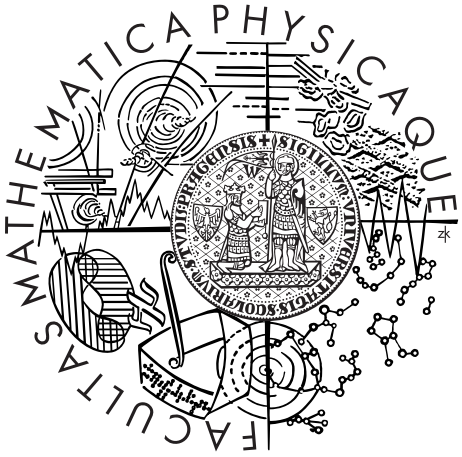
Univerzita Karlova v Praze

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.

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}
- $V_{w_{i-1}}$ 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 konkrétních větách

ruční analýza vět, přehled počtů chyb a vyhodnocení, které dělají překladu největší problémy ? POUŽÍT POZOROVANÉ N-BEST LISTY? ?

4. Experimenty

Data pro experimenty pochází z výstupů překladových systémů z WMT[1] 2006, které se účastnily překladu z angličtiny do němčiny. Některé z překladových hypotéz obsahovaly ručně ohodnocenou plynulost překladu. Právě tyto ohodnocené hypotézy používáme pro naše experimenty. 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, mohou být výsledky zkreslené malým vzorkem dat.

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

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. V jednom případě se hodnotitelé shodli, v druhém se lišili o 1.

Každý model natrénujeme jak metodou maximální entropie, tak standardními n-gramy. Pro trénování bude použit SRILM toolkit[2], který umí natrénovat standardní n-gramové modely, a jeho rozšíření od Tanela Alumäe[3] pro natrénování modelů maximální entropie (maxentových).

4.1 Způsob vyhodnocení

U každého natrénovaného modelu bude změřena perplexita pro každou větu zvlášť. Výsledky pak vykreslíme do grafu společně s odpovídající plynulostí pro

[1] WORKSHOP ON STATISTICAL MACHINE TRANSLATION

[2] The SRI Language Modeling Toolkit

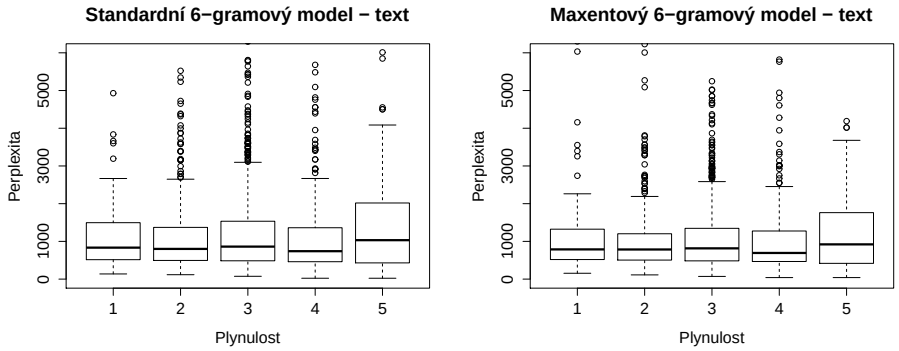
[3] Od verze 1.7.1 je toto rozšíření již součástí základní instalace SRILM toolkit.

znázornění jejich korelace. Pro lepší znázornění bude u každé plynulosti vykreslen boxplot[1] 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. Srovnání provedeme graficky umístěním dvou grafů přes sebe vykreslených odlišnou barvou.

Mimo srovnání korelace perplexity a ručně hodnocené plynulosti překladu budou oba typy modelů (standardní n-gramové a maxentové) srovnány z hlediska výpočetních nároků.

