Fifth Biennial Conference of the
Association for Machine Translation in the Americas

Tutorial Notes

The State of the Art in Language Modeling

Joshua Goodman
Microsoft Research
Machine Learning and Applied Statistics Group

October 9, 2002
Tiburon Lodge
Tiburon, California

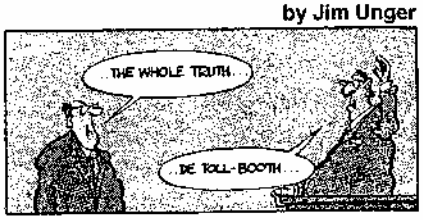
The State of The Art in Language Modeling
(Emphasis on Machine Translation)

Joshua Goodman
Microsoft Research
Machine Learning and Applied Statistics Group
http://www.research.microsoft.com/~joshuago

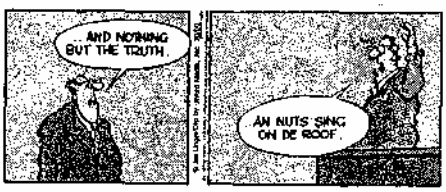
A bad language model

A bad language model

by Jim Unger

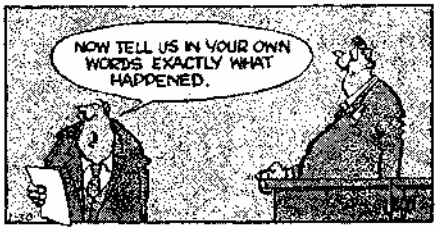


A bad language model



SA1-4

A bad language model



SA1-5

What's a Language Model

- A language model is a probability distribution over word sequences
- P("And nothing but the truth") ≈ 0.001
- P("And nuts sing on the roof") ≈ 0

SA1-6

Language Models for Machine Translation

- P("The spirit is willing") ≈ 0.001
- P("The wine is willing") ≈ 0.0001
- P("Wine the willing is") ≈ 0.000001
- Language model tells you which translations are likely, and which ones are not
  - a key part of statistical machine translation systems

What's a language model for?

- *Machine translation*
- Speech recognition
- Handwriting recognition
- Spelling correction
- Optical character recognition
- Typing in Chinese or Japanese

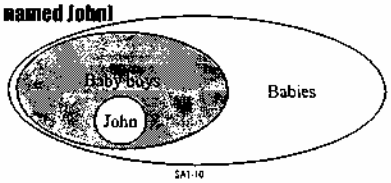
- (and anyone doing statistical modeling)

Really Quick Overview

- ✓ Humor
- ✓ What is a language model?
- ➢ Really quick overview
- Two minute probability overview
- One minute source channel overview
- How language models work (trigrams)
- Language models for machine translation
- Real overview
- More source channel, smoothing, caching, skipping, sentence-mixture models, clustering, parsing language models, tools

Everything you need to know about probability -- definition

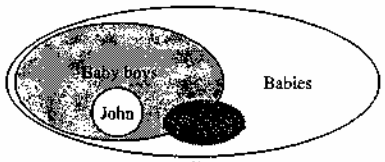
- P(X) means probability that X is true
  - P(baby is a boy) ≈ 0.5 (% of total that are boys)
  - P(baby is named John) ≈ 0.001 (% of total named John)



SAT-10

Everything about probability Joint probabilities

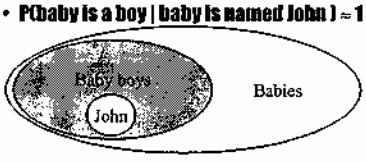
- P(X, Y) means probability that X and Y are both true, e.g. P(brown eyes, boy)



SAT-11

Everything about probability: Conditional probabilities

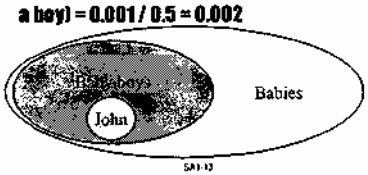
- P(X|Y) means probability that X is true when we already know Y is true
  - P(baby is named John | baby is a boy) ≈ 0.002
  - P(baby is a boy | baby is named John) ≈ 1



SAT-12

Everything about probabilities: math

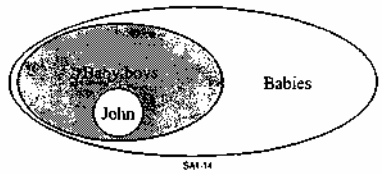
- P(X|Y) = P(X, Y) / P(Y)
  - P(baby is named John | baby is a boy) = P(baby is named John, baby is a boy) / P(baby is a boy) = 0.001 / 0.5 = 0.002



SA1-13

Everything about probabilities: Bayes Rule

- Bayes rule: P(X|Y) = P(Y|X) × P(X) / P(Y)
- P(named John | boy) = P(boy | named John) × P(named John) / P(boy)



SA1-14

THE Equation

$$\arg\max_{wordsequence} P(wordsequence \mid acoustics) =$$

$$\arg\max_{wordsequence} \frac{P(acoustics \mid wordsequence) \times P(wordsequence)}{P(acoustics)}$$

$$\arg\max_{wordsequence} P(acoustics \mid wordsequence) \times P(wordsequence)$$

$$\arg\max_{wordsequence} P(english \mid french) =$$

$$\arg\max_{wordsequence} \frac{P(french \mid english) \times P(english)}{P(french)}$$

$$\arg\max_{wordsequence} P(french \mid english) \times P(english)$$

Why use source channel?

- Example: trivial translation model
- 'the' → 'the' (1.0)
- 'spirit' → 'spirit' (0.3)
- 'spirit' → 'wine' (0.7)
- 'is' → 'is' (1.0)
- 'willing' → 'willing' (1.0)
- P('the spirit is willing' | 'the' 'spirit' 'is' 'willing') = 0.3
- P('the wine is willing' | 'the' 'spirit' 'is' 'willing') = 0.7

Why use source channel?

