Computational Linguistics and Chinese Language Processing
Vol. 9, No. 1 , February 2004, pp. 1-20 1

Bilingual Collocation Extraction Based on Syntactic and Statistical Analyses

Chien-Cheng Wu*, and Jason S. Chang*

Abstract

In this paper, we describe an algorithm that employs syntactic and statistical analysis to extract bilingual collocations from a parallel corpus. Collocations are pervasive in all types of writing and can be found in phrases, chunks, proper names, idioms, and terminology. Therefore, automatic extraction of monolingual and bilingual collocations is important for many applications, including natural language generation, word sense disambiguation, machine translation, lexicography, and cross language information retrieval.

Collocations can be classified as lexical or grammatical collocations. Lexical collocations exist between content words, while a grammatical collocation exists between a content word and function words or a syntactic structure. In addition, bilingual collocations can be rigid or flexible in both languages. Rigid collocation refers to words in a collocation must appear next to each other, or otherwise (flexible/elastic). We focus in this paper on extracting rigid lexical bilingual collocations. In our method, the preferred syntactic patterns are obtained from idioms and collocations in a machine-readable dictionary. Collocations matching the patterns are extracted from aligned sentences in a parallel corpus. We use a new alignment method based on punctuation statistics for sentence alignment. The punctuation-based approach is found to outperform the length-based approach with precision rates approaching 98%. The obtained collocations are subsequently matched up based on cross-linguistic statistical association. Statistical association between the whole collocations as well as words in collocations is used to link a collocation with its counterpart collocation in the other language. We implemented the proposed method on a very large Chinese-English parallel corpus and obtained satisfactory results.

* Department of Computer Science, National Tsing Hua University
Address: 101, Kuangfu Road, Hsinchu, Taiwan
E-mail: g904374@oz.nthu.edu.tw; jschang@cs.nthu.edu.tw

1. Introduction

Collocations, like terminology, tends to be lexicalized and to have a somewhat more restricted meaning than the surface forms suggest [Justeson and Katz, 1995]. Collocations are recurrent combinations of words that co-occur more often than they normally would based on chance. The words in a collocation may appear next to each other (rigid collocations) or in other locations (flexible/elastic collocations). On the other hand, collocations can also be classified as lexical or grammatical collocations [Benson, Benson, Ilson, 1986]. Lexical collocations exist between content words, while a grammatical collocation exists between a content word and function words or a syntactic structure. Collocations are pervasive in all types of writing and can be found in phrases, chunks, proper names, idioms, and terminology. Collocations in one language are usually difficult to translate directly into another language word for word; therefore, they present a challenge for machine translation systems and second language learners alike.

Automatic extraction of monolingual and bilingual collocations is important for many applications, including natural language generation, word sense disambiguation, machine translation, lexicography, and cross language information retrieval. Hank and Church [1990] pointed out the usefulness of mutual information for identifying monolingual collocations in lexicography. Justeson and Katz [1995] proposed to identify technical terminology based on preferred linguistic patterns and discourse properties of repetition. Among the many general methods presented by Manning and Schutze [1999], the best results can be achieved through filtering based on both linguistic and statistical constraints. Smadja [1993] presented a method called EXTRACT, based on the mean variance of the distance between two collocates , that is capable of computing elastic collocations. Kupiec [1993] proposed to extract bilingual noun phrases using statistical analysis of the co-occurrence of phrases. Smadja, McKeown, and Hatzivassiloglou [1996] extended the EXTRACT approach to handle bilingual collocation based mainly on the statistical measures of the Dice coefficient. Dunning [1993] pointed out the weakness of mutual information and showed that log likelihood ratios are more effective in identifying monolingual collocations, especially when the occurrence count is very low.

Both Smadja and Kupiec used the statistical association between whole collocations in two languages without examining the constituent words. For a collocation and its non-compositional translation equivalent, this approach is reasonable. For instance, with the bilingual collocation (“擠破頭”, “stop at nothing”) shown in Example 1, it will not be helpful to examine the statistical association between “stopping” and “擠” [ji, squeeze] (or “破” [bo, broken] and “頭” [tou, head] for that matter). However, for the bilingual collocation (“減薪”, “pay cut”) shown in Example 2, considering the statistical association between “pay” and “薪” [xin, wage] as well as between “cut” and “減” [jian, reduce] certainly makes sense. Moreover, we have more data with which to

make statistical inferences between words than between phrases. Therefore, measuring the statistical association of collocations based on constituent words will help us cope with the data sparseness problem. We will be able to extract bilingual collocations with high reliability even when they appear together in aligned sentences only once or twice.

Example 1

They are **stopping at nothing** to get their kids into "star schools"

他們**擠破頭**也要把孩子送進明星小學

Source: 1995/02 No Longer Just an Academic Question: Educational Alternatives Come to Taiwan

Example 2

Not only haven't there been layoffs or **pay cuts**, the year-end bonus and the performance review bonuses will go out as usual .

不但不虞裁員、**減薪**，年終獎金、考績獎金還都照發不誤

Source: 1991/01 Filling the Iron Rice Bowl

Since collocations can be rigid or flexible in both languages, there are, in general, three types of bilingual collocation matches. In Example 1, (“擠破頭”, “stop at nothing”) is a pair of rigid collocation, and (“把…送進”, “get … into”) is a pair of elastic collocation. In Example 3 ,(“走…的路線’, “take the path of”) is an example of a pair of elastic and rigid collocations.

Example 3

Lin Ku-fang, a worker in ethnomusicology, worries too, but his way is not to **take the path of** revolutionizing Chinese music or making it more "symphonic"; rather, he goes directly into the tradition, looking into it for "good music" that has lasted undiminished for a hundred generations.

民族音樂工作者林谷芳也非不感到憂心，但他的方法是：不**走**國樂改革或「交響化」**的路**，而是直接面對傳統、從中尋找歷百代不衰的「好聽音樂」。

Source: 1997/05 A Contemporary Connoisseur of the Classical Age--Lin Ku-fang's Canon of Chinese Classical Music

In this paper, we describe an algorithm that employs syntactic and statistical analyses to extract rigid lexical bilingual collocations from a parallel corpus. Here, we focus on bilingual collocations, which have some lexical correlation between them and are rigid in both languages. To cope with the data sparseness problem, we use the statistical association between two collocations as well as that between their constituent words. In Section 2, we describe how we obtain the preferred syntactic patterns from collocations and idioms in a machine-readable dictionary. Examples will be given to show how collocations matching the patterns are extracted and aligned for given aligned sentence pairs in a parallel corpus. We implemented the proposed method in an experiment on the Chinese-English parallel corpus of Sinorama Magazine and obtained satisfactory results. We describe the experiments and our evaluation in section 3. The limitations of the study and related issues are taken up in section 4. We conclude and give future directions of research in section 5.

2. Extraction of Bilingual Collocations

In this chapter, we will describe how we obtain bilingual collocations by using preferred syntactic patterns and associative information. Consider a pair of aligned sentences in a parallel corpus such as that shown in Example 4 below:

Example 4

The civil service rice bowl, about which people always said "you can't get filled up, but you won't starve to death either," is getting a new look with the economic downturn. Not only haven't there been layoffs or pay cuts, the year-end bonus and the performance review bonuses will go out as usual, drawing people to compete for their own "iron rice bowl."

以往一向被認爲「吃不飽、餓不死」的公家飯，值此經濟景氣低迷之際，不但不虞裁員、減薪，年終獎金、考績獎金還都照發不誤，因而促使不少人回頭競逐這隻「鐵飯碗」。

Source: 1991/01 Filling the Iron Rice Bowl

We can extract the following collocations and translation counterparts:

(civil service rice bowl, 公家飯)
(get filled up, 吃…飽)
(starve to death, 餓…死)
(economic downturn, 經濟景氣低迷)
(pay cuts, 減薪)

(year-end bonus, 年終獎金)
(performance review bonuses, 考績獎金)
(iron rice bowl, 鐵飯碗)

In section 2.1, we will first show how that process is carried out for Example 4 using the proposed approach. A formal description of our method will be given in section 2.2.

2.1 An Example of Extracting Bilingual Collocations

To extract bilingual collocations, we first run part of speech tagger on both sentences. For instance, for Example 4, we get the results of tagging shown in Examples 4A and 4B.

In the tagged English sentence, we identify phrases that follow a syntactic pattern from a set of training data of collocations. For instance, "jj nn" is one of the preferred syntactic structures. Thus, "civil service," "economic downturn," "own iron" etc are matched. See Table 1 for more details. For Example 4, the phrases shown in Examples 4C and 4D are considered to be potential candidates for collocations because they match at least two distinct collocations listed in LDOCE:

Example 4A

The/at civil/jj service/nn rice/nn bowl/nn ,/, about/in which/wdt people/nns always/rb said/vbd "/`` you/ppss can/md 't/* get/vb filled/vbn up/rp ,/, but/cc you/ppss will/md 't/* starve/vb to/in death/nn either/cc ,/rb "/" is/bez getting/vbg a/at new/jj look/nn with/in the/at economic/jj downturn/nn ./. Not/nn only/rb have/hv 't/* there/rb been/ben layoffs/nns or/cc pay/vb cuts/nns ,/, the/at year/nn -/in end/nn bonus/nn and/cc the/at performance/nn review/nn bonuses/nn will/md go/vb out/rp as/ql usual/jj ,/, drawing/vbg people/nns to/to compete/vb for/in their/pp$ own/jj "/`` iron/nn rice/nn bowl/nn ./. "/"

Example 4B

以往/Nd 一向/Dd 被/P02 認爲/VE2 「/PU 吃/VC 不/Dc 飽/VH 、/PU 餓不死/VR 」/PU 的/D5 公家/Nc 飯/Na ，/PU 值此/Ne 經濟/Na 景氣/Na 低迷/VH 之際/NG ，/PU 不但/Cb 不虞/VK 裁員/VC 、/PU 減薪/VB ，/PU 年終獎金/Na 、/PU 考績/Na 獎金/Na 還都/Db 照/VC 發/VD 不誤/VH ，/PU 因而/Cb 促使/VL 不少/Ne 人/Na 回頭/VA 競逐/VC 這/Ne 隻/Nf 「/PU 鐵飯碗/Na 」/PU

Example 4C

"civil service," " rice bowl," " iron rice bow," " fill up," " economic downturn," " end bonus," " year - end bonus," " go out," " performance

review," " performance review bonus," " pay cut," " starve to death," " civil service rice," " service rice," " service rice bowl," " people always," " get fill," " people to compete," " layoff or pay," " new look," " draw people"

Example 4D

"吃不飽," "餓不死," "公家飯," "經濟景氣," "景氣低迷," "經濟景氣低迷," "裁員," "減薪," "年終獎金," "考績獎金," "競逐," " 鐵飯碗."

Although "new look" and "draw people" are legitimate phrases, they are more like "free combinations" than collocations. That is reflected by their low log likelihood ratio values. For this research, we proceed to determine how tightly the two words in overlapping bigrams within a collocation are associated with each other; we calculate the minimum of the log likelihood ratio values for all the bigrams. Then, we filter out the candidates whose POS patterns appear only once or have minimal log likelihood ratios of less than 7.88. See Tables 1 and 2 for more details.

In the tagged Chinese sentence, we basically proceed in the same way to identify the candidates of collocations, based on the preferred linguistic patterns of the Chinese translations of collocations in an English-Chinese MRD. However, since there is no space delimiter between words, it is at times difficult to say whether a translation is a multi-word collocation or a single word, in which case it should not be considered as a collocation. For this reason, we take multiword and singleton phrases (with two or more characters) into consideration. For instance, in tagged Example 4, we extract and consider these candidates shown in Tables 1 and 2 as the counterparts of English collocations.

Notes that at this point, we have not pinned collocations down but allow overlapping and conflicting candidates such as "經濟景氣," "景氣低迷," and "經濟景氣低迷." See Tables 3 and 4 for more details.

Table 1. The initial candidates extracted based on preferred patterns trained on collocations listed in LDOCE (LDOCE example: the example for the POS pattern in LDOCE; Pattern Count: the number of POS patterns occurring in LDOCE ; Min LLR : the minimal LLR value of every two words in the candidate pairs.)

E-collocation Candidate Pairs	Part of Speech	LDOCE example	Pattern Count	Min LLR
civil service	jj nn	hard cash	1562	496.156856
rice bowl	nn nn	beef steak	1860	99.2231161

iron rice bowl	nn nn nn	tin pan alley	8	66.3654678
filled up	vbn rp	set down	84	55.2837871
economic downturn	jj nn	hard cash	1562	51.8600979
*end bonus	nn nn	beef steak	1860	15.9977283
year - end bonus	nn nn nn	tin pan alley	12	15.9977283
go out	vb rp	bang out	1790	14.6464925
performance review	nn nn	beef steak	1860	13.5716459
performance review bonus	nn nn nn	tin pan alley	8	13.5716459
pay cut	vb nn	take action	313	8.53341082
starve to death	vb to nn	bring to bay	26	7.93262494
civil service rice	jj nn nn	high water mark	19	7.88517791
*service rice	nn nn	beef steak	1860	7.88517791
*service rice bowl	nn nn nn	tin pan alley	8	7.88517791
* people always	nn rb	hand back	24	3.68739176
get filled	vb vbn	stay put	3	1.97585732
* people to compete	nn to vb	order to view	2	1.29927068
* layoff or pay	nn cc vb	wine and dine	14	0.93399125
* new look	jj nn	hard cash	1562	0.63715518
* draw people	vbg nn	dying wish	377	0.03947748

* indicates invalid candidate (with human judgment)

Table 2. The candidates of English collocations based on both preferred linguistic patterns and log likelihood ratios.

