CS 391L: Machine Learning
Natural Language Learning

Raymond J. Mooney
University of Texas at Austin

Sub-Problems in NLP

- Understanding / Comprehension
  - Speech recognition
  - Syntactic analysis
  - Semantic analysis
  - Pragmatic analysis
- Generation / Production
  - Content selection
  - Syntactic realization
  - Speech synthesis
- Translation
  - Understanding
  - Generation

Ambiguity is Ubiquitous

- Speech Recognition
  - "Recognize speech" vs. "Wreck a nice beach"
- Syntactic Analysis
  - "I ate spaghetti with a fork" vs. "I ate spaghetti with meat balls."
- Semantic Analysis
  - "The dog is in the pen." vs. "The ink is in the pen."
- Pragmatic Analysis
  - Pedestrian: "Does your dog bite?,"
    Clouseau: "No."
    Pedestrian pets dog and is bitten.
    Pedestrian: "I thought you said your dog does not bite?"
    Clouseau: "That, sir, is not my dog."

Humor and Ambiguity

- Many jokes rely on the ambiguity of language:
  - Groucho Marx: One morning I shot an elephant in my pajamas. How he got into my pajamas, I'll never know.
  - She criticized my apartment, so I knocked her flat.
  - Noah took all of the animals on the ark in pairs. Except the worms, they came in apples.
  - Policeman to little boy: "We are looking for a thief with a bicycle." Little boy: "Wouldn't you be better using your eyes."
  - Why is the teacher wearing sun-glasses. Because the class is so bright.

Ambiguity is Explosive

- Ambiguities compound to generate enormous numbers of possible interpretations.
- In English, a sentence ending in *n* prepositional phrases has *over* 2^n syntactic interpretations.
  - "I saw the man with the telescope": 2 parses
  - "I saw the man on the hill with the telescope.": 5 parses
  - "I saw the man on the hill in Texas with the telescope": 14 parses
  - "I saw the man on the hill in Texas with the telescope at noon.": 42 parses

Word Sense Disambiguation (WSD) as Text Categorization

- Each sense of an ambiguous word is treated as a category.
  - "play" (verb)
    - play-game
    - play-instrument
    - play-role
  - "pen" (noun)
    - writing-instrument
    - enclosure
- Treat current sentence (or preceding and current sentence) as a document to be classified.
  - "play":
    - play-game: "John played soccer in the stadium on Friday."
    - play-instrument: "John played guitar in the band on Friday."
    - play-role: "John played Hamlet in the theater on Friday."
  - "pen":
    - writing-instrument: "John wrote the letter with a pen in New York."
    - enclosure: "John put the dog in the pen in New York."

Learning for WSD

- Assume part-of-speech (POS), e.g. noun, verb, adjective, for the target word is determined.
- Treat as a classification problem with the appropriate potential senses for the target word given its POS as the categories.
- Encode context using a set of features to be used for disambiguation.
- Train a classifier on labeled data encoded using these features.
- Use the trained classifier to disambiguate future instances of the target word given their contextual features.

7

WSD "line" Corpus

- 4,149 examples from newspaper articles containing the word "line."
- Each instance of "line" labeled with one of 6 senses from WordNet.
- Each example includes a sentence containing "line" and the previous sentence for context.

8

Senses of "line"

- Product: "While he wouldn't estimate the sale price, analysts have estimated that it would exceed $1 billion. Kraft also told analysts it plans to develop and test a line of refrigerated entrees and desserts, under the Chillery brand name."
- Formation: "C-LD-R L-V-S V-NNA reads a sign in Caldor's book department. The 1,000 or so people fighting for a place in line have no trouble filling in the blanks."
- Text: "Newspaper editor Francis P. Church became famous for a 1897 editorial, addressed to a child, that included the line "Yes, Virginia, there is a Santa Clause."
- Cord: "It is known as an aggressive, tenacious litigator. Richard D. Parsons, a partner at Patterson, Belknap, Webb and Tyler, likes the experience of opposing Sullivan & Cromwell to "having a thousand-pound tuna on the line."
- Division: "Today, it is more vital than ever. In 1983, the act was entrenched in a new constitution, which established a tricameral parliament along racial lines, whith separate chambers for whites, coloreds and Asians but none for blacks."
- Phone: "On the tape recording of Mrs. Guba's call to the 911 emergency line, played at the trial, the baby sitter is heard begging for an ambulance."

9

Experimental Data for WSD of "line"

- Sample equal number of examples of each sense to construct a corpus of 2,094.
- Represent as simple binary vectors of word occurrences in 2 sentence context.
  - Stop words eliminated
  - Stemmed to eliminate morphological variation
- Final examples represented with 2,859 binary word features.

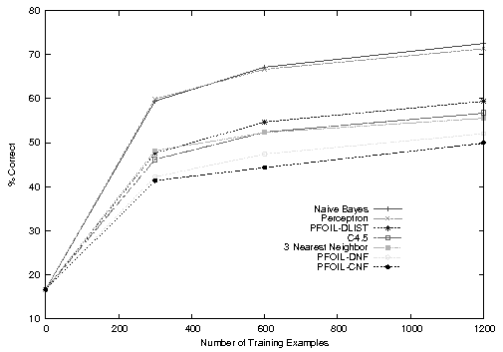
10

Learning Algorithms

- Naïve Bayes
  - Binary features
- K Nearest Neighbor
  - Simple instance-based algorithm with k=3 and Hamming distance
- Perceptron
  - Simple neural-network algorithm.
- C4.5
  - State of the art decision-tree induction algorithm
- PFOIL-DNF
  - Simple logical rule learner for Disjunctive Normal Form
- PFOIL-CNF
  - Simple logical rule learner for Conjunctive Normal Form
- PFOIL-DLIST
  - Simple logical rule learner for decision-list of conjunctive rules

11

Learning Curves for WSD of "line"



12

Discussion of Learning Curves for WSD of "line"

- Naïve Bayes and Perceptron give the best results.
- Both use a weighted linear combination of evidence from many features.
- Symbolic systems that try to find a small set of relevant features tend to overfit the training data and are not as accurate.
- Nearest neighbor method that weights all features equally is also not as accurate.
- Of symbolic systems, decision lists work the best.

