             6  det   _  _
6  Welt       Welt    N     NN    Fem|_|Sg             4  pn    _  _
7  .          .       $.    $.    _                    0  root  _  _
```

[2]The Zurich Dependency Parser for German, formálněji známý také pod názvem Pro3GresDE

[3]ZDROJ

Pro účely následujících experimentů nás bude zajímat pátý a šestý sloupec - rozšířený slovní druh a morfologická analýza.

Nahrazení slov různými morfologickými značkami z výstupu ParZu zajišťuje program MorfModel(VYMYSLET TŘEBA NĚJAKÝ TAKOVÝ NÁZEV), který vzniknul jako součást této práce a je k dispozici na přiloženém DVD.

Standardní n-gramové modely budeme vyrábět v již zmíněném SRILM toolkitu, nástrojem `ngram-count` s výchozím nastavením. Tj. technika back-off s Good-Turing vyhlazováním. Pro trénování modelů maximální entropie (maxentových) využijeme již taktéž zmíněné rozšíření SRILM toolkitu od Tanela Alumäe a Mikko Kurima. Pro vyhlazování používá toto rozšíření popsanou metodu $\ell_1 + \ell_2^2$ regularizace.

4.3 Způsob vyhodnocení

U každého natrénovaného modelu bude změřena perplexita pro každou větu zvlášť. Výsledky pak vykreslíme do grafu společně s odpovídající plynulostí pro znázornění jejich korelace. Pro lepší znázornění bude u každé plynulosti vykreslen boxplot[4] znázorňující oblast s nejvyšším výskytem hypotéz ohodnocených danou perplexitou. Čím vyšší je plynulost, tím nižší by měla být perplexita. Jednotlivé boxploty by proto měly, co se perplexity týče, klesat. Mediány boxplotů bude ještě proložena přímka, aby byla jejich tendence více zřetelná.

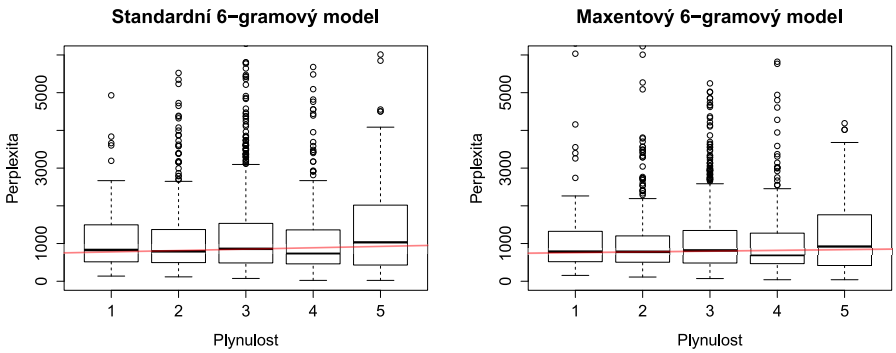
Srovnání standardních a maxentových modelů provedeme graficky umístěním dvou grafů přes sebe vykreslených odlišnou barvou. Mimo toho proveme ještě srovnání z hlediska výpočetních nároků[5]. Na závěr uvedeme shrnutí popisující naměřené hodnoty ze všech modelů.

4.4 Běžné modely se slovy

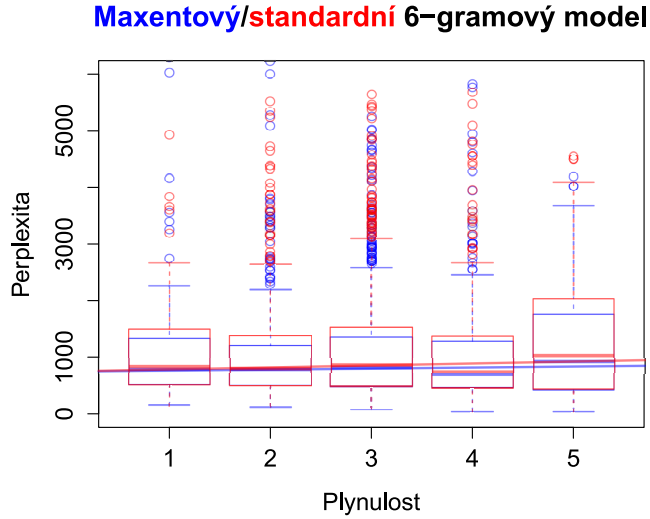
Jako první jsme zkusili natrénovat 6-gramové modely se slovy, abychom viděli, jak spolu souvisí perplexita a plynulost u takových modelů a mohli výsledek použít pro další srovnání.

[4]Boxplot (krabicový graf) - vykreslí obdélník v oblasti, kde se vyskytuje 50 % hodnot. Horní a dolní hranice obdélníku odpovídají hornímu a dolnímu kvartilu. Uprostřed obdélníku se vykresluje ještě tučně medián. Vertikálně vedou z obdélníků tzv. vousy, jejichž hranice leží v maximální (minimální) hodnotě, maximálně však v 1.5 násobku mezikvartilového rozmezí (horní - dolní kvartil) nad horním nebo pod dolním kvartilem. Body mimo tyto hranice se nazývají extrémní hodnoty a jsou vykresleny samostatně jako body.

[5]Všechny modely trénovány na netbooku Asus EEE 1201N - Intel Atom 330 - 1.6 GHz dual core, 2 GB RAM.



Z obou grafů je patrné, že plynulost nekoreluje s perplexitou tak, jak bychom chtěli. Perplexita by měla se zvyšující se plynulostí klesat - čím nižší perplexita, tím lepší a tedy i plynulejší překlad. Na obou grafech však boxploty neklesají, nýbrž kolísají. Dokonce hypotézy hodnocené plynulostí 5 mají rozsah nejčastějších perplexit nejvyšší. To ale může být částečně způsobeno malým počtem hypotéz ohodnocených plynulostí 5.



Srovnání ukazuje, že maxentový model dopadl o něco lépe, neboť jednotlivé boxploty mají nižší horní hranici nejčastějších perplexit než v případě standardních modelů. Rozdíly ve spodních hranicích jsou zanedbatelné. I proložená přímka stoupá v případě standardního n-gramového modelu více než v případě maxentového.

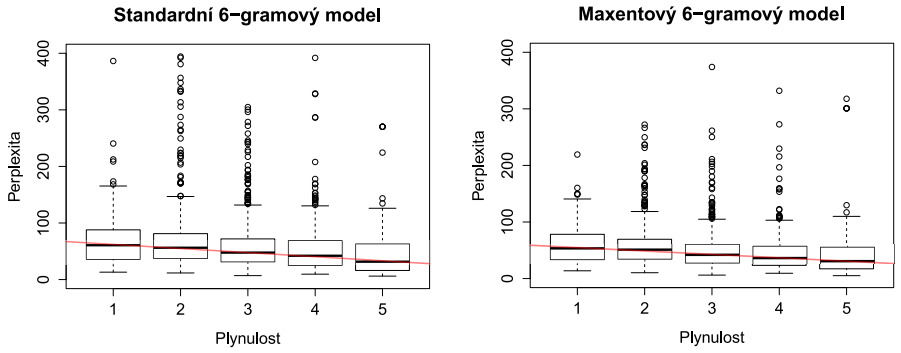
Čas nutný k natrénování se však výrazně liší - natrénování standardního n-gramového trvalo zhruba 3 minuty oproti téměř 12 hodinám u modelu maxentového.

