Automatic Lexicon Acquisition

(with Precision-Recall Maximization Techniques)

August 21, 2002, SWCL, Beijing

Jing-Shin Chang (shin@nlp.csie.ncnu.edu.tw)

Natural Language Processing & Machine Translation Labs
Department of Computer Science and Information Engineering
National Chi-Nan University
Puli, Nantou, Taiwan 545, ROC.

Table of Contents

- Why Automatic Lexicon Acquisition
- P-R (Precision-Recall) Maximization Problems
- Supervised Training: (for English Compound Word Extraction)
  - Learning Classifier Parameters for Precision-Recall Maximization
  - A Non-linear Adaptive Learning Algorithm for MaxPR Classification
  - Two-Stage Strategy -
    - Minimizing Classification Error: away from Naïve Bayesian
    - Non-linear Adaptive Learning for Precision- Recall Maximization
- Unsupervised Training: (for Chinese New Word Identification)
  - Unknown Word Problems
  - Word Segmentation using Viterbi Training with Augmented Dictionary
  - Iterative Integration of Word Segmentation Module && Bayesian Classifier for Precision-Recall Maximization
    - Now, well-known as Co-Training
- Knowledge Integration is the key

What is Lexicon Acquisition (English Compounds)

(Computer Manual) (“Microsoft Word User Guide”)

For information about installation, see **Microsoft Word** Getting Started. To choose a command from a menu, point to a menu name and click the **left mouse button** (滑鼠左鍵). For example, point to the File menu and click to display the File commands. If a command name is followed by an ellipsis, a **dialog box** (對話框) appears so you can set the options you want. You can also change the ***shortcut keys*** (快捷鍵) assigned to commands.

(1996/10/29 CNN)

Microsoft Corp. announced a major restructuring Tuesday that creates two worldwide **product groups** and shuffles the **top ranks** of **senior management**. Under the fourth realignment ..., the company will separate its **consumer products** from its business applications, creating a **Platforms and Applications group** and an **Interactive Media group**. ... **Nathan Myhrvold**, who also co-managed the **Applications and Content group**, was named to the newly created position of **chief technology officer**.

What is Lexicon Acquisition (Chinese New Words)

China Times 1997/7/26:

台經院指出，隨著股市活絡與景氣回溫，第一季車輛及零件營業額成長十六．八一％，顯示民間需求回升。再加上爲加入ＷＴＯ，開放進口已是時勢所趨，也將帶動消費成長。台經院預測今年民間消費全年成長率可提昇至六．七四％。

在投資方面，第一季國內投資出現回升走勢，固定資本形成實質增加六．五六％，其中民間投資實質增加八．九五％。在持續有民間大型投資計畫進行、國內房市 回溫、與政府開放投資、加速執行公共工程等多項因素下，預測今年全年民間投資將成長十一．八％。

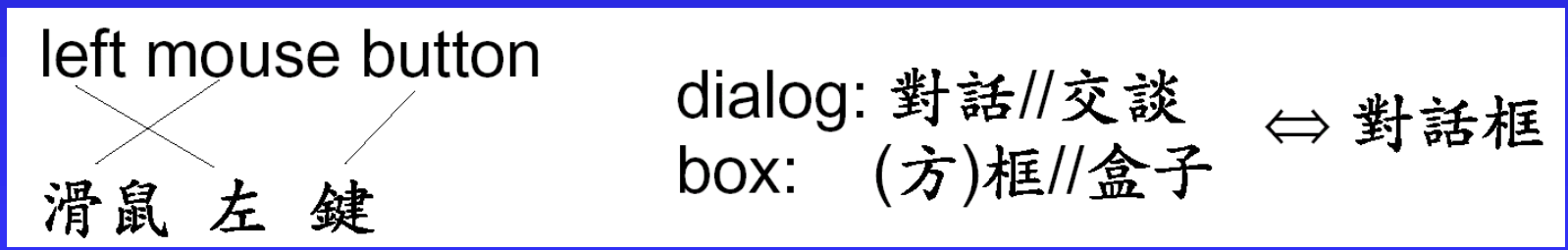
台經院表示，口蹄疫 連鎖效應在第二季顯現，使第二季出口貿易成長率比預期低，出口年增率二.一％，比去年低。而進口年增率爲七．三八％，因此第二季貿易出超僅十七．一四億美元，比去年第二季減少四十三．六五％。不過，由於第三、四季爲出口旺季，加上國際組織均預測今年世界貿易量擴大，台經院認爲我國商品出口應可轉趨順暢。

Why Automatic Lexicon Acquisition

1. A large-scale electronic dictionary is important to many NLP applications
   - machine translation, spoken language processing, spelling check, Chinese associated input methods
2. New (unknown) words && compound words increase rapidly (e.g., “網咖”、“網吧”、“凍蒜”、“奈米科技”、“奈米管”)
   - increase with time (every day)
   - vary with domain (every domain)
3. NLP systems prefer to lexicalize compound words for easier: analysis (disambiguation), generation (composition)

   e.g., **book** (*n*, v_i, v_t) + **store** (*n*, v_t) ⇔ **book store** (*n*)

   e.g., **green house** =\= **‘green’** + **‘house’**

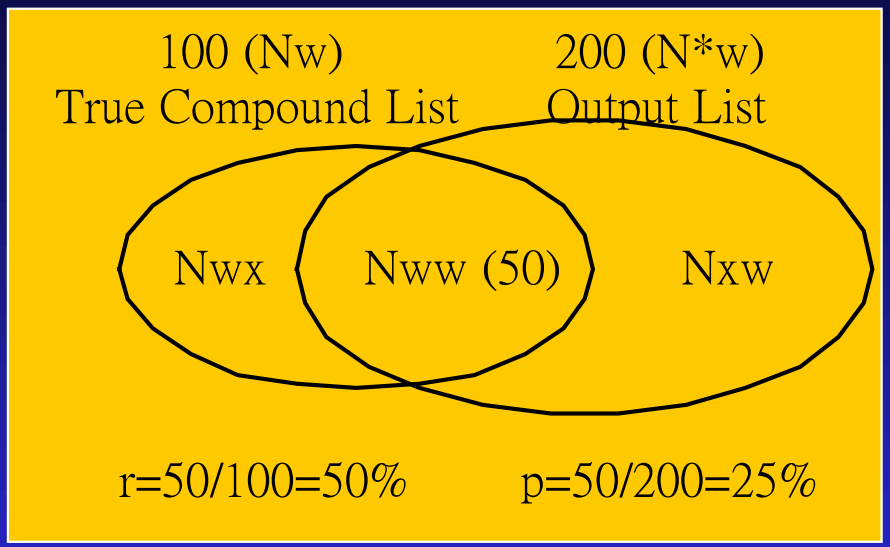


Why Automatic Lexicon Acquisition

4. Full human construction is costly, time consuming and inconsistent
5. Electronic text is becoming widely available

*. Examples of acquisition: Compound Words, Unknown Words

Precision-Recall Optimization Criteria



$$p = \frac{N_{ww}}{N_{ww} + N_{xw}} = \frac{1}{1 + N_{xw} / N_{ww}}$$

$$r = \frac{N_{ww}}{N_{ww} + N_{wx}} = \frac{1}{1 + N_{wx} / N_{ww}}$$

- Basic Criteria: Precision & Recall
  - $p = N_{ww}/(N_{ww}+N_{xw})$ = #correct_identification / #output_words
  - $r = N_{ww}/(N_{ww}+N_{wx})$ = #correct_identification / #all_words
  - (N_{ij}: # of class-i n-grams which are classified as class-j)
  - (i, j= w - word//compound ; x - non-word//non-compound)
- Most filtering approaches are unable to improve both *simultaneously*

Precision-Recall Optimization Criteria

- Typical Joint Criteria for Precision (p) and Recall (r) Maximization: WPR & F-measure
- WPR: W_p*p+W_r*r (weighted Precision/Recall)
  [W_p, W_r: weights (W_p+W_r=1)]
  - A weighting sum of precision and recall.
- F-metric (F-measure):
  - Definition:

$$F(\beta) = \frac{(\beta^2+1)pr}{\beta^2 p + r} = \frac{pr}{p\beta^2/(\beta^2+1) + r/(\beta^2+1)}$$

$$F(1) = \frac{2pr}{p+r}$$

  - A metric that appreciates a balance between precision and recall.
    [Maximal at p=r if β=1 and p+r is a constant.]
    (Prefer maximal product of p and r for a given weighted P/R)

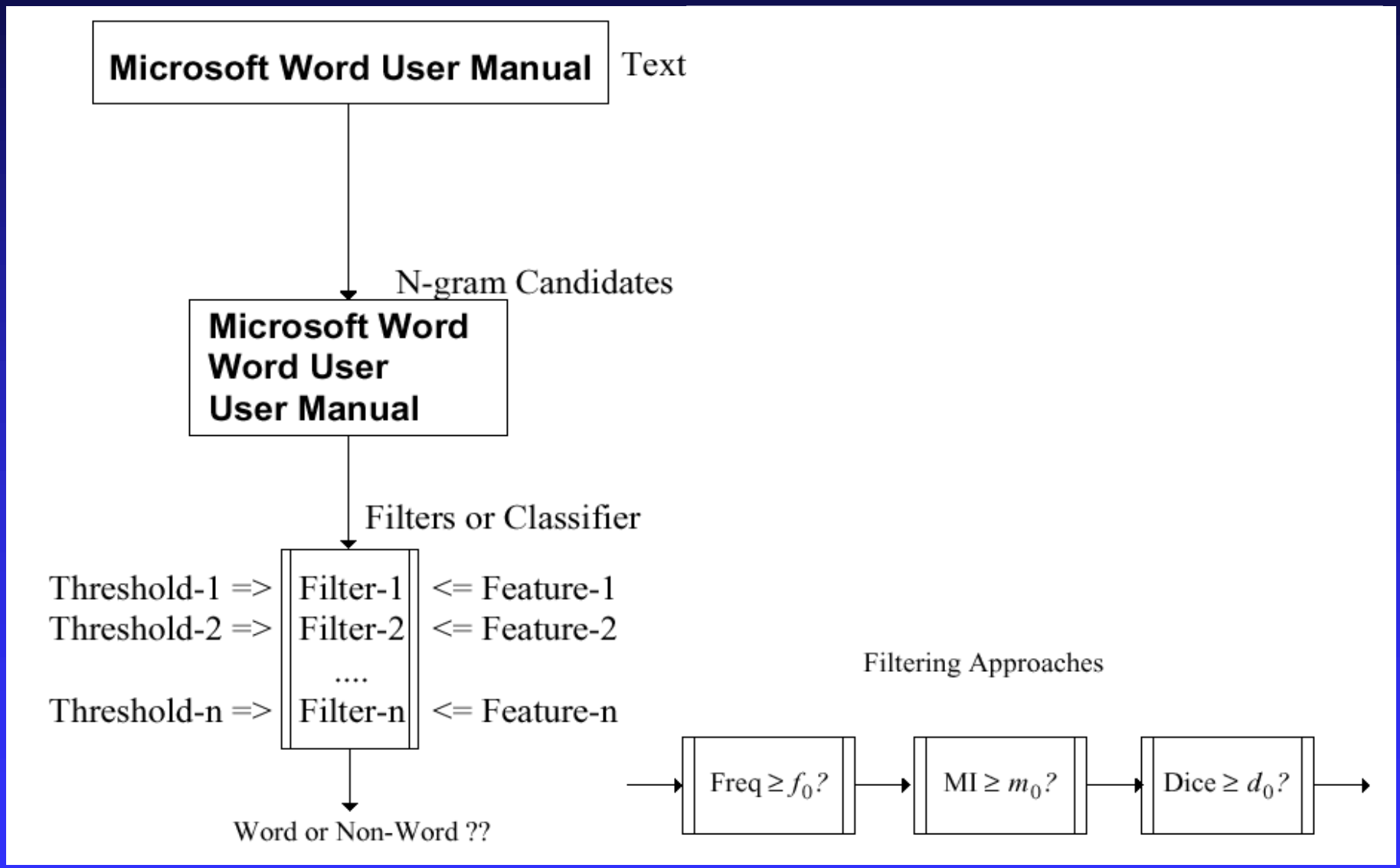
Precision-Recall Maximization Problems in Different Modes of Lexicon Acquisition

