

Two Empirical Methods for Measuring Adaptation

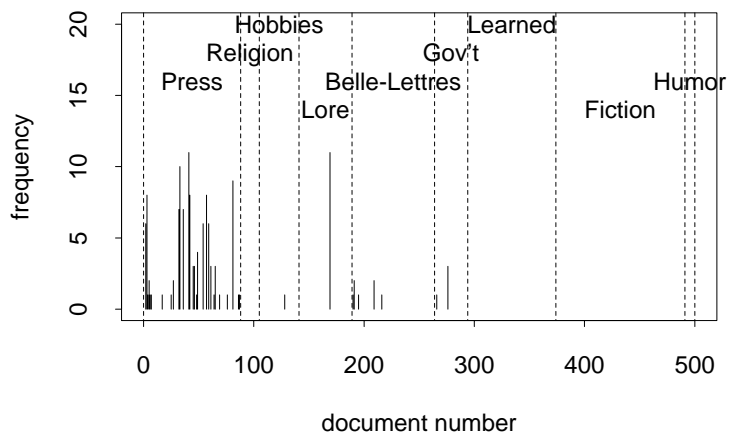
Kenneth Ward Church
AT&T Labs – Research
Florham Park, NJ
kwc@research.att.com

- Repetition is very common; ngrams/words (and their variant forms) appear in bursts.
- Adaptive language models (from Speech Recognition):
 - Cache Model(s): Jelinek (1997, p. 254)
 - Additive: $Pr(w) = \lambda Pr_L(w) + (1 - \lambda) Pr_G(w)$
 - Case-based: $Pr(w) = \begin{cases} \lambda_1 Pr_L(w) & \text{if } w \in \text{cache} \\ \lambda_2 Pr_G(w) & \text{otherwise} \end{cases}$
 - Introduced to account for topic change & repetition
 - Topic change: train on one domain (news); test on another (medicine).
 - Repetition: if document mentions *Noriega* once, it may well mention him again.

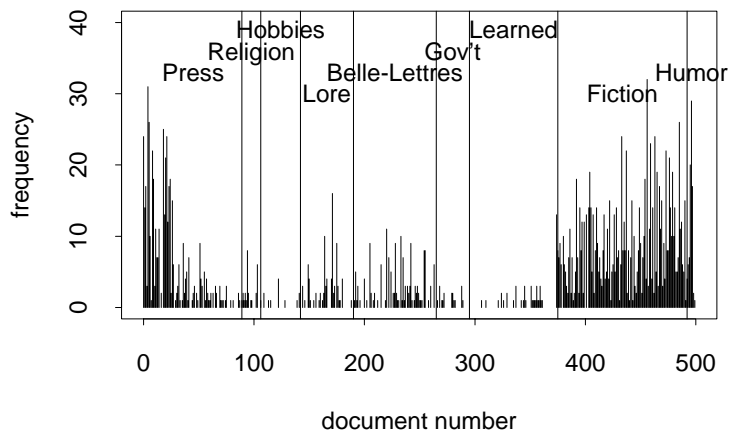
Adaptation

- Intuition: if a word has been mentioned recently
 - then the probability of that word (and its variant forms/neighbors) should go way up,
 - and many other words should go down a little bit.
- Adaptation Pattern:
 $Pr(+adapt) \gg Pr(prior) > Pr(-adapt)$
- Adaptation pattern is lexical
 - Stronger for content words:
 - proper nouns,
 - technical terminology and
 - good keywords for information retrieval.
 - Weaker for:
 - function words,
 - cliches and
 - common first names.

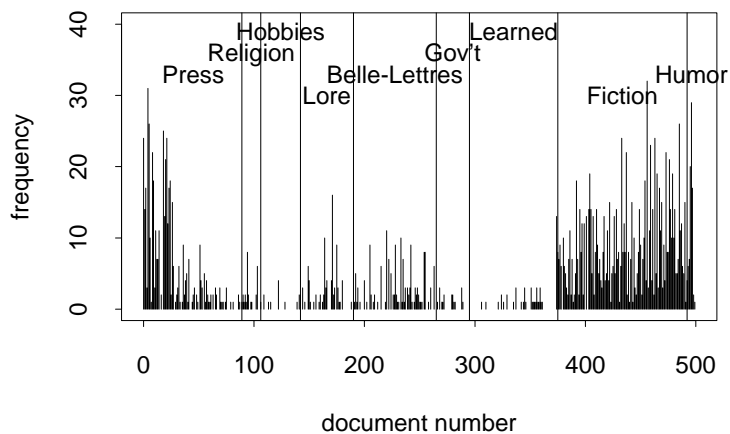
“Kennedy” in Brown Corpus



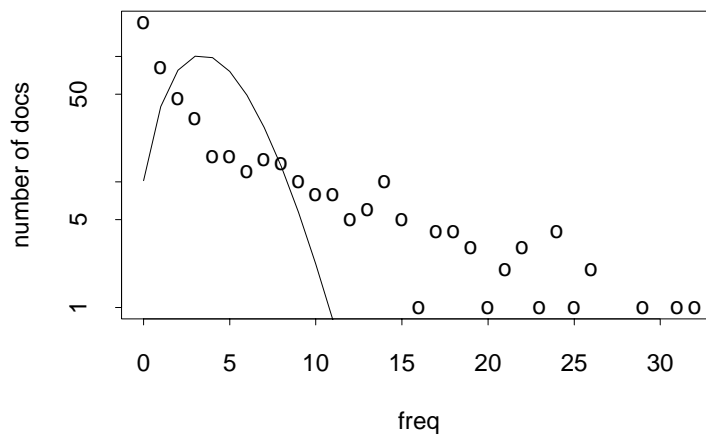
“said” in Brown Corpus



“said” in Brown Corpus



Poisson Doesn't Fit



Florian & Yarowsky (ACL-99)

- Observation: “peace” is more likely in some genres (International News) and less likely in others (Sports).

- Example: $\frac{Pr(x = \text{peace} \mid \text{context})}{Pr(x = \text{piece} \mid \text{context})}$

Context: “It is at least on the Serb side a real setback to the x”

- Florian & Yarowsky’s proposal:
 - Cluster documents into about 10^2 topics
 - Compute a language model for each topic
- Our suggestion:
 - Replace clustering with k nearest neighbors
 - 10^2 topics $\rightarrow \begin{bmatrix} D \\ k \end{bmatrix} \approx \begin{bmatrix} 10^5 \\ 10 \end{bmatrix}$ topics
 - Replace implicit independence (Poisson) assumptions with a non-parametric model

Non-Parametric Estimates of Adaptation: Method 1

- Split each document into two equal pieces:
 - history: first half of each document
 - test: second half of each document
- Task: given first half of a document (history), predict second half (test).
- Compute a contingency table for each word

Documents containing “hostages”
in 1990 AP News

	test	<u>test</u>
<u>history</u>	638	505
<u>history</u>	557	76787

<i>history</i>	test	
	<i>a</i>	<i>b</i>
	<i>c</i>	<i>d</i>

number of documents = $D = a + b + c + d$

document frequency = $df = a + b + c$

$$Pr(w \in test) \approx \frac{a + c}{D} \quad \text{prior}$$

$$Pr(w \in test | w \in history) \approx \frac{a}{a + b} \quad \text{+adapt}$$

$$Pr(w \in test | \neg w \in history) \approx \frac{c}{c + d} \quad \text{-adapt}$$

Pr(+adapt) >> Pr(prior) > Pr(-adapt)				
prior	+adapt	-adapt	source	w
0.014	0.56	0.0069	AP87	hostages
0.015	0.56	0.0072	AP90	
0.013	0.59	0.0057	AP91	
0.0044	0.39	0.0030	AP93	

