Automatic Construction of a Chinese Electronic Dictionary

*Jing-Shin Chang, *Yi-Chung Lin and **Keh-Yih Su

*Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, Taiwan 30043, ROC

**Behavior Design Corporation
Science-Based Industrial Park
Hsinchu, Taiwan 30077, ROC

1995/06/30, WVLC3, ACL-95

Table of Contents

Why Automatic Electronic Dictionary (ED) Construction for Chinese

1. A large-scale dictionary is important to many NLP applications
2. New lexical items grow rapidly with technologies
3. Field specific annotations may not be available in a general dictionary
4. Human construction is costly and time consuming
5. Chinese electronic dictionary is not widely available

Task: Automatic Dictionary Construction

Input: a Large Untagged Chinese Text Corpus

+ a Small Tagged/Segmented Seed Corpus

Output: Lexicon Entries with POS tags

Goal: cheap & quick acquisition of a large scale ED (word-tag list)

Learning: (essentially) unsupervised

=> Supervised training requires a large annotated corpus

=> Supervised learning usually assumes the existence of a large ED

Example:

Input: (unsegmented/untagged text corpus)

☞ 分析家對馬來西亞及泰國的預測有頗大的差異·如果將其中的
差異列入考慮，馬來西亞企業的成長率由一四％大跌到五％·
泰國亦由一二％滑落到三％·

☞ (English Translation)

The analysts’ predictions to Malaysia and Thailand bear great differences. If the differences are taken into consideration, the growth rate of the Malaysian enterprises will be dropped greatly from 14% to 5%, and that of Thailand will also be dropped from 12% to 3%.

☞ (Segmented Chinese)

分析家 對 馬來西亞 及 泰國 的 預測 有 頗大 的 差異·
如果 將 其中 的 差異 列入 考慮， 馬來西亞 企業 的 成長率
由 一四% 大 跌 到 五%· 泰國 亦 由 一二% 滑落 到 三%·

Output:

分析家	n
對	p
馬來西亞	n
及	cjw
泰國	n
的	ctm, ctn
預測	n, vi, vn, vs

Bad Output:

分析	n, vi, vn, vp
家	n
對	...
馬來	...
西亞	...

What's Special with Chinese Text

- Chinese text streams are not delimited by blanks
    - need to identify the lexical units within the streams

Example:

☞ 分析家對馬來西亞的預測... <=> 分析家 對 馬來西亞 的 預測...

English Analogy:

☞ Thisisabook. <=> This is a book.

- Chinese text streams contain various constructs (not purely "words") from morphological level to sentence level
    - 人 (people/person) 們 ("+s") 總是 (always) 喜新厭舊 (like new stuffs better than the old ones) 的 (adj. marker)
    - People always like new stuffs better than the old ones.
- need "word" association as well as other association metrics

