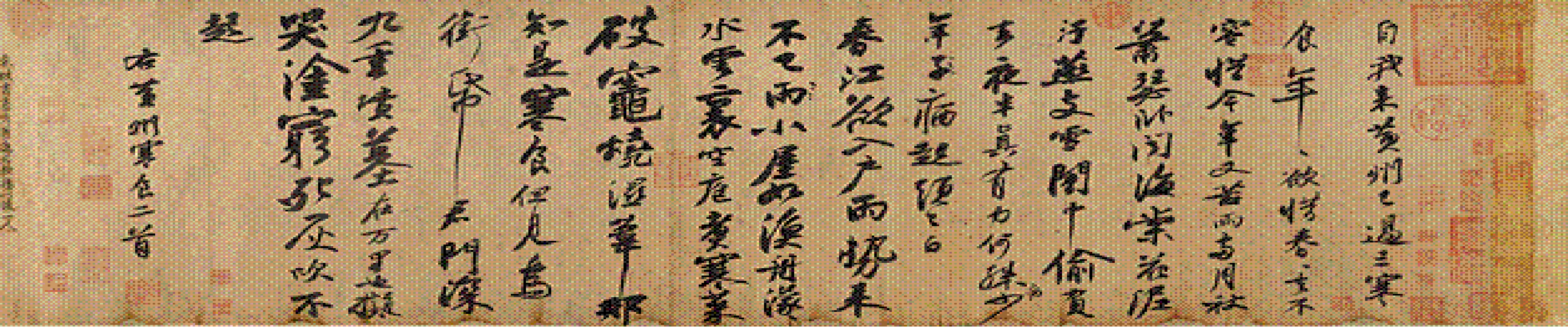

计算语言学
Computational Linguistics

教师：孙茂松

Tel:62781286

Email:sms@tsinghua.edu.cn

TA：林衍凯

Email:linyankai423@qq.com

郑重声明

- 此课件仅供选修清华大学计算机系研究生课《计算语言学》(70240052)的学生个人学习使用，所以只允许学生将其下载、存贮在自己的电脑中。未经孙茂松本人同意，任何人不得以任何方式扩散之（包括不得放到任何服务器上）。否则，由此可能引起的一切涉及知识产权的法律责任，概由该人负责。
- 此课件仅限孙茂松本人讲课使用。除孙茂松本人外，凡授课过程中，PTT文件显示此《郑重声明》之情形，即为侵权使用。

第四章
估计句子的概率-Markov模型
（Part 2）

4.4. Smoothing

- MLE: MLE is usually unsuitable for NLP because of the sparseness of the data
- p(z | xy) = ?
- Suppose our training data includes

  … xya ..

  … xyd …

  … xyd …

  but never xyz
- Should we conclude

  p(a | xy) = 1/3?

  p(d | xy) = 2/3?

  p(z | xy) = 0/3?
- NO! Absence of xyz might just be bad luck.

4.4. Smoothing

- Should we conclude

  $p(a \mid xy) = 1/3$? *reduce this*
  $p(d \mid xy) = 2/3$? *reduce this*
  $p(z \mid xy) = 0/3$? *increase this*

Baisc idea

- **Discount** the positive counts somewhat
- **Reallocate** that probability to the zeroes

4.4. Smoothing

N	Number of training instances
B	Number of bins training instances are divided into
w_{1n}	An n-gram $w_1 \cdots w_n$ in the training text
$C(w_1 \cdots w_n)$	Frequency of n-gram $w_1 \cdots w_n$ in training text
r	Frequency of an n-gram
$f(\cdot)$	Frequency estimate of a model
N_r	Number of bins that have r training instances in them
T_r	Total count of n-grams of frequency r in further data
h	'History' of preceding words