13

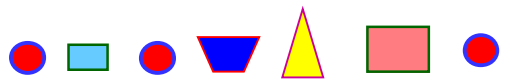
Beyond Classification Learning

- Standard classification problem assumes individual cases are disconnected and independent (i.i.d.: independently and identically distributed).
- Many NLP problems do not satisfy this assumption and involve making many connected decisions, each resolving a different ambiguity, but which are mutually dependent.
- More sophisticated learning and inference techniques are needed to handle such situations in general.

14

Sequence Labeling Problem

- Many NLP problems can viewed as sequence labeling.
- Each token in a sequence is assigned a label.
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors (not i.i.d).



foo bar blam zonk zonk bar blam

15

Part Of Speech Tagging

- Annotate each word in a sentence with a part-of-speech.
- Lowest level of syntactic analysis.

John saw the saw and decided to take it to the table.
PN V Det N Con V Part V Pro Prep Det N

- Useful for subsequent syntactic parsing and word sense disambiguation.

16

Information Extraction

- Identify phrases in language that refer to specific types of entities and relations in text.
- Named entity recognition is task of identifying names of people, places, organizations, etc. in text.

  people organizations places
  - Michael Dell is the CEO of Dell Computer Corporation and lives in Austin Texas.
- Extract pieces of information relevant to a specific application, e.g. used car ads:

  make model year mileage price
  - For sale, 2002 Toyota Prius, 20,000 mi, $15K or best offer. Available starting July 30, 2006.

17

Semantic Role Labeling

- For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.

  agent patient source destination instrument
  - John drove Mary from Austin to Dallas in his Toyota Prius.
  - The hammer broke the window.
- Also referred to a "case role analysis," "thematic analysis," and "shallow semantic parsing"

18

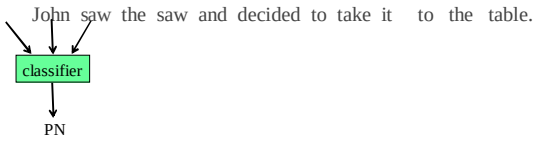
Bioinformatics

- Sequence labeling also valuable in labeling genetic sequences in genome analysis.
  extron intron
  – AGCTAACGTTCGATACGGATTACAGCCT

19

Sequence Labeling as Classification

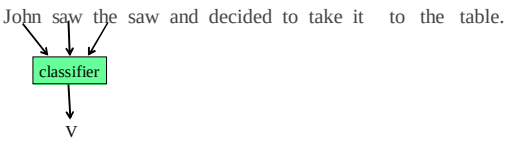
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



20

Sequence Labeling as Classification

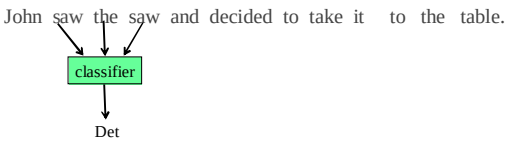
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



21

Sequence Labeling as Classification

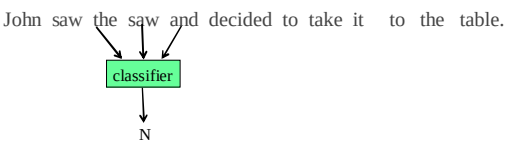
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



22

Sequence Labeling as Classification

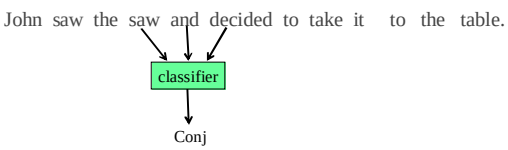
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



23

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

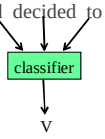


24

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

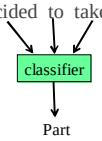


25

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

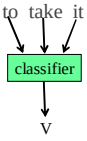


26

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

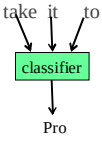


27

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

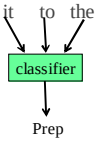


28

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

John saw the saw and decided to take it to the table.

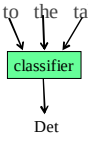


29

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

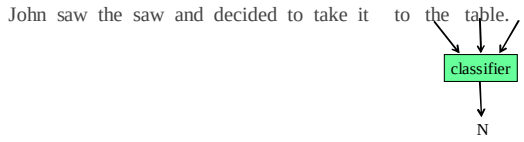
John saw the saw and decided to take it to the table.



30

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



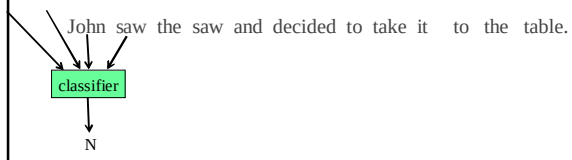
31

Sequence Labeling as Classification Using Outputs as Inputs

- Better input features are usually the categories of the surrounding tokens, but these are not available yet.
- Can use category of either the preceding or succeeding tokens by going forward or back and using previous output.

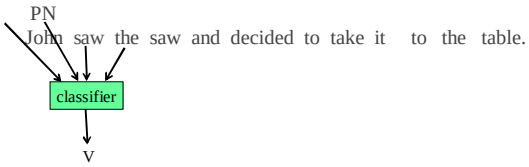
32

Forward Classification



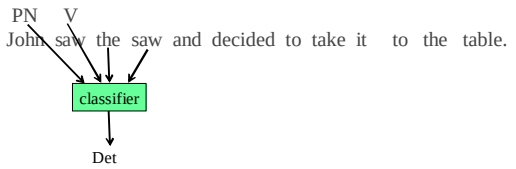
33

Forward Classification



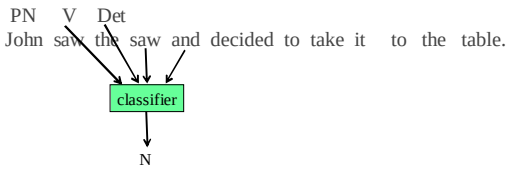
34

Forward Classification



35

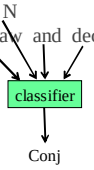
Forward Classification



36

Forward Classification

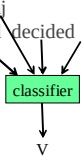
PN V Det N
John saw the saw and decided to take it to the table.



37

Forward Classification

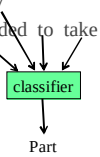
PN V Det N Conj
John saw the saw and decided to take it to the table.



