Interakce člověk–počítač v přirozeném jazyce (ICP)

LS 2013 — Jazykové modely

Tino Haderlein, Elmar Nöth

Katedra informatiky a výpočetní techniky (KIV)
Západočeská univerzita v Plzni

Lehrstuhl für Mustererkennung (LME)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Why Do We Need Syntactic structure?

Wouldn't the sentence
"I want to put a hyphen between the words Fish and And and And and Chips in my Fish–And–Chips sign"
have been clearer if quotation marks had been placed
before Fish, and between Fish and and, and and and And, and And and and, and and and And, and And and and, and and and Chips, as well as after Chips?

Yes, it makes sense.

Why Do We Need Syntactic structure?

John where James had had had had had had had had had had been correct

John, where James had had “had”, had had “had had”;
“had had” had been correct.

vs.

John, where James had had “had had”, had had “had”;
“had had” had been correct.

To Know What Was Said Without Listening

- utterance: “We must resolve some problems”
- we hear NOTHING!
- we count on 91243 sentences of the Wall Street Journal (WSJ) which words are at the beginning of a sentence, and how often
- the text contains 2265839 words; (47343 different)
- can we guess the correct utterance?
- the utterance is not contained in the data set

Which Words Start a Sentence? (7170)

Rank	Count	Word	Rank	Count	Word
1	14844	THE	11	881	THIS
2	3877	IN	12	849	I
3	3801	BUT	13	822	THEY
4	3307	MR.	14	820	AS
5	1776	HE	15	798	**WE**
6	1742	A	16	789	IF
7	1722	QUOTE	17	751	THAT
8	1631	IT	18	733	AT
9	1290	AND	19	622	SOME
10	1184	FOR	20	568	IT’S

Which Words Follow a “we” in WSJ? 416!

Rank	Count	Word	Rank	Count	Word
1	295	HAVE	11	50	WOULD
2	245	ARE	12	42	BELIEVE
3	119	DON’T	13	39	KNOW
4	108	WERE	14	39	DO
5	101	WILL	15	37	CAN’T
6	95	CAN	16	36	COULD
7	70	HAD	17	34	DIDN’T
8	66	WANT	18	33	EXPECT
9	55	THINK	19	31	SHOULD
10	52	NEED	20	29	**MUST**

Which Words Follow a “must” in WSJ? 247!

Rank	Count	Word
1	26	HAVE
2	15	PAY
3	15	MAKE
4	11	TAKE
5	11	ALSO
6	10	APPROVE
7	8	NOT
8	8	MEET
9	6	GO
10	6	FIND

Rank	Count	Word
11	6	COME
12	5	SHOW
13	5	DECIDE
14	5	CLEAR
15	4	STILL
16	4	REMAIN
17	4	PROVIDE
18	4	KNOW
19	4	GET
..	..	..
109	1	**RESOLVE**

Which Words Follow a “resolve” in WSJ? 36!

Rank	Count	Word	Rank	Count	Word
1	18	THE	11	2	ANY
2	6	A	12	1	TODAY
3	5	TO	13	1	THROUGH
4	3	THEIR	14	1	THESE
5	3	ITS	15	1	TECHNICALLY
6	3	DISPUTES	16	1	TECHNICAL
7	3	CASES	17	1	**SOME**
8	2	THIS	18	1	SOCIAL
9	2	OUR	19	1	SEVERAL
10	2	DIFFERENCES	20	1	REMAINING

Which Words Follow a “some” in WSJ? 1364!