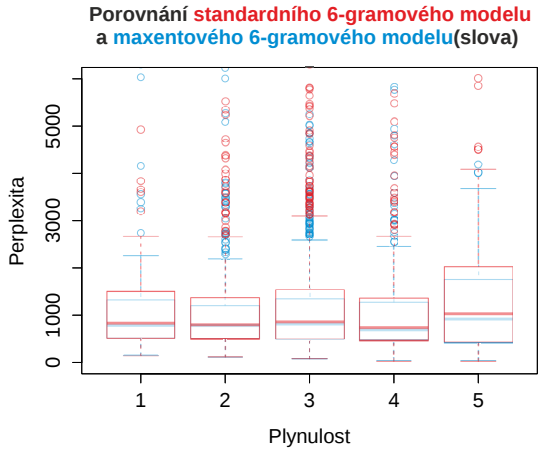
4.2 Běžné modely se slovy

Pro srovnání jsme zkusili natrénovat běžné 6-gramové modely se slovy, abychom viděli, jak spolu souvisí perplexita a plynulost. Jako trénovací data byl použit korpus z WMT[1] 2012 - News Commentary. Ten obsahuje 158 840 německých vět a je používán i u všech následujících modelů s morfologickými značkami.



Z obou grafů je patrné, že plynulost nekoreluje s perplexitou tak, jak jsme předpokládali. 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.

[1]Boxplot (krabicový graf) - vykreslí obdélník v oblasti, kde se vyskytuje 50 % hodnot. Horní a dolní hranice odpovídají hornímu a dolnímu kvartilu. Uprostřed boxplotu se vykresluje ještě tučně medián. Vertikálně vedou z obdélníků tzv. vousy, jejichž hranice leží v maximální (minimální) hodnotě, maximálně však v 1.5 násobku horní nebo dolní hranice. Body mimo tyto hranice se nazývají extrémní hodnoty



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.

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

4.3 Modely se slovy nahrazenými morfologickými značkami

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 morfologické značky.

Pro morfologickou analýzu použijeme parser ParZu[2]. Jedná se o nástroj, který kombinuje tagger Tree-Tagger a morfologický analyzátor Morphisto. ParZu za pomocí těchto dvou nástrojů vybere ze všech variant, které nabízejí, jedinou z nich a vrátí navíc větný rozbor.

připsat, že se používá předkompilovaný model morphista morphisto-02022011.a

ParZu spouští nejprve vlastní tokenizér. Vzhledem k tomu, že data z WMT 06, které používáme, jsou již tokenizovaná, tento tokenizér vynecháme a pouze upravíme formát - jeden token na řádku, věty oddělené prázdným řádkem.

Příklad výstupu ParZu:

[2]The Zurich Dependency Parser for German

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	_	_

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

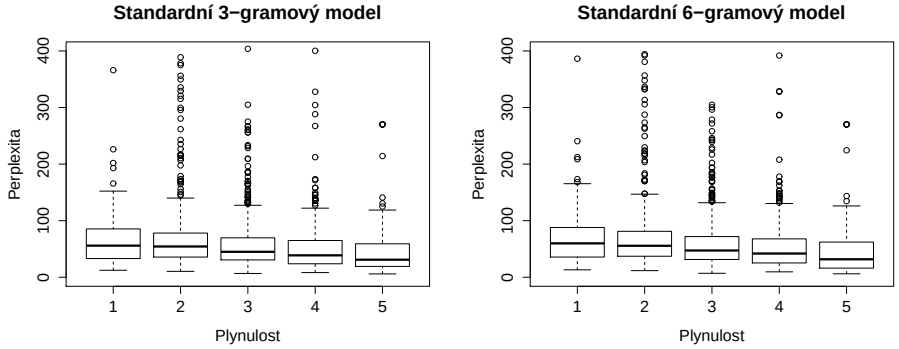
4.3.1 Rozšířený slovní druh + morfologické značky

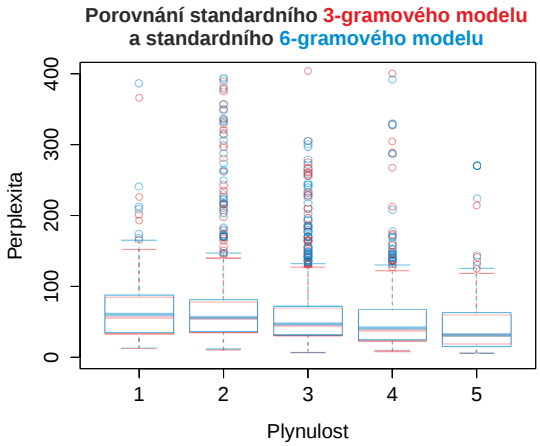
Jako první zkusíme natrénovat model, kde slova nahradíme rozšířeným slovním druhem a všemi morfologickými značkami z výstupu ParZu. Pro oddělení použijeme dvojtečku.