- 'spirit' → 'spirit' (0.3)
- 'wine' → 'spirit' (0.7)
- P('the' 'spirit' 'is' 'willing' | the *spirit* is willing) × P(the *spirit* is willing) =
  $0.3 \times 0.001 = .003$
- P('the' 'spirit' 'is' 'willing' | the *wine* is willing) × P(the *wine* is willing) =
  $0.7 \times 0.0001 = .0007$

Advantages of source channel

- Can train the language model with much more data than the translation model
- Can use two kinds of information – probability of translation, and probability of resulting strings
- Break the model in two pieces
- Can use everything we already know about language modeling

How Language Models work

- Hard to compute P("And nothing but the truth")
- Step 1: Decompose probability

P("And nothing but the truth") =
P("And") × P("nothing|and") × P("but|and nothing") × P("the|and nothing but") × P("truth|and nothing but the")

SAI-19

The Trigram Approximation

Assume each word depends only on the previous two words (three words total – tri means three, gram means writing)

P("the|... whole truth and nothing but") ≈
P("the|nothing but")
P("truth|... whole truth and nothing but the") ≈
P("truth|but the")

SAI-20

Trigrams, continued

- How do we find probabilities?
- Get real text, and start counting!
  - P("the | nothing but") ≈
    C("nothing but the") / C("nothing but")

SAI-21

Real Overview Overview

- ✓ Basics: probability, language model definition
- ➢ Real Overview (8 slides)
- The source channel model
- Evaluation
- Smoothing
- Caching, Skipping
- Clustering
- Sentence-mixture models
- Parsing language models
- Tools

Real Overview: Evaluation

- Need to compare different language models
- Speech recognition word error rate
- Machine translation accuracy
- Perplexity
- Entropy
- Coding theory

The Source Channel Model

- Use P(f|e) ×P(e) instead of P(e|f)
- Quick discussion of P(f|e) (channel model)
- You can use most of these techniques for the channel model, too.

Real Overview: Smoothing

- Got trigram for P("the" | "nothing but") from C("nothing but the") / C("nothing but")
- What about P("sing" | "and nuts") = C("and nuts sing") / C("and nuts")
- Probability would be 0: very *bad*!

Real Overview: Caching

- If you say something, you are likely to say it again later

Real Overview: Skipping

- Trigram uses last two words
- Other words are useful too – 3-back, 4-back
- Words are useful in various combinations (e.g. 1-back (bigram) combined with 3-back)

Real Overview: Clustering

- What is the probability
  P("Tuesday | party on")
- Similar to P("Monday | party on")
- Similar to P("Tuesday | celebration on")
- Put words in clusters:
  - WEEKDAY = Sunday, Monday, Tuesday, ...
  - EVENT=party, celebration, birthday, ...

Real Overview: Sentence Mixture Models

- In Wall Street Journal, many sentences
  "In heavy trading, Sun Microsystems fell 25 points yesterday"
- In Wall Street Journal, many sentences
  "Nathan Myhrvold, vice president of Microsoft, took a one year leave of absence."
- *Model each sentence type separately.*

Real Overview: Parsing Language Models

- Language has structure – noun phrases, verb phrases, etc.
- "The butcher from Albuquerque slaughtered chickens" – even though slaughtered is far from butchered, it is predicted by butcher, not by Albuquerque
- Recent, somewhat promising models

Real Overview: Tools

- You can make your own language models with tools freely available for research
- CMU language modeling toolkit
- SRI language modeling toolkit

SAI-31

Evaluation

- How can you tell a good language model from a bad one?
- Run a machine translation system, a speech recognizer (or your application of choice), calculate word error rate
  - Slow
  - Specific to your system

SAI-32

Evaluation: Perplexity Intuition

- Ask a speech recognizer to recognize digits: "0, 1, 2, 3, 4, 5, 6, 7, 8, 9" – easy – perplexity 10
- Ask a speech recognizer to recognize names at Microsoft – hard – 30,000 – perplexity 30,000
- Ask a speech recognizer to recognize "Operator" (1 in 4), "Technical support" (1 in 4), "sales" (1 in 4), 30,000 names (1 in 120,000) each – perplexity 54
- Perplexity is weighted equivalent branching factor.

SAI-33

Evaluation: perplexity

- "A, B, C, D, E, F, G...Z": perplexity is 26
- "Alpha, bravo, charlie, delta...yankee, zulu": perplexity is 26
- Perplexity measures language model difficulty, not acoustic difficulty.

Perplexity: Math

- Perplexity is geometric average inverse probability

$$\sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i \mid w_{1..i-1})}}$$

- Imagine "Operator" (1 in 4), "Technical support" (1 in 4), "sales" (1 in 4), 30,000 names (1 in 120,000)
- Model thinks all probabilities are equal (1/30,003)
- Average inverse probability is 30,003

Perplexity: Math

- Imagine "Operator" (1 in 4), "Technical support" (1 in 4), "sales" (1 in 4), 30,000 names (1 in 120,000)
- Correct model gives these probabilities
- ¾ of time assigns probability ¼, ¼ of time assigns probability 1/120,000
- Perplexity is 54 (compare to 30,003 for simple model)
- *Remarkable fact: the true model for data has the lowest possible perplexity*

Perplexity: Is lower better?

- Remarkable fact: the true model for data has the lowest possible perplexity
- Lower the perplexity, the closer we are to true model.
- Typically, perplexity correlates well with speech recognition word error rate
  - Correlates better when both models are trained on same data
  - Doesn't correlate well when training data changes
  - Franz Och's results show good correlation for MT

SA1-37

Perplexity: The Shannon Game

- Ask people to guess the next letter, given context. Compute perplexity.

Char n-gram	Low Char	Upper char	Low word	Upper word
1	9.1	16.3	191,237	4,702,511
5	3.2	6.5	653	29,532
10	2.0	4.3	45	2,998
15	2.3	4.3	97	2,998
100	1.5	2.5	10	142

  - (when we get to entropy, the "100" column corresponds to the "1 bit per character" estimate)

SA1-38

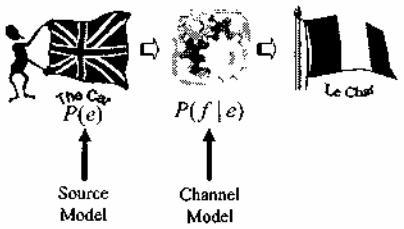
Evaluation: entropy

- Entropy = $\log_2$ perplexity $\log_2 \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i \mid w_{1..i-1})}}$
  - Should be called "cross-entropy of model on test data."
  - Remarkable fact: entropy is average number of bits per word required to encode test data using this probability model, and an optimal coder. Called bits.

SA1-39

The Source Channel Model

- Use P(f|e)×P(e) instead of P(e|f)



Source Model

Channel Model

Source Model

- Source model gives P(e)
- Just a language model
  - Typically, have used trigrams
  - One goal today: get people to use more interesting techniques
  - Speech recognizers have a lot of constraints that machine translation may not have

SA1-41

Channel Model

- Lots of different channel models
  - Most research in stat MT is on the channel
- IBM "Model 4" is a nice example
  - Translate each English word into some number of French words ("fertility model")
  - Select each French word
  - Place each French word ("distortion model")

SA1-42

Language model techniques for channel models

- This talk focuses on language modeling for source model
- Many language model techniques can be used for channel model
  - Channel model can be smoothed
  - Channel model can use clusters
  - Channel model can use cache

SA143

Smoothing: None

$$P(z \mid xy) \approx \frac{C(xyz)}{\sum_w C(xyw)} = \frac{C(xyz)}{C(xy)}$$

- Called Maximum Likelihood estimate.
- Lowest perplexity trigram on training data.
- Terrible on test data: If no occurrences of C(xyz), probability is 0.