4.5 Rozšířený slovní druh + morfologické značky

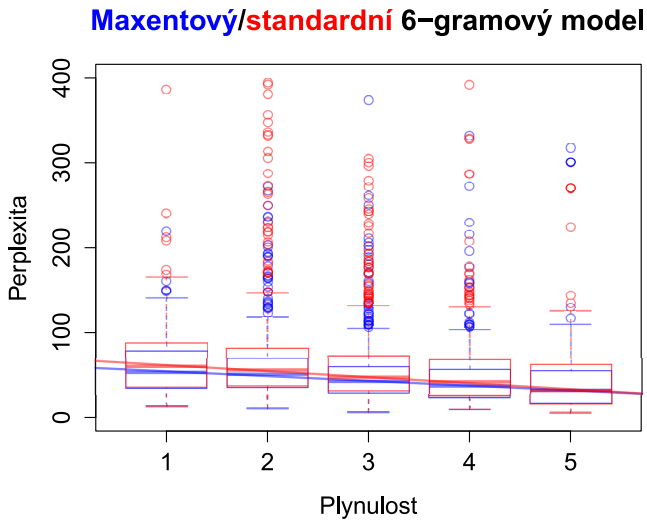
Jako první zkusíme natrénovat model, kde slova nahradíme rozšířeným slovním druhem a všemi morfologickými značkami z výstupu ParZu. Pro oddělení použijeme dvojtečku.

Příklad věty:

Die	unabhängige	Justiz	
ART:Def|Fem|Akk|Sg	ADJA:Pos|Fem|Akk|Sg|_|	NN:Fem|Akk|Sg	
und	die	freien	
KON:_	ART:Def|Neut|Akk|Pl	ADJA:Pos|Neut|Akk|Pl|_|	
Medien	zu	unterdrücken	.
NN:Neut|Akk|Pl	PTKZU:_	VVINF:_	$.:_



Oba modely dopadly lépe než modely trénované na slovech. Boxploty vykazují lehce klesavou tendenci.



Maxentové modely opět, co se perplexity týče, dopadají lépe než standardní n-gramové. Avšak proložená přímka klesá u standardních modelů strměji. Z hlediska

výpočetních nároků jsou na tom stadardní n-gramy oproti maxentovým znovu výrazně lépe - 72 sekund proti takřka 8 hodinám.

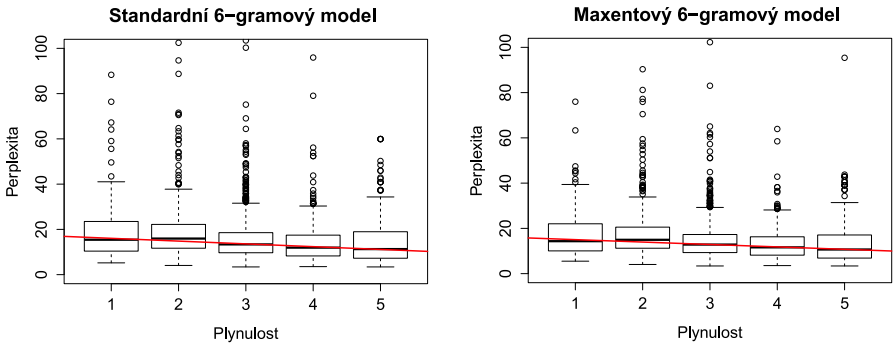
Tyto modely sice obsahují morfologickou analýzu, ale nerozumí jejímu obsahu. Nedokáží rozlišit, zda se sousední jména shodují v rodě, ale už ne v pádě apod. Natrénování maxentového modelu s rysy, které by vycházely z morfologické analýzy (rod, pád, číslo, ...), rozšíření SRILMu od Tanela Alumäe a Mikko Kurima bohužel neumožňuje a jiné dostupné toolkity, např. Maxent toolkit od LeZhanga, nejsou vhodné z hlediska výpočetních nároků na velká data. Zkusíme proto natrénovat další n-gramové modely, ve kterých nahradíme slova vždy jedním z potencionálních rysů. Takové modely by potom bylo možné kombinovat.

4.6 Rozšířený slovní druh

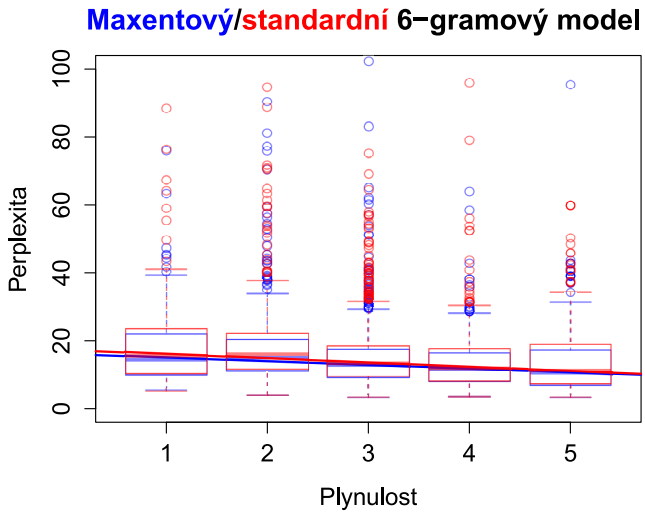
Zkusíme data ještě více zhustit a slova nahradit jen jejich rozšířeným slovním druhem.

Příklad věty:

Die	unabhängige	Justiz	
ART	ADJA	NN	
und	die	freien	
KON	ART	ADJA	
Medien	zu	unterdrücken	.
NN	PTKZU	VVINF	$.



Modely dopadly o něco hůře než v případě, kdy byl rozšířený slovní druh ještě upřesněn další analýzou. Bez dalšího určení nemůžeme např. kontrolovat správné vyskloňování. Stále je to však lepší než při natrénování na slovech.



V porovnání je na tom maxentový model z hlediska perplexity opět mírně lépe. Stejně jako proložená přímka klesá u standardního n-gramového modelu strměji.

Z hlediska výpočetních nároků to tentokrát není takový rozdíl. Standardní n-gramový model potřeboval pro natrénování 22 sekund, maxentový 14 minut.

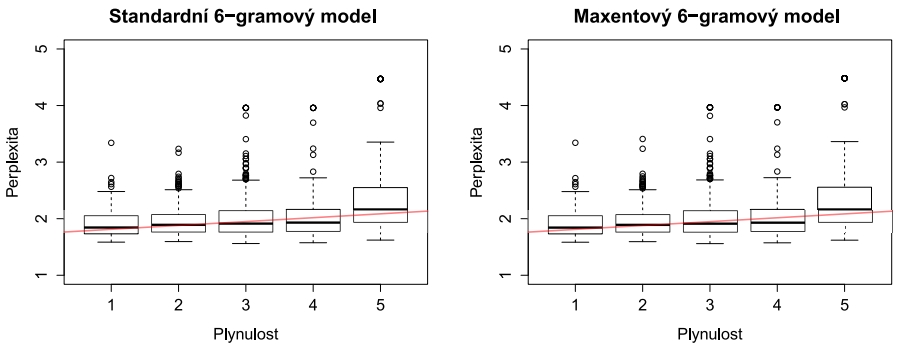
Zde je vidět, nakolik ovlivňuje velikost slovníku dobu trénování maxentových modelů. Oproti modelům se všemi morfologickými značkami potřebovaly standardní n-gramy 3.27x méně času, maxentové 34.29x, což je obrovský rozdíl.

