Natural Language Technology(*)

Lillian Lee

Cornell University

http://www.cs.cornell.edu/home/llee

(*) Some of this material comes from a joint tutorial, co-organized with John Lafferty, at the Sixteenth National Conference on Artificial Intelligence, 1999.

Outline

I. Overview of the Field

II. The Statistical Revolution

III. Language as a Statistical Source

IV. Tools of the Trade

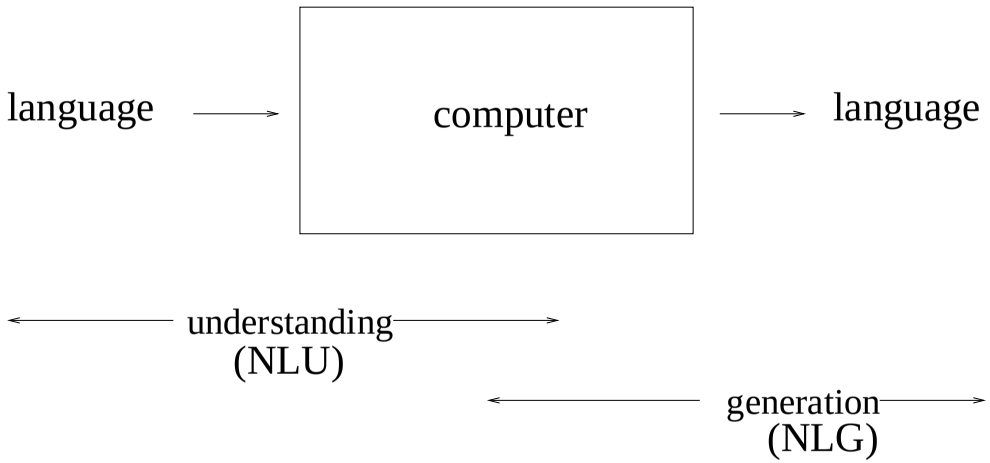
V. The Sparse Data Problem

VI. Conclusions and References

I. Overview of the Field

Natural Language Processing (NLP)

Goal: computers using natural language as input and/or output



NLU example: convert an utterance into a sequence of computer instructions.

NLG example: produce a summary of a patient's records.

Why NLP?

Lots of information is in natural language format.

- Documents
- News broadcasts
- User utterances

Lots of users want to communicate in natural language.

- “Do what I mean!”

NLP is Useful

Task	Input	Output
summarization	“document(s)” (CNN broadcasts)	summary
machine translation	signal in language 1	signal in language 2
question answering	query	answer to query
$\longrightarrow$ information retrieval	query	relevant documents
user interfaces	command in natural language	computer instructions

“Now we’re betting the company on these natural interface technologies”

– Bill Gates, 1997

NLP is Cross-Disciplinary

Excellent opportunities for interdisciplinary work.

- **Linguistics**: models of language
  - ▷ emphasizes 100% accuracy (*competence*)
- **Psychology**: models of cognitive processes
  - ▷ emphasizes biological/cognitive plausibility
- **Mathematics and statistics**: properties of models
  - ▷ emphasizes formal aspects

On the whole, NLP tends to be applications-oriented: 95% is OK; models need be neither biologically plausible nor mathematically satisifying.

NLP is Challenging

It is often said that NLP is "AI-complete":

> All the difficult problems in artificial intelligence manifest themselves in NLP problems.

This idea dates back at least to the Turing Test:

> "The question and answer method seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include" [Turing, "Computing Machinery and Intelligence", 1950]

Why is NLP hard?

- “Doesn’t Microsoft do that already?”
- Ad from the 70’s or 80’s (source: S. Shieber): the problem has already been solved ...

  “At last, a computer that understands you like your mother”

Ambiguity

"At last, a computer that understands you like your mother"

What can we infer about the computer?

1. (*) It understands you as well as your mother understands you
2. It understands (that) you like your mother
3. It understands you as well as it understands your mother

1 and 3: Does this mean well, or poorly?

Ambiguity at Many Levels (I)

At the acoustic level (*speech recognition*):

1. "... a computer that understands you like your mother"
2. "... a computer that understands your lie cured mother"

Ambiguity at Many Levels (II)

At the morphological (word-form) level:

"... a computer that understands you like ..."

= understand + s

$\neq$ under + stands (although derived historically from this)

$\neq$ un + derstands

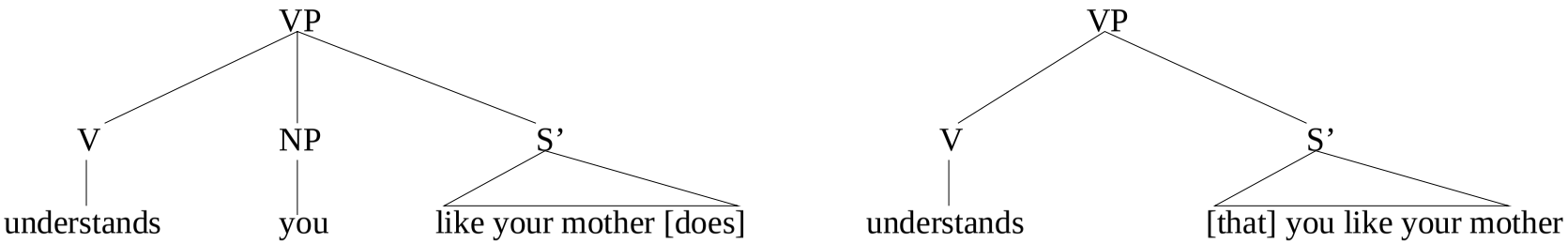
In practice, storing root forms reduces database size.

The morphological analysis problem is especially difficult in new domains:

unionized = ?

Ambiguity at Many Levels (III)

At the syntactic (structural) level:



Different structures lead to different interpretations.

Ambiguity at Many Levels (IV)

In fact, the identity of the syntactic material is ambiguous.

Ellipsis: missing (elided) syntactic material

"... a computer that understands you like ..." = ?

... understands you like your mother [understands you]

... understands you like [it understands] your mother

Ambiguity at Many Levels (V)

At the semantic (meaning) level:

mother = ? (OED)

A female parent

A cask or vat used in vinegar-making

This is an instance of word sense ambiguity.

A more typical example: “They put money in the bank”.

