Počítačové zpracování češtiny

Značkování (tagging)

Daniel Zeman

http://ufal.mff.cuni.cz/course/popj1/

Analýza češtiny on-line (+značkování)

http://ufal-point.mff.cuni.cz/services/morph/

http://quest.ms.mff.cuni.cz/cgi-bin/zeman/morfo/index.pl

http://nlp.fi.muni.cz/projekty/ajka/ajkacz.htm

http://lindat.mff.cuni.cz/services/morphodita/

Homonymie: víceznačné výsledky morfologické analýzy

- je = on + PPNS4-3, PPXP4-3 | být + VB-S-3P-AA
- stát = stát-1 (země) | stát-2 (na ulici) | stát-3 (milión) | stát4 (se)
- stane = stát-4 + VB-S-3P-AA | stanout + VB-S-3P-AA | stan + NNIS5-A
- hnát = hnát-1 (noha) | hnát-2 (honit)
- žena = žena + NNFS1-A | hnát-2 + VeYS-A
- kupuje = kupovat + VB-S-3P-AA | kupovat + VeYS-A
- růže = růže + NNFS1-A | NNFS2-A | NNFS5-A | NNFP1-A | NNFP4-A | NNFP5-A

Extrémní mnohoznačnost

- vlastní =
  - vlastní + AAMS1-1A | AAMS5-1A | AAMP1-1A | AAMP4-1A | AAMP5-1A | AAIS1-1A | AAIS4-1A | AAIS5-1A | AAIP1-1A | AAIP4-1A | AAIP5-1A | AAFS1-1A | AAFS2-1A | AAFS3-1A | AAFS4-1A | AAFS5-1A | AAFS6-1A | AAFS7-1A | AAFP1-1A | AAFP4-1A | AAFP5-1A | AANS1-1A | AANS4-1A | AANS5-1A | AANP1-1A | AANP4-1A | AANP5-1A
  - vlastnit + VB-S-3P-AA | VB-P-3P-AA

- Ani jiné jazyky nejsou jednoznačné.
  - books = book (kniha) + NNS | book (rezervovat) + VBZ
  - meeting = meeting (schůze) + NN | meet (sejít se) + VB

Klasifikační problémy

- (Zde se omezujeme na morfologickou rovinu.)
- Jeden tvar téhož hesla, více morfologických kategorií (čísel, pádů) — více značek.
  - Řešení: **značkování** (tagging).
- Více hesel, u každého jedna značka.
  - Řešení: **lemmatizace** (lemmatization).
- Více hesel, u některých více značek.
  - Řešení: kombinace značkování a lemmatizace.
- Stejné heslo a značka, ale různé významy.
  - Řešení: rozlišení významu slov (word sense disambiguation).

Podobné problémy, podobná řešení

- Všechny problémy mají podobné rysy:
  - Nejednoznačným prvkem je slovo.
  - Jsou možné různé interpretace tohoto slova.
  - Je třeba určit, která je ta správná.
  - K tomu je třeba znát kontext. Bez něj jsou všechny interpretace správné.
- I metody jsou tedy podobné.
- Nadále se budeme zabývat hlavně vlastním značkováním.

Primitivní řešení: vyhrává nejpravděpodobnější

- Přechodníky jsou v současném jazyku málo časté (málo pravděpodobné) ⇒ pro „žena“ vyhraje značka NNFS1-A.
- 5. pád neživotného podstatného jména je nepravděpodobný ⇒ pro „stane“ vyhraje VB-S-3P-AA.

☹ Bohužel, velmi často je toto „řešení“ chybné.

  – Víceznačné koncovky různých pádů, např. zaměnitelnost 1. a 4. pádu mužských neživotných jmen (⇒ rozdíl mezi podmětem a předmětem, to je významná chyba).

Jak získat pravděpodobnosti?