4.7 Rod

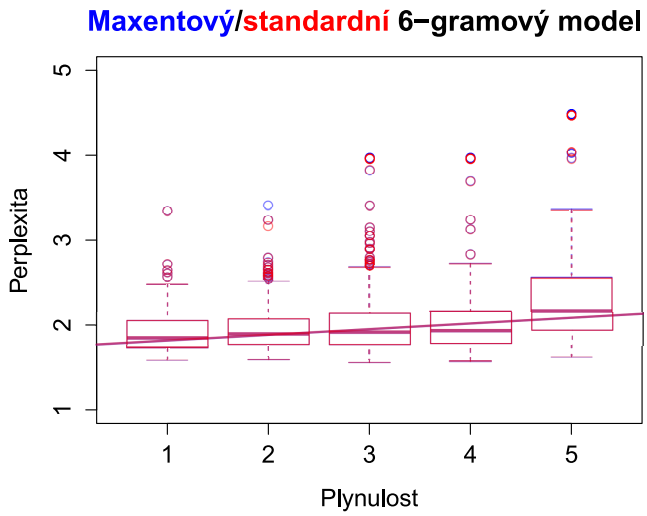
První z modelů s jedinou morfologickou značkou budou modely obsahující rod. Slova budou nahrazena znakem `w`, ke kterému se připojí patřičný rod, lze-li u slova určit. Tím dojde ke zhuštění dat a velikost slovníku se zmenší na pouhá 4 slova.

Příklad věty:

Die	unabhängige	Justiz	und	die	freien	Medien
wFem	wFem	wFem	w	wNeut	wNeut	wNeut
zu	unterdrücken	.				
w	w	w				



Modely ale dopadly přesně obráceně, než jsme chtěli. Boxploty neklesají, ale stoupají. S trochou nadsázky by se dalo říct, že zde čím je perplexita vyšší, tím je lepší plynulost. Ovšem skutečnost je taková, že perplexita pro plynulosti 1-4 vyšla velmi podobně a nejsme pouze na základě ní schopni rozlišit, o kterou plynulost by se mělo jednat.



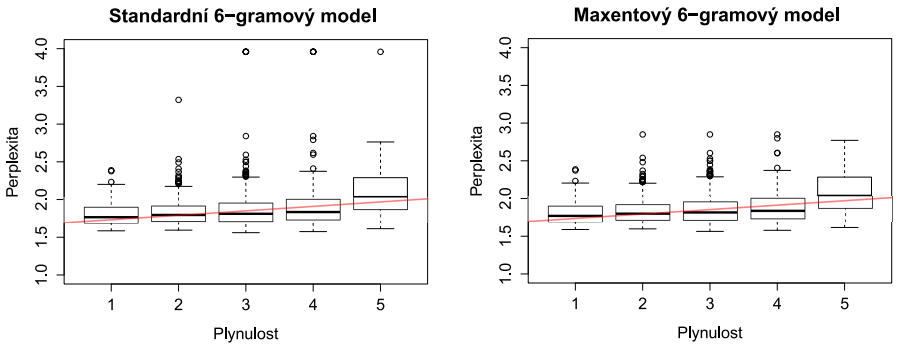
Rozdíl mezi standardními a maxentovými n-gramovými modely je prakticky neznatelný. Stejně je tomu tentokrát i u výpočetních nároků. Standardní n-gramy potřebovaly k natrénování 3 sekundy, maxentové n-gramy pak sekundy 4.

4.7.1 Rod stejný s předchozím

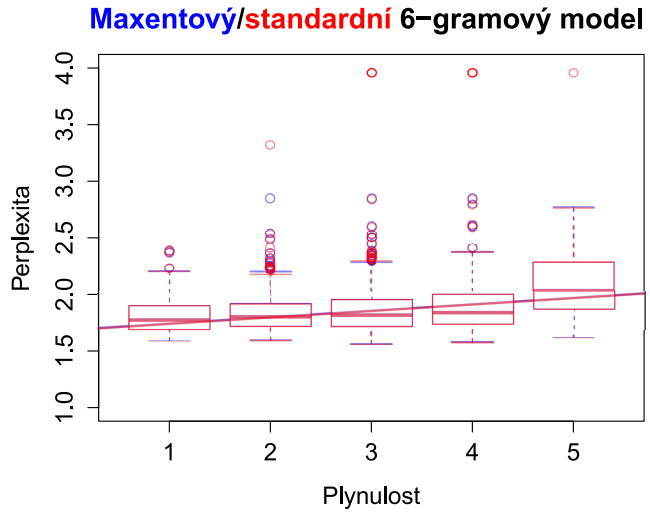
Ačkoliv modely pouze s rodem nebyly úspěšné, zkusíme ještě slova nenahrazovat jenom rodem daného slova, ale pokusíme se sledovat, zda se rody za sebou shodují. Slova tedy nahradíme opět písmenem `w`, k němuž přidáme slovo `rod`, lze-li u slova určit, a slovo `stejne`, pokud lze jednak rod u slova určit a jednak, pokud se rod shoduje s předchozím slovem.

Příklad věty:

Die	unabhängige	Justiz	und	die	freien	Medien
wrod	wstejny	wstejny	w	wrod	wstejny	wstejny
zu	unterdrücken	.				
w	w	w				



Výsledky jsou ale stejně špatné jako v případě samotného rodu. Grafy vypadají hodně podobně.



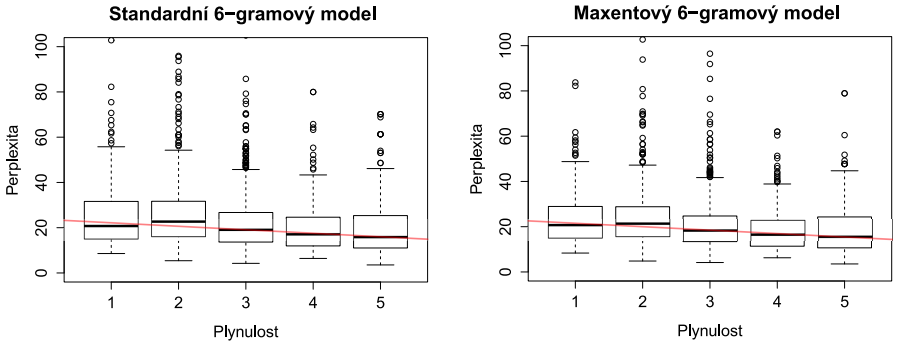
Rozdíly mezi standardním n-gramovým a maxentovým n-gramovým modelem je taktéž nepatrný. Doba nutná k natrénování obou typů byla v tomto případě shodně 3 sekundy.

4.7.2 S rozšířeným slovním druhem

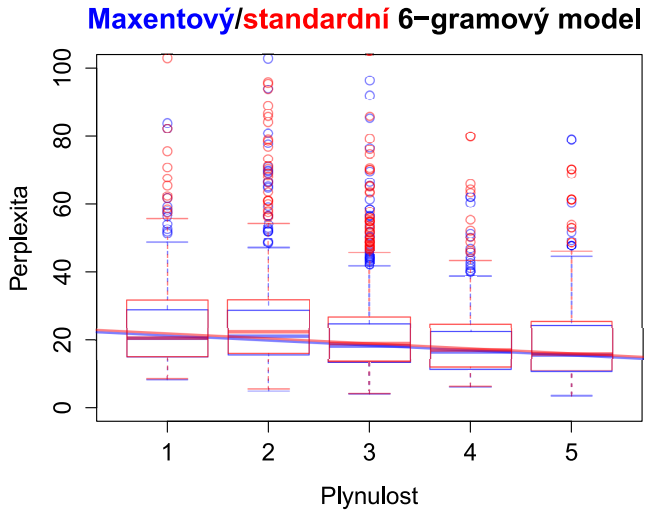
Jelikož samotný rod byla pro model nedostatečná informace, zkusíme namísto písmene `w` slova nahrazovat rozšířeným slovním druhem a k němu přidávat za dvojtečku rod.