E-collocation Candidate Pairs	Part of Speech	LDOCE example	Pattern Count	Min LLR
civil service	jj nn	hard cash	1562	496.156856
rice bowl	nn nn	beef steak	1860	99.2231161
iron rice bowl	nn nn nn	tin pan alley	8	66.3654678
filled up	vbn rp	set down	84	55.2837871
economic downturn	jj nn	hard cash	1562	51.8600979
*end bonus	nn nn	beef steak	1860	15.9977283
year - end bonus	nn nn nn	tin pan alley	12	15.9977283
go out	vb rp	bang out	1790	14.6464925
performance review	nn nn	beef steak	1860	13.5716459
performance review bonus	nn nn nn	tin pan alley	8	13.5716459
pay cut	vb nn	take action	313	8.53341082
starve to death	vb to nn	bring to bay	26	7.93262494
civil service rice	jj nn nn	high water mark	19	7.88517791
*service rice	nn nn	beef steak	1860	7.88517791

*service rice bowl	nn nn nn	tin pan alley	8	7.88517791

* indicates an invalid candidate(based on human judgment)

Table 3. The initial candidates extracted by the Chinese collocation recognizer.

C-collocation Candidate Pairs	POS	LDOCE example	Patter Count	Min LLR
不少 人	Ed Na	本國語	2	550.904793
*被 認爲	PP VE	待考慮	6	246.823964
景氣 低迷	Na VH	視力不良	97	79.8159904
經濟 景氣 低迷	Na Na VH	宗教信仰自由	3	47.2912274
經濟 景氣	Na Na	生活津貼	429	47.2912274
公家 飯	Nc Na	全國大選	63	42.6614685
*不 飽	Dc VH	毫無困難	24	37.3489687
考績 獎金	Na Na	生活津貼	429	36.8090448
不虞 裁員	VJ VA	引起爭吵	3	17.568518
回頭 競逐	VA VC	豎耳傾聽	26	14.7120606
*還都 照	Db VC	無法參與	18	14.1291893
*發 不誤	VD VH	供應充份	2	13.8418648
*低迷 之際	VH NG	兩可之間	10	11.9225789
*值此 經濟 景氣	VA Na Na	浮球活栓	2	9.01342071
*值此 經濟	VA Na	劃線支票	94	9.01342071
*照 發	VC VD	登記歸還	2	6.12848087
*人 回頭	Na VA	安危未卜	27	1.89617179

* indicates an invalid candidate (based on human judgment)

Table 4. The result of Chinese collocation candidates which are picked out. (The ones which have no Min LLR are singleton phrases.)

C-collocation Candidate Pairs	POS	LDOCE example	Patter Count	Min LLR
不少 人	Ed Na	本國語	2	550.904793
*被 認爲	PP VE	待考慮	6	246.823964
景氣 低迷	Na VH	視力不良	97	79.8159904
經濟 景氣 低迷	Na Na VH	宗教信仰自由	3	47.2912274
經濟 景氣	Na Na	生活津貼	429	47.2912274
公家 飯	Nc Na	全國大選	63	42.6614685
*不 飽	Dc VH	毫無困難	24	37.3489687
考績 獎金	Na Na	生活津貼	429	36.8090448
不虞 裁員	VJ VA	引起爭吵	3	17.568518

回頭 競逐	VA VC	豎耳傾聽	26	14.7120606
*還都 照	Db VC	無法參與	18	14.1291893
*發 不誤	VD VH	供應充份	2	13.8418648
*低迷 之際	VH NG	兩可之間	10	11.9225789
*值此 經濟 景氣	VA Na Na	浮球活栓	2	9.01342071
*值此 經濟	VA Na	劃線支票	94	9.01342071
之際	NG		5	
經濟	Na		1408	
景氣	Na		1408	
年終獎金	Na		1408	
考績	Na		1408	
獎金	Na		1408	
鐵飯碗	Na		1408	
公家	Nc		173	
以往	Nd		48	
值此	VA		529	
裁員	VA		529	
回頭	VA		529	
減薪	VB		78	
競逐	VC		1070	
認爲	VE		139	
低迷	VH		731	
不誤	VH		731	
不虞	VJ		205	
促使	VL		22	
餓不死	VR		14	

To align collocations in both languages, we employ the Competitive Linking Algorithm proposed by Melamed [1996] to conduct word alignment. Basically, the proposed algorithm **CLASS**, the Collocation Linking Algorithm based on Syntax and Statistics, is a greedy method that selects collocation pairs. The pair with the highest association value takes precedence over those with lower values. CLASS also imposes a one-to-one constraint on the collocation pairs selected. Therefore, the algorithm at each step considers only pairs with words that haven't been selected previously. However, CLASS differs with CLA(Competitive Linking Algorithm) in that it considers the association between the two candidate collocations based on two measures:

- the Logarithmic Likelihood Ratio between the two collocations in question as a whole;
- the translation probability of collocation based on constituent words.

In the case of Example 4, the CLASS algorithm first calculates the counts of collocation candidates in the English and Chinese parts of the corpus. The collocations are matched up randomly across from English to Chinese. Subsequently, the co-occurrence counts of these candidates matched across from English to Chinese are also tallied. From the monolingual collocation candidate counts and cross language concurrence counts, we produce the LLR values and the collocation translation probability derived from word alignment analysis. Those collocation pairs with zero translation probability are ignored. The lists are sorted in descending order of LLR values, and the pairs with low LLR value are discarded. Again, in the case of Example 4, the greedy selection process of collocation starts with the first entry in the sorted list and proceeds as follows:

1. The first, third, and fourth pairs, ("iron rice bowl," "鐵飯碗"), ("year-end bonus," "年終獎金"), and ("economic downturn," "經濟景氣低迷"), are selected first. Thus, conflicting pairs will be excluded from consideration, including the second pair, fifth pair and so on.
2. The second entry ("rice bowl," "鐵飯碗"), fifth entry ("economic downturn," "值此經濟景氣") and so on conflict with the second and third entries that have already been selected. Therefore, CLASS skips over these entries.
3. The entries ("performance review bonus," "考績獎金"), ("civil service rice," "公家飯"), ("pay cuts," "減薪"), and ("starve to death," "餓不死") are selected next.
4. CLASS proceeds through the rest of the list and the other list without finding any entries that do not conflict with the seven entries previously selected.
5. The program terminates and outputs a list of seven collocations.

Table 5. The extracted Chinese collocation candidates which are picked out. The shaded collocation pairs are selected by CLASS (Greedy Alignment Linking E).

English collocations	Chinese collocations	LLR	Collocation Translation Prob.
iron rice bowl	鐵飯碗	103.3	0.0202
rice bowl	鐵飯碗	77.74	0.0384
year-end bonus	年終獎金	59.21	0.0700
economic downturn	經濟 景氣 低迷	32.4	0.9359
economic downturn	值此 經濟 景氣	32.4	0.4359
...	. . .	...	...
performance review bonus	考績 獎金	30.32	0.1374
economic downturn	景氣 低迷	29.82	0.2500
civil service rice	公家 飯	29.08	0.0378

pay cuts	減薪	28.4	0.0585
year-end bonus	考績 獎金	27.35	0.2037
performance review	考績	27.32	0.0039
performance review bonus	年終獎金	26.31	0.0370
starve to death	餓不死	26.31	0.5670
...	. . .	...	...
rice bowl	公家 飯	24.98	0.0625
iron rice bowl	公家 飯	25.60	0.0416
…	…	…	…

2.2 The Method

In this section, we describe formally how CLASS works. We assume the availability of a parallel corpus and a list of collocations in a bilingual MRD. We also assume that the sentences and words have been aligned in the parallel corpus. We will describe how **CLASS** extracts bilingual collocations from such a parallel corpus. CLASS carries out a number of preprocessing steps to calculate the following information:

1. lists of preferred POS patterns of collocation in both languages;
2. collocation candidates matching the preferred POS patterns;
3. n-gram statistics for both languages, N = 1, 2;
4. log likelihood ratio statistics for two consecutive words in both languages;
5. log likelihood ratio statistics for a pair of candidates of bilingual collocations across one language to the other;
6. content word alignment based on the Competitive Linking Algorithm [Melamed, 1997.]

Figure 1 illustrates how the method works for each aligned sentence pair (*C*, *E*) in the corpus. Initially, part of speech taggers process *C* and *E*. After that, collocation candidates are extracted based on preferred POS patterns and statistical association between consecutive words in a collocation. The collocation candidates are subsequently matched up from one language to the other. These pairs are sorted according to the log likelihood ratio and collocation translation probability. A greedy selection process goes through the sorted list and selects bilingual collocations subject to one-to-one constraint. The detailed algorithm is given below:

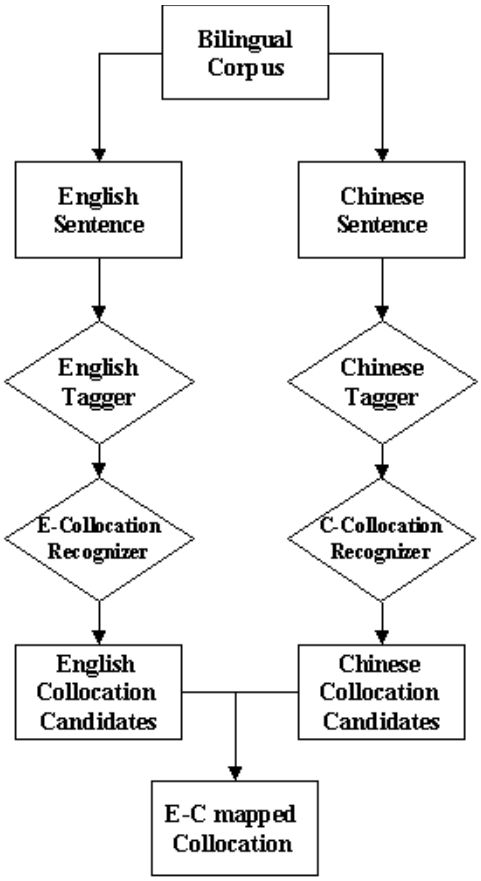


Figure 1. The major components in the proposed CLASS algorithm.

Preprocessing: Extracting preferred POS patterns *P* and *Q* in both languages

Input: A list of bilingual collocations from a machine-readable dictionary

Output:

1. Perform part of speech tagging for both languages.
2. Calculate the number of instances for all POS patterns in both languages.
3. Eliminate the POS patterns with instance counts of 1.

Collocation Linking Alignment based on Syntax and Statistics

Extract bilingual collocations from aligned sentences.

Input:

(1) A pair of aligned sentences (C, E), $C = (C_1\ C_2\ ...\ C_n)$ and $E = (E_1\ E_2\ ...\ E_m)$.

(2) Preferred POS patterns P and Q in both languages.

Output: Aligned bilingual collocations in (C, E)

1. C is segmented and tagged with part of speech information T.
2. E is tagged with part of speech sequences S.
3. Match T against P and match S against Q to extract collocation candidates $X_1, X_2,....X_k$ in English and $Y_1, Y_2, ...,Y_e$ in Chinese.
4. Consider each bilingual collocation candidate (X_i , Y_j) in turn and calculate the minimal log likelihood ratio LLR between X_i and Y_j:

   MLLR (D) = $\min_{i=1,n-1} LLR(W_i, W_{i+1})$.

5. Eliminate candidates with LLR that are smaller than a threshold (7.88).
6. Match up all possible links from English collocation candidates to Chinese ones: (D_1, F_1), (D_1, F_2), … (D_i, F_j), … (D_m, F_n).
7. Calculate LLR for (D_i, F_j) and discard pairs with LLR value that are lower than 7.88.
8. The only candidate list of bilingual collocations considered is the one with non-zero collocation translation probability P(D_i, F_j) values. The list is then sorted based on the LLR values and collocation translation probability.
9. Go down the list and select a bilingual collocation if it does not conflict with a previous selection.
10. Output the bilingual collocation selected in Step 9.

Log-likelihood ratio: LLR(x;y)

$$LLR(x,y) = -2\log_2 \frac{p_1^{k_1}(1-p_1)^{n_1-k_1}\, p_2^{k_2}(1-p_2)^{n_2-k_2}}{p^{k_1}(1-p)^{n_1-k_1}\, p^{k_2}(1-p)^{n_2-k_2}}$$

k_1 : # of pairs that contain x and y simultaneously.
k_2 : # of pairs that contain x but do not contain y.
n_1 : # of pairs that contain y
n_2 : # of pairs that does not contain y
$p_1 = k_1/n_1$, $p_2 = k_2/n_2$,
$p = (k_1+k_2)/(n_1+n_2)$

Collocation translation probability P(x | y)

$$P(D_i \mid F_j) = \frac{1}{k} \sum_{e \in F_j} \max_{c \in D_i} P(c \mid e)$$

k : number of words in the English collocation F_j

3. Experiments and Evaluation

We have implemented CLASS using the Longman Dictionary of Contemporary English, English-Chinese Edition, and the parallel corpus of Sinorama magazine. The articles from

Sinorama covered a wide range of topics, reflecting the personalities, places, and events in Taiwan for the previous three decades. We experimented on articles mainly dating from 1995 to 2002. Sentence and word alignment were carried out first to obtain the Sinorama Parallel Corpus.

Sentence alignment is a very important aspect of CLASS. It is the basis for good collocation alignment. We use a new alignment method based on punctuation statistics [Yeh & Chang, 2002]. The punctuation-based approach has been found to outperform the length-based approach with precision rates approaching 98%. With the sentence alignment approach, we obtained approximately 50,000 reliably aligned sentences containing 1,756,000 Chinese words (about 2,534,000 Chinese characters) and 2,420,000 English words in total.

