INTRODUCTION TO CORPUS-BASED STATISTICS-ORIENTED (CBSO) TECHNIQUES

(PART III: TECHNIQUES)

Keh-Yih Su
Tung-Hui Chiang
Jing-Shin Chang

Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, TAIWAN 30043, R.O.C.
email: kysu@bdc.com.tw

HOW TO IMPLEMENT A CBSO APPROACH?

- ☐ Language Modeling
  - Linguistic Problem Abstraction
  - Feature Selection
  - Mathematical Formulation
- ☐ Corpus Annotation (Optional)
  - Annotate corpora with required information
- ☐ Parameter Estimation
- ☐ Parameter Smoothing
- ☐ Parameter Learning
  - Supervised Learning:
    - —Discrimination Issues
    - —Robustness Issues
  - Unsupervised Learning:
    - —EM Training
    - —Viterbi Training

FEATURE SELECTION

- Goal: select the best set of *d* features which optimizes a criterion function from a large set of features.

  - select the most discriminative features for processing
  - reduce the dimension of the feature space and the size of the parameter space.
  - reduce redundant information without degrading system performance
  - eliminate irrelevant or noisy features to reduce their effects on performance

Sequential Forward Selection [Devijver 82]

☐ **Procedures**:

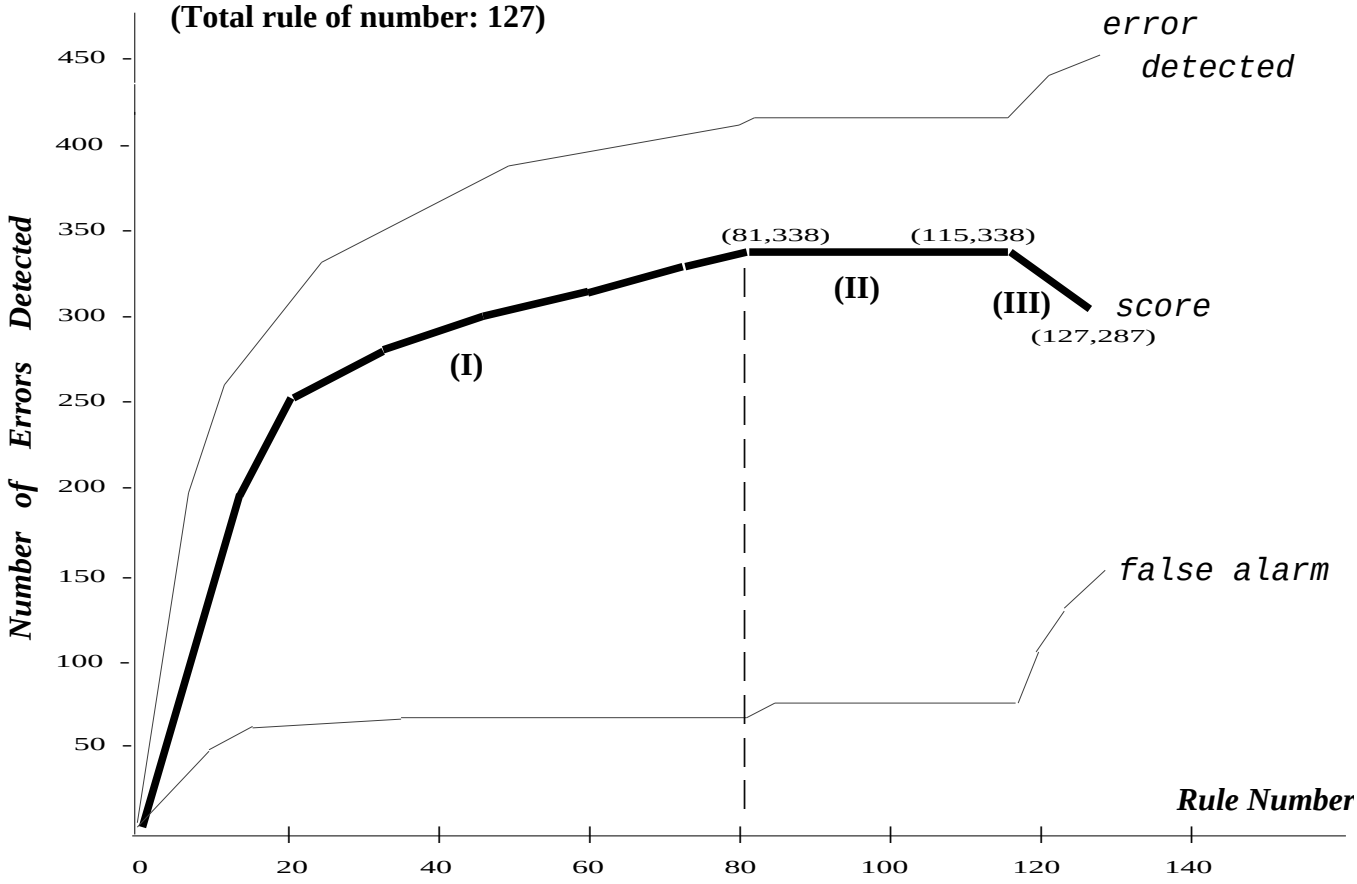
- Initially the feature set contains no feature.
- Add one feature to the current feature set to form an enlarged feature set.

  —the one being selected is the one that maximizes some criterion function (e.g., accuracy rate) when used jointly with the current feature set.

- Repeat until the feature set contains *d* features.

☐ Example of Rule Selection with SFS [Liu 93]:

Score = (number of error detected)- (number of false alarm)
Rule number: based on the score of a rule (i.e., rule-ordering)
(Total rule of number: 127)



Classification and Regression Tree (CART) [Breiman 1984]

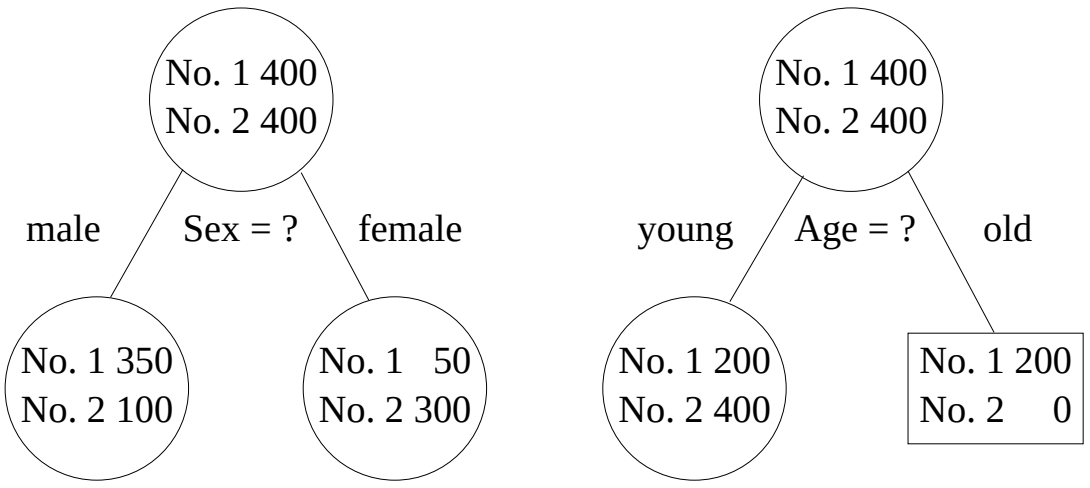
Binary Tree Construction

- Select features to split the node (see the following figure)
- Decide when to stop splitting the node
- Assign label to each node
- Use Cross-Validation to select the best tree structure

Classification Process

- Just follow the binary decision tree from the root to terminal nodes

Figure 1 Example of binary decision trees in CART method.



Splitting

- How to choose features to split the node
  - Minimize impurity criterion after splitting
  - Impurity criteria
    - —Estimated classification error after splitting
    - —Estimated entropy after splitting
      - —$\Delta I(s,t) = I(t) - P_L I(t_L) - P_R I(t_R)$
      - —$I(t) = -\sum_{j=1}^{N} P(j \mid t) \cdot \log P(j \mid t)$
    - —Others, e.g., minimum or maximum value.