System Overview

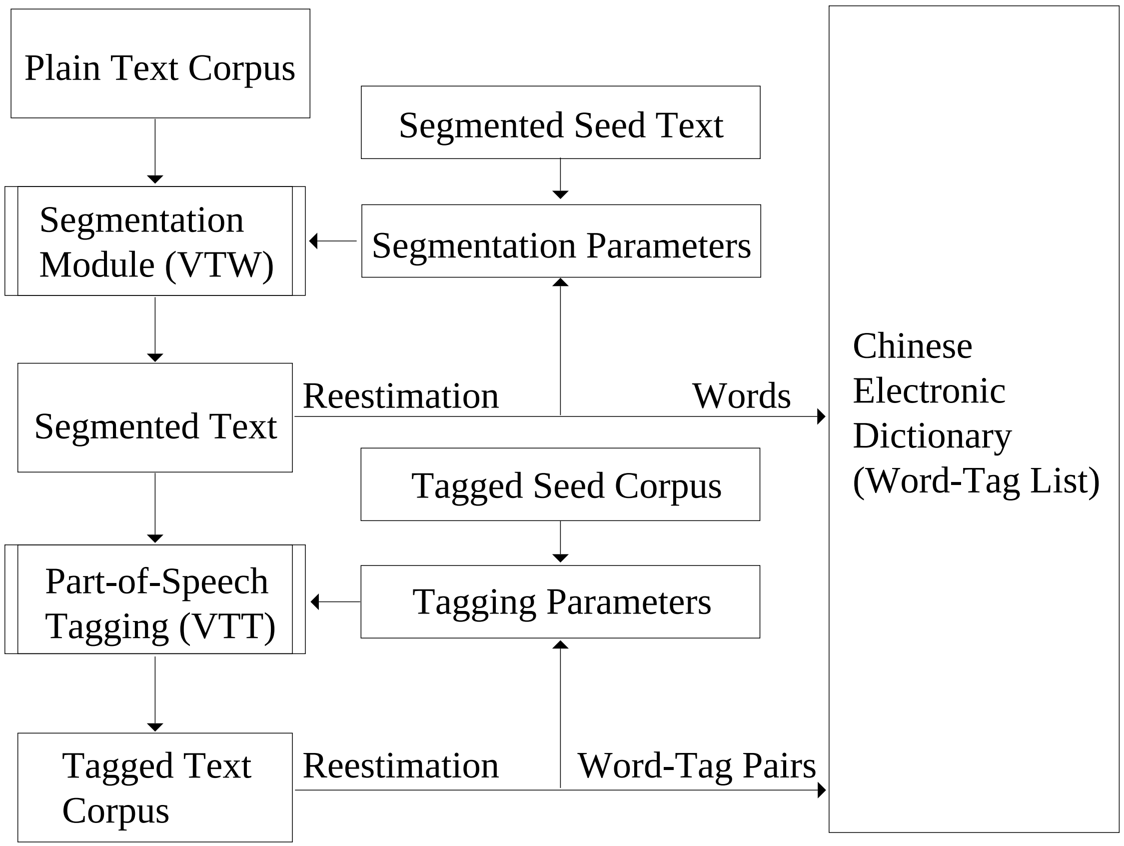

Figure 1 A Chinese Dictionary Construction Baseline System

Three ways to identify the lexical units

- word segmentation approaches:
  - produce the best segmentation pattern
    to optimize a segmentation criteria
  - identify qualified words based on contextual constraints
- feature-based 2-class classifier approaches:
  - ✔ classify the tokens with a discrimination function, or
  - test the likelihood between the word & non-word classes
  - identify qualified word tokens based on useful features
- a combination of the above two ways
  - apply discrimination function on words which are qualified
    by contextual constraints

Viterbi Training for Identifying Words (VTW)

- Reestimate the language parameters iteratively to improve the system performance.

- Labelling stage: Find the best word sequence $\widehat{W}$

$$\widehat{W} = \underset{W_j}{\operatorname{argmax}} P\left(W_j = w_{j,1}^{j,m_j} | c_1^n\right)$$

which maximizes the following score

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

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

$W_j = w_{j,1}^{j,m_j}$: segmented word pattern $w_{j,1}, w_{j,2}, \ldots, w_{j,m_j}$

Example Input: 分析家對馬來西亞及泰國的預測...

Possible patterns:

✓ W_1: 分析家 對 馬來西亞 及 泰國 的 預測 ...

✗ W_2: 分析 家 對馬 來 西亞 及 泰國 的 預測 ...

- Re-estimation stage: Find the best set of parameters $P(w_{j,i})$ which maximizes the likelihood of the segmented patterns $\hat{W}$ with the maximum likelihood estimation (MLE).

Example:

P(分析家) = N(分析家)/N(words) (is likely to increase)

P(分析) = N(分析)/N(words) (is likely to decrease)