- Dostatečně velká textová data — **korpus**.
- Někdo musí korpus projít a ručně rozhodnout všechny nejednoznačnosti — provést **anotaci**, anotovat korpus.
- Program spočítá výskyty jednotlivých značek.
- Je-li korpus dostatečně velký, relativní četnost značky odpovídá pravděpodobnosti jejího výskytu.

$$p(t) = \frac{c(t)}{\sum_i c(t_i)}$$

Nejpravděpodobnější značky

- Gender Study:

Jaký rod u podstatných jmen je v češtině nejběžnější?

Nejpravděpodobnější značky

Z:	.	0,16368
Db	*jak*	0,03514
J^	*ale*	0,03471
RR-6	*o*	0,03271
VB-S-3P-AA	*dělá*	0,02865
C=	*100*	0,02263
NNFS1-A	*žena*	0,02043
NNFS2-A	*ženy*	0,02039
RR-4	*pro*	0,02038
RR-2	*bez*	0,01922

1 výskyt: $p = 3.10^{-6}$ 133 značek

0 výskytů z 327597: 2112 značek z 3190 (66%)

Lze se vyhnout ruční práci anotátora?

- Lidská práce je drahá, anotace korpusu navíc úmorná.
- Proto kde to jde, je snaha využívat i tzv. *unsupervised methods* — učení bez lidského učitele.
- Pravděpodobnosti značek:
  - Brát v úvahu jenom jednoznačné případy.
  - Nebo: u nejednoznačných pravděpodobnost rozdělit rovným dílem. Výsledek bude ovšem stejný.
  - Pravděpodobnosti pak nejsou porovnatelné všechny navzájem, pouze v jednotlivých třídách nejednoznačnosti.

Jak testovat úspěšnost?

- Opět potřebujeme data, kde už jsou značky přiřazeny ručně.
- Projít testovací data a pro každé slovo nechat model přiřadit značku. Pak ji porovnat se značkou přiřazenou ručně.
- Úspěšnost: počet správně přiřazených značek / počet slov v testovacích datech.
- Testovací data nesmí být použita pro trénování! (Typicky: 10% dat pro testování, zbytek pro trénování.)
- Pro češtinu existují modely s úspěšností 95 %.

Zohlednění lexikální informace

- Pravděpodobnost značky (i relativní, vzhledem k pravděpodobnostem ostatních možností) se může lišit v závislosti na slovu.
- Příklad: slovo *hlavně* je buď příslovce (od přídavného jména *hlavní*, Dg-1A), nebo podstatné jméno *hlaveň* (NNFS2-A | NNFP1-A | NNFP4-A | NNFP5-A).
- Nelexikalizované pravděpodobnosti (výňatek z uvedené tabulky):
  - NNFS2-A 0,02039
  - Dg-1A 0,01343

Zohlednění lexikální informace

- Lexikalizované:
  - slovo *hlavně* se vyskytlo 45×, ani jednou jako *hlaveň*

$\Rightarrow p(\mathrm{Dg\text{-}1A} \mid hlavně) = 1$

$$p(t \mid w) = \frac{c(t \wedge w)}{\sum_i c(t_i \wedge w)}$$

Zohlednění lexikální informace

☹ Více parametrů:

- Dosud max. 3190 řádků tabulky (počet značek).
- Tvarů 20 000 000 (× 3190 značek).
- V praxi 1078 značek, 58347 tvarů.
- Ne všechny tvary umožňují všechny značky.
- Při průměru 27 značek / tvar (silně nadsazeno!) je 1 350 000 parametrů.