The content words were aligned using the Competitive Linking Algorithm. Alignment of content words resulted in a probabilistic dictionary with 229,000 entries. We evaluated 100 random sentence samples with 926 linking types, and the achieved precision rate was 93.3%. Most of the errors occurred with English words having no counterpart in the corresponding Chinese sentence. Translators do not always translate word for word. For instance, with the word “water” in Example 5, it seems that there is no corresponding pattern in the Chinese sentence. Another major cause of errors was collocations that were not translated compositionally. For instance, the word “State” in the Example 6 is a part of the collocation “United States,” and “美國” is more highly associated with “United” than “States”; therefore, due to the one-to-one constraint “States” will not be aligned with “美國”. Most often, it will be aligned incorrectly. About 49% of the error links were of this type.

Example 5

The boat is indeed a vessel from the mainland that illegally entered Taiwan waters. The words were a "mark" added by the Taiwan Garrison Command before sending it back.

編按：此船的確是大陸偷渡來台船隻，那八個字只不過是警總在遣返前給它加的「記號」！

Source: 1990/10 Letters to the Editor

Example 6

Figures issued by the American Immigration Bureau show that most Chinese immigrants had set off from Kwangtung and Hong Kong, which is why the majority of overseas Chinese in the United States to this day are of Cantonese origin.

由美國移民局發表的數字來看，中國移民以從廣東、香港出海者最多，故到現在爲止，美國華僑仍以原籍廣東者佔大多數。

Source: 1990/09 All Across the World: The Chinese Global Village

We obtained the word-to-word translation probability from the result of word alignment. The translation probability P(*c*|*e*) is calculated as followed:

$$P(c|e) = \frac{count\ (e,c)}{count\ (e)}, \text{ where}$$

count(e,c) : the number of alignment links between a Chinese word *c* and an English word *e*;

count(e) : the number of instances of e in alignment links.

Take "pay" as an example. Table 6 shows the various alignment translations for "pay" and the translation probability.

Table 6. The aligned translations for the English word "pay" and their translation probability.

Translation	Count	Translation Prob.	Translation	Count	Translation Prob.
代價	34	0.1214	花錢	7	0.025
錢	31	0.1107	出錢	6	0.0214
費用	21	0.075	租	6	0.0214
付費	16	0.0571	發給	6	0.0214
領	16	0.0571	付出	5	0.0179
繳	16	0.0571	薪資	5	0.0179
支付	13	0.0464	付錢	4	0.0143
給	13	0.0464	加薪	4	0.0143
薪水	11	0.0393	. . .	...	...
負擔	9	0.0321	積欠	2	0.0071
費	9	0.0321	繳款	2	0.0071
給付	8	0.0286			

Before running CLASS, we obtained 10,290 English idioms, collocations, and phrases together with 14,945 Chinese translations in LDOCE. After part of speech taggi ng, we had 1,851 distinct English patterns and 4326 Chinese patterns. To calculate the statistical association within words in a monolingual collocation and across the bilingual collocations,

we built N-grams for the Sinorama Parallel Corpus. There were 790,000 Chinese word bigrams and 669,000 distinct English bigrams. CLASS identified around 595,000 Chinese collocation candidates (184,000 distinct types) and 230,000 English collocation candidates (135,000 distinct types) through this process.

We selected 100 sentences to evaluate the performance. We focused on rigid lexical collocations. The average English sentence had 45.3 words, while the average Chinese sentence had 21.4 words. The two human judges, both master students majoring in Foreign Languages, identified the bilingual collocations in these sentences. We then compared the bilingual collocations produced by CLASS against the answer keys. The evaluation produced an average recall rate = 60.9 % and precision rate = 85.2 % (see Table 7).

Table 7. Experiment results of bilingual collocation from the Sinorama Parallel Corpus.

# keys	#answers	#hits	#errors	Recall	Precision
382	273	233	40	60.9%	85.2%

4. Discussion

This paper describes a new approach to the automatic acquisition of bilingual collocations from a parallel corpus. Our method is an extension of Melamed's Competitive Linking Algorithm for word alignment. It combines both linguistic and statistical information and uses it to recognize monolingual and bilingual collocations in a much simpler way than Smadja's work does. Our approach differs from previous work in the following ways:

1. We use a data-driven approach to extract monolingual collocations.
2. Unlike Smadja and Kupiec, we do not commit to two sets of monolingual collocations. Instead, we consider many overlapping and conflicting candidates and rely on cross linguistic statistics to revolve the issue.
3. We combine both type of information related to the whole collocation as well as to the constituent words to achieve more reliable probabilistic estimation of aligned collocations.

Our approach is limited by its reliance on training data consisting of mostly rigid collocation patterns, and it is not applicable to elastic collocations such as "jump on … bandwagon." For instance, the program cannot handle the elastic collocation in the following example:

Example 7

台灣幸而趕**搭**了一程獲利豐厚的**順風車**，可以將目前剛要起步的馬來西亞、中國大陸等國家遠拋身後。

Taiwan has had the good fortune to **jump on** this high-profit **bandwagon** and has been able to snatch a substantial lead over countries like Malaysia and mainland China, which have just started in this industry.

Source: Sinorama, 1996, Dec Issue Page 22, Stormy Waters for Taiwan’ s ICs

This limitation can be partially alleviated by matching nonconsecutive word sequences against existing lists of collocations for the two languages.

Another limitation has to do with bilingual collocations, which are not literal translations. For instance, “difficult and intractable” can not yet be handled by the program, because it is not a word for word translation of “桀傲不馴”.

Example 8

意思是說一個再怎麼**桀傲不馴**的人，都會有人有辦法制服他。

This saying means that no matter how **difficult and intractable** a person may seem, there will always be someone else who can cut him down to size.

Source: 1990/05 A Fierce Horse Ridden by a Fierce Rider

In the experiment, we found that this limitation may be partially solved by splitting the candidate list of bilingual collocations into two lists: one (NZ) with non-zero phrase translation probabilistic values and the other (ZE) with zero values. The two lists can then be sorted based on the LLR values. After extracting bilingual collocations from the NZ list, we could continue to go down the ZE list and select bilingual collocations that did not conflict with previously selection.

In the proposed method, we do no take advantage of the correspondence between POS patterns in one language with those in the other. Some linking mistakes seem to be avoidable if POS information is used. For example, the aligned collocation for “issue/**vb** visas/**nns**” is “簽證/**Na**”, not “發/**VD** 簽證/**Na.**” However, the POS pattern “vb nn” appears to be more compatible with “VD Na” than with “Na.”

Example 9

一九七二年澳洲承認中共，中華民國即於此時與澳斷交。因爲無正式邦交，澳洲不能在台灣**發簽證**，而由澳洲駐香港的使館代辦，然後將簽證送回台灣，簽證手續約需五天至一周。

The Republic of China broke relations with Australia in 1972, after the country recognized the Chinese Communists, and because of the lack of formal diplomatic relations, Australia felt it could not **issue visas** on Taiwan. Instead, they were handled through its consulate in Hong Kong and then sent back to Taiwan, the entire process requiring five days to a week to complete.

Source: 1990/04 Visas for Australia to Be Processed in Just 24 Hours

A number of mistakes are caused by erroneous word segments in the Chinese tagger. For instance, "大學及研究生出國期間" should be segmented as " 大學 / 及 / 研究生 / 出國 / 期間" but instead is segmented as "大學 / 及 / 研究 / 生出 / 國 / 期間 / 的 / 學業." Another major source of segmentation mistakes has to do with proper names and their transliterations. These name entities that are not included in the database are usually segmented into single Chinese characters. For instance, "...一書作者劉學銚指出..." is segmented as " ... / 一 / 書 / 作者 / 劉 / 學 / 銚 / 指出 / ...," while "...在匈牙利地區建國的馬札爾人..." is segmented as "...在 / 匈牙利 / 地區 / 建國 / 的 / 馬 / 札 / 爾 / 人 / ...." Therefore, handling these name entities in a pre-process should be helpful to avoid segmenting mistakes and alignment difficulties.

5. Conclusion and Future Work

In this paper, we have presented an algorithm that employs syntactic and statistical analyses to extract rigid bilingual collocations from a parallel corpus. Phrases matching the preferred patterns are extracted from aligned sentences in a parallel corpus. These phrases are subsequently matched up based on cross-linguistic statistical association. Statistical association between the whole collocations as well as words in the collocations is used jointly to link a collocation with its counterpart. We implemented the proposed method on a very large Chinese-English parallel corpus and obtained satisfactory results.

A number of interesting future directions suggest themselves. First, it would be interesting to see how effectively we can extend the method to longer and elastic collocations and to grammatical collocations. Second, bilingual collocations that are proper names and transliterations may need additional consideration. Third, it will be interesting to see if the performance can be improved using cross language correspondence between POS patterns.

References

Benson, Morton., Evelyn Benson, and Robert Ilson." The BBI Combinatory Dictionary of English: A Guide to Word Combinations. " *John Benjamins, Amsterdam, Netherlands,* 1986.

Choueka, Y. "Looking for needles in a haystack", *RIAO, Conference on User-Oriented Context Based Text and Image Handling, Cambridge*, 1988, pp. 609-623.

Choueka, Y.; Klein, and Neuwitz, E.. "Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus." *Journal of the Association for Literary and Linguistic Computing*, 4(1), 1983, pp34-38.

Church, K. W. and Hanks, P. "Word association norms, mutual information, and lexicography." *Computational Linguistics*, 16(1) , 1990, pp. 22-29.

Dagan, I. and K. Church. "Termight: Identifying and translation technical terminology". In *Proc. of the 4th Conference on Applied Natural Language Processing (ANLP)*, 1994, pages 34-40.

Dunning, T. "Accurate methods for the statistics of surprise and coincidence", *Computational Linguistics* 19:1, 1993, pp.61-75.

Haruno, M., S. Ikehara, and T. Yamazaki. "Learning bilingual collocations by word-level sorting." *In Proc. of the 16th International Conference on Computational Linguistics (COLING '96)*, 1996, pp. 525-530.

Huang, C.-R., K.-J. Chen, Y.-Y. Yang, "Character-based Collocation for Mandarin Chinese", *In Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, 2000, pp. 540-543.

Inkpen, Diana Zaiu and Hirst, Graeme. "Acquiring collocations for lexical choice between near-synonyms." *In Proceedings of the Workshop on Unsupervised Lexical Acquisition, 40th Annual Meeting of the Association for Computational Lin- guistics (ACL 2002)*, 2002, pp. 67-76.

Justeson, J.S. and Slava M. Katz. "Technical Terminology: some linguistic properties and an algorithm for identification in text." *Natural Language Engineering*, 1(1), 1995, pp. 9-27.

Kupiec, Julian. "An algorithm for finding noun phrase correspondences in bilingual corpora." *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993, pp. 17-22.

Lin, D. "Using collocation statistics in information extraction." *In Proc. of the Seventh Message Understanding Conference (MUC-7)*, 1998.

Manning and H. Schutze. "Foundations of Statistical Natural Language Processing," C., MIT Press, 1999.

Melamed, I. Dan. "A Word-to-Word Model of Translational Equivalence." *In Proceedings of the 35st Annual Meeting of the Association for Computational Linguistics*, 1997, pp 490-497.

Smadja, F. “Retrieving collocations from text: Xtract.” *Computational Linguistics*, 19(1) 1993, pp143-177.

Smadja, F., K.R. McKeown, and V. Hatzivassiloglou. “Translating collocations for bilingual lexicons: A statistical approach.” *Computational Linguistics*, 22(1) ,1996, pp 1-38.

Yeh, “Using Punctuation Marks for Bilingual Sentence Alignment.” Master thesis, 2003, National Tsing Hua University, Taiwan

Computational Linguistics and Chinese Language Processing
Vol. 9, No.1, February 2004, pp. 21-40

Automatic Pronominal Anaphora Resolution in English Texts

Tyne Liang* **and Dian-Song Wu***

Abstract

Anaphora is a common phenomenon in discourses as well as an important research issue in the applications of natural language processing. In this paper, anaphora resolution is achieved by employing WordNet ontology and heuristic rules. The proposed system identifies both intra-sentential and inter-sentential antecedents of anaphors. Information about animacy is obtained by analyzing the hierarchical relations of nouns and verbs in the surrounding context. The identification of animacy entities and pleonastic-it usage in English discourses are employed to promote resolution accuracy.

Traditionally, anaphora resolution systems have relied on syntactic, semantic or pragmatic clues to identify the antecedent of an anaphor. Our proposed method makes use of WordNet ontology to identify animate entities as well as essential gender information. In the animacy agreement module, the property is identified by the hypernym relation between entities and their unique beginners defined in WordNet. In addition, the verb of the entity is also an important clue used to reduce the uncertainty. An experiment was conducted using a balanced corpus to resolve the pronominal anaphora phenomenon. The methods proposed in [Lappin and Leass, 94] and [Mitkov, 01] focus on the corpora with only inanimate pronouns such as "it" or "its". Thus the results of intra-sentential and inter-sentential anaphora distribution are different. In an experiment using Brown corpus, we found that the distribution proportion of intra-sentential anaphora is about 60%. Seven heuristic rules are applied in our system; five of them are preference rules, and two are constraint rules. They are derived from syntactic, semantic, pragmatic conventions and from the analysis of training data. A relative measurement indicates that about 30% of the errors can be eliminated by applying heuristic module.

* Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan
Email: tliang@cis.nctu.edu.tw; gis90507@cis.nctu.edu.tw

1. Introduction

1.1 Problem description

Anaphora resolution is vital in applications such as machine translation, summarization, question-answering systems and so on. In machine translation, anaphora must be resolved in the case of languages that mark the gender of pronouns. One main drawback with most current machine translation systems is that the translation produced usually does not go beyond the sentence level and, thus, does not successfully deal with discourse understanding. Inter-sentential anaphora resolution would, thus, be of great assistance in the development of machine translation systems. On the other hand, many automatic text summarization systems apply a scoring mechanism to identify the most salient sentences. However, the task results are not always guaranteed to be coherent with each other. This could lead to errors if a selected sentence contained anaphoric expressions. To improve accuracy in extracting important sentences, it is essential to solve the problem of anaphoric references beforehand.

