Automatic Lexicon Acquisition and Precision-Recall Maximization for Untagged Text Corpora

Jing-Shin Chang (shin@hermes.ee.nthu.edu.tw)

Natural Language Processing Laboratory
Department of Electrical Engineering, National Tsing-Hua University
Hsinchu, Taiwan 30043, ROC.

July 29, 1997, CS/EE Bldg., NTHU

Table of Contents

- Why Automatic Lexicon Acquisition
- P-R (Precision-Recall) Maximization Problems
- English Compound Word Extraction (Supervised)
  - Learning Classifier Parameters for Precision-Recall Maximization
- Chinese New Lexicon Identification (Unsupervised)
  - Unknown Word Problems
  - Word Segmentation using Viterbi Training with Augmented Dictionary
  - Iterative Integration of Word Segmentor && Classifier for Precision-Recall Maximization

What is Lexicon Acquisition (English)

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. (Microsoft Word User Guide)

(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)

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, associated input methods

2. New (unknown) words && compound words increase rapidly (e.g., 修憲、凍省、反凍、反反凍、本尊、分身)

   - vary with *time*   - vary with *domain*

3. Prefer to lexicalize for easier: disambiguation (analysis), compositionality (generation)

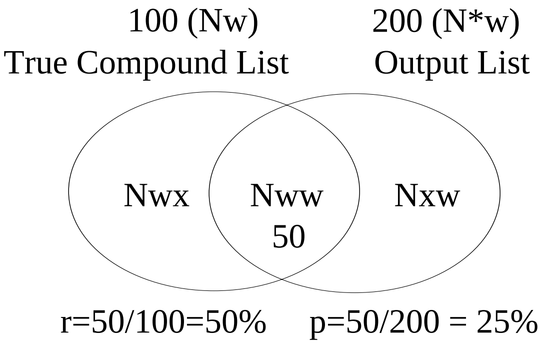
   e.g., book (n, vi, vt) + store (n, vt) <=> book store (n)
   e.g., green house ≠ ‘green’ + ‘house’

4. Human construction is costly, time consuming and inconsistent

5. Electronic text is becoming widely available

*. Target for acquisition: compound words, unknown words

Precision-Recall Optimization Criteria



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

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

p = Nww/(Nww+Nxw) = #correct_identification / #output_words

r = Nww/(Nww+Nwx) = #correct_identification / #all_words

(Nij: # of class-i n-grams which are classified as class-j)

(i, j= w - word//compound ; x - non-word//non-compound)

⇒ Typical Joint Criteria for Precision (p) and Recall (r) Maximization:

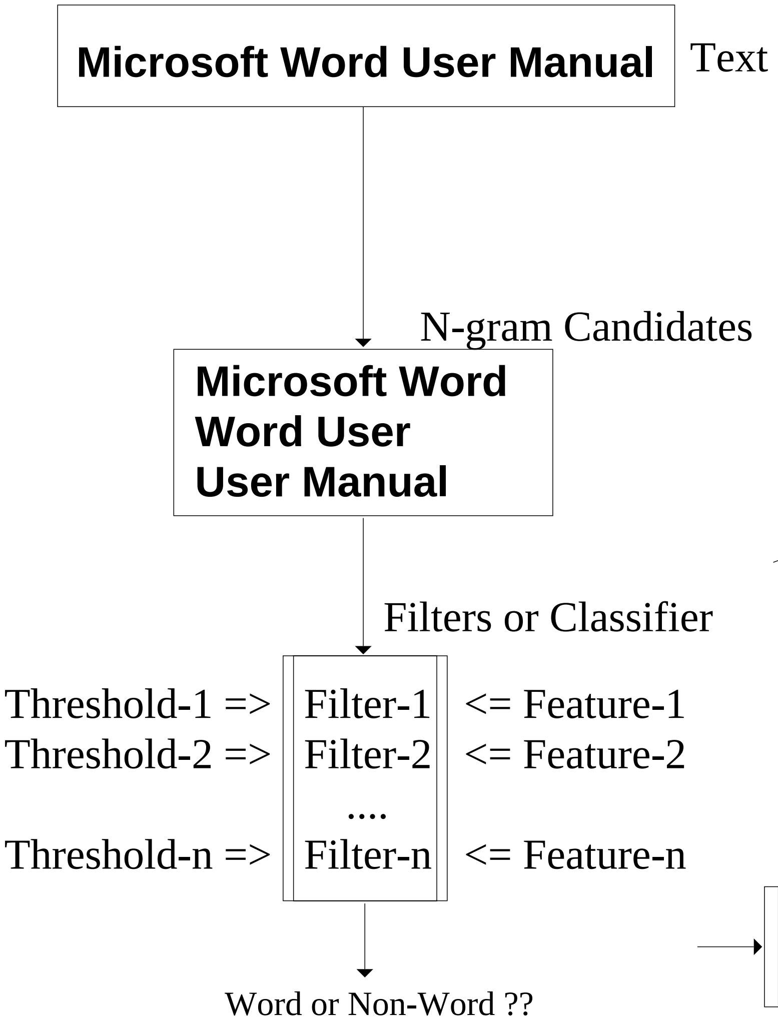
✏ WPR: Wp*p+Wr*r (weighted Precision/Recall)

[Wp, Wr: weights (Wp+Wr=1)]

— A weighting sum of precision and recall.

- F-metric (F-measure): $F(\beta) = \frac{(\beta^2+1)pr}{\beta^2 p + r} = \frac{pr}{p\beta^2/(\beta^2+1) + r/(\beta^2+1)}$
  - A metric that appreciate 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)

General Scheme in English Compound Extraction



Microsoft Word User Manual

Text

N-gram Candidates

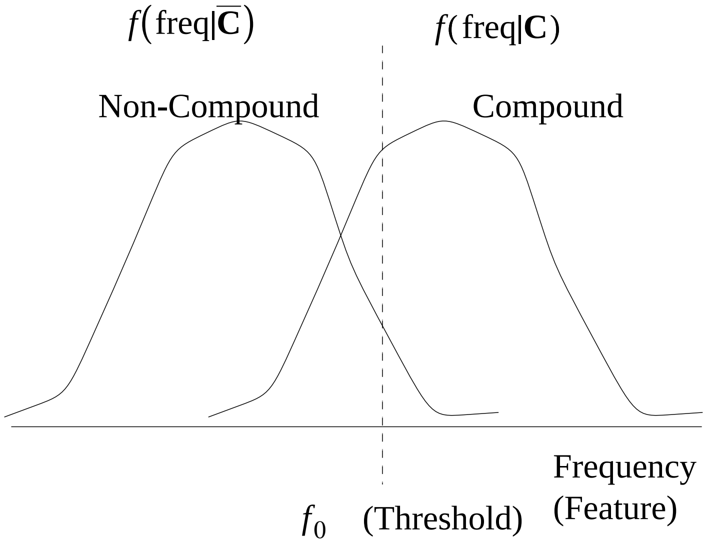
Microsoft Word
Word User
User Manual

Filters or Classifier

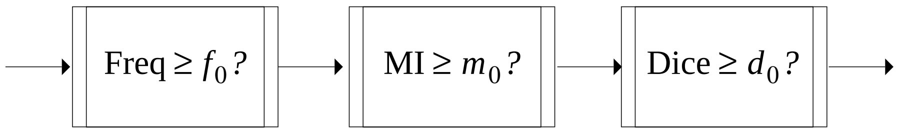
Threshold-1 => Filter-1 <= Feature-1
Threshold-2 => Filter-2 <= Feature-2
....
Threshold-n => Filter-n <= Feature-n

Word or Non-Word ??

Two-Class Classification Problem



Filtering Approaches

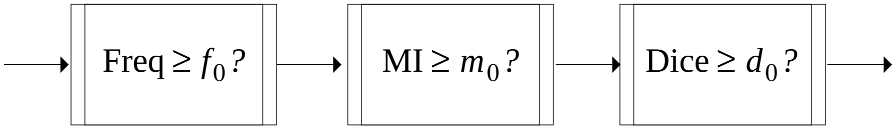


Freq ≥ f_0? → MI ≥ m_0? → Dice ≥ d_0?