SA144

Smoothing: Add One

- What is P(sing|nuts)? Zero? Leads to infinite perplexity!
- Add one smoothing: $P(z \mid xy) \approx \frac{C(xyz)+1}{C(xy)+V}$
- Works very badly. DO NOT DO THIS
- Add delta smoothing: $P(z \mid xy) \approx \frac{C(xyz)+\delta}{C(xy)+V\delta}$
- Still very bad. DO NOT DO THIS

SA145

Smoothing: Simple Interpolation

$$P(z \mid xy) \approx \lambda \frac{C(xyz)}{C(xy)} + \mu \frac{C(yz)}{C(y)} + (1-\lambda-\mu)\frac{C(z)}{C(\bullet)}$$

- Trigram is very context specific, very noisy
- Unigram is context-independent, smooth
- Interpolate Trigram, Bigram, Unigram for best combination
- Find $0<\lambda,\mu<1$ by optimizing on "held-out" data
- Almost good enough

SA1-46

Smoothing: Finding parameter values

- Split data into training, "heldout", test
- Try lots of different values for λ,μ on heldout data, pick best
- Test on test data
- Sometimes, can use tricks like "EM" (estimation maximization) to find values
- I prefer to use a generalized search algorithm, "Powell search" – see Numerical Recipes in C

SA1-47

Smoothing digression: Splitting data

- How much data for training, heldout, test?
- Some people say things like "1/3, 1/3, 1/3" or "80%, 10%, 10%" They are WRONG
- Heldout should have (at least) 100-1000 words per parameter.
- Answer: enough test data to be statistically significant. (1000s of words perhaps)

SA1-48

Smoothing digression: Splitting data

- Be careful: WSJ data divided into stories. Some are easy, with lots of numbers, financial, others much harder. Use enough to cover many stories.
- Be careful: Some stories repeated in data sets.
- Can take data from end – better – or randomly from within training. Temporal effects like "Elian Gonzalez"

SA1-49

Smoothing: Jelinek-Mercer

- Simple interpolation:

$$P_{smooth}(z \mid xy) = \lambda \frac{C(xyz)}{C(xy)} + (1-\lambda) P_{smooth}(z \mid y)$$

- Better: smooth a little after "The Dow", lots after "Adobe acquired"

$$P_{smooth}(z \mid xy) = \lambda(C(xy)) \frac{C(xyz)}{C(xy)} + (1-\lambda(C(xy))) P_{smooth}(z \mid y)$$

SA1-50

Smoothing: Jelinek-Mercer continued

$$P_{smooth}(z \mid xy) = \lambda(C(xy)) \frac{C(xyz)}{C(xy)} + (1-\lambda(C(xy))) P_{smooth}(z \mid y)$$

- Put λs into buckets by count
- Find λs by cross-validation on held-out data
- Also called "deleted-interpolation"

SA1-51

Smoothing: Katz

- Compute discount using "Good-Turing" estimate
- Only use bigram if trigram is missing

$$P_{Katz}(z \mid xy) = \begin{cases} \dfrac{C(xyz) - D(C(xyz))}{C(xy)} & \text{if } C(xyz) > 0 \\ \alpha(xy) P_{Katz}(z \mid y) & \text{otherwise} \end{cases}$$

- Works pretty well, except not good for 1 counts
- α is calculated so probabilities sum to 1

SAI-52

Smoothing: Interpolated Absolute Discount

- JM, Simple Interpolation overdiscount large counts, underdiscount small counts

$$\lambda(C(xy)) \frac{C(xyz)}{C(xy)} + (1 - \lambda(C(xy))) P_{smooth}(z \mid y)$$

- "San Francisco" 100 times, "San Joshua" once, should we use a big discount or a small one?
  - Absolute discounting takes the same from everyone

$$P_{abs-interp}(z \mid xy) = \frac{C(xyz) - D}{C(xy)} + \beta(xy) P_{abs-interp}(z \mid x)$$

Smoothing: Interpolated Multiple Absolute Discounts

- One discount is good

$$\frac{C(xyz) - D}{C(xy)} + \beta(xy) P_{abs-interp}(z \mid x)$$

- Different discounts for different counts

$$\frac{C(xyz) - D_{C(xyz)}}{C(xy)} + \beta(xy) P_{abs-interp}(z \mid y)$$

- Multiple discounts: for 1 count, 2 counts, >2

SAI-54

Smoothing: Kneser-Ney

P(Francisco | eggplant) vs P(stew | eggplant)

- "Francisco" is common, so backoff, interpolated methods say it is likely
- But it only occurs in context of "San"
- "Stew" is common, and in many contexts
- Weight backoff by number of contexts word occurs in

SA1-55

Smoothing: Kneser-Ney

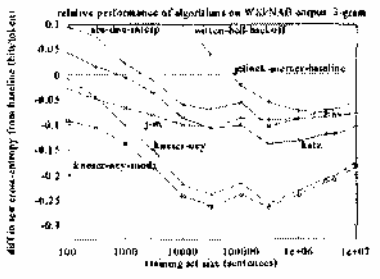
- Interpolated
- Absolute-discount
- Modified backoff distribution
- Consistently best technique

$$\frac{C(xyz)-D_{C(xyz)}}{C(xy)}+$$

$$\beta(xy)\frac{|\{w|C(wyz)>0\}|}{\sum_{v}|\{w|C(wyv)>0\}|}$$

SA1-56

Smoothing: Chart



Smoothing the Channel Model

- Channel model usually estimated with EM algorithm
- Results in fractional counts, e.g. 0.2, 0.8, 1.3, etc.
- Kneser-Ney not yet studied for fractional counts
- Use simple interpolation, or Jelinek-Mercer, or experiment with Kneser-Ney

SA1-58

Caching

- If you say something, you are likely to say it again later.
- Interpolate trigram with cache

$$P(z \mid history) \approx \lambda P_{smooth}(z \mid xy) + (1-\lambda) P_{cache}(z \mid history)$$

$$P_{cache}(z \mid history) = \frac{C(z \in history)}{length(history)}$$

SA1-59

Caching: Real Life

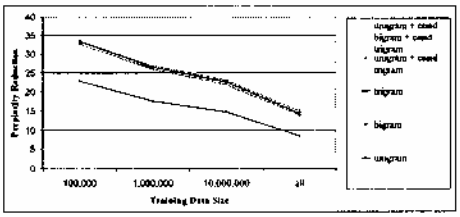
- Someone says "I swear to tell the truth"
- System hears "I swerve to smell the soup"
- Cache remembers!
- Person says "The whole truth", and, with cache, system hears "The whole soup." – errors are locked in.
- Caching works well when users correct as they go, poorly or even hurts without correction.