Zlepšení: kontext

- Řadu nejednoznačností lze rozhodnout (zjednoznačnit) na základě kontextu, ve kterém se slovo objevilo.
- Nezbytný rozsah kontextu je různý nejen pro jednotlivé uvedené problémy, ale i pro jednotlivá slova v rámci vybraného problému.
- Jednoduché modely
  - předcházející slovo
  - dvě předcházející slova
- Složitější modely — takřka libovolný dotaz
  - poslední sloveso bylo sloveso pohybu a před ním bylo zájmeno…

Příklady rozhodujícího kontextu

- *ženu krávu na trh*
  - Za slovem *ženu* následuje předmět (podstatné jméno, přídavné jméno, zájmeno nebo číslovka ve 4. pádě) nebo příslovečné určení místa (příslovce místa, předložka *na*, *do*, *k*, *přes*, …).
- *viděl starou ženu*; *ženu ani květinou neuhodíš*
  - V této klauzi (heuristika: po poslední čárce) se už vyskytlo sloveso.
  - Bezprostředně předchází přídavné jméno, které připouští stejný pád (akuzativ).
  - V okolí (levém ani pravém) nelze nalézt nic, co by mohlo být předmětem slovesa *hnát*. Naopak následuje jiné sloveso, vyžadující předmět.

Pravděpodobnosti s kontextem

- Při výběru značky se díváme na slovo, kterému značku přiřazujeme, a na slovo, které mu ve větě bezprostředně předchází.
- Na začátku věty jednoznačné pomocné slovo #.

$$p(t|w_{i-1}w_i) = \frac{c(t \wedge w_{i-1}w_i)}{\sum_j c(t_j \wedge w_{i-1}w_i)}$$

Počet parametrů modelu

- Četnost každé trojice tvar-tvar-značka.
- Při počtu tvarů 58347 a značek 1078 (nalezeno v našem korpusu) potenciálně $3{,}6\times10^{12}$ parametrů.
- Ale různých dvojic slov nemůže být více než slov v korpusu (v našem případě 327597)!
- Čili také: celý zbytek (stále $3{,}6\times10^{12}$) možností má p=0 (je považován za nemožný)! Totéž pro tvary, které jsme vůbec neviděli (2×10^{7}–58347 = 2×10^{7}). Ve skutečnosti je pro každý z nich jedna značka správně!
- $\Rightarrow$ vyhlazování (později); jiný kontext

Přesnost a úplnost modelu

- Obecnější kontext méně vystihuje pravidla pro konkrétní značku. Snadněji se v něm vyskytne i jiná značka.
  - menší **přesnost** (precision)
- Přesnější kontext snadněji spotřebuje dostupnou paměť.
- Kromě toho snadněji pomine případy, které jsou správně, ale z jeho zorného úhlu nebyly vidět (díval se „moc zblízka“).
  - menší **úplnost**, (recall)

