[Day-1] Introduction to Statistical Natural Language Processing

(Part IV: Typical Applications)

Keh-Yih Su and Jing-Shin Chang
kysu@bdc.com.tw, shin@nlp.csie.ncnu.edu.tw

(2002/8/17)

Behavior Design Corporation (BDC)
No. 5, 2F, Industrial East Road IV, Science-Based Industrial Park
Hsinchu, Taiwan 30077, R.O.C.

Department of Computer Science and Information Engineering
National Chi-Nan University
Puli, Nantou, Taiwan 545, R.O.C.

Day-1: Introduction to Statistical Natural Language Processing (mainly on Supervised Learning)

- Part I: Introduction (1)
  - Problems and Characteristics of Natural Language Processing
- Part II: Introduction (2)
  - What, When and Why Statistical Approach
- Part III: Basic Concepts and Background
  - Feature Space, Probability, Estimator, Stochastic Process, Data Set Classification, and Performance Measure
- **Part IV: Typical Applications**
  - **Word Segmentation, Tagging, Selecting Parse Tree, Aligning Bilingual Corpus**
- Part V: Techniques for Improving Performance
  - Smoothing, Class-Based Model, Adaptive Learning, Tips for Checking
- Part VI: Advanced Topics
  - Support Vector Machine, Maximum Entropy Models
- Appendix: Related Techniques
  - Parameter Estimation, Fractional Factorial Experiment Design, Decision Tree

Part IV: Typical Applications (Just some examples, not the full overview)

- Word Segmentation
  - Find the best token sequence for a given sentence
- Part of Speech Tagging
  - Find the best part-of-speech sequence for a given token sequence
- Parse Tree Selection (Structure Disambiguation)
  - Obtain the most appropriate syntactic relation for a given token sequence
- Bilingual Corpus Alignment
  - Search the best mapping between two sentence sequences of two languages

Word Segmentation Heuristics Approaches

- Matching Against Lexicons
  - Scan left-to-right (or right-to-left)
- Heuristic Matching Criteria
  - (1) Longest (Maximal) Match
    - Select the longest sub-string on multiple matches
  - (2) Minimum number of matches
    - Select the segmentation patterns with smallest number of words
  - (3) Combination of heuristic rules

Probabilistic Word Segmentation Models [Chiang et al., 1992]

- Bayesian Decision Rule:
  - Find the best segmentation pattern S*,

$$S^*(V) = \arg\max_{S_j} P(S_j = w_{j,1}^{j,m(j)} \mid c_1^n, V, \Lambda)$$

which maximizes the following likelihood function of the input corpus

$$P(S_j = w_{j,1}^{j,m(j)} \mid c_1^n, V, \Lambda)$$

  - c_1^n : input characters c_1, c_1, ..., c_n
  - S_j : j-th segmentation pattern, consisting of words { $w_{j,1}$, $w_{j,2}$, ..., $w_{j,mj}$ }
    - V: vocabulary (dictionary entries)
    - Λ: parameters (probabilities)
    - S*(V): the best segmentation (is a function of V)

Word Segmentation Features

- Adopted Features for Generalized WS Model
  - c_1^n : input characters c_1, c_1, ..., c_n
  - n: number of characters (length in characters)
  - S_j: j-th segmentation pattern, consisting of words { $w_{j,1}$, $w_{j,2}$, ..., $w_{j,mj}$ }
  - m_j: number of words in segmentation
  - l_{ji}: length of each words
  - t_{ji}: tag (part of speech) of each words

Generalized Probabilistic Word Segmentation Models [Chiang et al., 1992]

- Feature dependencies

$$P(\vec{L}_i, \vec{W}_i, m_i \mid c_1^n, n) = P(l_{i,1}^{i,m_i}, w_{i,1}^{i,m_i}, m_i \mid c_1^n, n) \equiv P_i(l_1^m, w_1^m, m \mid c_1^n, n)$$
$$= P_i(l_1^m, w_1^m \mid m, c_1^n, n) \times P_i(m \mid c_1^n, n)$$
$$= \prod_{k=1,m_i} P_i(l_k, w_k \mid l_1^{k-1}, w_1^{k-1}, m, c_1^n, n) \times P_i(m \mid c_1^n, n)$$
$$= \prod_{k=1,m_i} P_i(l_k \mid w_k, l_1^{k-1}, w_1^{k-1}, m, c_1^n, n) \times P_i(w_k \mid l_1^{k-1}, w_1^{k-1}, m, c_1^n, n) \times P_i(m \mid c_1^n, n)$$