Viterbi Training Procedure for Optimization

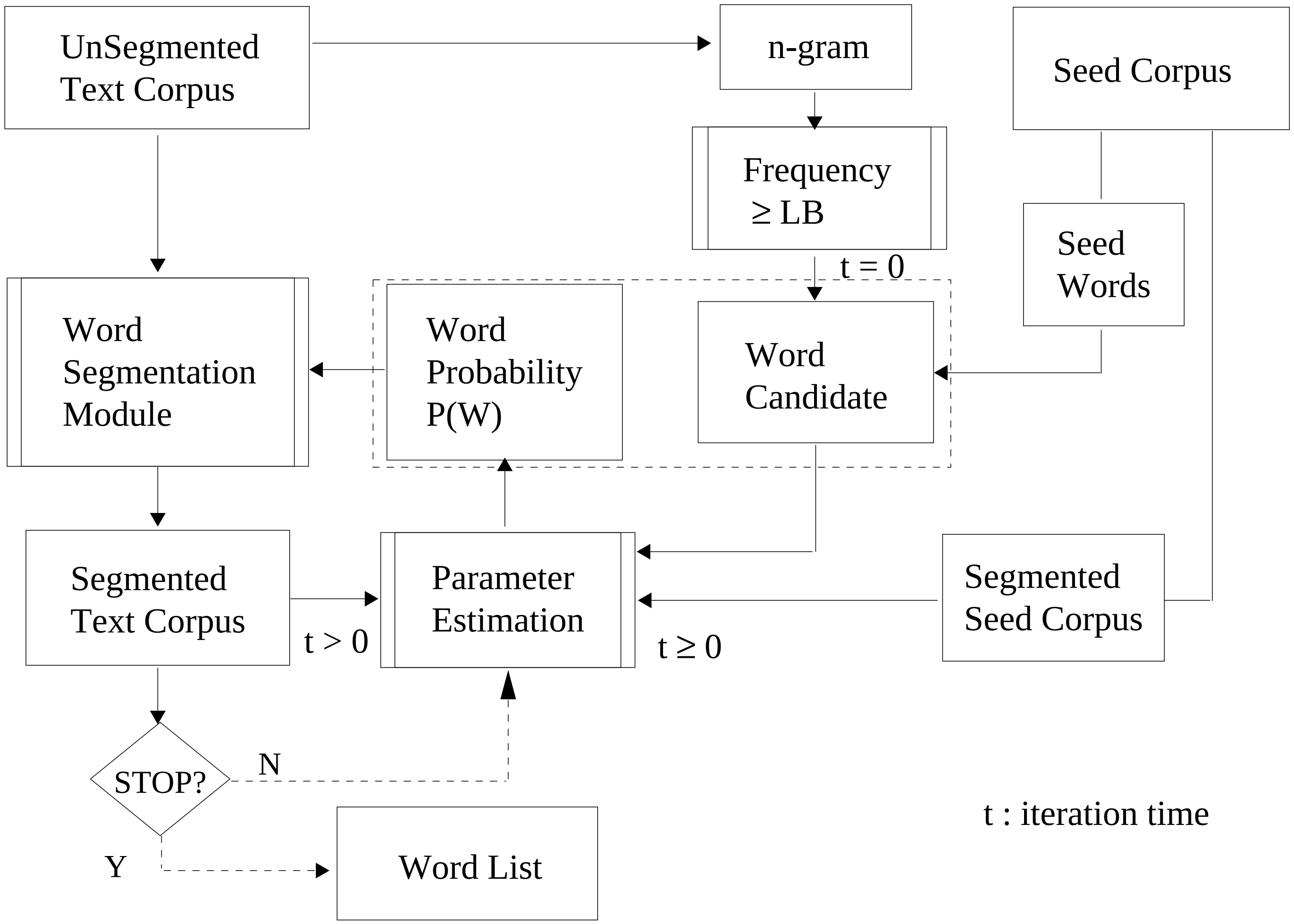

Figure 2 The block diagram of a Viterbi training model for word identification

Optimization Procedure

0. Collect n-grams (n=1,2,3,4) with Freq >= 5 + seed words + single characters as word candidates (initial word list/dictionary).
   Estimate initial probabilities of the n-grams based on seed.
1. Expand possible segmentation patterns for input text based on the candidate list
2. Find the best path with the highest score
3. Reestimate word probabilities based on the best path and the seed
4. Goto step 1 or Stop if parameters converge or max. iteration is reached
5. Extract Word List from the Segmented Text Corpus if Stopped

Two-Class Classifier for Identifying Words

Input: n-grams in the unlabelled text corpus

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

Classifier: a simple linear discrimination function

$$g(\vec{x}_s, \vec{w}_s) = \vec{w}_s \bullet \vec{x}_s$$

Decision Rules:

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

Learning the weights:

- A probabilistic descent method to optimize precision and recall
- Risk: R = Wp*(1-p)+Wr*(1-r) [p: precision r: recall]
- Adjust the weights in the $-\nabla R$ direction when an n-gram in the seed corpus is misclassified.

Features for the Classifier

— Frequency $f(x_i)$: a character n-gram is likely to be a word if it appears more frequently than the average.