- Supervised:
  - Language parameters can be well estimated with labeled data
  - Q: Can we find a set of parameters that maximizes a user-defined function of precision and recall?
  - Example Task:
    - English Compound Word Extraction
- Unsupervised:
  - Language parameters are not well known
  - Q: How to improve language parameters toward joint precision-recall maximization with the help of other knowledge sources
  - Example Task:
    - Chinese Unknown Word Extraction

English Compound Word Extraction with a Non-Linear Learning Method for Precision-Recall Maximization

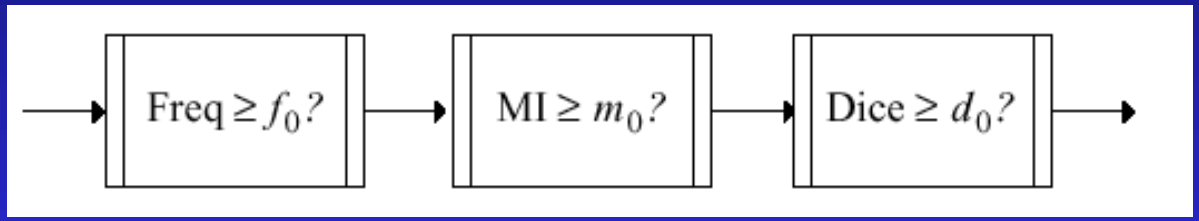
(Supervised Mode of Acquisition)

Traditional Filtering Scheme in English Compound Extraction

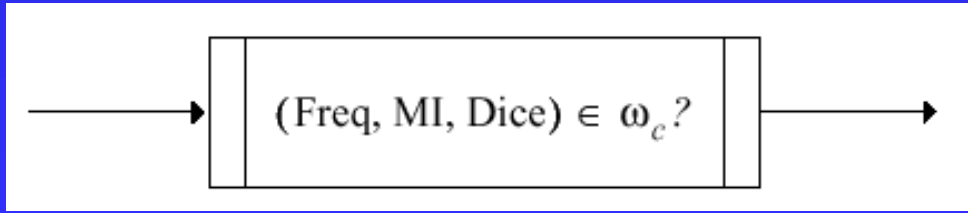


Problems with Traditional Scheme in Lexicon Acquisition

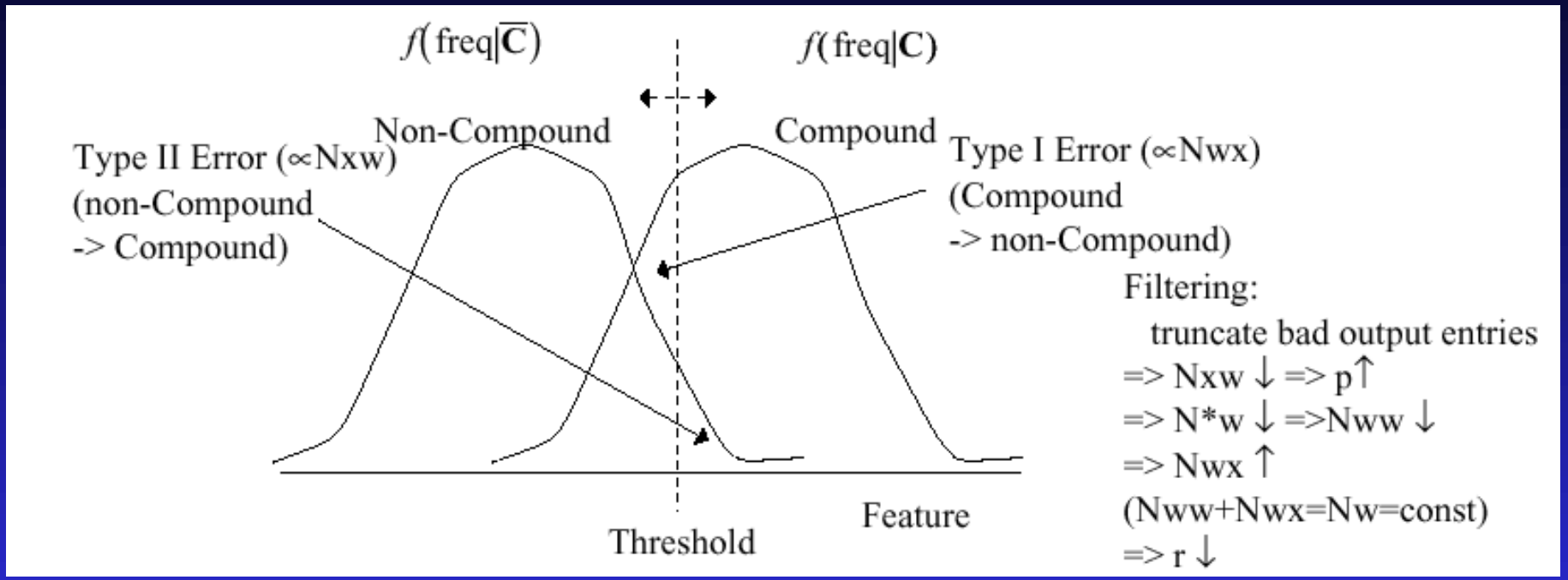
- Use simple *filtering* approaches and heuristic thresholds in extracting lexicon entries
  - Mostly based on step-by-step filtering approaches which filter out inappropriate candidates with one feature per step



  - Thresholds are determined by trial-and-error
  - No unified method for integrating known features
    - features are used *independent* of one another (e.g., cascaded)
    - no automatic method for identifying the best feature
  - A better approach: using a unified model to integrate all features



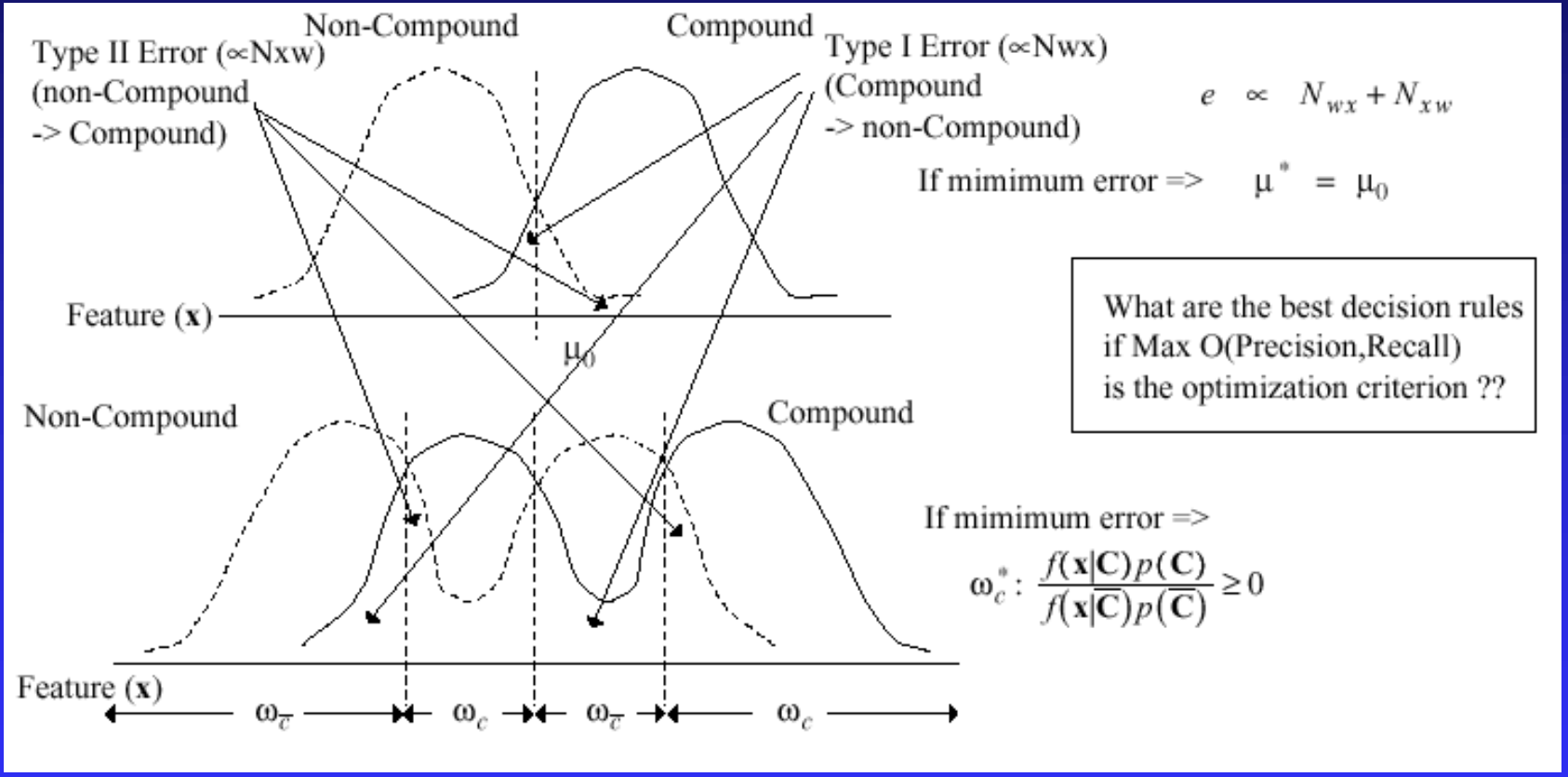
Precision-Recall Maximization Problems



- Precision and Recall cannot be tuned in an appropriate manner
  - precision and recall are nonlinear functions of error counts
  - adaptation to maximize different joint P/R preferences (such as F-metric) in different tasks had not been addressed
  - precision and recall cannot be improved at the same time
  - important thresholds for features are determined arbitrarily

Two-Stage P-R Maximization

- Designing MaxPR Classifier (??)
  - Minimum error classifier: is known to be Bayesian.
  - BUT what is a MaxPR classifier?? Does it exist??



Two-Stage P-R Maximization (cont.)

- Problem?
  - No simple analytical decision rules that are capable of achieving any user-specified joint criterion function of precision and recall
    - precision and recall are nonlinear functions of error counts
- Two-stage Strategy
  - 1st stage: Error Rate Minimization: Bayesian
  - 2nd stage: Precision-Recall Maximization: adapting parameters toward maximum precison-recall

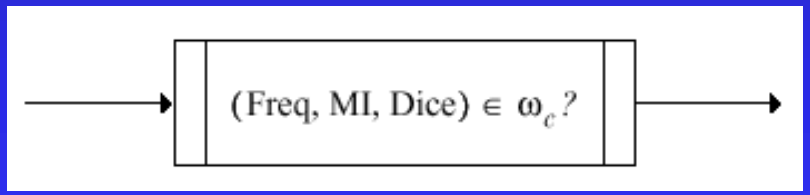
Two-Stage P-R Maximization (cont.)