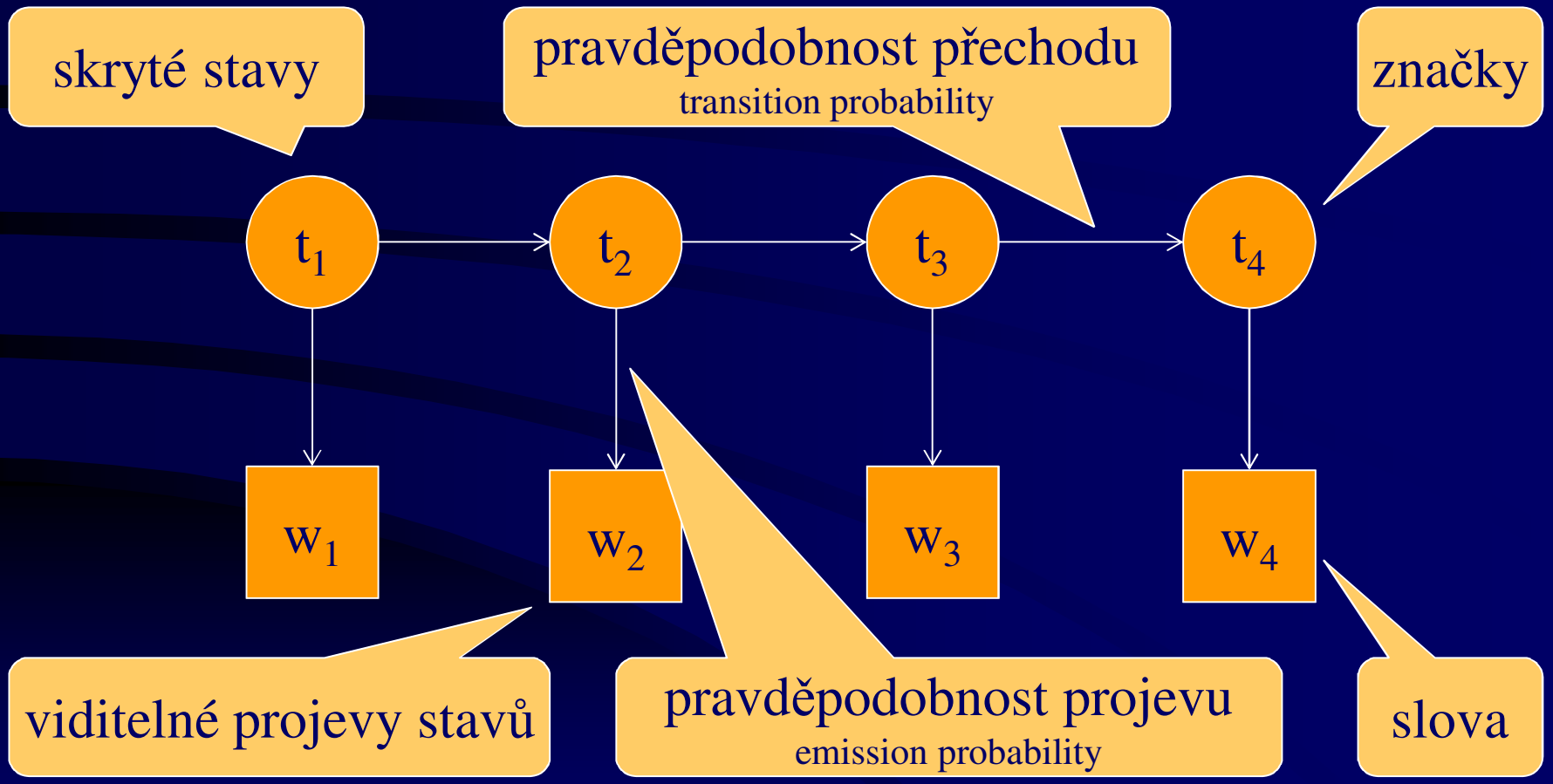
Jiný kontext: místo slova značka

- Při výběru značky se díváme na slovo, kterému značku přiřazujeme, a na značku, kterou jsme přiřadili předcházejícímu slovu.
- Na začátku věty jednoznačné pomocné slovo # se značkou Z#.

$$p(t_i | t_{i-1} w_i) = \frac{c(t_i \wedge t_{i-1} w_i)}{\sum_j c(t_j \wedge t_{i-1} w_i)}$$

Hidden Markov Model (HMM)

Skrytý Markovův model



Problém: co když jsme minule zvolili špatně?

- Výběr značky řídíme značkou předcházejícího slova.
- I ta mohla být víceznačná a museli jsme ji určit. Pokud jsme chybovali, je teď chyba ještě pravděpodobnější.

- Zotavení z chyb: bere se v úvahu i kontext vpravo.
- Pokud jsme minule vybrali značku A a pravděpodobnosti možných značek jsou teď výrazně nižší, než kdybychom zvolili značku B, vrátit se a zkusit B.
- Přesněji: snažíme se maximalizovat pravděpodobnost celé věty.

Pravděpodobnost věty

- Věta je posloupnost slov.
- Pravděpodobnost věty *S*: pravděpodobnost, že se v jedné větě vyskytnou daná slova v daném pořadí. Formálně: každé slovo *w* má svou pravděpodobnost výskytu (v libovolné větě) *p*(*w*).

$$p(S) = p(w_1 w_2 \ldots w_n)$$
$$= p(w_1) \cdot p(w_2|w_1) \cdots p(w_n|w_1 w_2 \ldots w_{n-1})$$