Termination

☐ When to stop

- Grow the tree until only one class of data points in a node, or
- Grow the tree until the number of data points in a node is less than a preset value, or
- Grow the tree until the levels of splitting is larger than a preset value, or
- Grow the tree until $\max_{s \in S} \Delta I(s,t) < \beta$.

☐ Cross Validation

- Construct sequence of subtrees, T_0 ,..., T_n, ranging from full tree to just the root node
- Estimate "honest" error rate for each subtree by using Cross-Validation
- Choose tree size with minimum "honest" error rate

☐ Terminal Node Assignment

- Majority Vote: Choose most frequent class to label the node, choose mean value for regression.

☐ Advantages of Tree-based Modeling [Breiman 84]

- It is very flexible, and can be applied to any data structure through the appropriate selection of the set of features (the set of questions).
- The final classification has a simple form which can be compactly stored and that efficiently classifies new data.
- The tree procedure output gives easily understood and interpreted information regarding the predictive structure of the data.

☐ Examples

- End of sentence detection
  - —Features: number of words, case before and after ".", etc.
  - —Performance: 99.84% (cross-validated)
- Vowel classification
  - —Features: vowel duration, phonemic context, etc.
  - —Performance: 84% (cross-validated)

MATHEMATICAL FORMULATION

BAYESIAN DECISION VS. MAXIMUM-LIKELIHOOD DECISION

☐ **Bayesian Decision Rule**:
Find the most probable source model (M) for a given observation (O) by choosing the one with the maximum conditional probability P(M|O): $\hat{M}_B = \underset{M}{\operatorname{argmax}} P(M \mid O)$

- It is the optimal classifier that minimizes the error rate.

☐ **Maximum Likelihood Decision Rule**:
Find the model (M) which is most likely to generate the observation (O): $\hat{M}_M = \underset{M}{\operatorname{argmax}} P(O \mid M)$.

- It is the same as the Bayesian Decision if prior probability, i.e., $P(M)$, is uniformly distributed (since $P(M \mid O) = \frac{P(O|M) \cdot P(M)}{P(O)}$)

PERFORMANCE EVALUATION

- Performance is an estimated value estimated from finite testing instances.
- Performance evaluation is affected by the size of the testing instances and the evaluation method.
- Performance measure:
  - training set performance
  - testing set performance
- Why testing set performance: over-tuning of the parameters to fit the training set
  - why get over-tuned: the number of adjustable parameters is greater than needed.
  - result in an over-optimistic estimation of the performance from the training set performance. (100% performance is possible if the number of parameter or the modeling complexity is too large.)

□ EVALUATION METHODS:

- **Resubstitution Estimate**:
  use the same set of samples to design and test a model (training set performance)
- **Holdout Estimate**:
  use two mutually exclusive sets of samples to design and test a model
- **Leave-one-out Estimate**:
  use one sample for testing and the other samples for design; test the model in rotation for each single sample
- **Rotation Estimate**:
  use one subset of the samples for testing and the other subsets for design; test the model in rotation for each subset

PARAMETER SMOOTHING

☐ Not Enough Data to Train Parameters

- IBM use 81 million parameters from 40,000 sentence pairs, about 800,000 words in each language.
- In general, use training set with size over $10 \cdot N_p$ to achieve good generalization, where N_p is the number of parameters.
- Smoothing is a very important issue in estimating parameters

☐ Parameter Smoothing Techniques:

- Good-Turing estimate [Good 53]
- Back-Off procedures [Katz 87]
- Adding a flattening constant [Su 89]
- Clipping with a floor value
- Deleted Interpolation [Bahl 83]

  —Use the information from correlated and less restricted parameters, thus better estimated.

Smoothing Techniques

- **Good-Turing Method** [Good 53]:

$$C^*(x) = r^* \approx (r+1)\frac{N_{r+1}}{N_r}$$

where $C(x) = r$ is the number of occurrence of event $X = x$; $C^*(x) = r^*$ is the estimated frequency count that x would occur, and N_r is the number of events that occurs r times.

- $P^*_{GT}(X = x_i) \approx \frac{C^*(X=x_i)}{\sum_j C^*(X=x_j)}$

- **Back-off Method** [Katz 87]
  - recursively reduce contextual window if the frequency of is zero for larger window size

$$P_{BF}\left(c_i \mid c_{i-m}^{i-1}\right) = \begin{cases} P_{GT}\left(c_i \mid c_{i-m}^{i-1}\right) & C\left(c_{i-m}^{i}\right) > 0 \\ \alpha \cdot P_{BF}\left(c_i \mid c_{i-(m-1)}^{i-1}\right) & C\left(c_{i-m}^{i}\right) = 0, C\left(c_{i-m}^{i-1}\right) > 0 \\ P_{BF}\left(c_i \mid c_{i-(m-1)}^{i-1}\right) & otherwise \end{cases}$$

PARAMETER LEARNING (SUPERVISED MODE)

- Adaptive learning is required to adjust the estimated parameters according to misjudged instances or unreliably recognized instances
- Why adaptive learning
  - Discrimination Issues
    - —System recognition rate is related to the ranks of the candidate. Maximum likelihood approaches achieve the performance indirectly through the estimation process.
    - —Hence, the criterion of maximizing likelihood is not equivalent to maximizing the recognition rate in the training corpus
  - Robustness Issues
    - —Statistic variation between training corpora and unseen text is not considered in estimation
    - —Hence, minimizing the error rate in the training corpora is not equivalent to maximizing the recognition rate in unseen text

☐ Discrimination Enhancement [Su 94]

- Find a discrimination function $g_j\left(O'_j, \lambda'_j\right)$ which can well preserve the correct ranking orders.

  —Three ways to search for a good discrimination function, starting from a preliminary parameter $\hat{\lambda}$: (1) change $\hat{\lambda} \rightarrow \lambda'$, (2) transform $O \rightarrow O'$, (3) adopt a good measuring function (e.g., $P(\cdot) \rightarrow g_j(\cdot)$).

  —Find a measuring function $g_j\left(O'_j, \lambda'_j\right)$ for the transformed observation vector O'_j and adjusted parameter set λ'_j that maximize the probability of getting the correct ranks

☐ Robustness Enhancement [Su 94]

- Enlarge the inter-cluster distance and reduce intra-cluster variance to achieve maximum separation in a measure space

  —Discard unreliable parameters

  —Enlarge the margin between the correct analysis and the competing candidates in its confusing set