38

Forward Classification

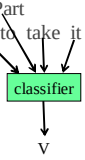
PN V Det N Conj V
John saw the saw and decided to take it to the table.



39

Forward Classification

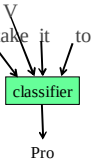
PN V Det N Conj V Part
John saw the saw and decided to take it to the table.



40

Forward Classification

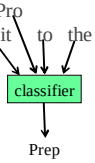
PN V Det N Conj V Part V
John saw the saw and decided to take it to the table.



41

Forward Classification

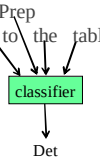
PN V Det N Conj V Part V Pro
John saw the saw and decided to take it to the table.



42

Forward Classification

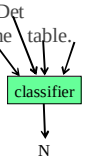
PN V Det N Conj V Part V Pro Prep
John saw the saw and decided to take it to the table.



43

Forward Classification

PN V Det N Conj V Part V Pro Prep Det
John saw the saw and decided to take it to the table.

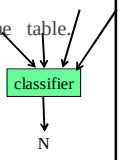


44

Backward Classification

- Disambiguating “to” in this case would be even easier backward.

John saw the saw and decided to take it to the table.

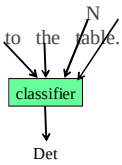


45

Backward Classification

- Disambiguating “to” in this case would be even easier backward.

John saw the saw and decided to take it to the table.

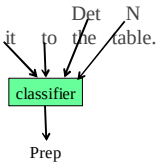


46

Backward Classification

- Disambiguating “to” in this case would be even easier backward.

Det N
John saw the saw and decided to take it to the table.

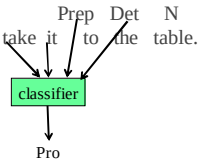


47

Backward Classification

- Disambiguating “to” in this case would be even easier backward.

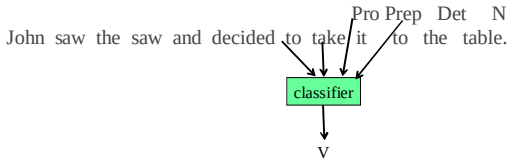
Prep Det N
John saw the saw and decided to take it to the table.



48

Backward Classification

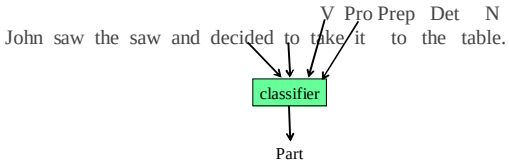
- Disambiguating “to” in this case would be even easier backward.



49

Backward Classification

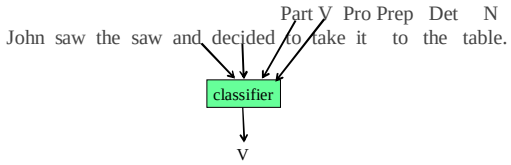
- Disambiguating “to” in this case would be even easier backward.



50

Backward Classification

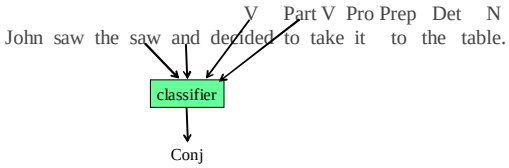
- Disambiguating “to” in this case would be even easier backward.



51

Backward Classification

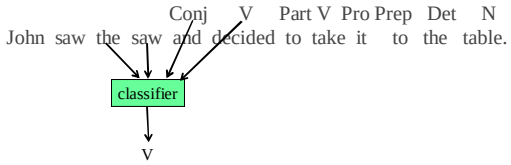
- Disambiguating “to” in this case would be even easier backward.



52

Backward Classification

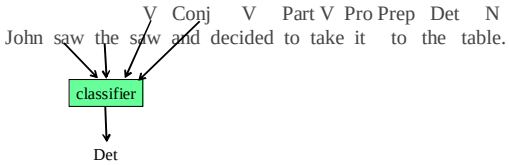
- Disambiguating “to” in this case would be even easier backward.



53

Backward Classification

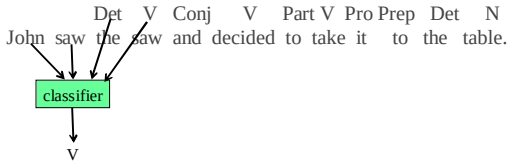
- Disambiguating “to” in this case would be even easier backward.



54

Backward Classification

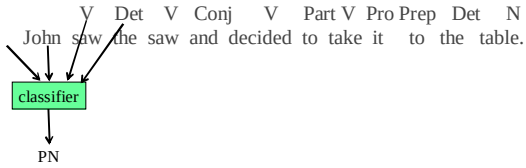
- Disambiguating “to” in this case would be even easier backward.



55

Backward Classification

- Disambiguating “to” in this case would be even easier backward.



56

Problems with Sequence Labeling as Classification

- Not easy to integrate information from category of tokens on both sides.
- Difficult to propagate uncertainty between decisions and “collectively” determine the most likely joint assignment of categories to all of the tokens in a sequence.

57

Probabilistic Sequence Models

- Probabilistic sequence models allow integrating uncertainty over multiple, interdependent classifications and collectively determine the most likely global assignment.
- Two standard models
  - Hidden Markov Model (HMM)
  - Conditional Random Field (CRF)

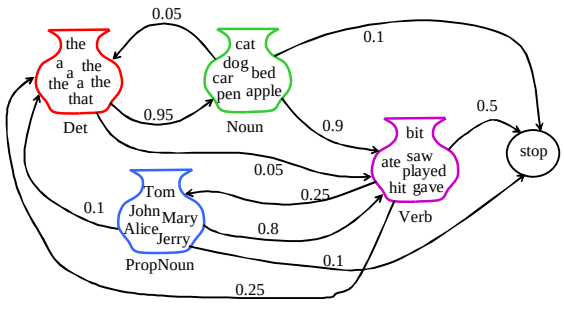
58

Hidden Markov Model

- Probabilistic generative model for sequences.
- A finite state machine with probabilistic transitions and probabilistic generation of outputs from states.
- Assume an underlying set of states in which the model can be (e.g. parts of speech).
- Assume probabilistic transitions between states over time (e.g. transition from POS to another POS as sequence is generated).
- Assume a probabilistic generation of tokens from states (e.g. words generated for each POS).

