n-gram models and statistical estimation

FSNLP, chapter 6

Christopher Manning and
Hinrich Schütze

Probabilistic language modeling

- Assigns probability $P(t)$ to a word sequence $t = w_1 w_2 \cdots w_n$
- Chain rule and joint/conditional probabilities for text t:

$$\begin{aligned} P(t) = P(w_1 \cdots w_n) &= P(w_1) \cdots P(w_n | w_1, \cdots w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i | w_1 \cdots w_{i-1}) \end{aligned}$$

where

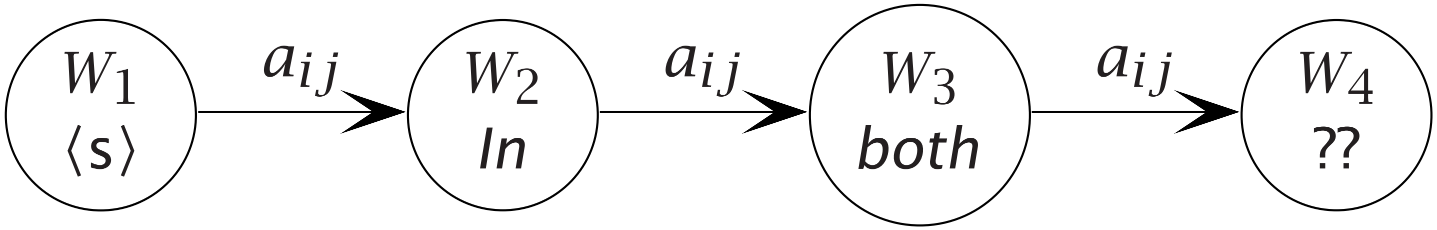
$$P(w_k | w_1 \ldots w_{k-1}) = \frac{P(w_1 \ldots w_k)}{P(w_1 \ldots w_{k-1})} \approx \frac{C(w_1 \ldots w_k)}{C(w_1 \ldots w_{k-1})}$$

- The chain rule leads to a *history-based* model: we predict following things from past things
- We cluster histories into equivalence classes to reduce the number of parameters to estimate

n-gram models: the classic example of a statistical model of language

- Each word is predicted according to a conditional distribution based on a limited context
- Conditional Probability Table (CPT): $P(X|both)$
  - $P(of|both) = 0.066$
  - $P(to|both) = 0.041$
  - $P(in|both) = 0.038$
- From 1940s onward (or even 1910s – Markov 1913)
- a.k.a. Markov (chain) models

n-gram models



- Simplest linear graphical model
- Words are random variables, arrows are dependencies between them (CPTs)
- These simple engineering models have just been amazingly successful.

n-gram models

- Core language model for the engineering task of better predicting the next word:
  - Speech recognition
  - OCR
  - Context-sensitive spelling correction
- It is only recently that they have been improved on for these tasks (Chelba and Jelinek 1998; Charniak 2001).

n-th order Markov models

- First order Markov assumption = bigram

$$P(w_k|w_1 \ldots w_{k-1}) \approx P(w_k|w_{k-1}) = \frac{P(w_{k-1}w_k)}{P(w_{k-1})}$$

- Similarly, n-th order Markov assumption
- Most commonly, trigram (2nd order):

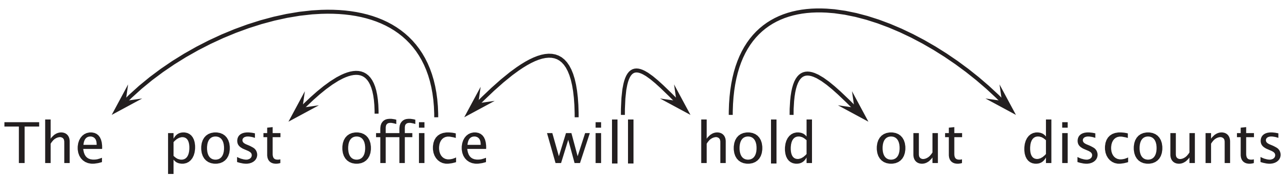
$$P(w_k|w_1 \ldots w_{k-1}) \approx P(w_k|w_{k-1}) = \frac{P(w_{k-1}w_k)}{P(w_{k-2}, w_{k-1})}$$

Why mightn't *n*-gram models work?