General Problems in Lexicon Acquisition

➨ Use simple *filtering* approaches and heuristic thresholds in extracting lexicon entries

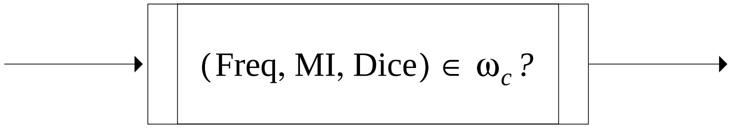
☆ mostly based on step-by-step filtering approaches which filter out inappropriate candidates with one feature per step



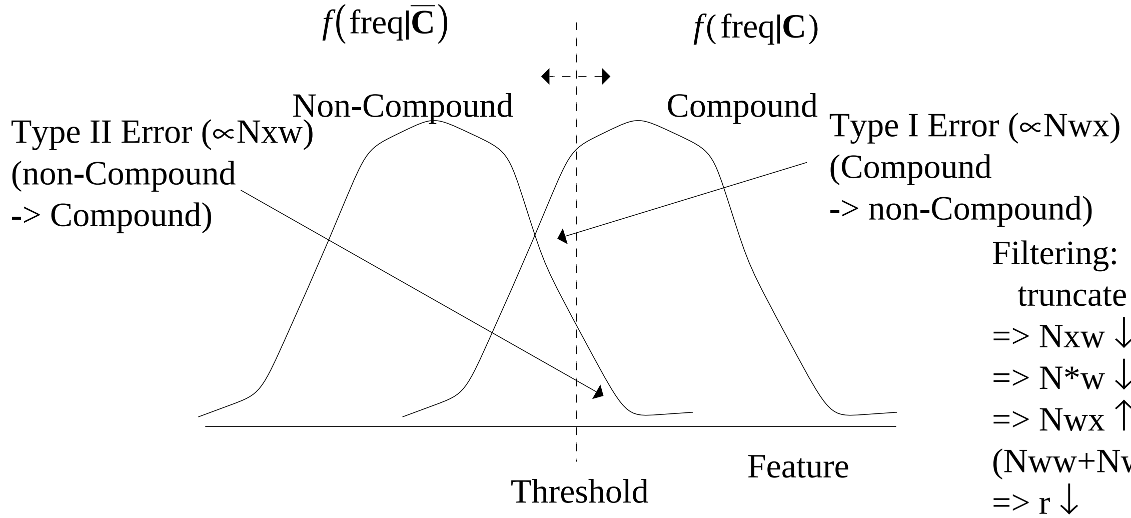
☆ thresholds are determined by trial-and-error

☆ no unified method for integrating known features

- known features are used *independent* of one another
- no automatic method for identifying the best feature



Precision-Recall Maximization Problems



$f(\text{freq}|\overline{\mathbf{C}})$

$f(\text{freq}|\mathbf{C})$

Non-Compound

Compound

Type II Error (∝Nxw)
(non-Compound
-> Compound)

Type I Error (∝Nwx)
(Compound
-> non-Compound)

Threshold

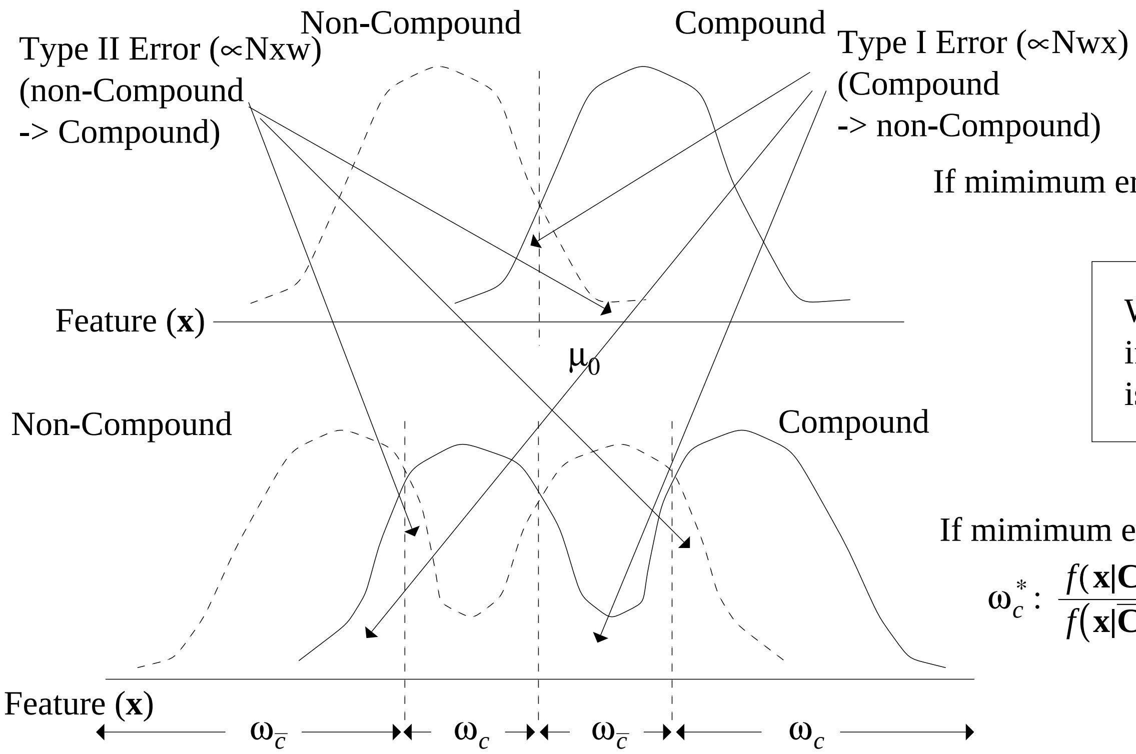
Feature

Filtering:
truncate bad output entries
=> Nxw ↓ => p↑
=> N*w ↓ =>Nww ↓
=> Nwx ↑
(Nww+Nwx=Nw=const)
=> r ↓

- 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

- Why two stage?
  - ☆ No simple analytical decision rules that are capable of achieving any user-specified criterion function of precision and recall



$$e \propto N_{wx} + N_{xw}$$

If mimimum error => $\mu^* = \mu_0$

What are the best decision rules
if Max O(Precision,Recall)
is the optimization criterion ??

If mimimum error =>

$$\omega_c^*: \frac{f(\mathbf{x}|\mathbf{C})\,p(\mathbf{C})}{f(\mathbf{x}|\bar{\mathbf{C}})\,p(\bar{\mathbf{C}})} \geq 0$$

➠ Which two stage?

☆ minimize classification error:

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

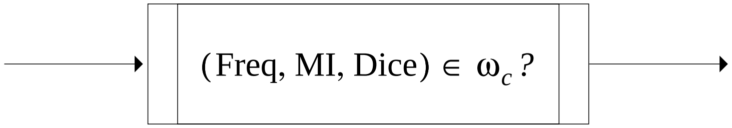
reduce error rate (Nwx+Nxw) generally improve P, R and other joint functions (Note: Maximize FM == Minimize $\left(n_{wx} + n_{xw}\right)/n_{ww}$)

☆ maximize precision-recall:

Min error classification ≠ MaxPR classification

➠ How to?

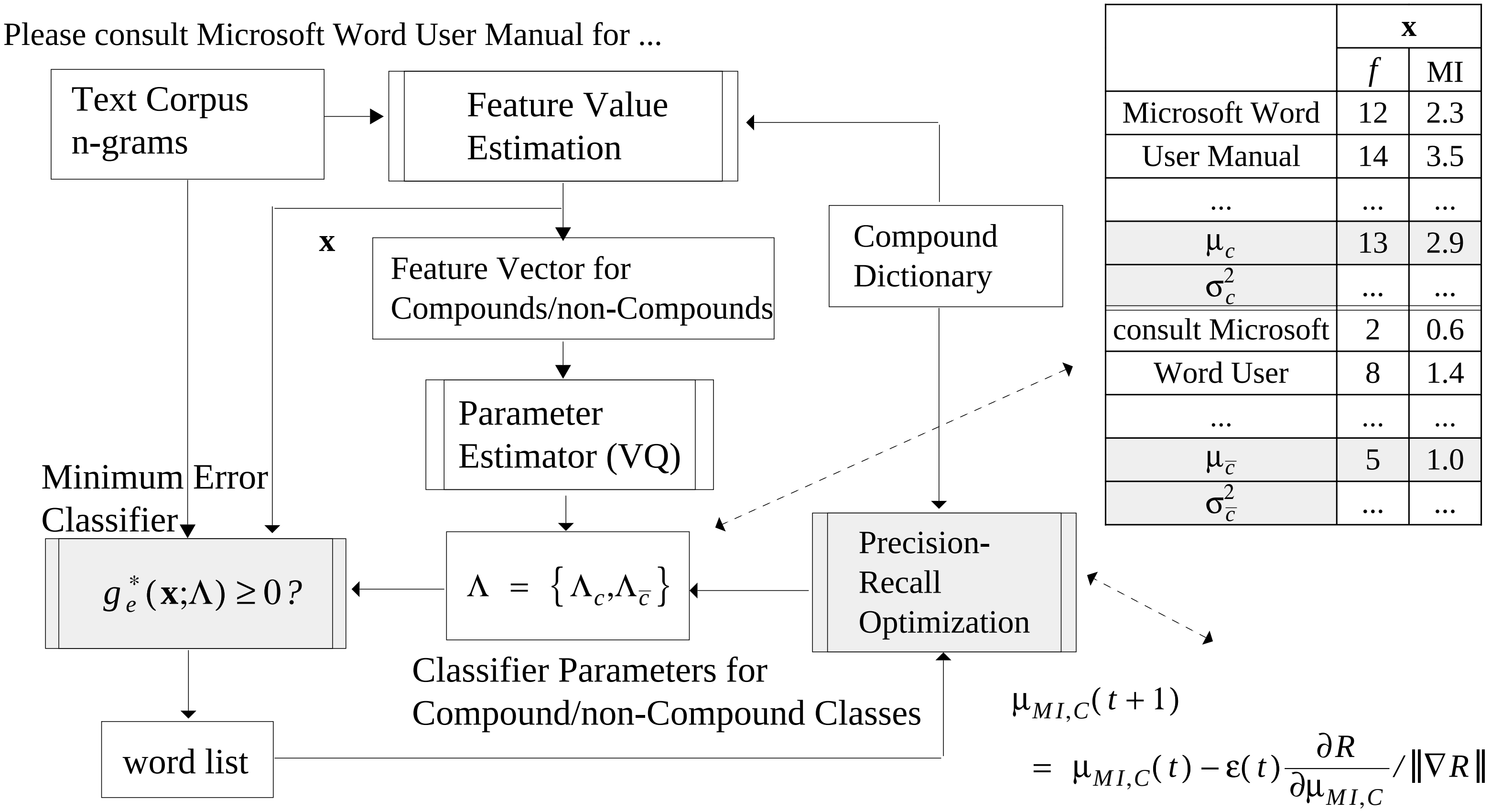
☆ minimum error classification: better features, better models for jointly combining all features, better estimation



$(\text{Freq, MI, Dice}) \in \omega_c\,?$

☆ maximize precision-recall: by parameter learning (nonlinear!!)