Příklad věty:

Die	unabhängige	Justiz	und	die	freien	Medien
ART:Fem	ADJA:Fem	NN:Fem	KON	ART:Neut	ADJA:Neut	NN:Neut
zu	unterdrücken	.				
PTKZU	VVINF	$.				



S rozšířeným slovním druhem už modely dopadly lépe, přesto perplexita s rostoucí plynulostí neklesá nijak výrazně, což dokazuje proložená přímka.



Ve srovnání jsou oba typy modelů na tom podobně, maxentové dostávaly jen o něco málo nižší perplexitu. Z hlediska výpočetní náročnosti je ale rozdíl velký - 36 sekund standardní n-gramový model naproti 27 minutám u modelu maxentového.

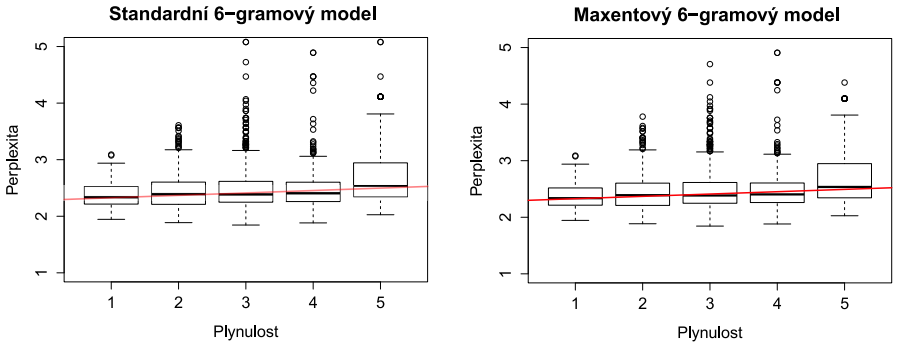
4.8 Číslo

Stejné modely zkusíme natrénovat i v případě čísla. Jako první znovu zkusíme, zda bude modelu postačovat informace pouze o čísle daného slova tj. `wSg`, `wPl` nebo `w`.

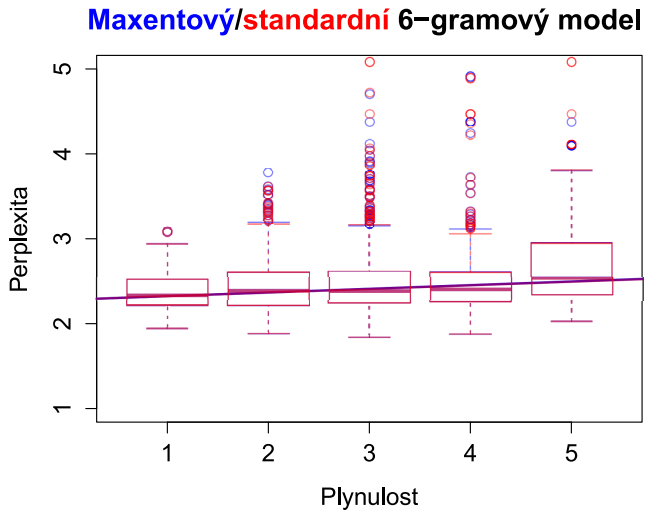
Příklad věty:

Die	unabhängige	Justiz	und	die	freien	Medien
wSg	wSg	wSg	w	wPl	wPl	wPl

zu	unterdrücken	.
w	w	w



Výsledky ale dopadly obdobně špatně jako v případě rodu. Pomocí perplexity nejsme schopni rozlišit, o jakou plynulost by se mělo jednat.



Oba typy modelů dopadly takřka stejně. Stejná byla i doba nutná k natrénování - shodně po třech sekundách.

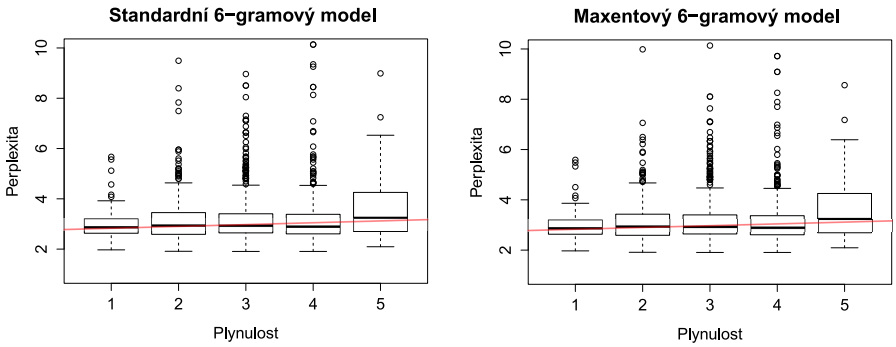
4.8.1 Přidání osoby

Číslo se bude často pojit s nějakou osobou. Zkusíme proto tuto informaci k číslu přidat. Slovo nahradíme písmenem `w`, poté bude následovat osoba a číslo, jdou-li u daného slova určit.

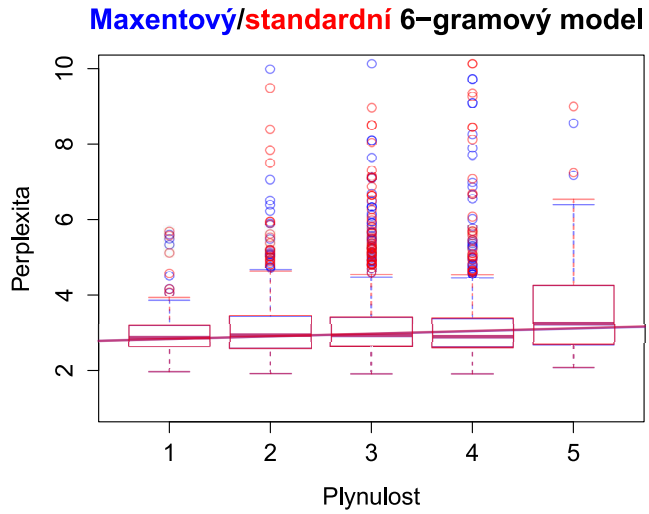
Příklad věty:

Zur	Belohnung	erhielt	Pakistan	von	Amerika
w	wSg	w3Sg	wSg	w	wSg

finanzielle	Unterstützung	und	Waffen	.
wSg	wSg	w	wPl	w



Modely ale nedopadly o nic lépe. Proložené přímky opět mírně stoupají, namísto aby klesaly.



