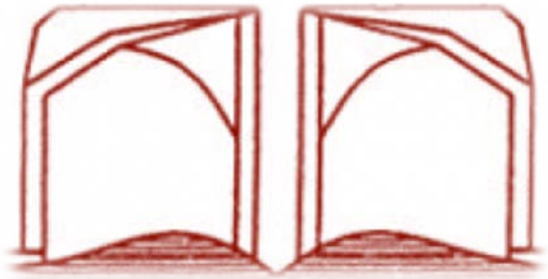


آشنایی با زبان‌شناسی رایانشی

مدل‌سازی زبانی

هادی ویسی

h.veisi@ut.ac.ir

دانشگاه تهران – دانشکده علوم و فنون نوین

فهرست

- معرفی مدل‌سازی زبانی و مدل‌سازی زبانی آماری
- شمارش کلمات (و قانون زیف)
- مدل N-Gram
  - نحوه محاسبه
  - نکات کاربردی
- هموارسازی
  - روش Add-One (هموارسازی لاپلاس)
  - روش تخفیف Good-Turing
  - روش Absolute Discounting Interpolation
  - روش Kneser-Ney
- ارزیابی مدل‌های زبانی (Perplexity)
- ابزارها

مدل‌سازی زبانی (Language Modeling) . . .

- مدل زبانی (Language Model)
  - مدل کردن نحوه رخداد توالی کلمات در زبان

- انواع مدل زبانی
  - آماری
    - به یک دنباله از کلمات زبان مانند $W=w_1w_2...w_m$ یک مقدار احتمال $P(W)$ نسبت می‌دهد
  - ساختاری
    - با استفاده از تعدادی قواعد زبانی، نحوه توالی لغات را مشخص می‌کند

مدل‌سازی زبانی

- سطوح مختلف مدل‌سازی زبانی
  - واژگانی محلی
  - نحوی
  - معنایی
- کاربردهای مدل زبانی
  - پیش‌بینی کلمات
  - بازشناسی گفتار
  - درک زبان طبیعی
  - ترجمهٔ ماشینی
  - بازشناسی نویسه‌های نوری
  - ...

مدل‌سازی زبانی آماری . . .

- چرا مدل زبانی آماری؟
  - انتساب مقدار احتمال به یک جمله
  - کاربرد در ترجمه
    - P(**high** winds tonight) > P(**large** winds tonight)
  - کاربرد در خطایاب
    - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
  - کاربرد در تشخیص گفتار
    - P(I saw a van) >> P(eyes awe of an)

مدل‌سازی زبانی آماری . . .

- هدف
  - محاسبه احتمال رخداد یک جمله (دنباله‌ای از کلمات)
    - $P(W) = P(w_1,w_2,w_3,w_4,w_5...w_m)$
    - P(بازیکنان تیم ملی نوجوانان جودو)
  - محاسبه احتمال کلمه بعدی (با این فرض که کلمات قبلی را داریم)
    - $P(w_5|w_1,w_2,w_3,w_4)$
    - P(بازیکنان تیم ملی نوجوانان |جودو)
- مدل زبانی (LM)
  - محاسبه $P(w_m|w_1,w_2...w_{m-1})$ یا $P(W)$

مدل‌سازی زبانی آماری . . .

- نحوه محاسبه P(W)
  - P(بازیکنان تیم ملی نوجوانان جودو)

- استفاده از قانون زنجیره‌ای (Chain Rule) احتمال
  - برای دو متغیر: $P(A,B) = P(A)P(B|A)$
    - P(بازیکنان تیم)=P(بازیکنان)P(بازیکنان|تیم)
  - برای چهار متغیر: $P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$
    - P(بازیکنان تیم ملی نوجوانان)=P(بازیکنان)P(بازیکنان|تیم)P(بازیکنان تیم|ملی)P(بازیکنان تیم ملی|نوجوانان)
  - حالت کلی: $P(x_1,x_2,x_3,\ldots,x_m) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_m|x_1,\ldots,x_{m-1})$

مدل‌سازی زبانی آماری . . .

- داریم

$$P(W) = P(w_1 w_2 \cdots w_m)$$
$$= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)\cdots P(w_m \mid w_1 \cdots w_{m-1})$$
$$= \prod_{i=1}^{m} P(w_i \mid w_1 \cdots w_{i-1})$$

- مثال فارسی
  - p (بازیکنان تیم ملی نوجوانان جودو)=
    P (بازیکنان) × P(بازیکنان|تیم) × P(بازیکنان تیم|ملی) × P(بازیکنان تیم ملی|نوجوانان)
    × P(بازیکنان تیم ملی نوجوانان|جودو)

- مثال انگلیسی
  - P(“its water is so transparent”) =
    = P(its) × P(water | its) × P(is | its water)
    × P(so | its water is) × P(transparent | its water is so)

مدل‌سازی زبانی آماری

نحوه محاسبه احتمال‌ها؟

- **شمارش**

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

- **این کار عملی نیست! چون ...**
  - تعداد جملات مختلف بی‌نهایت است
  - هیچ مجموعه داده‌ای برای شمارش آن موجود نیست

شمارش کلمات . . .

- اولین گام در مدل‌سازی آماری زبان، شمارش انواع کلمات در یک پیکره متنی است.

- پیکره متنی (Text Corpus)
  - مجموعه‌ای بزرگ از متون مختلف که از منابع گوناگون گردآوری شده‌اند
    - پیکره باید نماینده کل زبان باشد: تنوع از انواع متن و حجم قابل قبول
  - تعداد کل کلمات (word tokens) در پیکره‌های متنی در حد چند صد میلیون تا چند میلیارد
  - تعداد انواع کلمات (word types) یا کلمات یکتا در حد چند صد هزار یا چند میلیون (بسته به نوع متون)

- قبل از شمارش کلمات پیکره متنی باید واحدسازی و نرمال‌سازی شود

شمارش کلمات . . .

- شمارش کلمات
  - تعیین انواع word typeها در پیکره متنی و شمارش تعداد رخداد هر یک از آنها
  - با یک بار پیمایش پیکره متنی می‌توان تعداد انواع کلمات (word typeها) را در پیکره متنی شمارش کرد
  - در یک پیکرهٔ متنی بزرگ از متون واقعی، معمولاً تعداد کمی از کلمات با فراوانی بالا و تعداد زیادی از کلمات با فراوانی پایین رخ می‌دهند.
  - معمولاً ایست واژه‌ها (stop word)ها بیشترین فراوانی را در پیکره متنی دارند.
    - کلماتی مانند «از، به، اما، و، ...»

شمارش کلمات: قانون زیف . . .