MinErr Classifier+MaxPR Learning Approach



Please consult Microsoft Word User Manual for ...
Text Corpus n-grams
Feature Value Estimation
x
Feature Vector for Compounds/non-Compounds
Compound Dictionary
Parameter Estimator (VQ)
Minimum Error Classifier
$g_e^*(\mathbf{x};\Lambda) \geq 0\,?$
$\Lambda = \{\Lambda_c, \Lambda_{\bar{c}}\}$
Precision-Recall Optimization
Classifier Parameters for Compound/non-Compound Classes
word list

	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
$\sigma_{\bar{c}}^2$	...	...

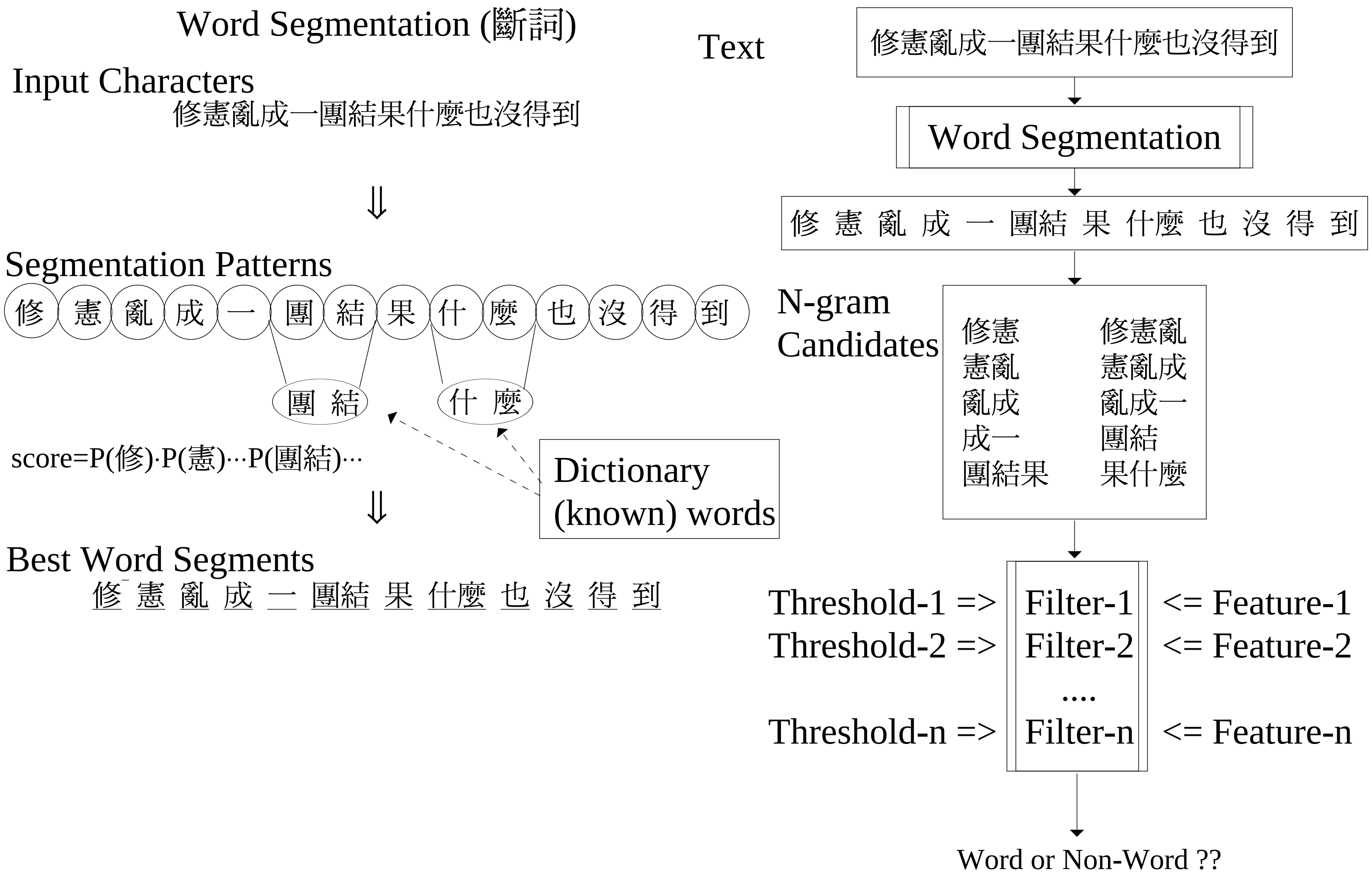
$$\mu_{MI,C}(t+1) = \mu_{MI,C}(t) - \varepsilon(t)\frac{\partial R}{\partial \mu_{MI,C}} / \|\nabla R\|$$

Figure 1 Supervised Training of Classifier Parameters for English Compound Extraction

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



Word Segmentation (斷詞)

Input Characters

修憲亂成一團結果什麼也沒得到

⇓

Segmentation Patterns

修 憲 亂 成 一 團 結 果 什 麼 也 沒 得 到

團 結 什 麼

score=P(修)·P(憲)···P(團結)···

Dictionary (known) words

⇓

Best Word Segments

修 憲 亂 成 一 團結 果 什麼 也 沒 得 到

Text

修憲亂成一團結果什麼也沒得到

Word Segmentation

修 憲 亂 成 一 團結 果 什麼 也 沒 得 到

N-gram Candidates

修憲	修憲亂
憲亂	憲亂成
亂成	亂成一
成一	團結
團結果	果什麼

Threshold-1 => Filter-1 <= Feature-1

Threshold-2 => Filter-2 <= Feature-2

....

Threshold-n => Filter-n <= Feature-n

Word or Non-Word ??

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’)

☞ Integration of Knowledge Sources:

- Combine information sources by cascading the above modules using one-pass, non-iterative cascaded scheme

Problems with Segmentation Using Known Words

☞ Incomplete Error Recovery Capability

- Two types of segmentation errors due to unknown word problems:

  - Over-segmentation: Split unknown words into short segments (e.g., single character regions ‘修憲’=> ‘修 憲’)

    ✏ 分析家 對 馬來西亞 的 預測
    <=> 分析 家 對 馬 來 西亞 的 預測

  - Under-Segmentation: Prefer long segment when combining segments (搶詞問題)

e.g., ‘土地 公有 政策’ =WS Error (‘公有’ unknown)=> ‘土地公 有 政策’
=Merge=> ‘土地公有’, ‘有政策’ (NOT: ‘土地’, ‘公有’, ‘政策’)
✏ 團結: mis-merge=> 修 憲 亂 成 一 團結 果 什麼 也 沒 得 到

✰ MERGE operation ONLY recover over-split candidates but NOT over-merged (under-segmented) candidates

Problems with Segmentation Using Known Words

☞ Use *known words* for segmentation without considering potential unknown words (zero word probabilities to unknown words)

- cannot take advantages of contextual constraints over unknown words to get the desired segmentation
- millions of randomly merged unknown word candidates for filter

  (-省都委會:) 獲省都委會同意 => 獲 省 都 委 會同 意

  =>省都|省都委|省都委會|都委|都委會同|委會同|委會同意

  (+省都委會:) 獲省都委會同意 => 獲 省都委會 同意
- an extra disambiguation step for resolving overlapping candidates

  e.g., 省都 vs 省都委會 (etc.)

  e.g., 彰化 縣 警 刑警隊 少年組

Problems with Non-iterative Scheme

☞ *Non-iterative* scheme without sharing information among modules and between iterations to progressively refine performance

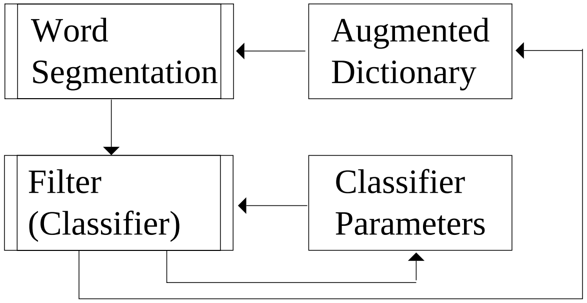
- Segmentor: Do not use association features of the filter to get better segmentation results (even with good initial guess on potential unknown words)

- Filter: Do not use contextual constraints over unknown words for forming likely candidates and resolving overlapping ambiguities

Strategies for Chinese Unknown Word Extraction

☞ Use an augmented dictionary to recover 2 types of segmentation errors

- augmented dictionary: system (known word) dictionary + *potential unknown words* in input corpus
- only those highly potential unknown words will be submitted to filter (without merging)



☞ Use the filter to truncate *a fraction of most unlikely candidates* from the augmented dictionary based on association metrics

☞ Iterative Scheme: Use the refined augmented dictionary for re-segmentation and re-filtering iteratively.