Ambiguity at Many Levels (VI)

At the discourse (multiple-clause) level:

1. Alice says they've built a computer that understands you like your mother.
2. But she ...

    2a. ... doesn't know any details.

    2b. ... doesn't understand me at all.

This is an instance of anaphora, where "she" co-refers to some other discourse entity.

What Will It Take?

The task seems so difficult! What resources do we need?

1. Knowledge about language
2. Knowledge about the world

Two veins of work to combat the knowledge acquisition bottleneck:

- handcrafted and expert-driven
- automated and data-driven

It often helps to restrict the domain.

Success Stories

Not an exclusive list!

The TAUM-METEO system [Chandioux 76]: essentially perfect French-English translation of weather reports.

JUPITER [MIT Spoken Language Systems group (1-888-573-TALK)]: conversational system for weather information. $\sim$80% "understanding" rate for novices.

Information Extraction: systems exist for analyzing and summarizing reports of joint business ventures [Message Understanding Conferences (MUC) 1994]

Note – restricted domains.

General NLP References

Not an exclusive list!

- Jurafsky and Martin, 2000. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*
- Allen, 1995. *Natural Language Understanding*, 2nd edition.
- `comp.ai.nat-lang` FAQ (Frequently Asked Questions). `http://www.cs.columbia.edu/~radev/nlpfaq.txt`
- The Association for Computational Linguistics (ACL) Universe, `http://www.cs.columbia.edu/~radev/u/db/acl/`

II. The Statistical Revolution

What is Statistical NLP?

Goal: Infer language properties from (annotated?) (text?) samples.

- Helps ease the knowledge acquisition bottleneck.

Draws on probability, statistics, information theory, machine learning.

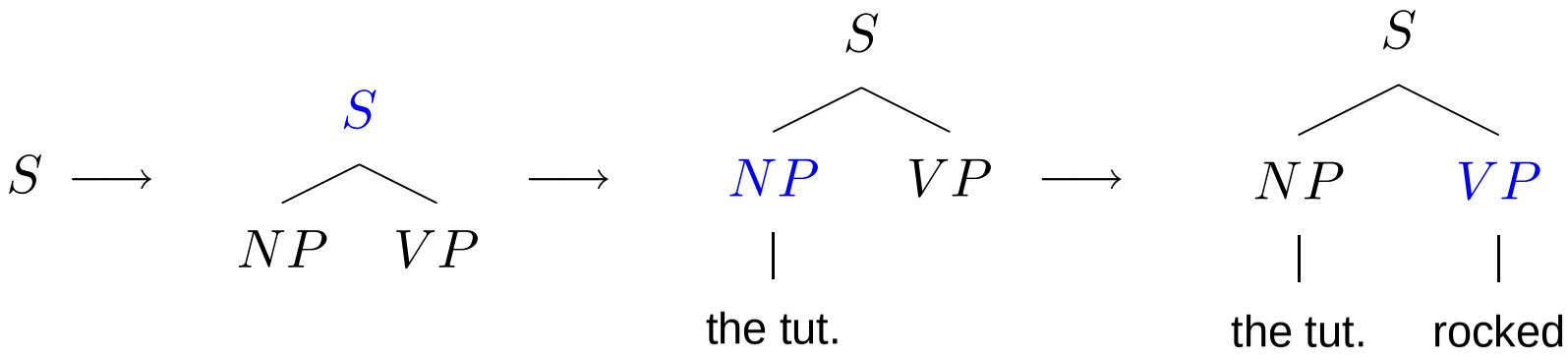
Two threads (often intertwined; not everyone distinguishes!):

- statistical *models* — language assumed generated by a statistical source
- statistical *methods* — no assumption on language source; sample statistics used to make decisions

Non-statistical Models Example: CFG's

Context-Free Grammars (CFG's): strings generated by choosing some sequence of rewriting rules.

S	$\rightarrow$	$NP\ VP$	VP	$\rightarrow$	rocked
NP	$\rightarrow$	the tutorial	VP	$\rightarrow$	bombed



Statistical Models Example: PCFG's

Probabilistic Context-Free Grammars (PCFG's): strings generated by randomly picking rules according to their probabilities

(1.0)	S	$\rightarrow$	$NP\ VP$	(.75)	VP	$\rightarrow$	rocked
(1.0)	NP	$\rightarrow$	the tutorial	(.25)	VP	$\rightarrow$	bombed

$$
\begin{aligned}
P(\text{“the tut. rocked”}) &= \underbrace{1}_{S \rightarrow NP\ VP} \times \underbrace{1}_{NP \rightarrow \text{the tut.}} \times \underbrace{.75}_{VP \rightarrow \text{rocked}} \\
&= .75
\end{aligned}
$$

Statistical Methods Example: WSD

Word sense disambiguation (WSD): find correct word sense from context

"They put money in the $\underbrace{\text{bank}}_{\text{savings? river?}}$ "

A statistical solution [Lesk 86]: estimate the likelihood of $\langle$savings bank$\rangle$ co-occurring with "money" from entries in a *machine-readable dictionary*

Why Statistical NLP?