Adaptation is Lexical

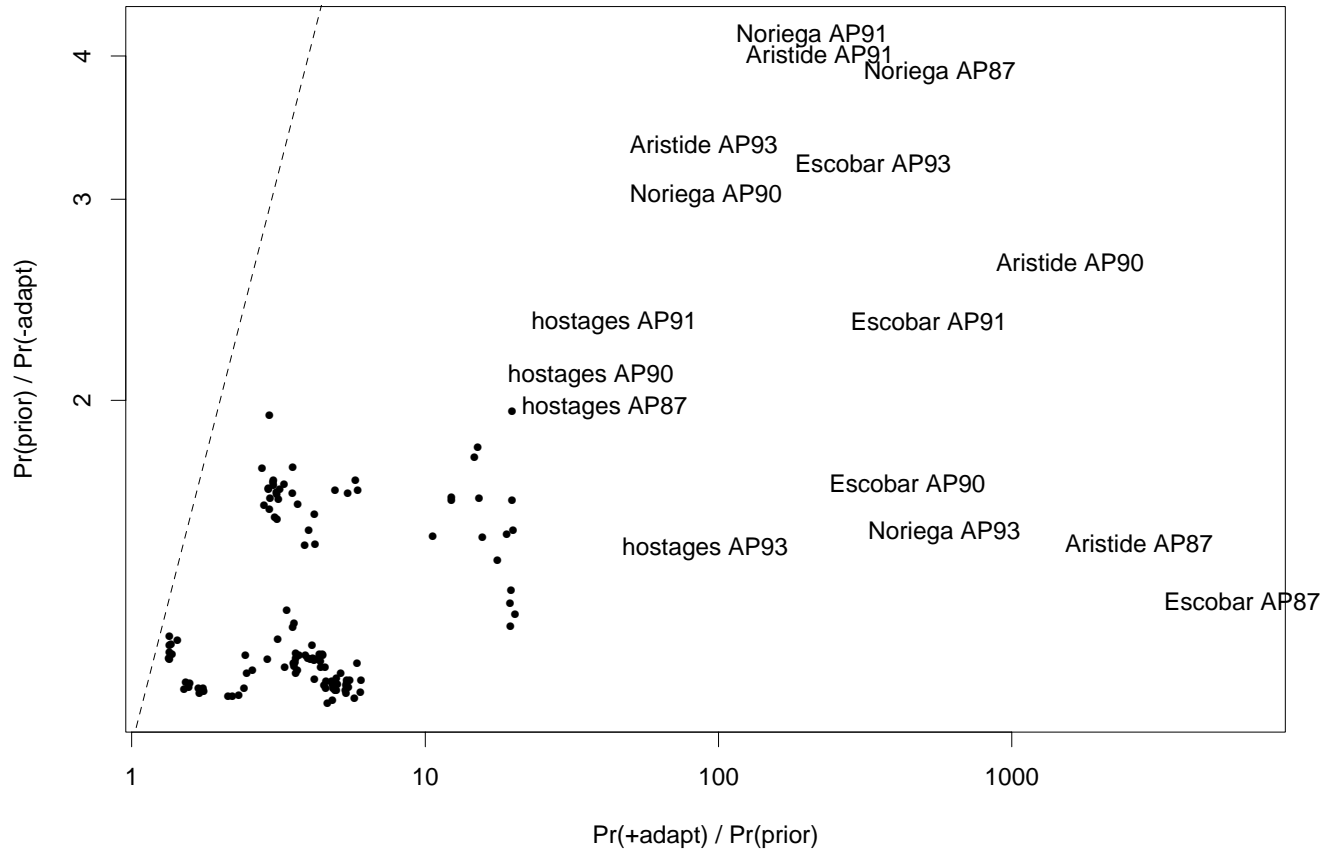
“Kennedy” adapts more than “except”

prior	+adapt	-adapt	source	w
0.012	0.27	0.0091	AP90	Kennedy
0.015	0.40	0.0084	AP91	Kennedy
0.014	0.32	0.0094	AP93	Kennedy
0.016	0.049	0.016	AP90	except
0.014	0.047	0.014	AP91	except
0.012	0.048	0.012	AP93	except

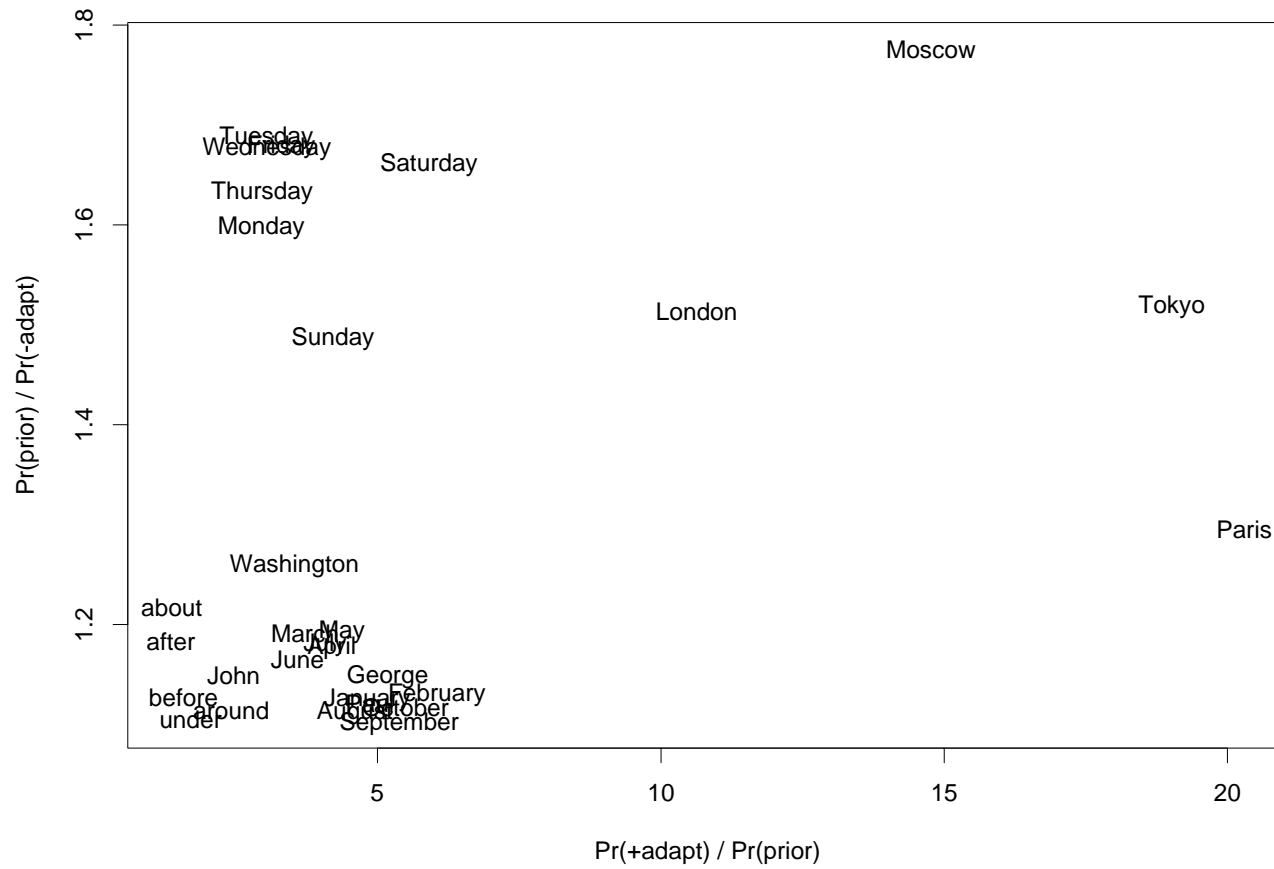
“said” adapts more than “for” & “and”

prior	+adapt	-adapt	source	w
0.81	0.90	0.44	AP90	said
0.81	0.89	0.44	AP91	said
0.76	0.88	0.39	AP93	said
0.78	0.84	0.58	AP90	for
0.77	0.83	0.58	AP91	for
0.74	0.82	0.54	AP93	for
0.94	0.96	0.66	AP90	and
0.93	0.95	0.70	AP91	and
0.91	0.95	0.60	AP93	and

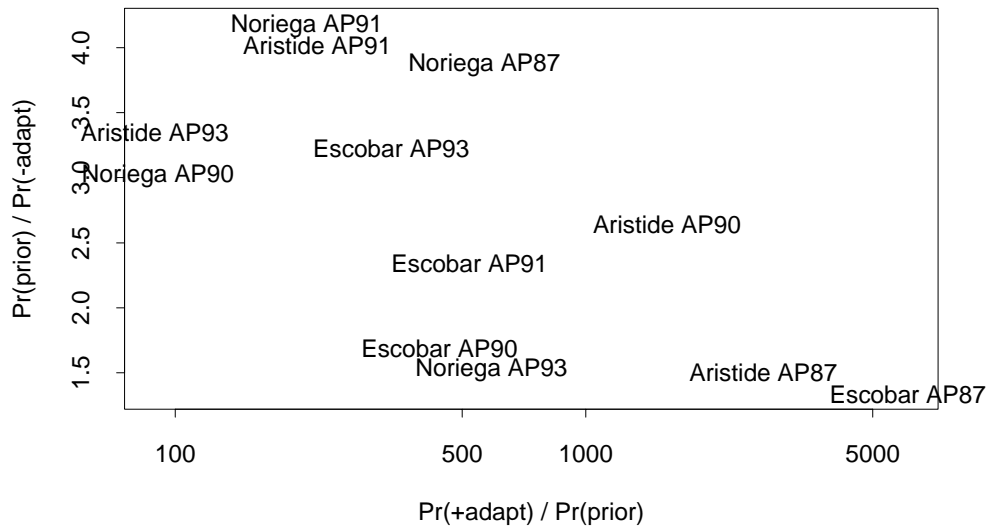
Adaptation Pattern: $\text{Pr}(+\text{adapt}) \gg \text{Pr}(\text{prior}) > \text{Pr}(-\text{adapt})$