Advantages

☞ Segmentor: Now use association features and progressively refined augmented dictionaries to get better segmentations.

☞ Filter: use contextual constraints for forming only likely candidates

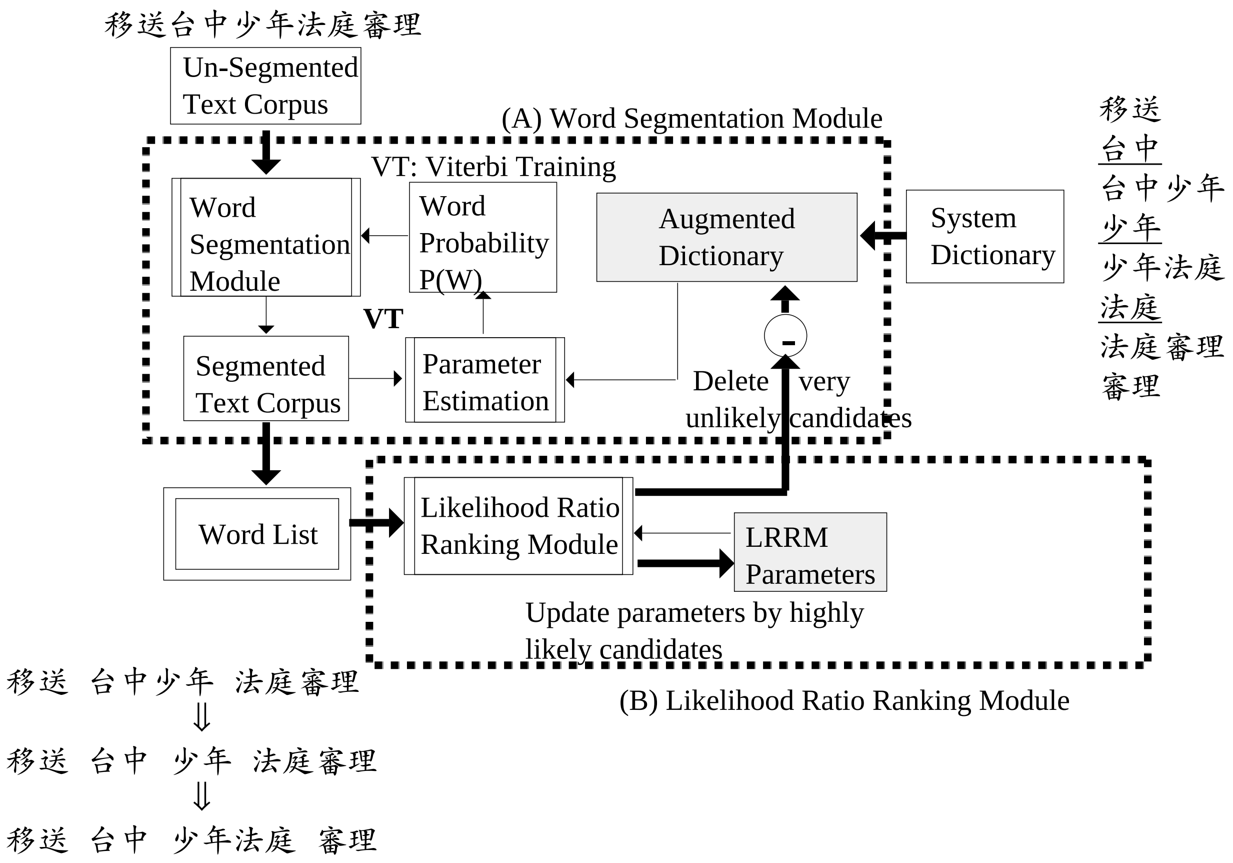
& Use improved segmentation to improve classifier parameters (=> better filtering)

☞ Do not need a merger & a disambiguator - merge & disambiguate is resolved by the segmentation module, following maximum likelihood constraints (contextual constraints) automatically

✰ Iterative scheme allows us to recover identification errors made in previous iterations, and thus improve recall (in addition to improving precision by filtering) e.g.,

(移送) 台中少年 法庭審理=>台中 少年 法庭審理=>台中 少年法庭 審理

A System for Chinese New Lexicon Acquisition



移送台中少年法庭審理

Un-Segmented Text Corpus

(A) Word Segmentation Module

VT: Viterbi Training

Word Segmentation Module

Word Probability P(W)

Augmented Dictionary

System Dictionary

VT

Segmented Text Corpus

Parameter Estimation

-

Delete very unlikely candidates

Word List

Likelihood Ratio Ranking Module

LRRM Parameters

Update parameters by highly likely candidates

(B) Likelihood Ratio Ranking Module

移送
台中
台中少年
少年
少年法庭
法庭
法庭審理
審理

移送 台中少年 法庭審理
⇓
移送 台中 少年 法庭審理
⇓
移送 台中 少年法庭 審理

Figure 2 Configuration for Automatic Chinese New Lexicon Acquisition

Summary: P-R Maximization in Typical Tasks

➨ Classifier+Features: English Compound Word Extraction

✰ Use a filter or classifier with multiple features for filtering

➨ Focus on design of an optimal (MaxPR) classifier

- exist ?? what if non-existent ??

➨ Segmentor+Classifier+Features: Chinese Unknown Word Extraction

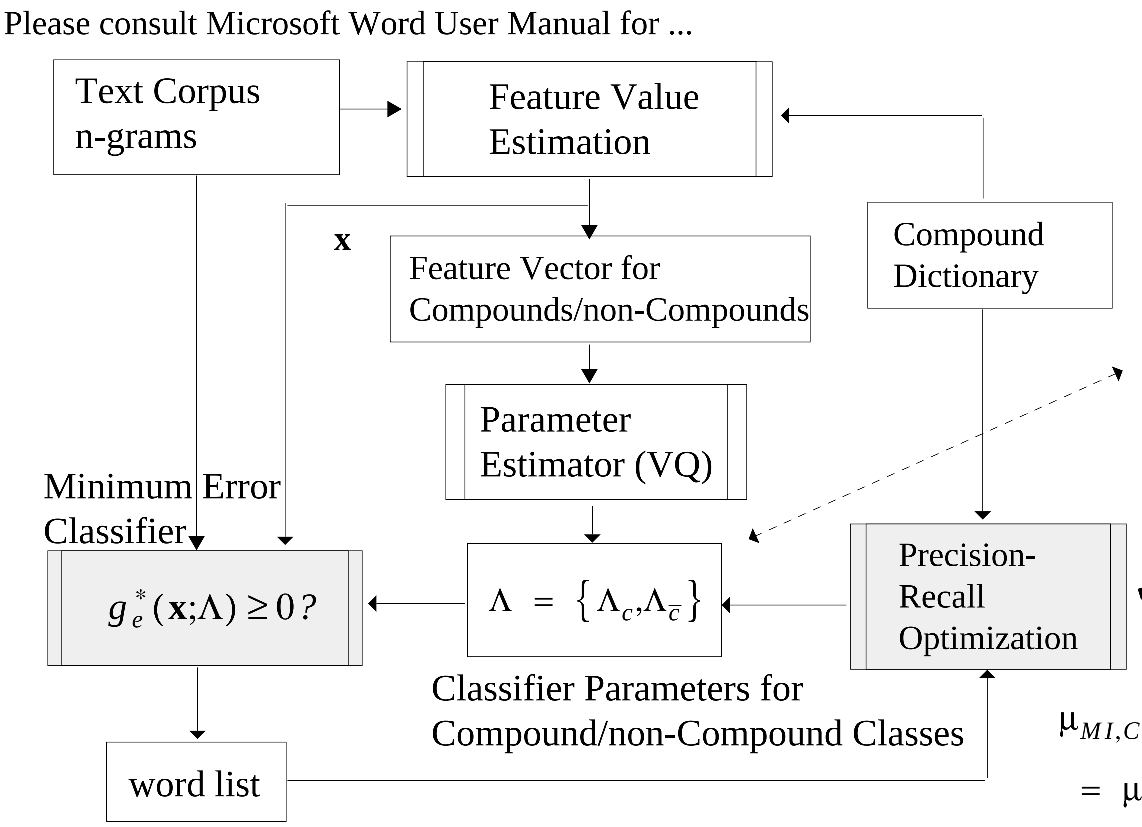
✰ Use additional information for submitting potential candidates to classifier (or other related subtasks)

➨Focus on integration of all information among modules for improve P-R simultaneously (by improving individual modules)

- cascade approaches (traditional)
- iterative integration (simpler, *recovering previous errors*)
  (!! we are here)
- joint modeling (optimal, to do...)

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

MinErr Classifier+MaxPR Learning Approach



Please consult Microsoft Word User Manual for ...