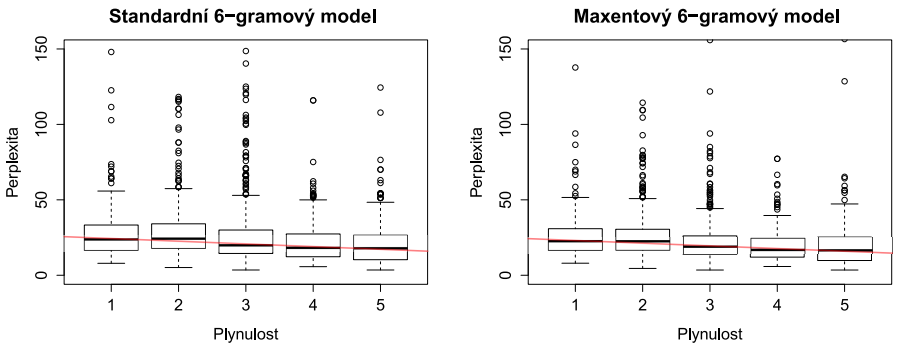
Ve srovnání obou typů modelů opět nejsou patrné výrazné rozdíly. Z hlediska výpočetních nároků se ale tentokrát trochu liší. Standardní n-gramové potřebovaly k natrénování 4 sekundy, maxentové 11 sekund.

4.8.2 S rozšířeným slovním druhem

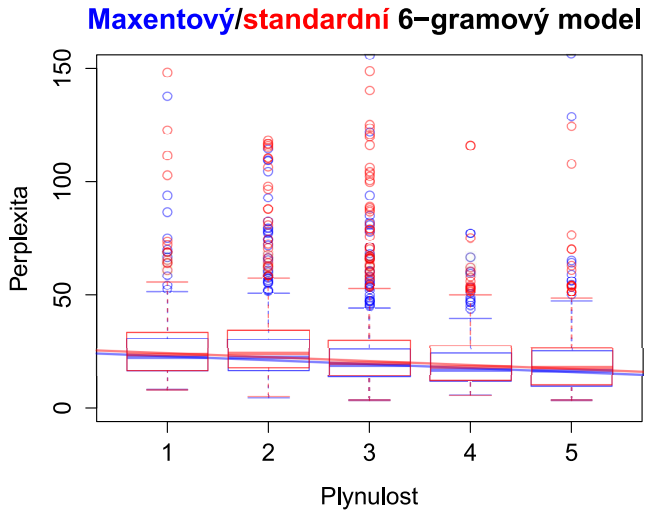
Vzhledem k tomu, že přidání osoby žádné patrné zlepšení nepřineslo, zkusíme znovu přidat rozšířený slovní druh. Slova tedy budeme nahrazovat jejich druhem a číslem, lze-li určit.

Příklad věty:

Zur	Belohnung	erhielt	Pakistan	von	Amerika
APPRART	NN:Sg	VVFIN:Sg	NE:Sg	APPR	NE:Sg
finanzielle	Unterstützung	und	Waffen	.	
ADJA:Sg	NN:Sg	KON	NN:Pl	$.	



Modely s rozšířeným slovním druhem dopadly znovu lépe. Ovšem výsledky stále nejsou nijak dobré.



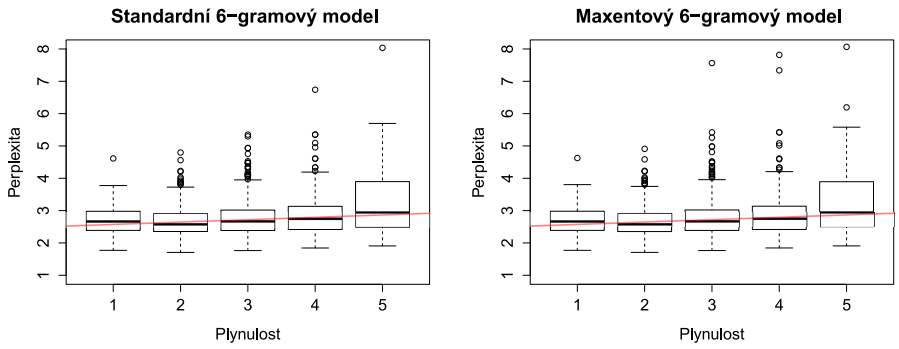
V porovnání dostávaly maxentové modely o něco nižší perplexitu. Čas potřebný k natrénování byl u standardních n-gramových modelů 33 s, u maxentových 24 minut.

4.9 Pád

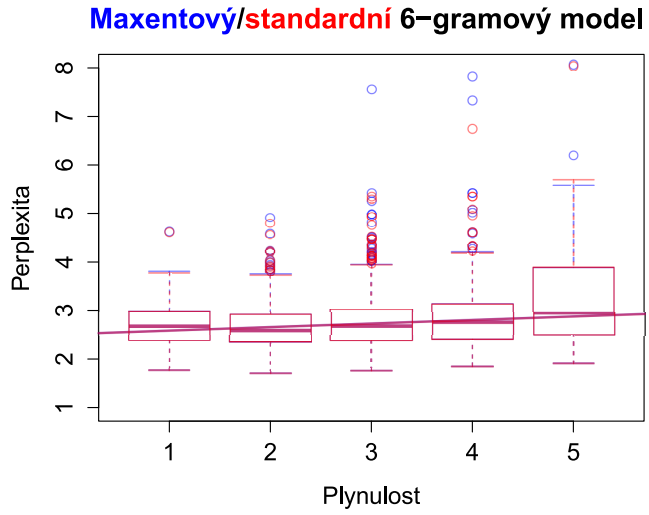
Jako poslední zkusíme ještě natrénovat modely, kde slova nahradíme znovu písmenem `w` a přidáme k němu pád, lze-li u daného slova určit.

Příklad věty:

Zur	Belohnung	erhielt	Pakistan	von	Amerika
wDat	w	w	wNom	wDat	wDat
finanzielle	Unterstützung	und	Waffen	.	
wAkk	wAkk	w	wAkk	w	



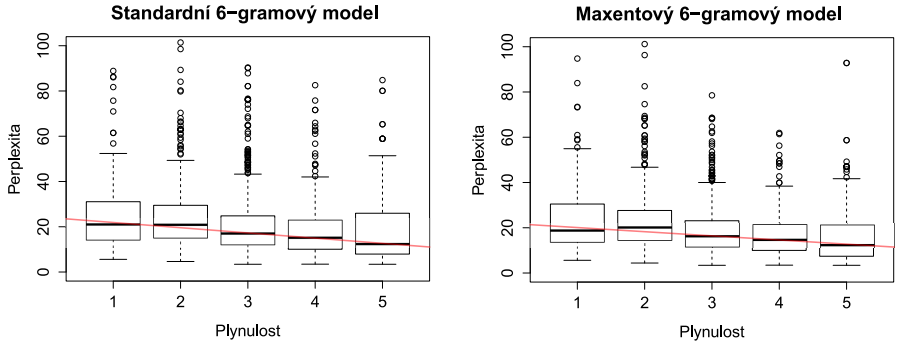
Výsledky opět nejsou dobré a jen potvrzují, že poskytnutí modelu značek pouze z jedné morfologické kategorie je nedostatečná informace.