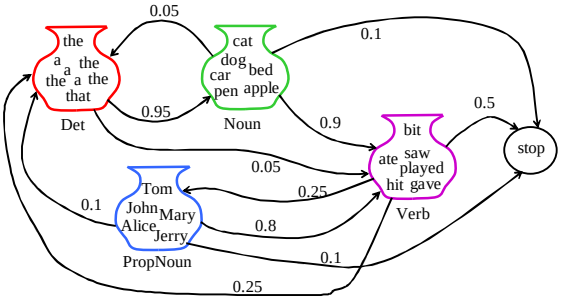
59

Sample HMM for POS



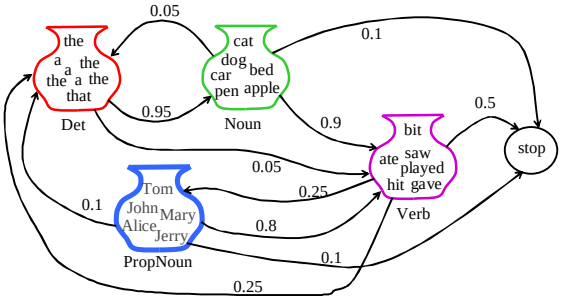
60

Sample HMM Generation



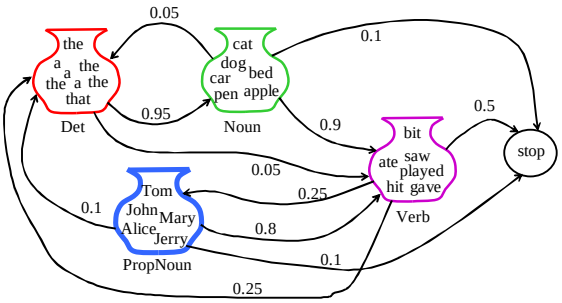
61

Sample HMM Generation



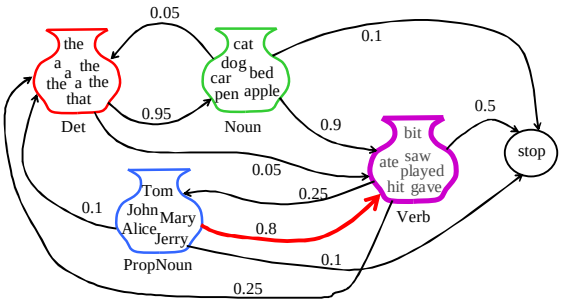
62

Sample HMM Generation



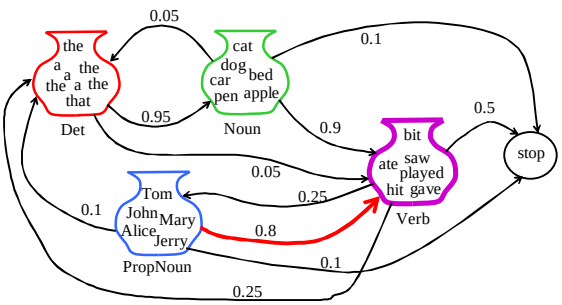
63

Sample HMM Generation



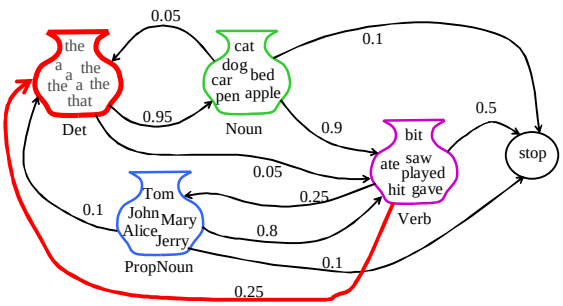
64

Sample HMM Generation



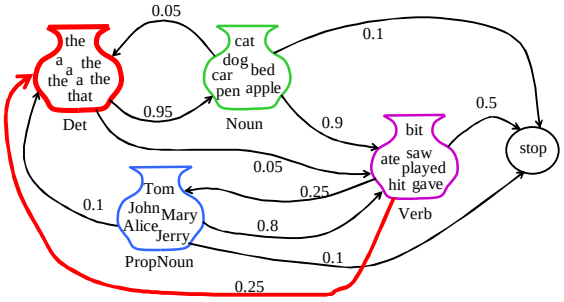
65

Sample HMM Generation



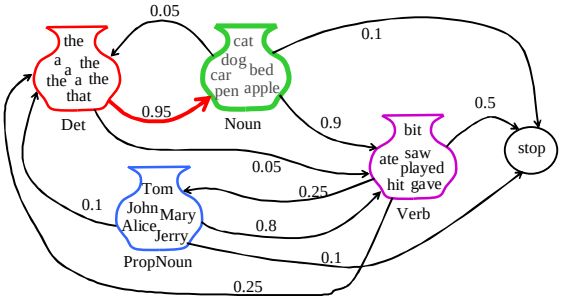
66

Sample HMM Generation



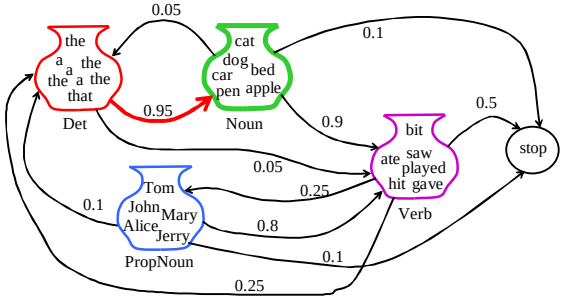
John bit the

Sample HMM Generation



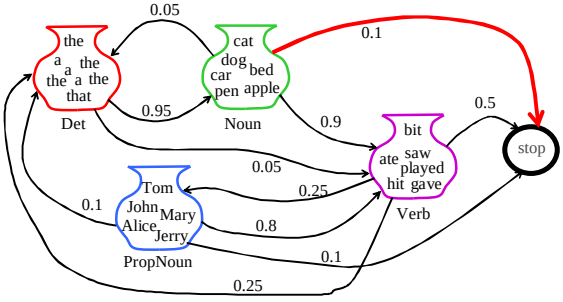
John bit the

Sample HMM Generation



John bit the apple

Sample HMM Generation



John bit the apple

Formal Definition of an HMM

- A set of N states $S=\{S_1, S_2, \ldots S_N\}$
- A set of M possible observations $V=\{V_1,V_2 \ldots V_M\}$
- A state transition probability distribution $A=\{a_{ij}\}$