Pravděpodobnost posloupnosti značek

- Kontext je předcházející značka, ne slovo.
- Místo posloupnosti slov nás zajímá posloupnost značek.
- Pravděpodobnost věty *S*: pravděpodobnost, že se v jedné větě vyskytnou dané značky v daném pořadí. Každá značka *t* má svou pravděpodobnost výskytu (v libovolné větě) $p(t)$.

$$p(S) = p(t_1 t_2 \ldots t_n)$$
$$= p(t_1) \cdot p(t_2|t_1) \cdots p(t_n|t_1 t_2 \ldots t_{n-1})$$

Zjednodušení: výskyt značky je částečně nezávislý jev

- Neumíme zjistit $p(t_i|t_1 \ldots t_{i\text{-}1})$. Model by měl příliš mnoho parametrů.
- Proto předpokládáme (chybně), že $p(t_i|t_1 \ldots t_{i\text{-}1}) = p(t_i|t_{i\text{-}1})$.
- Potom maximalizujeme součin

$$p(S) = p(t_1) \cdot p(t_2|t_1) \cdots p(t_i|t_{i-1})$$

N-gramové modelování

- Používáme kontext předcházejícího slova ke klasifikaci aktuálního slova.
- Takové metodě se říká **bigram** (dvougramové modelování). Dvougram: parametry modelu jsou četnosti *dvojic* aktuální slovo – předcházející slovo.
- Obdobně **trigram** využívá dvě předchozí slova.
- Pro N>3 už nepraktické: příliš mnoho parametrů, příliš řídká data (mnoho správných trojic nebylo nikdy vidět).
- Sedmice slov údajně jednoznačně rekonstruují trénovací (anglický) text.
- Samotné slovo bez kontextu: **unigram.**
- Za 0-gram můžeme považovat rovnoměrné rozdělení.

Využití n-gramového modelování

- Základní a nejjednodušší metoda statistického zpracování jazyka.
- Kromě značkování lze použít i jinde:
  - Lemmatizace. Pravděpodobnost lemmatu na základě tvaru a předcházejících lemmat. (Popř. i jiných věcí podle možností.)
  - Rozlišování významu slov.
  - Jazykové modelování, např. jako složka rozpoznávání řeči (spolu s akustickým modelem). Pravděpodobnost tvaru na základě předcházejících tvarů (či podle možností značek a lemmat).
  - Modelování syntaxe: nikoli dvojice po sobě jdoucích slov, ale dvojic řídící větný člen – závislý větný člen.