- قانون زیف (Zipf's Law)
  - کلمات موجود در پیکره متنی را برحسب فراوانی (از بیشتر به کمتر) مرتب می‌کنیم و به ترتیب به آنها رتبه (rank) 1 تا N می‌دهیم.
    - پرتکرارترین کلمه دارای رتبه ۱ و کم تکرارترین رتبه N

- بین فراوانی کلمات و رتبهٔ آنها یک تناسب معکوس وجود دارد.

$$f(w) = \frac{C}{z(w)^a}$$

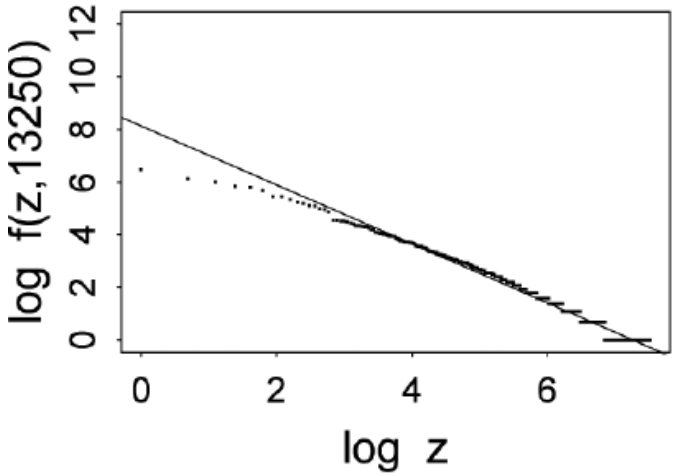
- f(w)= فراوانی کلمه w
- z(w)= رتبه کلمه w
- C و a= مقادیر ثابت (پارامترهای مدل)

شمارش کلمات: قانون زیف . . .

- حالت لگاریتمی

$$f(w) = \frac{C}{z(w)^a} \Rightarrow \log f(w) = \log C - a \log z(w)$$

- بین لگاریتم فراوانی و لگاریتم رتبه رابطهٔ خطی وجود دارد.
- پارامترهای C و a را می توان برای هر پیکرهٔ متنی محاسبه کرد.

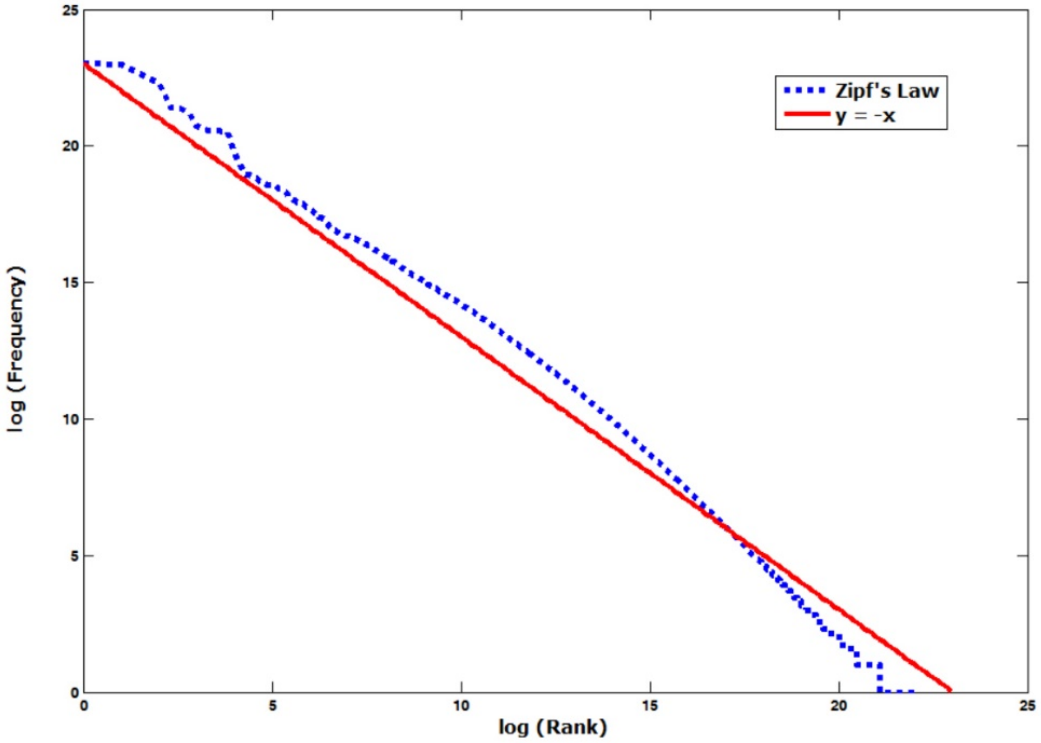


شمارش کلمات: قانون زیف . . .

فراوانی رتبه

.
.
.

- مثال: پیکره کُردی ئاسوسافت



شمارش کلمات

- **انتخاب واژگان (Lexicon) از پیکره**
  - در بسیاری از کاربردها، کلمات پرتکرارتر پیکره به عنوان واژگان انتخاب می‌شوند
  - دو راه برای انتخاب کلمات پرتکرار
    - گذاشتن حد آستانه برروی تعداد کلمات انتخابی (مثلاً ۱۰۰۰۰ کلمهٔ پرکاربردتر)
    - گذاشتن حد آستانه برروی فراوانی کلمات (مثلاً انتخاب کلمات با فراوانی بالاتر از ۱۰)
- **حذف برخی کلمات در برخی از کاربردها**
  - حذف ایست واژه‌ها از لیست کلمات در بازیابی اطلاعات
  - در برخی کاربردها فقط ریشهٔ کلمات انتخابی در واژگان قرار می‌گیرند.
- **افزودن برخی کلمات (علاوه بر کلمات پرکاربرد) به واژگان**
  - فرمان‌های صوتی در تشخیص گفتار

مدل‌سازی زبانی آماری . . .

- نحوه محاسبه احتمال‌ها؟

- نحوه محاسبه احتمال‌ها؟

  - شمارش

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

  - این کار عملی نیست! چون ...
    - تعداد جملات مختلف بی‌نهایت است
    - هیچ مجموعه داده‌ای برای شمارش آن موجود نیست

- فرض مارکوف (Markov)
  - مرتبه یک: فقط وابسته به یک متغیر (کلمه) قبل

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$$

  - مرتبه دو: فقط وابسته به دو متغیر (کلمه) قبل

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$$

مدل‌سازی زبانی آماری . . .

$$P(W) = P(w_1 w_2 \cdots w_m)$$
$$= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)\cdots P(w_m \mid w_1 \cdots w_{m-1})$$
$$= \prod_{i=1}^{m} P(w_i \mid w_1 \cdots w_{i-1})$$

- بر اساس فرض مارکوف

$$P(W) = P(w_1 w_2 \cdots w_m) \cong \prod_{i=1}^{m} P(w_i \mid w_{i-(k)} \cdots w_{i-1})$$

- که در محاسبه احتمال $p(w_i \mid w_1,w_2,w_3,w_4,w_5 \ldots w_{i-1})$، به جای در نظر گرفتن همه کلمات قبلی کلمه iام، تعداد کمتری کلمه (تنها k مورد نزدیک‌تر) را در نظر گرفته است

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

مدل‌سازی زبانی آماری: N-Gram . . .