In *person*	*she*		*was*		inferior		*to*		*both*		*sisters*	
1-gram	$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$	
	the*	0.034	the	0.034	the	0.034	the	*0.034*	the	*0.034*	the	0.034
	to	0.032	to	0.032	to	0.032	to	*0.032*	to	*0.032*	to	0.032
	and	0.030	and	0.030	and	0.030			and	*0.030*	and	0.030
	of	0.029	of	0.029	of	0.029			of	0.029	of	0.029
8	was	0.015	was	0.015	was	0.015			was	0.015	was	0.015
13	she	0.011			she	0.011			she	0.011	she	0.011
254					both	0.0005			both	0.0005	both	0.0005
435					sisters	0.0003					sisters	0.0003
1701					inferior	0.00005						
2-gram	$P(\cdot\mid person)$		$P(\cdot\mid she)$		$P(\cdot\mid was)$		$P(\cdot\mid inferior)$		$P(\cdot\mid to)$		$P(\cdot\mid both)$	
1	and	0.099	had	0.141	not	0.065	to	*0.212*	be	0.111	of	0.066
2	who	0.099	was	0.122	a	0.052			the	0.057	to	0.041
3	to	0.076			the	0.033			her	0.048	in	0.038
4	in	0.045			to	0.031			have	0.027	and	0.025
23	she	0.009							Mrs	0.006	she	0.009
41									what	0.004	sisters	0.006
293									**both**	0.0004		
∞					inferior	0						
3-gram	$P(\cdot\mid In,person)$		$P(\cdot\mid person,she)$		$P(\cdot\mid she,was)$		$P(\cdot\mid was,inf.)$		$P(\cdot\mid inferior,to)$		$P(\cdot\mid to,both)$	
	UNSEEN		did	0.5	not	*0.057*	UNSEEN		the	*0.286*	to	*0.222*
2			was	0.5	very	*0.038*			Maria	0.143	Chapter	0.111
					in	0.030			cherries	0.143	Hour	0.111
4					to	0.026			her	0.143	Twice	0.111
∞					inferior	0			both	0	sisters	0
4-gram	$P(\cdot\mid u,I,p)$		$P(\cdot\mid I,p,s)$		$P(\cdot\mid p,s,w)$		$P(\cdot\mid s,w,i)$		$P(\cdot\mid w,i,t)$		$P(\cdot\mid i,t,b)$	
	UNSEEN		UNSEEN		in	1.0	UNSEEN		UNSEEN		UNSEEN	
∞					inferior	0						

4.4. Smoothing

- Laplace's law (*adding one*)

$$P_{Lap}(w_1 \ldots w_n) = \frac{C(w_1 \ldots w_n) + 1}{N + B}$$

$$f_{Lap}(w_1 \ldots w_n) = \frac{C(w_1 \ldots w_n) + 1}{N + B} * N$$

$$N_0 * P_{Lap}(C(w_1 \ldots w_n) = 0)$$

4.4. Smoothing

unsmoothed bigram counts: *2nd word*

1st word

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	***Total (N)***
I	8	1087	0	13	0	0	0		3437
want	3	0	786	0	6	8	6		1215
to	3	0	10	860	3	0	12		3256
eat	0	0	2	0	19	2	52		938
Chinese	2	0	0	0	0	120	1		213
food	19	0	17	0	0	0	0		1506
lunch	4	0	0	0	0	1	0		459
…									

unsmoothed normalized bigram probabilities:

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	***Total***
I	.0023 (8/3437)	.32	0	.0038 (13/3437)	0	0	0		1
want	.0025	0	.65	0	.0049	.0066	.0049		1
to	.00092	0	.0031	.26	.00092	0	.0037		1
eat	0	0	.0021	0	.020	.0021	.055		1
Chinese	.0094	0	0	0	0	.56	.0047		1
food	.013	0	.011	0	0	0	0		1
lunch	.0087	0	0	0	0	.0022	0		1
…									

4.4. Smoothing

add-one smoothed bigram counts:

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total (N+V)**
I	~~8~~ 9	~~1087~~ 1088	1	14	1	1	1		~~3437~~ 5053
want	~~3~~ 4	1	787	1	7	9	7		2831
to	4	1	11	861	4	1	13		4872
eat	1	1	23	1	20	3	53		2554
Chinese	3	1	1	1	1	121	2		1829
food	20	1	18	1	1	1	1		3122
lunch	5	1	1	1	1	2	1		2075

add-one normalized bigram probabilities:

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total**
I	.0018 (9/5053)	.22	.0002	.0028 (14/5053)	.0002	.0002	.0002		1
want	.0014	.00035	.28	.00035	.0025	.0032	.0025		1
to	.00082	.00021	.0023	.18	.00082	.00021	.0027		1
eat	.00039	.00039	.0012	.00039	.0078	.0012	.021		1
Chinese	.0016	.00055	.00055	.00055	.00055	.066	.0011		1
food	.0064	.00032	.0058	.00032	.00032	.00032	.00032		1
lunch	.0024	.00048	.00048	.00048	.00048	.0022	.00048		1

4.4. Smoothing

* Data from the AP from (Church and Gale, 1991): AP data, 44 million words
    * Corpus of 22,000,000 bigrams
    * Vocabulary of 273,266 words (i.e. 74,674,306,760 possible bigrams - or bins)
    * 74,671,100,000 bigrams were unseen
    * And each unseen bigram was given a frequency of 0.000295

Freq. from training data

Freq. from held-out data

f_{MLE}	$f_{empirical}$	$f_{add\text{-}one}$
0	0.000027	**0.000295**
1	0.448	0.000589
2	1.25	0.000884
3	2.24	0.00118
4	3.23	0.00147
5	4.21	0.00177

Add-one smoothed freq.

too high

too low

* Total probability mass given to unseen bigrams =
  (74,671,100,000 x 0.000295) / 22,000,000 = **0.9997** !!!!

4.4. Smoothing

- Lidstone's law and the Jeffreys-Perks law (*adding* λ)

$$P_{Lid}(w_1 \ldots w_n) = \frac{C(w_1 \ldots w_n) + \lambda}{N + B\lambda}$$

$$\mu = \frac{N}{N + B\lambda}$$

$$P_{Lid}(w_1 \ldots w_n) = \mu \frac{C(w_1 \ldots w_n)}{N} + (1 - \mu)\frac{1}{B}$$

Jeffreys-Perks law： $\lambda = 0.5$

Expected Likelihood Estimation

$\lambda = 1 \rightarrow$ Laplace

$\lambda = 0.5 \rightarrow$ Jeffreys-Perks

$\lambda = 0 \rightarrow$ MLE

- *She was inferior to both sisters.*

- Unigram: p(S)=3.96*e-17
- Bigram: p(S)=0;
- ELE（Bigram）: p(S)=6.89*e-20

Poor estimates of context are worse than none.

Rank	Word	MLE	ELE
1	not	0.065	0.036
2	a	0.052	0.030
3	the	0.033	0.019
4	to	0.031	0.017
...			
=1482	inferior	0	0.00003

4.4. Smoothing

- Held out estimation: How do we know how much of the probability space to "hold out" for unseen events?

$C_1(w_1 \ldots w_n)$ = frequency of $w_1 \ldots w_n$ in the training data

$C_2(w_1 \ldots w_n)$ = frequency of $w_1 \ldots w_n$ in the held out data

N_r is the number of n-grams with frequency *r* in the training data

Let T_r is the total number of times that all n-grams that appeared *r* times in the training data appeared in the held out data. Then:

$$P_{ho}(w_1 \ldots w_n) = \frac{T_r}{N_r N}$$

4.4. Smoothing

training data vs. held out data (validation data)

development test data vs. final test data

- Training:
  - Training data (80% of total data)
    - To build initial estimates (frequency counts)
  - Held out data (10% of total data)
    - To refine initial estimates (smoothed estimates)
- Testing:
  - Development test data (5% of total data)
    - To test while developing
  - Final test data (5% of total data)
    - To test at the end