- Relationships (say between subject and verb) can be arbitrarily distant and convoluted, as linguists love to point out:
  - The ***man*** that I was watching without pausing to look at what was happening down the street, and quite oblivious to the situation that was about to befall him confidently ***strode*** into the center of the road.

Why do they work?

- That kind of thing doesn't happen much
- Collins (1997):
  - 74% of dependencies (in the Penn Treebank – WSJ) are with an adjacent word (95% with one $\leq$ 5 words away), once one treats simple NPs as units:
  - Below, 4/6 = 66% based on words



Why is that?

Sapir (1921: 14):

> ‘When I say, for instance, “I had a good breakfast this morning,” it is clear that I am not in the throes of laborious thought, that what I have to transmit is hardly more than a pleasurable memory symbolically rendered in the grooves of habitual expression. . . . It is somewhat as though a dynamo capable of generating enough power to run an elevator were operated almost exclusively to feed an electric doorbell.’

Evaluation

- Best evaluation of probability model is task-based
- As substitute for evaluating one component, standardly use corpus per-word cross entropy:

$$H(X, \mathrm{p}) = -\frac{1}{n} \sum_{i=1}^{n} \log_2 P(w_i | w_1, \ldots, w_{i-1})$$

- Or perplexity (measure of uncertainty of predictions):

$$PP(X, \mathrm{p}) = 2^{H(X,\mathrm{p})} = \left[\prod_{i=1}^{n} P(w_i | w_1, \ldots, w_{i-1}) \right]^{-1/n}$$

- *Needs to be assessed on independent, unseen, test data*

Relative frequency = Maximum Likelihood Estimate

$$P(w_2|w_1) = \frac{C(w_1, w_2)}{C(w_1)}$$

(or similarly for higher order or joint probabilities)

Makes training data as probable as possible

Limitations of Maximum Likelihood Estimator

Problem: often infinitely surprised when unseen word appears

- Problem: this happens commonly.
- Probabilities of zero count words are too low
- Probabilities of nonzero count words are too high
- Estimates for high count words are fairly accurate
- Estimates for low count words are unstable
- We need smoothing!

Adding one = Laplace's law

$$P(w_2|w_1) = \frac{C(w_1, w_2) + 1}{C(w_1) + V}$$

- V is the vocabulary size (assume fixed, closed vocabulary)
- This is the Bayesian estimator you get by assuming a uniform unit prior on events: Dirichlet prior

Add one estimator

- Problem: gives too much probability mass to unseens.
- Not good for large vocab, comparatively little data (i.e., NLP)
- e.g 10,000 word vocab, 1,000,000 words of training data, but *comes across* occurs 10 times. Of those, 8 times next word is *as*
  - $P_{\mathsf{MLE}}(\mathit{as}|\mathit{comes\ across}) = 0.8$
  - $P_{+1}(\mathit{as}|\mathit{comes\ across}) = \frac{8+1}{10+10000} \approx 0.0009$
- Quick fix: Lidstone's law (Mitchell's (1997) "m-estimate"):

$$P(w_2|w_1) = \frac{C(w_1, w_2) + \lambda}{C(w_1) + \lambda V}$$

for $\lambda < 1$, e.g., 1/2 or 0.05

Absolute discounting

- Idea is that we want to discount counts of seen things a little, and reallocate this probability mass to unseens
- By subtracting a fixed count, probability estimates for commonly seen things are scarcely affected, while probabilities of rare things are greatly affected
- If the discount is around $\delta = 0.75$, then seeing something once is not so different to not having seen it at all

$$P(w_2|w_1) = (C(w_1, w_2) - \delta)/C(w_1) \quad \text{if} \quad C(w_1, w_2) > 0$$

$$P(w_2|w_1) = (V - N_0)\delta/N_0 C(w_1) \quad \text{otherwise}$$

The frequency of previously unseen events

How do you know how likely you are to see a new word type in the future (in a certain context)?