Pronominal anaphora, where pronouns are substituted by previously mentioned entities, is a common phenomenon. This type of anaphora can be further divided into four subclasses, namely:

nominative: {he, she, it, they};

reflexive: {himself, herself, itself, themselves};

possessive: {his, her, its, their};

objective: {him, her, it, them}.

However, "it" can also be a non-anaphoric expression which does not refer to any previously mentioned item, in which case it is called an expletive or the pleonastic-it [Lappin and Leass, 94]. Although pleonastic pronouns are not considered anaphoric since they do not have antecedents to refer to, recognizing such occurrences is, nevertheless, essential during anaphora resolution. In [Mitkov, 01], non-anaphoric pronouns were found to constitute 14.2% of a corpus of 28,272 words.

Definite noun phrase anaphora occurs where the antecedent is referred by a general concept entity. The general concept entity can be a semantically close phrase, such as a synonym or super-ordinates of the antecedent [Mitkov, 99]. The word *one* has a number of different usages apart from counting. One of its important functions is as an anaphoric form. For example:

Mike has a white shirt and Jane has a red one.

Intra-sentential anaphora means that the anaphor and the corresponding antecedent occur

in the same sentence. Inter-sentential anaphora means the antecedent occurs in a sentence prior to the sentence with the anaphor. In [Lappin and Leass, 94], there were 15.9% inter-sentential cases and 84.1% intra-sentential cases in the testing results. In [Mitkov, 01], there were 33.4% inter-sentential cases and 66.6% intra-sentential cases.

Traditionally, anaphora resolution systems have relied on syntactic, semantic or pragmatic clues to identify the antecedent of an anaphor. Hobbs' algorithm [Hobbs, 76] was the first syntax-oriented method presented in this research domain. From the result of a syntactic tree, they checked the number and gender agreement between antecedent candidates and a specified pronoun. In RAP (Resolution of Anaphora Procedure) proposed by Lappin and Leass [94], an algorithm is applied to the syntactic representations generated by McCord's Slot Grammar parser, and salience measures are derived from the syntactic structure. It does not make use of semantic information or real world knowledge in choosing among the candidates. A modified version of RAP system was proposed by [Kennedy and Boguraev, 96]. It employed only part-of-speech tagging with a shallow syntactic parse indicating the grammatical roles of NPs and their containment in adjuncts or noun phrases.

Cardie *et al.* [99] treated coreferencing as a clustering task. Then a distance metric function was used to decide whether two noun phrases were similar or not. In [Denber, 98], an algorithm called Anaphora Matcher (AM) was implemented to handle inter-sentential anaphora in a two-sentence context. This method uses information about the sentence as well as real world semantic knowledge obtained from other sources. The lexical database system WordNet is utilized to acquire semantic clues about the words in the input sentences. It is noted that anaphora do not refer back more than one sentence in most cases. Thus, a two-sentence "window size" is sufficient for anaphora resolution in the domain of image queries.

A statistical approach to disambiguate pronoun "it" in sentences was introduced in [Dagan and Itai, 90]. The disambiguation is based on the co-occurring patterns obtained from a corpus to find the antecedent. The antecedent candidate with the highest frequency in the co-occurring patterns is selected as a match for the anaphor.

A knowledge-poor approach was proposed in [Mitkov, 98]; it can be applied to different languages (English, Polish, and Arabic). The main components of this method are the so-called "antecedent indicators" which are used to assign a score (2, 1, 0, -1) for each candidate noun phrase. The scores play a decisive role in tracking down the antecedent from a set of possible candidates. CogNIAC (COGnition eNIAC) [Baldwin, 97] is a system developed at the University of Pennsylvania to resolve pronouns using limited knowledge and linguistic resources. It is a high precision pronoun resolution system that is capable of achieving more than 90% precision with 60% recall for some pronouns. Mitkov [02] presented

a new, advanced and completely revamped version of his own knowledge-poor approach to pronoun resolution. In contrast to most anaphora resolution approaches, the system called MARS operates in the fully automatic mode. Three new indicators included in MARS are Boost Pronoun, Syntactic Parallelism and Frequent Candidates.

In [Mitkov, 01], the authors proposed an evaluation environment for comparing anaphora resolution algorithms. Performances are illustrated by presenting the results of a comparative evaluation conducted on the basis of several evaluation measures. Their testing corpus contained 28,272 words, with 19,305 noun phrases and 422 pronouns, of which 362 were anaphoric expressions. The overall success rate calculated for the 422 pronouns found in the texts was 56.9% for Mitkov's method, 49.72% for Cogniac and 61.6% for Kennedy and Boguraev's method.

2. System Architecture

2.1 Proposed System Overview

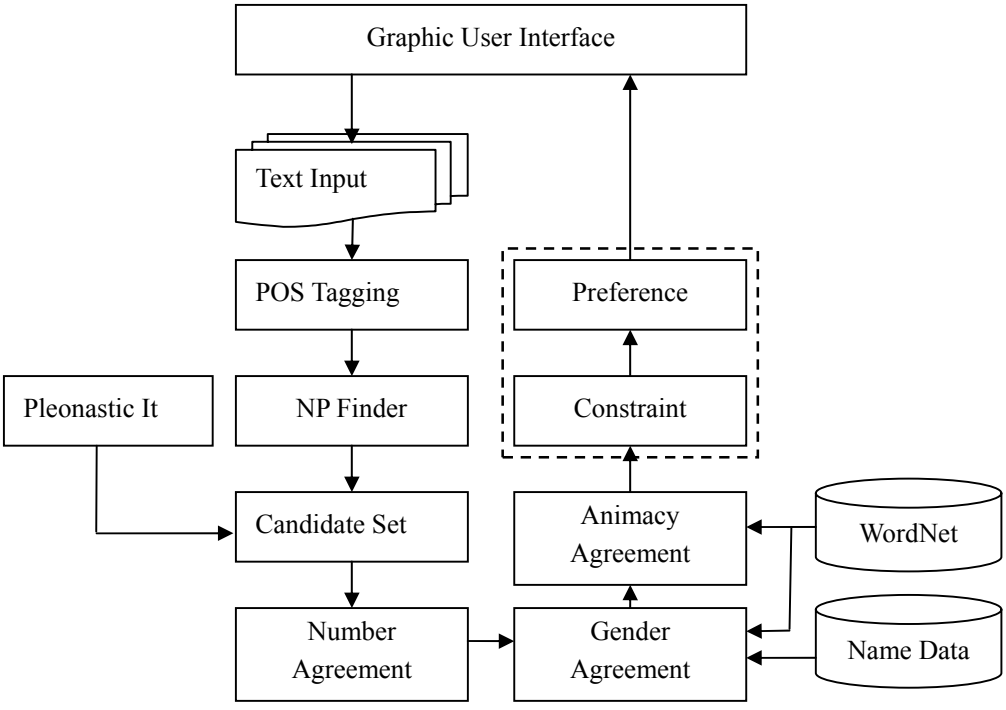


Figure 1. Architecture overview.

The procedure used to identify antecedents is described as follows:

1. Each text is parsed into sentences and tagged by POS tagger. An internal representation data structure with essential information (such as sentence offset, word offset, word POS, base form, etc.) is stored.
2. Base noun phrases in each sentence are identified by NP finder module and stored in a global data structure. Then the number agreement is applied to the head noun. Capitalized nouns in the name gazetteer are tested to find personal names. A name will be tagged with the gender feature if it can be found uniquely in male or female class defined in gender agreement module. In this phase, WordNet is also used to find possible gender clues for improving resolution performance. The gender attribute is ignored to avoid ambiguity when the person name can be masculine or feminine.
3. Anaphors are checked sequentially from the beginning of the first sentence. They are stored in a list with sentence offset and word offset information. Then pleonastic-it is checked so that no further attempts at resolution are made.
4. The remaining noun phrases preceding the anaphor within a predefined window size are collected as antecedent candidates. Then the candidate set is further filtered by means of gender and animacy agreement.
5. The remaining candidates are then evaluated by means of heuristic rules. These rules can be classified as preference rules or constraint rules. A scoring equation (equation 1) is used to evaluate how likely it is that a candidate will be selected as the antecedent. The scoring equation calculates the accumulated score of each possible candidate. The parameter $agreement_k$ denotes number agreement, gender agreement and animacy agreement output. If one of these three outputs indicates disagreement, the score will be set to zero. The parameter value enclosed in parentheses is the accumulated number of rules that fit our predefined heuristic rules:

$$score\,(can, ana) = \left(\sum_i rule\_pre_i - \sum_j rule\_con_j \right) \times \prod_k agreement_k, \quad (1)$$

where

can: each candidate noun phrase for the specified anaphor;
ana: anaphor to be resolved;
$rule\_pre_i$: the *i*th preference rule;
$rule\_con_i$: the *i*th constraint rule;
$agreement_k$: denotes number agreement, gender agreement and animacy agreement.

2.2 Main Components

2.2.1 POS Tagging

The TOSCA-ICLE tagger [Aarts *et al.*, 97] has been used to lemmatize and tag English learner corpora. The TOSCA-ICLE tag set consists of 16 major word classes. These major word

classes may be further specified by means of features of subclasses as well as a variety of syntactic, semantic and morphological characteristics.

2.2.2 NP Finder

According to the part-of-speech result, the basic noun phrase patterns are found to be as follows:

base NP → modifier + head noun

modifier → <article| number| present participle| past participle |adjective| noun>

At the beginning, our system identifies base noun phrases that contain no other smaller noun phrases within them. For example, *the chief executive officer of a financial company* is divided into *the chief executive officer* and *a financial company* for the convenience of judging whether the noun phrase is a prepositional noun phrase or not. This could be of help in selecting a correct candidate for a specific anaphor. Once the final candidate is selected, the entire modifier is combined together again.

The proposed base noun phrase finder is implemented based on a finite state machine (Figure 2). Each state indicates a particular part-of-speech of a word. The arcs between states indicate a word input from the first word of the sentence. If a word sequence can be recognized from the initial state and ends in a final state, it is accepted as a base noun phrase with no recursion; otherwise, it is rejected. An example of base noun phrase output is illustrated in Figure 3.

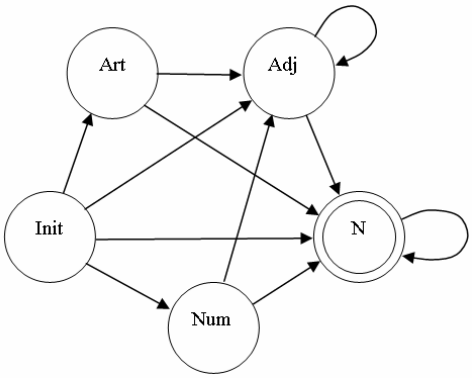


Figure 2. Finite state machine for a noun phrase.

The ministers agreed on a plan to boost screening at international departure points, bar travelers with SARS symptoms, and require health declaration forms for visitors from affected countries.

Figure 3. An example output of a base noun phrase.

2.2.3 Pleonastic-it Module

The pleonastic-it module is used to filter out those semantic empty usage conditions which are essential for pronominal anaphora resolution. A word “it” is said to be pleonastic when it is used in a discourse where the word does not refer to any antecedent.

References of “pleonastic-it” can be classified as state references or passive references [Denber, 98]. State references are usually used for assertions about the weather or the time, and this category is further divided into meteorological references and temporal references.

Passive references consist of modal adjectives and cognitive verbs. Modal adjectives (**Modaladj**) like advisable, convenient, desirable, difficult, easy, economical, certain, etc. are specified. The set of modal adjectives is extended by adding their comparative and superlative forms. Cognitive verbs (**Cogv**), on the other hand, are words like anticipate, assume, believe, expect, know, recommend, think, etc.

Most instances of "pleonastic-it" can be described by the following patterns:

1. It is **Modaladj** that **S.**
2. It is **Modaladj** (for **NP**) to **VP.**
3. It is **Cogv-ed** that **S.**
4. It seems/appears/means/follows (that) **S.**
5. **NP** makes/finds it **Modaladj** (for **NP**) to **VP.**
6. It is time to **VP.**
7. It is thanks to **NP** that **S.**

2.2.4 Number Agreement

The quantity of a countable noun can be singular (one entity) or plural (numerous entities). It makes the process of deciding on candidates easier since they must be consistent in number. With the output of the specific tagger, all the noun phrases and pronouns are annotated with number (single or plural). For a specified pronoun, we can discard those noun phrases that differ in number from the pronoun.

2.2.5 Gender Agreement

The gender recognition process can deal with words that have gender features. To distinguish the gender information of a person, we use an English first name list collected from (http://www.behindthename.com/) covering 5,661 male first name entries and 5,087 female ones. In addition, we employ some useful clues from WordNet results by conducting keyword search around the query result. These keywords can be divided into two classes：

Class_Female= {feminine, female, woman, women}

Class_Male= {masculine, male, man, men}

2.2.6 Animacy Agreement

Animacy denotes the living entities which can be referred to by some gender-marked pronouns (he, she, him, her, his, hers, himself, herself) in texts. Conventionally, animate entities include people and animals. Since it is hard to obtain the property of animacy with respect to a noun phrase by its surface morphology, we use WordNet [Miller, 93] to recognize animate entities in which a noun can only have one hypernym but can have many hyponyms. With twenty-five unique beginners, we observe that two of them can be taken as representations of animacy. These two unique beginners are {animal, fauna} and {person, human being}. Since all the hyponyms inherit properties from their hypernyms, the animacy of a noun can be determined by making use of this hierarchical relation. However, a noun may have several senses, depending on the context. The output result with respect to a noun must be employed to resolve this problem. First of all, a threshold value t_noun is defined (equation 2) as the ratio of the number of senses in animacy files to the number of total senses. This threshold value can be obtained by training a corpus, and the value is selected when the accuracy rate reaches its maximum:

$$t\_noun = \frac{the\_number\_of\_senses\_in\_animacy\_files}{the\_total\_senses\_of\_the\_noun}, \tag{2}$$

$$t\_verb = \frac{the\_number\_of\_senses\_in\_animacy\_files}{the\_total\_senses\_of\_the\_verb}, \tag{3}$$

$$accuracy = \frac{the\_number\_of\_animacy\_entities\_identified\_correctly}{the\_total\_number\_of\_animacy\_entities}. \tag{4}$$

Besides the noun hypernym relation, unique beginners of verbs are also taken into consideration. These lexicographical files with respect to verb synsets are {cognition}, {communication}, {emotion}, and {social} (Table 1). The sense of a verb, for example "read," varies from context to context as well. We can also define a threshold value t_verb as the ratio of the number of senses in animacy files (Table 1) to the number of total senses.