- ساده‌ترین حالت: در نظر نگرفتن وابستگی به کلمات قبل

$$P(W) = P(w_1 w_2 \cdots w_m) \cong \prod_{i=1}^{m} P(w_i \mid w_{i-(k)} \cdots w_{i-1}) \cong \prod_{i=1}^{m} P(w_i) = P(w_1)P(w_2)P(w_3)\cdots P(w_m)$$

k=0
k=N-1 → N=1

- مثال
  - P(بازیکنان تیم ملی نوجوانان جودو)=P(بازیکنان)P(تیم)P(ملی)P(نوجوانان)P(جودو)

Mono-Gram
UniGram
1-Gram

$$P_{monogram}(w_i) = \frac{Count(w_i)}{Count(AllWords)} = \frac{N(w_i)}{N_{total}}$$

- مثال
  - در متنی با ۱۰.۰۰۰ کلمه، تعداد تکرار کلمه «ملی» ۱۰ باشد، آنگاه p=10/10,000=0.001

Hadi Veisi (h.veisi@ut.ac.ir)

مدل‌سازی زبانی آماری: N-Gram . . .

- مدل دوتایی (Bi-Gram): در نظر گرفتن یک کلمه قبل از هر کلمه
  - N=2

$$P(w_1 w_2 \cdots w_m) \cong \prod_{i=1}^{m} P(w_i \mid w_{i-(k)} \cdots w_{i-1}) \cong \prod_{i=1}^{m} P(w_i \mid w_{i-1}) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)\cdots P(w_m \mid w_{m-1})$$

  - یعنی

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

  - مثال
    - p (بازیکنان تیم ملی نوجوانان جودو)=
      P (بازیکنان) × P(بازیکنان|تیم) × P(تیم|ملی) × P(ملی|نوجوانان) × P(نوجوانان|جودو)
  - محاسبه دوتایی

$$P_{bigram}(w_j \mid w_i) = \frac{N(w_i w_j)}{N(w_i)}$$

    - برای محاسبه P(ملی|نوجوانان)، تعداد تکرار «ملی نوجوانان» تقسیم بر تعداد تکرار «ملی»

مدل‌سازی زبانی آماری: N-Gram . . .

- مدل سه‌تایی (Tri-Gram): در نظر گرفتن دو کلمه قبل از هر کلمه
  - N=3

$$P(W) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)P(w_4 \mid w_2 w_3)\cdots P(w_m \mid w_{m-2} w_{m-1})$$

- امکان توسعه به تعداد بیشتر: 4-Gram، 5-Gram و ...
- مقدار معمول: 2-Gram تا 5-Gram
- در عمل وابستگی زبانی بیشتر از ۵ کلمه است
  - "The computer which I had just put into the machine room on the fifth floor crashed"

مدل‌سازی زبانی آماری: N-Gram . . .

- مدل‌های n-gram با استفاده از شمارش دنبالهٔ کلمات در یک پیکره متنی بزرگ به دست می‌آیند
  - ابتدا تمام انواع کلمات پیکره شمارش می‌شود و یک lexicon شامل V کلمه از کلمات پرکاربرد (و سایر کلمات موردنظر) تعیین می‌گردد.
  - سایر کلمات همگی با یک نماد مشخص به عنوان کلمه خارج از واژگان (OOV) جایگزین می‌شوند.
  - سپس پیکره از ابتدا تا انتها پیمایش شده و تمام ترکیبات دوتایی، سه تایی، ... و nتایی از کلمات واژگان (و همچنین نماد OOV) شمارش می‌شود.

مدل‌سازی زبانی آماری: محاسبه N-Gram . . .

- مدل monogram (n=1)

$$P(W) = P(w_1)P(w_2)P(w_3)\cdots P(w_m)$$

$$P_{monogram}(w_i) = \frac{N(w_i)}{N_{total}}$$

- مدل bigram (n=2)

$$P(W) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)P(w_4 \mid w_3)\cdots P(w_m \mid w_{m-1})$$

$$P_{bigram}(w_j \mid w_i) = \frac{N(w_i w_j)}{N(w_i)}$$

- مدل trigram (n=3)

$$P(W) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)P(w_4 \mid w_2 w_3)\cdots P(w_m \mid w_{m-2} w_{m-1})$$

$$P_{trigram}(w_k \mid w_i w_j) = \frac{N(w_i w_j w_k)}{N(w_i w_j)}$$

کل کلمات مورد نظر

$$i, j, k = 1, \cdots, V$$

مدل‌سازی زبانی آماری: محاسبه N-Gram . . .

- **مثال**
  - **متن**

شروع جمله

پایان جمله

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

- **محاسبه دوتایی**

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

$$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \frac{2}{3} = .67$$

$$P(\texttt{</s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \frac{1}{3} = .33$$

تعداد «I do»ها به تعداد «I»ها

مدل‌سازی زبانی آماری: محاسبه N-Gram . . .

- ماتریس شمارش‌های bigram

$p(w_{i-1})$ \ $p(w_i)$	w_1	w_2	$\cdots$	w_V
w_1	$N(w_1w_1)$	$N(w_1w_2)$	$\cdots$	$N(w_1w_V)$
w_2	$N(w_2w_1)$	$N(w_2w_2)$	$\cdots$	$N(w_2w_V)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_V	$N(w_Vw_1)$	$N(w_Vw_2)$	$\cdots$	$N(w_Vw_V)$

$$\sum_{w_j} N(w_i w_j) = N(w_i)$$

- ماتریس احتمالات bigram
  - تقسیم مقادیر ماتریس شمارش بر شمارش monogramها

	w_1	w_2	$\cdots$	w_V
w_1	$N(w_1 \mid w_1)$	$N(w_2 \mid w_1)$	$\cdots$	$N(w_V \mid w_1)$
w_2	$N(w_1 \mid w_2)$	$N(w_2 \mid w_2)$	$\cdots$	$N(w_V \mid w_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_V	$N(w_1 \mid w_V)$	$N(w_2 \mid w_V)$	$\cdots$	$N(w_V \mid w_V)$

مدل‌سازی زبانی آماری: محاسبه N-Gram . . .

- مثال: Berkeley Restaurant Project …
  - جملاتی حاوی درخواست اطلاعات رستوران‌ها
    - 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
  - ماتریس شمارش‌های bigram

	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

تعداد «to spend»ها

(h.veisi@ut.ac.ir)

مدل‌سازی زبانی آماری: محاسبه N-Gram . . .