- Examine some further text and find out [empirical held out estimators = validation]
- Use things you've seen once to estimate probability of unseen things:

$$P(\text{unseen}) = \frac{N_1}{N}$$

where N_1 is number of things seen once. (Good-Turing: Church and Gale 1991; Gale and Sampson 1995)

Good-Turing smoothing

- All words with same count get same probability
- Count mass of words with $r+1$ occurrences assigned to words with r occurrences
- $r*$ is corrected frequency estimate for word occurring r times
- $N_r \times r* = N_{r+1} \times (r+1)$ or
- $r* = \frac{N_{r+1} \times (r+1)}{N_r}$

Good-Turing smoothing

Derivation reflects leave-one out estimation (Ney et al. 1997):

- For each word token in data, call it the test set; remaining data is training set
- See how often word in test set has r counts in training set
- This will happen every time word left out has $r+1$ counts in original data
- So total count mass of r count words is assigned from mass of $r+1$ count words [$= N_{r+1} \times (r+1)$]
- Needs smoothing; accurate when lots of data

Smoothing: Rest of the story (1)

- Other methods: backoff (Katz 1987), cross-validation, Witten-Bell discounting, ... (Chen and Goodman 1998; Goodman 2001)
- Simple, but surprisingly effective: Simple linear interpolation (deleted interpolation; mixture model; shrinkage):

$$\hat{P}(w_3|w_1,w_2) = \lambda_3 P_3(w_3|w_1,w_2) + \lambda_2 P_2(w_3|w_2) + \lambda_1 P_1(w_3)$$

- The λ_i can be estimated on held out data
- They can be functions of (equivalence-classed) histories
- For open vocabulary, need to handle words unseen in any context (just use UNK, spelling models, etc.)

Smoothing: Rest of the story (2)

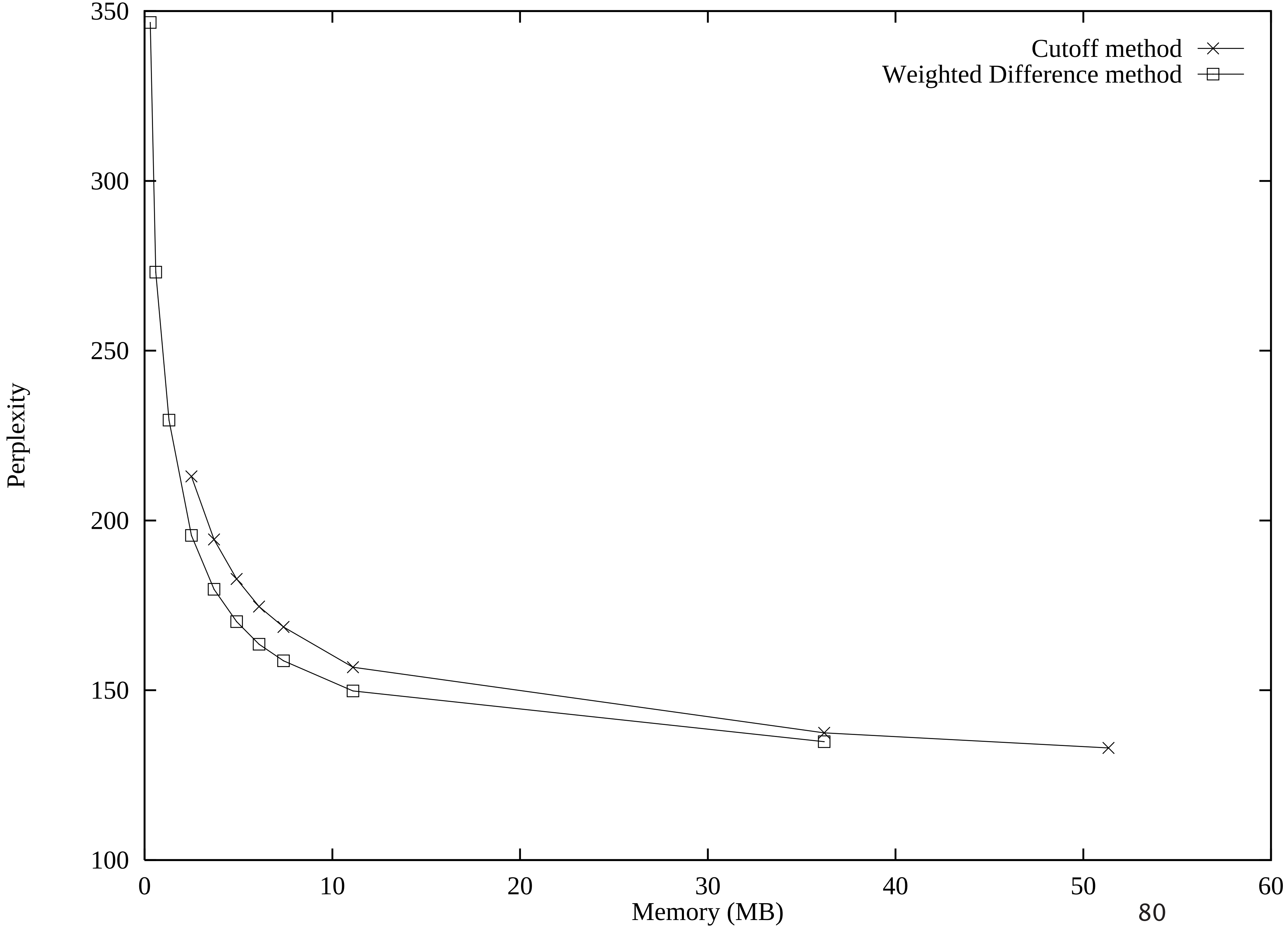
- Recent work emphasizes constraints on the smoothed model
- Kneser and Ney (1995): Backoff n-gram counts not proportional to frequency of n-gram in training data but to expectation of how often it should occur in novel trigram – since one only uses backoff estimate when trigram not found
- (Smoothed) maximum entropy models again place constraints on the distribution (Rosenfeld 1996, 2000)