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:_	$.:_

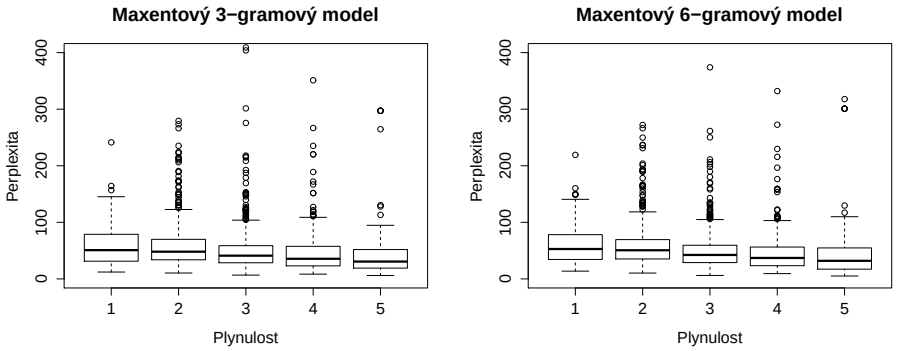
Běžné modely se slovy jsme trénovali jako 6-gramy. Zde se zkusíme podívat, jak moc se liší 3- a 6-gramové modely.

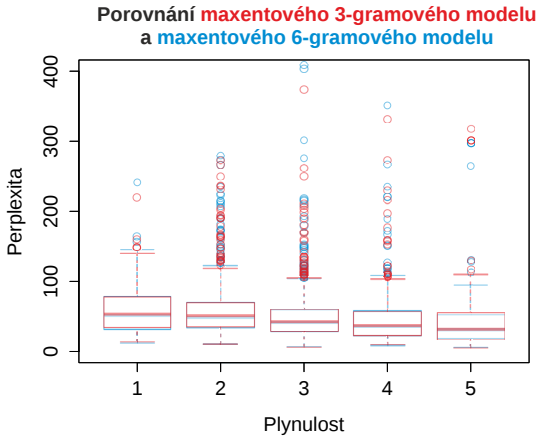


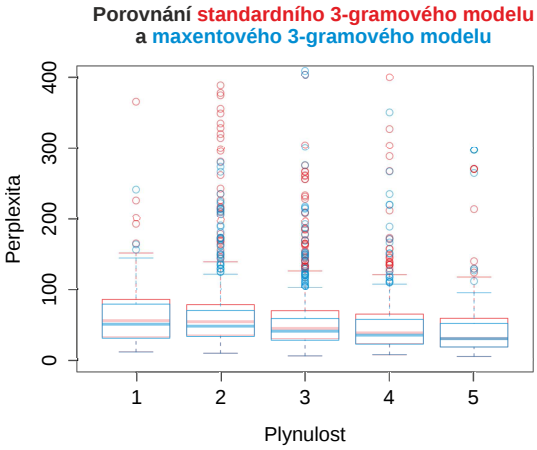


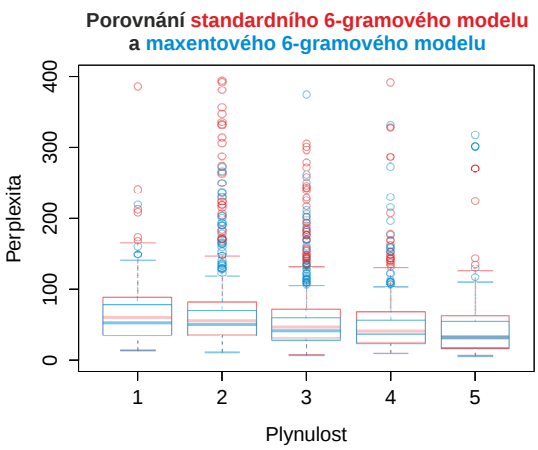
* graf je blbě - prohozené barvy

Standardní 6-gramové modely dopadají o něco lépe, rozdíly však nejsou nijak výrazné. Podobné je to i u maxentových modelů:









Maxentové modely opět dopadají lépe než standardní n-gramové. Celkově dopadly všechny modely lépe než běžné modely se slovy, neboť zde mají boxploty už klesavou tendenci.

Z hlediska výpočetních nároků TABULKA?