Rank	Count	Word
1	667	OF
2	152	ANALYSTS
3	58	TIME
4	57	PEOPLE
5	56	CASES
6	49	INVESTORS
7	47	OTHER
8	36	POINT
9	34	ONE
10	31	TWO

Rank	Count	Word
11	31	COMPANIES
12	28	TRADERS
13	24	INDUSTRY
14	23	U.
15	23	OBSERVERS
16	23	ARE
17	21	KIND
18	21	IN
19	20	ECONOMISTS
20	20	BIG
..	..	..
112	5	**PROBLEMS**

N-gramové jazykové modely

- Pro vyhodnocení Bayesovy rovnice je nutná pravděpodobnost
$$P(\boldsymbol{w}) = P(w_1, ..., w_N)$$
= pravděpodobnost, že posloupnost slov $w_1, .., w_N$ byla řečena.
- Rozvij $P(\boldsymbol{W})$ podle
$$P(\boldsymbol{w}) = P(w_1, .., w_N) = P(w_1) \cdot P(w_2|w_1) \cdot \ldots \cdot P(w_N|w_1, \ldots, w_{N-1}).$$
- Pravděpodobnost slova w_i je tedy závislá na všech předem řečených slovech (= historie).
- Nelze odhadnout $P(w_i|w_1, \ldots, w_{i-1})$ pro velké hodnoty i.
- Řešení: Definuj ekvivalenční třídy pro předchozí slova.

N-gramové jazykové modely

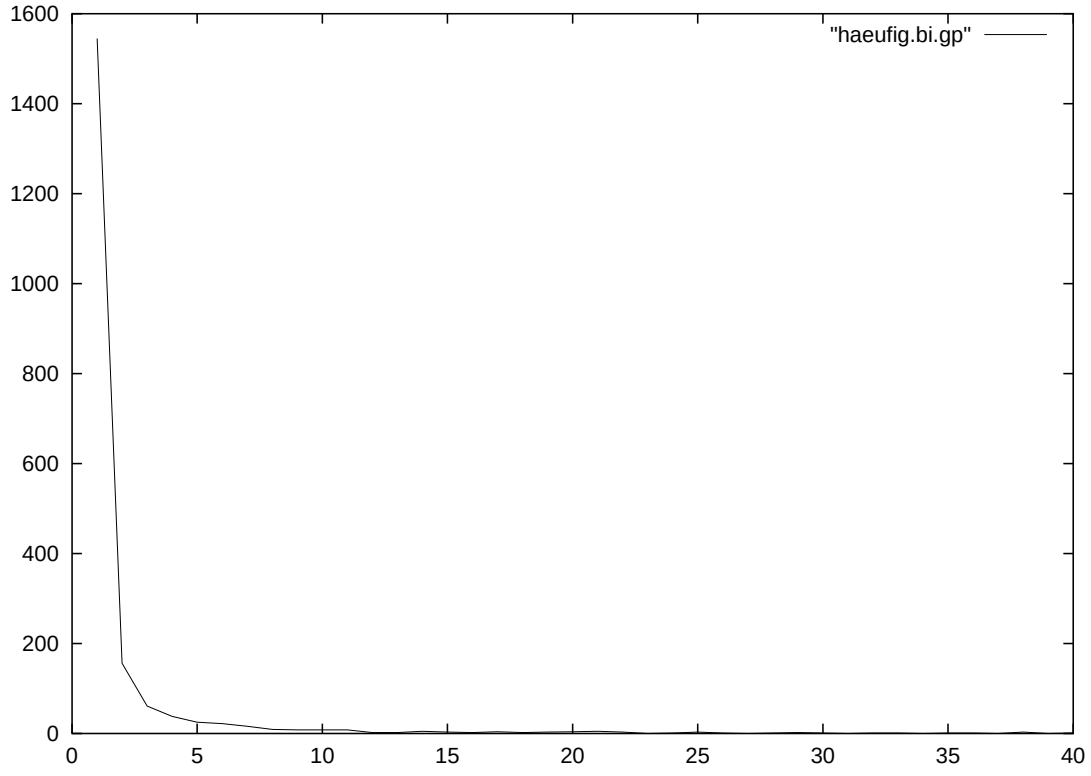
- Ekvivalenční třídy:
  w_i je závislé jen na zkrácené historii $\Phi(w_1, ..., w_{i-1})$:
  $P(\boldsymbol{w}) = P(w_1) \cdot \prod_{i=2}^{N} P(w_i|\Phi(w_1, ..., w_{i-1}))$
- Unigramový jazykový model: Všechny historie jsou ekvivalentní:
  $P(\boldsymbol{w}) = \prod_{i=1}^{N} P(w_i)$
- Bigramový jazykový model: Všechny historie, které skončí ve stejném slově, jsou ekvivalentní:
  $P(\boldsymbol{w}) = P(w_1) \cdot \prod_{i=2}^{N} P(w_i|w_{i-1})$
- Trigramový jazykový model: Všechny historie se stejnými dvěma posledními slovy jsou ekvivalentní:
  $P(\boldsymbol{w}) = P(w_1) \cdot P(w_2|w_1) \cdot \prod_{i=1}^{N} P(w_i|w_{i-2}, w_{i-1})$