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i) \qquad 1 \le i, j \le N$$

- Observation probability distribution for each state j $B=\{b_j(k)\}$

$$b_j(k) = P(v_k \text{ at } t \mid q_t = S_j) \quad 1 \le j \le N \quad 1 \le k \le M$$

- Initial state distribution $\pi = \{\pi_i\}$

$$\pi_i = P(q_1 = S_i) \qquad 1 \le i \le N$$

- Total parameter set $\lambda=\{A,B,\pi\}$

HMM Generation Procedure

- To generate a sequence of T observations:
  $O = O_1 O_2 \ldots O_T$

Choose an initial state $q_1=S_i$ according to π
For $t = 1$ to T
  Pick an observation $O_t=v_k$ based on being in state q_t using distribution $b_{qt}(k)$
  Transit to another state $q_{t+1}=S_j$ based on transition distribution a_{ij} for state q_t

Three Useful HMM Tasks

- Observation likelihood: To classify and order sequences.
- Most likely state sequence: To tag each token in a sequence with a label.
- Maximum likelihood training: To train models to fit empirical training data.

73

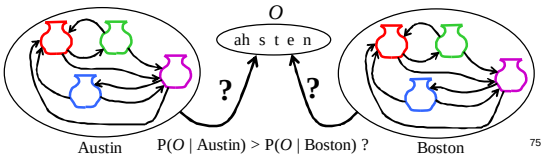
HMM: Observation Likelihood

- Given a sequence of observations, *O*, and a model with a set of parameters, λ, what is the probability that this observation was generated by this model: P(O| λ) ?
- Allows HMM to be used as a language model: A formal probabilistic model of a language that assigns a probability to each string saying how likely that string was to have been generated by the language.
- Useful for two tasks:
  - Sequence Classification
  - Most Likely Sequence

74

Sequence Classification

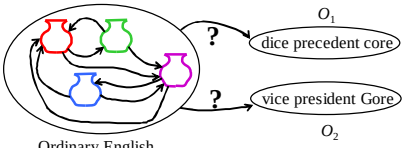
- Assume an HMM is available for each category (i.e. language).
- What is the most likely category for a given observation sequence, i.e. which category's HMM is most likely to have generated it?
- Used in speech recognition to find most likely word model to have generate a given sound or phoneme sequence.



75

Most Likely Sequence

- Of two or more possible sequences, which one was most likely generated by a given model?
- Used to score alternative word sequence interpretations in speech recognition.



76

HMM: Observation Likelihood Naïve Solution

- Consider all possible state sequences, *Q*, of length *T* that the model could have traversed in generating the given observation sequence.
- Compute the probability of this state sequence from π and *A*, and multiply it by the probabilities of generating each of given observations in each of the corresponding states in this sequence to get P(*O*,*Q*| λ).
- Sum this over all possible state sequences to get P(*O*| λ).
- Computationally complex: O(TN^T).

77

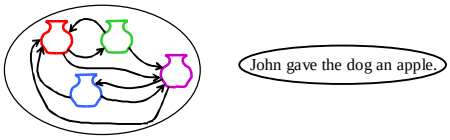
HMM: Observation Likelihood Efficient Solution

- Markov assumption: Probability of the current state only depends on the immediately previous state, not on any earlier history (via the transition probability distribution, *A*).
- Therefore, the probability of being in any state at any given time *t* only relies on the probability of being in each of the possible states at time *t*-1.
- Forward-Backward Algorithm: Uses dynamic programming to exploit this fact to efficiently compute observation likelihood in O(N^2T) time.

78

Most Likely State Sequence

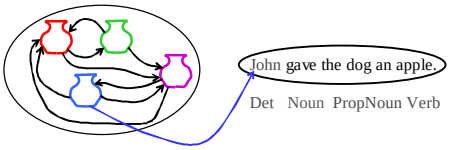
- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



79

Most Likely State Sequence

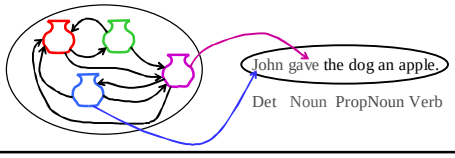
- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



80

Most Likely State Sequence

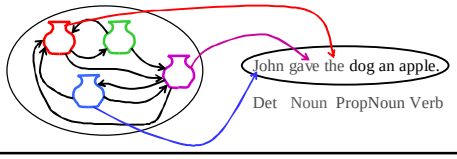
- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



81

Most Likely State Sequence

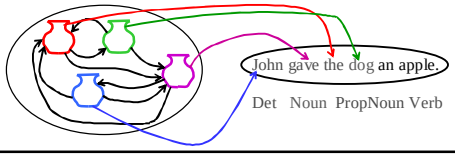
- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



82

Most Likely State Sequence

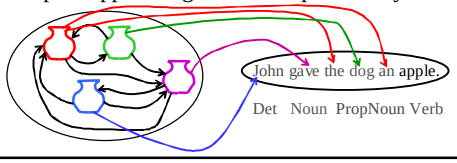
- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



83

Most Likely State Sequence

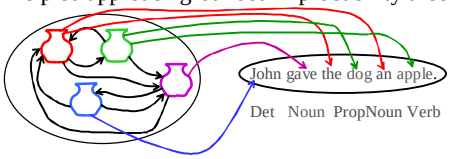
- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



84

Most Likely State Sequence

- Given an observation sequence, O, and a model, λ, what is the most likely state sequence, $Q=Q_1,Q_2,\ldots Q_T$, that generated this sequence from this model?
- Used for sequence labeling, assuming each state corresponds to a tag, it determines the globally best assignment of tags to all tokens in a sequence using a principled approach grounded in probability theory.



85

HMM: Most Likely State Sequence Efficient Solution

- Dynamic Programming can also be used to exploit the Markov assumption and efficiently determine the most likely state sequence for a given observation and model.
- Standard procedure is called the Viterbi algorithm (Viterbi, 1967) and also has $O(N^2T)$ time complexity.