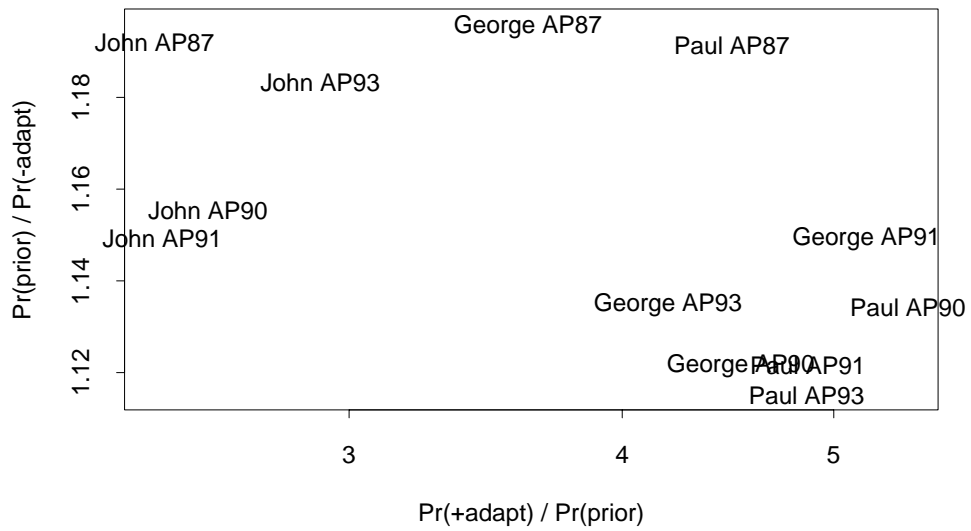
Content words adapt more than function words



Good Keywords



Poor Keywords



Degree of Adaptation Depends on Semantic Class

Pr(+adapt) >> Pr(prior) > Pr(-adapt)				
prior	+adapt	-adapt	source	w
0.091	0.31	0.070	AP87	Washington
0.083	0.30	0.066	AP90	
0.078	0.28	0.062	AP91	
0.086	0.27	0.080	AP93	
0.030	0.37	0.018	AP87	London
0.033	0.41	0.020	AP90	
0.035	0.37	0.023	AP91	
0.023	0.35	0.014	AP93	
0.024	0.47	0.012	AP87	Moscow
0.031	0.47	0.017	AP90	
0.030	0.44	0.017	AP91	
0.019	0.37	0.012	AP93	
0.016	0.29	0.011	AP87	Tokyo
0.019	0.30	0.013	AP90	
0.015	0.28	0.0097	AP91	
0.019	0.37	0.011	AP93	
0.013	0.26	0.010	AP87	Paris
0.015	0.30	0.011	AP90	
0.012	0.25	0.0096	AP91	
0.013	0.25	0.010	AP93	

Adaptation pattern over days of week

Pr(+adapt) >> Pr(prior) > Pr(-adapt)				
prior	+adapt	-adapt	source	w
0.08	0.26	0.05	AP87	Monday
0.10	0.27	0.06	AP90	
0.07	0.25	0.04	AP87	Tuesday
0.09	0.26	0.05	AP90	
0.07	0.26	0.04	AP87	Wednesday
0.09	0.26	0.05	AP90	
0.10	0.28	0.06	AP87	Thursday
0.09	0.26	0.05	AP90	
0.10	0.30	0.05	AP87	Friday
0.09	0.29	0.06	AP90	
0.05	0.26	0.03	AP87	Saturday
0.05	0.27	0.03	AP90	
0.07	0.29	0.05	AP87	Sunday
0.07	0.30	0.04	AP90	

Adaptation pattern over months

Pr(+adapt) >> Pr(prior) > Pr(-adapt)				
prior	+adapt	-adapt	source	w
0.04	0.16	0.03	AP87	January
0.03	0.17	0.03	AP90	
0.03	0.16	0.03	AP91	
0.06	0.23	0.05	AP87	March
0.06	0.23	0.05	AP90	
0.06	0.22	0.05	AP91	
0.07	0.23	0.06	AP87	May
0.06	0.23	0.05	AP90	
0.05	0.22	0.04	AP91	
0.05	0.20	0.05	AP87	July
0.05	0.23	0.04	AP90	
0.05	0.21	0.04	AP91	
0.03	0.15	0.03	AP87	September
0.03	0.16	0.03	AP90	
0.02	0.13	0.02	AP91	

**Names: “Good” keywords adapt more than “poor”
keywords**

Pr(+adapt) >> Pr(prior) > Pr(-adapt)				
prior	+adapt	-adapt	source	w
0.0079	0.71	0.0026	AP90	Noriega
0.0038	0.80	0.0009	AP91	
0.0006	0.90	0.0002	AP90	Aristide
0.0035	0.77	0.0009	AP91	
0.0011	0.47	0.0006	AP90	Escobar
0.0014	0.74	0.0006	AP91	
0.068	0.18	0.059	AP90	John
0.066	0.16	0.057	AP91	
0.025	0.11	0.022	AP90	George
0.025	0.13	0.022	AP91	
0.029	0.15	0.025	AP90	Paul
0.028	0.13	0.025	AP91	

Function words don't adapt as much as content words.

Pr(+adapt) >> Pr(prior) > Pr(-adapt)				
prior	+adapt	-adapt	source	w
0.36	0.48	0.29	AP90	about
0.35	0.48	0.29	AP91	
0.32	0.47	0.26	AP93	
0.14	0.24	0.12	AP90	under
0.12	0.21	0.11	AP91	
0.12	0.21	0.11	AP93	
0.27	0.37	0.23	AP90	after
0.26	0.35	0.22	AP91	
0.26	0.36	0.22	AP93	
0.17	0.26	0.15	AP90	before
0.16	0.26	0.15	AP91	
0.16	0.25	0.14	AP93	
0.07	0.17	0.07	AP90	around
0.07	0.17	0.06	AP91	
0.07	0.16	0.07	AP93	

Smoothing (for low frequency words)

- Problem: a , b , c , d and ratios of these quantities become unstable when the counts are small.
- Solutions: Good-Turing & Deleted Interpolation
- Let r be an observed count of an object (e.g., the frequency of a word)
- Let r^* be our best estimate of r in another corpus of the same size (all other things being equal)

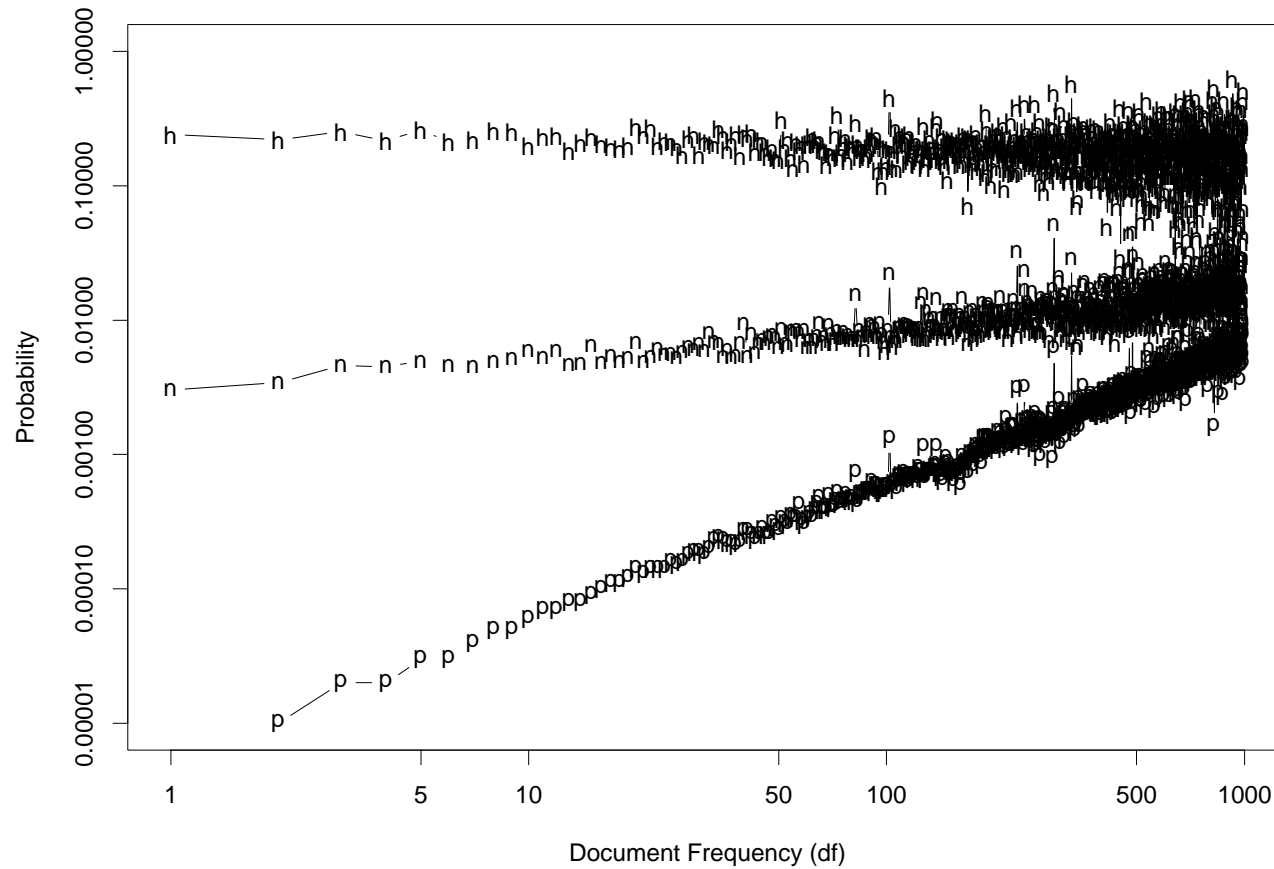
Simple Deleted Interpolation (for est freq of words/ngrams)

- Split training corpus in two halves.
- Use first half to count r for all objects (e.g., the frequency of all words in vocabulary)
- Use these counts to group objects into bins. The r^{th} bin contains all (and only) the words with freq r .
- Let N_r denote the size of the r^{th} bin.
- Use the second half of the training corpus to compute C_r , the aggregate frequency of all the words in the r^{th} bin.
- Set $r^* = \frac{C_r}{N_r}$
- If the two halves of the training corpora or the test corpora have different sizes, then scale r^* appropriately.