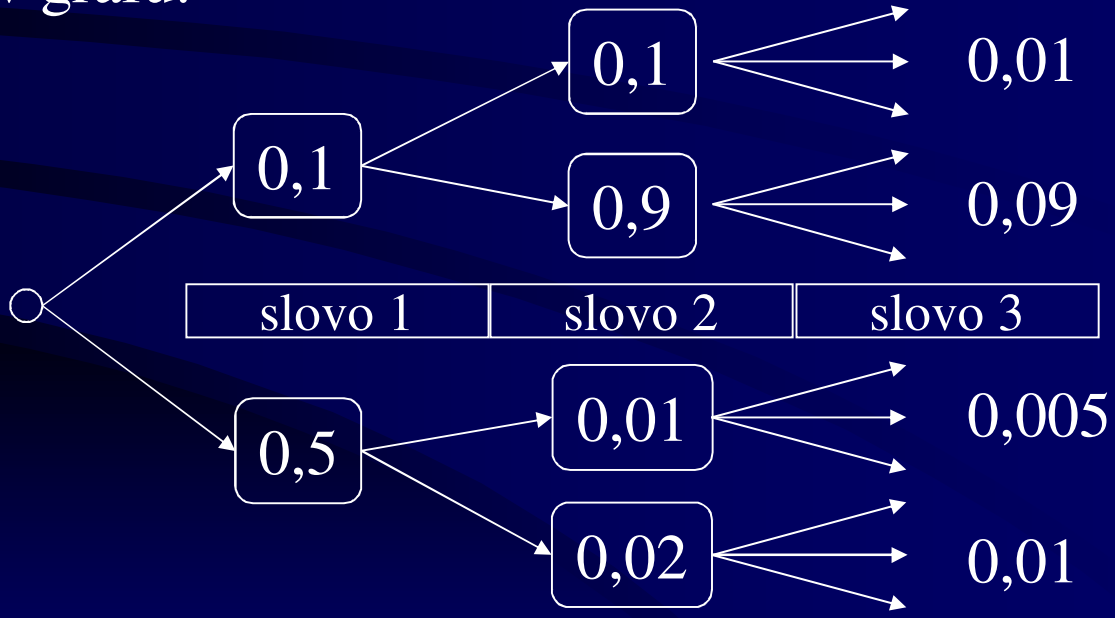
Hledání nejlepší posloupnosti značek

- Hladově.
- Vyzkoušet všechny možnosti (backtracking).
- Trellis (Viterbi).

- Hladově: pro každé slovo zvolit značku, která na základě předcházející značky má nevyšší pravděpodobnost.
  - 😐 Tak jsme to dělali dosud.
  - ☹ Nenajde maximum. Jakmile se vydáme po špatné cestě, už na ní zůstaneme.
  - ☺ Je to snadné a rychlé.

Hledání cesty projitím všech možností

- Hledání posloupnosti s nejvyšším součinem pravděpodobností je vlastně hledání nejlépe ohodnocené cesty v grafu:



Hledání cesty projitím všech možností

- ☹ Počet možných cest roste exponenciálně s počtem slov ve větě. Pro větu o 16 slovech (průměr) s průměrným počtem 2 značek na slovo je to 65536 cest. Pro větu o 32 slovech (nijak neobvyklé) už to budou 4 miliardy cest.
- ☹ Metoda je velmi náročná na čas.
- ☺ Ale najde opravdové maximum.
- ☺ Snižuje se chyba způsobená předpokladem, že značka slova závisí jen na jedné předcházející, ne na všech. Zvolená cesta do jisté míry reprezentuje i kontext ostatních předcházejících značek.

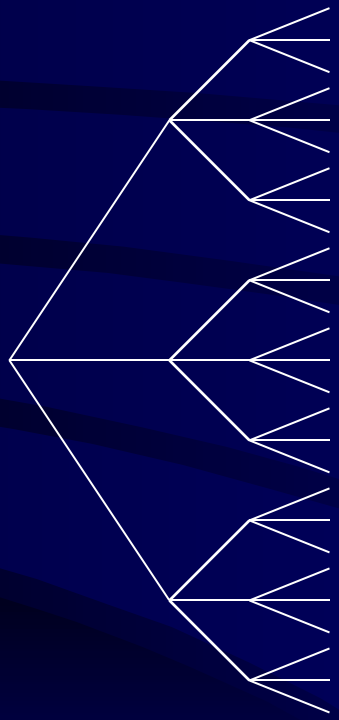
Kompromis: Viterbiho hledání (Trellis)

- Nepamatovat si všechny cesty, ale n (např. 5) nejlepších.
- Při zpracování nového slova každou z nich rozšířit nejvýše n způsoby (těmi nejlepšími možnými).
- Ze vzniklých $n \times n$ cest vybrat n nejlepších, ostatní zahodit.
- Na konci vybrat nejlepší přeživší cestu.
- Pro n=1 jde o hladový algoritmus.

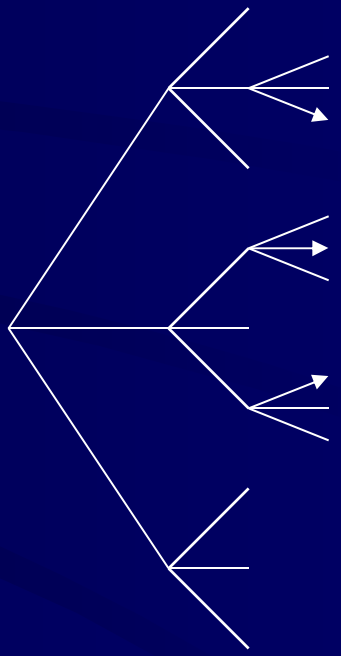
☺ Máme možnost opravit dřívější chybné rozhodnutí.
☺ Časovou náročnost lze regulovat velikostí n.