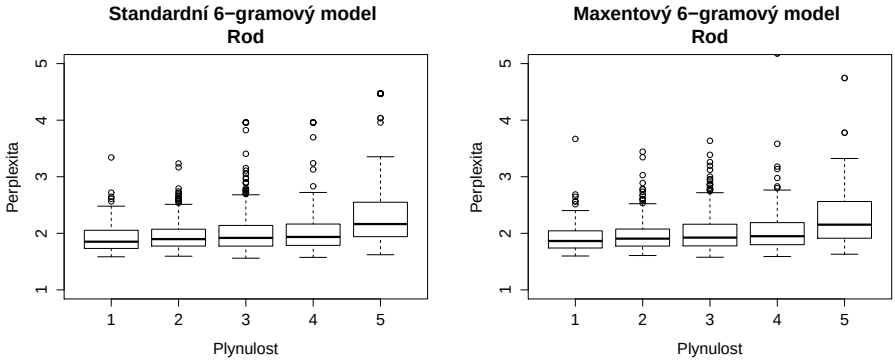
Tyto modely sice obsahují morfologickou analýzu, ale nerozumí gramatice jako takové. Nedokáží rozlišit, zda se sousední jména shodují v rodě, ale už ne v pádě apod. Natrénování maxentového modelu s rysy, které by vycházely z morfologické analýzy (rod, pád, číslo, ...), rozšíření SRILMu od Tanela Alumäe,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

4.3.2 Rod

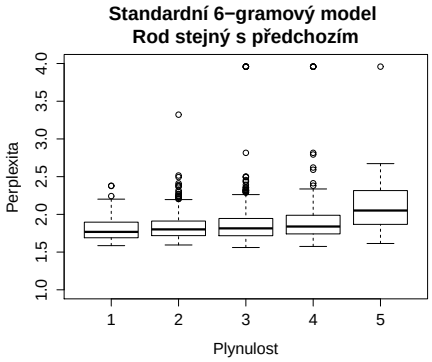
První z modelů s jednou morfologickou značkou budou modely obsahující rod. Slova budou nahrazena znakem `w`, ke kterému se připojí patřičný rod, lze-li u slova určit.

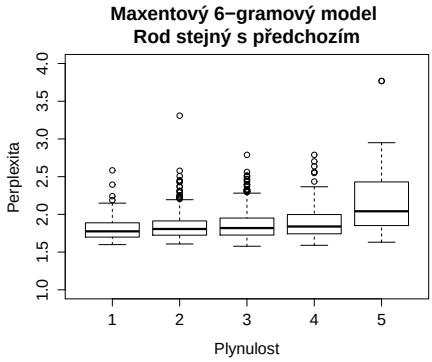
Příklad věty:

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

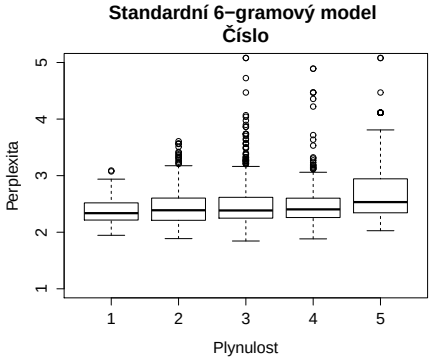


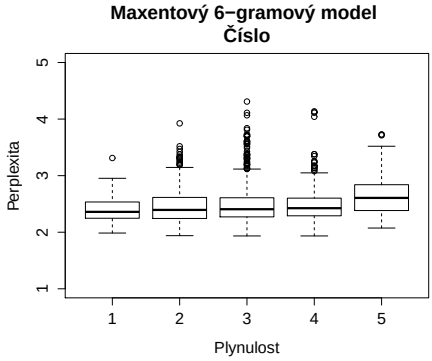
4.3.2.1 Rod stejný s předchozím



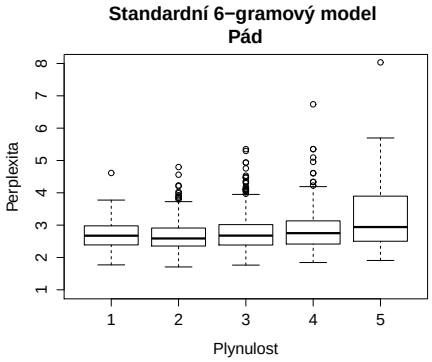


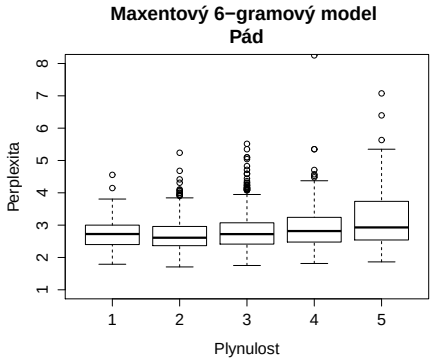
4.3.3 Číslo



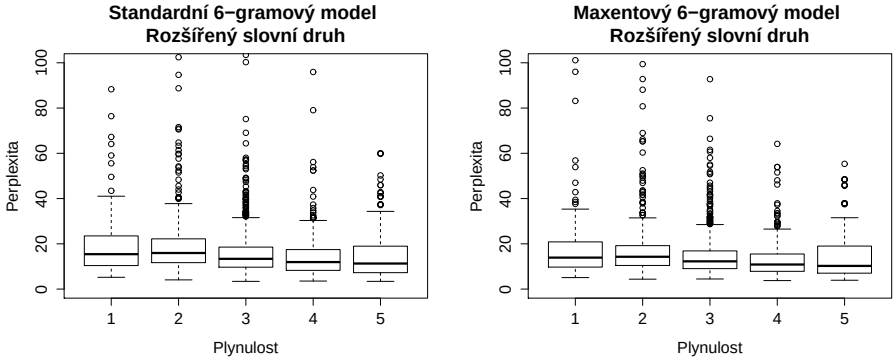


4.3.4 Pád

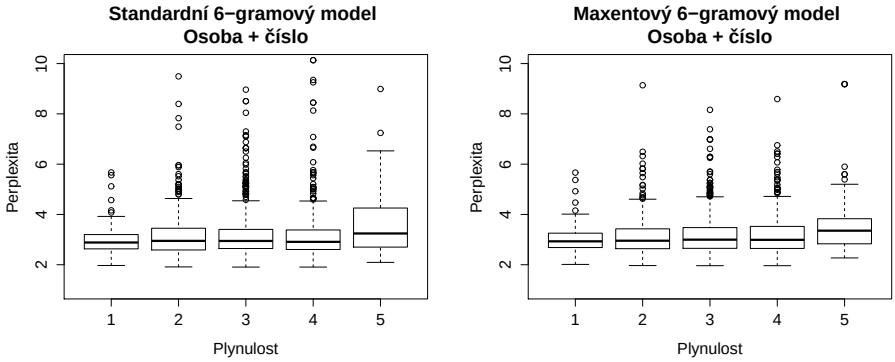




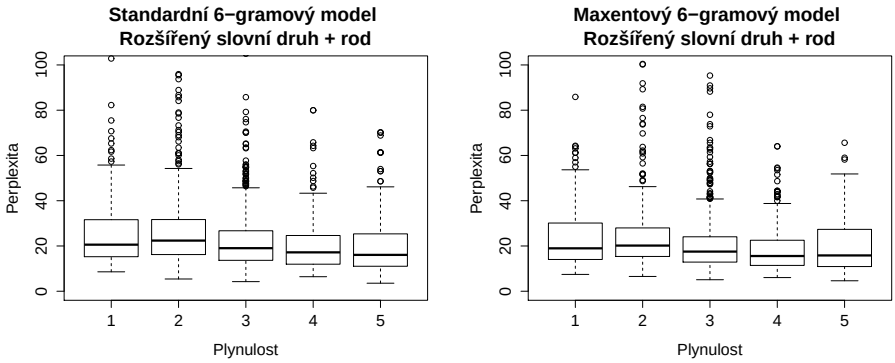
4.3.5 Rozšířený slovní druh



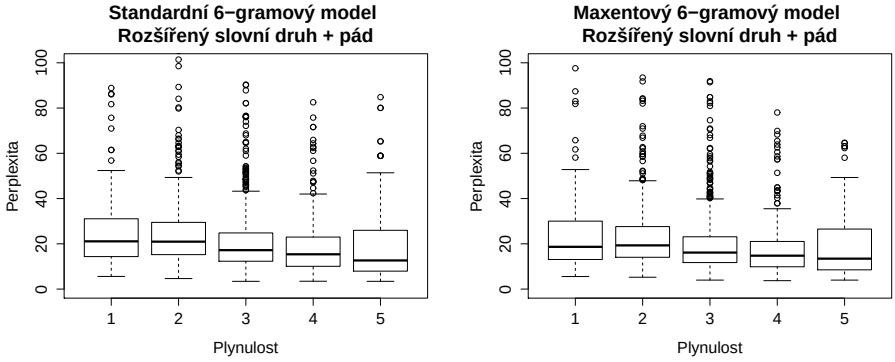
4.3.6 Osoba + číslo



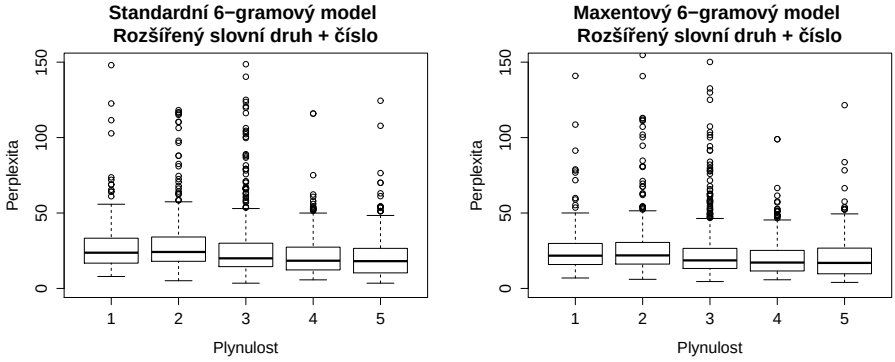
4.3.7 Rozšířený slovní druh + rod



4.3.8 Rozšířený slovní druh + pád



4.3.9 Rozšířený slovní druh + číslo



4.4 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.

Trénovat už nebudeme ve SRILMu, neboť ten nepodporuje jiné než n-gramové rysy. Pro tyto modely použijeme maxent toolkit od LeZhanga[3].

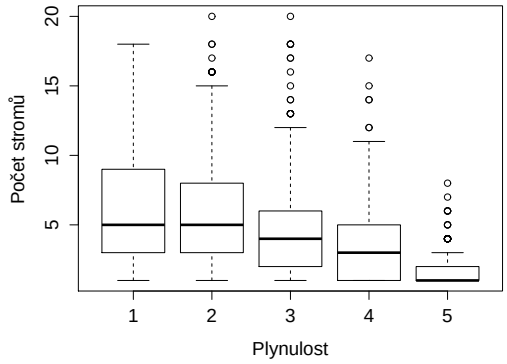
Pro následující experimenty jsme překladové hypotézy rozdělili na dva díly. Polovina tj. 1045 překladových hypotéz se použilo jako vývojová sada a druhá polovina jako sada testovací. Hypotézy byly rozděleny s ohledem na hodnocení plynulosti tak, aby vývojová i testovací množina vět obsahovala stejný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

[3]Lezhang

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**

4.4.1 Počet kořenů v naparsovaném stromu

Z gramaticky správné věty by měl jít vykreslit strom s větným rozborem. Předložíme-li ale ParZu větu, která správně není, vrátí nám stromů více. Zkusili jsme proto zjistit, jak spolu souvisí počet stromů z větného rozboru a ručně hodnocená plynulost překladu.



Výsledky vypadají slibně, neboť boxploty mají výrazně klesavou tendenci. Je tedy patrné, že počet stromů z větného rozboru souvisí s ručně hodnocenou plynulostí.

4.4.2 Chybějící infinitiv s zu

4.4.3 Chybějící podmět

4.4.4 Chybějící určité sloveso

4.4.5 Infinitiv po modálním slovese není na konci věty

4.4.6 Podmět se neshoduje s přísudkem

4.4.7 Příčestí minulé bez pomocného slovesa

4.4.8 Příčestí minulé není na konci věty

4.4.9 Jména a slovesa ve více osobách

4.4.10 Více určitých sloves ve větě

4.4.11 Určité sloveso není ve vedlejší větě na konci

4.4.12 Součtové rysy

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

Seznam použité literatury

Seznam tabulek

Seznam použitých zkratek

Přílohy