Application of Deleted Interpolation to Contingency Tables

- Split training corpus into two halves (we used two different years of AP news).
- Use first half to count df for all words in vocabulary (including words that didn't appear in the first half).
- Group words into bins by df .
- Let N_{df} denote the number of words in the df^{th} bin.
- Use the second half of the training corpus to compute $C_{df}^a, C_{df}^b, C_{df}^c, C_{df}^d$, the aggregate values of the contingency tables for all the words in the df^{th} bin.
- Let $a_{df}^* = \frac{C_{df}^a}{N_{df}}, b_{df}^* = \frac{C_{df}^b}{N_{df}}, c_{df}^* = \frac{C_{df}^c}{N_{df}}$, and $d_{df}^* = \frac{C_{df}^d}{N_{df}}$
- Compute $\text{Pr}(+adapt), \text{Pr}(prior), \text{Pr}(-adapt)$ as before, but replace a, b, c, d with a^*, b^*, c^*, d^* , respectively.

History (h) >> Neighborhood (n) >> Prior (p)



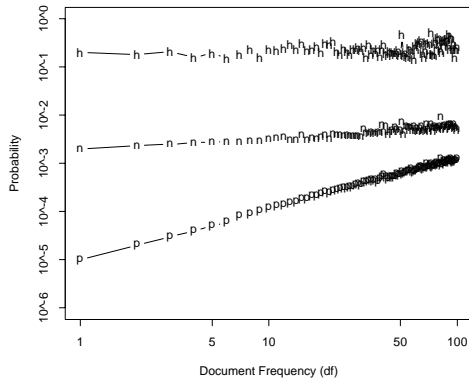
Concerns with Additive Cache Model

- Additive: $Pr(w) = \lambda Pr_L(w) + (1 - \lambda) Pr_G(w)$
- Adaptation is huge
 - $Pr(+adapt) \gg Pr(prior)$, often by 100-1000
 - Not clear how to build so much dynamic range into a small cache.
- $Pr(prior)$ depends strongly on frequency;
 $Pr(+adapt)$ does not.
- Adaptation is Lexical
 - Some words adapt more (Noriega, Aristide, Escobar)
 - Some words adapt less (John, George, Paul)
 - Words that adapt more in one year of AP news tend to adapt more in another year, and vice versa.
- Fix: allow λ to depend on w and whether or not $w \in cache$ (case-based cache model).

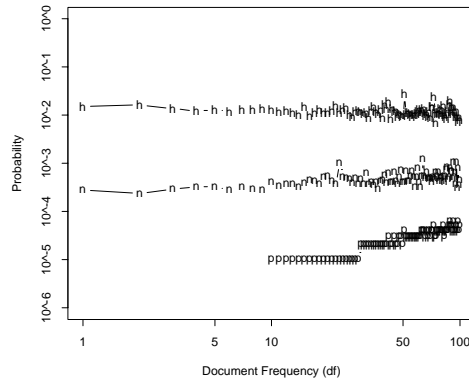
In a practical app, history won't be the same size as test

- Two approaches to variable length history
 - Approach 1: modify Method 1 to consider histories that are not the same size as test
 - Approach 2: introduce a second method for estimating $Pr(+adapt)$ that doesn't impose assumptions on the length of the history.

