Языковые модели

Павел Браславский

использованы слайды Dan Jurafsky

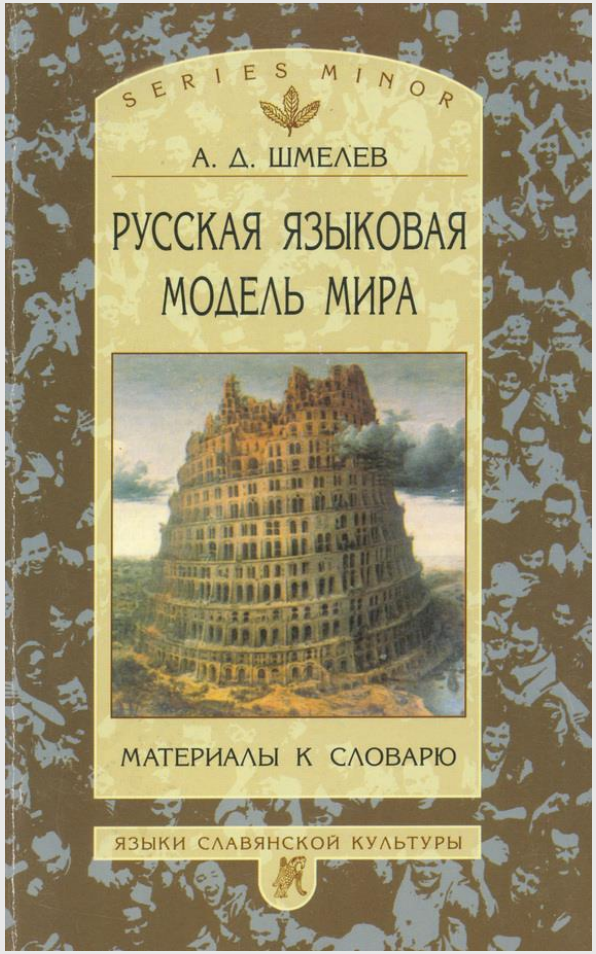
Терминологическое замечание

(statistical) language model (LM)

→

языковая модель

модель языка

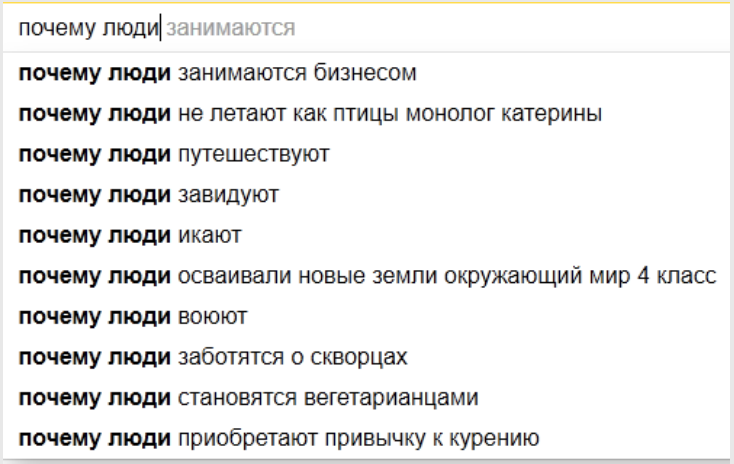


ВВЕДЕНИЕ

Языковые модели

Задача: определить вероятность последовательности слов

- распознавание речи
  скрипка лиса vs. *скрип колеса*
- исправление опечаток
  курсовая работа vs. *курсовая робота*
- генерация текста (в том числе спама ☺)
- машинный перевод
  крепкий чай vs. *сильный чай*
- …



"время и стекло"

All Videos Images News

About 2,100,000 results (0.30 seconds)

"время истекло"

All Videos Images News

About 291,000 results (0.29 seconds)

Предсказание следующего слова

- яблоко от яблони…
- круглый *год/сирота/…*
- выпил *чаю/воды/стакан/…*

309	и
192	стакан
138	еще
120	рюмку
86	залпом
82	с
69	водки
66	ее
66	за
63	две
55	его
48	и
45	а
43	чаю
42	бы
42	в
39	два
39	свой

Модели на основе n-грамм

- Как вычислить вероятность новой последовательности?

$P(S) = P(w_1, w_2, \ldots, w_n)$

- Произведение вероятностей:

$P(S) = P(w_1)P(w_2|w_1)\ P(w_3|w_1,w_2)\ldots P(w_n|w_1,w_2,\ldots,w_{n-1})$

- Предположим, что это Марковский процесс (т.е. ограничим влияние «истории»)
  - униграммная $P(w_n)$
  - биграммная $P(w_n|w_{n-1})$
  - триграммная $P(w_n|w_{n-2},w_{n-1})$

Модели на основе n-грамм –2

- моделируют *локальные* зависимости
- n-граммы более высоких порядков: мало данных, нет «обобщения»

ОЦЕНКА ВЕРОЯТНОСТЕЙ

Оценка вероятностей биграмм

- Метод максимального правдоподобия (Maximum Likelihood Estimate)

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Пример

<s>иду шагаю по москве</s>

<s>иду на вы</s>

<s>шагаю босиком по улице</s>

P(иду|<s>) = 0.67
P(шагаю|<s>) = 0.33
P(шагаю|иду) = 0.5

P(москве|по) = 0.5

P(по|босиком) = 1.0

…

Berkeley Restaurant Project corpus

9,222 предложений

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

[Dan Jurafsky]

Частоты биграмм

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

[Dan Jurafsky]

Вероятности биграмм

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

[Dan Jurafsky]

Вероятность предложения

P(<s> I want english food </s>) =

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

× P(</s>|food)

= .000031

[Dan Jurafsky]

О чем говорят вероятности?