- Simplification
  - Use log-scaled probability scores to avoid underflow

$$\arg\max_i P(\vec{W}_i, \vec{L}_i, m_i \mid c_1^n, n)$$
$$\equiv \arg\max_i \prod_{k=1,m_i} P_i(l_k \mid l_{k-1}) \times P_i(w_k \mid l_{k-1}) \times P_i(m \mid n)$$
$$= \arg\max_i \sum_{k=1,m_i} \log P_i(l_k \mid l_{k-1}) + \log P_i(w_k \mid l_{k-1}) + \log P_i(m \mid n)$$

Derived Segmentation Models

$$\arg\max_i P(\vec{W}_i, \vec{L}_i, m_i \mid c_1^n, n)$$
$$\equiv \arg\max_i \begin{cases} \sum_{k=1,m_i} \log P_i(w_k) & (M1) \\ \sum_{k=1,m_i} \log P_i(l_k \mid l_{k-1}) & (M2) \\ \sum_{k=1,m_i} \log P_i(m \mid n) & (M3) \\ \sum_{k=1,m_i} \log P_i(w_k \mid l_{k-1}) & (M4) \end{cases}$$

- M1: word unigram model (most frequently used)
- M2: length transition
- M3: length/word count correlation
- M4: word uni-gram with preceding length information
- Word bigram is not tried, as Backoff smoothing strategy is not known at that time

Testing Environment

Model	Number of Parameters	Model	Number of Parameters
P(Lk|Lk-1)	40	P(Tk|Tk-1)	625
P(m|n)	229	P(W)*P(Tk|Tk-1)	9,755 + 625
P(Wk)	9,755	P(W|L)*P(Tk|Tk-1)	14,437 + 625
P(Wk|LK-1)	14,473	P(W|T)*P(Tk|Tk-1)	10,231 + 625
Training Set	41599 words / 5608 sentences		
Testing Set	10134 words / 1402 sentences		
Dictionary	99441 entries		
Lexical Tags	22 parts of speech & 3 special tags		
Ambiguity	8.6 candidates / sentences (both training set & testing set)		

Table 7 Testing Environment

Baseline Performance WITHOUT Unknown Words [Chiang et al., 1992]

- Baseline Performance WITHOUT Unknown Words
  - Even the simple unigram model gives good result

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	Sentence (%)	word (%)	sentence (%)
Max Match-1	1.14 (98.86)	4.05 (95.95)	1.22 (98.78)	4.07 (95.93)
Max Match-2	1.14 (98.86)	4.07 (95.93)	1.12 (98.88)	3.78 (96.22)
P(Lk|Lk-1)	6.16 (93.84)	37.57 (62.43)	6.82 (93.18)	40.09 (59.91)
P(m|n)	5.24 (94.76)	28.53 (71.74)	5.71 (94.29)	29.60 (70.40)
P(Wk)	0.54 (99.46)	2.07 (97.93)	0.76 (99.24)	2.50 (97.50)
P(Wk|LK-1)	0.47 (99.53)	1.77 (98.23)	0.73 (99.27)	2.50 (97.50)

Table 2 Baseline Performance WITHOUT Unknown Words

Baseline Performance WITH Unknown Words [Chiang et al., 1992]

- Baseline Performance WITH Unknown Words
  - Unknown word is the main problem in word segmentation

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	Sentence (%)	word (%)	sentence (%)
Max Match-1	4.01 (95.99)	20.74 (79.26)	4.23 (95.77)	20.68 (79.32)
Max Match-2	4.01 (95.99)	20.77 (79.23)	4.15 (95.85)	20.54 (79.46)
P(Lk|Lk-1)	8.70 (91.30)	45.54 (54.46)	9.41 (90.59)	47.86 (52.14)
P(m|n)	7.19 (92.81)	38.61 (61.39)	7.82 (92.18)	39.30 (60.70)
P(Wk)	3.62 (96.38)	19.81 (80.19)	3.94 (96.06)	19.97 (80.03)
P(Wk|LK-1)	3.68 (96.32)	20.08 (79.92)	4.07 (95.93)	21.04 (78.96)
(*) The numbers in the parentheses Performance WITH Unknown Words				