— Mutual Information: characters with high mutual information tend to have high association [Church 90]

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

$$I(x,y,z) = log \frac{P_D(x,y,z)}{P_I(x,y,z)} = log \frac{P(x,y,z)}{P_I(x,y,z)}$$

$$P_I = P(x)P(y)P(z) + P(x)P(y,z) + P(x,y)P(z)$$

— Entropy: random distribution of the neighboring characters implies a natural break at the n-gram boundary [Tung 94]:

$$H_L(x) = -\sum_{c_i} P_L(c_i;x) log P_L(c_i;x)$$

$$H_R(x) = -\sum_{c_i} P_R(x;c_i) log P_R(x;c_i)$$

— Feature Vector: [log(f), log(I), $(H_L+H_R)/2$, 1]

Viterbi Training for POS Tagging

Labelling: Find the best POS pattern which maximizes the lexical score

$$\begin{aligned} S_{lex} &= P(T_j|W) = P(t_1^n|w_1^n) \\ &= P(W|t_1^n) \times P(t_1^n)/P(W) \\ &\cong 1/P(W) \times \prod P(t_i|t_{i-1}) P(w_i|t_i) \end{aligned}$$

W : input words $w_1, w_2, \ldots, w_n$

T_j : output tag (POS) sequence $t_1, t_2, \ldots, t_n$

Reestimation: Find the best $P(t_i|t_{i-1})$ and $P(w_i|t_i)$ with MLE.