Odhad ML n-gramových jazykových modelů

- počítej relativní četnosti $f(w_i|w_{i-2}, w_{i-1})$, $f(w_i|w_{i-1})$, $f(w_i)$ na tréninkovém korpu
- např. $f(w_i|w_{i-2}, w_{i-1}) = \frac{\#(w_{i-2}, w_{i-1}, w_i)}{\#(w_{i-2}, w_{i-1})}$
  $\#$ znamená hojnost výskytu v tréninkovém korpu
- Odhad ML pro $P(w_i|w_{i-n+1}, ..., w_{i-1}) = f(w_i|w_{i-n+1}, ..., w_{i-1})$
- Problém: V tréninkovém korpu se hodně z možných trigramů neobjevuje $\rightarrow P(\boldsymbol{w}) = 0 \Rightarrow$ rozpoznávací chyba.
- Příklad: Velikost slovní zásoby $= |\mathcal{V}| = 20000 \Rightarrow 8 \cdot 10^{12}$ možných trigramů
- V tréninkovém textu se 100 000 slovy se pozoruje maximálně 10^5 různých n-gramů, ve skutečnosti ale ještě mnohem méně.
- Řešení: hlazení odhadních hodnot nebo vytvoření kategorií

Četnost bigramů

histogram absolutních četností výskytu bigramů ve vzorku kolem 52000 slov (1780 slov ve slovní zásobě)



Hlazení n-gramů

- Nejjednodušší možnost jak zabránit případu $\#(w_{i-n+1}, ..., w_i) = 0$ je **Jeffreyovo hlazení**: Buď $\#'(w_{i-n+1}, ..., w_i) = 1 + \#(w_{i-n+1}, ..., w_i)$.

$$P(w_i|w_{i-n+1}, ..., w_{i-1}) = \frac{\#'(w_{i-n+1},...,w_i)}{\#'(w_{i-n+1},...,w_{i-1})} = \frac{1+\#(w_{i-n+1},...,w_i)}{|\mathcal{V}|+\sum_{w_i \in \mathcal{V}} \#(w_{i-n+1},...,w_i)}$$

- Problém: Nemožné n-gramy dostanou poměrně vysokou pravděpodobnost.
- Příklad nahoře: I když se při 1780 slovech (types) a 500 000 slovech postupného textu (tokens) každý bigram v tréninkovém textu objeví přesně jednou, stoupá „hmota“ pravděpodobnosti asi od 500 000 asi na 3 670 000, při 52 000 asi od 50 000 na 3 200 000.