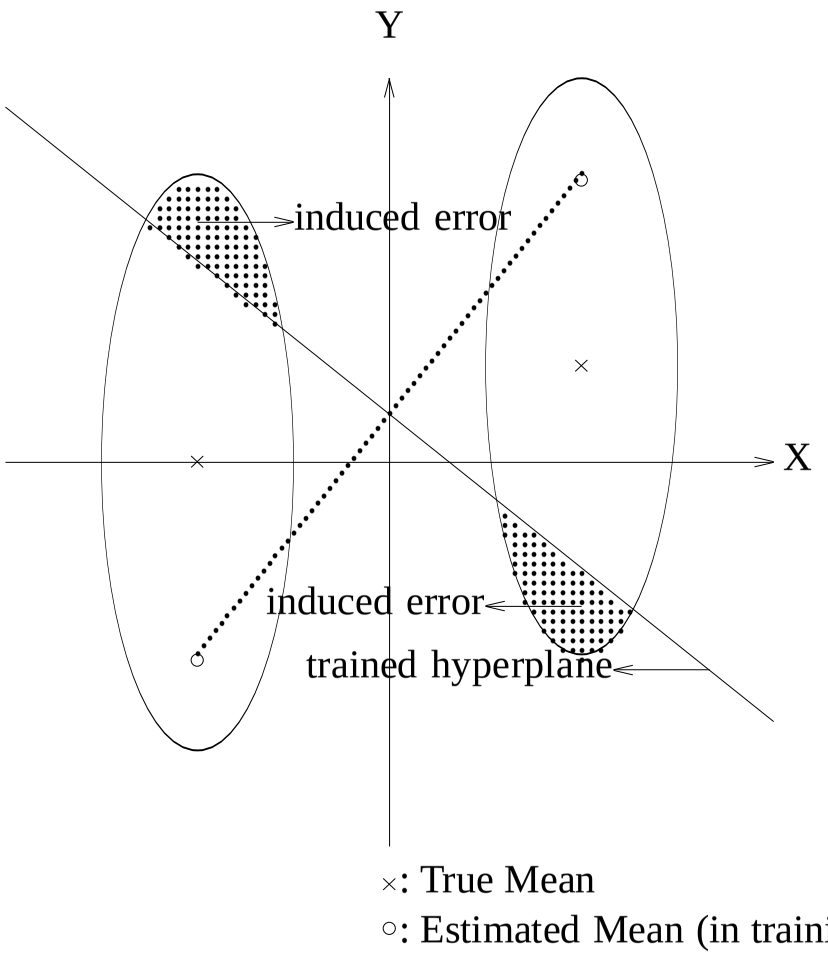
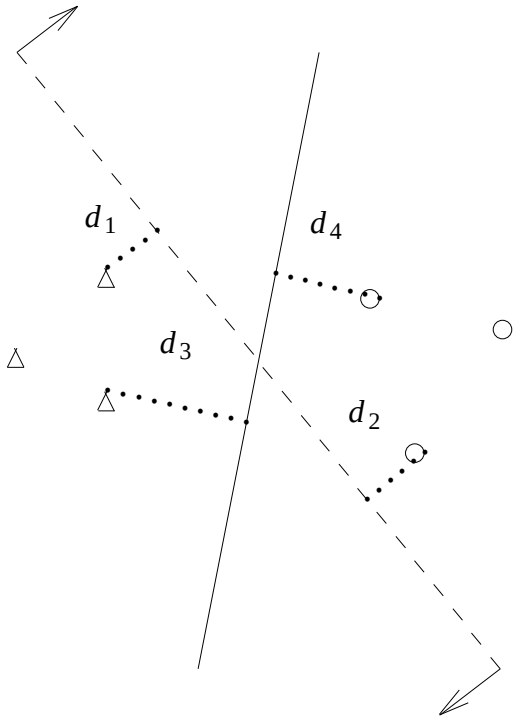


Figure 1. An illustration of projecting observations into subspace to reduce error rate



○ : Class 1

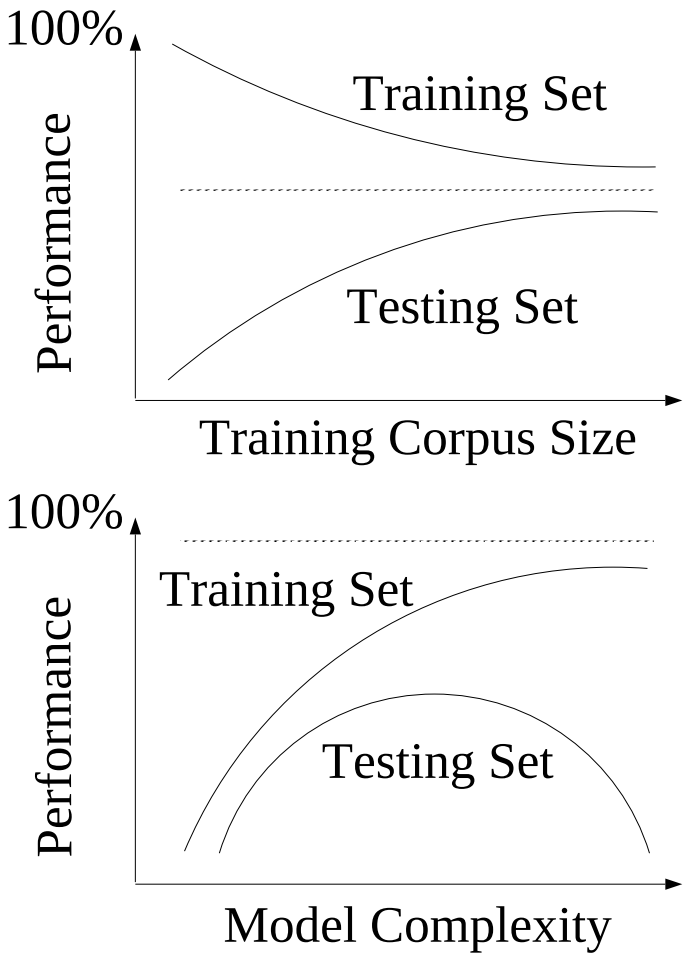
△ : Class 2

– – – – – : Decision boundary obtained by traditional adaptive learning algorithms

——— : Decision boundary obtained by robust adaptive learning method

Figure 2. An illustration of the concept of maximum separation classification

General Behavior of a (Statistical) Model



Learning Procedure

☐ **Initialization**: Initialize the parameters with maximum likelihood estimation (+ smoothing).

☐ **Calculate the miss-recognition distance**:
Let the highest two scores and the correct score be

$$^1SC_z, ^2SC_z, ^cSC_z, \qquad z \in \{syl, lex, syn, sem\},$$

then the distance d for incorrect recognition is defined as follows:

$$d = \begin{cases} ^1SC_z - ^cSC_z & if\ error \\ ^1SC_z - ^2SC_z & if\ correct\ and\ \frac{^1SC_z - ^2SC_z}{\max\{|^1SC_z|, |^2SC_z|\}} < \beta\% \end{cases}$$

☐ **Adjust the parameters**:

- Decide the amount of adjustment
  - —A loss $l(d)$, which is a function of the distance d, is defined for miss-recognition.
  - —The amount of adjustment of parameters $\triangle\Lambda^{(p)}$ in the p-th iteration is determined such that the risk function $R = E[l(\mathbf{d})] \approx \frac{1}{N}\sum_{i=1}^{N} l(d_i)$ (the expected loss) decrease:

$$\Lambda^{(p+1)} = \Lambda^{(p)} + \triangle\Lambda^{(p)},$$
$$\triangle\Lambda^{(p)} = -\epsilon U \nabla R,$$

- $\epsilon(p)$ is the learning rate function which is a mono-decreasing function of the iteration number p.

- $l(d) = \tan^{-1}\left(\frac{d}{d_0}\right)$, $l'(d) = \frac{d_0}{d_0^2+d^2}$, d_0 is a small positive constant.
- U is a positive definite matrix controlling convergent speed of parameters.

- The parameters are adjusted such that the score of the correct candidate is increased while the score of the top rank candidate is decreased.
- The learning procedure would converge in mean, which means the *average risk* would decrease as the learning procedure proceeds.

□ **Robustness enhencement:** the learning process continuously proceeds until ${}^{c}SC \geq {}^{2}SC + \delta$; that is, the margin between the correct analysis and the second highest candidate exceeds a preset threshold δ.

Example of Learning

- Sentence: “Press the left button”
- Correct tag sequence: “v art adj n”
- Score:

$$^{c}SC = S(\mathrm{v}|\mathrm{Press}) + S(\mathrm{art}|\mathrm{the}) + S(\mathrm{adj}|\mathrm{left})^{*} + S(\mathrm{n}|\mathrm{button}) + S(\mathrm{v}|@) + S(\mathrm{art}|\mathrm{v}) + S(\mathrm{adj}|\mathrm{art})^{*} + S(\mathrm{n}|\mathrm{adj})^{*}$$

- Tag sequence with the highest score: “v art v n”
- Score:

$$^{1}SC = S(\mathrm{v}|\mathrm{Press}) + S(\mathrm{art}|\mathrm{the}) + S(\mathrm{v}|\mathrm{left})^{*} + S(\mathrm{n}|\mathrm{button}) + S(\mathrm{v}|@) + S(\mathrm{art}|\mathrm{v}) + S(\mathrm{v}|\mathrm{art})^{*} + S(\mathrm{n}|\mathrm{v})^{*}$$

Parameters before learning

		Press	the	left	button	subtotal	total
candidate 1	@	v	art	v	n		
LS		0	0	-0.3*	0	-0.3	-2.38
CS		-0.7	-0.52	-0.7*	-0.16*	-2.08	
candidate 2	@	v	art	n	n		
LS		0	0	-0.7	0	-0.7	-2.92
CS		-0.7	-0.52	-0.3	-0.7	-2.22	
candidate 3	@	v	art	adj	n		
LS		0	0	-0.52*	0	-0.52	-2.42
CS		-0.7	-0.52	-0.52*	-0.16*	-1.90	