Table 1 Baseline Performance WITH Unknown Words

Performance of Adaptive Learning WITHOUT Unknown Words [Chiang et al., 1992]

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	Sentence (%)	word (%)	sentence (%)
P(Lk|Lk-1)	1.20 (98.80)	4.65 (95.35)	1.19 (98.81)	4.14 (95.86)
P(m|n)	1.26 (98.74)	4.99 (95.01)	1.23 (98.77)	4.21 (95.79)
P(Wk)	0.38 (99.62)	1.60 (98.40)	0.68 (99.32)	2.50 (97.50)
P(Wk|LK-1)	0.11 (99.89)	0.48 (99.52)	0.61 (99.39)	2.35 (97.65)

Table 4 Performance WITHOUT Unknown Words after LEARNING

Performance of Adaptive Learning WITH Unknown Words [Chiang et al., 1992]

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	Sentence (%)	word (%)	sentence (%)
P(Lk|Lk-1)	4.17 (95.83)	21.33 (78.67)	4.37 (95.63)	21.33 (78.67)
P(m|n)	4.33 (95.67)	22.18 (77.82)	4.43 (95.57)	21.47 (78.53)
P(Wk)	3.28 (96.72)	18.79 (81.21)	3.84 (96.16)	20.26 (79.74)
P(Wk|LK-1)	3.23 (96.77)	18.28 (81.72)	4.00 (96.00)	21.04 (78.96)

Table 3 Performance WITH Unknown Words after LEARNING

Unknown Word Problems in Word Segmentation

- Unknown Word Problems:
  - Over segmentation: segment into individual single characters
  - Word Competition (搶詞問題)
    - Unknown words are mis-merged as part of known words
    - Example, "土地 公有 政策" =WS Error ('公有' unknown)=> "土地公 有 政策"
- Unknown Word Detection:
  - Determine position and length (Lu) of unknown word (Wu) in suspect unknown word regions (assuming one unknown word per region)
  - Segmentation Score (Model 4):

$$Score = \cdots P(W_k = w_u \mid l_{k-1}) \times P(w_{k+1} \mid L_k = l_u) \times \cdots$$

$$P_u\left(c_i^j \text{ contains an unknown word of length } l_u \text{ at position } k \mid c_1^n\right)$$
$$\approx P\left(L_{uwr}\right) \times P\left(w_u \in c_i^{i+Luwr-1} \mid L_{uwr}\right) \times P\left(l_u \mid w_u \in c_i^{i+Luwr-1}\right)$$

Performance for Unknown Word Resolution

- Initial error rate (without learning) is raised significantly due to expansion of error patterns.
- Error rate reduction is significantly improved after adaptive learning is applied to the error-correcting model. (Compare with Table 1 & 3.)

Model	Training Set Error (*Accuracy)		Testing Set Error (*Accuracy)	
	word (%)	Sentence (%)	word (%)	sentence (%)
before learning	38.06 (61.94)	85.04 (14.96)	39.64 (60.36)	86.38 (13.62)
after learning	1.78 (98.22)	8.35 (91.65)	3.59 (96.41)	15.26 (84.74)

Table 6 Performance for Unknown Word Resolution (Baseline and Learning for 10 iterations)

PART-OF-SPEECH TAGGING

- POS Tagging Model:

$$\vec{C} = \arg\max_{c_1^n} P\left(c_1^n \mid w_1^n\right) \equiv \arg\max_{c_1^n} \prod_{i=1}^{n} P\left(c_i \mid c_1^{i-1}, w_1^n\right)$$

- Trigram Tagging Model [Garside87, Church 88]:
  - Formulation

$$C_1^n = \arg\max_{c_1^n} P\left(c_1^n \mid w_1^n\right) \equiv \arg\max_{c_1^n} \prod_{i=1}^{n} P\left(c_i \mid c_{i-1}, c_{i-2}\right) P\left(w_i \mid c_i\right)$$