- مثال: Berkeley Restaurant Project
  - شمارش monogramها

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

  - ماتریس احتمالات bigram

	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

  - آنگاه
    - P(<s> I want english food </s>) =
      P(I | <s>)×P(want | I)×P(english | want)×P(food | english)×P(</s> | food) = 0.000031

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- آمارها سخن می‌گویند (از Berkeley Restaurant Project)
  - مدل کردن واقعیات
    - P(english | want) = .0011
    - P(chinese | want) = .0065
    - معنی: درخواست برای غذای چینی بیشتر از انگلیسی است
  - مدل کردن نحو
    - P(to | want) = .66
    - P(eat | to) = .28
    - P(food | to) = 0
    - P(want | spend) = 0
    - P (i | <s>) = .25

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- محاسبات در عمل
  - احتمال جمله‌ای با ۱۰ کلمه = صفر
  - دلیل: underflow شدن
    - ضرب چند عدد کوچک در همدیگر آنقدر کوچک است که در محاسبات رایانه‌ای معادل صفر است

  - راه حل: استفاده از لگاریتم احتمال

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

  - مزیت دیگر: ساده (سریع) بودن محاسبات (جمع به جای ضرب)

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- کلمات خارج از واژگان (OOV)
  - همه کلماتی که در واژگان نیستند را به عنوان یک واحد (مثلاً با نام UNK) در نظر می‌گیریم
  - تمام کلمات پیکره متنی که واژگان نیستند را با کلمه UNK جایگزین کرده و احتمال‌های آن را مانند سایر کلمات حساب می‌کنیم

  - در زمان استفاده، در صورت برخود با کلمات خارج از واژگان از همان احتمال‌های بدست آمده برای UNK استفاده می‌کنیم

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- تعداد احتمال‌های مدل n-gram با افزایش n به طور نمایی رشد می‌کند
  - تعداد احتمال‌های مدل bigram: V^2 (V=تعداد کل کلمات واژگان)
  - تعداد احتمال‌های مدل trigram: V^3
  - تعداد احتمال‌های مدل n-gram: V^n
  - معمولاً V از مرتبهٔ چندصد هزار (چند میلیون) کلمه است
    - بنابراین تعداد احتمال‌های مدل n-gram بسیار زیاد است
- در عمل بسیاری از احتمال‌های صفر هستند (عدم رخداد برخی دنباله کلمات)
  - به دلیل کم بودن حجم پیکره متنی
  - به دلیل مجاز نبودن بعضی از دنباله‌های کلمات در زبان
  - ماتریس احتمالات به صورت یک ماتریس تنک(sparse) است
    - ماتریس دارای عناصر صفر با تعداد زیاد است

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- **مثال: پیکره شکسپیر**
  - تعداد 844,674 واحد و تعداد 29,066 کلمه یکتا
  - تعداد کل دوتایی (بایگرم)های ممکن = ۲۹۰۶۶*۲۹۰۶۶ = ۸۴۴ میلیون
  - تعداد کل دوتایی (بایگرم)های مورد استفاده توسط شکسپیر(از ۸۴۴ میلیون) = ۳۰۰ هزار
    - حدود ۹۹.۹۶٪ بایگرم‌های ممکن در پیکره شکسپیر وجود ندارد = مقدار احتمال آنها صفر است

- **مثال: پیکره زبان کردی (ئاسوسافت)**
  - اندازه پیکره: 207M (۲۰۷ میلیون)
  - تعداد کلمه یکتا: 4.66M (۴.۶۶ میلیون)
  - در واژگان 100K (پرکاربرد)
    - تعداد دوتایی‌ها: 5.463M (از ۱۰ میلیارد ممکن) ◀ حدود ۹۹.۹٤٪ حالت موجود نیست
      - با Cutoff=2
    - تعداد سه تایی‌ها: 9.9M (از $۱۰^{۱۵}$ حالت ممکن) ◀ حدود ۹۹.۹۹٪ حالت موجود نیست
      - Cutoff=2 2

Hadi Veisi (h.veisi@ut.ac.ir)

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- مدل N-Gram برای داده‌های آموزش خوب عمل می‌کند
  - فقط زمانی برای داده‌های آزمون خوب عمل می‌کند که داده آموزش و آزمون مشابه باشند
    - که در عمل این گونه نیست!
- قابلیت تعمیم (Generalization)
  - مدل برای داده دیده نشده (آزمون) هم خوب عمل کند
  - بهبود قابلیت تعمیم با رفع مشکل صفرها
    - مواردی که در داده آموزش وجود ندارد ولی در داده آزمون وجود دارند

مدل‌سازی زبانی آماری: نکات کاربردی . . .

- مثال

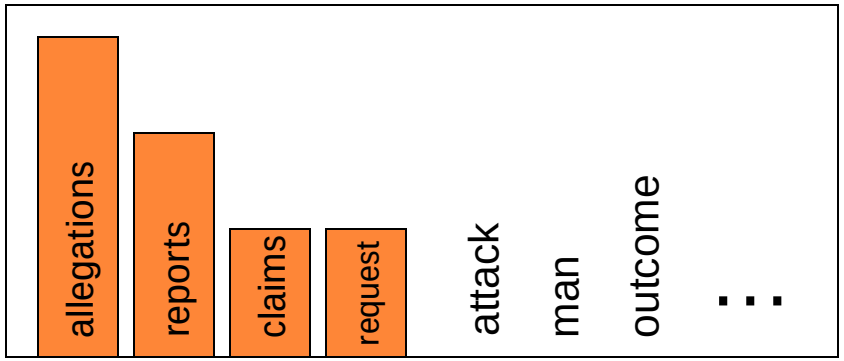
- **Training set**
  - … denied the allegations
  - … denied the reports
  - … denied the claims
  - … denied the request

- **Test set**
  - … denied the offer
  - … denied the loan

P(“offer” | denied the) = 0

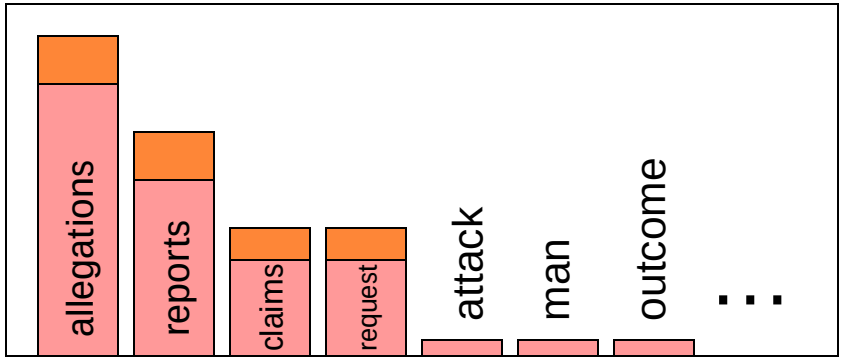
مدل‌سازی زبانی آماری: هموارسازی (Smoothing) . . .