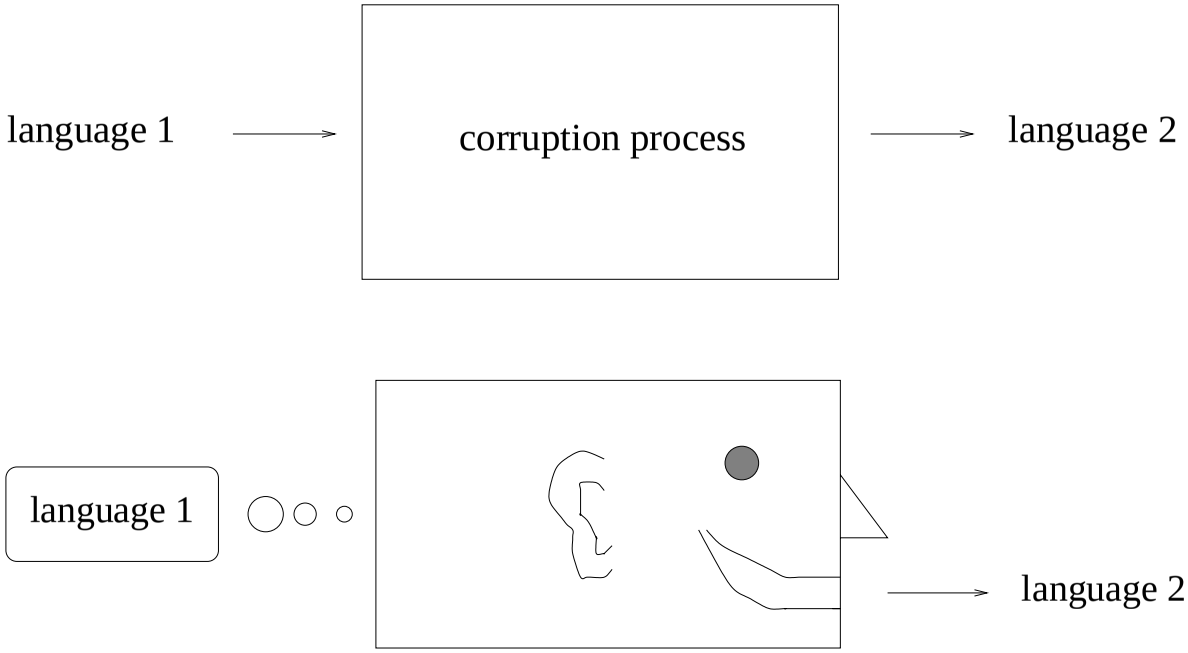
- Statistical models allow degrees of uncertainty (not just "grammatical/ungrammatical")
  - ▷ confidence can be assessed (helps combine knowledge sources)
  - ▷ models can be iteratively trained/updated
- Statistical methods reduce the *knowledge acquisition bottleneck*
  - ▷ transfer to new domains is easier

But statistical approaches were (are) not universally accepted ...

A Brief History

The 40's and 50's: statistical NLP popular

- Harris, Firth: empirical linguistics ("You shall know a word by the company it keeps" [Firth 57])
- Shannon, Weaver: cryptographic notions, the *noisy channel model*



A Brief History (cont.)

Late 50's–80's: statistical NLP in disfavor

> "It is fair to assume that neither sentence
>
> (1) *Colorless green ideas sleep furiously*
>
> nor
>
> (2) *Furiously sleep ideas green colorless*
>
> ... has ever occurred .... Hence, in any statistical model ... these sentences will be ruled out on identical grounds as equally "remote" from English. Yet (1), though nonsensical, is grammatical, while (2) is not." [Chomsky 1957]

A Brief History (cont.)

The 80's – present: statistical NLP once again mainstream

- revived by IBM: influenced by speech recognition
- confluence with interest in machine learning
- nowadays,

  "no one can profess to be a computational linguist without a passing knowledge of statistical methods .... anyone who cannot at least use the terminology persuasively risks being mistaken for kitchen help at the ACL [conference] banquet." [Abney 97]

Statistics on Statistical NLP

From Julia Hirschberg's AAAI-98 invited talk:

Source	Percentage of statistically-based papers
ACL 1990	12.8%
ACL 1998	63.5

(ACL is the main conference of the Association for Computational Linguistics)

1983 was the last year in which there were no such papers.

The “Opposite” of Statistical NLP?

Some draw contrasts with knowledge-based methods, higher-level processes, linguistics...

- Chomsky
- “I don’t believe in this statistics stuff”
- “that’s not learning, that’s statistics”

- “AI-NLP ...is going nowhere fast”
- “Every time I fire a linguist, my performance goes up”

Statistics Complements Other Approaches

- Knowledge-based models can be converted to stochastic versions
  - ▷ CFG's $\longrightarrow$ PCFG's
  - ▷ statistical semantics, discourse models [Miller 96]
- Statistical methods can make use of knowledge bases (don't confuse methods and models)
  - ▷ WSD using dictionaries

III. Language as a Statistical Source

A Recent Anniversary

Famous First Words

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages."

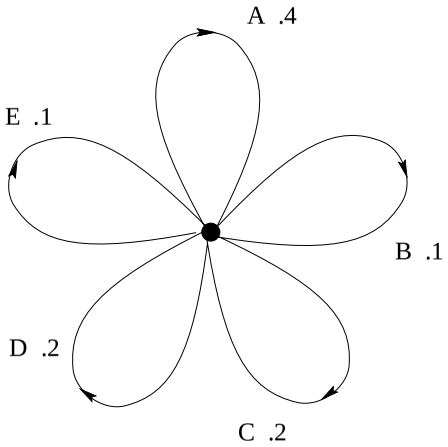
C.E. Shannon, A Mathematical Theory of Communication,
The Bell System Technical Journal, July 1948.

Generative Models

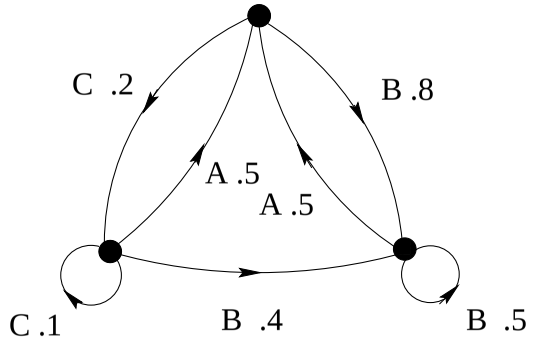
A useful conceptual and practical device: *coin-flipping models*

- A string is generated by a randomized algorithm
  - ▷ The generator can be in one of several “states”
  - ▷ A coin (or a bunch of coins) is flipped to choose the next state
  - ▷ Another coin is flipped to decide which letter or word to output
- Shannon: *“The states will correspond to the “residue of influence” from preceding letters”*