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

[Dan Jurafsky]

Практические соображения

- лучше оперировать логарифмами вероятностей
  - избежать переполнения
  - сложение быстрее, чем умножение

$$\log(p_1 \times p_2 \times p_3 \times p_4) =$$
$$= \log p_1 + \log p_2 + \log p_3 + \log p_4$$

[Dan Jurafsky]

Оценка

- «внешняя» оценка: насколько хорошо модель помогает решить задачу
- «внутренняя» оценка модели : коэффициент неопределенности (перплексия, perplexity):
  - хорошая модель приписывает более высокую вероятность последовательности (предложению), которая действительно встречается в тексте

Перплексия

- Обратная вероятность тестового набора, нормализованная на длину

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

- связана с энтропией

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_{i-1})}}$$

[Dan Jurafsky]

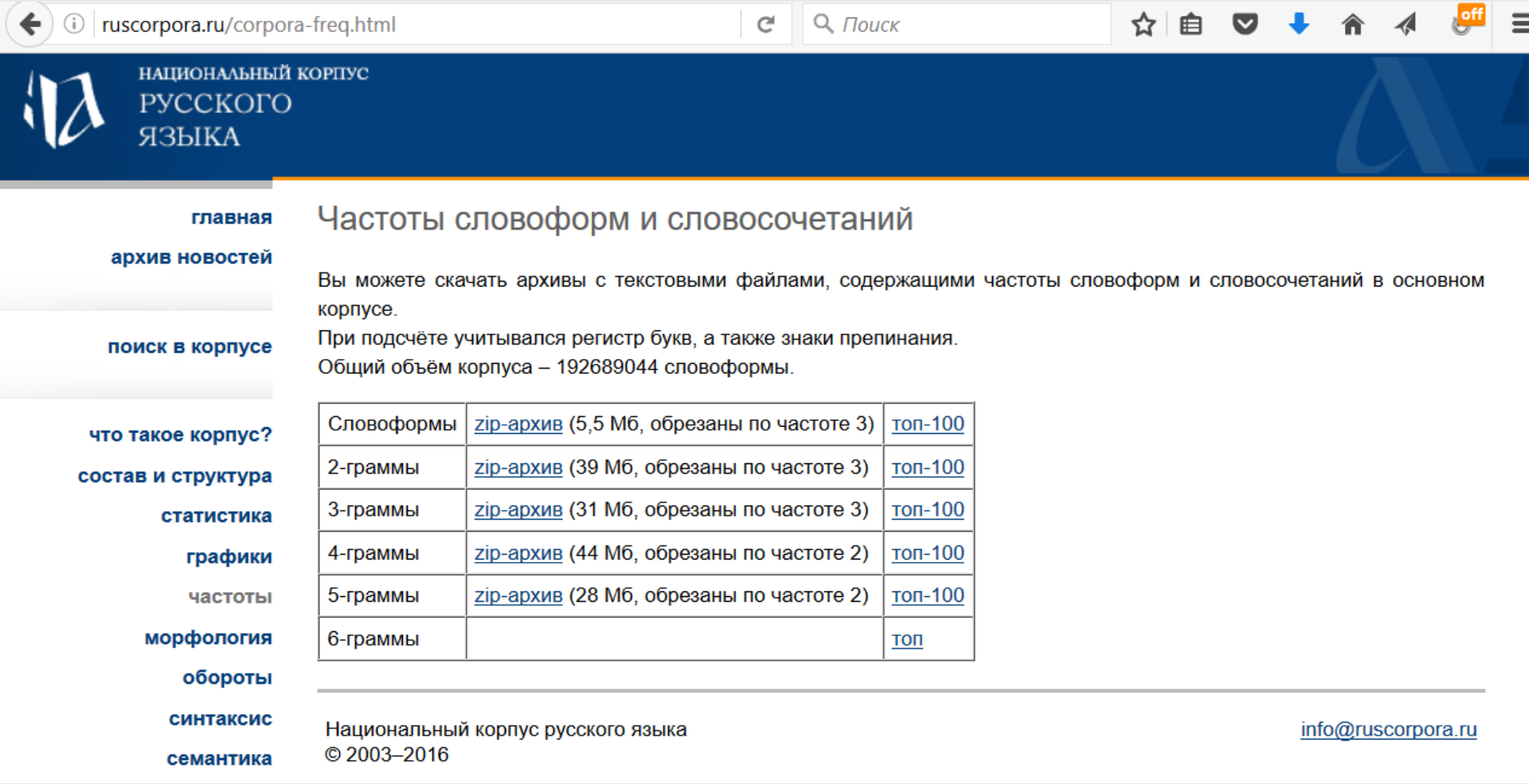
Ниже перплексия = лучше модель

- Обучение на 38 млн. слов, тестирование на 1.5 млн., WSJ