Table 1. Example of an animate verb.

Unique beginners	Example of verb
{cognition}	Think, analyze, judge …
{communication}	Tell, ask, teach …
{emotion}	Feel, love, fear …
{social}	Participate, make, establish …

The training data that we obtained from the Brown corpus consisted of 10,134 words, 2,155 noun phrases, and 517 animacy entities. We found that 24% of the noun phrases in the corpus referred to animate entities, whereas 76% of them referred to inanimate ones. We utilized the ratio of senses from the WordNet output to decide whether the entity was an animate entity or not. Therefore, the ratio of senses in the noun and its verb is obtained in the training phase to achieve the highest possible accuracy. Afterwards, the testing phase makes use of these two threshold values to decide on the animate feature. Threshold values can be obtained by training on the corpus and selecting the value when the accuracy rate (equation 4) reaches its maximum. Therefore, t_noun and t_verb were found to be 0.8 and 0.9, respectively, according to the distribution in Figure 4.

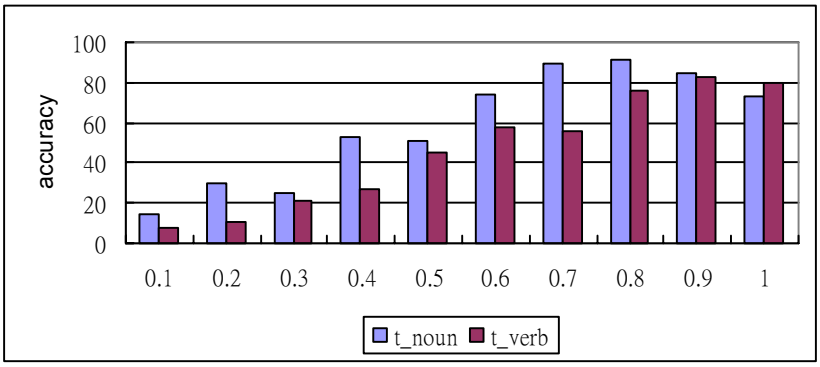
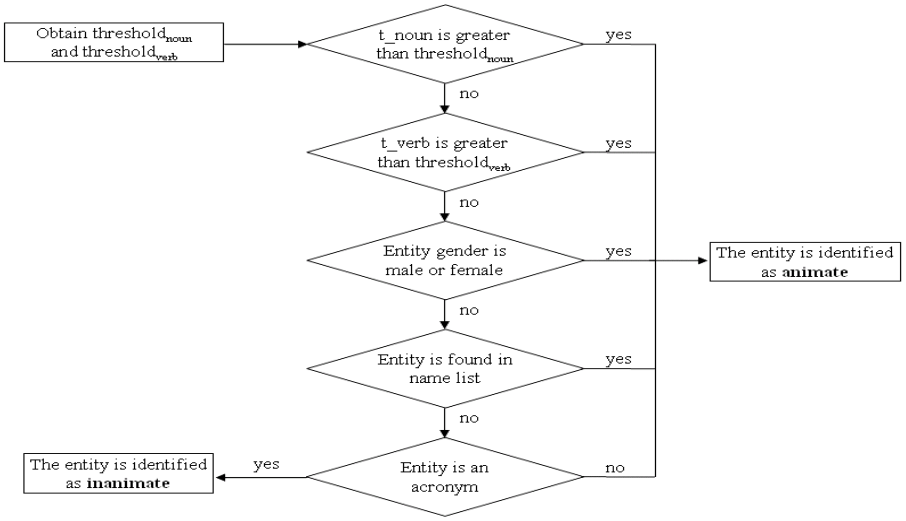


Figure 4. Thresholds of Animacy Entities.

The process of determining whether a noun phrase is animate or inanimate is described below：



2.2.7 Heuristic Rules

I. Syntactic parallelism rule

The syntactic parallelism of an anaphor and an antecedent could be an important clue when other constraints or preferences can not be employed to identify a unique unambiguous antecedent. The rule reflects the preference that the correct antecedent has the same part-of-speech and grammatical function as the anaphor. Nouns can function grammatically as subjects, objects or subject complements. The subject is the person, thing, concept or idea that is the topic of the sentence. The object is directly or indirectly affected by the nature of the verb. Words which follow verbs are not always direct or indirect objects. After a particular kind of verb, such as verb "be", nouns remain in the subjective case. We call these subjective completions or subject complements.

For example:

The security guard took off **the uniform** after getting off duty.

He put **it** in the bottom of the closet.

"**He**" (the subject) in the second sentence refers to "**The security guard,**" which is also the subject of the first sentence. In the same way, "**it**" refers to "**the uniform,**" which is the object of the first sentence. Empirical evidence also shows that anaphors usually match their antecedents in terms of their syntactic functions.

II. Semantic parallelism rule

This preference works by identifying collocation patterns in which anaphora appear. In this way, the system can automatically identify semantic roles and employ them to select the most appropriate candidate. Collocation relations specify the relations between words that tend to co-occur in the same lexical contexts. The rule emphasizes that those noun phrases with the same semantic roles as the anaphor are preferred answer candidates.

III. Definiteness rule

Definiteness is a category concerned with the grammaticalization of the identifiability and non-identifiability of referents. A definite noun phrase is a noun phrase that starts with the word "the"; for example, "the young lady" is a definite noun phrase. Definite noun phrases which can be identified uniquely are more likely to be antecedents of anaphors than indefinite noun phrases.

IV. Mention Frequency rule

Recurring items in a context are regarded as likely candidates for the antecedent of an anaphor. Generally, high frequency items indicate the topic as well as the most likely candidate.

V. Sentence recency rule

Recency information is employed in most of the implementations of anaphora resolution. In [Lappin, 94], the recency factor is the one with the highest weight among a set of factors that influence the choice of antecedent. The recency factor states that if there are two (or more) candidate antecedents for an anaphor, and that all of these candidates satisfy the consistency restrictions for the anaphor (i.e., they are qualified candidates), then the most recent one (the one closest to the anaphor) is chosen. In [Mitkov *et al.*, 01], the average distance (within a sentence) between the anaphor and the antecedent was found to be 1.3, and the average distance for noun phrases was found to be 4.3 NPs.

VI. Non-prepositional noun phrase rule

A noun phrase not contained in another noun phrase is considered a possible candidate. This condition can be explained from the perspective of functional ranking: subject > direct object > indirect object. A noun phrase embedded in a prepositional noun phrase is usually an indirect object.

VII. Conjunction constraint rule

Conjunctions are usually used to link words, phrases and clauses. If a candidate is connected with an anaphor by a conjunction, the anaphora relation is hard to be constructed between these two entities.

For example:

Mr. Brown teaches in a high school. Both **Jane** and **he** enjoy watching movies on weekends.

2.3 The Brown Corpus

The training and testing texts were selected randomly from the Brown corpus. The Corpus is divided into 500 samples of about 2000 words each. The samples represent a wide range of styles and varieties of prose. The main categories are listed in Figure 5.

A. Press: Reportage	J. Learned
B. Press: Editorial	K. General Fiction
C. Press: Reviews	L. Mystery and Detective Fiction
D. Religion	M. Science Fiction
E. Skills and Hobbies	N. Adventure and Western Fiction
F. Popular Lore	P. Romance and Love Story
G. Biography, Memoirs, etc.	R. Humor
H. Miscellaneous	

Figure 5. Categories of the Brown corpus.

2.4 System functions

The main system window is shown in Figure 6. The text editor is used to input raw text without any annotations and to show the analysis result. The POS tagger component takes the input text and outputs tokens, lemmas, most likely tags and the number of alternative tags. The NP chunker makes use of a finite state machine (FSM) to recognize strings which belong to a specified regular set.

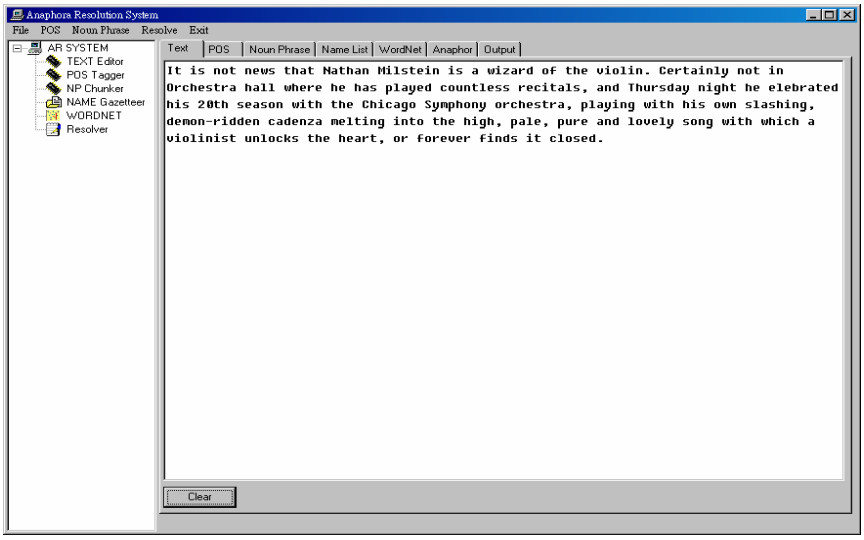


Figure 6. The main system window.

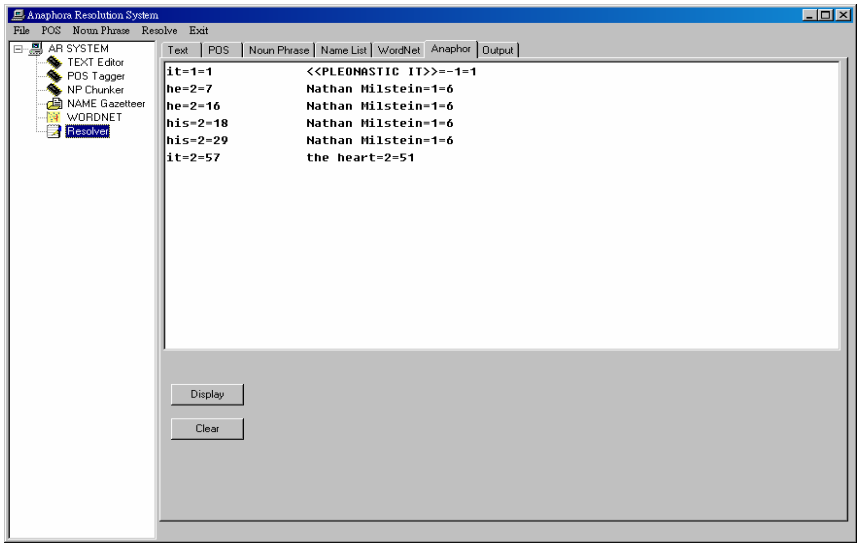


Figure 7. Anaphora pairs.

After the selection procedure is performed, the most appropriate antecedent is chosen to match each anaphor in the text. Figure 7 illustrates the result of anaphora pairs in each line, in which sentence number and word number are attached at the end of each entity. For example, "it" as the first word of the first sentence denotes a pleonastic-it, and the other "it," the 57th word of the second sentence refers to "the heart." Figure 8 shows the original text input with antecedent annotation following each anaphor in the text. All the annotations are highlighted to facilitate subsequent testing.

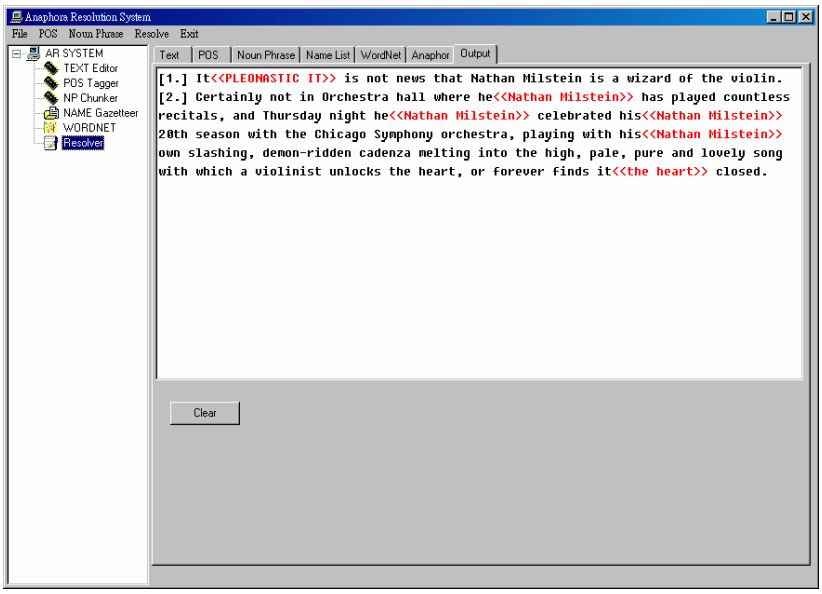


Figure 8. Anaphor with antecedent annotation.