SA1-60

Caching: Variations

- N-gram caches: $P_{cache}(z \mid history) \approx \frac{C(xyz \in history)}{C(xy \in history)}$
- Conditional n-gram cache: use n-gram cache only if $xy \in$ history
- Remove function-words like "the", "to"

Cache Results



Caching for machine translation Language model

- Someone translates "My' spirit' is' soaring'" to "My spirit is flying"
- Later, translate "My' spirit' is' willing'" to "My spirit is willing" – cache language model makes "my spirit" more likely than "my wine"
  - (But if it translated "My' spirit' is' soaring" to "My wine is flying" and user doesn't correct it, you make things worse)

Caching for machine translation Channel model

- Can use caching ideas in translation model as well
- P_{global} (spirit $\rightarrow$ spirit') = 0.4
- P_{global} (wine $\rightarrow$ spirit') = 0.6
- P_{cache} (spirit $\rightarrow$ spirit') = 1.0
- ½ P_{global} (spirit $\rightarrow$ spirit') + ½ P_{cache} (spirit $\rightarrow$ spirit') = 0.7

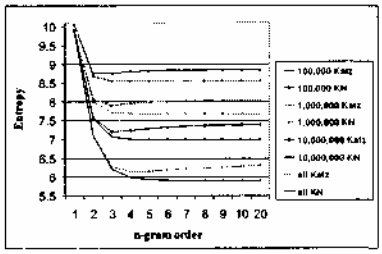
SAI-64

5-grams

- Why stop at 3-grams?
- If P(z|...rstuvwxy) ≈ P(z|xy) is good, then P(z|...rstuvwxy) ≈ P(z|vwxy) is better!
- Very important to smooth well
- Interpolated Kneser-Ney works much better than Katz on 5-gram, more than on 3-gram

SAI-65

N-gram versus smoothing algorithm



SAI-66

Speech recognizer mechanics

- Keep many hypotheses alive

"…tell the" (.01)
"…smell the" (.01)

- Find acoustic, language model scores
  - P(acoustics | truth = .3), P(truth | tell the) = .1
  - P(acoustics | soup = .2), P(soup | smell the) = .01

"…tell the truth" (.01 × .3 ×.1)
"…smell the soup" (.01 × .2 ×.01)

Speech recognizer slowdowns

- Speech recognizer uses tricks (dynamic programming) to merge hypotheses

Trigram:

"…tell the"
"…smell the"

Fivegram:

"…swear to tell the"
"…swerve to smell the"
"…swear too tell the"
"…swerve too smell the"
"…swerve to tell the"
"…swerve too tell the"
…

SA1-98

Speech recognizer vs. n-gram

- Recognizer can threshold out bad hypotheses
- Trigram works so much better than bigram, better thresholding, no slow-down
- 4-gram, 5-gram start to become expensive

SA1-99

5-grams for MT

- Very different search techniques
- Often can't do much dynamic programming anyway
- So, MT can use 5-grams, even if speech recognizer can't!

Speech recognizer with language model

- In theory,

$$\arg\max_{wordsequence} P(acoustics \mid wordsequence) \times P(wordsequence)$$

- In practice, language model is a better predictor -- acoustic probabilities aren't "real" probabilities
- In practice, penalize insertions

$$\arg\max_{wordsequence} \frac{P(acoustics \mid wordsequence) \times}{P(wordsequence)^{8} \times .1^{length(wordsequence)}}$$

Weighting LMs for MT

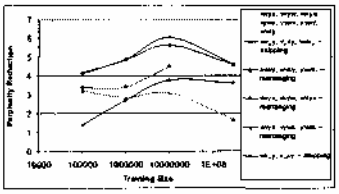
- Same thing applies for MT
- IBM '93 paper mentions insertion penalty
- Och '02 introduces LM weight, other weights, for MT. Och shows how to optimize them using maxent techniques

Skipping

- P(z|...rstuvwxy) ≈ P(z|vwxy)
- Why not P(z|v_xy) – "skipping" n-gram – skips value of 3-back word.
- Example: "P(time|show John a good)" -> P(time | show _____ a good)
- P(...rstuvwxy) ≈
  λP(z|vwxy) + μP(z|vw_y) + (1-λ-μ)P(z|v_xy)

SA1-73

5-gram Skipping Results



(Best trigram skipping result: 11% reduction)

SA1-74

Clustering

- CLUSTERING = CLASSES (same thing)
- What is P("Tuesday | party on")
- Similar to P("Monday | party on")
- Similar to P("Tuesday | celebration on")
- Put words in clusters:
  - WEEKDAY = Sunday, Monday, Tuesday, ...
  - EVENT=party, celebration, birthday, ...

SA1-75

Clustering overview

- Major topic, useful in many fields
- Kinds of clustering
  - Predictive clustering
  - Conditional clustering
  - IBM-style clustering
- How to get clusters
  - Be clever or it takes forever!

SA1-76

Predictive clustering

- Let "z" be a word, "Z" be its cluster
- One cluster per word: hard clustering
  - WEEKDAY = Sunday, Monday, Tuesday, ...
  - MONTH = January, February, April, May, June, ...
- P(z|xy) = P(Z|xy) × P(z|xyZ)
- P(Tuesday | party on) = P(WEEKDAY | party on) × P(Tuesday | party on WEEKDAY)
- $P_{smooth}(z|xy) \approx P_{smooth}(Z|xy) \times P_{smooth}(z|xyZ)$

SA1-77

Predictive clustering example

- Find P(Tuesday | party on)
  - P_{smooth}(WEEKDAY | party on) × P_{smooth}(Tuesday | party on WEEKDAY)
  - C(party on Tuesday) = 0
  - C(party on Wednesday) = 10
  - C(arriving on Tuesday) = 10
  - C(on Tuesday) = 100
- P_{smooth}(WEEKDAY | party on) is high
- P_{smooth}(Tuesday | party on WEEKDAY) backs off to P_{smooth}(Tuesday | on WEEKDAY)