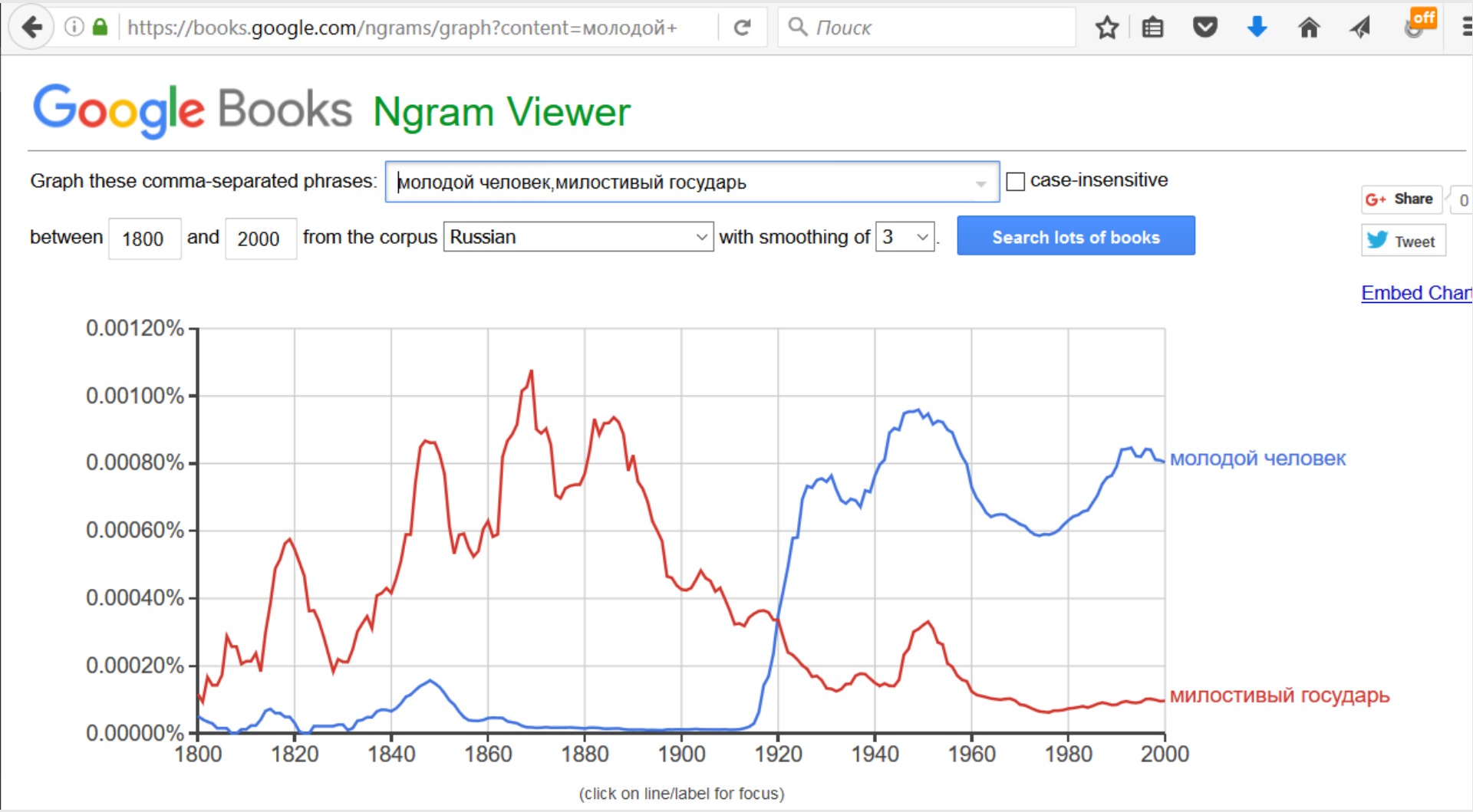
	Unigram	Bigram	Trigram
Perplexity	962	170	109

[Dan Jurafsky]

ДАННЫЕ


ruscorpora.ru/corpora-freq.html
Поиск
НАЦИОНАЛЬНЫЙ КОРПУС
РУССКОГО
ЯЗЫКА
главная
архив новостей
поиск в корпусе
что такое корпус?
состав и структура
статистика
графики
частоты
морфология
обороты
синтаксис
семантика
Частоты словоформ и словосочетаний
Вы можете скачать архивы с текстовыми файлами, содержащими частоты словоформ и словосочетаний в основном корпусе.
При подсчёте учитывался регистр букв, а также знаки препинания.
Общий объём корпуса – 192689044 словоформы.
Словоформы | zip-архив (5,5 Мб, обрезаны по частоте 3) | топ-100
2-граммы | zip-архив (39 Мб, обрезаны по частоте 3) | топ-100
3-граммы | zip-архив (31 Мб, обрезаны по частоте 3) | топ-100
4-граммы | zip-архив (44 Мб, обрезаны по частоте 2) | топ-100
5-граммы | zip-архив (28 Мб, обрезаны по частоте 2) | топ-100
6-граммы | | топ
Национальный корпус русского языка
© 2003–2016
info@ruscorpora.ru