3. Experimental Results and Analysis

The evaluation experiment employed random texts of different genres selected from the Brown corpus. There were 14,124 words, 2,970 noun phrases and 530 anaphors in the testing data. Two baseline models were established to compare the progress of performance with our proposed anaphora resolution (AR) system. The first baseline model (called the baseline subject) determined the number and gender agreement between candidates and anaphors, and then chose the most recent subject as the antecedent from the candidate set. The second baseline model (called baseline recent) performed a similar procedure, but it selected the most recent noun phrase as the antecedent which matched the anaphor in terms of number and gender agreement. The success rate was calculated as follows:

$$Success\ \ Rate = \frac{number\ \ of\ \ correctly\ \ resolved\ \ anaphors}{number\ \ of\ \ all\ \ anaphors} \tag{5}$$

The results obtained (Table 3) showed that there are 41% of the antecedents could be identified by finding the most recent subject; however, only 17% of the antecedents could be resolved by selecting the most recent noun phrase with the same gender and number agreement as the anaphor.

Table 3. Success rate of baseline models.

Genre	Baseline subject	Baseline recent
Reportage	52%	26%
Editorial	48%	15%
Reviews	32%	13%
Religion	44%	22%
Skills	41%	13%
Lore	31%	11%
Average	41%	17%

Figure 9 presents the distribution of the sentence distance between antecedents and anaphors. The value 0 denotes intra-sentential anaphora and other values indicate inter-sentential anaphora. In the experiment, a balanced corpus was used to resolve the pronominal anaphora phenomenon. The methods proposed in [Lappin and Leass, 94] and [Mitkov, 01] employ corpora with only inanimate pronouns, such as "it" or "its." Thus, the results for intra-sentential and inter-sentential anaphora distribution obtained using those methods are different. In our experiment on the Brown corpus, the distribution proportion of intra-sentential anaphora was about 60%. Figure 10 shows the average word distance distribution for each genre. The pleonastic-it could be identified with 89% accuracy (Table 4).

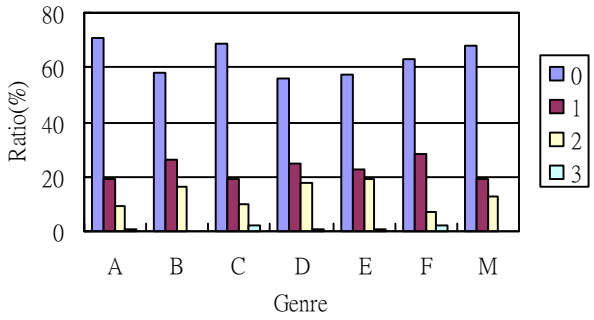


Figure 9. Referential sentence distance distribution.

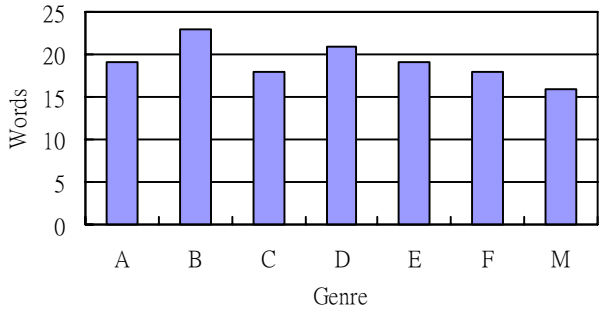


Figure 10. Referential word distance distribution.

Table 4. Pleonastic-it identification.

	Number of anaphora	Number of Anaphoric expressions	Number of Pleonastic-its	Ratio of Pleonastic-it to pronoun	Accuracy of identification
Total	530	483	47	9%	89%

The next experiment provided empirical evidence showing that the enforcement of agreement constraints increases the system's chances of selecting a correct antecedent from an initial candidate set. To access the effectiveness of each module, the total number of candidates in each genre was determined after applying the following four phases which include number agreement, gender agreement, animacy agreement, and heuristic rules (Figure 11). As shown in Figure 12, the error rates for two genres of testing data indicated the improvement in choosing correct antecedents following each resolution phase. Apparently, the animate module achieved more significant error rate reduction in the reportage domain than the other one.

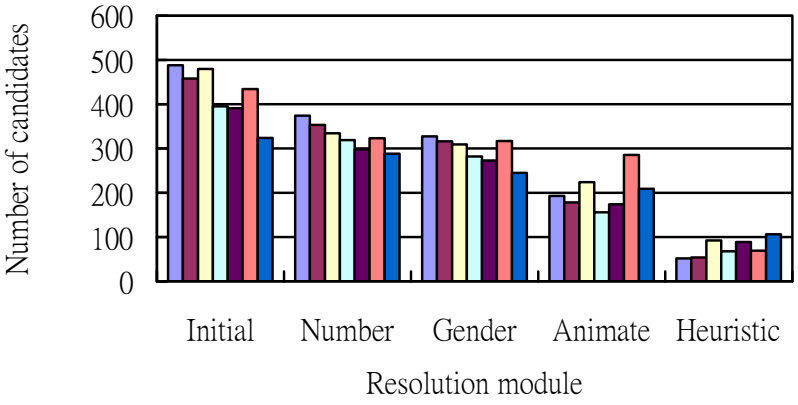


Figure 11. Candidate distribution after applying resolution modules.

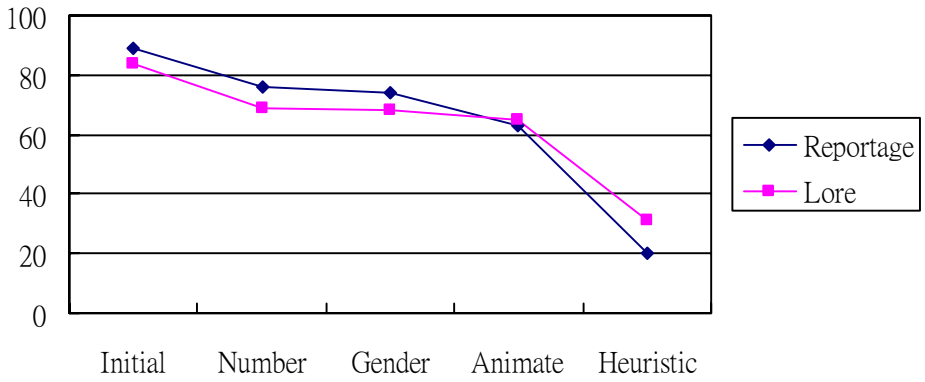


Figure 12. Error rate after applying resolution modules.

The final evaluation results obtained using our system, which applied animacy agreement and heuristic rules to resolution, are listed in Table 6. It also shows the results for each individual genre of testing data and the overall success rate, which reached 77%.

Table 6. Success rate of the AR system.

Genre	Words	Lines	NPs	Anims	Anaphors	Success Rate
Reportage	1972	90	488	110	52	80%
Editorial	1967	95	458	54	54	80%
Reviews	2104	113	480	121	92	79%
Religion	2002	80	395	75	68	76%
Skills	2027	89	391	67	89	78%
Lore	2018	75	434	51	69	69%
Fiction	2034	120	324	53	106	79%
Total	14124	662	2970	531	530	77%

Our proposed method makes use of the WordNet ontology to identify animate entities as well as essential gender information. In the animacy agreement module, each property is identified by the hypernym relation between entities and their unique beginners defined in WordNet. In addition, the verb of the entity is also an important clue for reducing the uncertainty. An overall comparison is shown below:

	Our method	[Kennedy and Boguraev, 96]	[Baldwin, 97]	[Mitkov, 98]
Hypernym relation in Noun	Y	N	N	N
Verb relation	Y	N	N	N
Name database	Y	N	N	N
Gender feature from WordNet	Y	N	N	Y
Full parsing	N	N	N	N
Heuristic	Y	Y	Y	Y
Accuracy	77%	62%	50%	57%

In the preprocessing phase, the accuracy of the POS tagger was about 95%. If a noun is misclassified as another part-of –speech, for example, if the noun “patient” is tagged as an adjective, then there is no chance for it to be considered as a legal antecedent candidate of an anaphor. The other problems encountered in the system are multiple antecedents and unknown word phenomena. In the case of multiple antecedents, the correct answer is composed of more than one entity, such as “Boys and girls are singing with pleasure.” In this case, additional

heuristic are needed to decide whether the entities should be combined into one entity or not. In the case of an unknown word, the tagger may fail to identify the part of speech of the word so that in WordNet, no unique beginner can be assigned. This can lead to a matching failure if the entity turns out to be the correct anaphoric reference.

4. Conclusion and Future Work

In this paper, the WordNet ontology and heuristic rules have been adopted to perform anaphora resolution. The recognition of animacy entities and gender features in discourses is helpful for improving resolution accuracy. The proposed system is able to deal with intra-sentential and inter-sentential anaphora in English texts and deals appropriately with pleonastic pronouns. From the experiment results, our proposed method is comparable in performance with prior works that fully parse the text. In contrast to most anaphora resolution approaches, our system benefits from the recognition of animacy agreement and operates in a fully automatic mode to achieve optimal performance. With the growing interest in natural language processing and its various applications, anaphora resolution is essential for further message understanding and the coherence of discourses during text processing.

Our future works will be as follows:

1. Extending the set of anaphors to be processed:
   This analysis aims at identifying instances (such as definite anaphors) that could be useful in anaphora resolution.
2. Resolving nominal coreferences:
   The language resource WordNet can be utilized to identify coreference entities by their synonymy/hypernym/hyponym relations.

Acknowledgement

This research is partially supported by National Science Council, R.O.C., under NSC contract 91-2213-E-009-082 and by MediaTek Research Center, National Chiao Tung University, Taiwan.

References

Jan, Aarts, Henk Barkema and Nelleke Oostdijk, “The TOSCA-ICLE Tagset: Tagging Manual,” *TOSCA Research Group for Corpus Linguistics*, 1997.

Baldwin, Breck, “CogNIAC: high precision coreference with limited knowledge and linguistic resources,” *In Proceedings of the ACL'97/EACL'97 workshop on Operational factors in practical, robust anaphora resolution*, 1997, pp. 38-45.

Bontcheva, Kalina, Marin Dimitrov, Diana Maynard and Valentin Tablan, "Shallow Methods for Named Entity Coreference Resolution," *In Proceedings of TRAITEMENT AUTOMATIQUE DES LANGUES NATURELLES (TALN)*, 2002, pp. 24-32.

Cardie, Claire and Kiri Wagstaff, "Noun Phrase Coreference as Clustering," In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

Chen, Kuang-hua and Hsin-Hsi Chen, "Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation," *In Proceedings of the 32nd ACL Annual Meeting*, 1994, pp. 234-241.

Dagan, Ido and Alon Itai, "Automatic processing of large corpora for the resolution of anaphora references," *In Proceedings of the 13th International Conference on Computational Linguistics (COLING'90), Vol. III, 1-3,* 1990.

Denber, Michel, "Automatic resolution of anaphora in English," *Technical report, Eastman Kodak Co.* , 1998.

Evans, Richard and Constantin Orasan, "Improving anaphora resolution by identifying animate entities in texts," *In Proceedings of DAARC*, 2000.

Ge, Niyu, John Hale and Eugene Charniak, "A Statistical Approach to Anaphora Resolution," *In Proceedings of the Sixth Workshop on Very Large Corpora (COLING-ACL98)*, 1998, pp.161-170.

Kennedy, Christopher and Branimir Boguraev, "Anaphora for everyone:

Pronominal anaphora resolution without a parser," In Proceedings of the 16th International Conference on Computational Linguistics, 1996, pp.113-118.

Lappin, Shalom and Herbert Leass, "An Algorithm for Pronominal Anaphora

Resolution," *Computational Linguistics*, Volume 20, Part 4, 1994, pp. 535-561.

Miller, George, "Nouns in WordNet: A Lexical Inheritance System," *Journal of Lexicography*, 1993, pp. 245-264.

Mitkov, Ruslan, "Robust pronoun resolution with limited knowledge, " In Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)/ACL'98 Conference Montreal, Canada. 1998, pp. 869-875.

Mitkov, Ruslan, "Anaphora Resolution: The State of the Art," Working paper (Based on the COLING'98/ACL'98 tutorial on anaphora resolution), 1999.

Mitkov, Ruslan and Catalina Barbu, "Evaluation tool for rule-based anaphora resolution methods," *In Proeedings of ACL'01, Toulouse*, 2001.

Mitkov, Ruslan, Richard Evans and Constantin Orasan, "A new fully automatic version of Mitkov's knowledge-poor pronoun resolution method," *In Proceedings of CICLing-2000, Mexico City, Mexico*.

Wang, Ning Yuan, Chunfa, Wang, K.F. and Li, Wenjie "Anaphora Resolution in Chinese Financial News for Information Extraction," *In Proceedings of 4th World Congress on Intelligent Control and Automation*, June 2002, pp.2422-2426.

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 1 , February 2004, pp. 41-64

Auto-Generation of NVEF Knowledge in Chinese

Jia-Lin Tsai*, Gladys Hsieh*, and Wen-Lian Hsu*

Abstract

Noun-verb event frame (NVEF) knowledge in conjunction with an NVEF word-pair identifier [Tsai *et al*. 2002] comprises a system that can be used to support natural language processing (NLP) and natural language understanding (NLU). In [Tsai *et al*. 2002a], we demonstrated that NVEF knowledge can be used effectively to solve the Chinese word-sense disambiguation (WSD) problem with 93.7% accuracy for nouns and verbs. In [Tsai *et al*. 2002b], we showed that NVEF knowledge can be applied to the Chinese syllable-to-word (STW) conversion problem to achieve 99.66% accuracy for the NVEF related portions of Chinese sentences. In [Tsai *et al*. 2002a], we defined a collection of NVEF knowledge as an NVEF word-pair (a meaningful NV word-pair) and its corresponding NVEF sense-pairs. No methods exist that can fully and automatically find collections of NVEF knowledge from Chinese sentences. We propose a method here for automatically acquiring large-scale NVEF knowledge without human intervention in order to identify a large, varied range of NVEF-sentences (sentences containing at least one NVEF word-pair). The auto-generation of NVEF knowledge (AUTO-NVEF) includes four major processes: (1) segmentation checking; (2) Initial Part-of-Speech (IPOS) sequence generation; (3) NV knowledge generation; and (4) NVEF knowledge auto-confirmation.