SA1-78

Conditional clustering

- P(z|xy) ≈ P(z|XY)
- P(Tuesday | party on) ≈
  P(Tuesday | EVENT PREPOSITION)
- $P_{smooth}(z|xy) \approx P_{smooth}(z|XY)$
  - λP_{ML} (Tuesday | EVENT PREPOSITION) +
    μP_{ML} (Tuesday | PREPOSITION) +
    $(1-\lambda-\mu) P_{ML}$ (Tuesday)

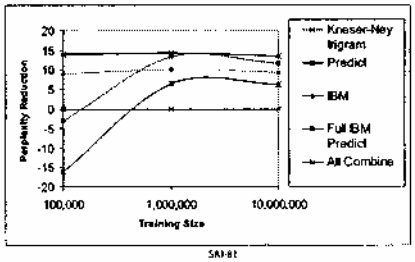
SA1-79

IBM Clustering

- $P(z|xy) \approx P_{smooth}(Z|XY) \times P(z|Z)$
- P(WEEKDAY|EVENT PREPOSITION) × P(Tuesday | WEEKDAY)
- Small, very smooth, mediocre perplexity
- $P(z|xy) \approx$
  $\lambda P_{smooth}(z|xy) + (1-\lambda) P_{smooth}(Z|XY) \times P(z|Z)$
- Bigger, better than no clusters

SA1-80

Cluster Results



SA1-81

Clustering: how to get them

- Build them by hand
  - Works ok when almost no data
- Part of Speech (POS) tags
  - Tends not to work as well as automatic
- Automatic Clustering
  - Swap words between clusters to minimize perplexity

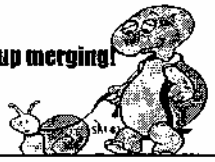
Clustering: automatic

- Minimize perplexity of P(z|Y)

Mathematical tricks speed it up

Use top-down splitting,

not bottom up merging!



Two actual WSJ classes

- MONDAYS
- FRIDAYS
- THURSDAY
- MONDAY
- EURODOLLARS
- SATURDAY
- WEDNESDAY
- FRIDAY
- TENTERHOOKS
- TUESDAY
- SUNDAY
- CONDITION

- PARTY
- FESCO
- CULT
- NILSON
- PETA
- CAMPAIGN
- WESTPAC
- FORCE
- CONRAN
- DEPARTMENT
- PENN
- GUILD

Clustering for the Channel Model

- Predictive clustering for channel
  - P(WEEKDAY→WEEKDAY')×
    P(Tuesday→Tuesday' |WEEKDAY→WEEKDAY')
- Conditional clustering for channel
  - P(to→to' | word before = LOCATIVE-VERB, go)
- Combine predictive and conditional
  - P(PARTY→EVENT'| word before = CELEBRATORY, birthday) × P(party→event'|EVENT')
  - P(PARTY→GROUP'(word before = POLITICAL, democratic) × P(party→group'|GROUP')

SA1-45

Clustering for distortion model

- Most statistical MT systems have "distortion model"
  - How far to move word
- French adjectives usually come after the noun
  - P(distort=3|ADJECTIVE) vs. P(distort=-2|NOUN)

SA1-46

Sentence Mixture Models

- Lots of different sentence types:
  - Numbers (The Dow rose one hundred seventy three points)
  - Quotations (Officials said "quote we deny all wrong doing "quote)
  - Mergers (AOL and Time Warner, in an attempt to control the media and the Internet, will merge)
- Model each sentence type separately

SA1-47

Sentence Mixture Models

- Roll a die to pick sentence type, s_k with probability λ_k
- Probability of sentence, given s_k

$$\prod_{i=1}^{n} P(w_i \mid w_{i-2} w_{i-1} s_k)$$

- Probability of sentence across types:

$$\sum_{k=1}^{m} \lambda_k \prod_{i=1}^{n} P(w_i \mid w_{i-2} w_{i-1} s_k)$$

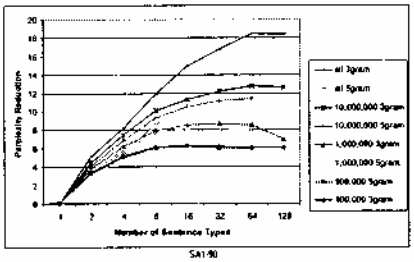
Sentence Model Smoothing

- Each topic model is smoothed with overall model.
- Sentence mixture model is smoothed with overall model (sentence type 0).

$$\sum_{k=0}^{m} \lambda_k \prod_{i=1}^{n} \begin{bmatrix} \mu_k P(w_i \mid w_{i-2} w_{i-1} s_k) \\ +(1-\mu_k) P(w_i \mid w_{i-2} w_{i-1}) \end{bmatrix}$$

SA1-48

Sentence Mixture Results



SA1-90

Sentence Clustering

- Same algorithm as word clustering
- Assign each sentence to a type, s_N
- Minimize perplexity of $P(z|s_N)$ instead of $P(z|Y)$

SAI-91

Topic Examples - 0 (Mergers and acquisitions)

- JOHN BLAIR &AMPERSAND COMPANY IS CLOSE TO AN AGREEMENT TO SELL ITS T. V. STATION ADVERTISING REPRESENTATION OPERATION AND PROGRAM PRODUCTION UNIT TO AN INVESTOR GROUP LED BY JAMES H. ROSENFIELD ,COMMA A FORMER C. B. S. INCORPORATED EXECUTIVE ,COMMA INDUSTRY SOURCES SAID .PERIOD
- INDUSTRY SOURCES PUT THE VALUE OF THE PROPOSED ACQUISITION AT MORE THAN ONE HUNDRED MILLION DOLLARS .PERIOD
- JOHN BLAIR WAS ACQUIRED LAST YEAR BY RELIANCE CAPITAL GROUP INCORPORATED ,COMMA WHICH HAS BEEN DIVESTING ITSELF OF JOHN BLAIR'S MAJOR ASSETS .PERIOD
- JOHN BLAIR REPRESENTS ABOUT ONE HUNDRED THIRTY LOCAL TELEVISION STATIONS IN THE PLACEMENT OF NATIONAL AND OTHER ADVERTISING .PERIOD
- MR. ROSENFIELD STEPPED DOWN AS A SENIOR EXECUTIVE VICE PRESIDENT OF C. B. S. BROADCASTING IN DECEMBER NINETEEN EIGHTY FIVE UNDER A C. B. S. EARLY RETIREMENT PROGRAM .PERIOD

Topic Examples - 1 (production, promotions, commas)