Reestimation Procedure for POS Tagging

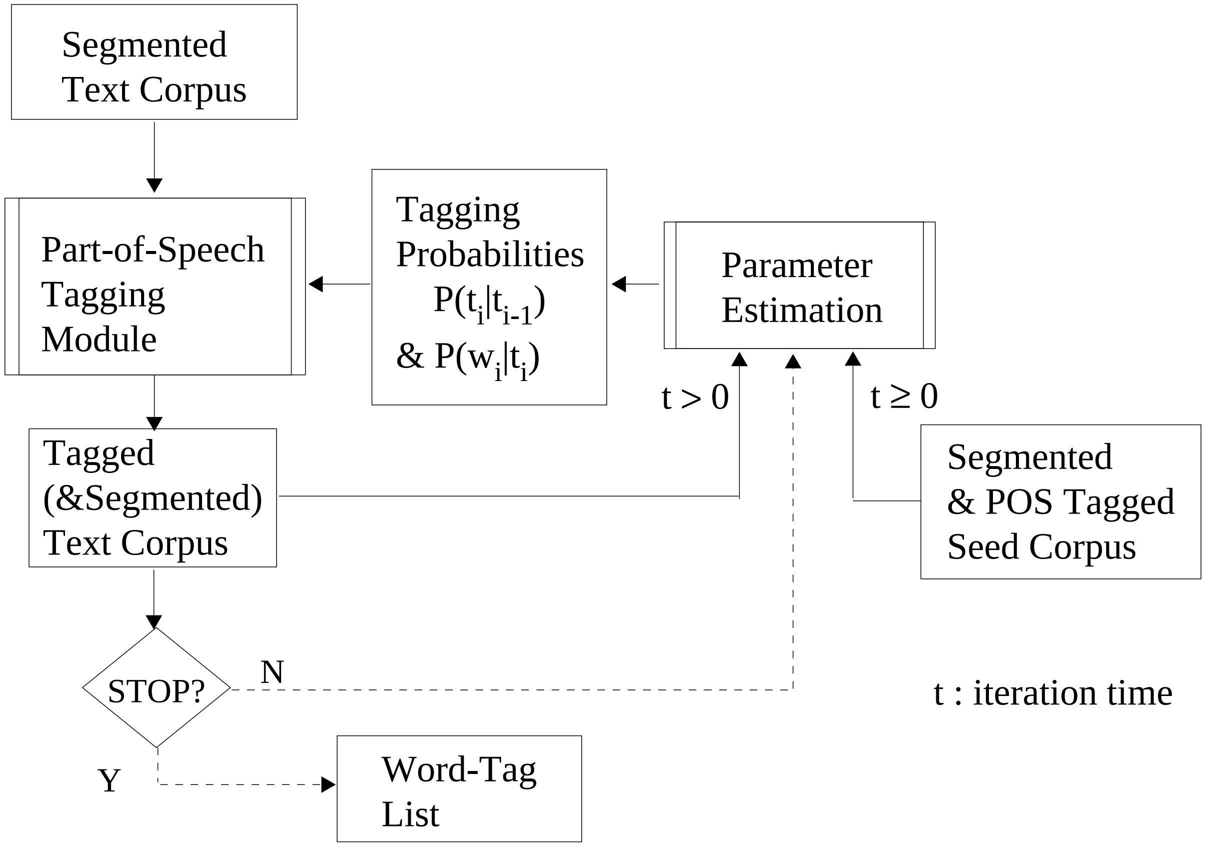

Figure 3 Block Diagram for a Viterbi POS Tag Training System

Reestimation Procedure for POS Tagging

0. Estimate initial POS tag transition probabilities and word-tag conditional probabilities according to the seed; and assign the most frequently used 10 POS in the seed to each segmented word tokens.
1. Expand all possible POS sequences according to the POS candiates.
2. Evaluate lexical scores for all POS sequences and find the best sequence.
3. Reestimate the probabilities according to the best sequence and the POS sequence for the seed.

Combining Viterbi Training and Two-Class Classifier for Word Identification

Basic Model: VTW (Viterbi Training for Words) + VTT (Viterbi Training for POS Tags)