- Which two stages?
  - Minimize classification error:

$$p = \left(1 + n_{xw}/n_{ww}\right)^{-1};\ r = \left(1 + n_{wx}/n_{ww}\right)^{-1}$$

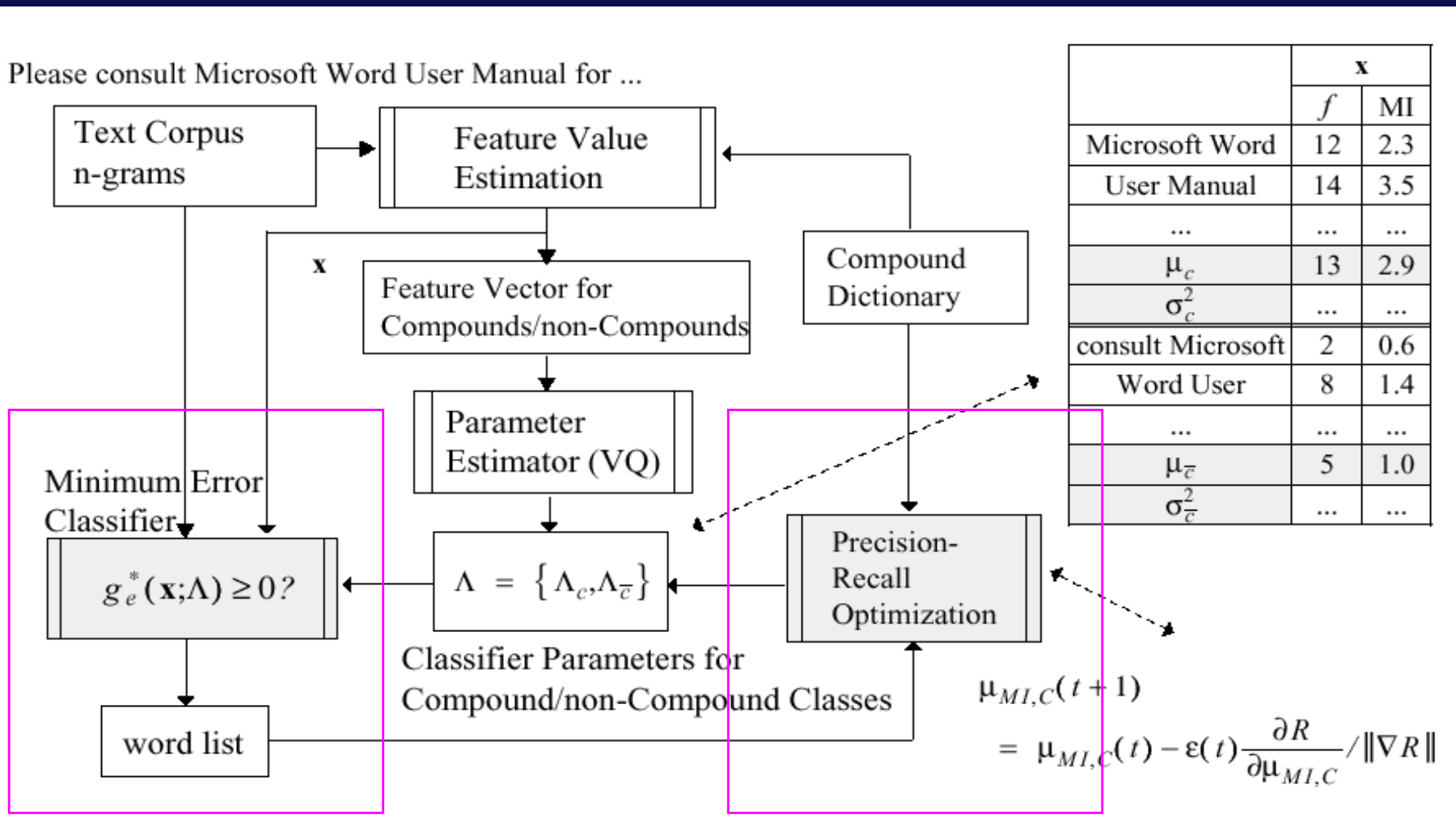
    reduce error rate (N_{wx}+N_{xw}) generally also improve *P*, *R* and other joint functions (Note: Maximize FM == Minimize (n_{wx}+n_{xw})/ n_{xw})
  - Maximize precision-recall:

    Min error classification ≠ MaxPR classification
- How to?
  - Minimum error classification: Bayesian classifier with better features, better models for jointly combining all features, better estimation



  - Maximize precision-recall: by parameter learning (nonlinear!!)

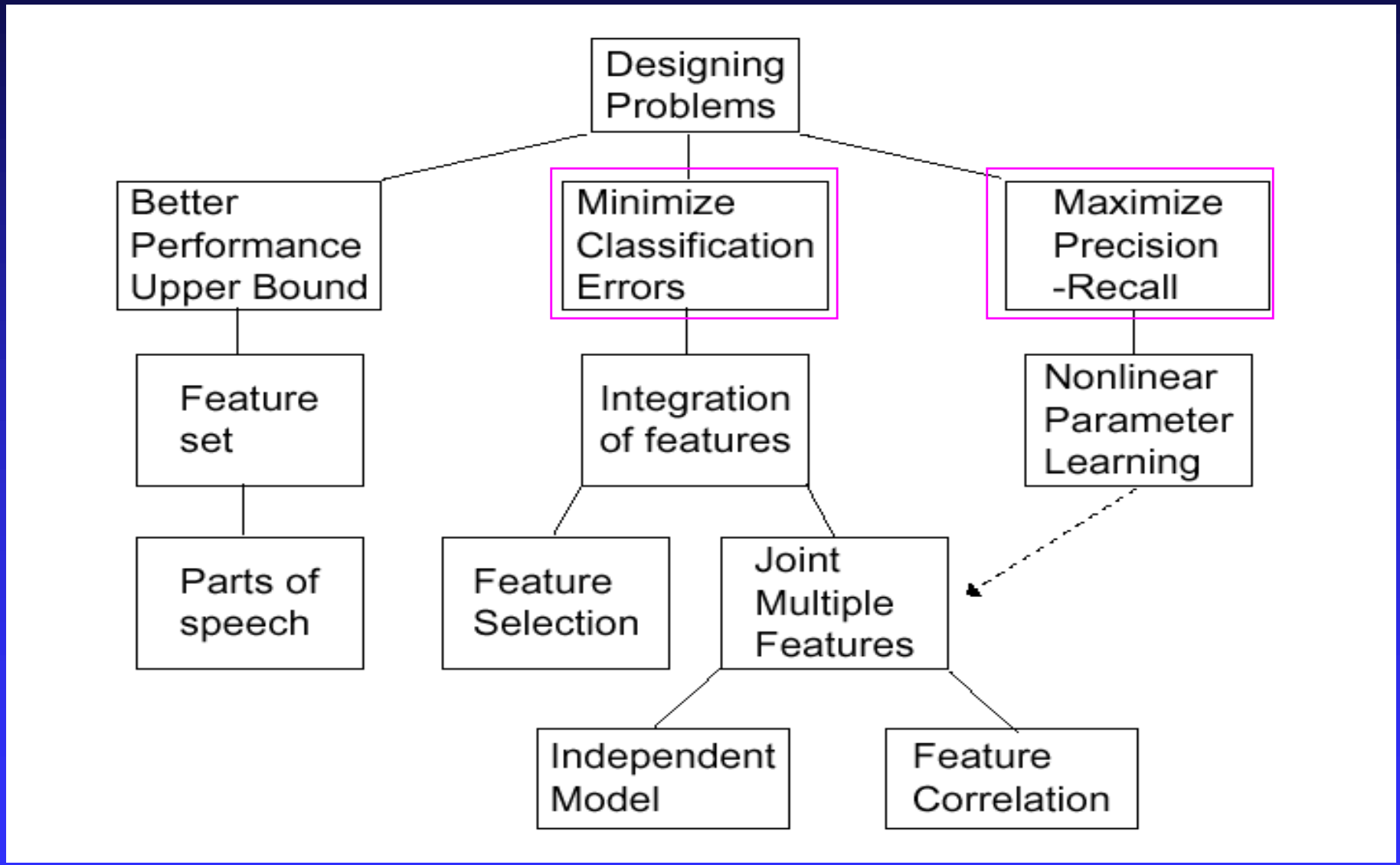
MinErr Classifier+MaxPR Learning Approach



	x	
	f	MI
Microsoft Word	12	2.3
User Manual	14	3.5
...	...	...
μ_c	13	2.9
σ_c^2	...	...
consult Microsoft	2	0.6
Word User	8	1.4
...	...	...
$\mu_{\bar{c}}$	5	1.0
$\sigma_{\bar{c}}^2$	...	...

Figure 1 Supervised Training of Classifier Parameters for English Compound Extraction

General Problems in Classifier Design



Stage-I: MinErr Classifier: Two-Class Classifier for Identifying New Words or Compound Words

Input: n-grams (n-word compounds, n-character words) in the text corpus

Output: assign a class label ("word" or "non-word") to each n-gram

Classifier: a log-likelihood ratio (LLR) tester (minimum error classifier)

$$g(\mathbf{x}) = LLR(\mathbf{x}) = \log \frac{f(\mathbf{x}|\mathbf{W})P(\mathbf{W})}{f(\mathbf{x}|\overline{\mathbf{W}})P(\overline{\mathbf{W}})}$$

Decision Rules:

$$class(w) = \begin{cases} +w \quad (word) & if \quad LLR(\bullet) \geq \lambda_0 \\ -w \quad (non-word) & if \quad LLR(\bullet) < \lambda_0 \end{cases}$$

Advantage: ensure minimum classification error (with λ_0 =0) if the distributions are known.

Features for the Classifier

- Normalized Frequency : a character n-gram, x, is likely to be a word if it appears more frequently than the average.
- Mutual Information: characters x and y with high mutual information tend to have high association [Church 90]

$$I(x,y) = log\frac{P(x,y)}{P(x)\times P(y)}$$

- Entropy: random distribution of the left/right neighbors (Ci) of an n-gram x implies a natural break at the n-gram boundary [Tung 94]:

$$H(x) = -\sum_{c_i} P(c_i;x) log P(c_i;x)$$

- Dice: similar to mutual information with non-occurring events (x=0,y=0) ignored [Smadja 96]:

$$D(x,y) = \frac{P(x=1,y=1)}{\frac{1}{2}[P(x=1)+P(y=1)]}$$

Features for the Classifier (cont.)

Part-of Speech Discrimination:

$$D_{pos}(x_i;\{P_{ij}\},\{P_j\}) = \sum_j P_{ij} log \frac{P_{ij}}{P_j}$$
$$P_{ij} \equiv P(j|w_i), \quad P_j \equiv P(j)$$

An n-gram, X_i, is likely to be a word if its parts-of-speech (詞類) distribution is "close to" the parts-of-speech distribution of the n-grams in the word-class, where closeness is measured in terms of the discrimination between two probability distributions.

P_{ij}: probability for X_i to be tagged with part-of-speech pattern j (e.g., j = [n n] for a noun-noun compound word).

P_j: probability for any n-grams to be tagged with part-of-speech pattern j.