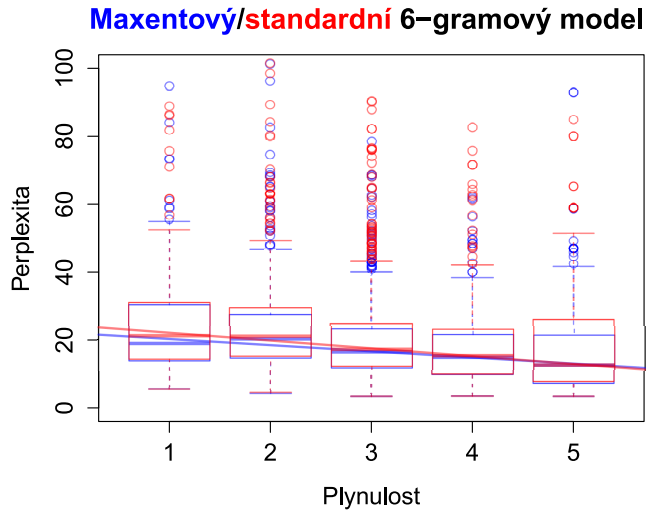
V porovnání jsou taktéž standardní n-gramové modely s modely maximálně entropie srovnatelné, bez větších rozdílů. Natrénování trvalo shodně 4 sekundy.

4.9.1 S rozšířeným slovním druhem

Jako v případech předchozích modelů se značkami z jedné morfologické kategorie, zkusíme přidat k pádu ještě rozšířený slovní druh.



Zlepšení je znatelné a podobá se modelům s rozšířeným slovním druhem a všemi morfologickými značkami. Výsledky jsou prozatím nejlepší ze všech modelů se značkami jedné morfologické kategorie s rozšířeným slovním druhem.



Maxentové modely dopadly o něco lépe a zvláště hypotézy hodnocené plynulostí 5 posunuly v perplexitě níže, což je správně. Nicméně proložená přímka je strmější u standardních n-gramů. Výpočetní nároky se ale výrazně liší - 34 sekund v případě standardních n-gramových modelů oproti 27 minutám v případě modelů maxentových.

4.10 Shrnutí

Zde uvedeme všechny naměřené hodnoty do tabulky. Tabulka bude obsahovat od každého modelu Pearsonův korelační koeficient[6], směrnici přímky proložené mediány boxplotů a čas potřebný k natrénování.

[6]Udává vztah mezi dvěma veličinami. Nabývá hodnot $< -1, 1 >$, přičemž 1 značí závislost přímou a -1 závislost nepřímou. Hodnota 0 indikuje, že vztah mezi veličinami nelze vyjádřit lineární funkcí.

Model	Pearsonův koeficient		Směrnice přímky		Čas trénování	
	standardní	maxentový	standardní	maxentový	standardní	maxentový
Slova	0.05	0.05	33.53	17.43	3 min	12 hod
RSD + všechny značky	-0.03	0.04	-7.00	-5.54	72 s	8 hod s
RSD	0.05	0.03	-1.23	-1.07	22 s	14 min
Rod	0.15	0.15	0.07	0.07	3 s	4 s
Rod stejný s předchozím	0.15	0.14	0.06	0.06	3 s	3 s
RSD + rod	0.04	0.03	-1.42	-1.41	36 s	27 min
Číslo	0.07	0.10	0.04	0.04	3 s	3 s
Osoba + číslo	0.06	0.09	0.07	0.07	4 s	11 s
RSD + číslo	0.03	0.02	-1.70	-1.70	33 s	24 min
Pád	0.15	0.15	0.07	0.07	4 s	4 s
RSD + pád	0.04	0.02	-2.25	-1.78	34 s	27 min

U Pearsonova koeficientu bychom rádi dosáhli hodnoty blížící se k -1. Jedinou zápornou hodnotu má ale jen standardní n-gramový model natrénovaný na rozšířeném slovním druhu a všech morfologických značkách. Směrnice přímky by měla být záporná, aby přímka klesala. Čím menší bude, tím bude klesat strměji. Opět je hodnota nejnižší u standardního modelu se slovním druhem a všemi značkami.

Ačkoliv modely maximální entropie obvykle dostávaly nižší perplexitu než standardní modely, korelace perplexity a ručně hodnocené plynulosti vycházela lépe u standardních modelů. Trénování modelů maximální entropie trvalo ve většině případů mnohonásobně déle a nepřineslo pro naše experimenty žádné výrazné zlepšení.

5. Modely s vlastní množinou rysů

Problémy s německou gramatikou jsme se prozatím snažili řešit nahrazením slov morfologickými značkami. Modely s rozšířeným slovním druhem + morfologická analýza dopadly sice lépe než běžné modely trénované na slovech, přesto zlepšení není nijak výrazné. V následující kapitole se proto pokusíme upustit od n-gramů a postihnout gramatiku z jiné stránky - vlastní množinou rysů.

5.1 Zdrojová data

Pro následující experimenty používáme stejná data s ručně hodnocenou plynulostí jako v předchozí kapitole. Zde jsme je rozdělili na dva díly. Polovina tj. 1045 překladových hypotéz se použije jako vývojová sada a druhá polovina jako sada testovací. Hypotézy byly rozděleny s ohledem na hodnocení plynulosti tak, aby vývojová i testovací množina vět obsahovala stejný počet hypotéz hodnocených plynulostí 1, 2, .., 5 (až na liché počty hypotéz některých plynulostí). Následující tabulka ukazuje přesné počty hypotéz a jejich rozdělení:

Plynulost	**Celkem hypotéz**	**Vývojová sada**	**Testovací sada**
1	150	75	75
2	445	222	223
3	932	466	466
4	387	194	193
5	176	88	88
CELKEM	**2090**	**1095**	**1095**

Vzhledem k tomu, že budou rysy vycházet z německé gramatiky, budeme často potřebovat znát hranice klauzí dané věty. Určit klauze z hypotézy, která není gramaticky správně, je však obtížné - parser je v takovém případě zmatený a neudělá větný rozbor správně. Na základě toho používáme identifikované klauze z výchozího anglického textu, který je gramaticky správně, a na německé je pak převádíme pomocí zarovnání na úrovni slov.

5.2 Vlastní rysy

Vlastní množina rysů sestává především z gramatických jevů, které n-gramy nemají možnost zachytit. Jedná se hlavně o tvorbu větného rámce. Mimo to ale budeme zkoumat, zda se ve větě třeba nevyskytuje více určitých sloves nebo naopak sloveso úplně chybí.

Celkem jsme sestavili 16 následujících rysů, u kterých vysvětlíme, jak fungují a co kontrolují, neboť ačkoliv jejich názvy intuitivně funkci napovídají, může být omezena jen na některé případy.