Text Corpus n-grams

Feature Value Estimation

$\mathbf{x}$

Feature Vector for Compounds/non-Compounds

Parameter Estimator (VQ)

Compound Dictionary

Minimum Error Classifier

$g_e^*(\mathbf{x};\Lambda) \geq 0\,?$

$\Lambda = \{\Lambda_c, \Lambda_{\bar{c}}\}$

Precision-Recall Optimization

Classifier Parameters for Compound/non-Compound Classes

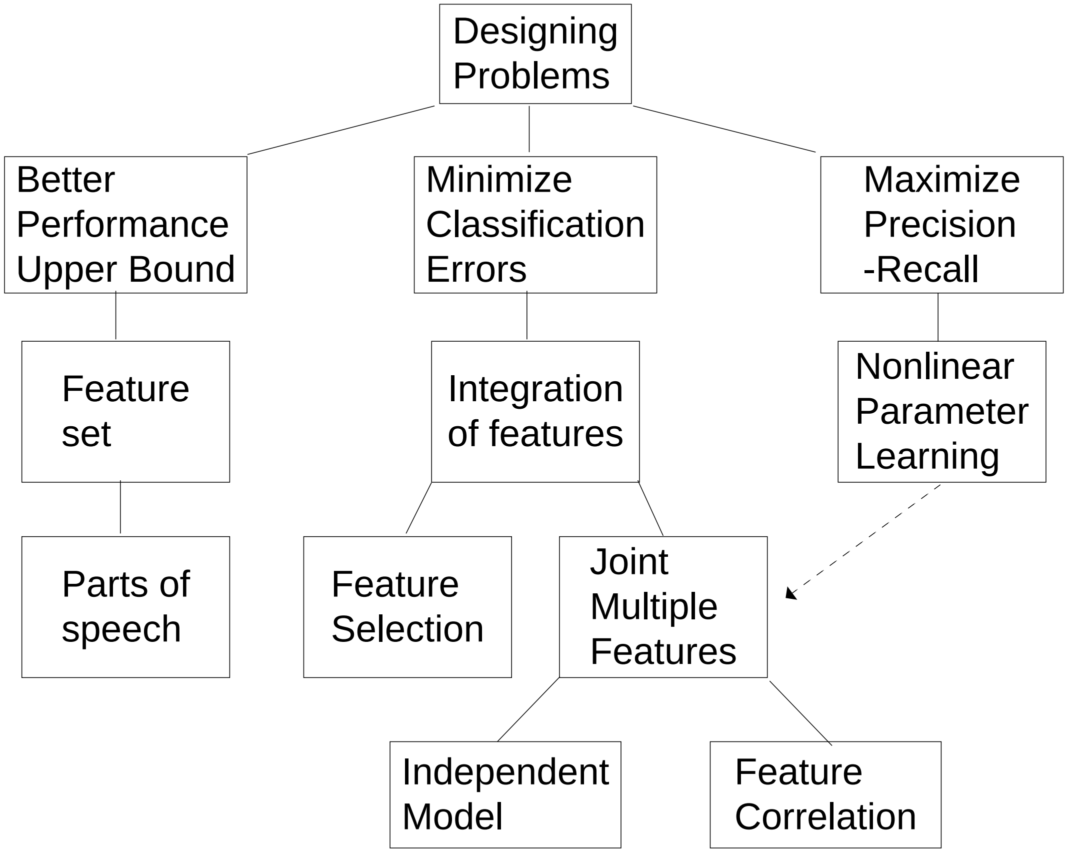
word list

	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
$\sigma_{\bar{c}}^2$	...	...

$$\begin{aligned}&\mu_{MI,C}(t+1)\\&= \mu_{MI,C}(t) - \varepsilon(t)\frac{\partial R}{\partial \mu_{MI,C}} / \|\nabla R\|\end{aligned}$$

Figure 3 Supervised Training of Classifier Parameters for English Compound Extraction

General Problems in Classifier Design



General Problems in Classifier Design

- ✏ Feature Extraction: (will not be addressed)
  - ✓ extract most discriminative features for the task
- ✏ Better Feature Set:
  - ✓ including high level features such as parts of speech
- ✏ Automatic Feature Selection:
  - ✓ adopt a unified feature selection mechanism for all available features so that (1) complementary features are used jointly, instead of being applied independently, and (2) the most appropriate features are applied automatically and less discriminative or redundant features are rejected

General Problems in Classifier Design (cont.)

- ✑ Classifier Design:
  - ✓ design appropriate decision rules for qualifying word candidates using known features jointly
- ✑ Parameter Estimation:
  - ✓ estimate statistical parameters for the classifier to fit particular *estimation criteria* (e.g., maximum likelihood estimation)
- ✑ Performance Maximization:
  - ✓ adjust statistical parameters to maximize desired *performance criteria* (e.g., a joint precision-recall performance such as F-metric)

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 $f(x) = \text{freq/avrg\_freq}$: 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 (C_i) of an n-gram x implies a natural break at the n-gram boundary [Tung 94]:

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

- 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:

$$D_{pos}\left(x_i;\{P_{ij}\},\{P_j\}\right) = \sum_j P_{ij} \, log \frac{P_{ij}}{P_j}$$

$$P_{ij} \equiv P\left(j|w_i\right), \qquad P_j \equiv P(j)$$

An n-gram, Xi, 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.

Pij: probability for Xi to be tagged with part-of-speech pattern j (e.g., j = [n n] for a noun-noun compound word).

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

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	0.0	4.87	6.01	12.33	8.07	0.0	1.35	2.69	36.77
	Precision	100.0	*	30.92	30.69	37.07	100.0	*	23.08	33.33	57.75
	Error Rate	11.03	12.41	13.15	13.34	13.47	21.20	23.06	23.78	23.68	20.79
	WPR(1:1)	55.54	*	17.90	18.35	24.70	54.03	*	12.22	18.01	47.26
	F-measure	19.97	*	8.41	10.05	18.50	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	0.0	13.99	10.20	7.58	0.0	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	*	*	35.20	35.06	40.37
	F-measure	*	*	21.00	14.05	11.69	*	*	20.00	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
Mx: Mixtures of Gaussian Density Functions

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

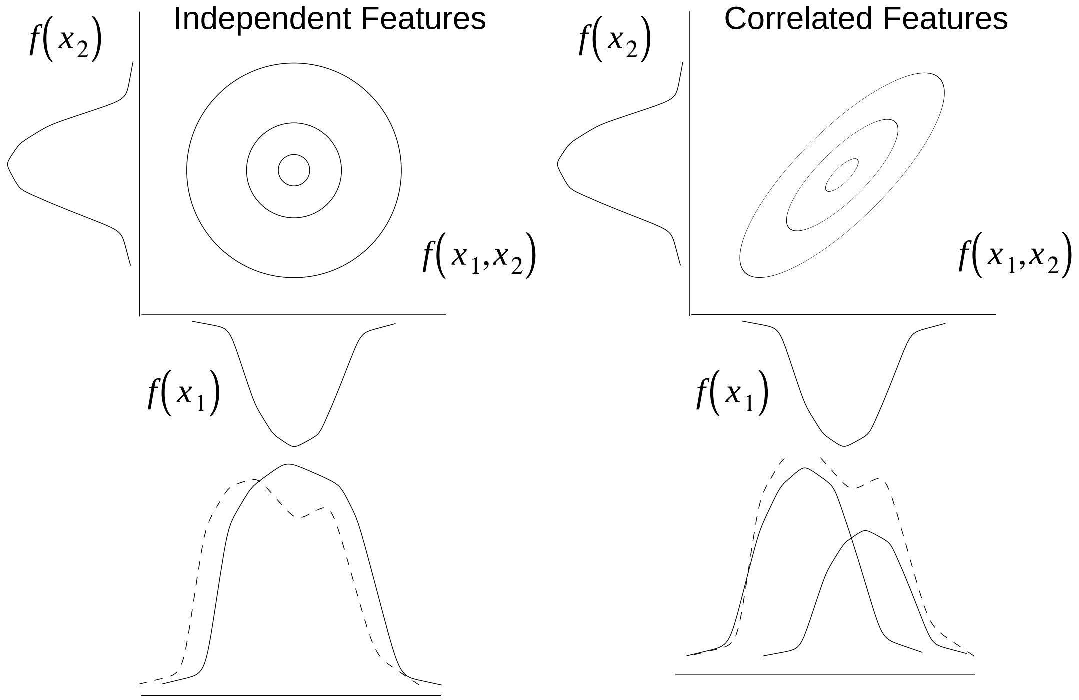
		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	11.09	40.41	54.61	35.34	31.30	8.07	35.43	60.54	33.63	50.67
	Precision	100.0	88.04	77.39	71.04	49.67	100.0	89.77	92.47	82.42	66.47
	Error Rate	11.03	8.07	7.61	9.81	12.46	21.20	15.82	10.24	16.96	17.27
	WPR(1:1)	55.54	64.23	66.00	53.19	40.49	54.04	62.60	76.51	58.03	58.57
	F-measure	19.97	55.39	64.03	47.20	38.40	14.93	50.81	73.17	47.77	57.50
Feature Sequence		Dpos	MI	H	D	NF	Dpos	MI	H	D	NF
3-gram	Recall	0.0	14.29	33.53	29.45	26.24	0.0	17.24	44.83	56.90	48.28
	Precision	*	100.0	70.99	46.98	33.83	*	100.0	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	12.10
	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	36.20	29.56	*	29.41	59.09	52.80	47.86

Table 2 Error rate performances 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)