Google Books n-gram viewer



Google Books n-grams

Russian

Version 20120701

total_counts

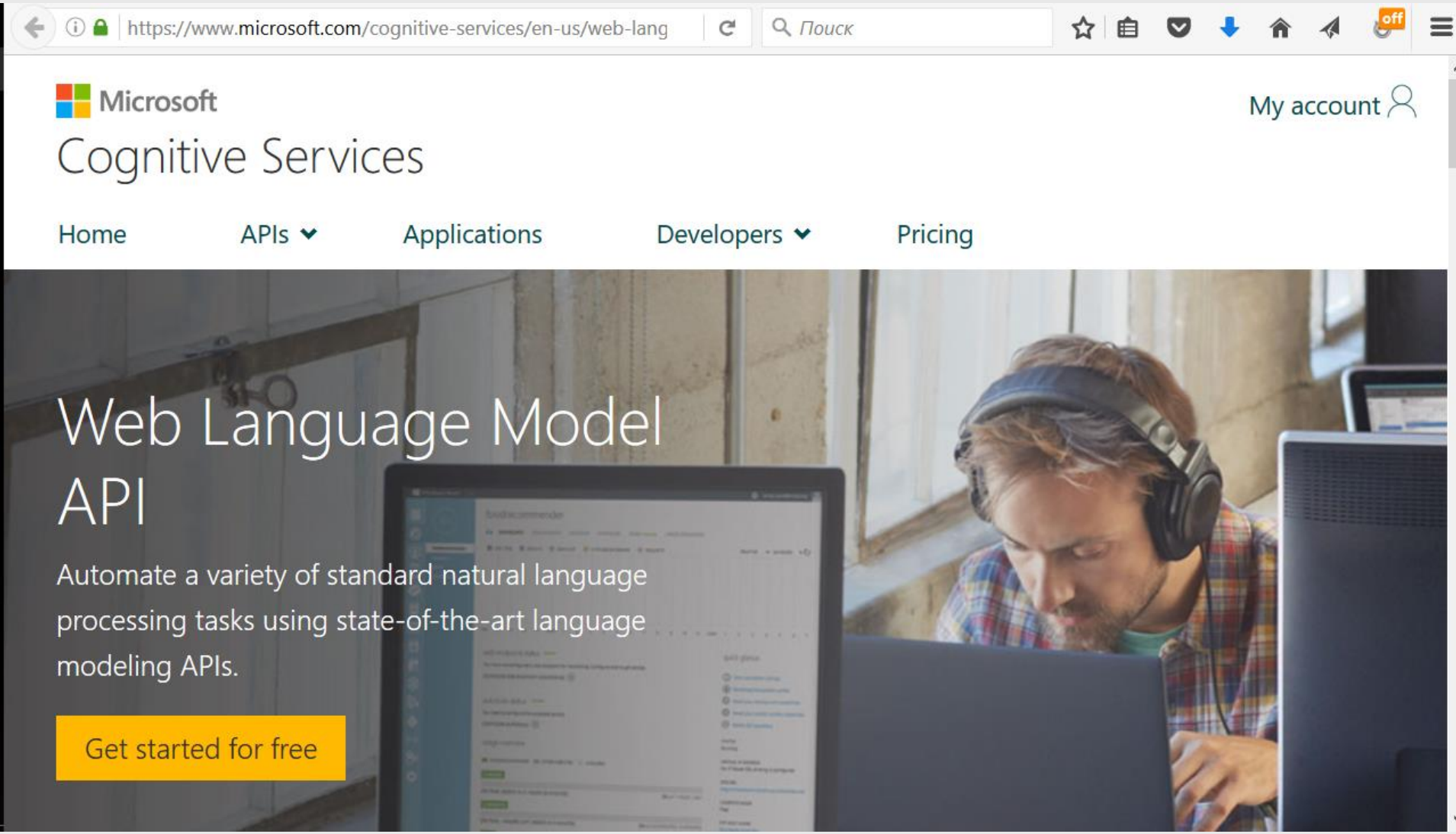
1-grams 0 1 2 3 4 5 6 7 8 9 a b c d e f g h i j k l m n o other p pos punctuation q r s t u v w x y z