Coin-Flipping Models



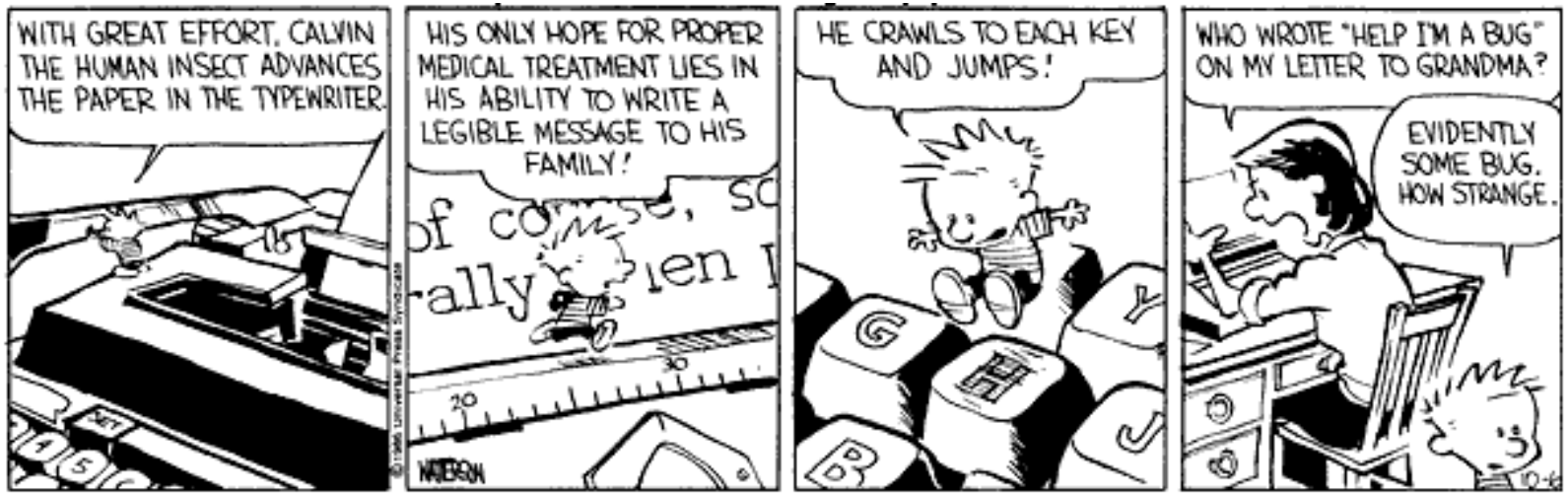
A A A C D C B D C E A A D A D A C E D A E A D C A B E D A D D C E C A A



A B B A B A B A B A B A B A B B B A B B B B B A B A B A B A B A B B A C

The Soul of a New Machine

When designing a new statistical model for an NLP task, it is often very helpful to simulate it in your mind.



Markov Approximations to English

From Shannon's original paper:

1. *Zero-order approximation:*

   XFOML RXKXRJFFUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD

2. *First-order approximation:*

   OCRO HLI RGWR NWIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH RBL

3. Second-order approximation:

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Markov Approximations (cont.)

From Shannon's original paper

4. *Third-order approximation:*

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTABIN IS REGOACTIONA OF CRE

Markov random field with 1000 "features," no underlying "machine" (Della Pietra et. al, 1997):

WAS REASER IN THERE TO WILL WAS BY HOMES THING BE RELOVERATED THER WHICH CONISTS AT FORES ANDITING WITH PROVERAL THE CHESTRAING FOR HAVE TO INTRALLY OF QUT DIVERAL THIS OFFECT INATEVER THIFER CONSTRANDED STATER VILL MENTTERING AND OF IN VERATE OF TO

Word-Based Approximations

1. *First-order approximation:*

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME
   CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO
   EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE T

2. *Second-order approximation:*

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER
   THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER
   METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD
   THE PROBLEM FOR AN UNEXPECTED

Shannon's comment:

"*It would be interesting if further approximations could be constructed, but the labor involved becomes enormous at the next stage.*"

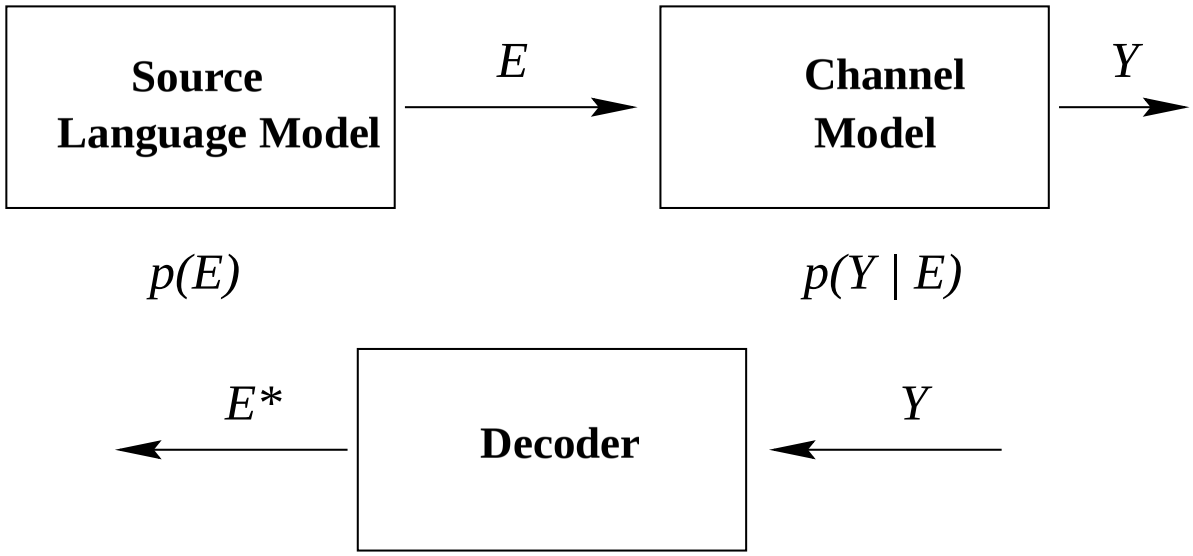
Estimating Redundancy

- Redundancy helps us communicate

  TH_R_ _S _NLY _N_ W_Y T_ F_LL _N TH_ V_W_LS _N TH_S S_NT_NC_

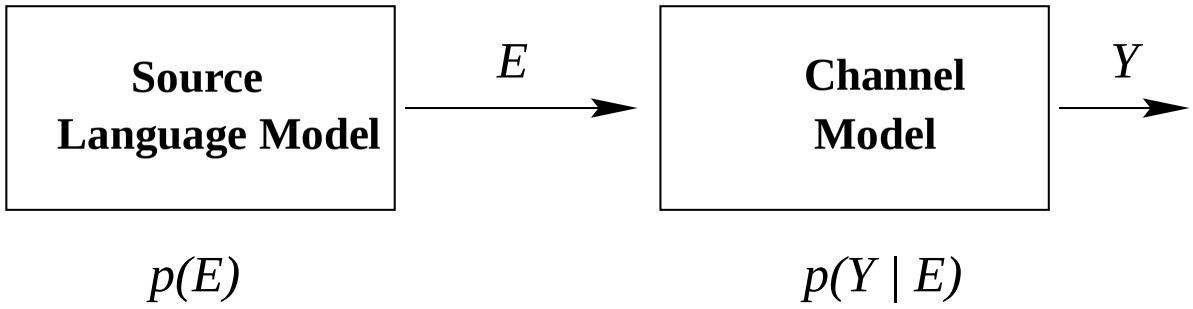
- Coin-flipping models can be used to estimate the redundancy or *entropy* of English:
  - ▷ First order model: 4.03 bits/letter
  - ▷ Fourth order model: 2.8 bits/letter
  - ▷ Trigram word model: 1.72 bits/letter