Baseline: Error Rate by Using One Feature

Baseline: Error Rate by Using One Feature											
		Training Set					Testing Set				
Feature		Dpos	MI	H	NF	D	Dpos	MI	H	NF	D
2-gram Baseline	Recall	11.09	0	4.87	6.01	12.33	8.07	0	1.35	2.69	36.77
	Precision	100	*	30.92	30.69	37.07	100	*	23.08	33.33	57.75
	Error Rate	11.03	12.41	13.15	13.34	13.47	21.2	23.06	23.78	23.68	20.79
	WPR(1:1)	55.54	*	17.9	18.35	24.7	54.03	*	12.22	18.01	47.26
	F-measure	19.97	*	8.41	10.05	18.5	14.93	*	2.55	4.98	44.93
Feature		Dpos	MI	H	NF	D	Dpos	MI	H	NF	D
3-gram Baseline	Recall	0	0	13.99	10.2	7.58	0	0	12.07	3.45	39.66
	Precision	*	*	42.11	22.58	25.49	*	*	58.33	66.67	41.07
	Error Rate	4.95	4.95	5.21	6.18	5.67	11.51	11.51	11.11	11.31	13.49
	WPR(1:1)	*	*	28.05	16.39	16.54	*	*	35.2	35.06	40.37
	F-measure	*	*	21	14.05	11.69	*	*	20	6.56	40.35

Table 1. Error Rate Performance Using only One Feature
(*:undefined, i.e.,all candidates are classified as non-compound.).

Use Features Jointly and Select Discriminative Features Automatically for the Classifier

0. Initialize current feature set as empty.

1. Classify training data by jointly (*) using current feature set and one of the remaining features not in the current feature set. Try all the remaining features one-by-one, and include the feature that performs best to the current feature set.

2. Stop including new features whenever the performance of the classifier begins to flatten or degrade due to the inclusion of redundant or contradictory features.

3. Use the selected features for lexicon acquisition.

(*) - Models for Jointly Integrating Features:
IN: Independent Normal Model (Naïve Bayesian)
Mx: Mixtures of Gaussian Density Functions

Error Rate by Using Independent Normal Model with Feature Selection for Joint Consideration

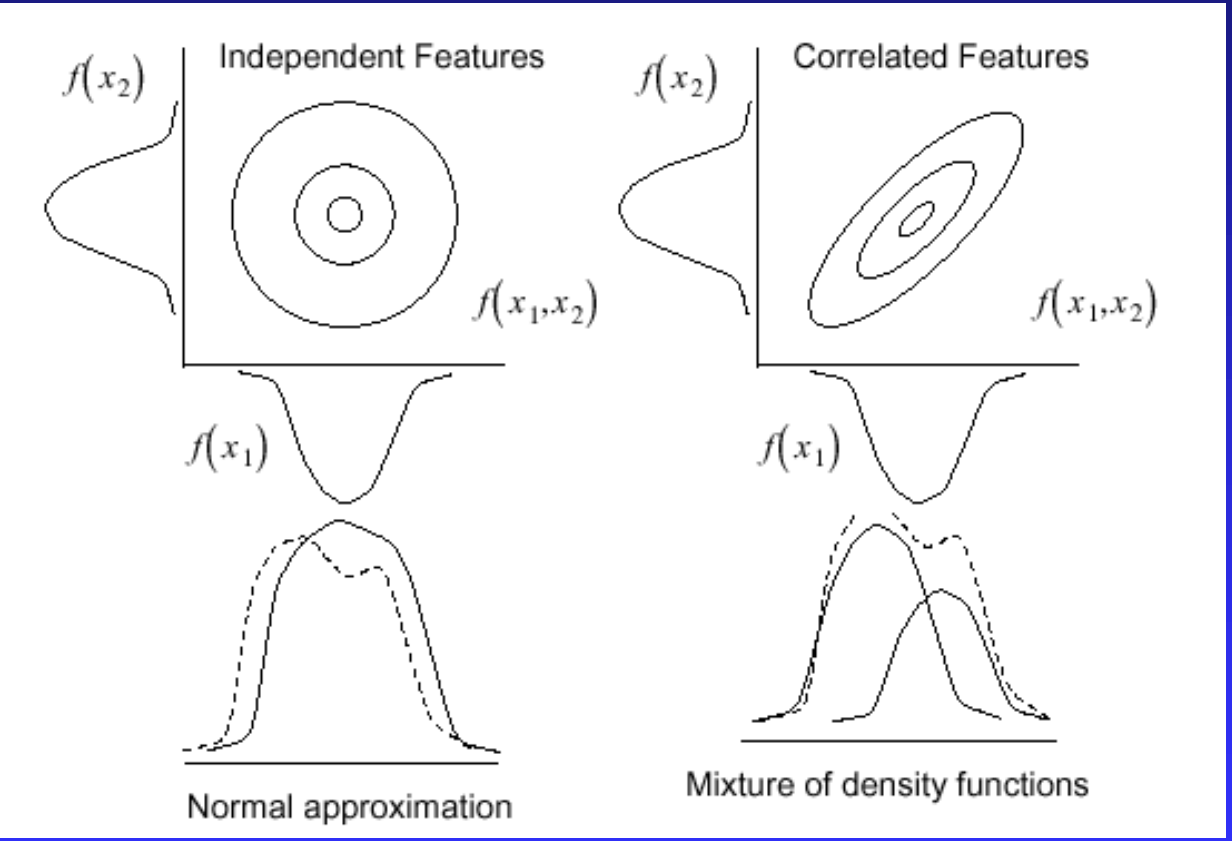
Error Rate by Using Independent Normal Model with Feature Selection for Joint Consideration											
		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	11.09	40.41	54.61	35.34	31.3	8.07	35.43	60.54	33.63	50.67
	Precision	100	88.04	77.39	71.04	49.67	100	89.77	92.47	82.42	66.47
	Error Rate	11.03	8.07	7.61	9.81	12.46	21.2	15.82	10.24	16.96	17.27
	WPR(1:1)	55.54	64.23	66	53.19	40.49	54.04	62.6	76.51	58.03	58.57
	F-measure	19.97	55.39	64.03	47.2	38.4	14.93	50.81	73.17	47.77	57.5
Feature Sequence		Dpos	MI	H	D	NF	Dpos	MI	H	D	NF
3-gram	Recall	0	14.29	33.53	29.45	26.24	0	17.24	44.83	56.90	48.28
	Precision	*	100	70.99	46.98	33.83	*	100	86.67	49.25	47.46
	Error Rate	4.95	4.24	3.97	5.14	6.19	11.51	9.52	7.14	11.71	12.1
	WPR(1:1)	*	57.15	52.26	38.22	30.04	*	58.62	65.75	53.08	47.87
	F-measure	*	25.01	45.55	36.2	29.56	*	29.41	59.09	52.8	47.86

Table 2. Error rate performance of the independent normal model.

Joint Consideration of the Features by Considering Feature Correlation

0. Why ?

- Features are not really independent (have correlation)
- Features are not really normally distributed (use mixtures)



Joint Consideration of the Features by Considering Feature Correlation

- 1. Model the distributions of the features with a k-mixture Gaussian Density Function to take correlations among features into consideration.
  - k is to be determined automatically by the feature selection mechanism.

$$f(x|\Lambda) \equiv \sum_{i=1}^{K} r_i \cdot N(x;\mu_i,\Sigma_i), \quad \sum_{i=1}^{K} r_i = 1$$

$$N(x;\mu,\Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

- 2. Estimate the parameters of the feature distributions using a clustering algorithm to maximize the likelihood of the input feature vectors.

Fixing K throughout Feature Selection Process

Fixing K throughout Feature Selection Process											
		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.5	71.61	50.67	51.71	69.06	71.3	69.96	67.26	47.09
	Precision	100	97.87	88.93	62.93	45.53	100	95.78	93.41	80.65	52.24
	Error Rate	3.74	3.73	4.63	9.82	13.67	7.14	7.34	8.07	11.27	22.13
	WPR(1:1)	84.92	84.69	80.27	56.8	48.62	84.53	83.54	81.68	73.95	49.66
	F-measure	82.24	82.63	79.34	56.14	48.42	81.7	81.75	80	73.34	49.53

Table 3. The Best Bigram Performance of the Minimum Error Rate Classifier Using a 2-Mixture Multivariate Normal Density Function (K=2).

Feature Sequence		Dpos	H	MI	D	NF	Dpos	H	MI	D	NF
3-gram	Recall	63.27	68.22	67.06	51.9	54.23	75.86	74.14	74.14	36.21	37.93
	Precision	100	95.12	90.91	80.91	39.08	100	97.73	95.56	95.45	41.51
	Error Rate	1.82	1.75	1.96	2.99	6.45	2.78	3.17	3.37	7.54	13.29
	WPR(1:1)	81.63	81.67	78.98	66.4	46.65	87.93	85.93	84.85	65.83	39.72
	F-measure	77.5	79.45	77.18	63.24	45.43	86.27	84.32	83.5	52.5	39.64

Table 4. The Best Trigram Performance of the Minimum Error Rate Classifier Using a 3-Mixture Multivariate Normal Density Function (K=3).

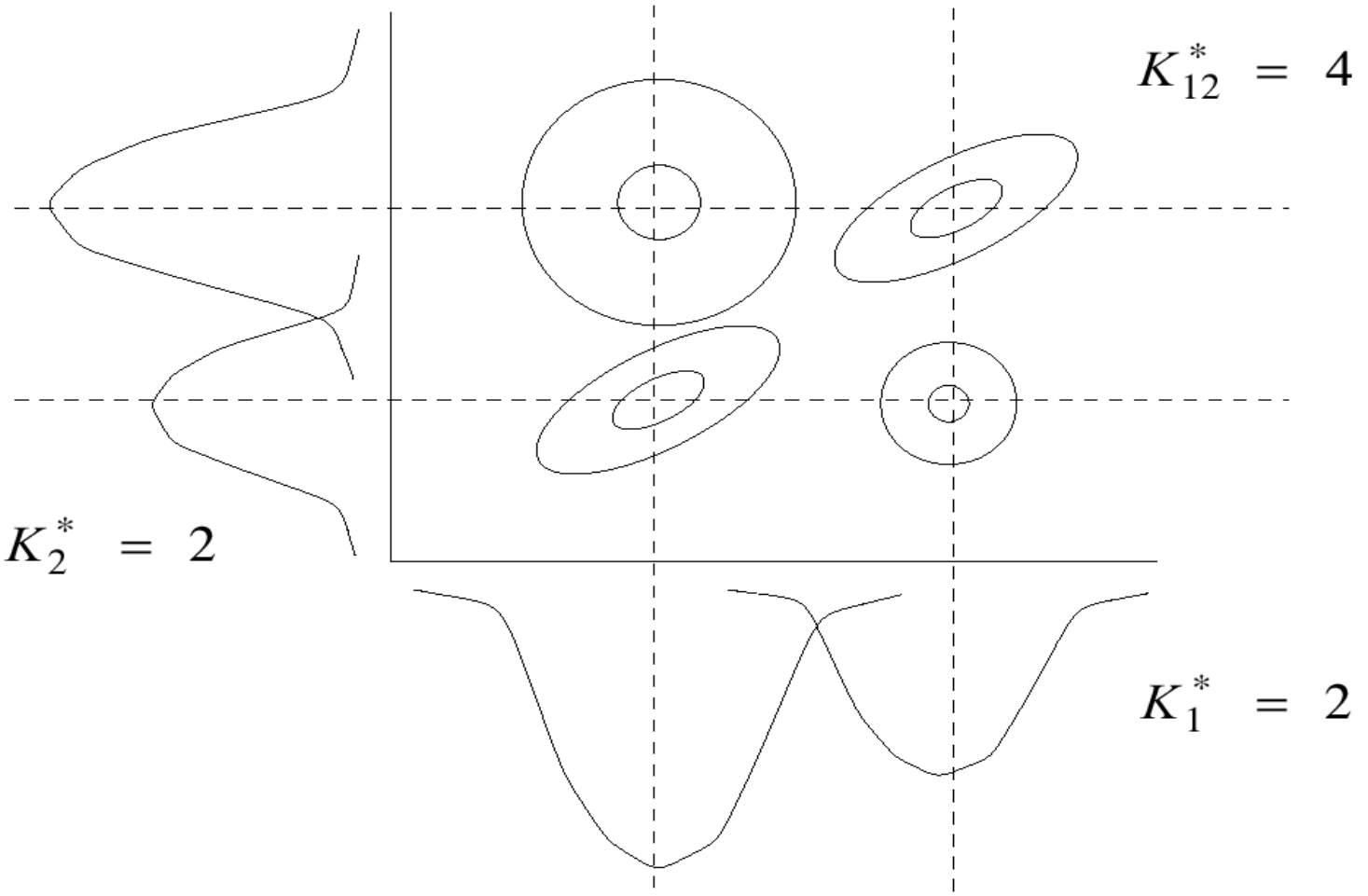
Comparison: Joint Consideration of the Features