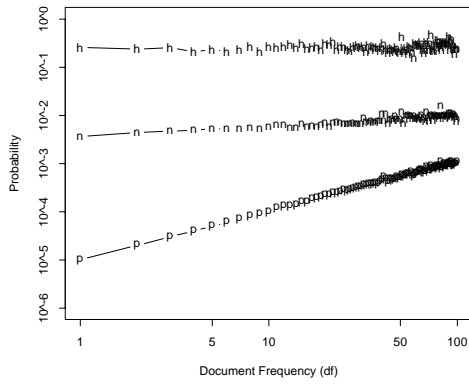
History = first 5%



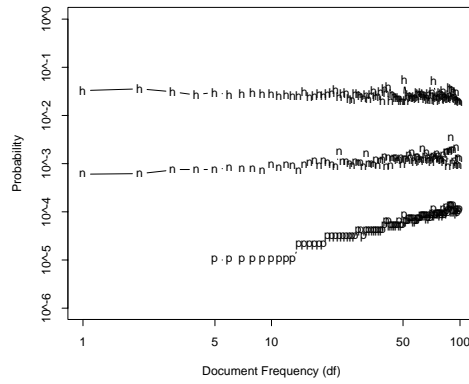
History = first 95%



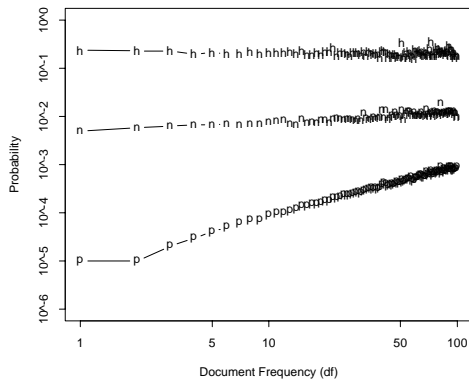
History = first 10%



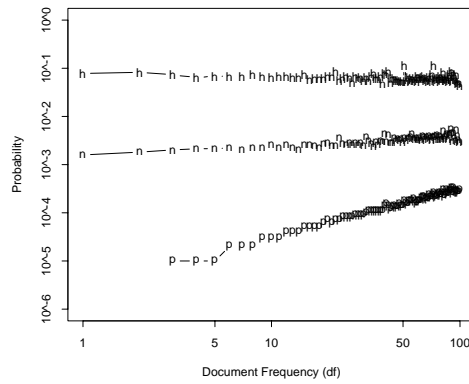
History = first 90%



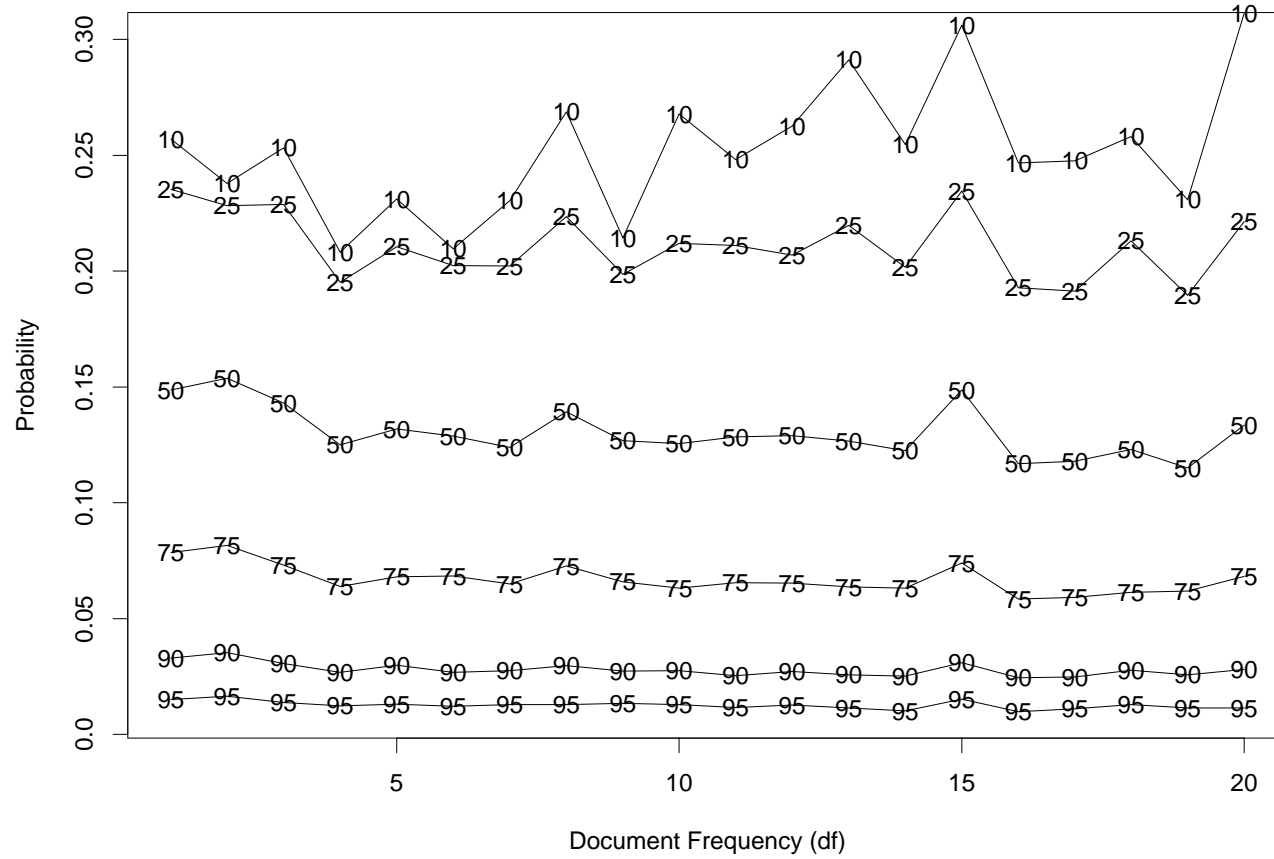
History = first 25%



History = first 75%

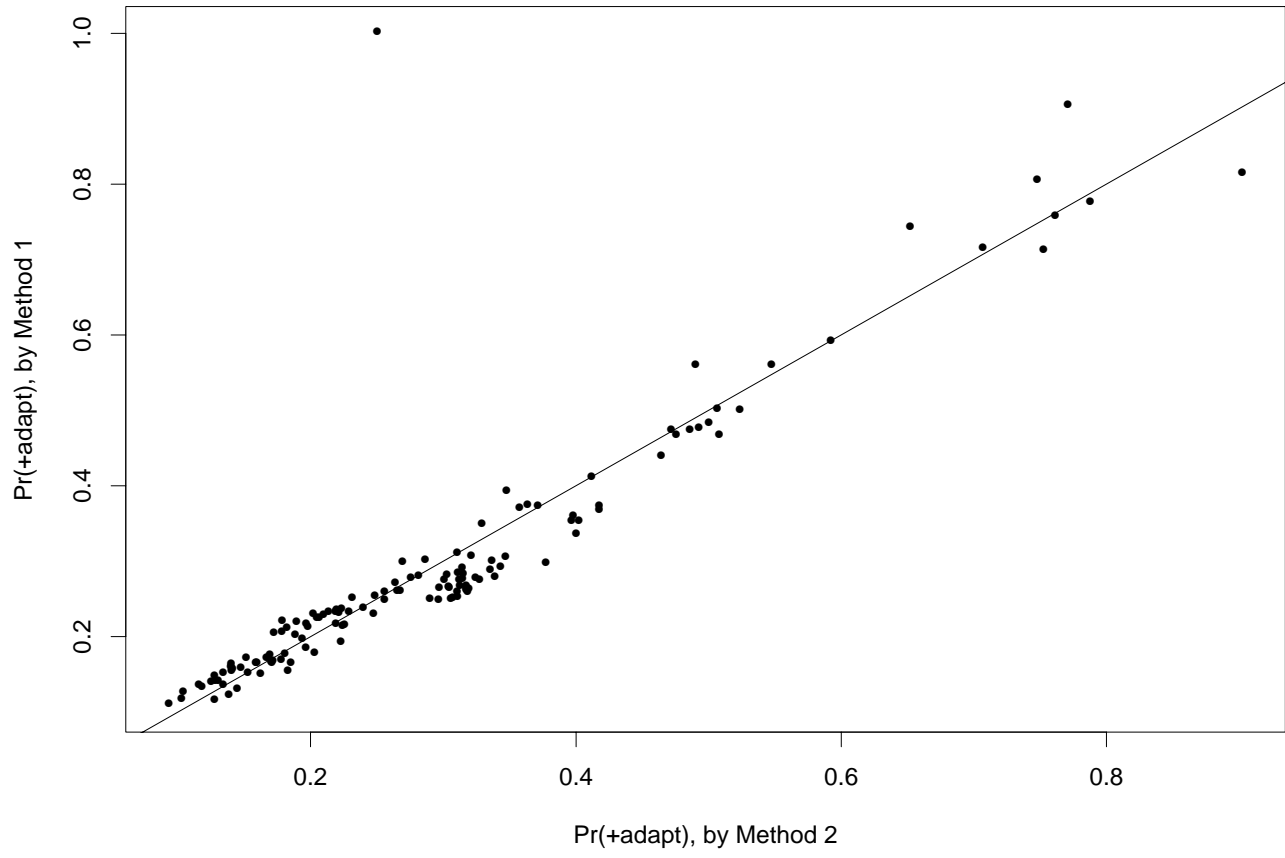


Pr(+adapt) falls as history increases (and test decreases)