Normal approximation

Mixture of density functions

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

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

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

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

		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.50	71.61	50.67	51.71	69.06	71.30	69.96	67.26	47.09
	Precision	100.0	97.87	88.93	62.93	45.53	100.0	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	56.80	48.62	84.53	83.54	81.68	73.95	49.66
	F-measure	82.24	82.63	79.34	56.14	48.42	81.70	81.75	80.00	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	51.90	54.23	75.86	74.14	74.14	36.21	37.93
	Precision	100.0	95.12	90.91	80.91	39.08	100.0	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	66.40	46.65	87.93	85.93	84.85	65.83	39.72
	F-measure	77.50	79.45	77.18	63.24	45.43	86.27	84.32	83.50	52.50	39.64

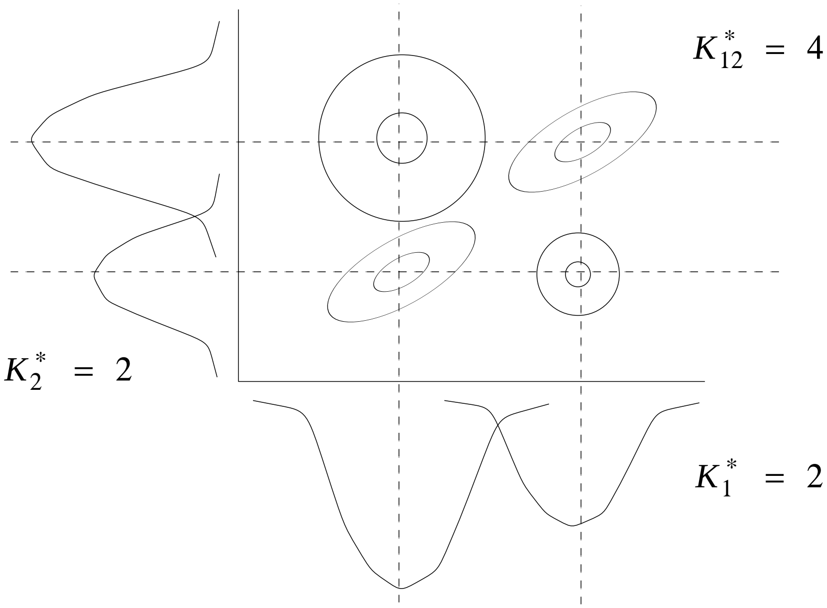
Table 4 The Best Trigram Performance of the Minimum Error Rate Classifier Using a 3-Mixture Multivariate Normal Density Function (K=3).

Comparison: Joint Consideration of the Features

N		Training Set					Testing Set				
	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	62.60	50.81
	IN: Dpos+H+MI	77.39	54.61	7.61	66.00	64.03	92.47	60.54	10.24	76.51	73.17
	Mx: Dpos+H (K=2)	97.87	71.50	3.73	84.69	82.63	95.78	71.30	7.34	83.54	81.75
3	IN: Dpos+MI	100.0	14.29	4.24	57.15	25.01	100.0	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 Best Number of Mixtures (K*)



Number of Mixtures increases rapidly with feature dimension

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	71.50	72.12	67.05	32.12	69.06	71.30	70.40	65.92	44.39
	Precision	100.0	97.87	90.74	83.70	56.78	100.0	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	81.70	81.75	80.51	77.37	53.80
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	51.90	24.49	75.86	74.14	74.14	36.21	44.83
	Precision	100.0	95.12	90.91	80.91	33.60	100.0	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	11.90
	WPR(1:1)	81.63	81.67	78.98	66.40	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	83.50	52.50	46.43

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

Precision and Recall Optimization Problem

⇒ 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 = Wp*(1-p)+Wr*(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 $-\nabla R$ direction when n-grams in the corpus are misclassified (∇: gradient w.r.t. the classifier parameters).

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

⇒ The risk will be non-increasing on average. ($\delta\overline{R} \le 0$)

⇒ 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)

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

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

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

⇒ and result in

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

⇒ 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 update the parameters in batch, not by sample.

⇒ Learning Constants for WPR maximization:

$$\begin{aligned} k_{21} &= \frac{w_p n_{11}}{\left(n_{11}+n_{21}\right)^2} \\ k_{12} &= \frac{w_p n_{21}}{\left(n_{11}+n_{21}\right)^2} + \frac{w_r}{n_1} \end{aligned}$$

⇒ Learning Constants for F-metric maximization:

$$\begin{aligned} k_{21} &\equiv \frac{\alpha_{21}}{\left(n_1 - n_{12}\right)} &= \frac{1}{\beta^2+1}\frac{1}{\left(n_1 - n_{12}\right)} \\ k_{12} &\equiv \frac{\left(\alpha_{12} n_1 + \alpha_{21} n_{21}\right)}{\left(n_1 - n_{12}\right)^2} &= \frac{1}{\beta^2+1}\frac{\left(\beta^2 n_1 + n_{21}\right)}{\left(n_1 - n_{12}\right)^2} \end{aligned}$$

Learning Parameters for Maximal Precision-Recall - Bigram Example

N		Training Set					Testing Set				
	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	62.60	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	71.50	3.73	84.69	82.63	95.78	71.30	7.34	83.54	81.75
	Mx+LRN: WPR (1:1)	99.57	72.75	3.42	86.16	84.07	100.0	71.75	6.52	85.87	83.55
	Mx+LRN: FM	99.43	72.85	3.42	86.14	84.09	100.0	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)

N		Training Set					Testing Set				
	Model	P	R	E	WPR	FM (β)	P	R	E	WPR	FM
2	Mx:Dpos+H(Kmax=3) before learning	97.87	71.50	3.73	78.10 (1:3) 84.69 (1:1) 91.28 (3:1)	91.15 (0.5) 82.63 (1.0) 75.58 (2.0)	95.78	71.30	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	82.90	97.58	72.20	6.83	78.54	82.99
	Mx+LRN: WPR (1:1)	99.57	72.75	3.42	86.16	84.07	100.0	71.75	6.52	85.87	83.55
	Mx+LRN: WPR (3:1)	99.71	72.02	3.50	92.79	83.63	100.0	71.30	6.62	92.83	83.25
	Mx+LRN: FM(0.5)	99.57	72.75	3.42	86.16	92.73	100.0	71.75	6.52	85.87	92.70
	Mx+LRN: FM(1.0)	99.43	72.85	3.42	86.14	84.09	100.0	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