Comparison: Joint Consideration of the Features

			Training Set					Testing Set				
N	Model & Features		P	R	E	WPR	FM	P	R	E	WPR	FM
2	IN:	Dpos+H	88.04	40.41	8.07	64.23	55.39	89.77	35.43	15.82	62.6	50.81
	IN:	Dpos+H+MI	77.39	54.61	7.61	66	64.03	92.47	60.54	10.24	76.51	73.17
	Mx:	Dpos+H(K=2)	97.87	71.5	3.73	84.69	82.63	95.78	71.3	7.34	83.54	81.75
3	IN:	Dpos+MI	100	14.29	4.24	57.15	25.01	100	17.24	9.52	58.62	29.41
	IN:	Dpos+MI+H	70.99	33.53	3.97	52.26	45.55	86.67	44.83	7.14	65.75	59.09
	Mx:	Dpos+H(K=3)	95.12	68.22	1.75	81.67	79.45	97.73	74.14	3.17	85.93	84.32

Table 5. Comparison between Independent Normal (IN) Model and K-mixture Multivariate Normal (Mx) Model.
(2: 2-gram, 3: 3-gram, P: Precision, R: Recall, E: Error Rate, WPR: Weighted Precision/Recall with equal weights, FM: F-measure.)

Searching for the Best Number of Mixtures (K*)



Number of Mixtures increases rapidly with feature dimension

Searching for the Best Number of Mixtures (K*)

- Why ?

  (1) As the number of features increases, K*, in general, will increase rapidly

  (2) The estimation algorithm can only achieve local maximum for likelihood value

  - using a larger K does not guarantee to reach better local maximum likelihood estimate than using a smaller K

  & even (global) maximum likelihood $\neq>$ minimum error rate

  ❑ => using larger K $\neq>$ smaller error rate

Searching for the Best Number of Mixtures (K*)

Searching for Best Number of Mixtures (K*)											
		Training Set					Testing Set				
Feature Sequence		Dpos(2)	H(2)	MI(3)	NF(3)	D(1)	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.5	72.12	67.05	32.12	69.06	71.3	70.4	65.92	44.39
	Precision	100	97.87	90.74	83.7	56.78	100	95.78	94.01	93.63	68.28
	Error Rate	3.74	3.73	4.37	5.71	11.45	7.14	7.34	7.86	8.89	17.58
	WPR(1:1)	84.92	84.69	81.43	75.37	44.45	84.53	83.54	82.21	79.77	56.34
	F-measure	82.24	82.63	80.37	74.46	41.03	81.7	81.75	80.51	77.37	53.8
Feature Sequence		Dpos(3)	H(3)	MI(3)	D(3)	NF(1)	Dpos	H	MI	D	NF
3-gram	Recall	63.27	68.22	67.06	51.9	24.49	75.86	74.14	74.14	36.21	44.83
	Precision	100	95.12	90.91	80.91	33.6	100	97.73	95.56	95.45	48.15
	Error Rate	1.82	1.75	1.96	2.99	6.13	2.78	3.17	3.37	7.54	11.9
	WPR(1:1)	81.63	81.67	78.98	66.4	29.04	87.93	85.93	84.85	65.83	46.49
	F-measure	77.51	79.45	77.19	63.24	28.34	86.27	84.32	83.5	52.5	46.43

Table 6. The Performance of the Minimum Error Rate Classifier Using Multivariate Normal Density Function up to 3 Mixtures (Kmax=3).

Stage-II: Precision and Recall Maximization

- Why: The *minimum error* classifier does not necessarily achieve *maximal O(precision, recall)*
  [O(.): a joint optimization function of precision and recall which reflects user preference]

- Precision (*p*) and Recall (*r*) (instead of error rate), however, are the major performance indices to maximize in text extraction or information retrieval tasks.

- Capable of maximizing any preference function of precision and recall is therefore an important issues, which had not been formally addressed in the literature.

Non-Linear Adaptive Learning for P-R Maximization

- A probabilistic descent method to maximize f(precision, recall).
- Define Risk for WPR: R = W_p*(1-p)+W_r*(1-r).
  (or risk for FM, etc.)
- Express the risk as a function of the parameters of the classifier.
- Adjust the classifier parameter vector in the -grad R direction when n-grams in the corpus are misclassified

(∇: gradient w.r.t. the classifier parameters).

$$\vec{\delta}_\Lambda(t) = -\varepsilon(t)\nabla R/\|\nabla R\|$$
$$\Lambda(t+1) = \Lambda(t)+\vec{\delta}_\Lambda(t)$$

  - The risk will be non-increasing on average. $\left(\delta\overline{R}\le 0\right)$
  - The same learning algorithm can be applied to other functions of precision/recall, such as F-metric, to improve the extraction tasks.
- It is non-linear since the parameters are updated in batch, not by sample, unlike most learning algorithms for minimizing error rate.

Learning Parameters for Maximal Precision-Recall (cont.)

- Gradient of risk can be expressed as a function of the numbers of classification errors, N12 and N21, and any differentiable approximation to N12 & N21 (f12, f21)

$$\begin{aligned}
\nabla R &= w_p \nabla \frac{n_{21}}{n_1 - n_{12} + n_{21}} + w_r \nabla \frac{n_{12}}{n_1} \\
&\approx w_p \frac{(n_1 - n_{12} + n_{21}) \nabla f_{21} - n_{21} \nabla (n_1 - f_{12} + f_{21})}{(n_1 - n_{12} + n_{21})^2} + w_r \nabla \frac{f_{12}}{n_1} \\
&= \frac{w_p n_{11}}{(n_{11} + n_{21})^2} \nabla f_{21} + \left[\frac{w_p n_{21}}{(n_{11} + n_{21})^2} + \frac{w_r}{n_1} \right] \nabla f_{12} \\
&\equiv k_{21} \nabla f_{21} + k_{12} \nabla f_{12}
\end{aligned}$$

- where the approximated error counts (f12, f21) are expressed as the sum of a zero-one loss function, *l(.)*, over each error, with

$$l(d_{\vec{x}}) = \frac{1}{\pi} \tan^{-1}\left(\frac{d_{\vec{x}}}{d_0}\right) + \frac{1}{2} \qquad d_{\vec{x}} \equiv g(\vec{x})$$

Learning Parameters for Maximal Precision-Recall (cont.)

- and result in

$$\nabla f_{12} = \sum_{\vec{x}:c(\vec{x})=1, g(\vec{x})<0} \nabla l(-d_{\vec{x}}) = -\sum l'(-d_{\vec{x}}) \nabla d_{\vec{x}}$$
$$\nabla f_{21} = \sum_{\vec{x}:c(\vec{x})=2, g(\vec{x})\geq 0} \nabla l(+d_{\vec{x}}) = +\sum l'(+d_{\vec{x}}) \nabla d_{\vec{x}}$$

- which depend on the decision of the classifier, i.e., depend on $g(\vec{x})$, and thus are functions of the parameters of the classifier.
  - The summation operator suggests that it is a non-linear learning algorithm which updates the parameters in batch, not by sample.
- Learning Constants for WPR maximization:

$$k_{21} = \frac{w_p n_{11}}{(n_{11}+n_{21})^2}$$
$$k_{12} = \frac{w_p n_{21}}{(n_{11}+n_{21})^2} + \frac{w_r}{n_1}$$

- Learning Constants for F-metric maximization:

$$k_{21} \equiv \frac{\alpha_{21}}{(n_1 - n_{12})} = \frac{1}{\beta^2+1}\frac{1}{(n_1 - n_{12})}$$
$$k_{12} \equiv \frac{(\alpha_{12} n_1 + \alpha_{21} n_{21})}{(n_1 - n_{12})^2} = \frac{1}{\beta^2+1}\frac{(\beta^2 n_1 + n_{21})}{(n_1 - n_{12})^2}$$

Learning Parameters for Maximal Precision-Recall - Bigram Example

Learning Parameters for Maximal Precision-Recall - Bigram Example											
		Training Set					Testing Set				
N	Model	P	R	E	WPR	FM	P	R	E	WPR	FM
2	IN: Dpos+H	88.04	40.41	8.07	64.23	55.39	89.77	35.43	15.82	62.6	50.81
	IN+LRN: WPR(1:1)	97.35	72.44	3.66	84.89	83.07	97.56	71.75	6.93	84.65	82.69
	Mx:Dpos+H(Kmax=3)	97.87	71.5	3.73	84.69	82.63	95.78	71.3	7.34	83.54	81.75
	Mx+LRN:WPR(1:1)	99.57	72.75	3.42	86.16	84.07	100	71.75	6.52	85.87	83.55
	Mx+LRN:FM	99.43	72.85	3.42	86.14	84.09	100	71.75	6.52	85.87	83.55

Table 7. Learning Results on Mixture of Multivariate Normal Model
(IN: Independent Normal Model, Mx: Mixture of Multivariate Normal Model, IN+LRN: Adaptive Learning on Independent Normal Model. Mx+LRN: Adaptive Learning on Multivariate Normal Mixtures)

Learning to Meet User Spec on O(p,r)

Learning to Meet User Spec on O(p,r)											
		Training Set					Testing Set				
N	Model	P	R	E	WPR	FM(beta)	P	R	E	WPR	FM
2	Mx:Dpos+H (Kmax=3) before learning	97.87	71.5	3.73	78.1(1:3) 84.69(1:1) 91.28(3:1)	91.15(0.5) 82.63(1.0) 75.58(2.0)	95.78	71.3	7.34	77.42 83.54 89.66	89.63 81.75 75.14
	Mx+LRN: WPR(1:3)	94.08	74.09	3.79	79.09	82.9	97.58	72.2	6.83	78.54	82.99
	Mx+LRN: WPR(1:1)	99.57	72.75	3.42	86.16	84.07	100	71.75	6.52	85.87	83.55
	Mx+LRN: WPR(3:1)	99.71	72.02	3.5	92.79	83.63	100	71.3	6.62	92.83	83.25
	Mx+LRN: FM(0.5)	99.57	72.75	3.42	86.16	92.73	100	71.75	6.52	85.87	92.7
	Mx+LRN: FM(1.0)	99.43	72.85	3.42	86.14	84.09	100	71.75	6.52	85.87	83.55
	Mx+LRN: FM(2.0)	89.51	75.13	4.18	82.32	77.62	97.02	73.09	6.72	85.06	76.89

Table 8. Learning Results for Different User Preferences over Precision and Recall