2-grams 0 1 2 3 4 5 6 7 8 9 _ADJ_ _ADP_ _ADV_ _CONJ_ _NOUN_ _NUM_ _PRT_ _VERB_ a_ aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az b_ ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp br bs bt bu bv bx by bz c_ ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cx cy cz d_ da db dc dd de df dg dh di dj dk dl dm dn do dp dr ds dt du dv dw dx dy dz e_ ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ew ex ey ez f_ fa fb fc fd fe ff fg fh fi fj fk fl fm fn fo fp fr fs ft fu fv fx fy fz g_ ga gb gc gd ge gf gg gh gi gj gk gl gm gn go gp gr gs gt gu gv gx gy gz h_ ha hb hc hd he hf hg hh hi hj hk hl hm hn ho hp hr hs ht hu hv hx hy hz i_ ia ib ic id ie if ig ih ii ij ik il im in io ip iq ir is it iu iv iw ix iy iz j_ ja jb jc jd je jf jg jh ji jj jk jl jm jn jo jp jr js jt ju jv jw jx jy jz k_ ka kb kc kd ke kf kg kh ki kj kk kl km kn ko kp kr ks kt ku kv kx ky kz l_ la lb lc ld le lf lg lh li lj lk ll lm ln lo lp lr ls lt lu lv lw lx ly lz m_ ma mb mc md me mf mg mh mi mj mk ml mm mn mo mp mr ms mt mu mv mw mx my mz n_ na nb nc nd ne nf ng nh ni nj nk nl nm nn no np nr ns nt nu nv nw nx ny nz o_ oa ob oc od oe of og oh oi oj ok ol om on oo op or os ot other ou ov ow ox oy oz p_ pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pr ps pt pu punctuation pv px py pz q_ qa qi qn qu qx r_ ra rb rc rd re rf rg rh ri rj rk rl rm rn ro rp rr rs rt ru rv rw rx ry rz s_ sa sb sc sd se sf sg sh si sj sk sl sm sn so sp sq sr ss st su sv sw sx sy sz t_ ta tb tc td te tf tg th ti tj tk tl tm tn to tp tr ts tt tu tv tw tx ty tz u_ ua ub uc ud ue uf ug uh ui uj uk ul um un uo up ur us ut uu uv ux uy uz v_ va vb vc vd ve vf vg vh vi vj vk vl vm vn vo vp vr vs vt vu vv vx vy vz w_ wa wc we wh wi wj wl wm wn wo wr ws wt wu ww wx wy x_ xa xb xc xd xe xf xg xh xi xj xk xl xm xn xo xp xr xs xt xu xv xx xy y_ ya yb yc yd ye yf yg yh yi yj yk yl ym yn yo yp yr ys yt yu yv yx yy yz z_ za zb zc zd ze zf zg zh zi zj zk zl zm zn zo zp zr zs zt zu zv zw zx zy zz

3-grams 0 1 2 3 4 5 6 7 8 9 _ADJ_ _ADP_ _ADV_ _CONJ_ _NOUN_ _NUM_ _PRT_ _VERB_ a_ aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az b_ ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp br bs bt bu bv bx by bz c_ ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cx cy cz d_ da db dc dd de df dg dh di dj dk dl dm dn do dp dr ds dt du dv dw dx dy dz e_ ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ex ey ez f_ fa fb fc fd fe ff fg fh fi fj fk fl fm fn fo fp fr fs ft fu fv fx fy fz g_ ga gb gc gd ge gf gg gh gi gj gk gl gm gn go gp gr gs gt gu gv gx gy gz h_ ha hb hc hd he hf hg hh hi hj hk hl hm hn ho hp hr hs ht hu hv hx hy hz i_ ia ib ic id ie if ig ih ii ij ik il im in io ip iq ir is it iu iv iw ix iy iz j_ ja jb jc jd je jf jg jh ji jj jk jl jm jn jo jp jr js jt ju jv jw jx jy jz k_ ka kb kc kd ke kf kg kh ki kj kk kl km kn ko kp kr ks kt ku kv kx ky kz l_ la lb lc ld le lf lg lh li lj lk ll lm ln lo lp lr ls lt lu lv lx ly lz m_ ma mb mc md me mf mg mh mi mj mk ml mm mn mo mp mr ms mt mu mv mx my mz n_ na nb nc nd ne nf ng nh ni nj nk nl nm nn no np nr ns nt nu nv nw nx ny nz o_ oa ob oc od oe of og oh oi oj ok ol om on oo op or os ot other ou ov ow ox oy oz p_ pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pr ps pt pu punctuation pv px py pz q_ qa qn qu qx r_ ra rb rc rd re rf rg rh ri rj rk rl rm rn ro rp rr rs rt ru rv rw rx ry rz s_ sa sb sc sd se sf sg sh si sj sk sl sm sn so sp sq sr ss st su sv sw sx sy sz t_ ta tb tc td te tf tg th ti tj tk tl tm tn to tp tr ts tt tu tv tw tx ty tz u_ ua ub uc ud ue uf ug uh ui uj uk ul um un uo up ur us ut uu uv ux uy uz v_ va vb vc vd ve vf vg vh vi vj vk vl vm vn vo vp vr vs vt vu vv vx vy vz w_ wa we wh wi wj wl wn wo wr ws wt wu ww wx x_ xa xc xd xe xf xg xh xi xj xk xl xm xn xo xp xr xs xt xu xv xx xy y_ ya yb yc yd ye yf yg yh yi yj yk yl ym yn yo yp yr ys yt yu yv yx yy yz z_ za zb zc zd ze zf zg zh zi zj zk zl zm zn zo zp zr zs zt zu zv zw zx zy zz

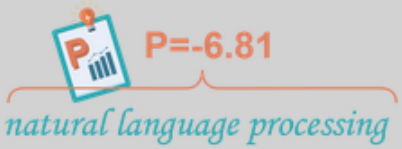
4-grams 0 1 2 3 4 5 6 7 8 9 _ADJ_ _ADP_ _ADV_ _CONJ_ _NOUN_ _NUM_ _PRT_ _VERB_ a_ aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az b_ ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp br bs bt bu bv bx by bz c_ ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cx cy cz d_ ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ex ey ez f_ fa fb fc fd

storage.googleapis.com/books/ngrams/books/googlebooks-rus-all-3gram-20120701-gp.gz


https://www.microsoft.com/cognitive-services/en-us/web-lang
Поиск
Microsoft
Cognitive Services
My account
Home
APIs
Applications
Developers
Pricing
Web Language Model API
Automate a variety of standard natural language processing tasks using state-of-the-art language modeling APIs.
Get started for free

Joint probabilities

Calculate how often a particular sequence of words appear together.