- use contextual constraints for identifying words and assign parts of speech

Post-Filtering Model: VTW-I + TCC (Two-Class Classifier) + VTW-II + VTT

- filtering out unlikely candidate produced by the VTW-I module

Post-Filtering Configuration

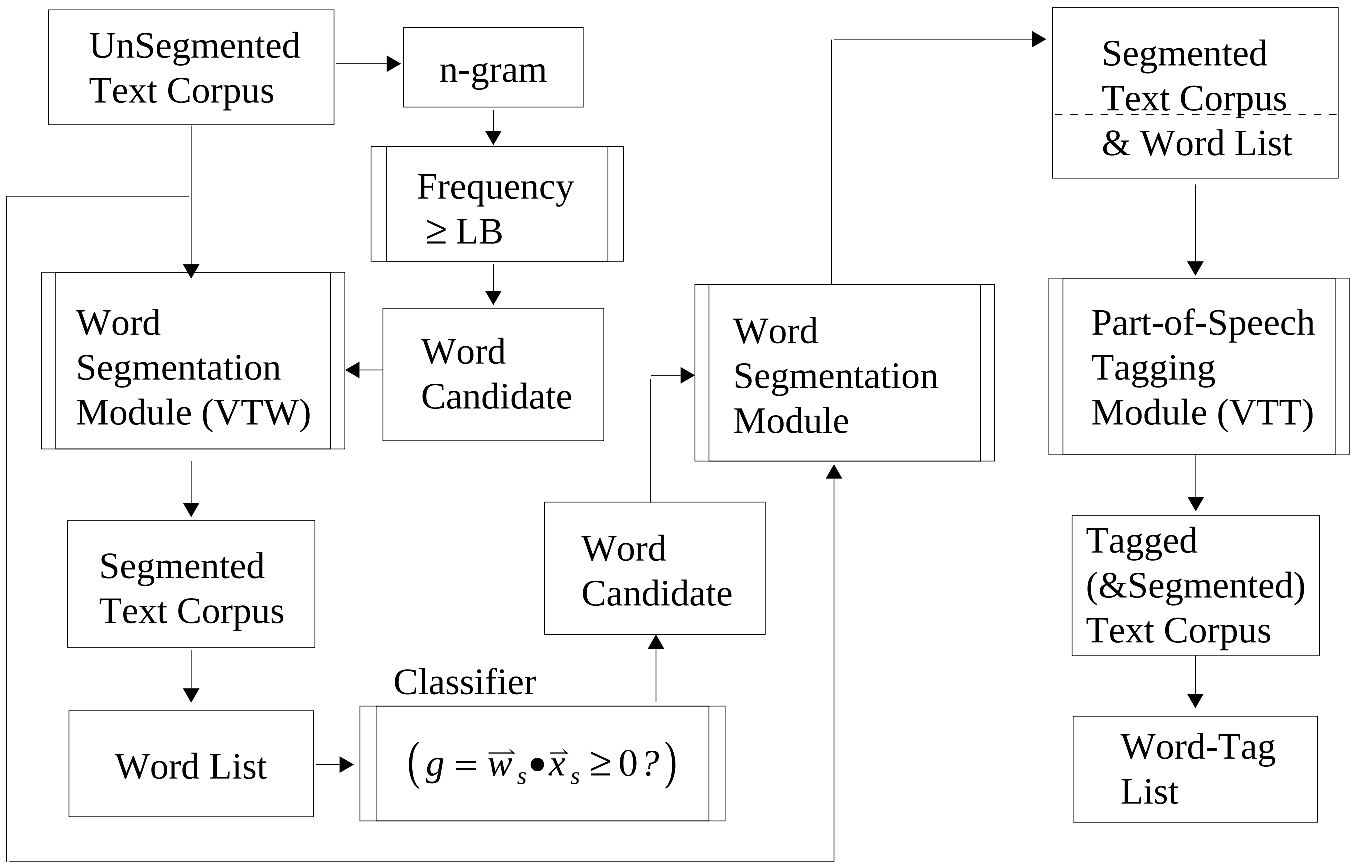

Figure 4 VTW + TCC + VTT Configuration for Automatic Construction of an Electronic Dictionary

Experiments

- Untagged Corpus: 311,591sentences (about 1,670,000 words, 9 M bytes)

  Domain: News Articles [China Times]

n	1	2	3	4	Total
#n-grams	3,994	99,407	99,211	43,424	246,036

- Seed-I : 9,676 sentences (127,052 words, 415 K)
- Seed-II:1,000 sentences (12,849 words) sampled from Seed-I

  Domain: Computer Manuals [BDC MT corpora]

n	1	2	3	4	Total
#n-grams	893	7,782	12,289	12,989	33,953

Evaluation against manually constructed Standard dictionaries

- Why not manual evaluation?
  - avoid subjective judgement & reduce costs in inspecting the long (80K) list
- Standard word dictionary: {BDC ED}∪{CKIP ED}∪{Seed words} excluding {{Words not in seed-II} ∪ {words not in the untagged input corpus} ∪ {words with low frequency (< 5)}}.

  Size: 17,005 bigram words, 2,524 trigram words and 1,612 4-gram words.
- Standard word-tag dictionary: {BDC English-Chinese E.D.}

  Size: 87,551 entries, including 35,722 bigram words, 19,858 trigram words, and 24,092 4-gram words

Tagset: 62 tags (only 42 tags appear in the 1000-sentence seed, 47 tags in the 9676-sentence seed)

Note: The standard dictionaries are constructed independent of the corpus used. In other words, they are not the ideal dictionaries constructed by an expert lexicographer after inspecting the whole untagged corpus.

Word Identification Performance

n-gram	Processing Step	Freq. LB. Filtering (LB=5)	VTW	Two-Class Classifier Pos-tFiltering (TCC)	VTW-2
2	Precision	17.07 (17,005/ 99,601)	38.21 (15,283/ 39,999)	56.80 (13,091/23,049)	56.88 (13,156/23,130)
	Recall	100.0 (17,005/ 17,005)	89.87 (15,283/ 17,005)	76.98 (13,091/17,005)	77.37 (13,156/17,005)
3	Precision	2.54 (2,524/99,460)	6.01 (2,171/36,123)	6.02 (2,171/36,067)	6.12 (2,170/35,443)
	Recall	100.0 (2,524/2,524)	86.01 (2,171/ 2,524)	86.01 (2,171/2,524)	85.97 (2,170/2,524)
4	Precision	3.71 (1,612/43,454)	5.81 (1,503/25,891)	6.21 (1,497/24,099)	6.31 (1,497/23,713)
	Recall	100.0 (1,612/1,612)	93.24 (1,503/1,612)	92.87 (1,497/1,612)	92.87 (1,497/1,612)

Table 1 Word Identification Performance for the VTW+VTT and VTW+TCC+VTT topologies (seed=1000 sentences)

Note: only 2.5% and 3.7% 3-grams and 4-grams in the initial candidate lists are considered "word" in the standard dictionary

Note: only 317 and 40 word tokens are available in the training (seed) corpus

n	2	3	4
#n-grams	7,782	12,289	12,989
#words	1,275	317	40
#non-words	6,507	11,972	12,949