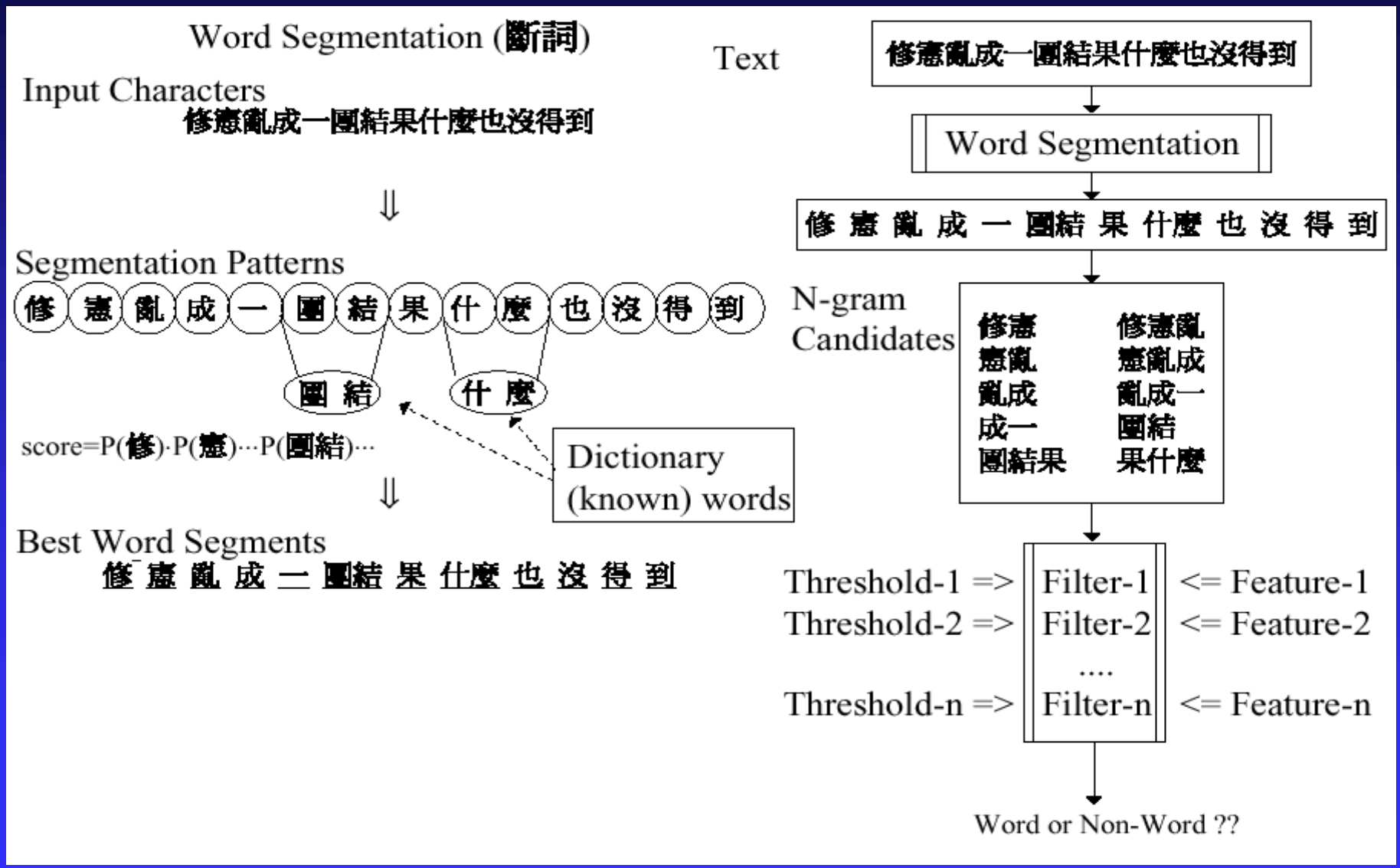
An Iterative Precision-Recall Maximization Method for Chinese New Word Identification

(Unsupervised Mode of Acquisition)

Chinese-Specific Problems

- More difficult than English in identifying lexical units
  - No natural delimiters (like spaces) between lexical entries
  - Need word segmentation (斷詞) for identifying new words
- Unknown Word Problems during Word Segmentation (WS)
  - Most word segmentation algorithms produce over-segmented single character regions when there are unknown (new) words
  - Some tokens are mis-merged during segmentation
- Need extra information for word segmentation: WS+filter

General Scheme in Chinese Lexicon Extraction



General Scheme for Chinese New Word Identification

- Segmentation-Merging-Filtering-Disambiguation Scheme [Tung 94, Wang 95]:
  1. Segmentation with (known words in) system dictionary
  2. Merge adjacent n-grams to form unknown word candidates
  3. Filter out inappropriate candidates with character association metrics
  4. Disambiguation on overlapped candidates
     (e.g., `漁業 區 附近')

- Method of Knowledge Source Integration:
  - Combine information sources by cascading the above modules using one-pass, non-iterative cascaded scheme

Integration of Knowledge Sources

- Conventional System Schemes:
  - Segmentation (with known words) + Merge adjacent characters + Qualification with a filter
- Characteristics:
  - Independent knowledge sources, one-pass, non-iterative
    - Word Segmentation: Use contextual constraints (or contextual probabilities) to find the best segmentation
    - Filter: Use word association features (e.g., mutual information, dice) to filter out unlikely compound words
      - many filtering approaches filter out unlikely candidates in a feature-by-feature filtering manner, one feature one filtering step
  - No information sharing between the two modules

Problems with Segment-Merge-Filtering Schemes

- Merge-type errors cannot be recovered:
  - Types of errors: over-segmentation, under-segmentation (mis-merging)
  - **New words** may be merged with neighbors into **known words** in a system dictionary, and thus will not be extracted
    - Example: known word:土地公 & new word:公有
    - [土地公有政策] => [土地公][有][政策]
- **Simple filtering will *never* improve recall**
  - Successful filtering ⇒ precision improved, recall **unchanged**
  - Unsuccessful filtering ⇒ both precision and recall **degraded**

Problems with Segment-Merge-Filtering Schemes

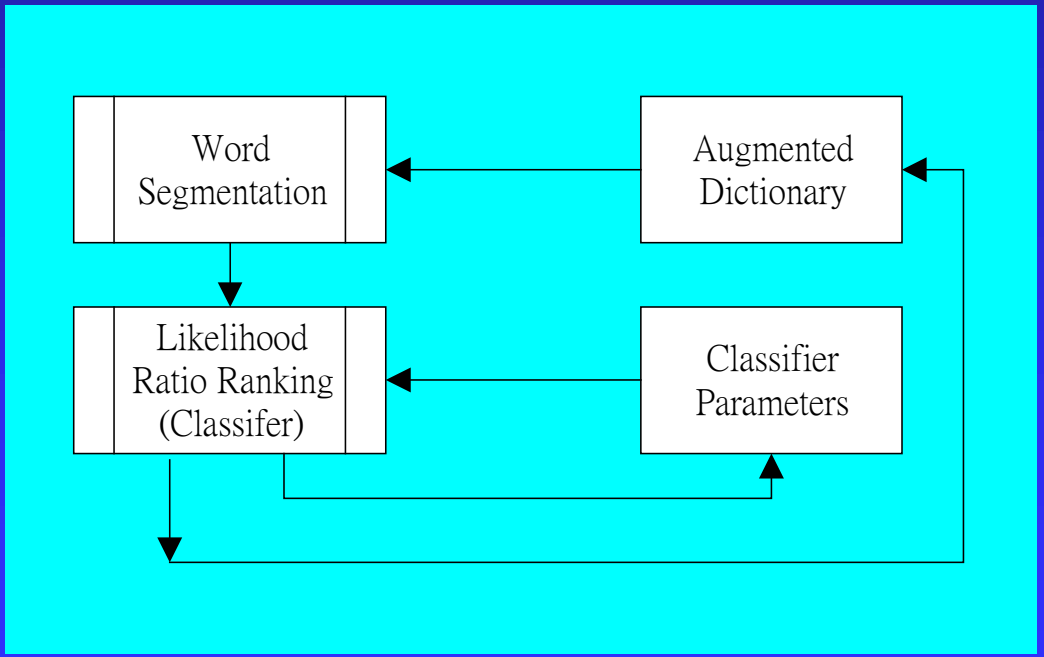
- Association features not used jointly; instead, used independently
  - Worse than jointly considering all association features
- Information cannot be shared between word segmentation and filtering
  - Inherent contextual constraints cannot be used by filter
  - Word association features do not help select candidate word for segmentation module
- Model parameters are not improved iteratively
  - Performance of segmentation and filtering is unlikely to be perfect in only one pass with unsupervised mode

Strategies for Extracting Chinese New Words

- Strategies
  - Use augmented dictionary (system dictionary+high frequency n-grams)
    - to prevent from pre-mature rejection of new words by using only known words for segmentation
    - new words have the chance to compete with known words during segmentation
  - Iterative Approach to provide a chance for improving **recall**:
    - Word Segmentation → Qualification (→ Re-estimate Parameters) → Segmentation → Qualification (→ Re-estimate Parameters) …
    - Why: (See Next Slide)
  - Use a two-class classifier which jointly considering all features: likelihood ratio test
  - Use ranks of likelihood ratio to identify very likely or very unlikely candidates, instead of using the value for filtering out candidates with non-positive values
  - Filter => Likelihood Ratio Ranking Module (aka LRRM)

Basic Language Models and System Architecture

- Integration of the Modules
  - Iteratively apply word segmentation and use the relative rank information of the segments to improve the augmented dictionary for segmentation
    - improve the segmentation parameters and classifier parameters as well



Extracting Chinese New Words

- Why Iterative ?
  - **Recall Improvement:** Truncated candidates could be replaced by other more likely segments (judged by contextual probability) at later segmentation iterations, thus extracting likely new words
    - **Recall** could be improved, in addition to improving precision (by filtering)
    - Joint improvement of precision-recall becomes possible
  - **Information Sharing:** Contextual probability used by Word Segmentation and association features used by filter help each other in improving the model parameters
    - WS: producing better segments iteration by iteration, highly probable new words are moved to the word-class, thus refine two-class classifier model
    - Filter: provide correct candidate ranking for truncating unlikely n-grams, thus improve the dictionary used by the word segmentation module
    - Contextual information and Association features are iteratively integrated

Unsupervised Training for New Word Extraction

- **Initialization:**
  - Initial augmented dictionary = {system dictionary + high frequency n-grams in text (frequency count >=5)}
  - Initial word segmentation probability = relative frequency in text corpus
  - Initial two-class classifier parameters: divide n-grams into word & non-word according to system dictionary & estimate feature distribution for the two classes
- **Jointly train & improve two modules:**
  - Word Segmentation+Ranking Module
    - LRRM: a two-class classifier, using likelihood ratio between word-class and non-word class to rank possibility of an n-gram being a word

Unsupervised Training for New Word Extraction (cont.)