Parameters after learning

		Press	the	left	button	subtotal	total
candidate 1	@	v	art	v	n		
LS		0	0	-0.35*	0	-0.35	-2.51
CS		-0.7	-0.52	-0.74*	-0.20*	-2.16	
candidate 2	@	v	art	n	n		
LS		0	0	-0.7	0	-0.7	-2.92
CS		-0.7	-0.52	-0.3	-0.7	-2.22	
candidate 3	@	v	art	adj	n		
LS		0	0	-0.48*	0	-0.48	-2.29
CS		-0.7	-0.52	-0.48*	-0.11*	-1.81	

- LS: Lexical Score
- CS: Context Score
- @: *beginning of sentence* marker

PARAMETER LEARNING (UNSUPERVISED MODE)

EM Training [Dempster 77]

- EM (Expectation and Maximization) algorithm: an unsupervised training process which consists of an expectation step followed by a maximization step.
- There is a many-to-one mapping $\mathbf{x} \rightarrow \mathbf{y}$ from $\mathcal{X}$ to $\mathcal{Y}$.
  - $\mathbf{x}$: is the *complete* data with density $\mathbf{x} \sim f(\mathbf{x}|\, \Phi)$ depending on the parameter set Φ.
  - $\mathbf{y}$: the *incomplete* data with the sampling density $g(\mathbf{y}|\, \Phi)$:

$$g(\mathbf{y}|\, \Phi) = \int_{\mathcal{X}(\mathbf{y})} f(\mathbf{x}|\, \Phi)\, d\mathbf{x}.$$

The EM training procedure in the p-th iteration:

- **E-step**: Estimate the complete-data sufficient statistics $\mathbf{t}(\mathbf{x})$ by finding

$$\mathbf{t}^{(p)} = E\left[\mathbf{t}(\mathbf{x}) \mid \mathbf{y}, \Phi^{(p)}\right].$$

- **M-step**: Determine $\Phi^{(p+1)}$ which maximize $f\left(\mathbf{x}^{(p)}|\, \Phi^{(p+1)}\right)$.

□ Example:

- The complete data $\mathbf{x}$:

  —$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$.
  —cell probabilities: $\left(\frac{1}{2}, \frac{1}{4}\pi, \frac{1}{4}(1-\pi), \frac{1}{4}(1-\pi), \frac{1}{4}\pi\right)$

- The observed (incomplete) data $\mathbf{y}$: 197 animals which are distributed multinomially into 4 categories.

  —$\mathbf{y} = (y_1, y_2, y_3, y_4) = (125, 18, 20, 34)$.
  —$y_1 = x_1 + x_2$, e.g., $(x_1, x_2) = (125, 0), (124, 1), \cdots$.
  —$y_2 = x_2, y_3 = x_3, y_4 = x_4$.
  —cell probabilities: $\left(\frac{1}{2} + \frac{1}{4}\pi, \frac{1}{4}(1-\pi), \frac{1}{4}(1-\pi), \frac{1}{4}\pi\right)$

- To find $\pi^{(p+1)}$ from $\pi^{(p)}$ ($\pi^{(p)}$ denotes the value of π after p iterations):

$$x_1^{(p)} = E\left[X_1 \mid X_1 + X_2 = 125, \pi^{(p)}\right] = 125\frac{\frac{1}{2}}{\frac{1}{2} + \frac{1}{4}\pi^{(p)}};$$

$$x_2^{(p)} = E\left[X_2 \mid X_1 + X_2 = 125, \pi^{(p)}\right] = 125\frac{\frac{1}{4}\pi^{(p)}}{\frac{1}{2} + \frac{1}{4}\pi^{(p)}};$$

$$\pi^{(p+1)} = \frac{x_2^{(p)} + 34}{x_2^{(p)} + 34 + 18 + 20}.$$

- $x_1^{(p)}$ and $x_2^{(p)}$ are usually not integers.
- The reestimation process converge to π^* when $p > 5$, where $\pi^* \approx 0.6268214980$ is the MLE of π.

Viterbi Training [Rabiner 86]

The viterbi training procedure of model parameters includes:

- **Initial Step**: Select an initial parameter set. The initial parameters are usually assumed to be uniformly distributed or have some distribution determined by some priori knowledge.
- **Decoding Process**: Search for the optimal path using the current parameters. Generally, the optimal path is referred to a process that decodes the underlying states, such as the best assignment of part-of-speech sequence for a input word sequence.
- **Maximization Process**: Re-estimate new parameter values from the training data and the decoded states.
- **Repeat**: repeat the Decoding and Maximization Processes until converge.

Viterbi Training: Proof of Convergence

- For the given observation x, let s^* and $\bar{s}$ be the two optimal state sequences corresponding to the parameter sets λ and $\bar{\lambda}$, respectively:

$$s^* = \arg\max_s f(x, s|\lambda)$$
$$\bar{s} = \arg\max_s f\left(x, s|\bar{\lambda}\right).$$

Then

$$\begin{aligned}\max_s f\left(x, s|\bar{\lambda}\right) &\geq f\left(x, s^*|\bar{\lambda}\right) \\ &= \max_{\lambda'} f\left(x, s^*|\lambda'\right) \\ &= \max_{\lambda'}\left[\max_s f\left(x, s|\lambda'\right)\right] \\ &\geq \max_s f(x, s|\lambda).\end{aligned}$$

EM Training vs. Viterbi Training

The comparison of EM and Viterbi training for part-of-speech tagging:

- **Initial Step:**
  Assume that the lexical probability (e.g., P(n|design)) and the contextual probability (e.g., $P(C_i=n|C_{i-1}=art)$) are uniformly distributed.

- **Expectation (EM) / Decoding (Viterbi) Process:**

  — EM: Compute the sufficient statistics:

  (1) the expected No. of transitions from tag t_i to tag t_j, and
  (2) the expected No. of transitions from tag t_i.

  — Viterbi: Use the Viterbi algorithm to tag the corpus with the criterion: $\max_T P(W,T)$

  where W and T correspond to the word and part-of-speech sequences, respectively.

☐ Maximization Process:

— EM: Re-estimate the new parameters by using the method described in M-step, e.g.,

$$\bar{P}\left(t_i \mid t_j\right) = \frac{expected\ no.\ of\ transitions\ from\ tag\ t_i\ to\ tag\ t_j}{expected\ no.\ of\ transitions\ from\ state\ tag\ t_i}$$

— Viterbi: Use the tagged corpus to re-estimate the new parameter values, e.g.,

$$\bar{P}\left(t_i \mid t_j\right) = \frac{total\ no.\ of\ transitions\ from\ tag\ t_i\ to\ tag\ t_j}{total\ no.\ of\ transitions\ from\ state\ tag\ t_i}$$

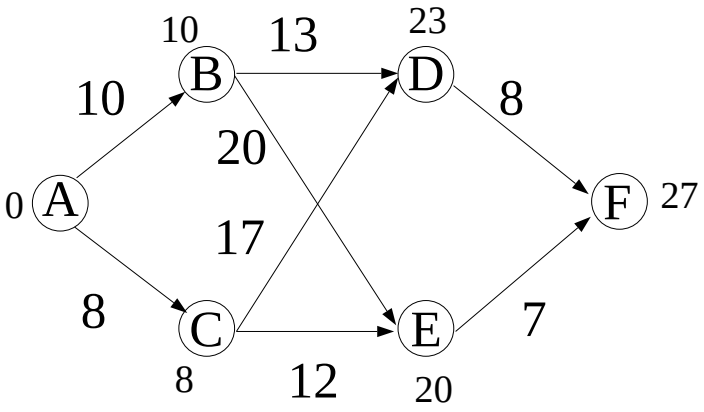
☐ **Loop:** Repeat the Expectation (EM) / Decoding (Viterbi) and the Maximization Processes until converge.