Word Identification Performance Metric

Word Precision/Recall

$$P = \frac{\#(words\ in\ stddic\ \cap\ words\ in\ extdic)}{\#(words\ in\ extdic)}$$

$$R = \frac{\#(words\ in\ stddic\ \cap\ words\ in\ extdic)}{\#(words\ in\ stddic)}$$

stddic: standard word dictionary

extdic: extracted word dictionary

POS Tagging Performance

	Seed-I = 1000 Sentences		Seed-II = 9676 Sentences	
	Basic Model	Post-Filtering	Basic Model	Post-Filtering
Praw	46.40 (7944/17119)	46.37 (7241/15615)	51.79 (8873/17132)	52.53 (8390/15973)
Rraw	60.40 (7944/13153)	60.82 (7241/11906)	64.22 (8873/13816)	64.61 (8390/12986)
Pavrg	53.07	53.17	60.21	61.25
Ravrg	68.69	69.54	72.79	73.59
Pwavrg	57.20	57.55	71.16	71.81
Rwavrg	71.29	71.58	73.42	73.83

Table 2 Performance for Part-of-Speech Extraction of the Two Models

Note: only 42 tags appear in the 1000-sentence seed, 47 tags in the 9676-sentence seed

Note: only Top-10 POS are considered as the POS candidates

POS Tagging Performance Metrics

POS Raw Precision/Recall

$$P_{raw} = \frac{\#(word-tag\ pairs\ in\ stddic\ \cap\ word-tag\ pairs\ in\ extdic)}{\#(word-tag\ pairs\ in\ extdic)}$$

$$R_{raw} = \frac{\#(word-tag\ pairs\ in\ stddic\ \cap\ word-tag\ pairs\ in\ extdic)}{\#(word-tag\ pairs\ in\ stddic)}$$

POS Per-word Tag Precision

$$P(w) = \frac{\#(tags\ for\ w\ in\ stddic\ \cap\ tags\ for\ w\ in\ extdic)}{\#(tags\ for\ w\ in\ extdic)}$$

$$R(w) = \frac{\#(tags\ for\ w\ in\ stddic\ \cap\ tags\ for\ w\ in\ extdic)}{\#(tags\ for\ w\ in\ stddic)}$$

POS Average Precision/Recall

$$P_{avrg} = \frac{1}{N_w} \sum_w P(w) \qquad R_{avrg} = \frac{1}{N_w} \sum_w R(w)$$

POS Weighted Average Precision/Recall

$$P_{wavrg} = \sum_w P(w) \times Pr(w) \qquad R_{wavrg} = \sum_w R(w) \times Pr(w)$$

Error Types for Word Identification

0. Non-Error: New words not known to the standard dictionary and constructs with regular syntax (names, number+units, date/time)

互惠原則、內閣閣員、天佑證券、仁警分局、卜蜂集團、少量多樣、心路歷程、戶口謄本、毋枉毋縱、乙太網路、 分析師、分隊長、夫妻檔、月成長、水源區 (...)。

1. Compositional: the n-gram can be decomposed into legal words (e.g., 『今天』『下午』 ("this afternoon")、『今天』『發表』 ("announce ... today")、『介入』『選舉』("intervene the election")).

2. Collocational: parts of the n-gram are legal words, the other parts are highly flexible (e.g.,"do not + VERBS"：不予起訴、不予採納、不予理會；"many + NOUNS"：不少民眾、不少台商、不少住户；"is not + ADJECTIVES"：不公平、不切實、不友善).

3. Idiomatic: none of the substrings are legal words, all single characters are highly flexible (e.g., 不 一 而 足 ("cannot be enumerated one-by-one")).

Distribution of Errors

%	Non-Error				Error		
Types	new words	names	number +units	date/time	comp.	collo.	misc.
n = 2	18.0	10.5	1.0	0.0	70.5		
n = 3	6.0	7.5	6.5	2.0	0.0	10.5	67.5
n = 4	28.0	3.5	3.0	1.0	12.5	15.5	36.5

- sample size: 200 n-grams for each n (sampled with a random number generator)
- idiomatic type and some arguable n-grams are included in the "misc." (miscellaneous) type.

Concluding Remarks & Future Research

1. Unsupervised training based on contextual constraints and features for character associations could be used to extract the lexicon entries.
2. The proposed method could be used as a quick and cheap aid for constructing a large POS annotated dictionary.
3. The sub-patterns of the misidentified n-grams might provides some evidences for better identification.
4. 3-grams and 4-grams, which have very little training tokens, need be improved with better models
5. Better classifiers based on likelihood ratio test and other discriminative features are likely to minimize the error rates better than the simple linear discrimination function.
6. Semantic Tags could be extracted with a similar method.