- **Jointly train & improve two modules (cont.)**
  - **Viterbi Training:** for Training Word Segmentation Module:
    - Use initial probabilities for finding the best word segments
    - Re-estimate word probabilities from best segments
    - **Repeat:** until converge or running a specified iterations
  - Sort word list in Word Segmentation results by Likelihood Ratio
  - Delete unlikely words (not in system dictionary) from augmented dictionary
  - **Update word/non-word class parameters of LRRM:** with highly likely new words (change the estimates to the word-class)
- **Repeat:** Joint Training to Iteratively improve the Viterbi-Training and LRRM modules

A System for Chinese New Lexicon Acquisition

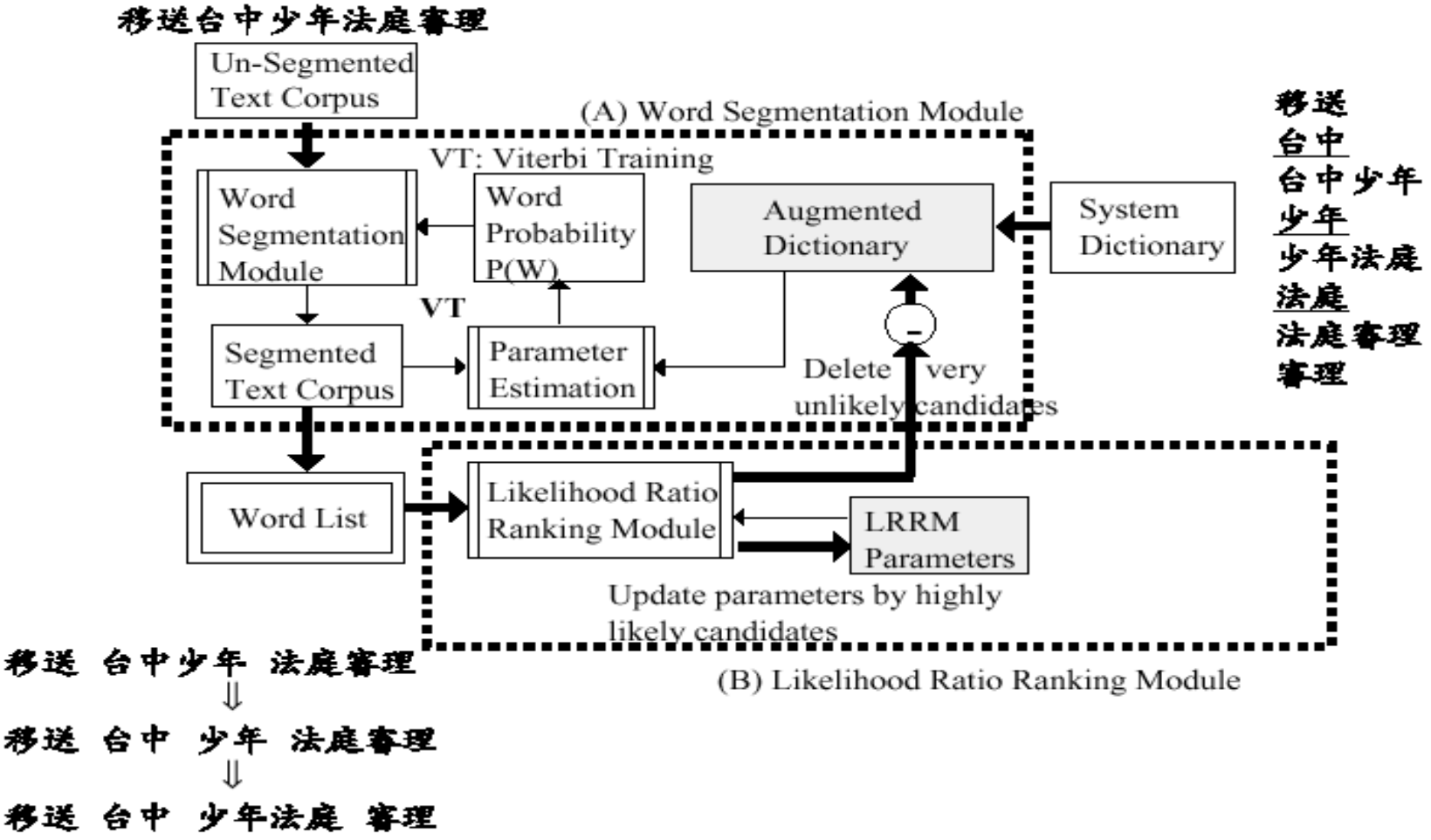


Figure 2 Configuration for Automatic Chinese New Lexicon Acquisition

Viterbi Training for Extracting New Words

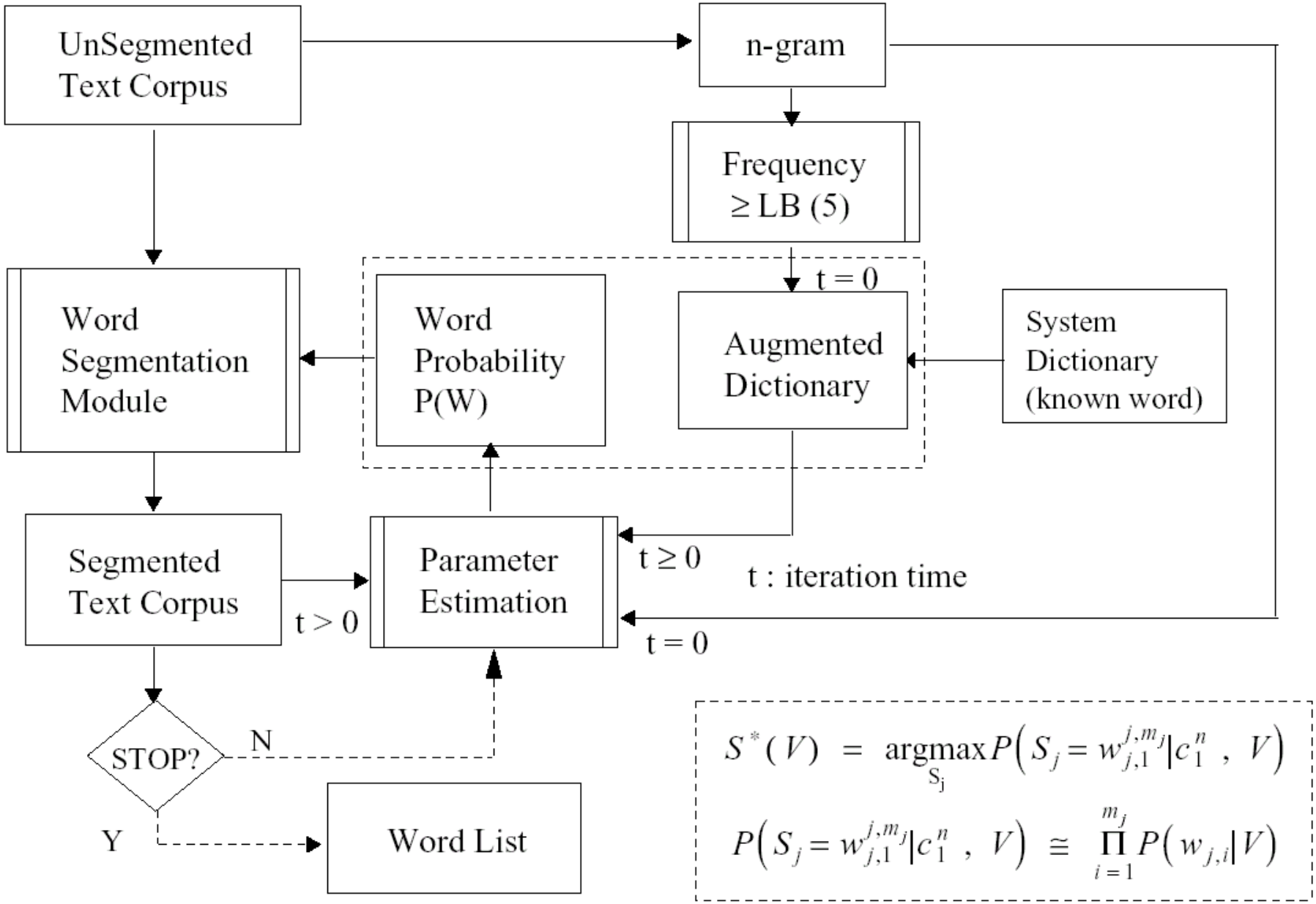


Figure 1 The Viterbi training model for unsupervised new word identification

Viterbi Training for Identifying New Words

- Criteria:
  - 1. produce words that maximizes the likelihood of the input corpus
  - 2. avoid producing over-segmented entries due to unknown words
- Viterbi Training Approach:

  Reestimate the parameters of the segmentation model iteratively to improve the system performance, where the word candidates in the augmented dictionary contain known words and potential words in the input corpus.
- Potential unknown words will be assigned non-zero probabilities automatically in the above process.

Viterbi Training for Identifying Words (cont.)

- Segmentation Stage: Find the best segmentation pattern S*

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

which maximizes the following likelihood function of the input corpus

$$P(S_j = w_{j,1}^{j,m(j)} | c_1^n, V) \approx \prod_{i=1,m(j)} P(w_{j,i} | V)$$

c_1^n : input characters c_1, c_1, ..., c_n

S_j : j-th segmentation pattern, consisting of { $w_{j,1}$, $w_{j,2}$, ..., $w_{j,mj}$ }

$V(t)$: vocabulary (n-grams in the augmented dictionary used for segmentation)

$S^*(V)$: the best segmentation (is a function of V)

Viterbi Training for Identifying Words (cont.)

- Reestimation Stage: Estimate the word probability which maximizes the likelihood of the input text:

Initial Estimation:

$$P(w_{j,i}|V) = \frac{Number(w_{j,i})\ in\ corpus}{Number\ of\ all\ w_{j,i}\ in\ corpus}$$

Reestimation:

$$P(w_{j,i}|V) = \frac{Number(w_{j,i})\ in\ best\ segmentation}{Number\ of\ all\ w_{j,i}\ in\ best\ segmentation}$$

Model for Two-Class Classifier (Log-Likelihood Ratio Ranking Module)

Input: n-grams in the unsegmented text corpus

Output: assign a class label ("word" or "non-word") to each n-gram

Classifier: a log-likelihood ratio (LLR) tester (minimum error classifier)

$$g(\mathbf{x}) = LLR(\mathbf{x}) = \log \frac{f(\mathbf{x}|\mathbf{W})P(\mathbf{W})}{f(\mathbf{x}|\overline{\mathbf{W}})P(\overline{\mathbf{W}})}$$

Decision Rules:

$$class(w) = \begin{cases} +w \quad (word) & if \quad LLR(\bullet) \geq \lambda_0 \\ -w \quad (non-word) & if \quad LLR(\bullet) < \lambda_0 \end{cases}$$

Advantage: ensure minimum classification error (with λ_0 =0) if the distributions are known.

NOTE: the associated LLR's are used for sorting to identify **relative ranking order** of character n-grams. We don't really use it for assigning class label.

Features for the Classifier

- Mutual Information: characters *x* and *y* with high mutual information tend to have high association [Church 90]

$$I(x,y) = log\frac{P(x,y)}{P(x) \times P(y)}$$

- Entropy: random distribution of the left/right neighbors (C_i) of an *n*-gram *x* implies a natural break at the *n*-gram boundary [Tung 94]:

$$H(x) = -\sum_{c_i} P(c_i;x) log P(c_i;x)$$

Combining Viterbi Training and Two-Class Classifier

- Why: "Viterbi Training+Two-Class Classification" iteratively?
  - using individual module or cascading them does not fully use information of other modules
- How: The best segmentation is a function of the vocabulary && Classifier performance is a function of its parameters ... so ...
- An iterative integration approach:
  - Segment input with the augmented dictionary using Viterbi Training
  - Filtering out unlikely candidates from the current augmented dictionary
  - Update class labels of the classifier's training n-grams, according to the best segmentation, to improve the estimated classifier parameters.
  - Repeat the segmentation-classification sessions using progressively refined augmented dictionaries and classifier parameters.