Our experimental results show that AUTO-NVEF achieved 98.52% accuracy for news and 96.41% for specific text types, which included research reports, classical literature and modern literature. AUTO-NVEF automatically discovered over 400,000 NVEF word-pairs from the 2001 *United Daily News* (2001 ***UDN***) corpus. According to our estimation, the acquired NVEF knowledge from 2001 ***UDN*** helped to identify 54% of the NVEF-sentences in the *Academia Sinica Balanced Corpus* (***ASBC***), and 60% in the 2001 ***UDN*** corpus.

* Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan, R.O.C.
E-mail: {tsaijl,gladys,hsu}@iis.sinica.edu.tw

We plan to expand NVEF knowledge so that it is able to identify more than 75% of NVEF-sentences in ***ASBC***. We will also apply the acquired NVEF knowledge to support other NLP and NLU researches, such as machine translation, shallow parsing, syllable and speech understanding and text indexing. The auto-generation of bilingual, especially Chinese-English, NVEF knowledge will be also addressed in our future work.

Keywords: natural language understanding, verb-noun collection, machine learning, HowNet

1. Introduction

The most challenging problem in natural language processing (NLP) is programming computers to understand natural languages. For humans, efficient syllable-to-word (STW) conversion and word sense disambiguation (WSD) occur naturally when a sentence is understood. In a natural language understanding (NLU) system is designed, methods that enable consistent STW and WSD are critical but difficult to attain. For most languages, a sentence is a grammatical organization of words expressing a complete thought [Chu 1982; Fromkin *et al*. 1998]. Since a word is usually encoded with multiple senses, to understand language, efficient word sense disambiguation (WSD) is critical for an NLU system. As found in a study on cognitive science [Choueka *et al*. 1983], people often disambiguate word sense using only a few other words in a given context (frequently only one additional word). That is, the relationship between a word and each of the others in the sentence can be used effectively to resolve ambiguity. From [Small *et al*. 1988; Krovetz *et al*. 1992; Resnik *et al*. 2000], most ambiguities occur with nouns and verbs. Object-event (i.e., noun-verb) distinction is the most prominent ontological distinction for humans [Carey 1992]. Tsai *et al*. [2002a] showed that knowledge of meaningful noun-verb (NV) word-pairs and their corresponding sense-pairs in conjunction with an NVEF word-pair identifier can be used to achieve a WSD accuracy rate of 93.7% for NV-sentences (sentences that contain at least one noun and one verb).

According to [胡裕樹 *et al*. 1995; 陳克健 *et al*. 1996; Fromkin *et al*. 1998; 朱曉亞 2001;陳昌來 2002; 劉順 2003], the most important content word relationship in sentences is the noun-verb construction. For most languages, subject-predicate (SP) and verb-object (VO) are the two most common NV constructions (or meaningful NV word-pairs). In Chinese, SP and VO constructions can be found in three language units: compounds, phrases and sentences [Li *et al*. 1997]. Modifier-head (MH) and verb-complement (VC) are two other meaningful NV word-pairs which are only found in phrases and compounds. Consider the meaningful NV word-pair ***汽車-進口***(**car**, **import**). It is an MH construction in the Chinese compound 進口汽車(import car) and a VO construction in the Chinese phrase ***進口***許多***汽車***(**import** many **cars**). In [Tsai *et al*. 2002a], we called a meaningful NV word-pair a *noun-verb event frame* (*NVEF*)

word-pair. Combining the NV word-pair *汽車-進口* and its sense-pair **Car-Import creates** a *collection* of NVEF knowledge. Since a complete event frame usually contains a predicate and its arguments, an NVEF word-pair can be a full or a partial event frame construction.

In Chinese, syllable-to-word entry is the most popular input method. Since the average number of characters sharing the same phoneme is 17, efficient STW conversion has become an indispensable tool. In [Tsai *et al*. 2002b], we showed that NVEF knowledge can be used to achieve an STW accuracy rate of 99.66% for converting NVEF related words in Chinese. We proposed a method for the semi-automatic generation of NVEF knowledge in [Tsai *et al*. 2002a]. This method uses the NV frequencies in sentences groups to generate NVEF candidates to be filtered by human editors. This process becomes labor-intensive when a large amount of NVEF knowledge is created. To our knowledge, no methods exist that can be used to fully auto-extract a large amount of NVEF knowledge from Chinese text. In the literature, most methods for auto-extracting Verb-Noun collections (i.e., meaningful NV word-pairs) focus on English [Benson *et al.* 1986; Church *et al*. 1990; Smadja 1993; Smadja *et al*. 1996; Lin 1998; Huang *et al*. 2000; Jian 2003]. However, the issue of VN collections focuses on extracting meaningful NV word-pairs, not NVEF knowledge. In this paper, we propose a new method that *automatically* generates NVEF knowledge from running texts and constructs a large amount of NVEF knowledge.

This paper is arranged as follows. In section 2, we describe in detail the auto-generation of NVEF knowledge. Experiment results and analyses are given in section 3. Conclusions are drawn and future research ideas discussed in section 4.

2. Development of a Method for NVEF Knowledge Auto-GenerationFor our auto-generate NVEF knowledge (AUTO-NVEF) system, we use HowNet 1.0 [Dong 1999] as a system dictionary. This system dictionary provides 58,541 Chinese words and their corresponding parts-of-speech (POS) and word senses (called DEF in HowNet). Contained in this dictionary are 33,264 nouns and 16,723 verbs, as well as 16,469 senses comprised of 10,011 noun-senses and 4,462 verb-senses.

Since 1999, HowNet has become one of widely used Chinese-English bilingual knowledge-base dictionaries for Chinese NLP research. Machine translation (MT) is a typical application of HowNet. The interesting issues related to (1) the overall picture of HowNet, (2) comparisons between HowNet [Dong 1999], WordNet [Miller 1990; Fellbaum 1998], Suggested Upper Merged Ontology (SUMO) [Niles *et al*. 2001; Subrata *et al*. 2002; Chung *et al*. 2003] and VerbNet [Dang *et al*. 2000; Kipper *et al*. 2000] and (3) typical applications of HowNet can be found in the 2nd tutorial of *IJCNLP-04* [Dong 2004].

2.1 Definition of NVEF Knowledge

The sense of a word is defined as its definition of concept (DEF) in HowNet. Table 1 lists three different senses of the Chinese word 車(Che[surname]/car/turn). In HowNet, the DEF of a word consists of its main feature and all secondary features. For example, in the DEF "character|文字,surname|姓,human|人,ProperName|專" of the word 車(Che[surname]), the first item "character|文字" is the main feature, and the remaining three items, surname|姓, human|人, and ProperName|專, are its secondary features. The main feature in HowNet inher-its features from the hypernym-hyponym hierarchy. There are approximately 1,500 such features in HowNet. Each one is called a *sememe*, which refers to the smallest semantic unit that cannot be reduced.

Table 1. The three different senses of the Chinese word (Che[surname]/car/turn).

C.Word [a]	E.Word [a]	Part-of-speech	Sense (i.e. DEF in HowNet)
車	Che[surname]	Noun	character|文字,surname|姓,human|人,ProperName|專
車	car	Noun	LandVehicle|車
車	turn	Verb	cut|切削

[a] C.Word means Chinese word; E.Word means English word.

As previously mentioned, a meaningful NV word-pair is a noun-verb event-frame word-pair (*NVEF word-pair*), such as **車 - 行駛**(Che[surname]/car/turn, move). In a sentence, an NVEF word-pair can take an SP or a VO construction; in a phrase/compound, an NVEF word-pair can take an SP, a VO, an MH or a VC construction. From Table 1, the only meaningful NV sense-pair for **車 - 行駛**(car, move) is **LandVehicle|車 - VehicleGo|駛**. Here, combining the NVEF sense-pair **LandVehicle|車 - VehicleGo|駛** and the NVEF word-pair **車 - 行駛** creates a *collection* of NVEF knowledge.

2.2 Knowledge Representation Tree for NVEF Knowledge

To effectively represent NVEF knowledge, we have proposed an NVEF knowledge representation tree (NVEF KR-tree) that can be used to store, edit and browse acquired NVEF knowledge. The details of the NVEF KR-tree given below are taken from [Tsai *et al*. 2002a].

The two types of nodes in the KR-tree are ***function nodes*** and ***concept nodes***. Concept nodes refer to words and senses (DEF) of NVEF knowledge. Function nodes define the relationships between the parent and children concept nodes. According to each main feature of noun senses in HowNet, we can classify noun senses into fifteen subclasses. These subclasses are 微生物(bacteria), 動物類(animal), 人物類(human), 植物類(plant), 人工物(artifact), 天

然物(natural), 事件類(event), 精神類(mental), 現象類(phenomena), 物形類(shape), 地點類(place), 位置類(location), 時間類(time), 抽象類(abstract) and 數量類(quantity). Appendix A provides a table of the fifteen main noun features in each noun-sense subclass.

As shown in Figure 1, the three function nodes that can be used to construct a collection of NVEF knowledge (LandVehicle|車- VehcileGo|駛) are as follows:

(1) **Major Event** (主要事件): The content of the major event parent node represents a noun-sense subclass, and the content of its child node represents a verb-sense subclass. A noun-sense subclass and a verb-sense subclass linked by a Major Event function node is an NVEF subclass sense-pair, such as LandVehicle|車 and VehicleGo|駛 shown in Figure 1. To describe various relationships between noun-sense and verb-sense subclasses, we have designed three subclass sense-symbols: =, which means *exact*; &, which means *like*; and %, which means *inclusive*. For example, provided that there are three senses, S_1, S_2, and S_3, as well as their corresponding words, W_1, W_2, and W_3, let

S_1 = LandVehicle|車,*transport|運送,#human|人,#die|死 W_1=靈車(hearse);

S_2 = LandVehicle|車,*transport|運送,#human|人 W_2=客車(bus);

S_3 = LandVehicle|車,police|警 W_3=警車(police car).

Then, S_3/W_3 is in the *exact*-subclass of =LandVehicle|車,police|警; S_1/W_1 and S_2/W_2 are in the *like*-subclass of &LandVehicle|車,*transport|運送; and S_1/W_1, S_2/W_2, and S_3/W_3 are in the *inclusive*-subclass of %LandVehicle|車.

(2) **Word Instance** (實例): The contents of word instance children consist of words belonging to the sense subclass of their parent node. These words are self-learned through the sentences located under the Test-Sentence nodes.

(3) **Test Sentence** (測試題): The contents of test sentence children consist of the selected test NV-sentence that provides a language context for its corresponding NVEF knowledge.

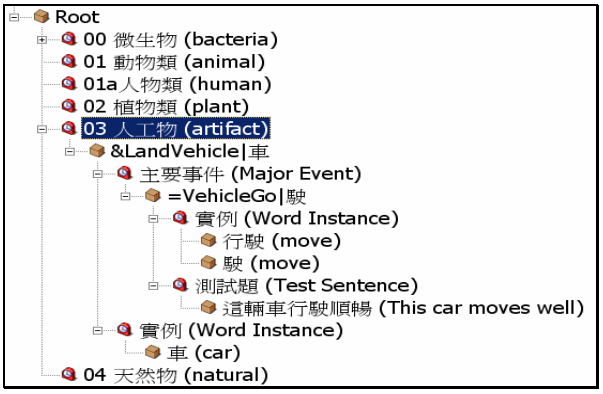


Figure 1. An illustration of the KR-tree using 人工物 (artifact) as an example of a noun-sense subclass. The English words in parentheses are provided for explanatory purposes only.

2.3 Auto-Generation of NVEF Knowledge

AUTO-NVEF automatically discovers meaningful NVEF sense/word-pairs (NVEF knowledge) in Chinese sentences. Figure 2 shows the AUTO-NVEF flow chart. There are four major processes in AUTO-NVEF. These processes are shown in Figure 2, and Table 2 shows a step by step example. A detailed description of each process is provided in the following.

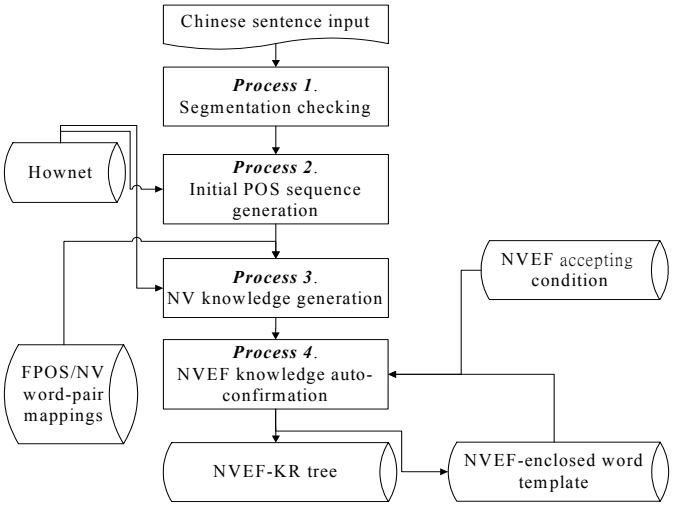


Figure 2. AUTO-NVEF flow chart.