- MR. DION ,COMMA EXPLAINING THE RECENT INCREASE IN THE STOCK PRICE ,COMMA SAID ,COMMA "DOUBLE-QUOTE OBVIOUSLY ,COMMA IT WOULD BE VERY ATTRACTIVE TO OUR COMPANY TO WORK WITH THESE PEOPLE .PERIOD
- BOTH MR. BRONFMAN AND MR. SIMON WILL REPORT TO DAVID G. SACKS ,COMMA PRESIDENT AND CHIEF OPERATING OFFICER OF SEAGRAM .PERIOD
- JOHN A. KROL WAS NAMED GROUP VICE PRESIDENT ,COMMA AGRICULTURE PRODUCTS DEPARTMENT ,COMMA OF THIS DIVERSIFIED CHEMICALS COMPANY ,COMMA SUCCEEDING DALE E. WOLF ,COMMA WHO WILL RETIRE MAY FIRST .PERIOD
- MR. KROL WAS FORMERLY VICE PRESIDENT IN THE AGRICULTURE PRODUCTS DEPARTMENT .PERIOD
- RAPESEED ,COMMA ALSO KNOWN AS CANOLA ,COMMA IS CANADA'S MAIN OILSEED CROP .PERIOD
- YALE E. KEY IS A WELL -HYPHEN SERVICE CONCERN .PERIOD

Topic Examples - 2 (Numbers)

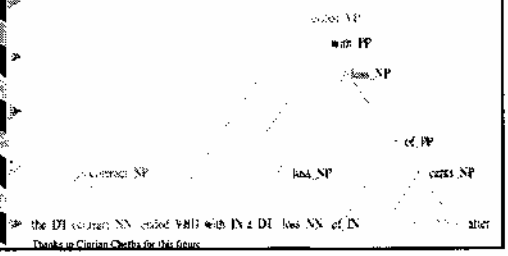
- SOUTH KOREA POSTED A SURPLUS ON ITS CURRENT ACCOUNT OF FOUR HUNDRED NINETEEN MILLION DOLLARS IN FEBRUARY ,COMMA IN CONTRAST TO A DEFICIT OF ONE HUNDRED TWELVE MILLION DOLLARS A YEAR EARLIER ,COMMA THE GOVERNMENT SAID .PERIOD
- THE CURRENT ACCOUNT COMPRISES TRADE IN GOODS AND SERVICES AND SOME UNILATERAL TRANSFERS .PERIOD
- COMMERCIAL -HYPHEN VEHICLE SALES IN ITALY ROSE ELEVEN .POINT FOUR %PERCENT IN FEBRUARY FROM A YEAR EARLIER ,COMMA TO EIGHT THOUSAND ,COMMA EIGHT HUNDRED FORTY EIGHT UNITS ,COMMA ACCORDING TO PROVISIONAL FIGURES FROM THE ITALIAN ASSOCIATION OF AUTO MAKERS .PERIOD
- INDUSTRIAL PRODUCTION IN ITALY DECLINED THREE .POINT FOUR %PERCENT IN JANUARY FROM A YEAR EARLIER ,COMMA THE GOVERNMENT SAID .PERIOD
- CANADIAN MANUFACTURERS' NEW ORDERS FELL TO TWENTY .POINT EIGHT ON BILLION DOLLARS (LEFT-PAREN CANADIAN RIGHT-PAREN IN JANUARY

Topic Examples – 3 (quotations)

- NEITHER MR. ROSENFIELD NOR OFFICIALS OF JOHN BLAIR COULD BE REACHED FOR COMMENT .PERIOD
- THE AGENCY SAID THERE IS "DOUBLE-QUOTE SOME INDICATION OF AN UPTURN "DOUBLE-QUOTE IN THE RECENT IRREGULAR PATTERN OF SHIPMENTS ,COMMA FOLLOWING THE GENERALLY DOWNWARD TREND RECORDED DURING THE FIRST HALF OF NINETEEN EIGHTY SIX .PERIOD
- THE COMPANY SAID IT ISN'T AWARE OF ANY TAKEOVER INTEREST .PERIOD
- THE SALE INCLUDES THE RIGHTS TO GERMAINE MONTEIL IN NORTH AND SOUTH AMERICA AND IN THE FAR EAST ,COMMA AS WELL AS THE WORLDWIDE RIGHTS TO THE DIANE VON FURSTENBERG COSMETICS AND FRAGRANCE LINES AND U. S. DISTRIBUTION RIGHTS TO LANCASTER BEAUTY PRODUCTS .PERIOD
- BUT THE COMPANY WOULDN'T ELABORATE .PERIOD
- HEARST CORPORATION WOULDN'T COMMENT ,COMMA AND MR. GOLDSMITH COULDN'T BE REACHED .PERIOD
- A MERRILL LYNCH SPOKESMAN CALLED THE REVISED QUOTRON AGREEMENT

Structured Language Model

"The contract ended with a loss of 7 cents after"



Thanks to Ciprian Chelba for this figure

How to get structure data?

- Use a Treebank (a collection of sentences with structure hand annotated) like Wall Street Journal, Penn Tree Bank.
- Problem: need a treebank.
- Or – use a treebank (WSJ) to train a parser; then parse new training data (e.g. Broadcast News)
- Re-estimate parameters to get lower perplexity models.

SAI-97

Parsing vs. Trigram Eugene Charniaks's Experiments

Model	Perplexity
Trigram poor smoothing	167
Trigram deleted-interpolation	155
Trigram Kneser-Ney	145
Parsing	119

18%

All experiments are trained on one million words of Penn tree-bank data, and tested on 80,000 words.

SAI-98

Thanks to Eugene Charniak for this slide

Structured Language Models

- Promising results
- But: time consuming: language is right branching; 5-grams, skipping, capture similar information.
- Interesting applications to parsing
  - Combines nicely with parsing MT systems

SAI-99

MT and Parsing Language Models

$$\arg\max_e p(e \mid f) = \arg\max_e p(e)p(f \mid e)$$

$$= \arg\max_e \sum_{\pi} p(e,\pi)p(f \mid e,\pi)$$

$$= \arg\max_e \sum_{\pi(e)} p(\pi)p(f \mid \pi)$$

$$\approx \arg\max_e p(\pi)p(f \mid \pi)$$

Puts the fragments together into a parse.

Finds tree fragments that match the French.

Thanks to Eugene Charniak for this slide

Some Successes (Eugene Charniak recent results)

Correct: This is not possible.

Trigram: Impossibility.

Parser: This is impossible.

Correct: This is the globalization of production.

Trigram: This globalization of production.

Parser: This is globalization of production.

Thanks to Eugene Charniak for this slide

Some Less than Successes (Eugene Charniak recent results)

Correct: He said he often eats Chinese dishes.

Trigram: He said China frequently tastes food.

Parser: He said recurrent taste of Chinese cuisine.