SEARCHING STRATEGY

Dynamic Programming

- An important searching technique in searching a large solution space
- Tasks that are suitable for dynamic programming
  - Tasks that can be resolved in multistage and each stage has a finite number of states only.
  - Once two paths reach the same node, the behavior of these two paths will be the same after that.
  - If a path A-C-E-F is optimal among all the possible paths from A to F, then the path A-C-E is also a local optimal path among all the possible paths from A to E.

- Example of Using Dynamic Programming
  - Chart Parsing
  - Parts of Speech Tagging
  - Find the path with the shortest length



CLASS-BASED MODELING

☐ Goal:

- To reduce the number of parameters such that the parameters can be estimated more reliably.
- To improve statistical language modeling:

  —to provide a partial solution in dealing with the estimation of parameters for unseen events.

☐ Clustering Procedure:

- Define similarity metric
- Cluster data iteratively
- Stop when desired criteria are matched.

☐ Applications:

- Part-of-speech Tagging
- Word sense disambiguation
- Word association
- Machine Translation

CLUSTERING

Dynamic Clustering

- Employs an iterative algorithm to optimize a clustering criterion function.
- At each iteration, data points are assigned to clusters.
- Then, the cluster representatives are updated to reflect any change in the data point assignment.
- The new cluster models are used in the next iteration.
- Continue until a stable partition is obtained.
- The number of clusters is known beforehand.
- Example: K-means clustering

1. Choose the number of classes, K
2. Choose initial $\mu_1, \mu_2, \cdots, \mu_K$.
3. Classify each data x_i to one of the K classes.
4. Recompute the estimates for $\hat{\mu}_i$ using the results of 3.
5. If the $\hat{\mu}_i$ are consistent then STOP; otherwise go to 1, 2, or 3.

Hierarchical Clustering

- Initially, every point in the data set is considered as a separate cluster.
- At any stage of a hierarchical clustering algorithm the two of the existing clusters which are most similar are merged to creat a new cluster, thus reducing the number of potential clusters by one.
- Terminate when the desired criteria are matched.
- The number of clusters is unknown beforehand.
- Examples [Brown 92]
  - Friday Monday Thursday ...
  - people guys folks fellows ...
  - water gas coal liquid acid ...
  - man woman boy girl ...
  - head body hands eyes ...

BOOTSTRAPING

☐ Goal:

- Estimate parameters with a small corpus

☐ Why:

- The performance of supervised learning is better than that of unsupervised learning.
- However, annotating (e.g., tagging) the corpus is a tedious, boring and time consuming task.

Bootstraping With a Seed Corpus

- Use a small annotated corpus as a seed to estimate the initial parameter values.
- Annotate a large untagged corpus with the parameters acquired.
- Use EM or Viterbi training algorithms to re-estimate the parameters, from the large corpora (holding the original seed unchanged).
- Extend the seed to include the newly tagged part.
- Repeat the process until all the untagged instances are tagged.

Bootstraping Without any Seed Corpus

☐ **Approach**:
Suppose we have a random sample $X_1, X_2, \cdots, X_n$ taken from an unknown distribution function $F(\cdot)$. The function to be evaluated is $\theta(F)$.

- Use the random sample to estimate the distribution function by some estimator $\hat{F}$.
- Use $\hat{F}$ to generate new data.
- Estimate $\hat{\theta}$ from the new data.

☐ **Procedures**:

- Randomly select N samples from the small corpus which has N samples.
- Repeat the random sampling process M times. (M is a large number.)
- Get an estimation of the parameter from those M $\times$ N samples.

APPLICATIONS OF CBSO APPROACHES

- ☐ PART-OF-SPEECH TAGGING
- ☐ SYNTACTIC AMBIGUITY RESOLUTION
- ☐ SEMANTIC LANGUAGE MODEL
- ☐ WORD ASSOCIATION
- ☐ MACHINE TRANSLATION
- ☐ SPOKEN LANGUAGE PROCESSING
- ☐ INFORMATION RETRIEVAL
- ☐ BILINGUAL CORPORA ANALYSIS
- ☐ TREEBANK CONVERSION
- ☐ OTHERS (e.g., OCR, OLCR)

PART-OF-SPEECH TAGGING

☐ Tagging Part of Speech

- Model: $P\left(c_{k_1^n} \mid w_1^n\right) = \prod_{i=1}^{n} P\left(c_{k_i} \mid c_{k_1^{i-1}}, w_1^n\right)$.
- Proposed Reduced Model
  - —[Garside87]:
    $P\left(Lex_k \mid w_1^n\right) \approx \prod_{i=1}^{n} \left[P\left(c_{k_i} \mid c_{k_{i-1}}\right) \cdot P\left(c_{k_i} \mid w_i\right)\right]$,
    96% accuracy was reported.
  - —Example: Computer (n) design (n/v) is (be) a (art) hard (adj/adv) task (n).

  =>the score for the most preferred part of speech (POS) sequence "n_1 n_2 be art adj n_3" is: $P\left(n_1 \mid computer\right) \cdot P\left(n_1 \mid BOS\right) \times P\left(n_2 \mid design\right) \cdot P\left(n_2 \mid n_1\right) \times \cdots \times P\left(adj \mid hard\right) \cdot P\left(adj \mid art\right) \times \cdots$ (BOS: begin-of-sentence)
- Procedure:
  - —Take log of $P\left(c_{k_1^n} \mid w_1^n\right)$
  - —Use Dynamic Programming technique to find the best part of speech sequence.

SYNTACTIC AMBIGUITY RESOLUTION

- Basic Model: $P(PSR_1, \cdots, PSR_n \mid catg_1, \cdots, catg_k)$
- Stochastic Grammar [IWPT 89]
- Context Free Model [IWPT 89]
- Context Sensitive Model [Su 88] (with Normalization)
- Others, e.g. Tree Closing algorithm [Garside 87]

Syntactic Structure Disambiguation

Syntactic Score [Su 88]

$$S_{syn} \equiv P\left(Syn, Lex \mid Wrd\right)$$

Decomposition of Syntax Tree

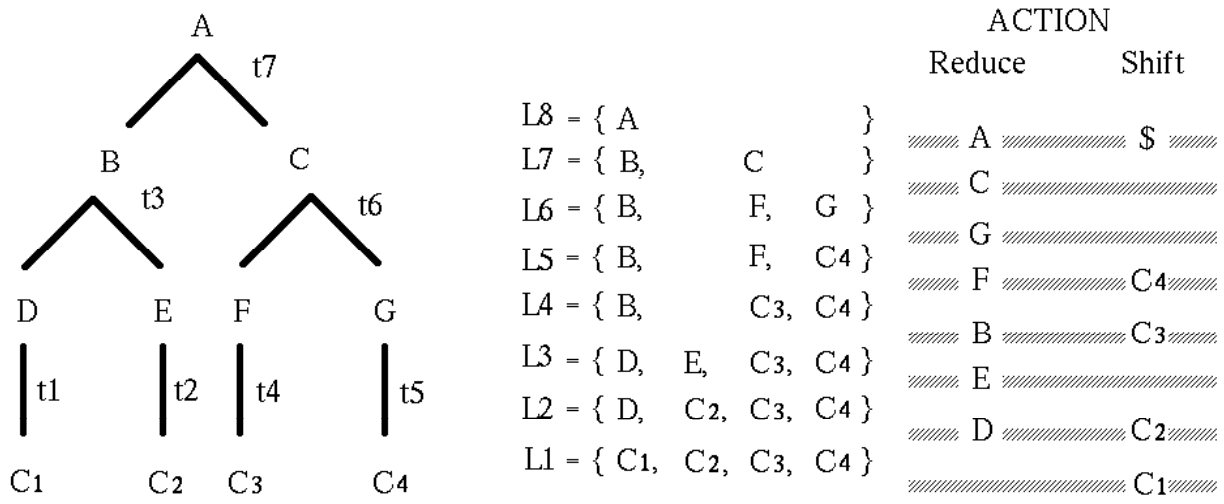


Figure 2 Decomposition of a syntax tree into phrase levels for score computation in bottom–up GLR parser.