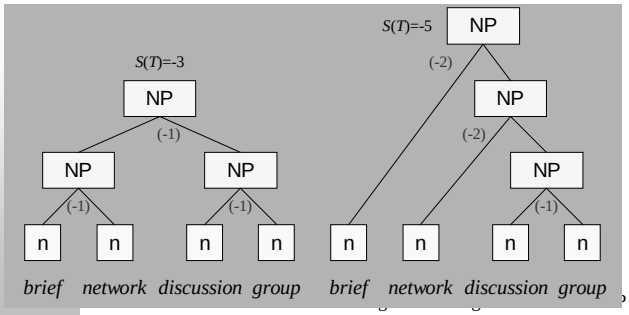
  - 96% accuracy was reported.

Stochastic Context-free Grammar (SCFG)

- Formulation

$$P(T) \approx \prod_{A \to \alpha \in T} P(\alpha \mid A), \qquad \sum_{\alpha : A \to \alpha \in G} P(\alpha \mid A) = 1, \forall A \in N$$

$$S(T) \equiv \sum_{A \to \alpha \in T} S(\alpha \mid A), \text{ where } S(\alpha \mid A) = \log P(\alpha \mid A)$$



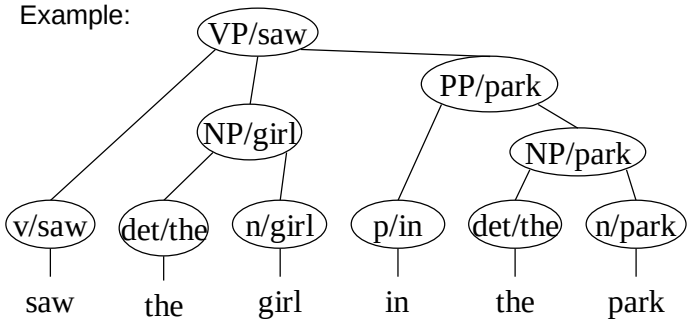
NP → n n (-1)
NP → NP n (-3)
NP → n NP (-2)
NP → NP NP (-1)

Problems with SCFG

- Context Sensitivity Issues ⇔ Context Sensitive Model
  - SCFG: rewriting rules in a context free manner does not imply that the preferences to rules (stochastic behavior of rewriting) is context free
- Normalization Issues ⇔ Token Synchronized Parsing
  - SCFG tends to assign a higher score to a parse-tree with simpler syntactic structure (with fewer number of nodes, i.e., fewer number of rule applications)
  - Such preference is completely irrelevant to linguistic interpretation
- Semantic Issues ⇔ Lexicalized Stochastic Parsers
  - Purely syntactic constrains do not resolve certain ambiguity well
  - Semantic information is required to provide better discrimination power
  - Head lexicon provides the main knowledge source about the semantics of a constituent

Lexicalized SCFG/PCFG [Collins 97, Charniak 97,00]

- Lexicalized SCFG
  - An extension of PCFG with Lexicalized LHS and RHS symbols
  - Lexicalized with one (and only one) head word that best characterizes the constituent rooted at a non-terminal
  - Lexical head was percolated from terminal nodes
- Example:



Lexicalized SCFG/PCFG [Collins 97, Charniak 97,00]

- Parsing Model
  - Generative model: conceptually, top-down generation process
  - Head-driven parsing, middle-out: not strictly left-to-right
  - For each non-terminal node in a parse, predict the head tag and head word first, then predict left and right siblings of the head child
  - Conditioning almost all predictions based on the head of the constituent
- Performance [Charniak 00, 01]
  - Best performed over the Penn Wall Street Journal treebank
    - In terms of labeled precision, recall, cross-brackets (not sentence accuracy)

Lexicalized Models for Statistical Parsing [Collins 97]

- The most widely cited recent work on lexicalized SCFG, with good performance on Penn WSJ Treebank.
- Parsing Model:

$$T_{best} = \arg\max_{T} P(T \mid S) = \arg\max_{T} P(T, S)$$
$$P(T \mid S) = \prod_{i} P(RHS_i \mid LHS_i) = \prod_{i} P(LHS_i \rightarrow RHS_i)$$