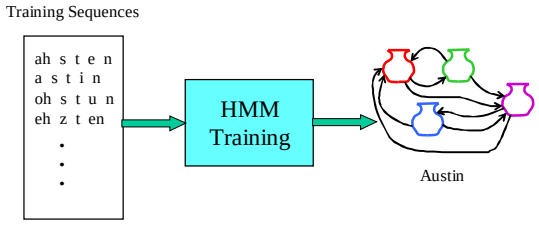
86

Maximum Likelihood Training

- Given an observation sequence, O, what set of parameters, λ, for a given model maximizes the probability that this data was generated from this model ($P(O|\lambda)$)?
- Used to train an HMM model and properly induce its parameters from a set of training data.
- Only need to have an unannotated observation sequence (or set of sequences) generated from the model. Does not need to know the correct state sequence(s) for the observation sequence(s). In this sense, it is unsupervised.

87

Maximum Likelihood Training



88

HMM: Maximum Likelihood Training Efficient Solution

- There is no known efficient algorithm for finding the parameters, λ, that truly maximize $P(O|\lambda)$.
- However, using iterative re-estimation, the Baum-Welch algorithm, a version of a standard statistical procedure called Expectation Maximization (EM), is able to *locally* maximize $P(O|\lambda)$.
- In practice, EM is able to find a good set of parameters that provide a good fit to the training data in many cases.

89

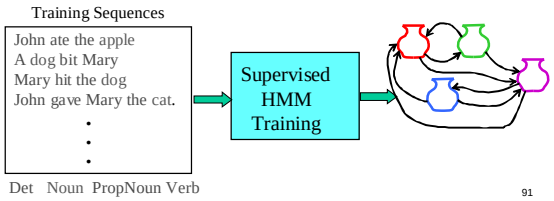
Sketch of Baum-Welch (EM) Algorithm for Training HMMs

Assume an HMM with N states.
Randomly set its parameters $\lambda=\{A,B,\pi\}$
(so that they represent legal distributions)
Until converge (i.e. λ no longer changes) do:
E Step: Use the forward/backward procedure to determine the probability of various possible state sequences for generating the training data
M Step: Use these probability estimates to re-estimate values for all of the parameters λ

90

Supervised HMM Training

- If training sequences are labeled (tagged) with the underlying state sequences that generated them, then the parameters, $\lambda=\{A,B,\pi\}$ can all be estimated directly from counts accumulated from the labeled sequences (with appropriate smoothing).



Generative vs. Discriminative Models

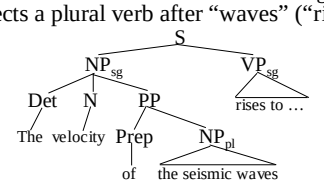
- HMMs are generative models and are *not* directly designed to maximize the performance of sequence labeling. They model the *joint distribution* P(*O*,*Q*)
- HMMs are trained to have an accurate probabilistic model of the underlying language, and not all aspects of this model benefit the sequence labeling task.
- Discriminative models are specifically designed and trained to maximize performance on a particular inference problem, such as sequence labeling. They model the *conditional distribution* P(*Q* | *O*)

Conditional Random Fields

- Conditional Random Fields (CRFs) are discriminative models specifically designed and trained for sequence labeling.
- Experimental results verify that they have superior accuracy on various sequence labeling tasks.
  - Noun phrase chunking
  - Named entity recognition
  - Semantic role labeling
- However, CRFs are much slower to train and do not scale as well to large amounts of training data.

Limitations of Finite-State Models

- Finite state models like HMMs and CRFs are unable to model all aspects of natural-language.
- The complexity and nested phrasal structure of natural language require recursion and the power of context free grammars (CFGs).
- For example "The velocity of the seismic waves rises to…" is hard for a HMM POS tagger since it expects a plural verb after "waves" ("rise")

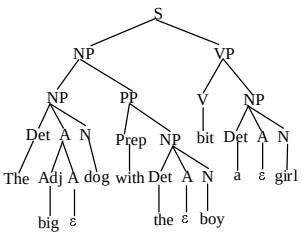


Probabilistic Context Free Grammar (PCFG)

- A PCFG is a probabilistic version of a CFG where each production has a probability.
- Probabilities of all productions rewriting a given non-terminal must add to 1, defining a distribution for each non-terminal.
- String generation is now probabilistic where production probabilities are used to non-deterministically select a production for rewriting a given non-terminal.

Sample PCFG

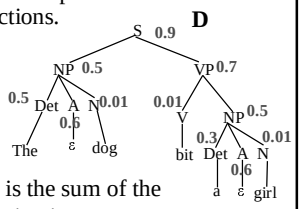
S → NP VP	0.9	+ = 1
S → VP	0.1	
NP → Det A N	0.5	+ = 1
NP → NP PP	0.3	
NP → PropN	0.2	
A → ε	0.6	+ = 1
A → Adj A	0.4	
PP → Prep NP	1.0	+ = 1
VP → V NP	0.7	+ = 1
VP → VP PP	0.3	



Sentence Probability

- Assume productions for each node are chosen independently.
- Probability of derivation is the product of the probabilities of its productions.

P(D) = 0.9 x 0.5 x 0.7 x 0.5 x 0.6 x 0.01 x 0.01 x 0.5 x 0.3 x 0.6 x 0.01 = 8.505 x 10^{-9}



- Probability of a sentence is the sum of the probability of all of its derivations.

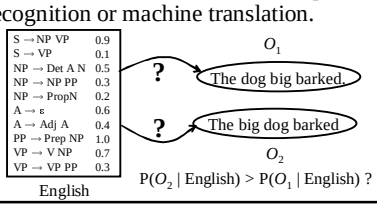
Since it is unambiguous, P("The dog bit a girl") = 8.505 x 10-9

Three Useful PCFG Tasks

- Observation likelihood: To classify and order sentences.
- Most likely derivation: To determine the most likely parse tree for a sentence.
- Maximum likelihood training: To train a PCFG to fit empirical training data.

PCFG: Observation Likelihood

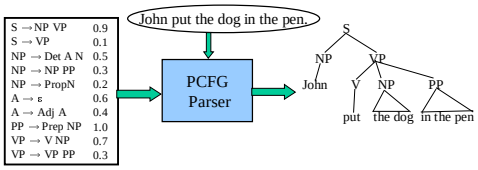
- There is an analog to Forward/Backward called the Inside/Outside algorithm for efficiently determining how likely a string is to be produced by a PCFG.
- Can use a PCFG as a language model to choose between alternative sentences for speech recognition or machine translation.