☐ Basic Context-Sensitive Formulation

$$
\begin{aligned}
& P\left(L_j | L_1^{j-1}, c_1^n, w_1^n\right) \\
& \quad \approx P\left(L_j | L_{j-1}\right) \\
& \quad \approx P\left(\{l_A, A, r_A\} \,|\, \{l_A, X_1, X_2 \ldots X_m, r_A\}\right)
\end{aligned}
$$

- L_j: the *j*th phrase level
- $L_j : \{l_A A\, r_A\} \leftarrow L_{j-1} : \{l_A X_1 X_2 \cdots X_m r_A\}$
- encode context-sensitivity within context-free framework
- evaluated after each **reduce** action

☐ Example:

$$
\begin{aligned}
& SCORE_{syn}\left(Syn_A\right) \\
& \quad = P\left(L_{8,} L_{7 \ldots} L_2 | L_1\right) \\
& \quad = P\left(L_8 | L_{7 \ldots} L_2, L_1\right) \times P\left(L_7 | L_{6 \ldots} L_1\right) \times \ldots \times P\left(L_2 | L_1\right) \\
& \quad \approx P\left(L_8 | L_7\right) \times P\left(L_7 | L_6\right) \times \ldots \times P\left(L_2 | L_1\right) \\
& \quad \approx P\left(\{A\} | \, \{l_7, B, C, r_7\}\right) \times P\left(\{C\} | \, \{l_6, F, G, r_6\}\right) \times \ldots \times P\left(\{D\} | \, \{l_1, c_1, r_1\}\right)
\end{aligned}
$$

- Run-Time Formulation (Normalization Form)
  - compact multiple highly correlated phrase levels when evaluating score
  - evaluated after each **shift** action
  - avoid “normalization problem” (same input with different number of transition probabilities)

- Example:

$$
\begin{aligned}
&SCORE_{syn}\left(Syn_A\right) \\
&\quad = P\left(L_{8,}L_{7...}L_2|L_1\right) \\
&\quad = P\left(L_{8,}L_{7,}L_6|L_{5,}L_{4...}L_1\right) \times P\left(L_5|L_{4,}L_{3...}L_1\right) \times P\left(L_{4,}L_3|L_{2,}L_1\right) \times P\left(L_2|L_1\right) \\
&\quad \approx P\left(L_{8,}L_{7,}L_6|L_5\right) \times P\left(L_5|L_4\right) \times P\left(L_{4,}L_3|L_2\right) \times P\left(L_2|L_1\right) \\
&\quad \approx P\left(L_8|L_5\right) \times P\left(L_5|L_4\right) \times P\left(L_4|L_2\right) \times P\left(L_2|L_1\right)
\end{aligned}
$$

SEMANTIC LANGUAGE MODEL

- Preliminary result on PP-attachment problem [Liu 89, 90]
  - Find attachment of "V N PP"
  - About 95% was observed in a limited domain.

- Word Sense Disambiguation
  - Deciding word sense based on simple contextual information [Brown 91].
  - Resolving lexical ambiguities in one language using statistical data on lexical relations in another language [Dagan 91].

- Semantic N-tuple Model [Chang 90, 92]
  - Applying selectional restrictions jointly on collocationally related word N-tuples.

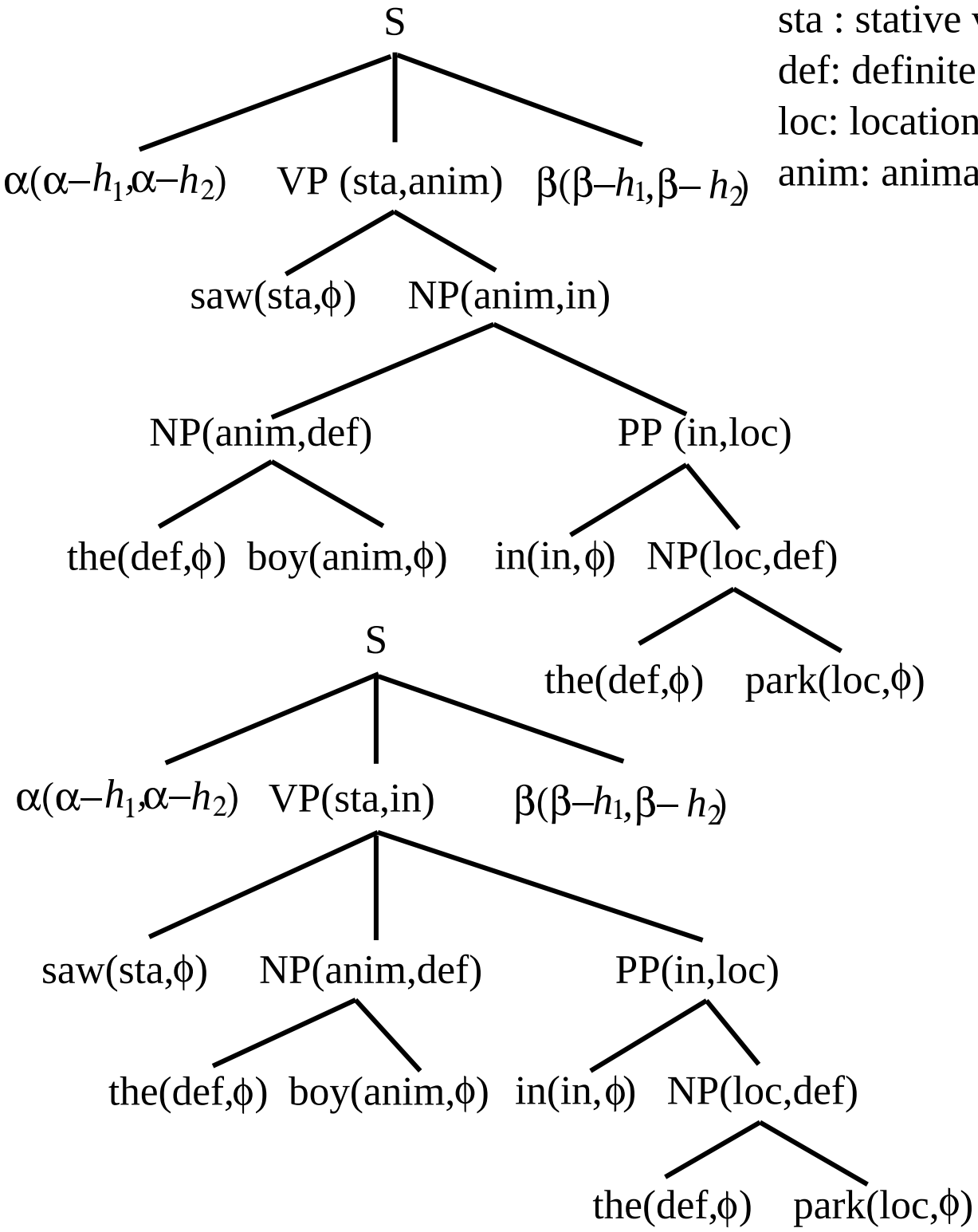
Semantic N-tuple Model

☐ Semantic Score

$$
\begin{aligned}
S_{sem} &\equiv P\left(Sem \mid Syn, Lex, Words\right) \\
&\equiv P\left(\Gamma_1^m \mid L_1^m, c_1^n, w_1^n\right) \\
&= \prod_i P\left(\Gamma_i | \Gamma_1^{i-1}, L_1^m, c_1^n, w_1^n\right) \\
&\approx \prod_i P\left(\Gamma_i | \Gamma_{i-1}\right) \\
&\approx \prod_i P\left(\left\{\bar{l}_A, A\left(S_1^N\right), \bar{r}_A\right\} \mid \left\{\bar{l}_A, \bar{X}_1, \bar{X}_2 \ldots \bar{X}_M, \bar{r}_A\right\}\right)
\end{aligned}
$$

☐ Formulation

- Γ_i (annotated phrase level): the semantic information is encoded as a form of annotation to the phrase levels
- the annotation is the feature structure associated with each constituent in the phrase level.
- a semantic N-tuple (S_1, S_2, ..., S_N) is used as the feature structure of constituent A.
- S_j is the "head feature" (the 1st feature) of the "*j*th head" of the current node A.



sta : stative verb
def: definite article
loc: location
anim: animate

WORD ASSOCIATION

- Word Association [Church 89]
  - Procedures
    - —Select a window with length n
    - —Evaluate Mutual Information of any word pair (w_x , w_y) by shifting the window across the corpus.
  - Application
    - —Lexicography : Collins English Dictionary
    - —OCR, Spelling Correction, e.g. {Financial Form} versus {Financial Farm}