	System 1	System 2
scores	71, 61, 55, 60, 68, 49, 42, 72, 76, 55, 64	42, 55, 75, 45, 54, 5 55, 36, 58, 55, 67
total	609	526
n	11	11
mean $\bar{x}_i$	55.4	47.8
$s_i^2 = \sum(x_{ij} - \bar{x}_i)^2$	1,375.4	1,228.8
df	10	10

$$\text{Pooled } s^2 = \frac{1375.4 + 1228.8}{10 + 10} \approx 130.2$$

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{2s^2}{n}}} = \frac{55.4 - 47.8}{\sqrt{\frac{2 \cdot 130.2}{11}}} \approx 1.56$$

t-test

alpha=0.05

$$t = 1.56 < 1.725$$

4.4. Smoothing

- Cross-validation (deleted estimation)

  Held out estimation is useful if there is a lot of data available

  If not, we can use each part of the data both as training data and as held out data.

$$P_{ho}(w_1 \ldots w_n) = \frac{T_r^{01}}{N_r^0 N} \quad \text{or} \quad \frac{T_r^{10}}{N_r^1 N}$$

$$P_{del}(w_1 \ldots w_n) = \frac{T_r^{01} + T_r^{10}}{N(N_r^0 + N_r^1)}$$

4.4. Smoothing

- Good-Turing estimation

$$r^* = (r+1)\frac{E(N_{r+1})}{E(N_r)} \qquad P_{GT} = \frac{r^*}{N}$$

Two strategies:

1. only for very low frequencies
2. (r, Nr) fit S

$$\text{If } C(w_1 \ldots w_n) = r > 0$$

$$P_{GT}(w_1 \ldots w_n) = \frac{r^*}{N} \qquad \text{where } r^* = (r+1)\frac{S(r+1)}{S(r)}$$

$$\text{else} \qquad P_{GT}(w_1 \ldots w_n) = \frac{1 - \sum_{r=1}^{\infty} N_r \frac{r^*}{N}}{N_0} \approx \frac{N_1}{N_0 N}$$

4.4. Smoothing

Empirical results for bigram data (Church and Gale)

f	f_{emp}	f_{GT}	f_{add1}	f_{del}
0	0.000027	0.000027	0.000295	0.000037
1	0.448	0.446	0.000589	0.396
2	1.25	1.26	0.000884	1.24
3	2.24	2.24	0.00118	2.23
4	3.23	3.24	0.00147	3.22
5	4.21	4.22	0.00177	4.22
6	5.23	5.19	0.00206	5.20
7	6.21	6.21	0.00236	6.21
8	7.21	7.24	0.00265	7.18
9	8.26	8.25	0.00295	8.18

Bigrams				Trigrams			
r	N_r	r	N_r	r	N_r	r	N_r
1	138741	28	90	1	404211	28	35
2	25413	29	120	2	32514	29	32
3	10531	30	86	3	10056	30	25
4	5997	31	98	4	4780	31	18
5	3565	32	99	5	2491	32	19
6	2486		...	6	1571		..
7	1754	1264	1	7	1088	189	1
8	1342	1366	1	8	749	202	1
9	1106	1917	1	9	582	214	1
10	896	2233	1	10	432	366	1
	...	2507	1		...	378	1

Table 6.7 Extracts from the frequencies of frequencies distribution for bigrams and trigrams in the Austen corpus.

r	r^*	$P_{GT}(\cdot)$
0	0.0007	1.058×10^{-9}
1	0.3663	5.982 x10−7
2	1.228	2.004×10^{-6}
3	2.122	3.465×10^{-6}
4	3.058	4.993×10^{-6}
5	4.015	6.555×10^{-6}
6	4.984	8.138×10^{-6}
7	5.96	9.733×10^{-6}
8	6.942	1.134×10^{-5}
9	7.928	1.294×10^{-5}
10	8.916	1.456×10^{-5}
...		
28	26.84	4.383 x10−5
29	27.84	4.546×10^{-5}
30	28.84	4.709×10^{-5}
31	29.84	4.872×10^{-5}
32	30.84	5.035×10^{-5}
...		
1264	1263	0.002062
1366	1365	0.002228
1917	1916	0.003128
2233	2232	0.003644
2507	2506	0.004092

$$P(she|person) = \frac{f_{\mathrm{GT}}(person\ she)}{C(person)} = \frac{1.228}{223} = 0.0055$$

$P(she\|person)$	0.0055
$P(was\|she)$	0.1217
$P(inferior\|was)$	6.9×10^{-8}
$P(to\|inferior)$	0.1806
$P(both\|to)$	0.0003956
$P(sisters\|both)$	0.003874

4.4. Smoothing

- Katz's backing-off (1987)
  - Why are we treating all novel events as the same?
  - p(zygote | see the) vs. p(baby | see the)
    - Suppose both trigrams have zero count
  - baby beats zygote as a unigram
  - the baby beats the zygote as a bigram
  - see the baby beats see the zygote ?