$P(O_2 \mid \text{English}) > P(O_1 \mid \text{English})$?

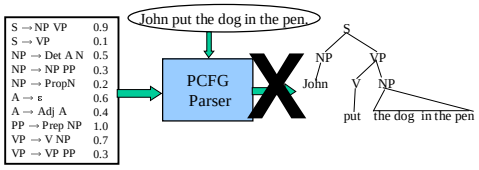
PCFG: Most Likely Derivation

- There is an analog to the Viterbi algorithm to efficiently determine the most probable derivation (parse tree) for a sentence.
- Time complexity is $O(N^3T^3)$ where N is the number of non-terminals in the grammar and T is the length of the sentence.



PCFG: Most Likely Derivation

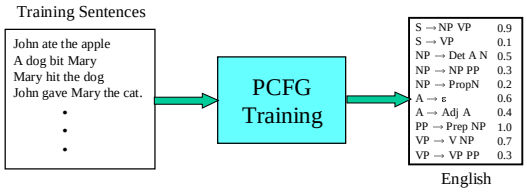
- There is an analog to the Viterbi algorithm to efficiently determine the most probable derivation (parse tree) for a sentence.
- Time complexity is $O(N^3T^3)$ where N is the number of non-terminals in the grammar and T is the length of the sentence.



PCFG: Maximum Likelihood Training

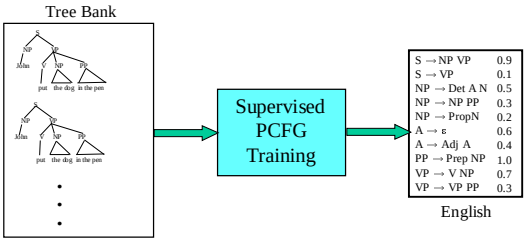
- Given a set of sentences, induce a grammar that maximizes the probability that this data was generated from this grammar.
- Assume the number of non-terminals in the grammar is specified.
- Only need to have an unannotated set of sequences generated from the model. Does not need correct parse trees for these sentences. In this sense, it is unsupervised.

PCFG: Maximum Likelihood Training



PCFG: Supervised Training

- If parse trees are provided for training sentences, a grammar and its parameters can be can all be estimated directly from counts accumulated from the tree-bank (with appropriate smoothing).

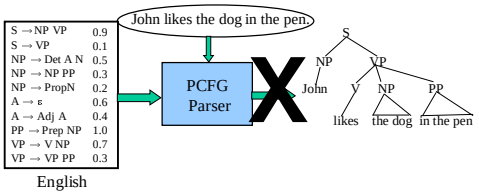


PCFG Comments

- Unsupervised training (of PCFGs or HMMs) do not to work very well. They tend to capture alternative structure in the data that does not directly reflect general syntax.
- Since probabilities of productions do not rely on specific words or concepts, only general structural disambiguation is possible.
- Consequently, vanilla PCFGs cannot resolve syntactic ambiguities that require semantics to resolve, e.g. ate with fork vs. meatballs.
- In order to work well, PCFGs must be lexicalized, i.e. productions must be specialized to specific words by including their head-word in their LHS non-terminals (e.g. VP-ate).

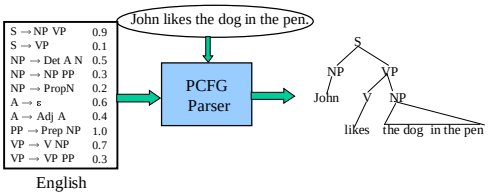
Example of Importance of Lexicalization

- A general preference for attaching PPs to verbs rather than NPs in certain structural situations could be learned by a vanilla PCFG.
- But the desired preference can depend on specific words.



Example of Importance of Lexicalization

- A general preference for attaching PPs to verbs rather than NPs in certain structural situations could be learned by a vanilla PCFG.
- But the desired preference can depend on specific words.



Treebanks

- English Penn Treebank: Standard corpus for testing syntactic parsing consists of 1.2 M words of text from the Wall Street Journal (WSJ).
- Typical to train on about 40,000 parsed sentences and test on an additional standard disjoint test set of 2,416 sentences.
- Chinese Penn Treebank: 100K words from the Xinhua news service.
- Other corpora existing in many languages, see the Wikipedia article “Treebank”

Treebank Results

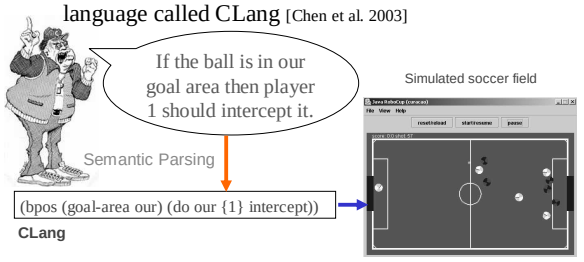
- Standard accuracy measurements judge the fraction of the constituents that match between the computed and human parse trees. If *P* is the system's parse tree and *T* is the human parse tree (the "gold standard"):
  - Recall = (# correct constituents in *P*) / (# constituents in *T*)
  - Precision = (# correct constituents in *P*) / (# constituents in *P*)
- Labeled Precision and labeled recall require getting the non-terminal label on the constituent node correct to count as correct.
- Results of current state-of-the-art systems on the English Penn WSJ treebank are about 90% labeled precision and recall.

Semantic Parsing

- ***Semantic Parsing***: Transforming natural language (NL) sentences into computer executable complete logical forms or meaning representations (MRs) for some application.
- Example application domains
  - CLang: Robocup Coach Language
  - Geoquery: A Database Query Application

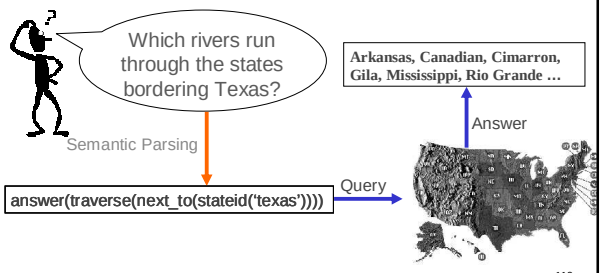
CLang: RoboCup **C**oach **Lang**uage

- In RoboCup Coach competition teams compete to coach simulated players [*http://www.robocup.org*]
- The coaching instructions are given in a formal language called CLang [Chen et al. 2003]



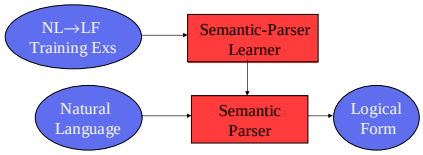
Geoquery: A Database Query Application

- Query application for U.S. geography database containing about 800 facts [Zelle & Mooney, 1996]



Learning Semantic Parsers

- Manually programming robust semantic parsers is difficult due to the complexity of the task.
- Semantic parsers can be learned automatically from sentences paired with their logical form.