PROBABILISTIC TRANSLATION MODEL

- Translation problem is an optimization problem
- Let
  - $CW_1^{n_C}$: Chinese sentence of length n_C
  - $EW_1^{n_E}$: English sentence of length n_E
  - *CLM*: Chinese Language Model
  - *TM*: Transfer Model from English to Chinese
  - *ELM*: English Language Model
  - I_C: Intermediate Representation of Chinese (i.e., normalized Chinese annotated syntax tree)
  - I_E: Intermediate Representation of English (i.e., normalized English annotated syntax tree)

- Find the mapping from English to Chinese is equivalent to finding $CW_1^{n_C}$ that maximizes the *translation score*:

$$
\begin{aligned}
& P\left(CW_1^{n_C} \mid EW_1^{n_E}, ELM, TM, CLM\right) \\
&= \sum\sum P\left(CW_1^{n_C}, I_C, I_E \mid EW_1^{n_E}, ELM, TM, CLM\right) \\
&= \sum\sum P\left(CW_1^{n_C} \mid I_C, I_E, CLM, \cdots\right) \\
&\quad \times P\left(I_C \mid I_E, TM, EW_1^{n_E}, \cdots\right) \\
&\quad \times P\left(I_E \mid EW_1^{n_E}, ELM, TM, CLM\right) \\
&\approx \sum\sum P\left(CW_1^{n_C} \mid I_C, CLM\right) && (generation\ phase) \\
&\quad \times P\left(I_C \mid I_E, TM\right) && (transfer\ phase) \\
&\quad \times P\left(I_E \mid EW_1^{n_E}, ELM\right) && (analysis\ phase) \\
&= \sum\sum P_G\left(.\right) \times P_T\left(.\right) \times P_A\left(.\right)
\end{aligned}
$$

- Searching Strategy: to find the desirable candidates of $CW_1^{n_C}$, we can achieve our goal in three separate phases.
  - Analysis: find the intermediate representation of English sentence that maximizes the *analysis score*: $P_A(\cdot)$
  - $I_{Emax} \stackrel{\Delta}{=} \arg\max_{I_E} P\left(I_E \mid EW_1^{n_E}, ELM\right)$
  - Transfer: find the intermediate representation of Chinese sentence that maximizes the *transfer score*: $P_T(\cdot)$
  - $I_{Cmax} \stackrel{\Delta}{=} \arg\max_{I_C} P\left(I_C \mid I_{E_{max}}, TM\right)$
  - Generation: find the Chinese sentence that maximizes the *generation score*: $P_G(\cdot)$
  - $CW_{max} \stackrel{\Delta}{=} \arg\max_{CW_1^{n_C}} P\left(CW_1^{n_C} \mid I_{C_{max}}, CLM\right)$
- In each phase, the Best-N candidates can be selected for further processing.
- The probabilistic translation model provides a standard and systematic method to improve the system in a predictable way.

SPOKEN LANGUAGE PROCESSING

☐ The criterion for a Chinese phonetic typewriter: to find the character sequence $\widehat{Chr}$ such that

$$\begin{aligned}\widehat{Chr} &\equiv \underset{Chr_m}{\operatorname{argmax}}\{\, P\,(Chr_m \mid Acu)\} \\ &= \underset{Chr_m}{\operatorname{argmax}}\left\{\sum_k \sum_l \sum_n P\,(Lex_k, Wrd_l, Chr_m, Syl_n \mid Acu)\right\} \\ &\approx \underset{Chr_m}{\operatorname{argmax}}\{\sum_k \sum_l \sum_n P\,(Lex_k \mid Wrd_l) \times P\,(Wrd_l \mid Chr_m) \\ &\qquad \times P\,(Chr_m \mid Syl_n) \times P\,(Syl_n \mid Acu)\}\end{aligned}$$

- Acu: the input acoustic signals.
- Syl_n: the n-th syllable sequence.
- Chr_m: the m-th character sequence.
- Wrd_l: the l-th word sequence.
- Lex_k the k-th lexical (or part of speech) sequence.

☐ $P\,(Lex_k, Wrd_l, Chr_m, Syl_n \mid Acu)$ is called the *integrated score function.*

☐ The Computational Model:

$$\widehat{Chr} \equiv \underset{Chr_m}{\operatorname{argmax}}\{w_{lex} \cdot \log P\left(Lex_k \mid Wrd_l\right) + w_{wrd} \cdot \log P\left(Wrd_l \mid Chr_m\right) + w_{chr} \cdot \log P\left(Chr_m \mid Syl_n\right) + w_{syl} \cdot \log P\left(Syl_n \mid Acu\right)\},$$

- $w_{lex}, w_{wrd}, w_{chr}, w_{syl}$ denotes the lexical, word, character and syllable weights, respectively.
- The reasons for assigning different weights to the various parameters (log probabilities):
  - —Different parameters contribute differently to the overall discrimination power; therefore, they should be emphasized (weighted) appropriately.
  - —The dynamic ranges of the parameters in different modules varying in a wide range should be compensated (normalized).
- The values of the weights are trained via the robust learning procedure.

On-Line Character Recognition (OLCR)

- The criterion for a OLCR system:
  to find the character $\hat{c}$ satisfying:

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} P\left(c_i | s_1^n\right)$$
$$= \underset{c_i}{\operatorname{argmax}} \left\{ \sum_j P\left(c_i, T_{i,j} | s_1^n\right) \right\},$$

  - $s_1^n \left(= s_1, s_2, \cdots, s_n\right)$ represents n successive segments corresponding to the hand-written character.
  - $T_{i,j}$ is the j-th template associated with c_i.

- The computational model:to find the character $\hat{c}$ such that

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} \left\{ \max_j f\left(s_1^n | c_i, T_{i,j}\right) P\left(c_i, T_{i,j}\right) \right\}.$$

CONCLUSIONS

- NLP needs a huge amount of knowledge. However, it is very difficult to acquire and manipulate the information by human. Automatic learning seems to be the only way to go.
- The cooperative approach is recommended:
  - Human is competent in establishing high level linguistic models, but awkward in dealing with a large amount of fine-grained knowledge consistently and cost-effectively.
  - Machines are good at processing massive data, but have difficulty to induce concepts and models.
  - Statistics theories provide well-established techniques for systematically processing corpus.
- CBSO approach:
  - Setup probabilistic language models by the human.
  - Estimate the parameters by machines.
- Combining the deduction capability of linguists and induction capability of the machine is probably the best way to make large scale MT/NLP systems feasible.
- Corpus-Based Statistics-Oriented approaches will play an important role in knowledge acquisition for large MT/NLP systems.

BIBLIOGRAPHY

[Amari 67] Amari, S., *"A theory of adaptive pattern classifiers,"* IEEE Trans. on Electronic Computers, Vol. EC-16, pp. 299-307, June 1967

[Bahl 83] Bahl, L. R., F. Jelinek, and R. Mercer., *"A Maximum Likelihood Approach to Continuous Speech Recognition,"* IEEE Trans. on Pattern Analysis and Machine Intelligence, , Vol. PAMI-5, No. 2, pp. 179-190, March 1993.

[Breiman 84] Breiman, L., J. H. Friedman, R.A. Olshen and C. J. Stone, *"Classification And Regression Trees,"* Wadsworth Inc., CA, USA, 1984.

[Brown 90] Brown, P. et al., *"A Statistical Approach to Machine Translation,"* Computational Linguistics, vol. 16, no. 2, pp. 79-85, June 1990.

[Brown 91] Brown, P. et al., *"Word-Sense Disambiguation Using Statistical Methods,"* Proceedings of 29th Annual Meeting of the Association for Computational Linguistics, pp. 264-270, California, USA, June 18-21, 1991.

[Brown 92] Brown, P. et al., *"Class-Based n-gram Models of Natural Language,"* Computational Linguistics, vol. 18, no. 4, pp. 467-479, 1992.