Hlazení n-gramů: Good-Turingův odhad

- základní myšlenka: Všechny n-gramy se stejnou četností r v tréninkovém korpu by měli dostat stejnou pravděpodobnost.
- Good-Turingova odhadní hodnota dodá n-gramu, který se objeví r-krát v tréninkovém korpu, hodnotu r^*:

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

- r^* je očekávaná hodnota četnosti vyskytování tohoto n-gramu v jiném korpu se stejnou velikostí jako tréninkový korpus.
- n_l je počet n-gramů, které se v tréninkovém korpu vyskytují přesně l-krát.
- konverze/přeměna na pravděpodobnost: pravděpodobnost, že se n-gram vyskytuje k-krát v datech:

$$p_k = \frac{r^*}{N}, \quad \text{kde } N = \text{počet n-gramů v tréninkovém korpu}$$

Hlazení n-gramů: Good-Turingův odhad

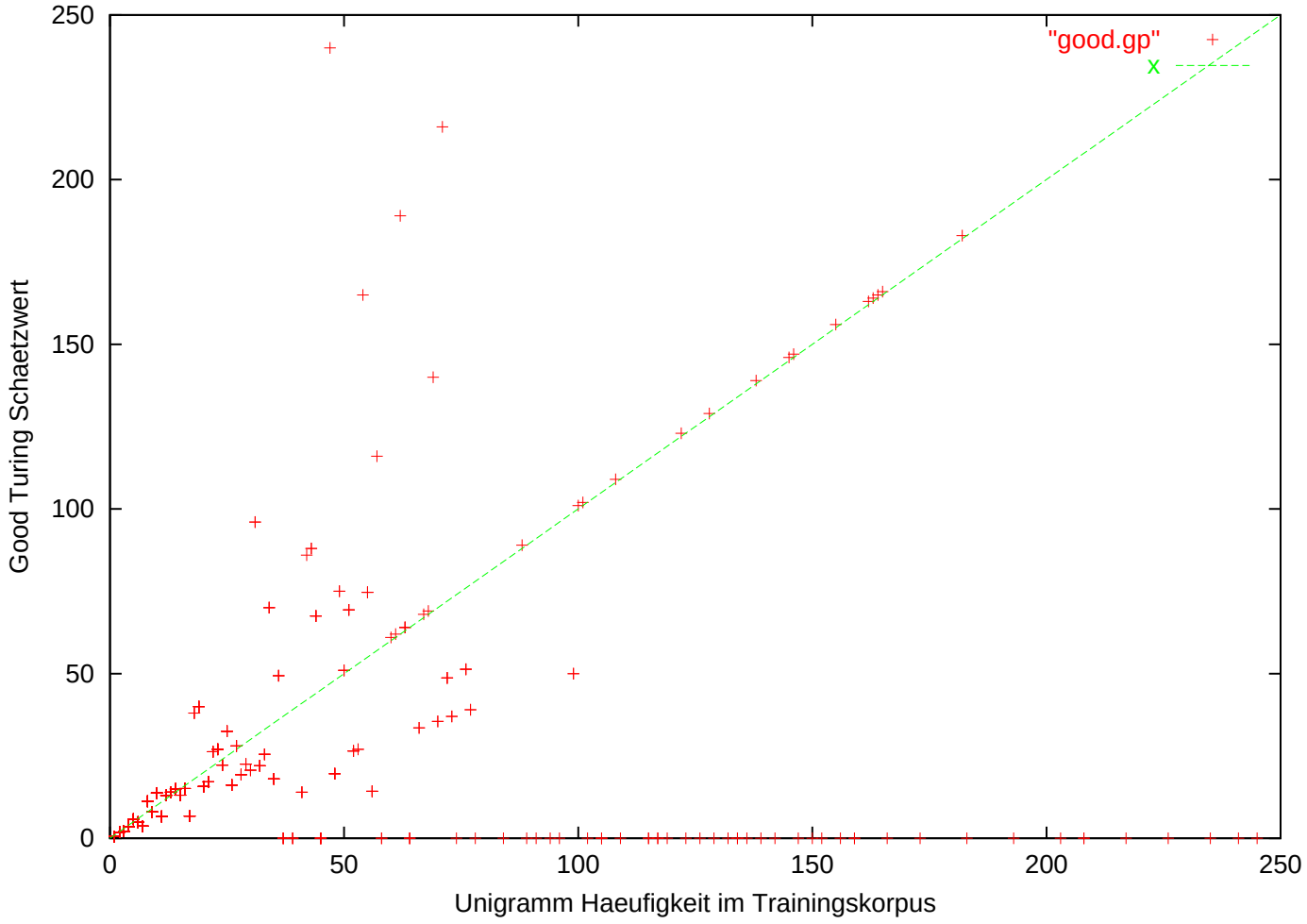
- Good-Turingův odhad není možný, když $n_r = 0$, tj. před aplikací se musí použít jiná metoda hlazení, aby bylo $n_r \neq 0$ pro všechny n-gramy.
- Důležité: I pro nové, modifikované hodnoty četností zůstane suma všech četností stejná:

$$(N)^* = \sum_{k=1}^{K} (N_k)^* = \sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1) n_{r+1} = \sum_{r=1}^{\infty} r n_r = N$$

- To znamená, že Good-Turingův odhad ty vypočítané četnosti nemění libovolně, ale jen je **přerozdělovává**.

Hlazení n-gramů: Good-Turingův odhad

osa x: četnost unigramů v tréninkovém korpu; osa y: Good-Turingova odhadní hodnota



Hlazení n-gramů: strategie

- Všeobecně: N-gramy, které se nevyskytnou v tréninkovém korpu, se zobrazují na n-gramy nižšího řádu (= se zkrácenou historií).
- Všeobecná strategie (= backoff smoothing):

$$P_{smooth}(w \mid \boldsymbol{v}) = \begin{cases} \hat{q}(w \mid \boldsymbol{v}), & \text{když } \#(\boldsymbol{v}w) > 0 \\ \beta(\boldsymbol{v}) \cdot P_{smooth}(w \mid \boldsymbol{v}'), & \text{když } \#(\boldsymbol{v}w) = 0 \end{cases}$$

- $\hat{q}(\cdot)$ je modifikovaná funkce odhadu jemnějšího modelu.
- **Znovurozdělování** ušetřené hmoty pravděpodobnosti je proporcionální k $P_{smooth}(w|\boldsymbol{v}')$.
- $\boldsymbol{v}'$ je „**zhublá**“ (zredukovaná) podmínková část hrubého modelu.
- Váha $\beta(\boldsymbol{v})$ garantuje, že pravděpodobnosti $P_{smooth}(w|\boldsymbol{v})$ dodržují pravidla stochastiky.

Hlazení n-gramů: Katz smoothing

- Kombinace základů Good-Turingova odhadu se strategií backoff
- Postup:
  - Nemění se relativní četnost velmi častých n-gramů s $r > k$ ($k = 5, ..., 8$).
  - Relativní četnost n-gramů s $k \geq r > 0$ se sníží ve prospěch n-gramů, které se neobjeví ($r = 0$; **discounting**).
  - Každý n-gram s $r = 0$ dostane hmotu pravděpodobnosti; dostane vyšší (popř. nižší) pravděpodobnost, když se příslušný $(n-1)$-gram vyskytuje v korpu velmi často (popř. málo).
- Katz smoothing:

$$P_{katz}(w \mid \boldsymbol{v}) = \begin{cases} \frac{\#(\boldsymbol{v},w)}{\#(\boldsymbol{v})}, & \text{když } r > k \\ d_r \cdot \frac{\#(\boldsymbol{v},w)}{\#(\boldsymbol{v})}, & \text{když } k \geq r > 0 \\ \alpha(\boldsymbol{v}) \cdot P(w \mid \boldsymbol{v}'), & \text{když } r = 0 \end{cases}$$