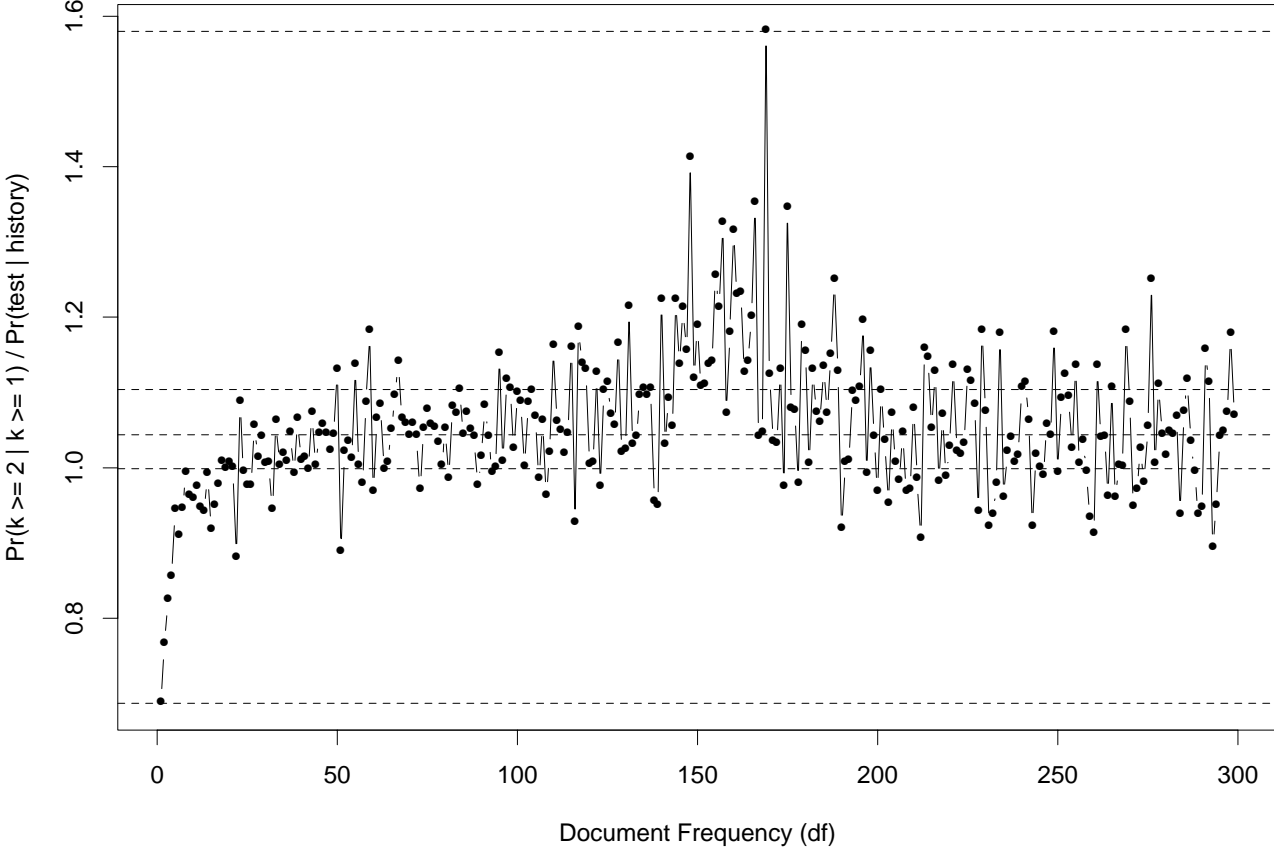


Method 2

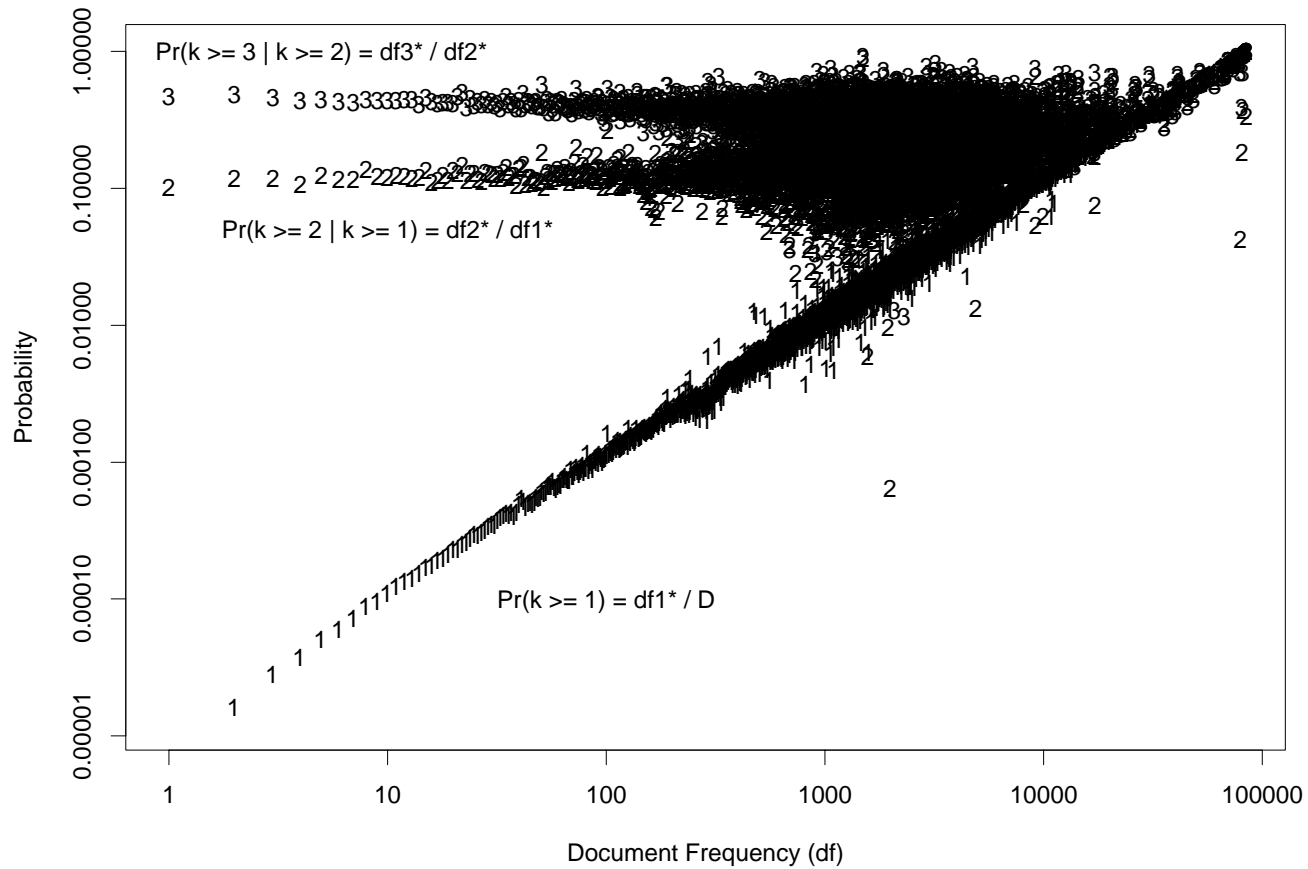
- $Pr(+adapt_2) = Pr(k \geq 2 | k \geq 1) = \frac{df_2}{df_1}$
- where $df_j(w)$ is the number of documents with j or more instances of w (df_1 is the standard notion of df)
- We will show that $Pr(+adapt_1) \approx Pr(+adapt_2)$
 - similar shapes, and
 - within a factor of two
- Suggests that $Pr(+adapt)$ isn't very sensitive to the size of the history (factors of two or so, rather than orders of magnitude).
- Method 2 can be generalized to compute the chance of a third instance $Pr(k \geq 3 | k \geq 2)$.
- Open question: don't know how to use method 2 to estimate $Pr(-adapt_2)$



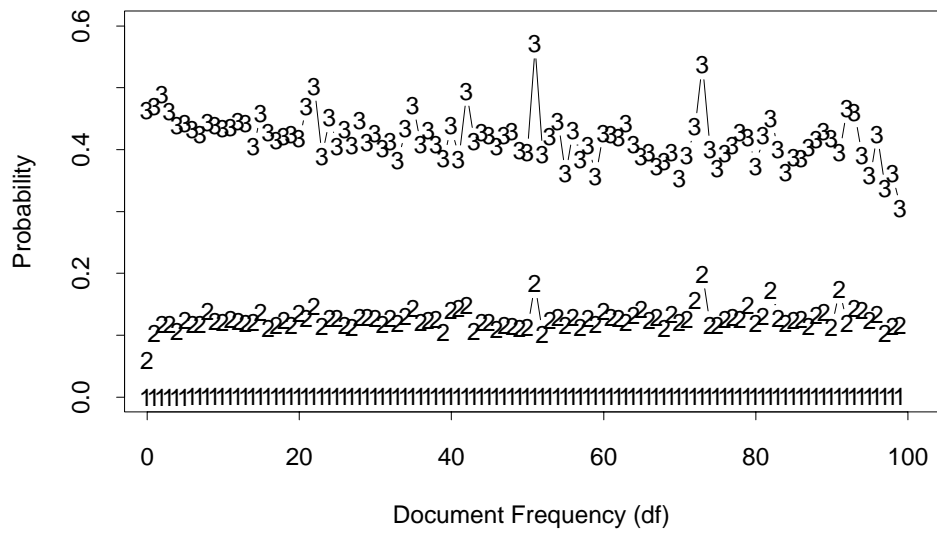
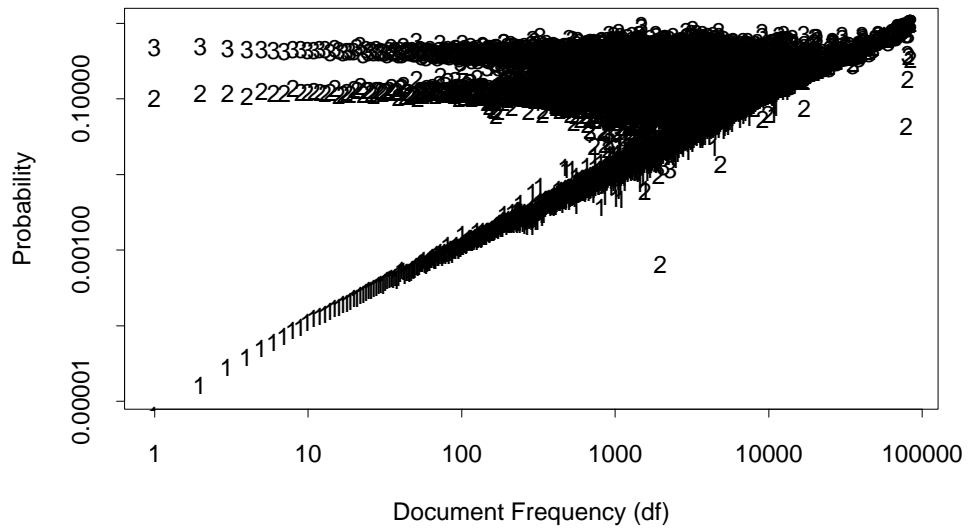
$\Pr(k \geq 2 | k \geq 1)$ is well within a factor of two of $\Pr(\text{test} | \text{history})$



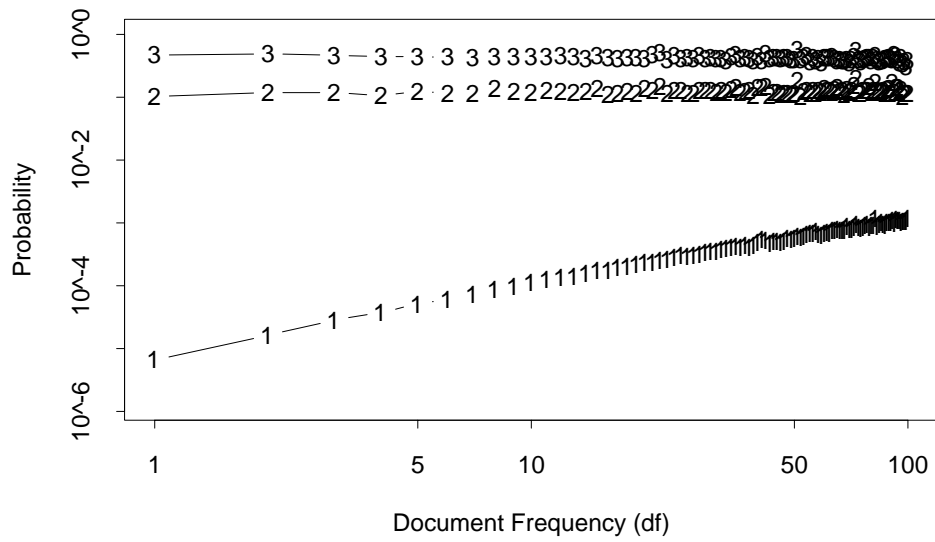
Adaptation is huge (and hardly dependent on frequency)