- Head Annotated Production Rule:

$$C(h) \rightarrow \Delta L_n(l_n) \cdots L_1(l_1) H(h) R_1(r_1) \cdots R_m(r_m) \Delta$$

  - C: phrase/constituent/chunk under consideration
  - H: head child of C
  - h: head = <head-word, head-tag> pair
  - L,R: Left/Right modifiers, Δ: stop symbol

Lexicalized Models for Statistical Parsing [Collins 97]

- Decomposition: Model-I (zero-th order Markov assumption)
  - Generating Head constituent label with probability

$$P_H(H \mid C, h)$$

  - Generating Right Modifiers, with probability

$$\prod_{i=1,m+1} P_R(R_i(r_i) \mid H, C, h, distance_r(i-1)) \quad [R_{m+1}(r_{m+1}) \equiv \Delta(\text{stop})]$$

  - Generating Left Modifiers, with probability

$$\prod_{i=1,n+1} P_L(L_i(L_i) \mid H, C, h, distance_l(i-1)) \quad [L_{n+1}(l_{n+1}) \equiv \Delta(\text{stop})]$$

  - Distance: function of the surface string from the head word to the edge of the constituent
  - Model-II: modeling complement/adjunct distinction and sub-categorization
  - Model-III: modeling traces and Wh-Movement

Lexicalized Models for Statistical Parsing [Collins 97]

- Example: “Last week, Marks bought Brooks”
- Head Child => Left Sibling(s) => Right Sibling(s) of Head Child

S(bought) → ΔNP(week) NP(Marks) VP(bought) Δ

$$P_h(VP \mid S, \text{bought}) \times P_l(NP(\text{Marks}) \mid S, VP, \text{bought}) \times P_l(NP(\text{week}) \mid S, VP, \text{bought}) \times P_l(\Delta \mid S, VP, \text{bought}) \times P_r(\Delta \mid S, VP, \text{bought})$$

Parsers Performance Comparison [Charniak 2000]

- Penn WSJ Treebank Parsing
  - LR: Labeled Recall, LP: Labeled Precision, CB: average cross-brackets per sentence, 2CB: <= 2 CB’s, 0CB: zero CB

Parser	LR	LP	CB	0CB	2CB
<= 40 words (2245 sentences)					
Char97	87.5	87.4	1.00	62.1	86.1
Coll99	88.5	88.7	0.92	66.7	87.1
Char00	90.1	90.1	0.74	70.1	89.6
<= 100 words (2416 sentences)					
Char97	86.7	86.6	1.20	59.9	83.2
Coll99	88.1	88.3	1.06	64.0	85.1
Ratna99	86.3	87.5			
Char00	89.6	89.5	0.88	67.6	87.7

Parsing results compared with previous work

Bilingual Sentences Alignment [Kueng & Su 2002]

- Task: Aligning English and Chinese sentences
- Criteria: find the best match among *m* English sentences (ES) and *n* Chinese sentences (CS):

$$M^* = \arg\max_{M_i} P\left(M_i \left| ES_1^m, CS_1^n \right.\right)$$

$M_i = \left\{type_1^{(i)}, ..., type_{N_i}^{(i)}\right\}$: the *i*-th possible alignment-candidate, consisting of aligned passage pairs of $type_j^{(i)}$

Bilingual Sentences Alignment (Cont.)

- Reduced Model:

$$\hat{M} \equiv \arg\max_{M_i} \prod_{j=1}^{N_i} P\left([AES_j^{(i)}, ACS_j^{(i)}] \left| type_j^{(i)} \right.\right) P\left(type_j^{(i)}\right)$$

- $[AES_j^{(i)}, ACS_j^{(i)}]$: the *j*-th aligned English-Chinese Bilingual-Sentences-Groups-Pair

Robust Sentence Alignment [Kueng & Su 2002]

- Baseline Model

$$\arg\max_{M_i} \prod_{j=1}^{N_i} f\left(\delta_c, \delta_w \left| type_j^{(i)} \right.\right) \times P\left(\delta_{cognate}\right) \times P\left(type_j^{(i)}\right)$$

δ_c, δ_w: differences of lengths, in characters or in words ~ $N(0,1)$

$\delta_{cognate}$: differences in number of cognates (English strings) ~ Poission

$= \left(Num(\text{cognates in Chinese passage}) - Num(\text{cognates in English passage})\right)$

- Since almost all English cognates found in Chinese sentences can also be found in the corresponding English sentences, δ(cognate) had better be modeled as a Poisson distribution for a rare event (rather than Normal distribution as some works did).