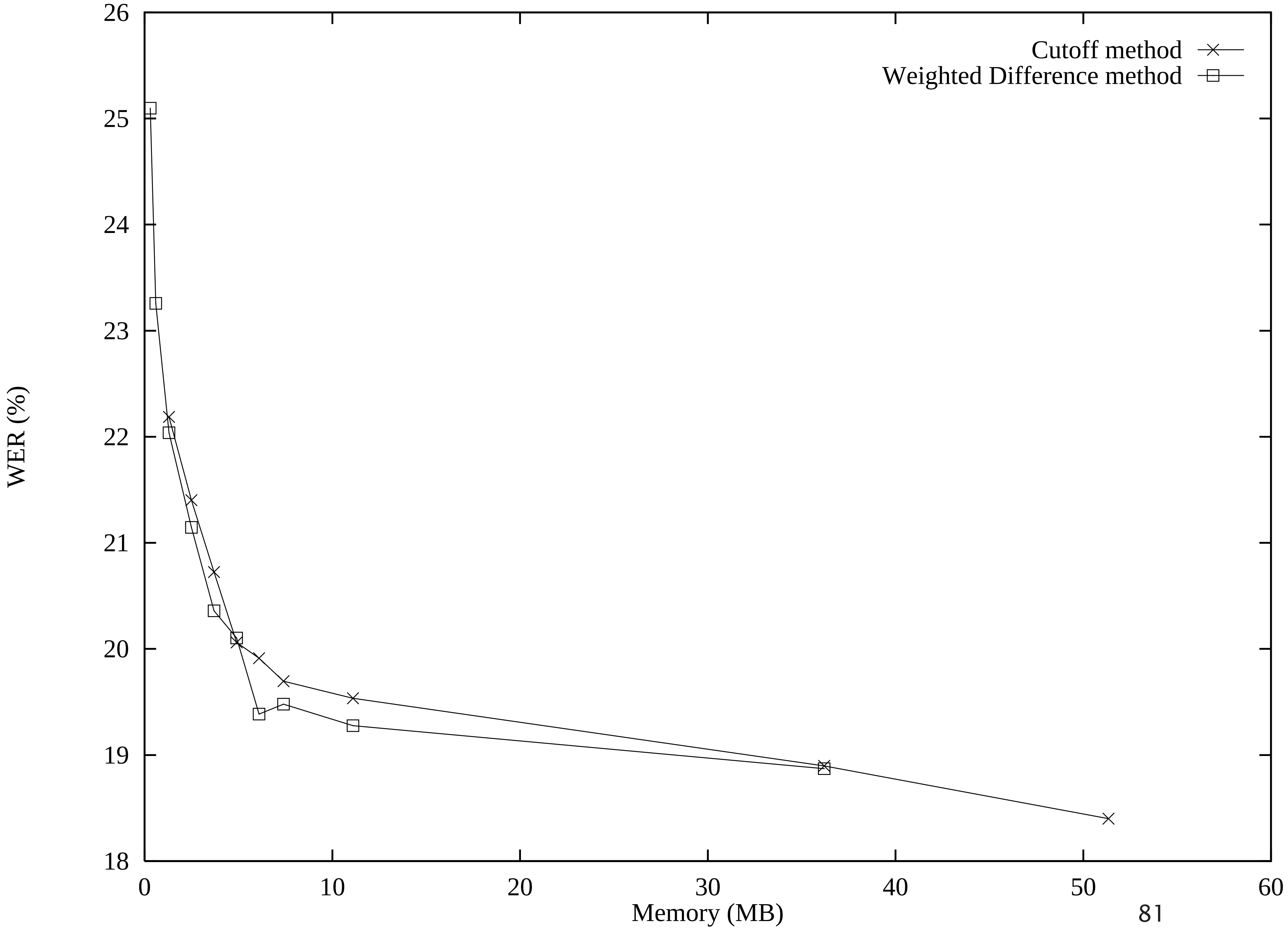
Size of language models with cutoffs

Seymore and Rosenfeld (*ICSLP*, 1996): 58,000 word dictionary, 45 M words of training data, *WSJ*, Sphinx II

Bi/Tri-gram cutoff	# Bigrams	# Trigrams	Memory (Mb)
0/0	4,627,551	16,838,937	104
0/1	4,627,551	3,581,187	51
1/1	1,787,935	3,581,187	29