Hlazení n-gramů: Katz smoothing

- Přesné rovnice získáme z následujících podmínek:
  - Musí zůstat splněna pravidla stochastiky.
  - Faktor d_r má být proporcionální ke Good-Turingovu odhadu.
  - Hmota pravděpodobnosti, kterou odebíráme od častých n-gramů, má být stejná jako ta, kterou přidělíme nevyskytujícím n-gramům.
- Je jediné řešení:

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \; ; \quad \alpha(\boldsymbol{v}) = \frac{1 - \sum_{w|\#(\boldsymbol{v},w)>0} P_{katz}(w|\boldsymbol{v})}{1 - \sum_{w|\#(\boldsymbol{v},w)>0} P(w|\boldsymbol{v}')}$$

Hlazení n-gramů: Interpolace

- Motivace: Pro malé r je odhad ML velmi nepřesný, i když $r > 0$.
- Řešení: zahrnovat statistiky nižšího řádu, i když $r > 0$
- Lineární interpolace:

$$P_I(w_n|w_1,..,w_{n-1}) =$$
$$\rho_o \cdot \frac{1}{|\mathcal{V}|} + \rho_1 \cdot f(w_n) + \rho_2 \cdot f(w_n|w_{n-1}) + \cdots + \rho_n \cdot f(w_n|w_1..w_{n-1})$$
$$\text{kde } \sum_i \rho_i = 1$$

- Váhy ρ_i můžeme určit např. algoritmem EM na separátním validačním (ohodnocovacím) vzorku.

Další důležité metody interpolace

- Kneser-Neyova interpolace: nelineární interpolace
- interpolace s principem maximální entropie
- log-lineární interpolace
- racionální interpolace (jen v Erlangenu!)

Tvoření kategorií

- další metoda zabránění nespolehlivým odhadním hodnotám
- $L = 1000$ slov $\implies$ 1 000 000 000 trigramů!

$\rightarrow$ sniž efektivní velikost slovní zásoby

- Skupiny slov s podobnými funktionálními a statistickými vlastnostmi se sdružují do **kategorií**

$$\mathcal{C} = \{C_1, \ldots, C_N\}, \quad \text{kde} \quad \bigcup_{k=1}^{N} C_k = \mathcal{V}.$$

- Příklad: Všechny číslovky, jména měst, osob, měsíců si všechny pro sebe tvoří jednu kategorii.
- Kategorie mají být po párech disjunktní $\rightarrow$ tvoření kategorií je jednoznačné.
  Problém: Je např. „Teplá“ jméno města nebo ne?

$$w \in \mathcal{V} \longrightarrow C(w) \in \mathcal{C} \quad \text{s vlastností } w \in C(w)$$

Tvoření kategorií

- příklad: tvoření kategoriálního bigramového jazykového modelu:

$$P(\boldsymbol{w}) = P(w_1) \cdot P(w_1 \mid C(w_1)) \cdot \prod_{i=2}^{m} P(w_i \mid C(w_i)) \cdot P(C(w_i) \mid C(w_{i-1}))$$

- nutné statistiky:
  - při bigramech $N^2 - 1$ pravděpodobností $P(C_i)$, $P(C_j \mid C_i)$ pro přechody mezi kategoriemi
  - $|\mathcal{V}| - N$ příslušností $P(W_k \mid C(W_k))$ ke kategoriím
- „lepení“ při n-gramových gramatikách analogicky
- komplexnější: překrývající se kategorie $\rightarrow$ příslušnost ke kategorii je nepozorovatelný stav $\rightarrow$ HMM
- Na kategorie se dají přenášet všechny interpolační a backoff strategie.