Our Semantic-Parser Learners

- CHILL+WOLFIE (Zelle & Mooney, 1996; Thompson & Mooney, 1999, 2003)
  - Separates parser-learning and semantic-lexicon learning.
  - Learns a deterministic parser using ILP techniques.
- COCKTAIL (Tang & Mooney, 2001)
  - Improved ILP algorithm for CHILL.
- SILT (Kate, Wong & Mooney, 2005)
  - Learns symbolic transformation rules for mapping directly from NL to LF.
- SCISSOR (Ge & Mooney, 2005)
  - Integrates semantic interpretation into Collins' statistical syntactic parser.
- WASP (Wong & Mooney, 2006)
  - Uses syntax-based statistical machine translation methods.
- KRISP (Kate & Mooney, 2006)
  - Uses a series of SVM classifiers employing a string-kernel to iteratively build semantic representations.

Experimental Corpora

- CLang
  - 300 randomly selected pieces of coaching advice from the log files of the 2003 RoboCup Coach Competition
  - 22.52 words on average in NL sentences
  - 14.24 tokens on average in formal expressions
- GeoQuery [Zelle & Mooney, 1996]
  - 250 queries for the given U.S. geography database
  - 6.87 words on average in NL sentences
  - 5.32 tokens on average in formal expressions

115

Experimental Methodology

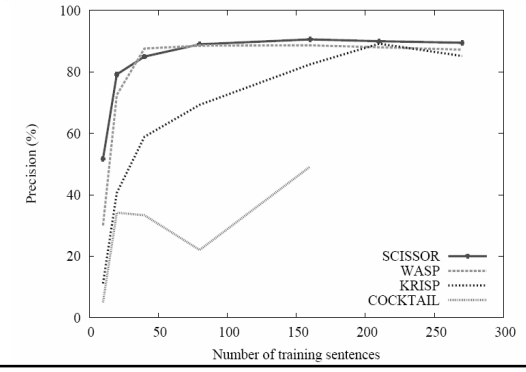
- Evaluated using standard 10-fold cross validation
- Correctness
  - CLang: output ***exactly matches*** the correct representation
  - Geoquery: the resulting query retrieves the same answer as the correct representation
- Metrics

$$Precision = \frac{|Correct\ Completed\ Parses|}{|Completed\ Parses|}$$

$$Recall = \frac{|Correct\ Completed\ Parses|}{|Sentences|}$$

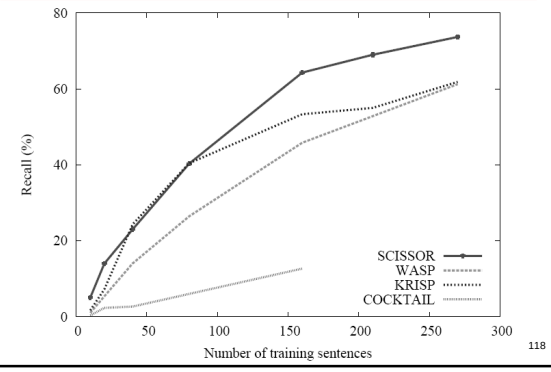
116

Precision Learning Curve for CLang



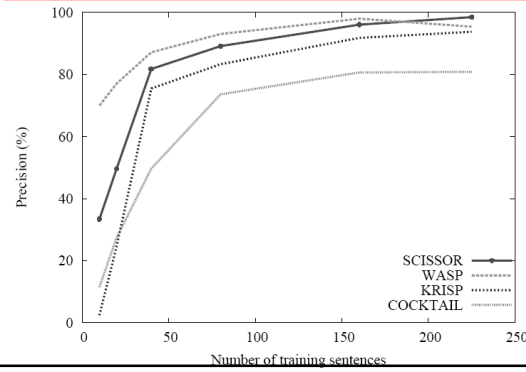
117

Recall Learning Curve for CLang



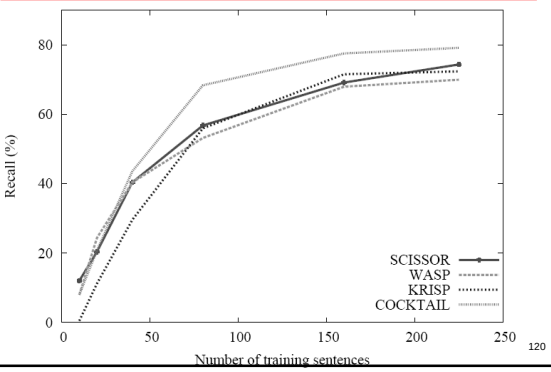
118

Precision Learning Curve for GeoQuery



119

Recall Learning Curve for Geoquery



120

Issues for Future Research

- Manual annotation of large corpora is difficult. Potential solutions include:
  - Active learning
  - Unsupervised learning
  - Semi-supervised learning
  - Learning from natural context
- Most progress has involved syntactic analysis. More work is needed on semantic and pragmatic analysis.
  - Semantic role labeling: PropBank and FrameNet
  - Semantic parsing: OntoNotes?
- What are the implications for our understanding of human language learning?
  - Nativism vs. empiricism
- What are the implications for our understanding of human language evolution?

121

Conclusions

- Resolving ambiguity in natural language is the most difficult aspect of NLP.
- Properly resolving ambiguity requires many types of knowledge that must be efficiently and effectively integrated during processing.
- Manually encoding this knowledge is very difficult.
- Machine learning methods can learn the requisite knowledge from various types of annotated and unannotated corpora.
- Learning methods have proven more successful for building accurate, robust NLP systems than manual knowledge acquisition.

122