Viterbiho hledání

místo



třeba



Viterbi: pozor na nuly!

- Pozor na řídká data!
  - Má-li něco četnost 0, vynuluje se celý součin.
- Nuly dokážou Viterbiho úplně zničit.
- Řešení: **vyhlazování** (za chvíli).

Problém: nulové pravděpodobnosti

- Řada správných dvojic se vůbec nevyskytla v trénovacích datech $\Rightarrow$ má pravděpodobnost 0.
- Znamená to, že se nevyskytnou nikdy? Ne!
- Např. u značkování je rozdíl mezi značkou, kterou pro dané slovo nepřipouští ani morfologická analýza, a značkou, kterou jsme pouze v daném kontextu ještě neviděli.
- Proto je snaha se nulovým pravděpodobnostem vyhnout. Tomu se říká **vyhlazování**.

Problém: nulové pravděpodobnosti

- Příklad: rozpoznávání řeči.
  - Akustický model: p_a = p(slovo|zvuk).
  - Jazykový model: p_j = p(slovo|předcházející slovo).
  - p(slovo) = p_a . p_j
- Pokud p_j = 0, pak i p = 0 bez ohledu na p_a. Co když p_a = 1?

- Možné řešení: přičíst ke všem četnostem malou konstantu, např. 1. Pokud trénovací data obsahovala 10000 dvojic slov (tj. 10000 slov), pak vše, co se nevyskytlo, má pravděpodobnost 1/10001.

Problém: jak jemněji rozlišit neznámá slova?

- ☺ Přičtení konstanty je jednoduché a rychlé, neklade žádné nové nároky na zpracování dat.
- ☹ Nedělá ale rozdíl mezi tím, co jsme neviděli a tím, o čem předem *víme*, že je to nemožné.
  - Řešení: těmto opravdu nemožným dát p = 0. (Opatrně!)
- ☹ Neznámé dvojice slov jsou všechny hodnoceny stejně (rovnoměrné rozdělení pravděpodobností). Co když víme, že některé z nich jsou pravděpodobnější?

Problém: jak jemněji rozlišit neznámá slova?

- Neznámé dvojice slov jsou všechny hodnoceny stejně (rovnoměrné rozdělení pravděpodobností). Co když víme, že některé z nich jsou pravděpodobnější?
- Konkrétně: dvojice slov sice v trénovacích datech nebyla, ale slova tam byla samostatně.
- Příklad: věta *„To/PDNS1 je/VB-S-3P-AA hlavně/Dg-1A vaše/PSHS1-P2 věc/NNFS1-A ./Z:“*
  - Nevyskytlo se ani *(VB-S-3P-AA hlavně/Dg-1A)*, ani *(VB-S-3P-AA hlavně/NNFP1-A)*.
  - Avšak 45× se vyskytlo (v jiném kontextu) *hlavně/Dg-1A*, druhá možnost ani jednou.

Řešení: lepší méně specifická informace než žádná

- Neznáme-li pravděpodobnost dvojice slov, spokojíme se s pravděpodobností aktuálního slova.
  - Poznámka: v anglické literatuře ***back off*** *to less specific info*.
  - Kombinace n-gramového modelu s (n–1)-gramovým.
  - Pokud ani 1-gramový model (**unigram**) nepomůže, pak teprve nasadíme rovnoměrné rozdělení pravděpodobností (0-gram).

- Potíž: kombinujeme několik pravděpodobnostních rozdělení, v každém z nich je součet p roven 1, tj. v kombinovaném modelu tomu tak není!