- رفع مشکل احتمال‌های صفر در n-gram: هموارسازی (smoothing)
  - تخمین احتمال رخدادهای دیده نشده
  - داده‌های موجود



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

  - تغییر آمارها
    - برای تخمین موارد دیده نشده



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

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Add-One (هموارسازی لاپلاس) . . .
  - ساده‌ترین روش هموارسازی
  - اضافه کردن عدد ۱ به تمام شمارش‌ها
    - موارد دیده نشده = ۱
  - برای دوتایی

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

تعداد کل کلمات

- مثال: Berkeley Restaurant Project

	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

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Add-One (هموارسازی لاپلاس) . . .

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

  - مثال: Berkeley Restaurant Project

	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

- شمارش جدید بعد از هموارسازی

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

	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

$$c_i^* = (c_i+1)\frac{N}{N+V}$$

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Add-One (هموارسازی لاپلاس) . . .
  - تغییر زیاد در آمارها

	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

قبل از هموارسازی

	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

بعد از هموارسازی

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Add-One توسعه یافته: به حالت kتایی (Add-k)

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

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

- روش Unigram prior

$$P_{\text{UnigramPrior}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + mP(w_i)}{c(w_{i-1}) + m}$$

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Add-One (هموارسازی لاپلاس)
  - روش مؤثری نیست چون تغییر زیادی در احتمالات غیرصفر می‌دهد
  - از این روش برای هموارسازی مدل N-Gram استفاده نمی‌شود
  - کاربرد
    - مواردی که تعداد صفرها خیلی زیاد نیست
    - دسته‌بندی متون
- روش‌های دیگر هموارسازی
  - تخفیف (Discounting)
    - روش Good-Turing
    - روش پرکاربرد: Kneser-Ney
  - برهم‌نهی (Interpolation)
  - عقب‌گرد (Backoff)
    - برای پیکره‌های بسیار بزرگ مانند وب از نوعی از این روش با نام Stupid backoff استفاده می‌شود

مدل‌سازی زبانی آماری: هموارسازی . . .

- هموارسازی تخفیف (Discounting)
  - استفاده از شمارش‌های دیده شده برای تخمین شمارش‌های دیده نشده
  - از شمارش‌های غیرصفر کاسته شده و بر روی شمارش‌های صفر توزیع می‌گردد

ضریب تخفیف

$$c^* = c.d_c$$

شمارش اولیه

- تعریف N_c: تعداد چیزهایی که c بار دیده شده‌اند

Sam I am I am Sam I do not eat

I	3
Sam	2
am	2
do	1
not	1
eat	1

$N_1 = 3$ تعداد کلماتی که یک بار تکرار شده‌اند: do, not, eat

$N_2 = 2$

$N_3 = 1$

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش تخفیف Good-Turing . . .

تعداد کلماتی که یک بار تکرار شده‌اند

$$P^*_{GT}(\text{things with zero frequency}) = \frac{N_1}{N} \Rightarrow c^* = \frac{(c+1)N_{c+1}}{N_c}$$

- مثال: یک ماهی‌گیر ۱۸ ماهی زیر را صید کرده است (در متنی کلمات با فراوانی زیر تکرار شده)
  - 10 carp, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel

دیده شده (ماهی trout)

- c = 1
- MLE p=1/18
- C*(trout) = 2 * N2/N1= 2 * 1/3=2/3
- P^*_{GT}(trout) = 2/3 /18= 1/27

دیده نشده (ماهی bass یا catfish)

- c = 0
- MLE p=0/18=0
- P^*_{GT} (unseen) = N_1/N = 3/18

- عدم استفاده به تنهایی و استفاده به صورت ترکیبی با روش‌های برهم‌نهی و Backoff

مدل‌سازی زبانی آماری: هموارسازی . . .

روش تخفیف Good-Turing

- توزیع شمارش‌های کسرشده برروی شمارش‌های صفر

c (MLE)	N_c	c^* (GT)
0	74,671,100,000	0.0000270
1	2,018,046	0.446
2	449,721	1.26
3	188,933	2.24
4	105,668	3.24
5	68,379	4.22
6	48,190	5.19

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

- در عمل

نمونه مقدار برای k =۵

$$\begin{cases} c^* = c \quad \text{for } c > k \\ c^* = \dfrac{(c+1)\frac{N_{c+1}}{N_c} - c\frac{(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}}, \quad \text{for } 1 \le c \le k \end{cases}$$

مدل‌سازی زبانی آماری: هموارسازی . . .

روش تخفیف Good-Turing

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

- نتایج استفاده از روش Good-Turing روی ۲۲ میلیون کلمه

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

- نتایج نشان می‌دهد برای c>1 تقریبا داریم c*=c-0.75

- روش Absolute Discounting Interpolation
  - کم کردن مقدار مشخصی از شمارش (صرفه جویی در زمان)

Hadi Veisi (h.veisi@ut.ac.ir)

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Absolute Discounting Interpolation
  - جمع وزن‌دار bigram و unigram

مقدار ثابت، وابسته به پیکره نمونه مقدار = 0.75

وزن برهم‌نهی

$$P_{\text{AbsoluteDiscounting}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) - d}{c(w_{i-1})} + \lambda(w_{i-1})P(w_i)$$

دوتایی هموار شده با روش تخفیف

unigram

- برای تعداد ۱ و ۲ به صورت جداگانه مقادیر دیگری در نظر گرفته می‌شود
- استفاده از unigram استاندارد در رابطه فوق مناسب نیست. چرا؟
  - روش Kneser-Ney

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Kneser-Ney . . .
  - استفاده از unigram استاندارد در هموارسازی برهم‌نهی مناسب نیست
    - ممکن است unigram یک کلمه بالا باشد اما در بافت جمله مناسب نباشد
  - Shannon game: I can't see without my reading glasses | Francisco ?
  - "Francisco" is more common (higher unigram) than "glasses"
  - … but "Francisco" always follows "San"

- تعریف $P_{continuation}(w)$ به جای $P(w)$
  - بیانگر میزان پیوستگی است تا میزان اهمیت (تکرار) کلمه
  - تعداد بافت‌های مختلفی که کلمه w در آنها رخ می‌دهد

تعداد کلماتی که قبل از کلمه w می آیند

$$P_{CONTINUATION}(w) = \frac{\left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|}{\left|\{(w_{j-1}, w_j) : c(w_{j-1}, w_j) > 0\}\right|} = \frac{\left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|}{\sum_{w'} \left|\{w'_{i-1} : c(w'_{i-1}, w') > 0\}\right|}$$

تعداد کل bi-gramها (انواع bi-gramها نه تکرار آنها)

مدل‌سازی زبانی آماری: هموارسازی . . .