The Source-Channel Model



$$E^* = \arg\max_E p(E \mid Y) = \text{argmax } p(E)\ p(Y \mid E)$$

Source-Channel Examples



Application	*Observation* Y
speech recognition	acoustic signal
French translation	French text
spelling correction	typed text
OCR	processed image

IV. Tools of the Trade

- PCFG's
- HMM's
- EM
- Special cases

Predicting String Probabilities

“The tutorial was a roaring success”

vs.

“The tutorial was a boring address”

Which is more likely? (both are *grammatical*)

Language Modeling

Language model: method for assigning probabilities to strings; want to approximate *source* probabilities

$$P(\text{“ ... roaring success”}) = .003$$

$$P(\text{“ ... boring address”}) = .000001$$

Classic applications: speech, handwriting, and optical character recognition

Standard models: PCFG's, n-grams/HMM's

PCFG's

Prob.	Rule	Prob.	Rule
(1.0)	$S \rightarrow NP\,VP$	(.75)	$VP \rightarrow$ rocked
(1.0)	$NP \rightarrow$ the tutorial	(.25)	$VP \rightarrow$ bombed

$$P(\text{“the tut. rocked”}) = P\left(\begin{array}{c} S \\ \diagup\ \diagdown \\ NP \quad\quad VP \\ | \quad\quad\ | \\ \text{the tutorial} \quad \text{rocked} \end{array}\right)$$

$$= 1 \times 1 \times .75$$

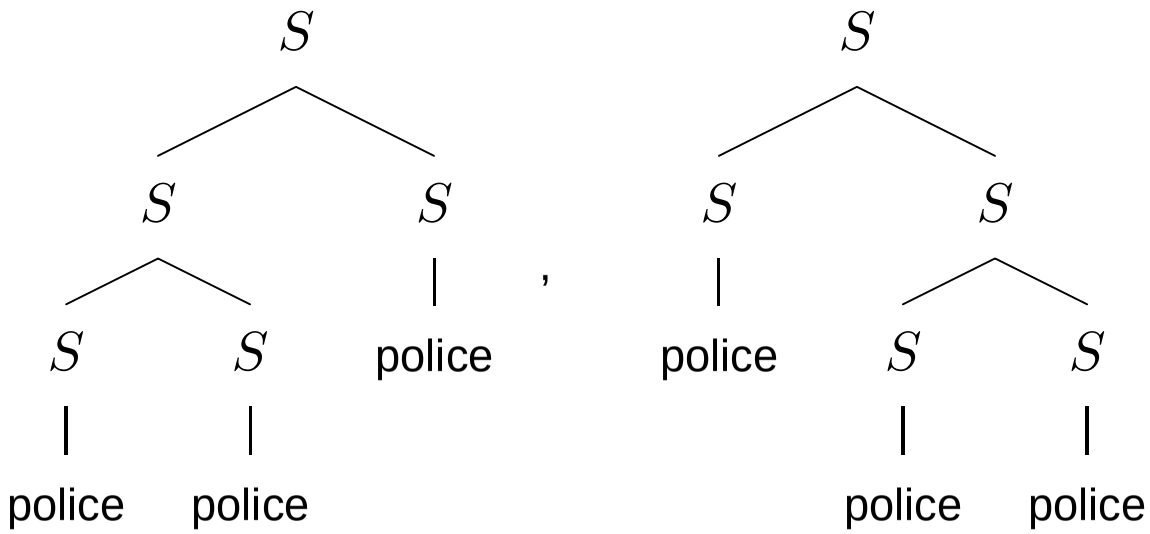
PCFG's give both parse structures and probability.

PCFG Facts

Probabilities of all rules with the same lefthand side must sum to one.

Ambiguous PCFG's: multiple parse trees for same sentence

.1	S	$\rightarrow$	SS
.9	S	$\rightarrow$	police



Police, Police police, Police police police, ...

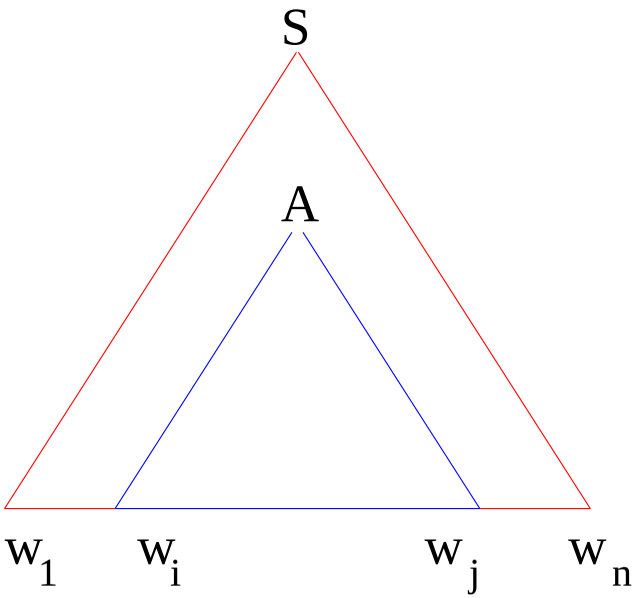
PCFG Language Modeling

For sentence $w_1 \cdots w_n$ and PCFG G,

$$P_G(w_1 \cdots w_n) = \sum_{\text{parse trees } \pi \text{ for } w_1 \cdots w_n} P(\pi)$$

How can we compute this efficiently? Dynamic programming!