[Chang 92] Chang, J.-S., Y.-F. Luo and K.-Y. Su, *"GPSM: A Generalized Probabilistic Semantic Model for Ambiguity Resolution,"* Proceedings of ACL-92, pp. 177–184, 30th Annual Meeting of the Association for Computational Linguistics, University of Delaware, Newark, DE, USA, 28 June–2 July, 1992.

[Chang 93] Chang, J.-S. and K.-Y. Su, *"A Corpus-Based Statistics-Oriented Transfer and Generation Model for Machine Translation,"* Proceedings of TMI-93, pp. 3-14, 1993.

[Chen 91] Chen, S.-C., J.-S. Chang, J.-N. Wang and K.-Y. Su, *"ArchTran: A Corpus-Based Statistics-Oriented English-Chinese Machine Translation System,"* Proceedings of Machine Translation Summit III, pp. 33–40, Washington, D.C., USA, July 1–4, 1991.

[Chiang 92a] Chiang, T.-H., Y.-C. Lin and K.-Y. Su, *"Syntactic Ambiguity Resolution Using A Discrimination and Robustness Oriented Adaptive Learning Algorithm",* Proceedings of COLING-92, vol. I, pp. 352–358, 14th Int. Conference on Computational Linguistics, Nantes, France, July 23–28, 1992.

[Chiang 92b] Tung-Hui Chiang, Jing-Shin Chang, Ming-Yu Lin and Keh-Yih Su, *"Statistical Models for Word Segmentation and Unknown Word Resolution,"* Proceedings of ROCLING-V, pp. 123-146, Taipei, Taiwan, R.O.C., 1992.

[Church 88] Church, K., *"A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text,"* ACL Proc. 2nd Conf. on Applied Natural Language Processing, pp. 136-143, Austin, Texas, USA, 9-12 Feb. 1988.

[Church 89] Church, K. and P. Hanks, *"Word Association Norms, Mutual Information, and Lexicography,"* Proc. 27th Annual Meeting of the ACL, pp. 76-83, University of British Columbia, Vancouver, British Columbia, Canada, 26-29 June 1989.

[Dempster 77] Dempster A. P., N. M. Laird and D. H. Rubin, *"Maximum Likelihood from Incomplete Data via the EM Algorithm,"* Journal of the Royal Statistical Society, 39(B), 1–38, 1977.

[DeRose 88] DeRose, Steven. J., *"Grammatical Category Disambiguation by Statistical Optimization,"* Computational Linguistics, vol. 14, no. 1, pp. 31-39, 1988.

[Devijver 82] Devijver, P.A., and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall, London, 1982.

[Dagan 91] Dagan I. A. Itai, U. Schwall, *"Two Language Are More Informative Than One,"* Proceedings of 29th Annual Meeting of the Association for Computational Linguistics, pp. 130-137, California, USA, June 18–21, 1991.

[Duda 73] Duda, O. R., P. E. Hart, *Pattern Classification and Scene Analysis*, Hohn Wiley and Sons, Inc., USA, 1973.

[Garside 87] Garside, Roger, Geoffrey Leech and Geoffrey Sampson (eds.), *The Computational Analysis of English: A Corpus-Based Approach*, Longman Inc., New York, 1987.

[Good 53] Good, I. J., *"The population frequencies of species and the estimation of population parameters,"* Biometrika, vol.40, no. 3, 4, pp. 237–264, 1953.

[Hoel 71a] Hoel, P. G., S. C. Port, C. J. Stone, *"Introduction to Probability Theory,"* Houghtone Mifflin Company, USA. 1971.

[Hoel 71b] Hoel, P. G., S. C. Port, C. J. Stone, *"Introduction to Statistical Theory,"* Houghtone Mifflin Company, USA. 1971.

[Hoel 72] Hoel, P. G., S. C. Port, C. J. Stone, "*"Introduction to Stochastic Process,"* Houghtone Mifflin Company, USA. 1971.

[IWPT 89] Proceedings of International Workshop on Parsing Technologies (IWPT-89) CMU, Pittsburgh, PA, USA, August 28-31, 1989.

[Katagiri 91] Katagiri, S., C. H. Lee, and B. H. Juang, *"New Discriminative Training Algorithm Based on the Generalized Probabilistic Decent Method,"* Proceedings of 1991 IEEE Workshop Neural Networks for Signal Processing, pp. 299-308, Piscataway, NJ, Aug. 1991.

[Katz 87] Katz, S. M., *"Estimation of Probabilities From Sparse Data for the Language Model Component of a Speech Recognizer,"* IEEE Transactions on Acoustics, Speech and Signal Processing, No. 35, pp. 400–401, 1987.

[Liu 90] Liu, C.-L., J.-S. Chang and K.-Y. Su, *"The Semantic Score Approach to the Disambiguation of PP Attachment Problem,"* Proceedings of ROCLING-III, pp. 253-270, National Tsing-Hua Univ., Taipei, R.O.C., Sept. 21-23, 1990.

[Papoulis 84] Papoulis A., " *Probability, Random Variable, and Stochastic Processes*, 2nd Ed. McGraw-Hill, USA. 1984.

[Rabiner 86] Rabiner, L. R., J. G. Wilpon, and B. H. Juang, *"A Segmental k-Means Training Procedure for Connected Word Recognition,"* AT&T Tech. Journal, Vol. 65, No. 3, pp.21-31, May-June 1986.

[Rabiner 89] Rabiner, L. R., *"A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,"* Proceeding of IEEE, pp.1404-1413, Vol. 77, No. 2, Feb. 1989.

[Schalkoof 92] Schalkoof, R., *"Pattern Recognition: statistical, structural and neural approaches,"* John Wiley & Sons, Inc. Singapore 1992.

[Su 88a] Su, K.-Y. and J.-S. Chang, "*Semantic and Syntactic Aspects of Score Function,*" Proc. of COLING-88, vol. 2, pp. 642–644, 12th Int. Conf. on Computational Linguistics, Budapest, Hungary, August 22–27, 1988.

[Su 89a] Su, K.-Y., J.-N., Wang, M.-H. Su and J.-S. Chang, "*A Sequential Truncation Parsing Algorithm Based on the Score Function,*" Proceedings of International Workshop on Parsing Technologies (IWPT-89), pp. 95–104, CMU, Pittsburgh, PA, USA, August 28-31, 1989.

[Su 89b] Su, K.-Y., M.-H. Su and L.-M. Kuan., *"Smoothing Statistic Databases in a Machine Translation System,"* Proceedings of ROCLING-II, pp. 333-347, Academia Sinica, Taipei, Taiwan, R.O.C., Sept. 22-24, 1989.

[Su 90] Su, K.-Y. and J.-S. Chang, *"Some Key Issues in Designing MT Systems,"* Machine Translation, vol. 5, no. 4, pp. 265-300, 1990.

[Su 91a] Su, K.-Y. and C.-H. Lee, *"Robustness and Discrimination Oriented Speech Recognition Using Weighted HMM and Subspace Projection Approach,"* Proceedings of IEEE ICASSP-91, vol. 1, pp. 541-544, Toronto, Ontario, Canada. May 14-17, 1991.

[Su 91b] Su, K.-Y., J.-N. Wang, M.-H. Su and J.-S. Chang, *"GLR Parsing with Scoring,"* In M. Tomita (ed.), Generalized LR Parsing, Chapter 7, pp. 93-112, Kluwer Academic Publishers, 1991.

[Su 92] Su, K.-Y and J.-S. Chang, *"Why Corpus-Based Statistics-Oriented Machine Translation,"* Proceedings of TMI-92, pp. 249–262, 4th Int. Conf. on Theoretical and Methodological Issues in Machine Translation, Montreal, Canada, June 25–27, 1992.

[Su 92b] Su, K.-Y., M.-W. Wu and J.-S. Chang, *"A New Quantitative Quality Measure for Machine Translation Systems,"* Proceedings of COLING-92, vol. II, pp. 433–439, 14th Int. Conference on Computational Linguistics, Nantes, France, July 23–28, 1992.

[Su 94] Su, K. Y., and C. H. Lee, *"Speech Recognition Using Weighted HMM and Subspace Projection Approaches,"* IEEE Trans. on speech and audio processing, Vol. 2, No. 1, pp. 69-79, Jan. 1994.