- روش Kneser-Ney . . .
  - محاسبه احتمال bigram با برهم‌نهی

مقدار ثابت، وابسته به پیکره نمونه مقدار = 0.75

$$P_{KN}(w_i \mid w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1}) P_{CONTINUATION}(w_i)$$

  - ضریب (وزن) برهم‌نهی

$$\lambda(w_{i-1}) = \frac{d}{c(w_{i-1})} \left| \{w : c(w_{i-1}, w) > 0\} \right|$$

مقدار تخفیف نرمال شده

تعداد کلماتی که بعد از کلمه w_{i-1} می آیند

روش پرکاربرد

مدل‌سازی زبانی آماری: هموارسازی . . .

روش Kneser-Ney

- حالت کلی (بازگشتی)
  - برای n=2 همان رابطه قبل

کلمه i-n+1 تا کلمه i-1

$$P_{KN}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\max(c_{KN}(w_{i-n+1}^{i}) - d, 0)}{c_{KN}(w_{i-n+1}^{i-1})} + \lambda(w_{i-n+1}^{i-1}) P_{KN}(w_i \mid w_{i-n+2}^{i-1})$$

$$c_{KN}(\bullet) = \begin{cases} count(\bullet) & \text{for the highest order} \\ continuationcount(\bullet) & \text{for lower order} \end{cases}$$

درحالت محاسبه سه تایی، برای خود سه تایی

یک تایی و دوتایی

- برای محاسبه یک تایی (پایان تکرار)

$$P_{KN}(w) = \frac{\max(c_{KN}(w) - d, 0)}{\sum_{w'} c_{KN}(w')} + \lambda(\varepsilon)\frac{1}{V}$$

رشته خالی

مدل‌سازی زبانی آماری: هموارسازی . . .

- هموارسازی برهم‌نهی (Interpolation)
  - ترکیب خطی n-gramهای پایین برای تخمین n-gram بالاتر
  - تخمین 3-gram بر اساس ترکیب آمار 1-gram، 2-gram و 3-gram

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

$$\sum_i \lambda_i = 1$$

  - محاسبه ضرایب λ با استفاده از داده
    - استفاده از داده Held-Out (یا Dev. Set)- مجزا از مجموعه داده آموزش
    - محاسبه N-gramها با داده آموزش و سپس تخمین λها
      - تخمین با استفاده از روش‌های تخمین مانند EM

مدل‌سازی زبانی آماری: هموارسازی . . .

هموارسازی عقب‌گرد (Backoff)

- زمانی که مقدار یک n-gram را نداشته باشیم (یا مقدار آن به اندازه کافی معتبر نباشد)، سراغ n-gramهای با درجهٔ پایین‌تر می‌رویم

- تخمین 3-gram با روش عقب‌گرد Katz

احتمال هموارشده با تخفیف

ضریب عقب‌گرد

$$P_{\text{katz}}(z|x,y) = \begin{cases} P^*(z|x,y), & \text{if } C(x,y,z) > 0 \\ \alpha(x,y)P_{\text{katz}}(z|y), & \text{else if } C(x,y) > 0 \\ P^*(z), & \text{otherwise.} \end{cases}$$

$$P_{\text{katz}}(z|y) = \begin{cases} P^*(z|y), & \text{if } C(y,z) > 0 \\ \alpha(y)P^*(z), & \text{otherwise.} \end{cases}$$

ارزیابی مدل‌های زبانی . . .

- آیا مدل زبانی محاسبه شده، زبان را به خوبی مدل می‌کند؟
  - احتمال بیشتری به جملات خوب می دهد (در مقابل جملات بد)
  - جملات خوب: جملات واقعی در زبان و دارای تکرار بالا
  - جملات بد: جملات غیر گرامری و با تکرار پایین

- ارزیابی
  - ایجاد مدل زبانی با مجموعه متنی به نام مجموعه آموزش (Train Set)
  - ارزیابی با مجموعه جملاتی متفاوت با مجموعه آموزش= مجموعه آزمون (Test Set)

  - نیاز به یک معیار ارزیابی (evaluation metric)
    - ارزیابی صحت مدل کردن جملات تست توسط مدل آموزش داده شده با مجموعه آموزش

ارزیابی مدل‌های زبانی . . .

- معیار ارزیابی
  - استفاده از کارایی سیستم پردازش زبان (بازشناسی گفتار، ترجمه، ...)
    - درصد تشخیص درست کلمات
  - مقایسه کارایی سیستم برای دو مدل زبانی مختلف و انتخاب مدل با کارایی بالاتر

- سرگشتگی (perplexity)
  - یک معیار مستقل از سیستم و متناسب با احتمال‌های نسبت داده شده به جملات
  - بیانگر اینکه در یک جمله، کلمه بعدی را چقدر دقیق پیش‌بینی کنیم

I always order pizza with cheese and ____

mushrooms 0.1
pepperoni 0.1
anchovies 0.01
….
fried rice 0.0001
….
and 1e-100

Shanon Game

ارزیابی مدل‌های زبانی: سرگشتگی . . .

تعریف

- مدلی بهتر است که درست کلمه بعدی را پیش‌بینی کند: احتمال بیشتری بدهد
- بیشینه کردن احتمال: $p(W) = P(w_1 w_2 \ldots w_N)$

دنباله کلمات جمله تست

- نرمال کردن به تعداد کلمات $p(W) = p(w_1 w_2 \ldots w_N)^{\frac{1}{N}}$
- سرگشتگی = معکوس احتمال (باید کمینه شود)

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

- با توجه به قاعده زنجیری احتمال $\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$
- برای دوتایی (بایگرام) داریم $\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_{i-1})}}$

ارزیابی مدل‌های زبانی: سرگشتگی . . .

- **مفهوم سرگشتگی**
  - **میانگین فاکتور انشعاب: میانگین تعداد کلمات ممکن بعد از هر (رشته) کلمه**

- **مثال**
  - **برای اعداد ۰ تا ۹: احتمال آمدن هر کدام برابر ۰.۱ است**
    - در یک رشته عددی N تایی داریم

$$pp(W) = p(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \left(\frac{1}{10}\cdot\frac{1}{10}\cdot\ldots\cdot\frac{1}{10}\right)^{-\frac{1}{N}} = \left(\left(\frac{1}{10}\right)^N\right)^{-\frac{1}{N}} = \left(\frac{1}{10}\right)^{-1} = 10$$

    - بعد از هر رشته عددی، هرکدام از ۱۰ عدد می‌توانند بیایند: سرگشتگی = ۱۰

  - **برای تشخیص اسامی در یک سامانه تشخیص گفتار با ۱۰۰۰ اسم**
    - سرگشتگی = ۱۰۰۰

ارزیابی مدل‌های زبانی: سرگشتگی . . .