Computing Sentence Probabilities



Inside probability $\mathrm{In}(A, i, j)$: probability that A generates string $w_i \cdots w_j$

Outside probability $\mathrm{Out}(A, i, j)$: probability that S generates $w_1 \cdots w_{i-1} A w_{j+1} \cdots w_n$

Combining inside probs bottom-up yields $\mathrm{In}(S, 1, n)$, prob of the sentence

Training PCFG's

Where do we get the rule probabilities?

Inside-Outside algorithm [Baker 79]:

- iterative re-estimation using inside and outside probabilities for large training corpus T
- each step increases T's likelihood according to the PCFG

Note: algorithms for taking advantage of structural annotation [Pereira and Schabes 92]

PCFG's: Overall Effectiveness

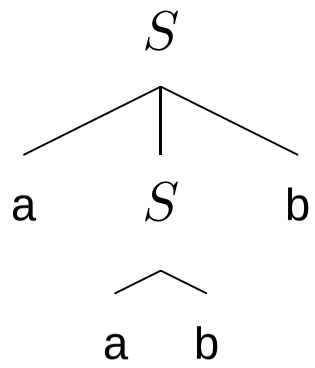
PCFG's: linguistically intuitive, provide parse structures

If don't need parses, simpler models more effective at estimating sentence probabilities (so far)

But cf. [Chelba and Jelinek 98]

Restricting Generative Capacity

CFG's: top-down generation. $S \rightarrow aSb$; $S \rightarrow ab$

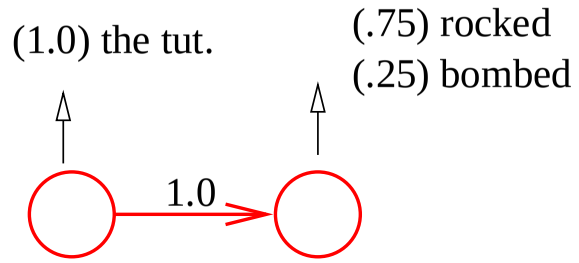


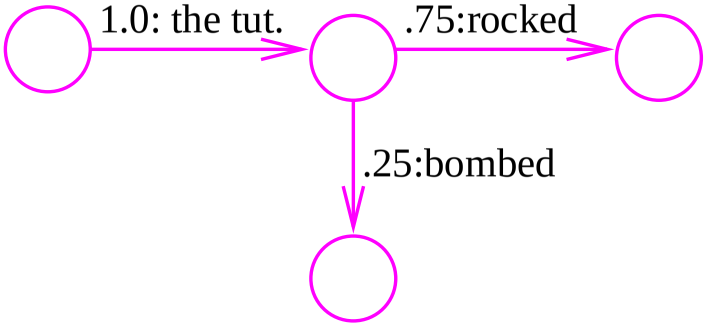
"Matching" a's and b's not necessarily adjacent.

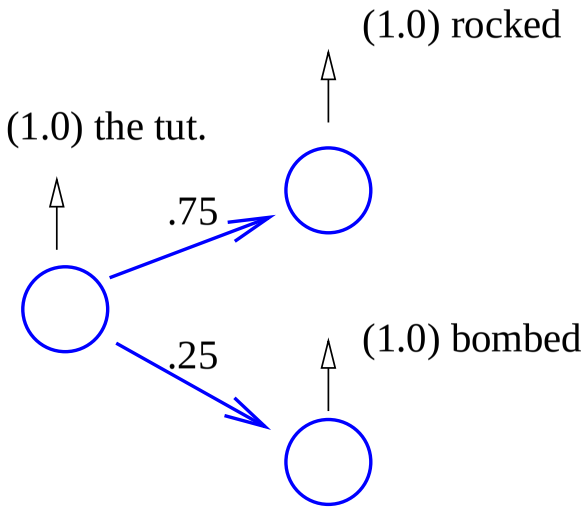
What if we limit modeling capacity to *local* correlations?

HMM's

Hidden Markov Models (HMM's): states, state transitions, outputs.







$$P(\text{“the tut. rocked”}) = 1 \times 1 \times .75 = 1 \times .75 \times 1 = 1 \times .75 = .75$$

HMM Facts

- Sum of all transition probs out of a state must sum to one. Same for output probs. (cf. PCFG rule probs)
- Same string may be generated via different *paths* (cf. ambiguous PCFG's)
- HMM's cannot simulate all PCFG's.
  - ▷ ex: $(.1)\ S \rightarrow aSb;\ \ (.9)\ S \rightarrow ab$

HMM Language Modeling

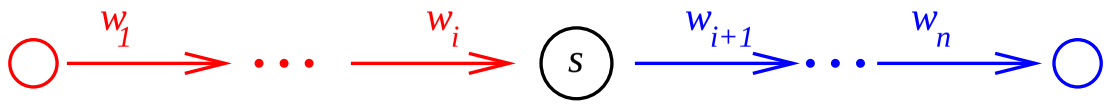
For sentence $w_1 \cdots w_n$ and HMM h (arbitrary states),

$$P_h(w_1 \cdots w_n) = \sum_{\text{paths } p \text{ for } w_1 \cdots w_n} P(w_1 \cdots w_n | p) P(p)$$

(cf. PCFG language modeling)

How can we compute this efficiently? Dynamic programming!

Computing Sentence Probabilities



Forward prob $\mathrm{For}(s, i)$: prob of generating $w_1 \cdots w_i$, ending at state s

Backward prob $\mathrm{Back}(s, i)$: prob of generating $w_{i+1} \cdots w_n$, starting at s

Combining forward probs left-to-right yields $\mathrm{For}(\text{start}, n)$, prob of the sentence

Alternatively, combining backward probs right-to-left yields $\mathrm{Back}(\text{start}, 0)$

Training HMM's

Where do we get transition/output probs?

Forward-Backward, or Baum-Welch:

- iterative re-estimation using forward and backward probabilities for large training corpus T
- each step increases T's likelihood according to the HMM

(cf. Inside-Outside)

Training Commonalities

Inside-Outside (PCFG's) and Forward-Backward (HMM's) look very similar.

In general, training a probabilistic model: find parameter settings maximizing (locally) training data likelihood

EM

The *EM* algorithm [Dempster, Laird, Rubin 77]: used when it is difficult to calculate likelihood directly.

Ex: Inside-Outside, Forward-Backward

θ: model parameters (e.g., transition probs)

T: training data

Goal: find

$$\theta^* = \arg\max_{\theta} P_\theta(T)$$

Settle for finding local stationary point via hillclimbing.