Length-Based Sentence Alignment Models

- Length correlation: [Gale 91a]

$$\delta(l_1, l_2) = (l_2 - l_1 \cdot c) / \sqrt{l_1 \cdot s^2} \sim N(0,1)$$

$$c, s^2 : \text{mean \& variance of } (l_2 / l_1)$$

- Mean: c, the expected number of chars in L_2 per char in L_1
  - G-E =1.1 , F-E =1.06, C-E = 0.506 (characters)
- Variance: s^2, the the variance of number of chars in L_2 per char in L_1

Length-Based Sentence Alignment Models (Cont.)

- [Gale 91a] Indo-European Language-Pairs
  - 36/621 (5.8%) error rate for English-French
  - 19/695 (2.7%) error rate for English-German
  - Overall: 55/1316 (4.2%)
- [Wu 94] English-Chinese
  - 86.4% accuracy (or 95.2%, if some headers were discarded)
- [Kueng & Su 2002] English-Chinese
  - 98.2% accuracy in technical manual, 89.2 accuracy in magazine (cross style)

Robust Sentence Alignment [Kueng & Su 2002]

- Robust alignment model with lexicon matches:
  - Length distribution, alignment-type distribution (used by length-based approaches) and cognate frequency vary significantly across *document styles* and *domains*
  - Length based features are unlikely to be used by human for alignment
  - Transfer-lexicons are usually more reliable cues for human to align sentences

$$\arg\max_{M_i} \prod_{j=1}^{N_i} f_1\left(\delta_c, \delta_w \middle| type_j^{(i)}\right) \times P\left(\delta_{cognate}\right) \times f_2\left(\delta_{Transfer-Lexicons}\right) \times P\left(type_j^{(i)}\right)$$

$$\delta_{TransLexicons}: \text{ differences in number of matched lexicons} \sim N(0,1)$$
$$= \left(Num(\text{transfer-lexicons matched}) - Num(\text{transfer-lexicons unmatched})\right)$$
$$/\, Sum(\text{transfer-lexicons})$$

Robust Sentence Alignment [Kueng & Su 2002]

- Training Set:
  - 7,331 pairs (Chinese-English) (Caterpillar User Manual, machinery)
- Test Set:
  - 1,514 pairs (Caterpillar User Manual) (inside-domain testing set)
  - 274 pairs (Sinorama Magazine, general domain) (for cross-style testing)
- Sub-model features
  - CTL: use Chinese transfer lexicon plus matching type prior (P(match_type))
  - CL: use character count based length feature
  - WL: use word count based length feature
  - EC: use English cognates
- Feature combination:
  - A Sequential-Forward-Selection (SFS) procedure [Devijver, 82], based on the performance measured from the Caterpillar User Manual, is adopted to rank different feature combinations.

Robust Sentence Alignment [Kueng & Su 2002]

- Feature Selection Sequence: CTL < CL < WL < EC
  - The selection sequence verifies that the transfer-lexicon is a more reliable feature and contributes most to the aligning task.
- Robust model achieves a 60% *F*-measure error reduction (from 14.4% to 5.8%) compared with the baseline model (i.e., improving the cross-style performance from 85.6% to 94.2% in *F*-measure)

	Training Set [Caterpillar User Manual]	Testing Set I [Caterpillar User Manual]	Testing Set II [Sinorama Magazine]
Baseline Model	98.9%	98.2%	85.6%
CTL	98.3%	97.5%	82.4%
CTL+CL	99.3%	98.2%	89.6%
CTL+CL+WL	99.6%	98.8%	94.1%
CTL+CL+WL+EC (Robust Model)	99.8%	99.1%	94.2%

F-measure Performance of Baseline Model and SFS Selected Submodels