10/10	347,647	367,928	4

80% of unique trigrams occur only once!


Cutoff method
Weighted Difference method
Perplexity
Memory (MB)
350
300
250
200
150
100
0
10
20
30
40
50
60

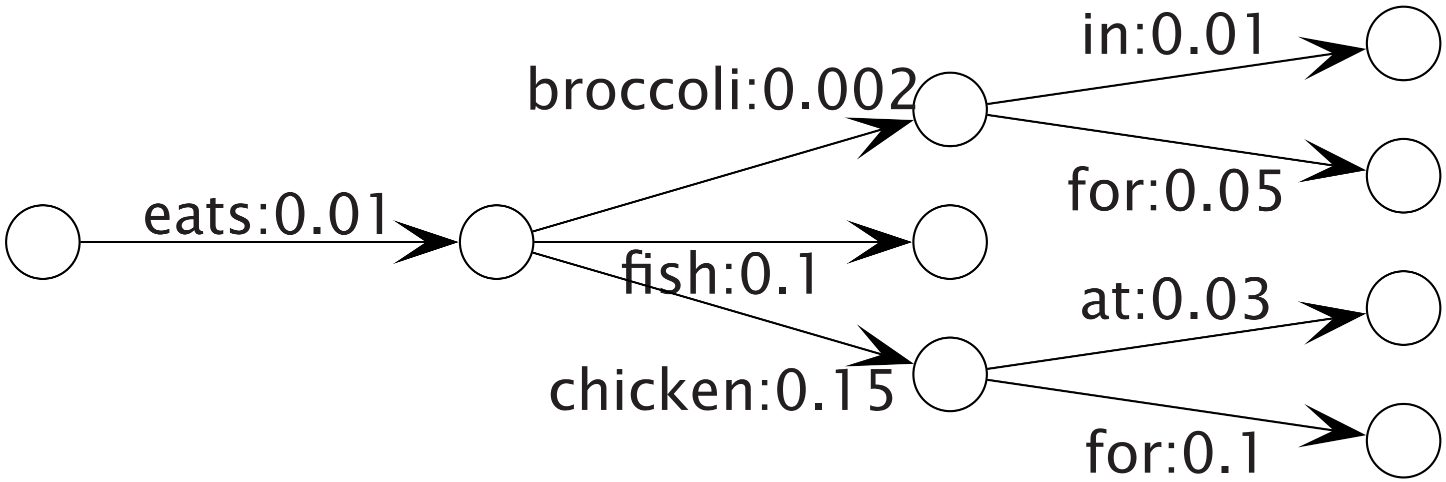

Cutoff method
Weighted Difference method
WER (%)
Memory (MB)
26
25
24
23
22
21
20
19
18
0
10
20
30
40
50
60