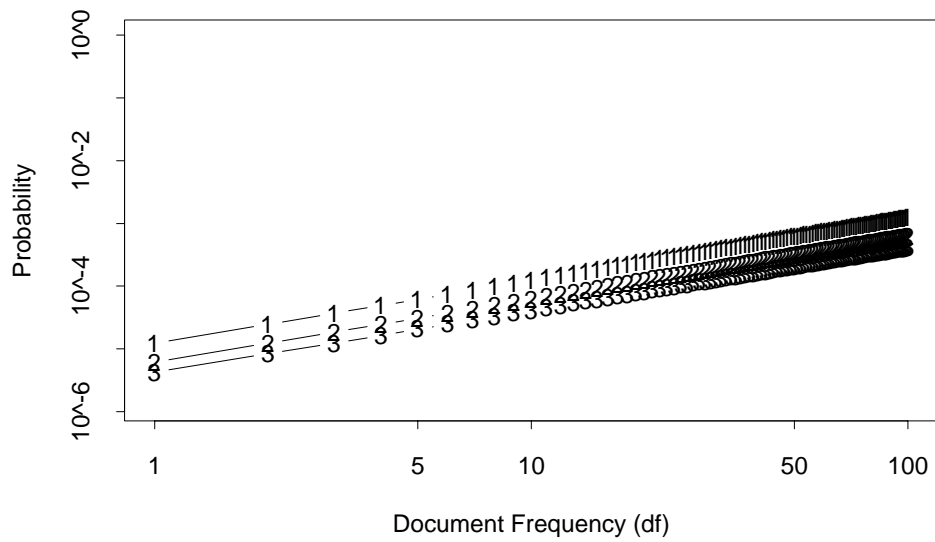
Adaptation is huge (and hardly dependent on frequency)



Adaptation is huge (and hardly depends on freq)



Adaptation is tiny under a Poisson (and depends on freq)



Neighborhoods

- Motivation: suppose history mentions a bunch of words related to a peace process, but doesn't mention the word "peace."
- Context: "It is at least on the Serb side a real setback to the x "
- Probability of "peace" should go up
- Partition vocabulary into three classes: history, neighborhood, otherwise
- neighborhood = IR query expansion of history – history (query expansion uses a different year of AP news)
- $Pr(\textit{peace} \in \textit{test} \mid \textit{peace} \in \textit{history}) \gg$
 $Pr(\textit{peace} \in \textit{test} \mid \textit{peace} \in \textit{neighborhood}) \gg$
 $Pr(\textit{peace} \in \textit{test} \mid \textit{peace} \in \textit{otherwise})$
- nearest neighbors as opposed to clustering

Documents containing “peace”
in 1991 AP News

	test	<u>test</u>
<u>history</u>	2125	2160
<u>history</u>	1963	74573

Documents containing “peace”
in 1991 AP News

	test	<u>test</u>
<i>history</i>	2125	2160
<i>neighborhood</i>	1479	22516
<i>otherwise</i>	484	52057

	test	
<i>history</i>	<i>a</i>	<i>b</i>
<i>history</i>	<i>c</i>	<i>d</i>

	test	
<i>history</i>	<i>a</i>	<i>b</i>
<i>neighborhood</i>	<i>e</i>	<i>f</i>
<i>otherwise</i>	<i>g</i>	<i>h</i>

$$c = e + g, d = f + h$$

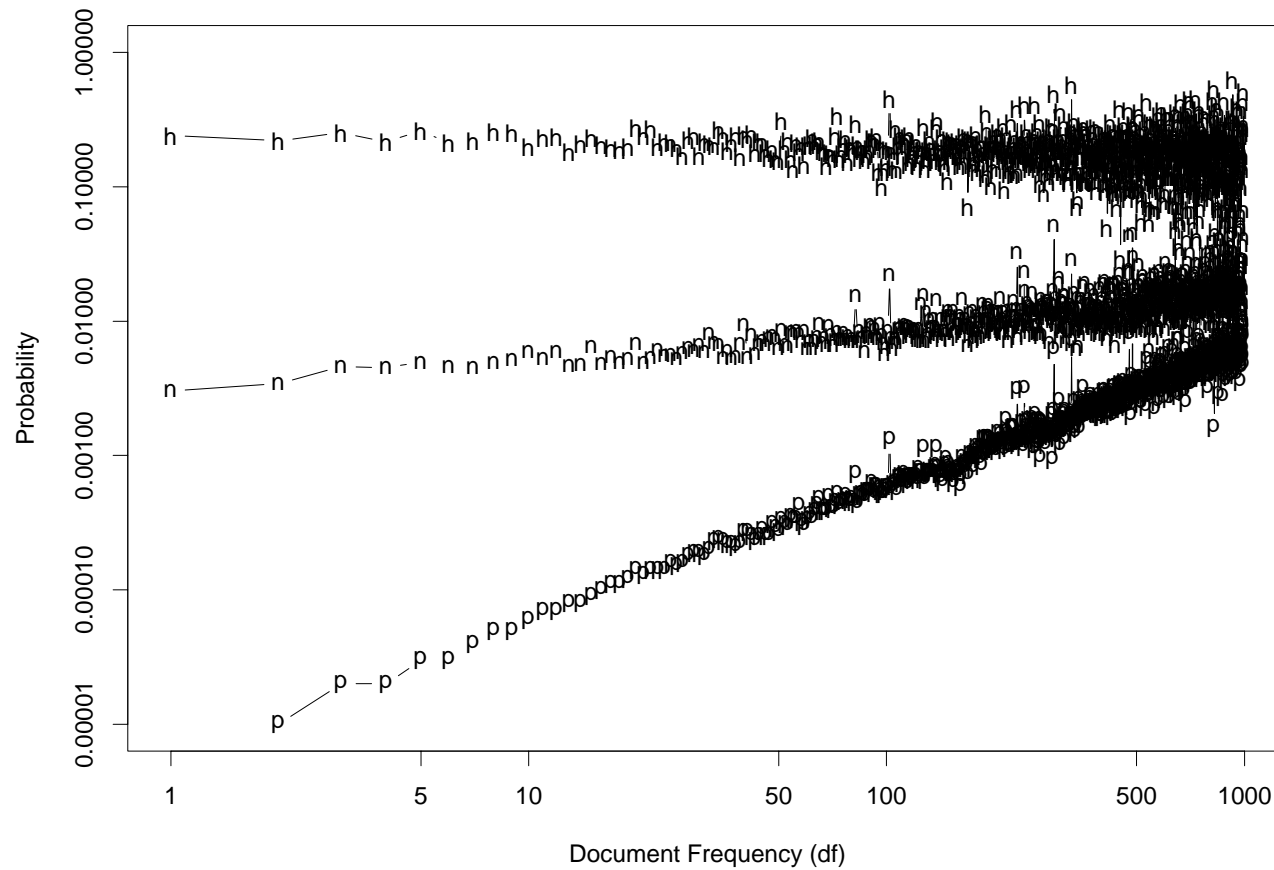
$$Pr(w \in test) \approx \frac{a + c}{D} \quad \text{prior}$$

$$Pr(w \in test | w \in history) \approx \frac{a}{a + b} \quad \text{history}$$

$$Pr(w \in test | w \in neighborhood) \approx \frac{e}{e + f} \quad \text{neighborhood}$$

$$Pr(w \in test | w \in otherwise) \approx \frac{g}{g + h} \quad \text{otherwise}$$

History (h) >> Neighborhood (n) >> Prior (p)



- “Kennedy” adapts more than “except”
- “peace” adapts more than “piece”

adapts more →

$\text{Pr}(\text{history}) \gg \text{Pr}(\text{neighborhood}) \gg \text{Pr}(\text{otherwise})$

prior	history	n'hood	otherwise	src	w
0.026	0.40	0.022	0.0050	AP91	Kennedy
0.020	0.32	0.025	0.0038	AP93	
0.026	0.05	0.018	0.0122	AP91	except
0.019	0.05	0.014	0.0081	AP93	
0.077	0.50	0.062	0.0092	AP91	peace
0.074	0.49	0.066	0.0069	AP93	
0.015	0.10	0.014	0.0066	AP91	piece
0.013	0.08	0.015	0.0046	AP93	

Size of Neighborhood (k)

	test	
<i>history</i>	<i>a</i>	<i>b</i>
<i>neighborhood</i>	<i>e</i>	<i>f</i>
<i>otherwise</i>	<i>g</i>	<i>h</i>

Hostages

a	b	e	f	g	h	e/(e+f)	g/(g+h)	ratio	k
627	436	190	2365	263	76940	0.074	0.003	21.83	1
627	436	236	4077	217	75228	0.055	0.003	19.02	2
627	436	255	5499	198	73806	0.044	0.003	16.56	3
627	436	287	7899	166	71406	0.035	0.002	15.12	5
627	436	310	9963	143	69342	0.030	0.002	14.66	7
627	436	336	12538	117	66767	0.026	0.002	14.92	10
627	436	381	19203	72	60102	0.019	0.001	16.26	20
627	436	417	32506	36	46799	0.013	0.001	16.48	50
627	436	438	46890	15	32415	0.009	0.000	20.01	100