Correct: Wishful thinking out of touch with reality.

Trigram: Divorce practical delusion.

Parser: Practical delusion divorced.

Thanks to Eugene Charniak for this slide

Using Complex Language Models

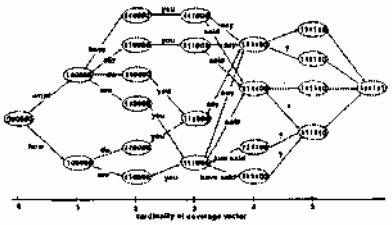
- For speech recognition, complex models often interact poorly with dynamic programming
  - Showed how bad 5-grams are for speech DP
  - Skipping, sentence mixtures have same problems
  - Clusters interact badly with phonetic trees
- Many MT algorithms don't have these limitations! (But some do.)

N-best lists

- Make list of 100 best translation hypotheses using simple bigram or trigram
- Rescore using any model you want
  - Cheaply apply complex models
  - Perform Source research separately from Channel
- For long, complex sentences, need exponentially many more hypotheses

SA1-104

Lattices for MT
Compact version of n-best list



From Ueffing, Och and Ney, EMNLP '02

Lattices in MT versus Speech

- Lattices or n-best rescoring causes "latency" in speech recognizer
  - Second pass over lattice doesn't start until first pass is finished
  - Can change results of incremental recognition
  - Rarely used in speech products (more common in research)
- Not a problem for MT!
  - Complex LMs better for MT systems than for speech systems.

Tools: CMU Language Modeling Toolkit

- Can handle bigram, trigrams, more
- Can handle different smoothing schemes
- Many separate tools -- output of one tool is input to next: easy to use
- Free for research purposes
- http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html

Tools: SRI Language Modeling Toolkit

- More powerful than CMU toolkit
- Can handles clusters, lattices, n-best lists, hidden tags
- Free for research use
- http://www.speech.sri.com/projects/srilm

Small enough

- Real language models are often huge
- 5-gram models typically larger than the training data
- Use count-cutoffs (eliminate parameters with fewer counts) or, better
- Use Stolcke pruning – finds counts that contribute least to perplexity reduction,
  - P(City | New York) ≈ P(City | York)
  - P(Friday | God it's) ≠ P(Friday | It's)
- Remember, Kneser-Ney helped most when lots of 1 counts

Combining Data

- Often, you have some "in domain" data and some "out of domain data"
- Example: Microsoft is working on translating computer manuals
  - Only about 3 million words of Brazilian computer manuals
- Can combine computer manual data with hundreds of millions of words of other data
  - Newspapers, web, encyclopedias,usenet...

SA1-110

How to combine

- Just concatenate – add them all together
  - Bad idea – need to weight the "in domain" data more heavily
- Take out of domain data and multiple copies of in domain data (weight the counts)
  - Bad idea – doesn't work well, and messes up most smoothing techniques

SA1-111

How to combine

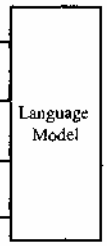
- A good way: take weighted average, e.g.

$\lambda P_{manuals}(z|xy) + \mu P_{web}(z|xy) + (1-\lambda-\mu) P_{newspaper}(z|xy)$

- Can apply to channel models too (e.g. combine Hansard with computer manuals for French translation)
- Lots of research in other techniques
  - Maxent-inspired models, non-linear interpolation (log domain), cluster models, etc. Minimal improvement (but see work by Rukmini Iyer)

Other Language Model Uses

- Handwriting Recognition
  - P(observed ink|words) × P(words)
- Telephone Keypad Input
  - P(numbers|words) × P(words)
- Spelling Correction
  - P(observed keys|words) × P(words)
- Chinese/Japanese text entry
  - P(phonetic representation|characters) × P(characters)



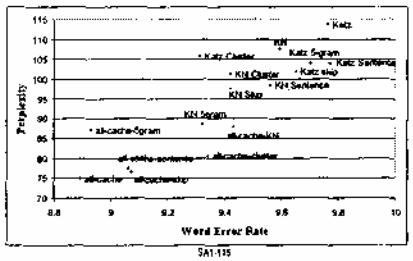
SA1-113

Some Experiments

- I re-implemented almost all techniques
- Trained on 260,000,000 words of WSJ
- Optimize parameters on heldout
- Test on separate test section
- Some combinations extremely time-consuming (days of CPU time)
  - Don't try this at home, or in anything you want to ship
- Rescored N-best lists to get results
  - Maximum possible improvement from 10% word error rate absolute to 5%

SA1-114

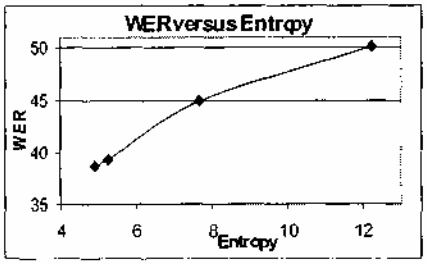
Overall Results: Perplexity



Franz Och's Results

Model type	PP	WER	PER	mWER	BLEU
zerogram	4781	50.1	38.1	45.9	29
Unigram	203.1	45	30.2	40.9	37.7
Bigram	38.3	39.3	26.9	32.9	53
Trigram	29.9	38.7	26.8	31.8	55.2
Trigram+ CLM		37.7	26.5	30.9	56.1

Franz Och's Results



Shannon Revisited

Char n-gram	Low Char	Upper char	Low word	Upper word
1	9.1	16.3	191,237	4,702,511
5	3.2	6.5	653	29,532
10	2.0	4.3	45	2,998
15	2.3	4.3	97	2,998
100	1.5	2.5	10	142

- People can make GREAT use of long context
- With 100 characters, computers get very roughly 50% word perplexity reduction.

SA1-118

Conclusions

- Machine translation can use much more complex techniques than speech recognition
- Lots of fun techniques – caching, sentence mixture models – that have not been applied to MT yet, and others, like clustering, that have.
- Parsing LMs are a promising technique.
- Can apply LM ideas and techniques to channel model too.