More LM facts

- Seymore, Chen, Eskenazi and Rosenfeld (1996)
- HUB-4: Broadcast News 51,000 word vocab, 130M words training. Katz backoff smoothing (1/1 cutoff).
- Perplexity 231
- 0/0 cutoff: 3% perplexity reduction
- 7-grams: 15% perplexity reduction
- Note the possibilities for compression, if you're confident that you'll be given English text (and the encoder/ decoder can use very big tables)

Extra slides

Markov models = n-gram models



Markov models

- Deterministic FSMs with probabilities
- No long distance dependencies
- No notion of structure or syntactic dependency
- But lexical
- (And: robust, have frequency information, ...)

Probability estimates for StatNLP

$$P(X_2|X_1 = w_1) = P(w_2|w_1)$$

Relative frequency = MLE

$$P(w_2|w_1) = \frac{C(w_1, w_2)}{C(w_1)}$$

Problem: often infinitely surprised when unseen word appears

Problem: this happens commonly

Adding one = Laplace's law

$$P(w_2|w_1) = \frac{C(w_1, w_2) + 1}{C(w_1) + V}$$

(Is Bayesian estimator assuming a uniform prior on events.)

Problem: gives too much probability mass to unseens.

Quick fix: Lidstone's law:

$$P(w_2|w_1) = \frac{C(w_1, w_2) + \lambda}{C(w_1) + \lambda V}$$

for $\lambda < 1$, e.g., 1/2

Better methods

- Lots – in chapter 6. We won't dwell on them.
- Simple but surprisingly effective: mixture models. Simple linear interpolation:

$$P(w_2|w_1) = \lambda_1 P_1(w2) + \lambda_2 P_2(w2|w_1)$$

- Open vocabulary. Unknown words.

Managing data

- Training data
- Validation data
- Final testing data
- Cross-validation
  - One score doesn't allow system comparison
  - This allows confidence ranges to be computed
  - And systems to be compared with confidence!

Dealing with unseens: Add one estimator = Laplace's law (1814)

- Bayesian estimator assuming a uniform prior on events:

$$P(w|h) = \frac{C(h,w) + 1}{C(h) + V}$$

- Problem: gives too much probability mass to unseens.
- Not good for large vocab, comparatively little data
- E.g., word trigrams: 10,000 word vocab; 10,000,000 words of training data; *comes across* occurs 10 times. Of those, 8 times next word is *as*
  - $P_{\text{MLE}}(\textit{as}|\textit{comes across}) = 0.8$
  - $P_{+1}(\textit{as}|\textit{comes across}) = \frac{8+1}{10+10000} \approx 0.0009$

The frequency of previously unseen events

- Quick fix: Lidstone's law: $P(w|h) = \frac{C(h,w)+\lambda}{C(h)+\lambda V}$ for $\lambda < 1$, e.g., 1/2 or 0.05
- How do you know how likely you are to see a new word type in the future (in a certain context)?
  - Examine some further text and find out [held out estimators]
  - Use things you've seen once to estimate probability of unseen things:

$$P(\text{unseen}) = \frac{N_1}{N}$$

    N_1 is number of things seen once. (Good-Turing: Church and Gale 1991; Gale and Sampson 1995)

Bibliography

Brants, Thorsten. 2000. TnT – a statistical part-of-speech tagger. In *ANLP 6*, pp. 224–231.

Breiman, L., and J. H. Friedman. 1988. Discussion on article by Loh and Vanichsetakul: 'tree-structured classification via generalized discriminant analysis'. *Journal of the American Statistical Association* 83: 715–727.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1991. Word-sense disambiguation using statistical methods. In *ACL 29*, pp. 264–270.

Bruce, Rebecca, and Janyce Wiebe. 1994. Word-sense disambiguation using decomposable models. In *ACL 32*, pp. 139–145.

Bruce, Rebecca F., and Janyce M. Wiebe. 1999. Decomposable modeling in natural language processing. *Computational Linguistics* 25:195–208.