Conditional probabilities

Given a sequence of words, calculate how often a particular word tends to follow.

Next word completions

Given a sequence of words, get the list of words most likely to follow.

СГЛАЖИВАНИЕ ЛАПЛАСА

Новые последовательности

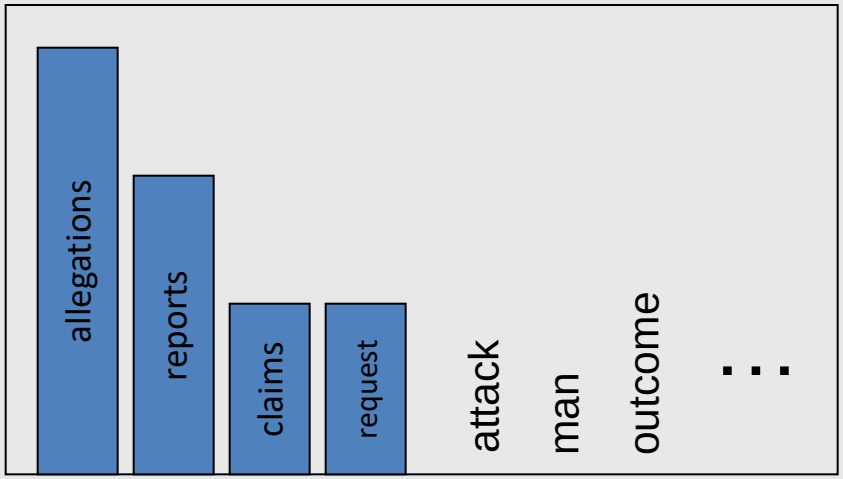
- языковая модель должна *обобщать* (а не повторять) данные, на которых она обучалась
- всегда будут новые последовательности слов, которые не встречались в корпусе для обучения
- «новые» n-граммы (в тестовых данных, но не в данных для обучения) «ломают» модель

Сглаживание

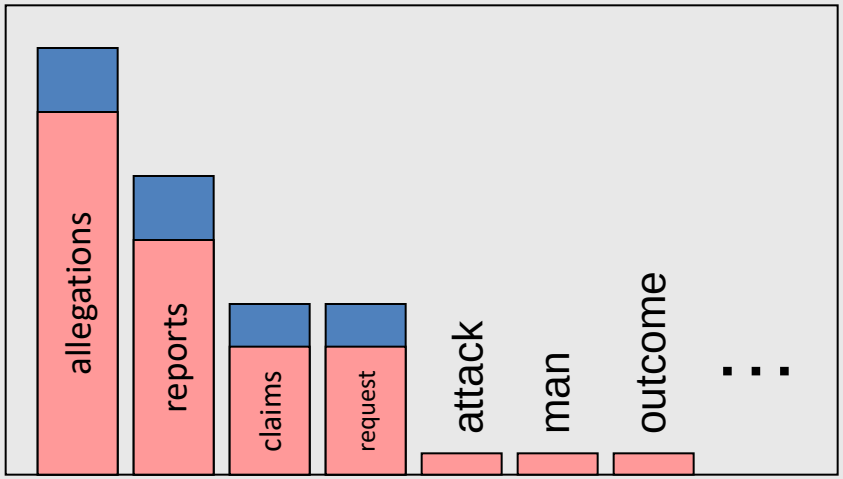
- «зарезервировать» часть вероятностной массы для событий, которые еще не встречались

Пример

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



[Dan Jurafsky]

Сглаживание Лапласа

- или сглаживание «+1»
- Предположим, что мы встречали каждую пару слов на один раз больше

- оценка максимального правдоподобия:

$$P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

- оценка «+1»:

$$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$$

[Dan Jurafsky]

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

[Dan Jurafsky]

Сглаженные вероятности

$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

[Dan Jurafsky]

Восстановленные частоты

$$c^*(w_{n-1}w_n) = \frac{[C(w_{n-1}w_n)+1] \times C(w_{n-1})}{C(w_{n-1})+V}$$

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

[Dan Jurafsky]

Сравнение

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

[Dan Jurafsky]

Сглаживание «+1»

- плохо работает для языковых моделей
  - более мягкий вариант «+α»:

$$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + \alpha}{c(w_{i-1}) + \alpha V}$$

- используется для классификации текстов
  - и там, где меньше нулевых оценок

ОТКАТ И ИНТЕРПОЛЯЦИЯ

Откат и интерполяция

- Откат (backoff)
  - переход к с более низким n, для которых достаточно данных (3→2→1)
- Интерполяция
  - смесь униграмм, биграмм, триграмм