Integrated System for New Word Identification

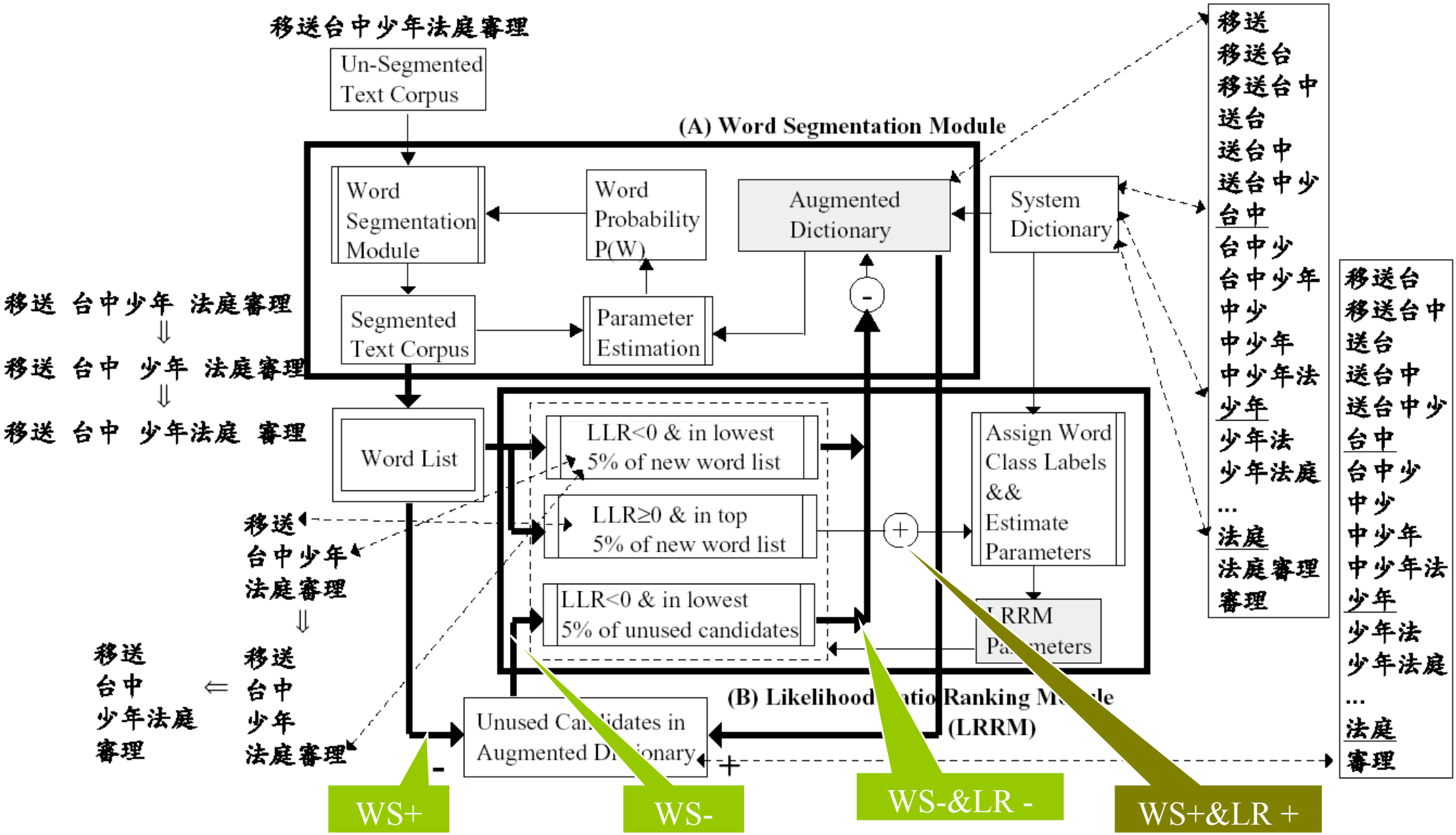


Figure 2 The integrated system for unsupervised new word identification

Refinement of Augmented Dictionary & Refinement of Classifier Parameters

- Refine augmented dictionary
  - truncate the worst 5% new words of the segmentation output from the augmented dictionary
    - so that they won't appear in later segmentation sessions
  - truncate the worst 5% augmented dictionary entries which do not appear in the segmentation output
    - so as to reduce processing time
- Refine class labels && classifier parameters
  - re-assign the class labels of the best 5% new words of the segmentation output to "word"
    - so that classifier parameters will be more reliably estimated

Performance of the Integrated System (cont.)

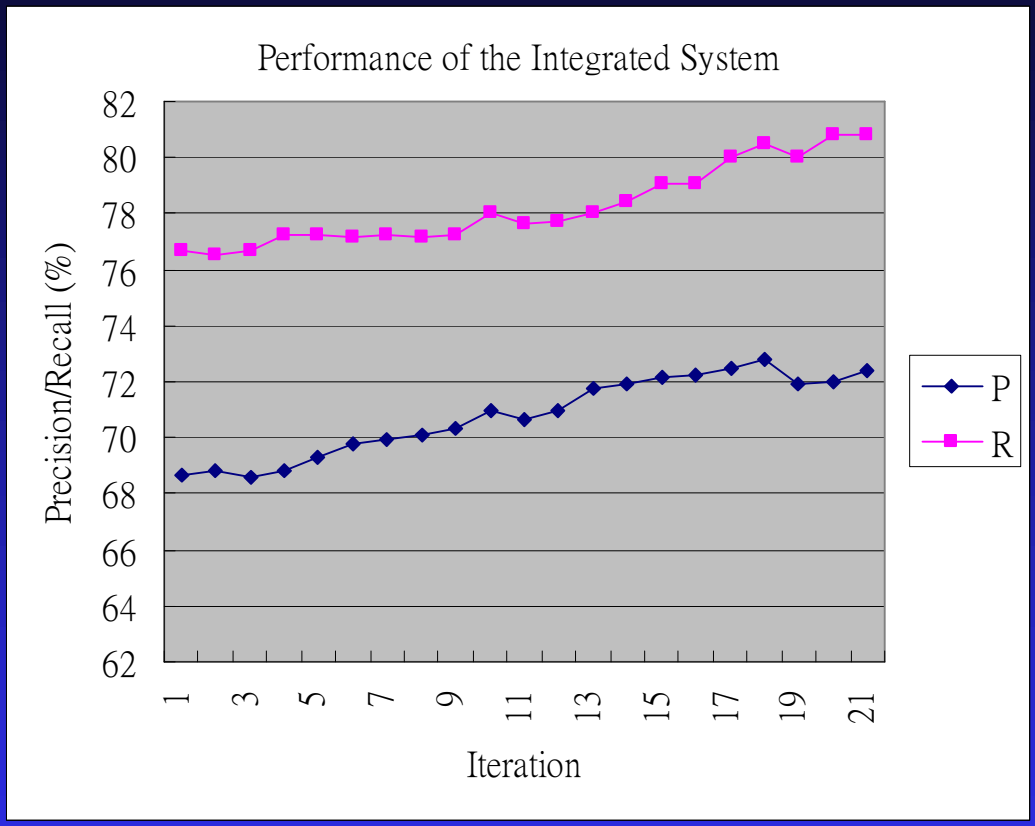


Figure 6. Performance for Identifying New Words in Each Iteration (bigram new words).

- Precision and recall are both improved almost monotonically without sacrificing one performance for another.

Summary on Quantitative Analysis

$$p = \frac{N_{ww}}{N_{ww} + N_{xw}} = \frac{1}{1 + N_{xw} / N_{ww}}$$

$$r = \frac{N_{ww}}{N_{ww} + N_{wx}} = \frac{1}{1 + N_{wx} / N_{ww}}$$

- Most contribution of the F-measure and WPR comes from the improvement in precision
- n_{ww}: +5% (2-gram), +8% (3-gram), about constant for 4-grams
- n_{xw}: -12% (2-gram), -30% (3-gram), and -52% (4-gram)
  - the improvement in precision is mostly attributed to the decrease in n_{xw}.(i.e., truncating unlikely candidates from augmented dictionary)
- True words for truncated words are *recovered* via *re-segmentation*:
  - => N_{ww} increased => N_{wx} decreased => recall increased

Example of Extracted New Words

Example of Extracted New Words					
Bigram New Word		Trigram New Words		Quadgram New Word	
Proper Names					
鹿谷	Lu-Gu; a county name	中新社	China News Service	曾蔡美佐	a female name
蓋茲	(Bill) Gates	富士通	Fujitsu	新興分局	Hsin-Hsing police office
住友	a company name	翁秀卿	a female name	富岡國小	Fu-Gang Primary School
Ordinary Words					
護法	guard	管理局	Bureau of Administration	年度預算	annual budget
幹員	talented (police)men	養豬戶	pig-raising farmers	全球股市	global stock markets
鑑於	in view of	下半年	second half of the year	貨幣市場	monetary market
共舞	dance (with somebody)	投機風	opportunism	國家公園	national park
責令	command	收盤價	closing price	生命安全	personal security

Example of Extracted New Words (cont.)

Example of Extracted New Words (cont.)					
Abbreviation					
市警	city policemen	國台辦	Taiwan-Affair Office of National Affair House	省都委會	provincial city development committee
中菲	Sino-Philippine	消基會	the Consumer Protection Committee	紅會人員	the Red Cross staffs
鄉代	county representatives	上下班	go-to-and/or-come-back-from the office	投開票所	polls
Collocational Strings					
就會	will then ...	據指出	it was indicated that ...	絕大多數	overwhelming majority
既非	neither	並沒有	do not	一片混亂	a mess
Derivational Words					
廠方	authority of the company	壽險業	life insurance companies	所有權人	owner
		複雜化	complicate		
Numerical Strings					
一萬	ten thousands	十四日	14th day of the month	八十年度	1991 accounting year

Distribution of Acquired New Words

Distribution of Acquired New Words						
n-gram	P(%)	A(%)	D(%)	C(%)	O(%)	#(%)
2	9	2	0	16	67	6
3	34	5	21	7	23	10
4	5	4	1	5	82	3

Table 14. Distribution of correctly identified words (P: proper names, A: abbreviational words, D: derived words, C: collocational strings, O: other ordinary new words)

Distribution of Errors

Distribution of Errors						
n-gram	P(%)	A(%)	D(%)	C(%)	O(%)	#(%)
2	13	6	4	26	37	14
3	25	8	5	3	17	42
4	24	5	0	0	24	47

Table 15. Distribution of incorrectly identified words. (Word => non-Word)

n-gram	P(%)	A(%)	D(%)	C(%)	O(%)	#(%)
2	25	0	0	47	12	16
3	5	0	0	13	59	23
4	10	3	2	41	20	24

Table 16. Distribution of spurious words that are recognized as words. (non-Word=>Word) (The P, A, D, C, O, # types indicate the major origin of the non-Word

Concluding Remarks

- Supervised Learning for Precision-Recall Maximization:
  1. Two-stage strategy can be used to maximize precision-recall by first minimizing classification error using a well designed two-class classifier, and then maximizing the joint precision-recall.

  2. When designing the minimum error classifier, various association metrics should be used *jointly* to minimize the classification error. The feature correlation should also be considered in modeling the density function.

  3. Joint Precision-Recall performance can be maximized by *learning (I.e.,* adjusting) the classifier parameters to reduce a risk function defined on precision and recall.

Concluding Remarks

- Unsupervised Learning for Precision-Recall Maximization:

  1. An iterative scheme for precision-recall maximization can be used to integrate two knowledge sources (the segmentor and filter information, by truncating unlikely candidates in the augmented dictionary and updating the filter/classifier parameters.)

  2. Precision can be improved by filtering out inappropriate candidates; Recall can be improved by re-segmentation (using contextual information). Iterative integration thus improve both without sacrificing precision for recall or *vice versa*.

Thanks …