Cardie, Claire. 1993. Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 25–32.

Charniak, Eugene. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI '97)*, pp. 598–603.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *NAACL 1*, pp. 132–139.

Charniak, Eugene. 2001. Immediate-head parsing for language models. In *ACL 39*.

Chelba, Ciprian, and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *ACL 36/COLING 17*, pp. 225–231.

Chen, Stanley F., and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University.

Church, Kenneth W., and William A. 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.

Collins, Michael John. 1996. A new statistical parser based on bigram lexical dependencies. In *ACL 34*, pp. 184–191.

Collins, Michael John. 1997. Three generative, lexicalised models for statistical parsing. In *ACL 35/EACL 8*, pp. 16–23.

Daelemans, Walter, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger generator. In *WVLC 4*, pp. 14–27.

Dagan, Ido, and Alon Itai. 1994. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics* 20: 563–596.

Domingos, Pedro, and Michael Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29: 103–130.

Duda, Richard O., and Peter E. Hart. 1973. *Pattern classification and scene analysis*. New York: Wiley.

Franz, Alexander. 1997. Independence assumptions considered harmful. In *ACL 35/EACL 8*, pp. 182–189.

Friedman, Jerome H. 1997. On bias, variance, 0/1–loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery* 1:55–77. Also, Technical Report, Stanford University, 1996.

Friedman, Nir, and M. Goldszmidt. 1996. Building classifiers using Bayesian networks. In *AAAI '96*, pp. 1277–1284.

Frietag, Dayne, and Andrew McCallum. 1999. Information extraction with HMMs and shrinkage. In *AAAI '99 Workshop on Machine Learning for Information Extraction*.

Gale, William A., Kenneth W. Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities* 26:415–439.

Gale, William A., and Geoffrey Sampson. 1995. Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics* 2:217–237.

Goodman, Joshua. 2001. A bit of progress in language modeling, extended version. Technical Report MSR-TR-2001-72, Microsoft Research.

Johnson, Mark. 1998. The effect of alternative tree representations on tree bank grammars. In *Proceedings of Joint Conference on New Methods in Language Processing and Computational Natural Language Learning (NeMLaP3/CoNLL98)*, pp. 39–48, Macquarie University.

Katz, Slava M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-35:400–401.

Kilgarriff, Adam. 1997. "I don't believe in word senses". *Computers and the Humanities* 31:91–113.

Kneser, Reinhard, and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 181–184.

Koller, Daphne, and Mehran Sahami. 1996. Toward optimal feature selection. In *ICML '96*, pp. 284–292.

Laplace, Pierre Simon marquis de. 1814. *Essai philosophique sur les probabilites*. Paris: Mme. Ve. Courcier.

Lau, Raymond. 1994. Adaptive statistical language modelling. Master's thesis, Massachusetts Institute of Technology.

Lesk, Michael. 1986. Automatic sense disambiguation: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, pp. 24–26, New York. Association for Computing Machinery.

Magerman, David M. 1995. Review of 'Statistical language learning'. *Computational Linguistics* 11:103–111.

Manning, Christopher D., and Bob Carpenter. 1997. Probabilistic parsing using left corner language models. In *Proceedings of the Fifth International Workshop on Parsing Technologies (IWPT-97)*, pp. 147–158, MIT.

McCallum, Andrew, and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Working Notes of the 1998 AAAI/ICML Workshop on Learning for Text Categorization*.

Mitchell, Tom M. (ed.). 1997. *Machine Learning*. New York: McGraw-Hill.

Ney, Hermann, Sven Martin, and Frank Wessel. 1997. Statistical language modeling using leaving-one-out. In Steve Young and Gerrit Bloothooft (eds.), *Corpus-Based Methods in Language and Speech Processing*, pp. 174–207. Dordrecht: Kluwer Academic.

Ratnaparkhi, Adwait. 1997. A linear observed time statistical parser based on maximum entropy models. In *EMNLP 2*, pp. 1–10.

Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

Rosenfeld, Roni. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language* 10: 187–228.

Rosenfeld, Ronald. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE* 88.

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics* 24:97–124.

Yarowsky, David. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *COLING 14*, pp. 454–460.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL 33*, pp. 189–196.