Линейная интерполяция

$$\hat{P}(w_n|w_{n-1}w_{n-2}) = \lambda_1 P(w_n|w_{n-1}w_{n-2}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n)$$

$$\sum_i \lambda_i = 1$$

- λ могут зависеть от контекста

$$\hat{P}(w_n|w_{n-2}w_{n-1}) = \lambda_1(w_{n-2}^{n-1}) P(w_n|w_{n-2}w_{n-1}) + \lambda_2(w_{n-2}^{n-1}) P(w_n|w_{n-1}) + \lambda_3(w_{n-2}^{n-1}) P(w_n)$$

[Dan Jurafsky]

Для очень больших корпусов (веб)

- Наивный откат (*stupid backoff*) [Brants *et al*. 2007]
- Использование относительных частот

$$S(w_i \mid w_{i-k+1}^{i-1}) = \begin{cases} \dfrac{\text{count}(w_{i-k+1}^{i})}{\text{count}(w_{i-k+1}^{i-1})} & \text{если } \text{count}(w_{i-k+1}^{i}) > 0 \\ 0.4S(w_i \mid w_{i-k+2}^{i-1}) & \text{иначе} \end{cases}$$

$$S(w_i) = \frac{\text{count}(w_i)}{N}$$

[Dan Jurafsky]

СГЛАЖИВАНИЕ ГУДА-ТЬЮРИНГА

Идея метода: рыбалка

- пример Josh Goodman
- Улов: 10 карпов, 3 окуня, 2 сига, 1 форель, 1 лосось, 1 угорь = 18 рыб
- Какова вероятность, что следующая рыба – форель? 1/18
- Какова вероятность поймать новую рыбу (сома или щуку)?
  - Предположим, что вероятность поймать новую равна вероятности поймать рыбу, которая до сих пор попадалась только один раз: 3/18
- Тогда какова вероятность поймать форель? < 1/18

[Dan Jurafsky]

Пример

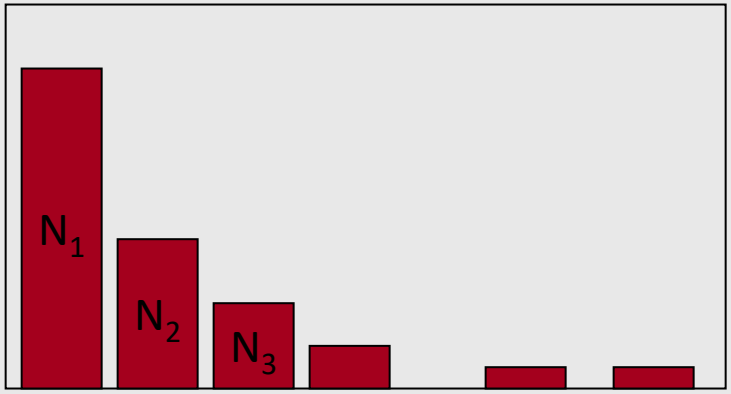
$$P^*_{GT}(\text{нулевая вероятность}) = \frac{N_1}{N} \qquad c^* = \frac{(c+1)N_{c+1}}{N_c}$$

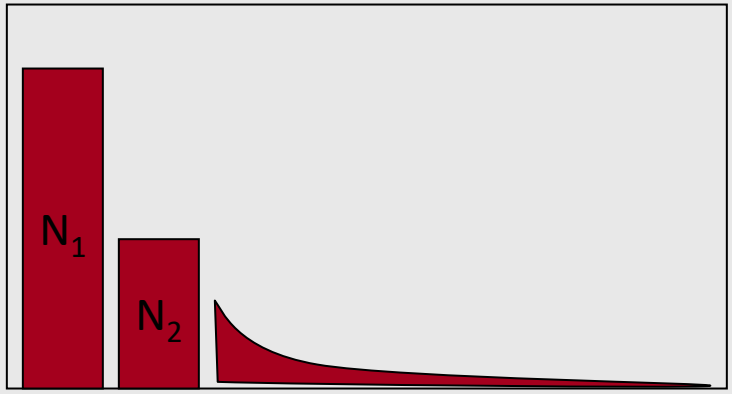
- не видели (сом или щука)
  - c = 0:
  - MLE p = 0/18 = 0
  - P^*_{GT} (не видели) = N_1/N = 3/18
- видели однажды (форель)
  - c = 1
  - MLE p = 1/18
  - C*(форель) = 2 * N2/N1
    = 2 * 1/3
    = 2/3
  - P^*_{GT}(форель) = 2/3 / 18 = 1/27

[Dan Jurafsky]

Аппроксимация низких частот

- В области больших k N_k идут не подряд (высокие частоты – редкие) и не позволяют использовать формулу напрямую
- Аппроксимация для больших k с помощью степенной функции





[Dan Jurafsky]

Пример

- [Church and Gale, 1991]
- новостной корпус 22 млн. слов

$$c^* = \frac{(c+1)N_{c+1}}{N_c}$$

Count c	Good Turing c*
0	.0000270
1	0.446
2	1.26
3	2.24
4	3.24
5	4.22
6	5.19
7	6.21
8	7.24
9	8.25

[Dan Jurafsky]