- **مثال: سیستم تشخیص گفتار منشی تلفنی**
  - **سیستمی که یکی از کلمات «اپراتور»، «فروش»، «پشتیبانی» و «اسم یکی از افراد شرکت» را تشخیص می‌دهد و داخلی آن را وصل می‌کند.**
  - **آمارها**
    - کلمه «اپراتور»: به طور متوسط ۵۰٪ حالات
    - کلمه «فروش»: به طور متوسط ۲۰٪
    - کلمه «پشتیبانی»: به طور متوسط ۱۵٪
    - «اسامی افراد»: به طور متوسط ۱۵٪ حالات اسم یکی از ۱۰۰ نفر شرکت بیان می‌شود
  - **سرگشتگی؟**

$$pp(W) = p(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \left(0.5 \times 0.2 \times 0.15 \times \frac{0.15}{100}\right)^{-\frac{1}{4}} = 14.5$$

ارزیابی مدل‌های زبانی: سرگشتگی

- مدل زبانی بهتر (قوی‌تر)، سرگشتگی کمتری را نتیجه می‌دهد.
  - مدل‌های مختلف N-Gram روی پیکره WSJ (انگلیسی)
    - آموزش: ۳۸ میلیون کلمه – آزمون: ۱.۵ میلیون کلمه

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

  - مدل‌های مختلف N-Gram روی پیکره فارسی
    - آموزش: ۸۲۰ میلیون کلمه – آزمون: ۱۰ هزار کلمه (نامه اداری)

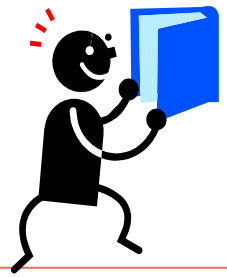
N-gram Order	Trigram
Perplexity	166

- معیار دیگر ارزیابی مدل‌های زبانی: آنتروپی (entropy)

$$H = \log_2 PP = \log_2 p(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = -\frac{1}{N}\log_2 p(w_1 w_2 \ldots w_N)$$

ابزارها . . .

- **SRI-LM**
  - http://www.speech.sri.com/projects/srilm
- **CMU Statistical Language Modeling (SLM)**
  - http://www.speech.cs.cmu.edu/SLM/toolkit.html
- **Google Book N-grams**
  - http://ngrams.googlelabs.com (https://books.google.com/ngrams)

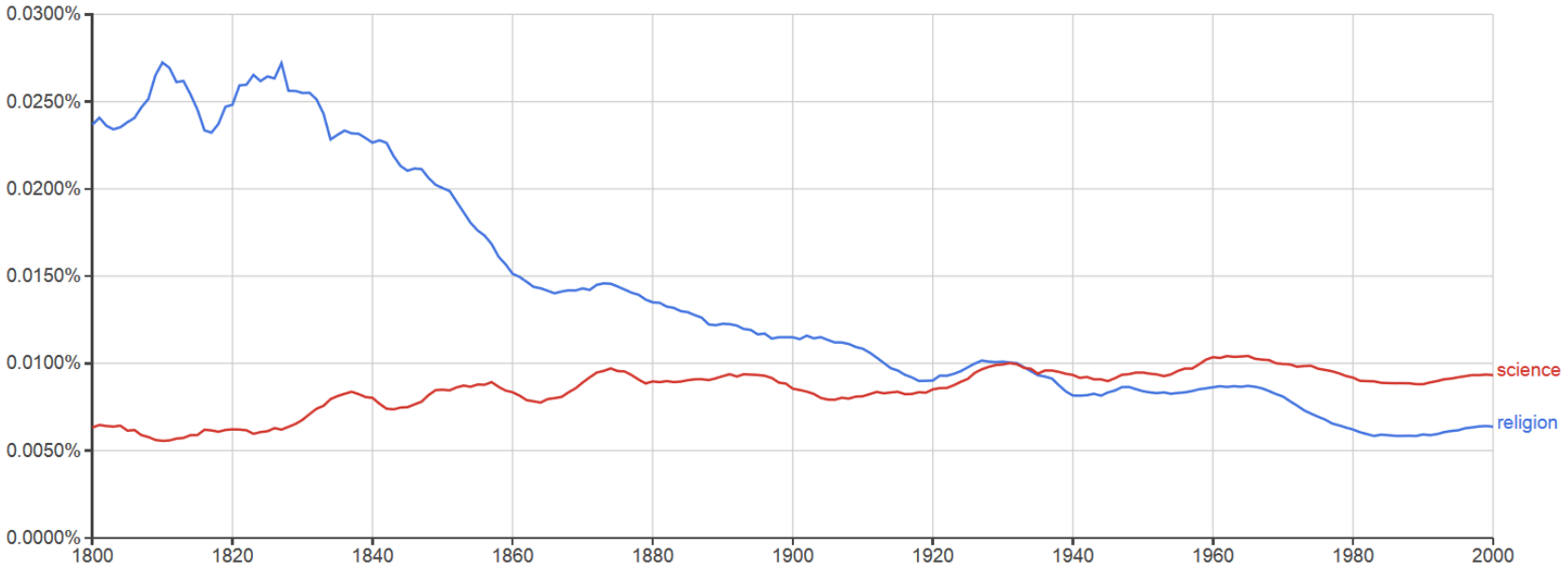


ابزارها: Google Book N-grams . . .

- استخراج شده از کتاب‌های کتابخانه گوگل (http://books.google.com)
  - از سال ۱۸۰۰ تا ۲۰۱۲
  - حاوی ۵ میلیون کتاب (برای ۸ زبان)
    - کتابخانه گوگل در ۲۰۱۳ حاوی ۳۰ میلیون کتاب بوده است (تخمین وجود ۱۳۰ میلیون کتاب در دنیا)
  - نسخه ۲ (در سال ۲۰۱۲)