SA1-119

More Resources

- Joshua's web page: www.research.microsoft.com/~joshuago
  - Smoothing technical report: good introduction to smoothing and lots of details too.
  - "A Bit of Progress in Language Modeling," which is the journal version of much of this talk.
  - Papers on fuzzy keyboard, language model compression, and maximum entropy.
  - Clustering tool

SA1-120

More Resources: Language Models for MT

- *Very little research on this*
  - H. Sawaf, K. Schütz, H. Ney. "On the Use of Grammar Based Language Models for Statistical Machine Translation", IWPT '00
  - Franz Josef Och and H. Ney "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation", ACL '02
  - Franz Josef Och's thesis, Statistical Machine Translation: From Single-Word Models to Alignment Templates (in preparation)

SA1-121

Lattices for MT

- *For n-gram style models*
  - Nicola Ueffing, Franz Josef Och and Hermann Ney. "Generation of Word Graphs in Statistical Machine Translation", EMNLP '02
- *For parsing style models*
  - Irene Langkilde. Forest-based Statistical Sentence Generation, ACL '00

SA1-122

More Resources

- Eugene Charniaks's web page: http://www.cs.brown.edu/people/ec
  - Papers on statistical parsing for it's own sake and for language modeling, as well as using language modeling to measure contextual influence.
  - Pointers to software for statistical parsing as well as statistical parsers optimized for language-modeling

SA1-123

More Resources: Books

- Books (all are OK, none focus on language models)
  - Statistical Language Learning by Eugene Charniak
  - Speech and Language Processing by Dan Jurafsky and Jim Martin (especially Chapter 6)
  - Foundations of Statistical Natural Language Processing by Chris Manning and Hinrich Schütze.
  - Statistical Methods for Speech Recognition, by Frederick Jelinek
  - Spoken Language Processing by Huang, Acero and Hon

More Resources

- Sentence Mixture Models: (also, caching)
  - Rukmini Iyer. EE Ph.D. Thesis, 1998 "Improving and predicting performance of statistical language models in sparse domains"
  - Rukmini Iyer and Mari Ostendorf. Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Acoustics, Speech and Audio Processing* 7:30–39, January 1999.
- Caching: Above, plus
  - R. Kuhn. Speech recognition and the frequency of recently used words: A modified markov model for natural language. In *12th International Conference on Computational Linguistics*, pages 348–350, Budapest, August 1988.
  - R. Kuhn and R. D. Mori. A cache-based natural language model for speech reproduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(6):570–583, 1990.
  - R. Kuhn and R. D. Mori. Correction to a cache-based natural language model for speech reproduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(6):691–692, 1992.

More Resources: Clustering

- The seminal reference
  - P. F. Brown, V. J. DellaPietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December 1992.
- Two-sided clustering
  - H. Yamamoto and Y. Sagisaka. Multi-class composite n-gram based on connection direction. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, May 1999.
- Fast clustering
  - D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *SIGIR 92*, 1992.
- Other:
  - R. Kneser and H. Ney. Improved clustering techniques for class-based statistical language modelling. In Eurospeech 93, volume 2, pages 973–976, 1993.

More Resources

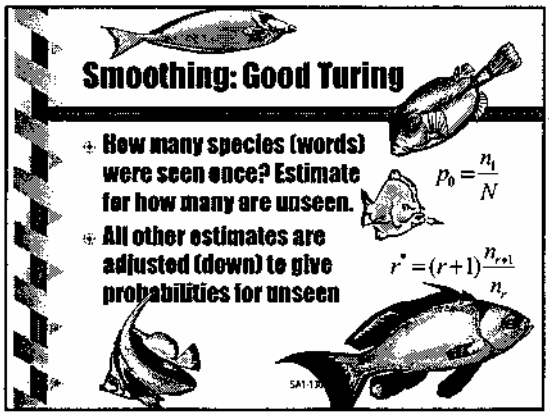
- Structured Language Models
  - Eugene's web page
  - Ciprian Chelba's web page:
    - http://www.clsp.jhu.edu/people/chelba/
- Maximum Entropy
  - Roni Rosenfeld's home page and thesis http://www.cs.cmu.edu/~roni/
  - Joshua's web page
- Stolcke Pruning
  - A. Stolcke (1998), Entropy-based pruning of backoff language models. *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp. 270-274, Lansdowne, VA. NOTE: get corrected version from http://www.speech.sri.com/people/stolcke

More Resources: Skipping

- Skipping:
  - X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The SPHINX-II speech recognition system: An overview. *Computer, Speech, and Language*, 2:137--148, 1993.
- Lots of stuff:
  - S. Martin, C. Hamacher, J. Liermann, F. Wessel, and H. Ney. Assessment of smoothing methods and complex stochastic language modeling. In *6th European Conference on Speech Communication and Technology*, volume 5, pages 1939--1942, Budapest, Hungary, September 1999. H. Ney, U. Essen, and R. Kneser.
  - On structuring probabilistic dependences in stochastic language modelling. *Computer, Speech, and Language*, 8:1--38, 1994.

Smoothing: Good Turing

- Imagine you are fishing
- You have caught 10 Carp, 3 Cod, 2 tuna, 1 trout, 1 salmon, 1 eel.
- How likely is it that next species is new? 3/18
- How likely is it that next is tuna? Less than 2/18



Smoothing: Good Turing Example

- 10 Carp, 3 Cod, 2 tuna, 1 trout, 1 salmon, 1 eel.
- How likely is new data (p_0). Let n_1 be number occurring once (3), N be total (18). p_0=3/18

$$p_0 = \frac{n_1}{N}$$

- How likely is eel? 1^*
- n_1=3, n_2=1
- 1^*=2×1/3 = 2/3
- P(eel) = 1^*/N = (2/3)/18 = 1/27

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

Smoothing: Katz

- Use Good-Turing estimate

$$P_{Katz}(z \mid xy) = \begin{cases} \frac{C^*(xyz)}{C(xy)} & \text{if } C(xyz) > 0 \\ \alpha(xy) P_{Katz}(z \mid y) & \text{otherwise} \end{cases}$$

- Works pretty well.
- Not good for 1 counts
- α is calculated so probabilities sum to 1

SA1-132

Smoothing:Absolute Discounting

- Assume fixed discount

$$P_{absolute}(z \mid xy) = \begin{cases} \dfrac{C(xyz) - D}{C(xy)} & \text{if } C(xyz) > 0 \\ \alpha(xy) P_{absolute}(z \mid y) & \text{otherwise} \end{cases}$$

- Works pretty well, easier than Katz.
- Not so good for 1 counts

SA1-131

Smoothing: Interpolated Absolute Discount

- Backoff: ignore bigram if have trigram

$$P_{absolute}(z \mid xy) = \begin{cases} \dfrac{C(xyz) - D}{C(xy)} & \text{if } C(xyz) > 0 \\ \alpha(xy) P_{absolute}(z \mid y) & \text{otherwise} \end{cases}$$

- Interpolated: always combine bigram, trigram

$$P_{abs\text{-}interp}(z \mid xy) = \frac{C(xyz) - D}{C(xy)} + \beta(xy) P_{abs\text{-}interp}(z \mid x)$$