A System for Chinese New Lexicon Acquisition

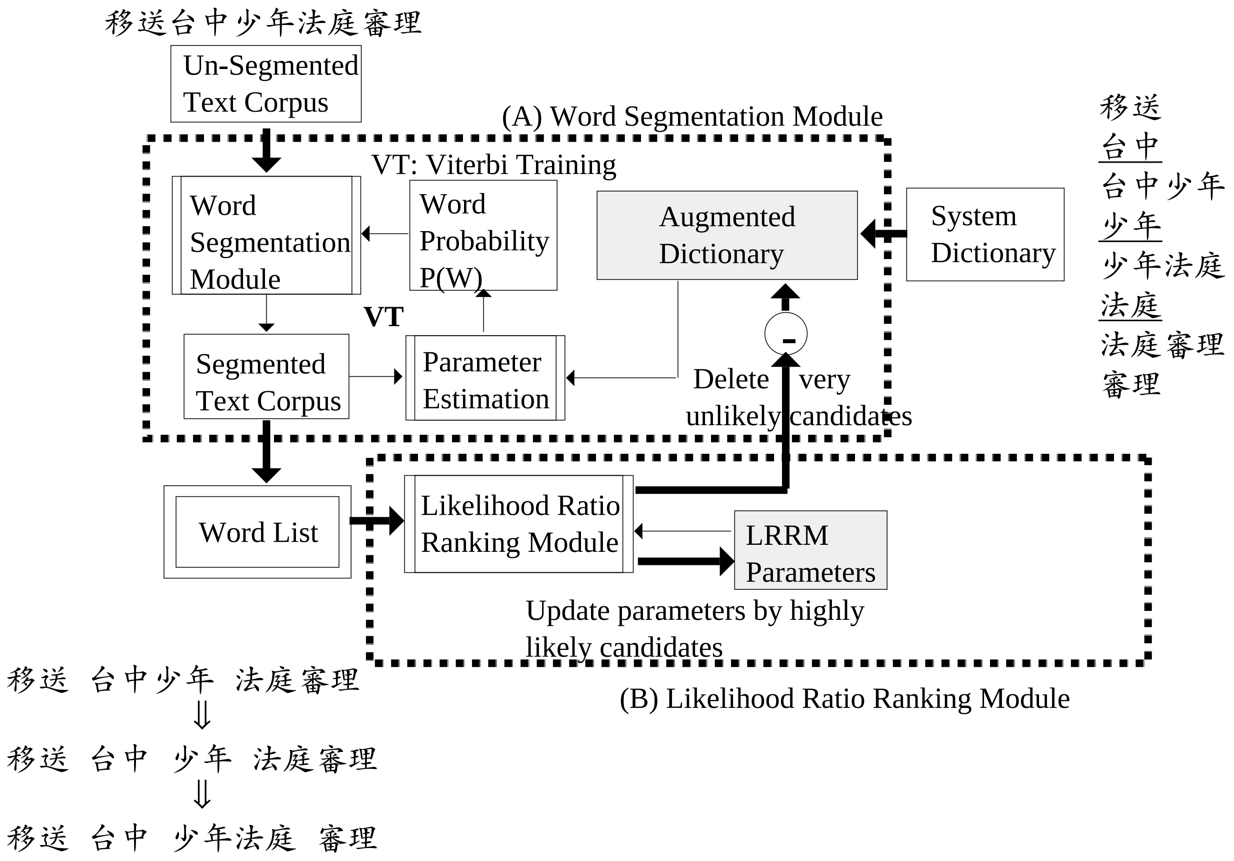
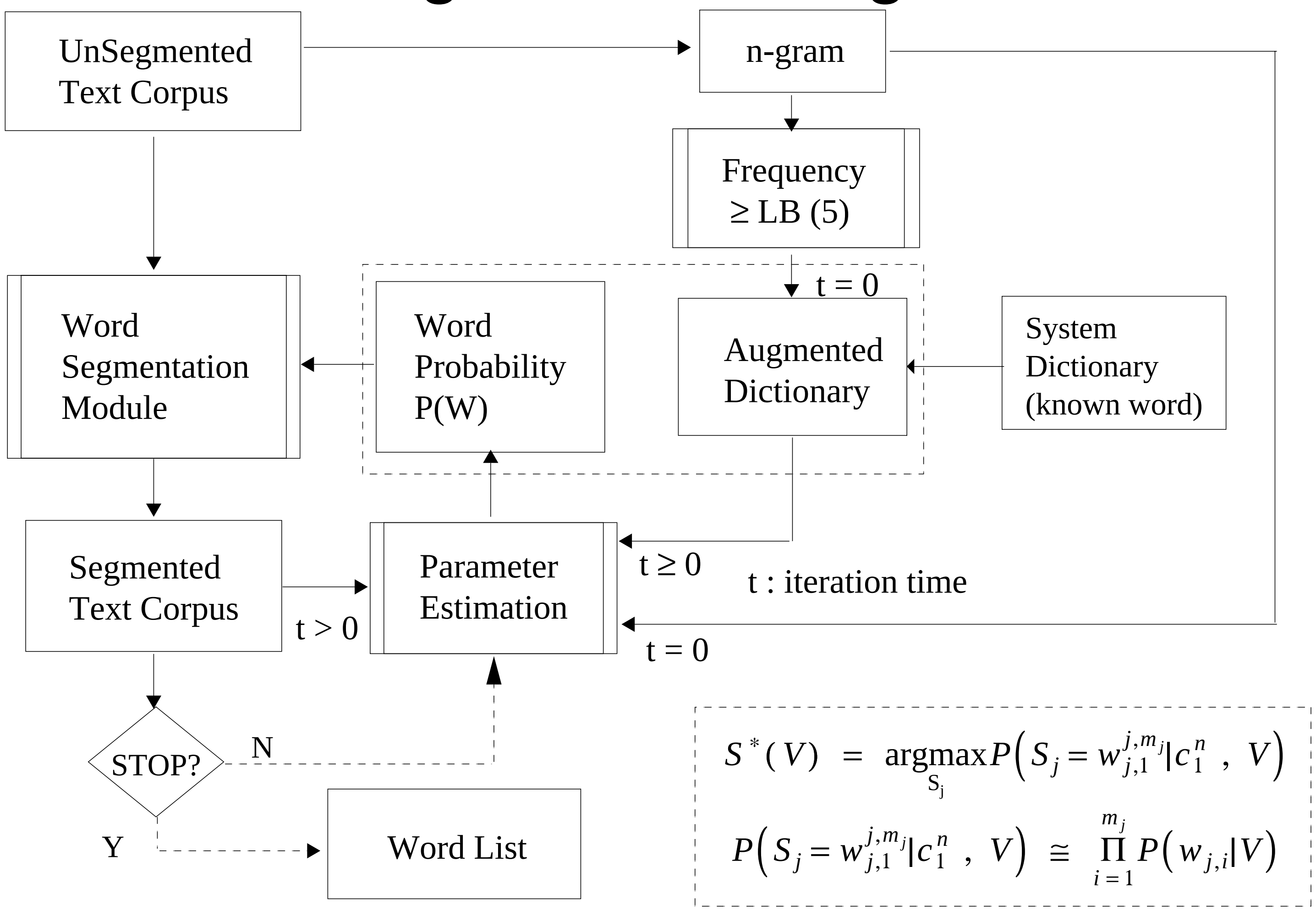


Figure 4 Configuration for Automatic Chinese New Lexicon Acquisition

Viterbi Training for Extracting New Words



$$S^*(V) = \underset{S_j}{\operatorname{argmax}} P\left(S_j = w_{j,1}^{j,m_j} | c_1^n, V\right)$$

$$P\left(S_j = w_{j,1}^{j,m_j} | c_1^n, V\right) \cong \prod_{i=1}^{m_j} P\left(w_{j,i} | V\right)$$

Figure 5 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^*

$$S^*(V) \;=\; \operatorname*{argmax}_{\mathrm{S_j}} P\left(S_j = w_{j,1}^{j,m_j} | c_1^n \;,\; V\right)$$

which maximizes the following likelihood function of the input corpus

$$P\left(S_j = w_{j,1}^{j,m_j} | c_1^n \;,\; V\right) \;\cong\; \prod_{i=1}^{m_j} P\left(w_{j,i} | V\right)$$

c_1^n : input characters $c_1, c_2, \ldots, c_n$

$S_j \;=\; w_{j,1}^{j,m_j}$: *j*-th segmentation pattern, consisting of $w_{j,1}, w_{j,2}, \ldots, w_{j,m_j}$

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

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

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

Initial Estimation:

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

Reestimation:

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

Performance of the Viterbi Training Procedure

n-gram	iteration number	p (%)	r (%)	WPR	FM
2	1	65.83	72.75	69.29	69.12
	2	68.67	76.67	72.67	72.45
	*	68.72	78.41	73.57	73.25
3	1	26.45	78.64	52.55	39.59
	2	28.81	80.68	54.75	42.46
	*	29.63	81.36	55.50	43.44
4	1	36.57	93.09	64.83	52.51
	2	38.24	93.45	65.85	54.27
	*	38.96	93.09	66.03	54.93

Table 9 Performance of the Viterbi Training Procedure for Identifying Unregistered Words
(*: performance at converge; convergence is reached at iteration #13.)

p: precision, r: recall, WPR=(p+r)/2: FM=2pr/(p+r)

Filter: 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.

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]:

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

Performance of the Classifier (Unsupervised Mode)

- Use system dictionary to assign initial class labels to the n-grams in the input corpus, and estimate initial parameters for the two classes
- Classify the input n-grams with current classifier parameters, and re-estimate classifier parameters after classification, and repeat.

n-gram	p (%)	r (%)	WPR	FM
2	54.28	90.99	72.63	68.00
3	33.78	63.01	48.40	43.98
4	51.17	81.42	66.30	62.84

Table 10 Performance of the two-class classification module for Identifying Unregistered Words (with $\log \lambda = 0$ as the decision boundary, 21 iterations)

- Performance is comparable with Viterbi Training but the curves are less monotonic because such updating strategy does not guarantee monotonic increasing in p, r, WPR or FM.

Simple Cascade Model (non-iterative)

n-gram	Models	p (%)	r (%)	WPR	FM
2	WS-only	68.72	78.41	73.57	73.25
	LRRM-only	54.28	90.99	72.63	68.00
	Non-Iterative	73.56	73.90	73.73	73.73
3	WS-only	29.63	81.36	55.50	43.44
	LRRM-only	33.78	63.01	48.40	43.98
	Non-Iterative	31.90	80.34	56.12	45.66
4	WS-only	38.96	93.09	66.03	54.93
	LRRM-only	51.17	81.42	66.30	62.84
	Non-Iterative	42.38	93.09	67.74	58.25

Table 11 Performance by cascading the segmentation module and the ranking module, in comparison with other models (WS-only: segmentation only; LRRM-only: likelihood ratio ranking module only; Non-Iterative: cascading the two modules and truncating the most unlikely 10% from the segmentation output). (Note: WS model applies 13 iterations to converge, and LRRM model is iterated by 21 iterations.)