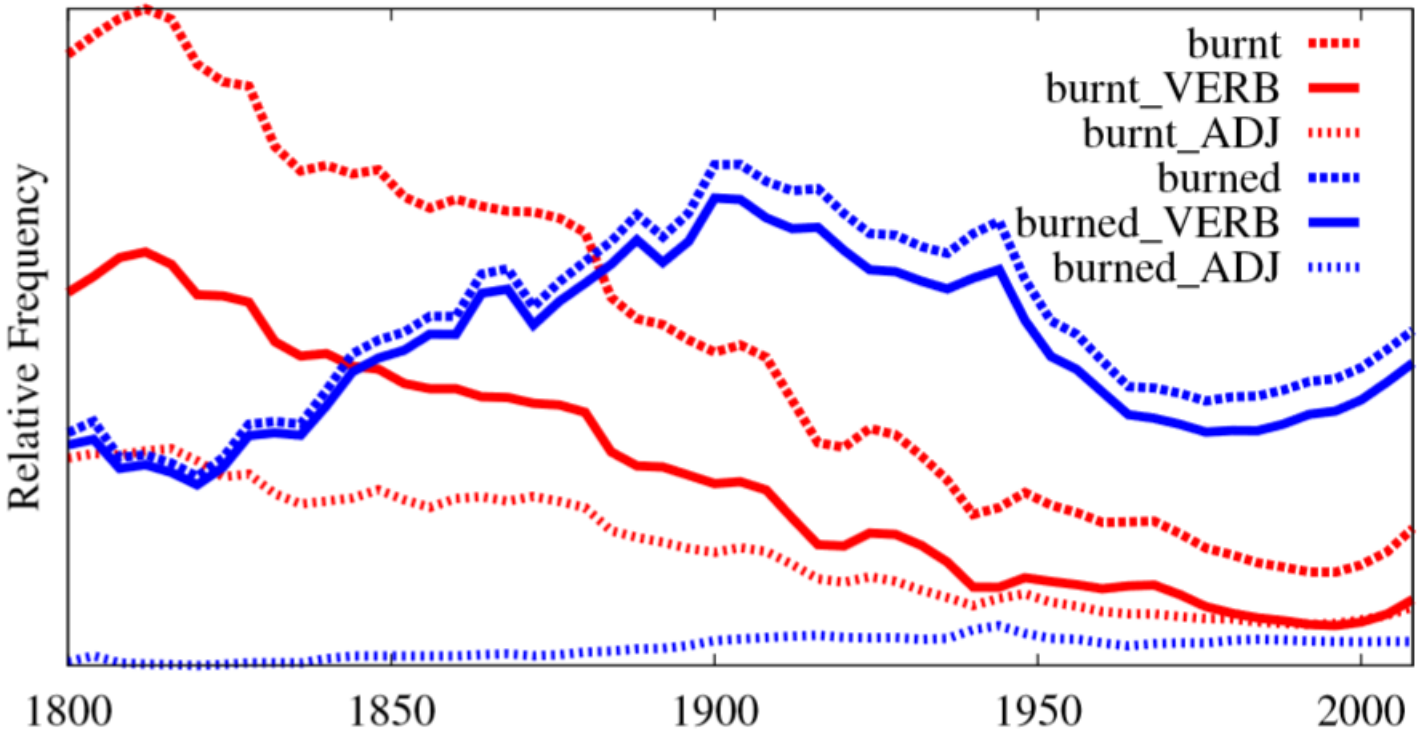
ابزارها: Google Book N-grams ...

- معرفی و دانلود داده‌ها
  - http://storage.googleapis.com/books/ngrams/books/datasetsv2.html
- بررسی تغییرات زبان و واژه‌ها از سال ۱۸۰۰ تا ۲۰۱۲



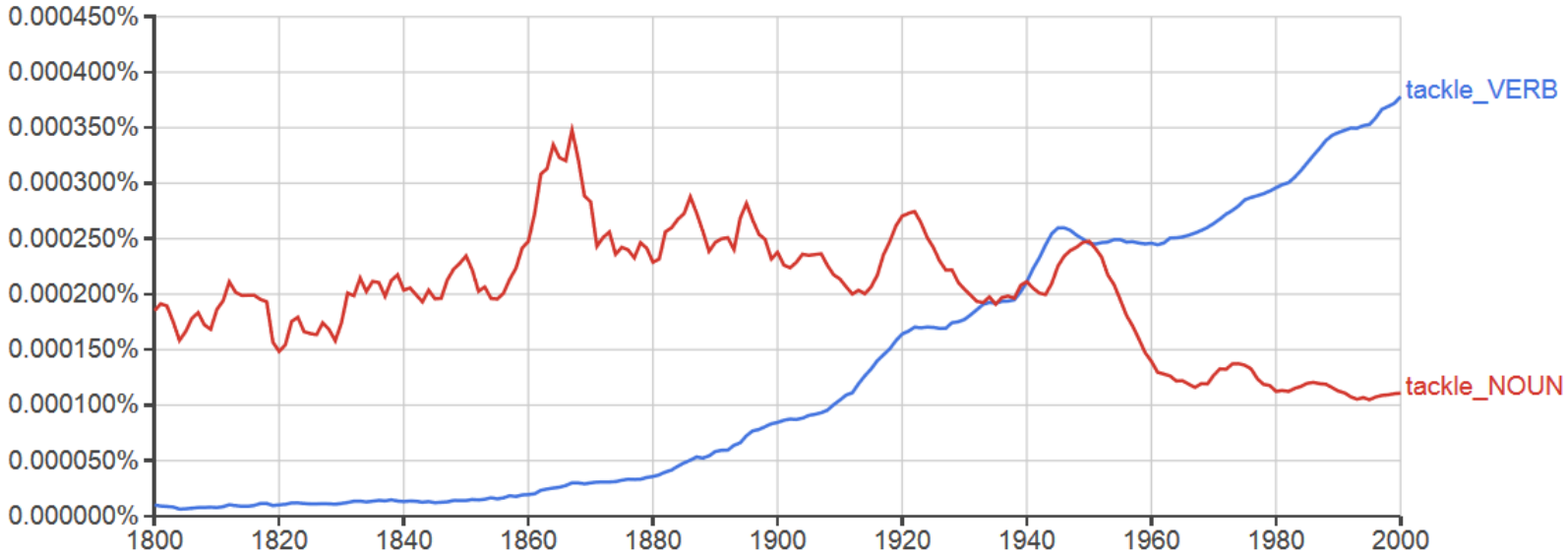
ابزارها: Google Book N-grams ...

- بررسی تغییرات زبان و واژه‌ها از سال ۱۸۰۰ تا ۲۰۱۲
  - استفاده از فعل بی قاعده burnt و با قاعده burned



ابزارها: Google Book N-grams ...

- بررسی تغییرات زبان و واژه‌ها از سال ۱۸۰۰ تا ۲۰۱۲
  - استفاده از کلمه tackle به عنوان اسم یا فعل



باز هم گوگل!

Web 1T 5-gram Version 1

- داده‌های N-Gram جمع آوری شده از حدود یک تریلیون کلمه (token) وب

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

```
File sizes: approx. 24 GB compressed (gzip'ed) text files

Number of tokens:    1,024,908,267,229
Number of sentences:    95,119,665,584
Number of unigrams:         13,588,391
Number of bigrams:         314,843,401
Number of trigrams:        977,069,902
Number of fourgrams:     1,313,818,354
Number of fivegrams:     1,176,470,663
```

نمونه 4-Gramها

باز هم گوگل!

Web 1T 5-gram Version 1

- https://catalog.ldc.upenn.edu/LDC2006T13

Web 1T 5-gram Version 1

Item Name:	Web 1T 5-gram Version 1
Author(s):	Thorsten Brants, Alex Franz
LDC Catalog No.:	LDC2006T13
ISBN:	1-58563-397-6
ISLRN:	831-344-220-094-6
Release Date:	September 19, 2006
Member Year(s):	2006
DCMI Type(s):	Text
Data Source(s):	web collection
Application(s):	language modeling
Language(s):	English
Language ID(s):	eng