1. `chybi_infszu` - kontroluje, zda byla klauze uvozena spojkou vyžadující infinitiv s zu (rozšířený slovní druh *KOUI*[1]), typicky se jedná o zkrácené vedlejší věty, a sleduje, zda ve větě takový infinitiv byl

2. `chybi_podmet` - označuje skutečnost, že se v klauzi nevyskytlo slovo v prvním pádě

3. `chybi_vfin` - v klauzi chybí určité sloveso

4. `chybi_sum` - sčítá hodnoty všech rysů typu `chybi_*`

5. `inf_po_vm_neni_na_konci` - kontroluje, zda se za infinitivem po modálním slovese nevyskytlo ještě další slovo mimo sloves

6. `infszu_neni_na_konci` - byla-li spojka vyžadující infinitiv s zu a zároveň se za infinitivem ještě vyskytlo další slovo mimo sloves

7. `vv_sloveso_neni_na_konci` - po podřadících spojkách (*KOUS*, *PRELS*, *PRELAT*) kontroluje, zda po určitém slovesu nešlo ještě další slovo mimo sloves

8. `pp_neni_na_konci` - pokud věta obsahovala pomocné sloveso, očekáváme příčestí minulé a kontrolujeme proto, zda se za ním ještě nevyskytlo další slovo mimo sloves

9. `neni_na_konci_sum` - sčítá hodnoty všech rysů typu `*_neni_na_konci`

10. `pp_bez_av` - bylo-li ve větě příčestí minulé bez pomocného slovesa a zároveň nebyla před příčestím minulým souřadící spojka, která by mohla oddělovat dvě věty se dvěma příčestími minulými, ale pomocným slovesem jen v první z nich (*např.: Ich habe gekocht und gelernt.*)

11. `neshoda_podmet_prisudek` - sledujeme výskyt prvního slova v nominativu nebo prvního slovesa, od těchto prvních výskytů si zapamatujeme číslo a osobu (u podstatných jmen ručně nastavíme, že se jedná o třetí osobu), pokud se nenajde shoda v čísle a osobě mezi prvním nalezeným nominativem a slovesem, pak daná klauze dostane tento rys

12. `vice_osob` - funguje stejně jako předchozí, jenom s jiným vyhodnocením - tj. tehdy, když po první nalezené osobě nalezneme ještě další (samozřejmě v prvním pádě)

13. `vice_vfin` - indikuje výskyt více určitých sloves v jedné klauzi

14. `vice_sum` - sčítá hodnoty všech rysů typu `vice_*`

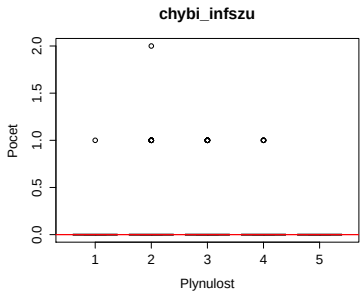
15. `root` - projde větný rozbor z výstupu ParZu a spočítá počet kořenů, je-li totiž věta gramaticky správně, nalezneme kořen pouze jeden, v opačném případě je jich více a představují pomyslný počet chyb ve větě

16. `sum` - sčítá hodnoty všech předchozích rysů včetně rysu `root`

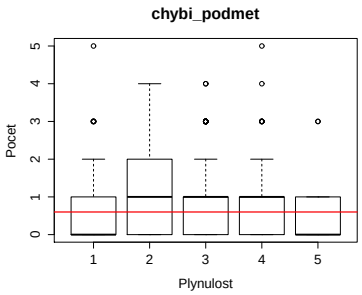
[1]ParZu používá pro označení rozšířeného slovního druhu Stuttgart/Tübinger Tagsets ZDROJ http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/stts.asc

Pomocí programu Chyby, který vzniknul k této práci, jsme změřili na vývojových datech korelaci každého rysu s ručně hodnocenou plynulostí. Každý rys kromě rysů součtových a rysu `root` jsou určovány pro každou klauzi zvlášť a mohou v ní vždy dostat jen hodnotu `true`/`false`. Rysy celé věty jsou pak součtem hodnot rysů ze všech klauzí - přičteme vždy jedničku, když daný rys nabyl v klauzi hodnoty `true`.

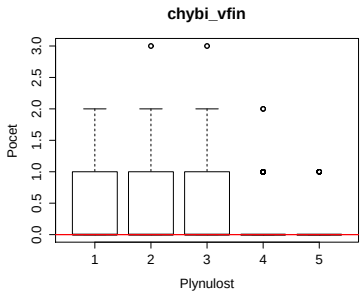
5.2.1 Rysy typu chybi_*



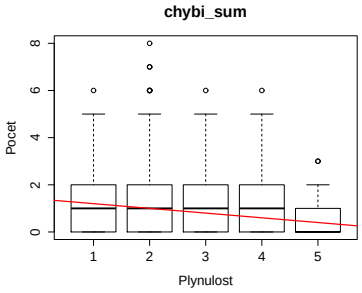
Obrázek 5.1: Korelace hodnoty rysu chybi_infszu a plynulosti



Obrázek 5.2: Korelace hodnoty rysu chybi_podmet a plynulosti



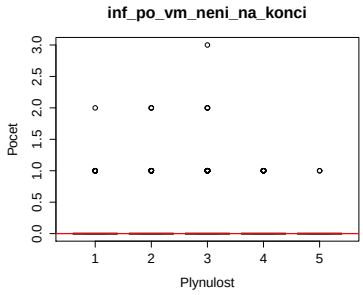
Obrázek 5.3: Korelace hodnoty rysu chybi_vfin a plynulosti



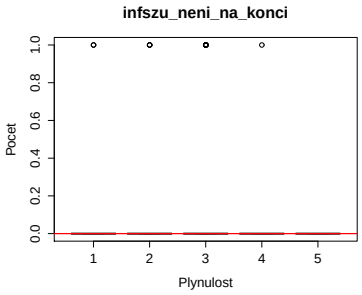
Obrázek 5.4: Korelace součtového rysu chybi_sum a plynulosti

Jednotlivé rysy samostatně neukázaly závislost s plynulostí (obrázky 5.1, 5.2, 5.3). V součtu je u hypotéz s plynulostí 5 medián na nule, což je správně, neboť tyto hypotézy by měly být úplně bez chyb (obrázek 5.4). Hodnoty nad nulou jsou způsobené jednak díky chybám v hranicích klauzí, protože se v nich spoléháme na identifikaci klauzí anglických a na zarovnání slov. Jednak také díky chybám při morfologické analýze z ParZu a samozřejmě také díky speciálním případům, se kterými náš program na vyhledávání hodnot rysů, nepočítá.

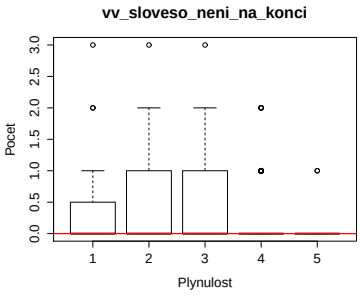
5.2.2 Rysy typu *_neni_na_konci



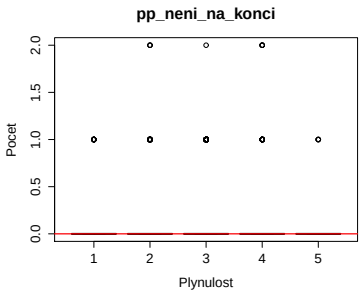
Obrázek 5.5: Korelace hodnoty rysu inf_po_vm_neni_na_konci a plynulosti



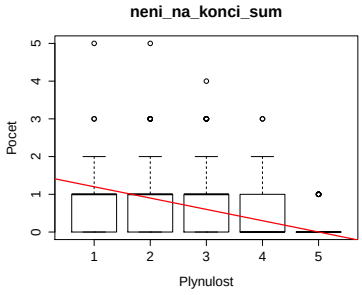
Obrázek 5.6: Korelace hodnoty rysu infszu_neni_na_konci a plynulosti



Obrázek 5.7: Korelace hodnoty rysu vv_sloveso_neni_na_konci a plynulosti



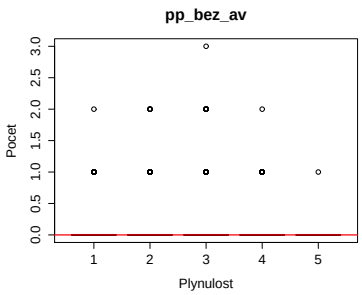
Obrázek 5.8: Korelace hodnoty rysu pp_neni_na_konci a plynulosti



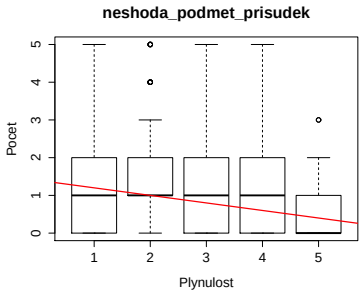
Obrázek 5.9: Korelace součtového rysu neni_na_konci_sum a plynulosti

Zde je situace podobná jako u předchozí skupiny rysů. Jednotlivé rysy samostatně (obrázky 5.5, 5.6, 5.7, 5.8) mají mediány na nule. U součtového rysu (obrázek 5.9) alespoň hypotézy s hodnocením plynulosti 5 dostávaly oproti ostatním plynulostem častěji nulu.