EM (cont.)

Use *auxiliary* variable Y, *dependent on* θ. If

$$\sum_y P_{\theta_i}(y|T)\log P_{\theta_{i+1}}(y,T) > \sum_y P_{\theta_i}(y|T)\log P_{\theta_i}(y,T) \quad (1)$$

then

$$P_{\theta_{i+1}}(T) > P_{\theta_i}(T)$$

(training likelihood increased!)

EM algorithm

Iterative process:

Expectation: calculate $E(\log_\theta(Y,T))$, with respect to $P_{\theta_i}(y|T)$, as a function of θ

Maximization: find θ_{i+1} maximizing this

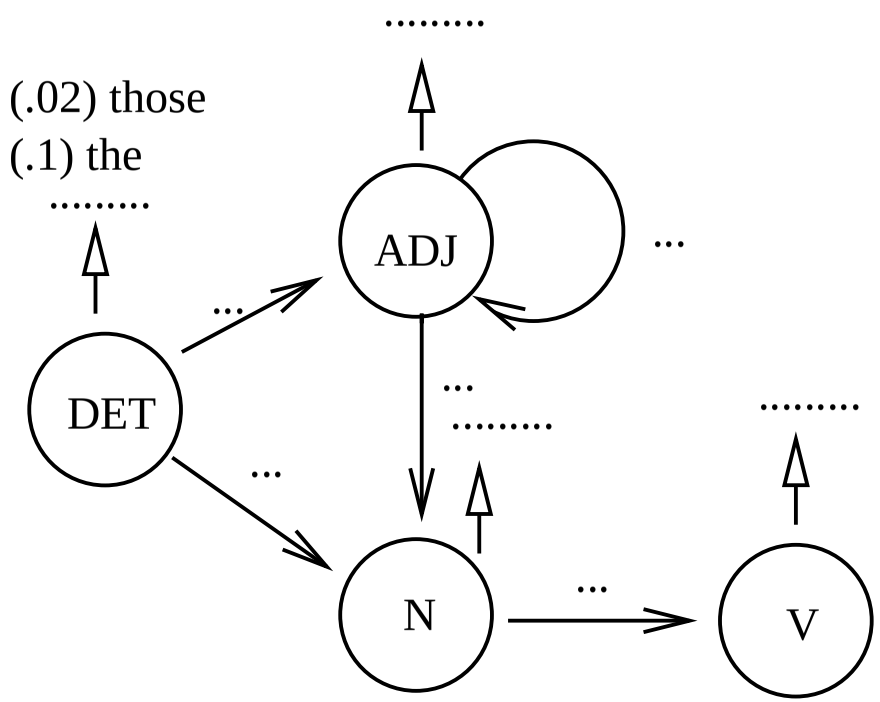
Trick is to find auxiliary variable Y making these computations easy (ex: HMM's: the paths)

Special Cases

For simpler versions of HMM's (nothing hidden), EM is not necessary.

- Part-of-speech HMM's
- n-gram models

HMM POS Model



State-of-the art for POS tagging ($\sim 95\%$ accuracy): "reverse" the HMM to find most likely tag (state) sequence

Training: state semantics *known* – use POS-tagged training data, not EM

N-gram Models

Special simple case of HMM's: state represents $N-1$ previous words.

Calculations much simpler (avoid Forward-Backward, EM)

Bigram model for $P(\text{the tut. was a roaring success})$:

$$P(\text{the}) \cdot P(\text{tut.}|\text{the}) \cdot P(\text{was}|\text{tut.}) \cdot P(\text{a}|\text{was}) \cdot P(\text{roaring}|\text{a}) \cdots$$

Trigram model:

$$P(\text{the}) \cdot P(\text{tut.}|\text{the}) \cdot P(\text{was}|\text{the tut.}) \cdot P(\text{a}|\text{tut. was}) \cdot P(\text{roaring}|\text{was a}) \cdots$$

Bigrams/trigrams: dominant language-modeling technology

Training N-gram models

Estimates for $P(w_n|w_1 w_2 \cdots w_{n-1})$ are typically based on the *maximum likelihood estimate*

$$\frac{\#(w_1 w_2 \cdots w_{n-1} w_n)}{\#(w_1 w_2 \cdots w_{n-1})},$$

where $\#(\cdot)$ indicates frequency in a large training corpus.

Standard techniques: *interpolation* [Jelinek and Mercer 80], *backoff* [Katz 87]

(more later ...)

V. The Sparse Data Problem

Predicting Probabilities

"It's hard to recognize speech"

vs.

"It's hard to wreck a nice beach"

Which is more likely? (both are *grammatical*)

Applications: speech recognition, handwriting recognition, spelling correction, ...

General problem in statistical NLP: density estimation

P("I saw her duck [with a telescope]" $\longrightarrow$ verb attachment)

P("L'avocat general" $\longrightarrow$ "the general avocado")

Maximum-Likelihood Estimation

Training: find parameters maximizing the likelihood of training set T.

A simple model:

$$P_{\text{true}}(\text{“informative brown bag seminar”}) \approx \frac{\#(\text{“informative brown bag seminar”})}{|T|}$$

1,060,000,000 web pages indexed ...

Advanced Search Language, Display, & Filtering Options Search Tips

"informative brown bag seminar"

Google Search I'm Feeling Lucky

Tip: Get the most out of Google's capabilities -- try our new Advanced Search page.

Your search - **"informative brown bag seminar"** - did not match any documents.

- Make sure all words are spelled correctly.
- Try using fewer words.
- Try using more general keywords.
- Try different keywords.

Try our Web Directory - Cool Jobs - Advertise with Us! - Add Google to your Site - Google Browser Buttons - **Everything Else**

Sparse Data Problems

Why care about unseen strings?

- For a 350M-word sample of English, an estimated 14% of triples in any new sample would be unseen [Brown et al. 1992].
- A standard corpus of word 4-tuples has a 95% unseen rate for the test set [Collins and Brooks 1995, PP-attachment].

The aggregate probability of unseen events can be very large, so we need to accurately model them.

Sparse Data Problems (cont.)

Chomsky: the sparse data problem is insurmountable!

"It is fair to assume that neither sentence

(1) Colorless green ideas sleep furiously

nor

(2) Furiously sleep ideas green colorless

... has ever occurred .... Hence, in any statistical model ... these sentences will be ruled out on identical grounds as equally "remote" from English." [Chomsky 1957]

Similarity Information

Key idea: look at information provided by similar words.