Другие подходы

- Witten-Bell:
  учет разнообразия продолжения
  (низкое разнообразие слов, которые следуют за «високосный», по сравнению с «жадному»
  → новая биграмма с «идиотический» более вероятна)
- Kneser-Ney:
  «в другую сторону» – учет вероятности слова быть продолжением (биграммы), разнообразия «истории»

Оценка

Perplexity for language models trained on the Europarl corpus

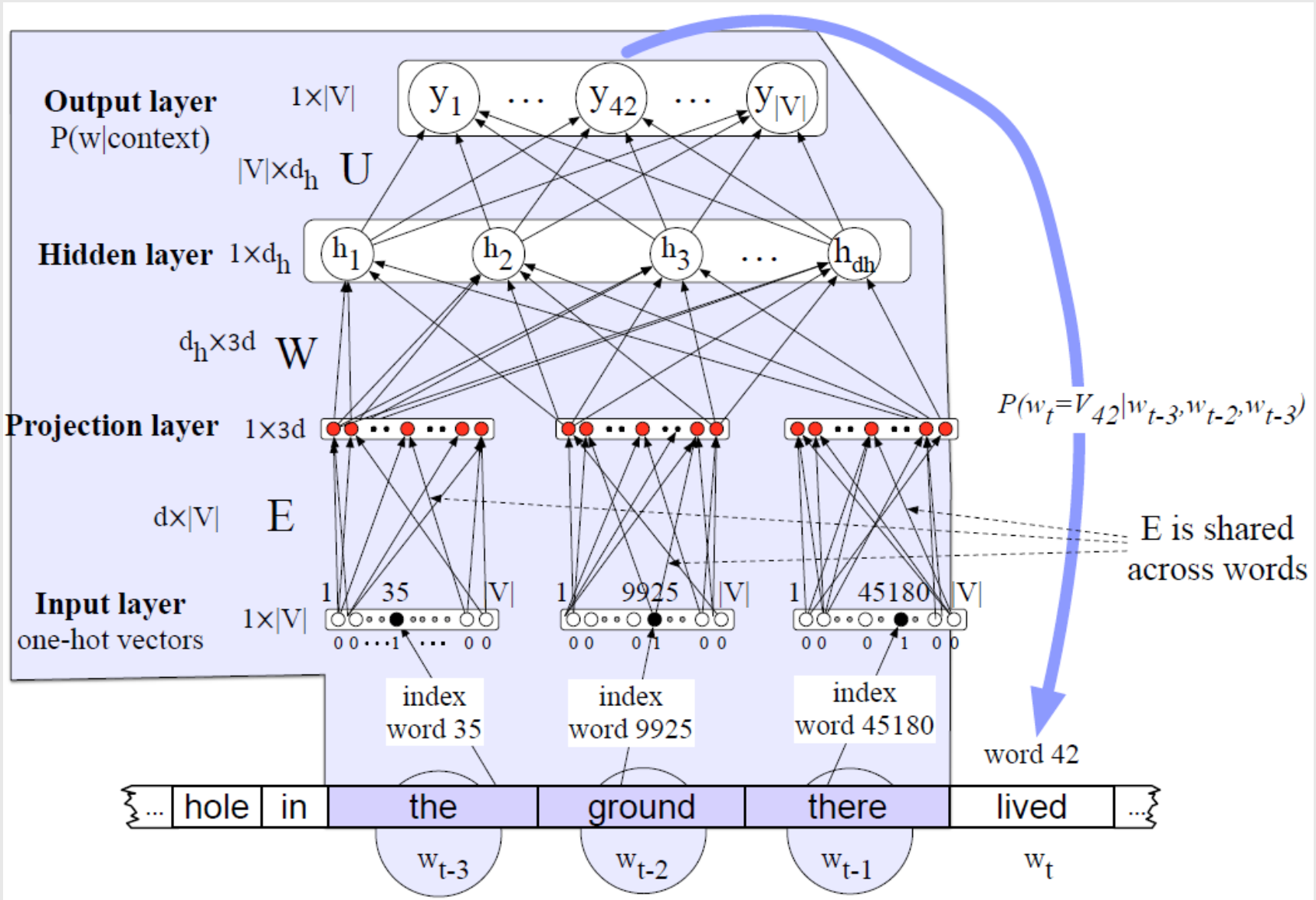
Smoothing method	**bigram**	**trigram**	**4-gram**
Good-Turing	96.2	62.9	59.9
Witten-Bell	97.1	63.8	60.4
Modified Kneser-Ney	95.4	61.6	58.6
Interpolated Modified Kneser-Ney	94.5	59.3	54.0

[Philipp Koehn]

НЕЙРОННЫЕ ЯЗЫКОВЫЕ МОДЕЛИ

Преимущества

- не надо сглаживать
- учет более длинных «историй»
- обобщение


Output layer
P(w|context)
1×|V|
|V|×dh U
Hidden layer 1×dh
dh×3d W
Projection layer 1×3d
d×|V| E
Input layer
one-hot vectors 1×|V|
y1 ... y42 ... y|V|
h1 h2 h3 ... hdh
1 35 |V|
1 9925 |V|
1 45180 |V|
E is shared across words
index word 35
index word 9925
index word 45180
word 42
... hole in the ground there lived ...
wt-3 wt-2 wt-1 wt

SLP, 3ed