Vyhlazování

- Jednotlivé modely (rozdělení) zkombinovat do jednoho, v němž bude opět součet pravděpodobností = 1. Jak?
- Celková pravděpodobnost je součet pravděpodobností v jednotlivých modelech normalizovaných **váhovým koeficientem λ_i**.
- Váhy jsou čísla mezi 0 a 1, jejich součet je 1.

Vyhlazování

$$p(t_i|w_i,t_{i-1}) = \lambda_3 \frac{c(t_i,w_i,t_{i-1})}{\sum_{j=1}^{|T|} c(t_j,w_i,t_{i-1})} + \lambda_2 \frac{c(t_i,w_i)}{\sum_{j=1}^{|T|} c(t_j,w_i)} + \lambda_1 \frac{c(t_i)}{\sum_{j=1}^{|T|} c(t_j)} + \lambda_0 \frac{1}{|T|}$$

Jak zjistit vyhlazovací váhy?

- Odhadem, např. 0,9 + 0,09 + 0,009 + 0,001. Obvykle funguje přijatelně.
- Iteračním algoritmem.
  - Vyčlenit část dat, která nepoužijeme ani pro trénování ani pro test („vyčleněná data“, held-out data). Např. Trénink = 80%, Vyčl = 10%, Test = 10%.
  - Stanovit počáteční váhy a pro vzniklý model vyzkoušet, jak dobře odpovídá vyčleněným datům.
  - Na základě tohoto pozorování upravit váhy a opakovat.
  - Proces konverguje k optimálním vahám. (Jsou optimální pro tato vyčleněná data. Na testovacích datech to bude horší.)

Jak zjistit vyhlazovací váhy?

- *V* jsou vyčleněná data.

$$\Lambda_3 = \sum_{i=1}^{|V|} \lambda_3 \frac{c_V(t_i, w_i, t_{i-1})}{\sum_{j=1}^{|T|} c_V(t_j, w_i, t_{i-1})}$$

$$\Lambda_2 = \sum_{i=1}^{|V|} \lambda_2 \frac{c_V(t_i, w_i)}{\sum_{j=1}^{|T|} c_V(t_j, w_i)}$$

$$\Lambda_1 = \sum_{i=1}^{|V|} \lambda_1 \frac{c_V(t_i)}{\sum_{j=1}^{|T|} c_V(t_j)}$$

$$\Lambda_0 = \sum_{i=1}^{|V|} \lambda_0 \frac{1}{|T|}$$

Jak zjistit vyhlazovací váhy?

$$\lambda_i' = \frac{\Lambda_i}{\sum_j \Lambda_j}$$

- S upravenými vahami lze průchod vyčleněnými daty opakovat a získat ještě lepší váhy (EM algoritmus, estimation-maximization).
- Kdybychom použili trénovací data, vyšly by váhy 1-0-0-0.

Předpovídání každé kategorie zvlášť

- Předvedli jsme současně:
  - jednu metodu značkování (lze řešit i jinak)
  - jedno použití n-gramového modelování (lze použít i jinde)
- Jedna možnost jiného značkování:
  - Modelovat každou kategorii zvlášť.

Předpovídání každé kategorie zvlášť

- Dosud:
  - „Za přídavným jménem ženského rodu v jednotném čísle a třetím pádě následuje podstatné jméno ženského rodu v jednotném čísle a třetím pádě.“
- Nyní:
  - „Za třetím pádem následuje třetí pád.“
  - …
- Není dáno, že to tak musí být lepší. Ale lze to zkusit. (Hajič a Hladká, 1998)

Domácí úkol

- Sežeňte si tagger (např. Morphodita, TreeTagger, RF-Tagger, TnT, Stanford)
  - Google: download pos tagger
- Zvolte si jazyk
- Sežeňte označkovaný korpus
  - Např. https://lindat.mff.cuni.cz/repository/xmlui/
- Tagger natrénujte / nebo stáhněte hotový natrénovaný model
- Vyhodnoťte úspěšnost, prohlédněte si časté druhy chyb
- Napište mi zprávu mailem
  - (do 27. listopadu)