Kennedy

a	b	e	f	g	h	e/(e+f)	g/(g+h)	ratio	k
574	868	137	2556	530	76156	0.051	0.007	7.36	1
574	868	188	4529	479	74183	0.040	0.006	6.21	2
574	868	225	6207	442	72505	0.035	0.006	5.77	3
574	868	276	9238	391	69474	0.029	0.006	5.18	5
574	868	306	11981	361	66731	0.025	0.005	4.63	7
574	868	351	15507	316	63205	0.022	0.005	4.45	10
574	868	452	24733	215	53979	0.018	0.004	4.52	20
574	868	574	41955	93	36757	0.013	0.003	5.35	50
574	868	619	56113	48	22599	0.011	0.002	5.15	100

Enhanced Smoothing

- Simple Good-Turing and Deleted Interpolation use a single variable (df) for binning.
- Enhanced smoothing methods use 2+ variables.
- The table below uses both df_1 and df_2 for binning.
- Note that adaptation varies by nearly an order of magnitude for words with the same df .

df_1	df_2	N_{df_1,df_2}	$Pr(+adapt_1)$	$Pr(+adapt_2)$
5	0	6011	0.06	0.07
5	1	1586	0.18	0.20
5	2	864	0.23	0.25
5	3	491	0.40	0.41
5	4	248	0.40	0.43
5	5	109	0.42	0.50
6	0	4300	0.05	0.05
6	1	1248	0.21	0.20
6	2	674	0.24	0.24
6	3	440	0.33	0.35
6	4	268	0.40	0.41
6	5	119	0.38	0.51
6	6	56	0.56	0.54

Experimental Results

- Train on AP90 & AP91
- Test on AP92

- Measurement: $score(doc) = \sum_{w \in test(doc)} \log_2 \frac{Pr_P(w)}{Pr_B(w)}$

- Proposed:

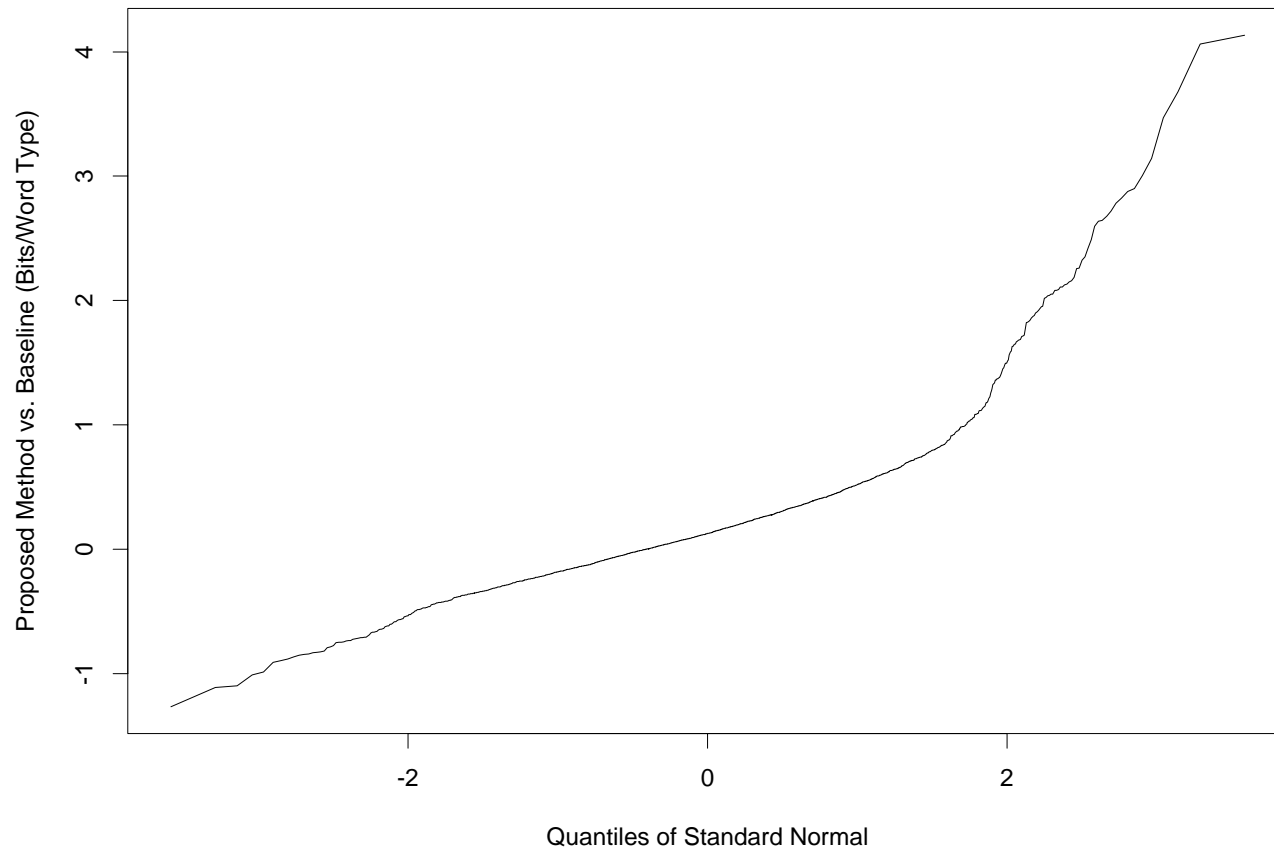
$$Pr_P(w) = \begin{cases} Pr(w|history) & \text{if } w \in history \\ Pr(w|neighborhood) & \text{if } w \in neighborhood \\ Pr(w|otherwise) & \text{otherwise} \end{cases}$$

- Baseline: $Pr(w) = \frac{df}{D}$

- Findings

- Average information gain is small but significant (0.05 bits/word type)
- A few documents gain a lot (2 bits/word type)
- And there are many more big winners (0.8%) than big losers (0.1%)

Proposed method helps a few documents by a lot



- Subjective Observations
 - Examples of big winners:
 - Lists of major cities and their temperatures
 - Lists of major currencies and their prices
 - Lists of commodities and their prices
 - Examples of big losers:
 - Articles that summarize the major stories of the day/week/year.
 - Articles that were garbled in transmission
- Big winners: test is very predictable from history
- Big losers: history is misleading
- Suggestion for future work: measure topic shifts (Hearst, Florian & Yarowsky)

Symmetric ($b \approx c$)

a	b	c	d	src	term
57345	6571	6437	8134	AP90	said
58544	7101	6615	8561	AP91	
44173	5820	6729	10452	AP93	
157	1290	1344	75696	AP90	above
113	1303	1215	78190	AP91	
139	992	1053	64990	AP93	
262	702	704	76819	AP90	Kennedy
574	868	667	78712	AP91	
308	652	624	65590	AP93	

Asymmetric ($b > c$)

a	b	c	d	src	term
2809	70679	273	4726	AP90	AP
2743	72518	220	5340	AP91	
3459	56389	113	7213	AP93	
537	1229	982	75739	AP90	Tokyo
422	1085	768	78546	AP91	
495	852	749	65078	AP93	
1897	6611	5126	64853	AP90	spokesman
1700	6692	5356	67073	AP91	
1135	4732	4080	57227	AP93	

Conclusions

- Described two methods of estimating $Pr(+adapt)$
 - Method 1: $Pr(+adapt_1) = Pr(test|history)$
 - Method 2: $Pr(+adapt_2) = Pr(k \geq 2 | k \geq 1)$
 - Both methods produce similar results (well within factor of two, as opposed to 100-1000)
- Neighborhoods \equiv IR Query Expansion of History

- Cache Model(s):
 - Additive: $Pr(w) = \lambda Pr_L(w) + (1 - \lambda) Pr_G(w)$
 - Case-based: $Pr(w) = \begin{cases} \lambda_1 Pr_L(w) & \text{if } w \in \text{cache} \\ \lambda_2 Pr_G(w) & \text{otherwise} \end{cases}$
- Agree with Jelinek (1997, p. 254): case-based cache models are better than additive cache models
 - λ should depend on w (adaptation is lexical).
 - λ should depend on $w \in \text{cache}$
 - because adaptation is huge (factors of 100-1000)
 - and because $Pr(+adapt)$ is independent of frequency, unlike $Pr(prior)$

Future Work

- Stratify words in neighborhood (not all neighbors are equally valuable for predicting test):
 - Good neighbors appear lots of times in lots of documents near history
 - and hardly anywhere else.
- We split documents into history & test; could do the same for other linguistic units: paragraphs, sentences.
- Measure topic shifts (Hearst, Florian & Yarowsky)
- Model asymmetry
- Unsupervised Approach to Information Retrieval (IR) Research (radical)
 - $neighborhood = query\ expansion(history, IR\ method)$
 - $opt\ IR\ method = \underset{\{IR\ methods\}}{ARGMAX} f(Pr(test|neighborhood))$
- Unsupervised Approach to Language Modeling (less radical)

- Theory of document length
 - Exactly how does $Pr(+adapt_1)$ decrease with $|test|$
- Theory of neighborhood size
 - Hypothesis: $|neighborhood|$ should grow inversely with $|history|$
- Negative Adaptation
 - Method 2: $Pr(+adapt_2) = Pr(k \geq 2 | k \geq 1) = \frac{df_2}{df_1}$
 - But can this method be generalized to estimate $Pr(-adapt_2)$?