"informative brown bag talk"

"informative brown bag presentation"

$\Rightarrow$ "informative brown bag seminar" is reasonable.

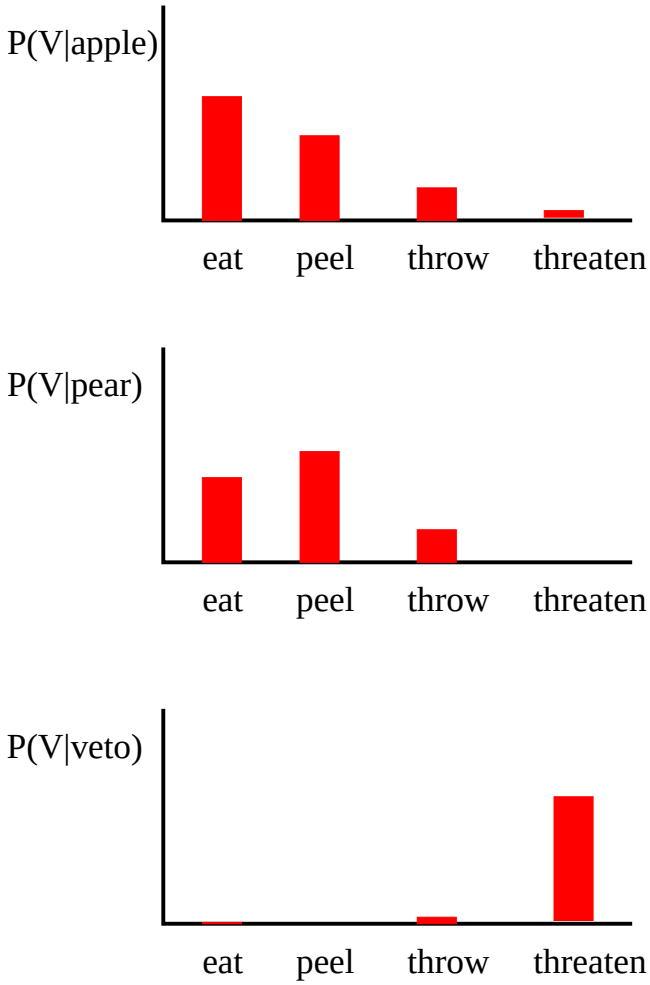
We wish to determine similarity automatically, not with pre-existing thesauri:

- domain variance (apple, sun)
- unknown words

Distributional Similarity

We are interested in distributional similarity:

x and x' are similar means $P(Y|x) \approx P(Y|x')$



Not (necessarily) semantic similarity: "You used me!" $\not\Rightarrow$ "You utilized me!")

Distributional Similarity Models

- Clustering [Brown et al. 92; Schütze 92, Pereira-Tishby-Lee 93; Karov-Edelman 96, Li-Abe 97; Rooth et al 99, Lee-Pereira 99]
  - ▷ Group words into global clusters; use clusters as models
  - ▷ Compresses the data
- Nearest neighbors [Dagan-Marcus-Markovitch 93, Dagan-Pereira-Lee 94, Dagan-Lee-Pereira 97, Lee-Pereira 99, Lee 99]
  - ▷ For each word, use words in its specific local neighborhood as model

Example: two clusters vs. two neighbors

Example: Nearest Neighbors of “Company”

euc	tau	conf	cos	var	JS	jac
city	year	govt.	talk	business	business	state
airline	state	year	hostage	airline	airline	business
industry	people	people	primary	state	firm	govt.
program	govt.	percent	referendum	bank	bank	group
org.	group	syndrome	lead	firm	state	country
bank	country	business	hearing	agency	agency	program
system	business	today	stake	govt.	group	people
today	program	firm	discussion	city	govt.	nation

Underline: unique to a function. Verb-noun pairs from AP newswire.

VI. Conclusions and References

No Myths...Only a Beginning

"The linguistic content of our program thus far is scant indeed. It is limited to one set of rules for analyzing a string of characters into a string of words, and another set of rules for analyzing a string of words into a string of sentences. Doubtless even these can be recast in terms of some information theoretic objective function. But *it is not our intention to ignore linguistics, neither to replace it. Rather, we hope to enfold it in the embrace of a secure probabilistic framework so that the two together may draw strength from one another and guide us to better natural language processing systems* in general and to better machine translation systems in particular."

— *The Mathematics of Statistical Machine Translation*
[Brown, Della Pietra, Della Pietra, and Mercer, 1993]

For Further Information ...

“The $64,000 question in computational linguistics these days is: “What should I read to learn about statistical natural language processing?”” [Magerman 95]

Short Overviews

- C. Cardie and R. Mooney, “Machine Learning and Natural Language”, introdction to *Machine Learning* 34(1-3), special issue on natural language learning, 1999.
- S. Abney, “Statistical Methods and Linguistics”, in *The Balancing Act*, J. Klavans and P. Resnik, eds., 1997.
- E. Brill and R. Mooney, “An Overview of Empirical Natural Language Processing”, *AI Magazine* 18(4), 1997.
- K. Church and R. Mercer, “Introduction to the Special Issue on Computational Linguistics Using Large Corpora”, *Computational Linguistics* 19(1), 1993.

Books

- E. Charniak, *Statistical Language Learning*, MIT Press, 1993.
  - ▷ Reviewed by D. Magerman in *Computational Linguistics* 21(1), 1995.
- T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Interscience, 1991.
- F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, 1997.
- C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.
  - ▷ Reviewed by L. Lee in *Computational Linguistics* 26(2), 2000.

Conferences

- Empirical Methods in Natural Language Processing (EMNLP).
- Workshop on Very Large Corpora (WVLC).
- General NLP conferences:
  - ▷ Association for Computational Linguistics (ACL)
  - ▷ North American Chapter of the ACL (NAACL)
  - ▷ European Chapter of the ACL (EACL)
  - ▷ Applied Natural Language Processing (ANLP)
  - ▷ International Conference on Computational Linguistics (COLING)

Many recent papers are posted on the cmp-lg server, `http://xxx.lanl.gov/cmp-lg/`, later absorbed into the Computing Research Repository computer science holdings, `http://xxx.lanl.gov/archive/cs`.