4.4. Smoothing

Basic smoothing (e.g., add-λ or Good-Turing):

* Holds out some probability mass for novel events
* Divided up **evenly** among the novel events

Backoff smoothing

* Holds out same amount of probability mass for novel events
* But divide up **unevenly** in proportion to backoff prob.
* For p(z | xy):
  * Novel events are types z that were never observed after xy
  * Backoff prob for p(z| xy) is p(z| y) … which in turn backs off to p(z)!

4.4. Smoothing

- In back-off models, different models are consulted in order depending on their specificity.
- If the n-gram of concern has appeared more than k times, then an n-gram estimate is used but an amount of the MLE estimate gets discounted (it is reserved for unseen n-grams).
- If the n-gram occurred k times or less, then we will use an estimate from a shorter n-gram (back-off probability), normalized by the amount of probability remaining and the amount of data covered by this estimate.
- The process continues recursively.

$$P_{bo}(w_i \mid w_{i-n+1} \ldots w_{i-1}) = \begin{cases} (1-d_{w_{i-n+1}\ldots w_{i-1}}) \dfrac{C(w_{i-n+1}\ldots w_i)}{C(w_{i-n+1}\ldots w_{i-1})} & \text{if } C(w_{i-n+1}\ldots w_i) > k \\ \alpha_{w_{i-n+1}\ldots w_{i-1}} P_{bo}(w_i \mid w_{i-n+2}\ldots w_{i-1}) & \text{Otherwise} \end{cases}$$

	$P(she\|h)$	$P(was\|h)$	$P(inferior\|h)$	$P(to\|h)$	$P(both\|h)$	$P(sisters\|h)$	Product
Unigram	0.011	0.015	0.00005	0.032	0.0005	0.0003	3.96×10^{-17}
Bigram	0.00529	0.1219	0.0000159	0.183	0.000449	0.00372	3.14×10^{-15}
n used	2	2	1	2	2	2	
Trigram	0.00529	0.0741	0.0000162	0.183	0.000384	0.00323	1.44×10^{-15}
n used	2	3	1	2	2	2	

Table 6.11 Probability estimates of the test clause according to various language models. The unigram estimate is our previous MLE unigram estimate. The other two estimates are back-off language models. The last column gives the overall probability estimate given to the clause by the model.

4.4. Smoothing

- Simple linear interpolation (*deleted interpolation*)

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

where $0 \le \lambda_i \le 1$ and $\Sigma_i \lambda_i = 1$

- General linear interpolation
- EM（Expectation maximization

"Whenever data sparsity is an issue, smoothing can help performance, and data sparsity is almost always an issue in statistical modeling. In the extreme case where there is so much training data that all parameters can be accurately trained without smoothing, one can almost always expand the model, such as by moving to a higher n-gram model, to achieve improved performance. With more parameters data sparsity becomes an issue again, but with proper smoothing the models are usually more accurate than the original models. Thus, no matter how much data one has, smoothing can almost always help performace, and for a relatively small effort."

Chen & Goodman (1998)

4.4. Smoothing

- Chapter 6: Statistical Inference: n-gram Models over Sparse Data, Foundations of Statistical NLP

Stanley F. Chen, Joshua Goodman (1996, Harvard University), *An Empirical Study of Smoothing Techniques for Language Modeling*, Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics

Church, K., and Gale, W. (1990) , *Poor Estimates of Context are Worse than None,* Third Darpa Workshop on Speech and Natural Language, Hidden Valley, PA.

K. Church and W. Gale. (1991). *A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams*. Computer Speech and Language 5:19-54.