5.2.3 Rysy pp_bez_av a neshoda_podmet_prisudek



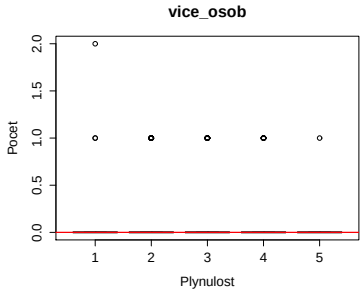
Obrázek 5.10: Korelace hodnoty rysu pp_bez_av a plynulosti



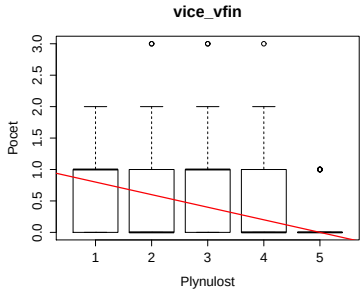
Obrázek 5.11: Korelace hodnoty rysu neshoda_podmet_prisudek a plynulosti

Oba rysy opět nevykazují samostatně souvislost s plynulostí (obrázky 5.10, 5.11).

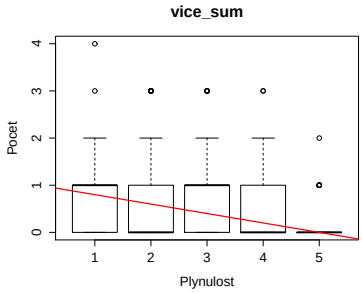
5.2.4 Rysy typu vice_*



Obrázek 5.12: Korelace hodnoty rysu vice_osob a plynulosti



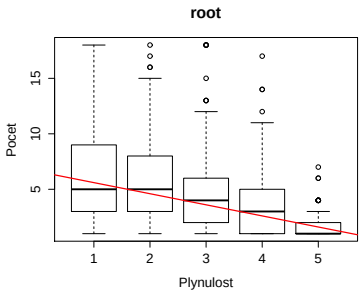
Obrázek 5.13: Korelace hodnoty rysu vice_vfin a plynulosti



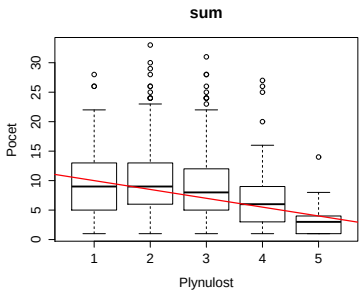
Obrázek 5.14: Korelace součtového rysu vice_sum a plynulosti

Shodně dopadly i rysy typu `vice_*`. Mediány rysu `vice_osob` jsou nulové (obrázek 5.12). U rysu `vice_vfin` mediány kolísají mezi nulou a jedničkou (obrázek 5.13), stejně jako i u součtového rysu (obrázek 5.14).

5.2.5 Rysy sum a root



Obrázek 5.15: Korelace hodnoty rysu root a plynulosti



Obrázek 5.16: Korelace hodnoty rysu sum a plynulosti

Oba rysy vykazují závislost jejich hodnot s plynulostí (obrázky 5.15, 5.16). Mediány mají pro různé plynulosti odlišné hodnoty (až na plynulosti 4 a 5). V následujících experimentech proto zkusíme, zda se dá predikovat plynulost třeba jen na základě mediánu vypočteného pro jednotlivé plynulosti v trénovacích datech.

5.3 Princip experimentů

- popsat identifikaci klauzí dle alignmentu a rysy

Princip experimentů bude následující:

- identifikovat německé klauze v hypotézách na základě anglických s využitím zarovnání na úrovni slov
- provést morfologickou analýzu a pokusit se o větný rozbor hypotéz

- pokusit se vyhledat některé gramatické chyby a použít je jako rysy
- natrénovat maxentový model s těmito rysy
- provést stejný postup na testovacích datech a model otestovat

Pro identifikaci klauzí vzniknul k bakalářské práci program Klauze, který na základě anglických klauzí a zarovnání na úrovni slov identifikuje klauze německé a vydá na výstup jejich hranice. Ze zarovnání bereme jen ty nejvíce jisté shody (int?), neboť nám nejde o kompletní zarovnání, ale jen o hranice klauzí. Na základě anglických klauzí jsou nejdříve slova roztříděna podle zarovnání do klauzí. Slova v klauzích se potom setřídí podle pořadí, v jakém byly v hypotéze. Poté se dle minimálního a maximálního indexu slova (počítáno zleva od nuly) rozhoduje, zda se jedná o větu vloženou nebo zda má začínat až pozdějším slovem, neboť se kryje s větou předchozí. Tímto způsobem se určí hranice klauzí tak, aby zahrnuly všechny slova ve větě.

Na morfologickou analýzu a větný rozbor použijeme opět parser ParZu. Tentokrát ale využijeme ještě dalších informací, které poskytuje. Zkusíme využít v náš prospěch i skutečnosti, že u gramaticky špatné věty postaví větný rozbor chybně.

Pro hledání chyb v jednotlivých klauzích jsme stvořili program Chyby. Ten se na základě provedené analýzy z ParZu a hranicí klauzí snaží určit některé gramatické chyby, pro celou větu pak vydá tyto chyby jako součet všech klauzí. Výstupem je soubor ve formátu ihned použitelném pro natrénování v Maxent Toolkitu od Le Zhanga. Za pomocí rysů vycházejících z gramatiky se budeme snažit predikovat plynulost. Výstupní formát bude tedy např. následující:

```
1  chybi_vfin:2  pp_bez_av:3  chybi_infszu:1
4  chybi_vfin:0  pp_bez_av:0  chybi_infszu:1
2  chybi_vfin:1  pp_bez_av:2  chybi_infszu:0
```

Jak vidíme z příkladu, druhý řádek nám říká, že hypotéza hodnocená plynulostí 4 má hodnotu rysu `chybi_vfin` 0, `pp_bez_av` 0 a `chybi_infszu` 1. Význam jednotlivých rysů bude ještě dále upřesněn, zde prozatím slouží jako příklad.

5.4 Způsob vyhodnocení

přesnost, lišící se o jedna + přehled nabízených fluency

5.5 Modely se všemi rysy

5.6 Modely se všemi rysy kromě rysů součtových

5.6.1 Bez rysu root

5.7 Modely se součtovými rysy

5.7.1 S rysem root

5.8 Modely s rysem root

5.9 Modely s rysem sum

5.9.1 S rysem root

NAPSAT, ŽE SPOUSTU CHYB V PARZU ZAPŘÍČIŇUJE LOWERCASE!!!
zahlen x Zahlen

Seznam použité literatury

Seznam tabulek

Seznam použitých zkratek

Přílohy