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:

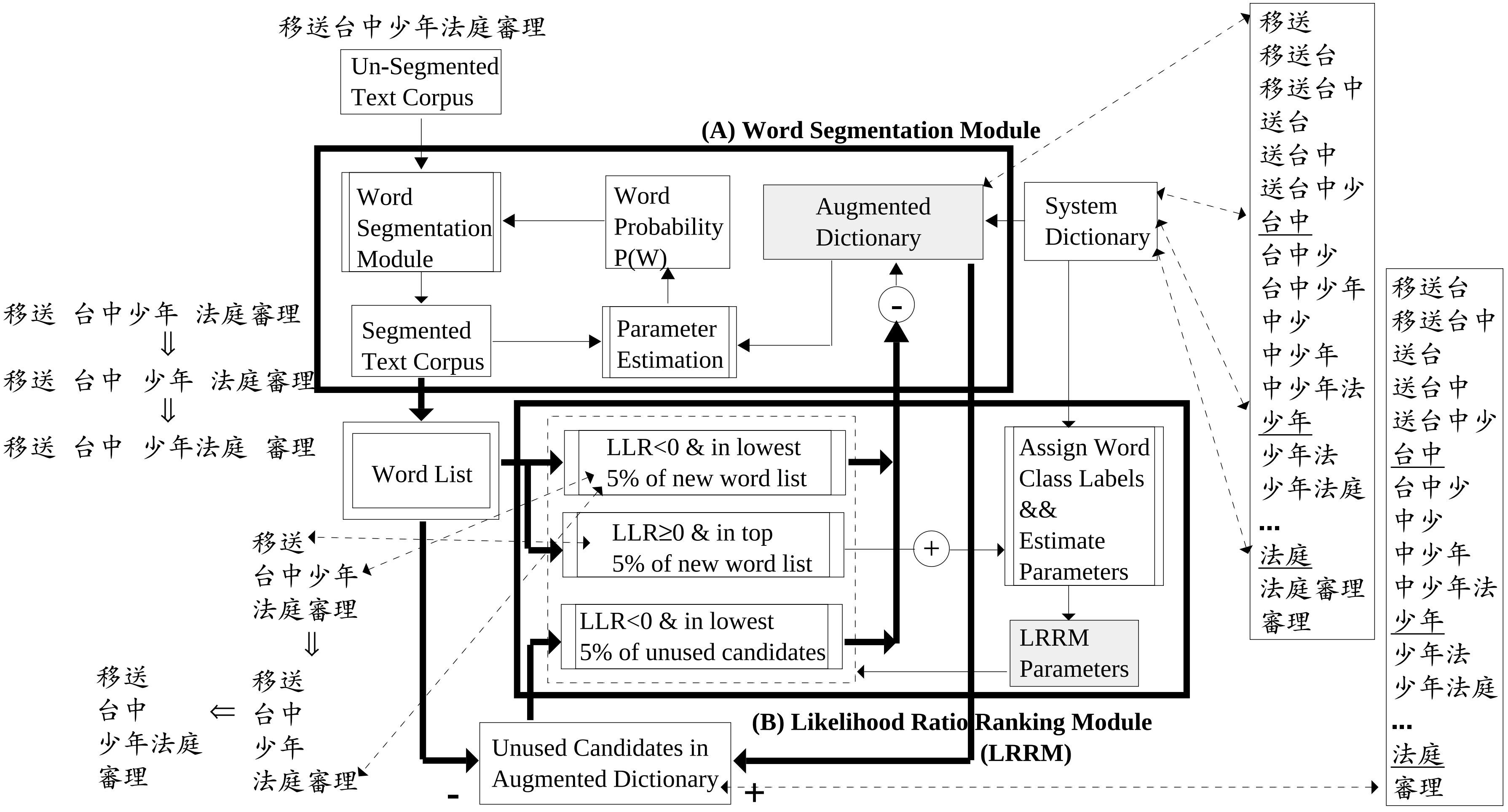
· Segment input with the augmented dictionary using Viterbi Training

· 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



移送台中少年法庭審理

Un-Segmented Text Corpus

(A) Word Segmentation Module

Word Segmentation Module

Word Probability P(W)

Augmented Dictionary

System Dictionary

Segmented Text Corpus

Parameter Estimation

-

移送 台中少年 法庭審理
⇓
移送 台中 少年 法庭審理
⇓
移送 台中 少年法庭 審理

Word List

LLR<0 & in lowest 5% of new word list

LLR≥0 & in top 5% of new word list

LLR<0 & in lowest 5% of unused candidates

+

Assign Word Class Labels && Estimate Parameters

LRRM Parameters

(B) Likelihood Ratio Ranking Module (LRRM)

移送
台中少年
法庭審理
⇓
移送
台中
少年
法庭審理

移送
台中
少年法庭
審理
⇐

Unused Candidates in Augmented Dictionary

-

+

移送
移送台
移送台中
送台
送台中
送台中少
台中
台中少
台中少年
中少
中少年
中少年法
少年
少年法
少年法庭
...
法庭
法庭審理
審理

移送台
移送台中
送台
送台中
送台中少
台中
台中少
中少
中少年
中少年法
少年
少年法
少年法庭
...
法庭
審理

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

n-gram	iteration number	p (%)	r (%)	WPR	FM
2	1	68.67	76.67	72.67	72.45
	21	72.39	80.83	76.61	76.38
	Difference	3.72	4.16	3.94	3.93
3	1	28.81	80.68	54.75	42.46
	21	38.60	87.80	63.20	53.62
	Difference	9.79	7.12	8.45	11.16
4	1	38.24	93.45	65.85	54.28
	21	56.21	93.82	75.01	70.30
	Difference	17.97	0.37	9.17	16.03

Table 12 Performance of the Integrated Viterbi-TCC System for Identifying Unregistered New Words (‘Difference’: difference in performance between iteration 21 and iteration 1)

Performance of the Integrated System (cont.)

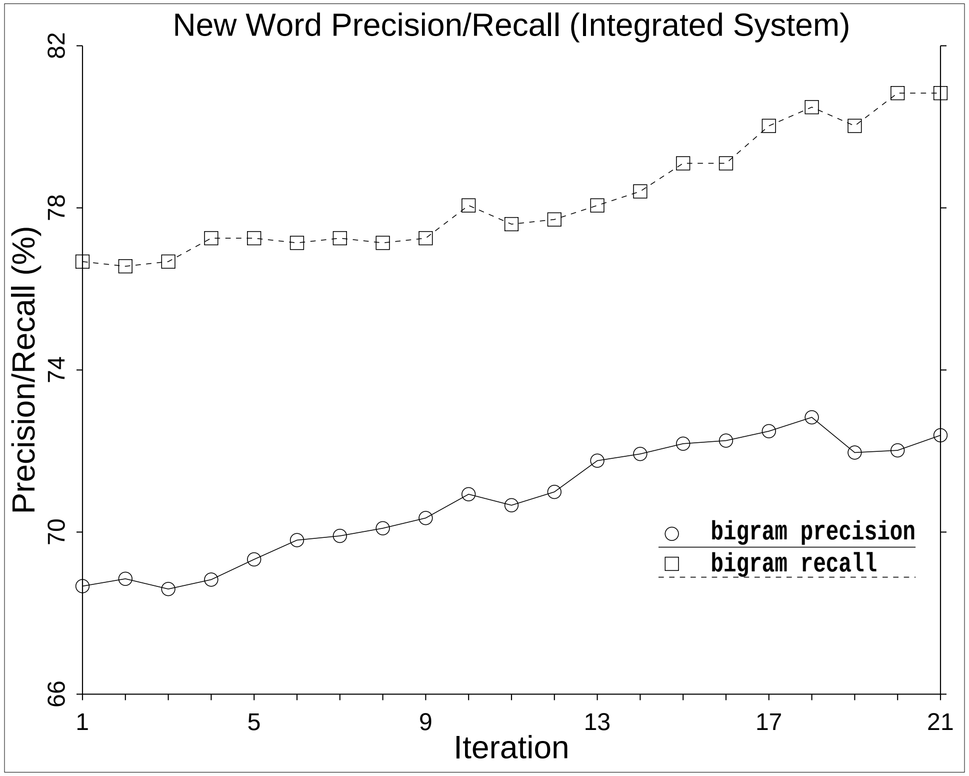


Figure 6 Performance for Identifying New Words in Each Iteration (bigram new words).

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

Comparison of the Various Models

n-gram	Models	p (%)	r (%)	WPR	FM
2	WS-only	68.72	78.41	73.57	73.25
	LRRM-only	54.28	90.99	72.63	68.00
	Non-Iterative	73.56	73.90	73.73	73.73
	Iterative	72.39	80.83	76.61	76.38
3	WS-only	29.63	81.36	55.50	43.44
	LRRM-only	33.78	63.01	48.40	43.98
	Non-Iterative	31.90	80.34	56.12	45.66
	Iterative	38.60	87.80	63.20	53.62
4	WS-only	38.96	93.09	66.03	54.93
	LRRM-only	51.17	81.42	66.30	62.84
	Non-Iterative	42.38	93.09	67.74	58.25
	Iterative	56.21	93.82	75.01	70.30

Table 13 Comparison of performance between various models for new word extraction.
(WS-only: Word-segmentation only, LRRM-only: ranking module only, Non-Iterative: cascading the WS and LRRM and truncating worst 10% of the segmentation output; Iterative: Iteratively integrating WS and LRRM modules.)

Summary on Quantitative Analysis

$$
\begin{aligned}
p &= \frac{n_{ww}}{n_{ww} + n_{xw}} = \frac{1}{1 + n_{xw}/n_{ww}} \\
r &= \frac{n_{ww}}{n_{ww} + n_{wx}} = \frac{1}{1 + n_{wx}/n_{ww}}.
\end{aligned}
$$

➨ 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:

=> Nww increased => Nwx decreased => recall increased

Example of Extracted New Words

Bigram New Words		**Trigram New Words**		**Quadgram New Words**	
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.)

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

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

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-Words)

Concluding Remarks

1. Various association metrics can be used jointly to rank word candidates by using a two-class classification model, which could minimize the classification error.

2. Joint Precision-Recall performance can be maximized by adjusting the classifier parameters to reduce a risk function defined on precision and recall.

3. An iterative scheme for precision-recall maximization can be used to integrate the segmentor and filter information by truncating unlikely candidates in the augmented dictionary and updating the filter/classifier parameters.

4. 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*.