Parts of Speech (POS)

- syntakticky/sémanticky/pragmaticky orientované slovní třídy (50–100 POS)
- druh slova, pád, číslo, rod, čas, způsob/vid, ...
- efektivně v jazycích s komplexním skloňováním, jako čeština, němčina, francouzština, italština

Cache Models

- metoda dynamické **adaptace** jazykového modelu na téma, o kterém se mluví
- statický jazykový model P_{static}: lineárně interpolován cache modelem P_{cache}

$$P_{total}(w_i|w_{i-n+1},..,w_{i-1}) = \rho_c P_{static}(w_i|w_{i-n+1},..,w_{i-1}) + (1-\rho_c)P_{cache}(w_i|w_{i-1})$$

- odhad cache modelu P_{cache} na slovech, která byla řečena jako **poslední** $\rightarrow$
- proto je typicky jen bi- nebo trigram, protože je pro cache model k dispozici jen málo dat
- interpolační váha ρ_c se mění s velikostí cache
- především zlepšení v diktovacích systémech

Perplexita

- perplexita: míra pravděpodobnosti, kterou jazykový model přiděluje větám z testovacího vzorku (testovací množiny)
- dáno: jazykový model, který větám z testovacího vzorku $\boldsymbol{W}$ přiděluje pravděpodobnost $P(\boldsymbol{W})$
- aproximace entropie $H(\boldsymbol{W})$ modelu $P(w_i|w_{i-n+1},..,w_{i-1})$ na datech $\boldsymbol{W}$:

$$H(\boldsymbol{W}) = -\frac{1}{N_W}\log_2 P(\boldsymbol{W}); \quad N_W \text{ je počet slov ve } \boldsymbol{W}$$

- (empirická) test-set perplexity $PP(\boldsymbol{W})$ definovaná jako

$$PP(\boldsymbol{W}) = 2^{H(\boldsymbol{W})}$$

- Můžeme ji počítat s $PP(\boldsymbol{W}) = P(\boldsymbol{W})^{-\frac{1}{N_W}}$.
- Čím menší je empiricky aproximovaná perplexita jazykového modelu, tím lépe umí model předpovídat testovací vzorek a tím lépe je vhodný.

Perplexita

- Skutečná perplexita PP je vždycky menší než empirická aproximace $PP(\boldsymbol{W})$: $PP(\boldsymbol{W}) \geq PP$ (kvůli Jensenově nerovnici)
- Skutečná perplexita $PP = 2^H$ s

$$H = \lim_{m\to\infty} H_m = -\lim_{m\to\infty} \frac{1}{m} \sum_{\boldsymbol{w}\in\mathcal{V}^m} P(\boldsymbol{w}) \cdot \log_2 P(\boldsymbol{w})$$
$$= -\lim_{m\to\infty} \frac{1}{m} E[\log_2 P(\boldsymbol{w})] = -\lim_{m\to\infty} \frac{1}{m} \log_2 P(\boldsymbol{w})$$

- Náhodný proces má maximální entropii $H = \log_2 L$, když $P(v|\boldsymbol{w}) = \frac{1}{L}$, kde $L = |\mathcal{V}|$.
- Perplexita může proto být interpretována jako střední stupeň rozvětvení jazyka.
- Čím lépe se slova jednoho jazyka dají předpovídat, tím menší je jeho perplexita.

Perplexita

- Skutečná perplexita anglického jazyka je domněle kolem 30-50.
- Empirická perplexita jazykového modelu je pro trigramy typicky mnohem menší než pro bigramy a unigramy.
- Empirická perplexita jazykového modelu je silně závislá na datech:
  - Na textech z novin (Wall Street Journal) se slovní zásobou 5000 slov dosahují jazykové modely hodnoty perplexity kolem 128 (trigram) popř. 176 (bigram).
  - Při hlasovým dialogovým systémech (např. ATIS – Air Travel Information System, EVAR – informace o vlacích) je perplexita trigramových modelů pod 20.

Perplexita

- Pokud se počítá perplexita nezávisle na akustické možnosti záměny slov, může i při nízké perplexitě výkon rozpoznávače řeči být velmi špatný.
- Jsou ale metody, které např. zkouší vybrat kategorie tak, že se akusticky snadně záměnitelná slova dostanou do různých kategorií a tím je jazykový model umí rozlišovat.