Process 1. Segmentation checking: In this stage, a Chinese sentence is segmented according to two strategies: *forward (left-to-right) longest word first* and *backward (left-to-right) longest word first*. From [Chen *et al*. 1986], the "longest syllabic word first strategy" is effective for Chinese word segmentation. If both forward and backward segmentations are equal (forward=backward) and the word number of the segmentation is greater than one, then this segmentation result will be sent to ***process 2***; otherwise, a *NULL* segmentation will be sent. Table 3 shows a comparison of the word-segmentation accuracy for forward, backward and forward=backward strategies using the *Chinese Knowledge Information Processing* (CKIP) lexicon [CKIP 1995]. The word segmentation accuracy is the ratio of the correctly segmented sentences to all the sentences in the *Academia Sinica Balancing Corpus* (***ASBC***) [CKIP 1996]. A correctly segmented sentence means the segmented result exactly matches its corresponding segmentation in ***ASBC***. Table 3 shows that the forward=backward technique achieves the best word segmentation accuracy.

Table 2. An illustration of AUTO-NVEF for the Chinese sentence 音樂會現場湧入許多觀眾(There are many audience members entering the locale of the concert). The English words in parentheses are included for explanatory purposes only.

Process	Output
(1)	音樂會(concert)/現場(locale)/湧入(enter)/許多(many)/觀眾(audience members)
(2)	$N_1N_2V_3ADJ_4N_5$, where N_1 =[音樂會]; N_2 =[現場]; V_3=[湧入]; ADJ_4=[許多]; N_5=[觀眾]
(3)	NV1 = 現場/place|地方,#fact|事情/N - 湧入(yong3 ru4)/GoInto|進入/V NV2 = 觀眾/human|人,*look|看,#entertainment|藝,#sport|體育,*recreation|娛樂/N - 湧入(yong3 ru4)/GoInto|進入/V
(4)	NV1 is the 1st collection of NVEF knowledge confirmed by NVEF accepting-condition; the learned NVEF template is [音樂會 NV 許多] NV2 is athe 2nd collection of NVEF knowledge confirmed by NVEF accepting-condition; the learned NVEF template is [現場V許多N]

Table 3. A comparison of the word-segmentation accuracy achieved using the backward, forward and backward = forward strategies. Test sentences were obtained from ASBC, and the dictionary used was the CKIP lexicon.

	Backward	Forward	Backward = Forward
Accuracy	82.5%	81.7%	86.86%
Recall	100%	100%	89.33%

Process 2. Initial POS sequence generation: This process will be triggered if the output of ***process 1*** is not a *NULL* segmentation. It is comprised of the following steps.

1) For segmentation result $w_1/w_2/\ldots/w_{n-1}/w_n$ from ***process 1***, our algorithm computes the POS of w_i, where i = 2 to n. Then, it computes the following two sets: a) the *following POS/frequency set* of w_{i-1} according to ***ASBC*** and b) the *HowNet POS set* of w_i. It then computes the POS intersection of the two sets. Finally, it selects the POS with the highest frequency in the POS intersection as the POS of w_i. If there is zero or more than one POS with the highest frequency, the POS of w_i will be set to *NULL* POS.
2) For the POS of w_1, it selects the POS with the highest frequency in the POS intersection of the *preceding POS/frequency set* of w_2 and the *HowNet POS set* of w_1.
3) After combining the determined POSs of w_i obtained in first two steps, it then generates the ***initial POS sequence*** (***IPOS***). Take the Chinese segmentation 生/了 as an example. The following POS/frequency set of the Chinese word 生(to bear) is {N/103, PREP/42,

STRU/36, V/35, ADV/16, CONJ/10, ECHO/9, ADJ/1}(see Table 4 for tags defined in HowNet). The HowNet POS set of the Chinese word 了(a Chinese satisfaction indicator) is {V, STRU}. According to these sets, we have the POS intersection {STRU/36, V/35}. Since the POS with the highest frequency in this intersection is STRU, the POS of 了 will be set to STRU. Similarly, according to the intersection {V/16124, N/1321, ADJ/4} of the preceding POS/frequency set {V/16124, N/1321, PREP/1232, ECHO/121, ADV/58, STRU/26, CONJ/4, ADJ/4} of 了 and the HowNet POS set {V, N, ADJ} of 生, the POS of 生will be set to V. Table 4 shows a mapping list of CKIP POS tags and HowNet POS tags.

Table 4. A mapping list of CKIP POS tags and HowNet POS tags.

	Noun	Verb	Adjective	Adverb	Preposition	Conjunction	Expletive	Structural Particle
CKIP	N	V	A	D	P	C	T	De
HowNet	N	V	ADJ	ADV	PP	CONJ	ECHO	STRU

Process 3. NV knowledge generation: This process will be triggered if the ***IPOS*** output of ***process 2*** does not include any *NULL* POS. The steps in this process are given as follows.

1) Compute the ***final POS sequence*** (***FPOS***). This step translates an ***IPOS*** into an ***FPOS***. For each continuous noun sequence of ***IPOS***, the last noun will be kept, and the other nouns will be dropped. This is because a contiguous noun sequence in Chinese is usually a compound, and its head is the last noun. Take the Chinese sentence 音樂會(N_1)現場(N_2)湧入(V_3)許多(ADJ_4)觀眾(N_5) and its IPOS $N_1N_2V_3ADJ_4N_5$ as an example. Since it has a continuous noun sequence音樂會(N_1)現場(N_2), the ***IPOS*** will be translated into ***FPOS*** $N_1V_2ADJ_3N_4$, where N_1=現場, V_2=湧入, ADJ_3=許多and N_4=觀眾.
2) Generate NV word-pairs. According to the ***FPOS*** mappings and their corresponding NV word-pairs (see Appendix B), AUTO-NVEF generates NV word-pairs. In this study, we created more than one hundred ***FPOS*** mappings and their corresponding NV word-pairs. Consider the above mentioned ***FPOS*** $N_1V_2ADJ_3N_4$, where N_1=現場, V_2=湧入, ADJ_3=許多 and N_4=觀眾. Since the corresponding NV word-pairs for the ***FPOS*** $N_1V_2ADJ_3N_4$ are N_1V_2 and N_4V_2, AUTO-NVEF will generate two NV word-pairs **現場(N)湧入(V)** and**湧入(V)觀眾(N)**. In [朱曉亞 2001], there are some useful semantic structure patterns of Modern Chinese sentences for creating ***FPOS*** mappings and their corresponding NV word-pairs.
3) Generate NV knowledge. According to HowNet, AUTO-NVEF computes all the NV sense-pairs for the generated NV word-pairs. Consider the generated NV word-pairs **現場(N)湧入(V)** and **湧入(V)觀眾(N).** AUTO-NVEF will generate two collections of NV knowledge:

NV1 = [現場(locale)/place|地方,#fact|事情/N] - [湧入(enter)/GoInto|進入/V], and
NV2 = [觀眾(audience)/human|人,*look|看,#entertainment|藝,#sport|育,*recreation|娛樂/N] - [湧入(enter)/GoInto|進入/V].

Process 4. NVEF knowledge auto-confirmation: In this stage, AUTO-NVEF automatically confirms whether the generated NV knowledge is or is not NVEF knowledge. The two auto-confirmation procedures are described in the following.

(a) NVEF accepting condition (NVEF-AC) checking: Each NVEF accepting condition is constructed using a noun-sense class (such as 人物類[human]) defined in [Tsai et al. 2002a] and a verb main feature (such as GoInto|進入) defined in HowNet [Dong 1999]. In [Tsai et al. 2002b], we created 4,670 NVEF accepting conditions from manually confirmed NVEF knowledge. In this procedure, if the noun-sense class and the verb main feature of the generated NV knowledge can satisfy at least one NVEF accepting condition, then the generated NV knowledge will be auto-confirmed as NVEF knowledge and will be sent to the NVEF KR-tree. Appendix C lists the ten NVEF accepting conditions used in this study.

(b) NVEF enclosed-word template (NVEF-EW template) checking: If the generated NV knowledge cannot be auto-confirmed as NVEF knowledge in procedure (a), this procedure will be triggered. An NVEF-EW template is composed of all the left side words and right side words of an NVEF word-pair in a Chinese sentence. For example, the NVEF-EW template of the NVEF word-pair 汽車-行駛(car, move) in the Chinese sentence 這(this)/汽車(car)/似乎(seem)/行駛(move)/順暢(well) is 這 N 似乎 V 順暢. In this study, all NVEF-EW templates were auto-generated from: 1) the collection of manually confirmed NVEF knowledge in [Tsai et al. 2002], 2) the on-line collection of NVEF knowledge automatically confirmed by AUTO-NVEF and 3) the manually created NVEF-EW templates. In this procedure, if the NVEF-EW template of a generated NV word-pair matches at least one NVEF-EW template, then the NV knowledge will be auto-confirmed as NVEF knowledge.

3. Experiments

To evaluate the performance of the proposed approach to the auto-generation of NVEF knowledge, we define the NVEF accuracy and NVEF-identified sentence ratio according to Equations (1) and (2), respectively:

NVEF accuracy = # of meaningful NVEF knowledge / # of total generated NVEF knowledge; (1)

NVEF-identified sentence ratio =# of NVEF-identified sentences / # of total NVEF-sentences. (2)

In Equation (1), meaningful NVEF knowledge means that the generated NVEF knowledge has been manually confirmed to be a collection of NVEF knowledge. In Equation (2), if a Chinese sentence can be identified as having at least one NVEF word-pair by means of the generated NVEF knowledge in conjunction with the NVEF word-pair identifier proposed in [Tsai *et al*. 2002a], this sentence is called an **NVEF-identified sentence**. If a Chinese sentence contains at least one NVEF word-pair, it is called an **NVEF-sentence**. We estimate that about 70% of the Chinese sentences in ***ASBC*** are NVEF-sentences.

3.1 User Interface for Manually Confirming NVEF Knowledge

A user interface that manually confirms generated NVEF knowledge is shown in Figure 3. With it, evaluators (native Chinese speakers) can review generated NVEF knowledge and determine whether or not it is meaningful NVEF knowledge. Take the Chinese sentence 高度壓力(High pressure)使(make)有些(some)人(people)食量(eating capacity)減少(decrease) as an example. AUTO-NVEF will generate an NVEF knowledge collection that includes the NVEF sense-pair [attribute|屬性,ability|能力,&eat|吃] - [subtract|削減] and the NVEF word-pair [食量(eating capacity)] - [減少(decrease)]. The principles for confirming meaningful NVEF knowledge are given in section 3.2. Appendix D provides a snapshot of the designed user interface for evaluators for manually to use to confirm generated NVEF knowledge.

Chinese sentence	高度壓力(High pressure)使(make)有些(some)人(people)食量(eating capacity)減少(decrease)		
名詞詞義 (Noun sense)	attribute|屬性,ability|能力,&eat|吃	動詞詞義 (Verb sense)	subtract|削減
名詞 (Noun)	食量 (eating capacity)	動詞 (Verb)	減少 (decrease)

Figure 3. The user interface for confirming NVEF knowledge using the generated NVEF knowledge for the Chinese sentence 高度壓力(High pressure)使(makes)有些(some)人(people)食量(eating capacity)減少(decrease). The English words in parentheses are provided for explanatory purposes only. [ ] indicate nouns and <> indicate verbs.

3.2 Principles for Confirming Meaningful NVEF Knowledge

Auto-generated NVEF knowledge can be confirmed as meaningful NVEF knowledge if it satisfies all three of the following principles.

Principle 1. The NV word-pair produces correct noun(N) and verb(V) POS tags for the given Chinese sentence.

Principle 2. The NV sense-pair and the NV word-pair make sense.

Principle 3. Most of the inherited NV word-pairs of the NV sense-pair satisfy Principles 1 and 2.

3.3 Experiment Results

For our experiment, we used two corpora. One was the 2001 ***UDN*** corpus containing 4,539,624 Chinese sentences that were extracted from the *United Daily News* Web site [On-Line United Daily News] from January 17, 2001 to December 30, 2001. The other was a collection of specific text types, which included research reports, classical literature and modern literature. The details of the training, testing corpora and test sentence sets are given below.

(1) **Training corpus.** This was a collection of Chinese sentences extracted from the 2001 ***UDN*** corpus from January 17, 2001 to September 30, 2001. According to the training corpus, we created thirty thousand manually confirmed NVEF word-pairs, which were used to derive 4,670 NVEF accepting conditions.

(2) **Testing corpora.** One corpus was the collection of Chinese sentences extracted from the 2001 ***UDN*** corpus from October 1, 2001 to December 31, 2001. The other was a collection of specific text types, which included research reports, classical literature and modern literature.

(3) **Test sentence sets**. From the first testing corpus, we randomly selected all the sentences extracted from the news of October 27, 2001, November 23, 2001 and December 17, 2001 in 2001 ***UDN*** as our first test sentence set. From the second testing corpus, we selected a research report, a classical novel and a modern novel for our second test sentence set.

Table 5a. Experiment results of AUTO-NVEF for news.

News article date	NVEF accuracy		
	NVEF-AC	NVEF-EW	NVEF-AC + NVEF-EW
October 27, 2001	99.54%(656/659)	98.43%(439/446)	99.10% (1,095/1,105)
November 23, 2001	98.75%(711/720)	95.95%(379/395)	97.76% (1,090/1,115)
December 17, 2001	98.74%(1,015/1,028)	98.53%(1,141/1,158)	98.63% (2,156/2,186)
Total Average	98.96%(2,382/2,407)	98.00%(1,959/1,999)	98.52% (4,341/4,406)

All the NVEF knowledge acquired by AUTO-NVEF from the testing corpora was manually confirmed by evaluators. Tables 5a and 5b show the experiment results. These tables show that our AUTO-NVEF achieved 98.52% NVEF accuracy for news and 96.41% for specific text

types.

Table 5b. Experiment results of AUTO-NVEF for specific text types.

Text type	NVEF accuracy		
	NVEF-AC	NVEF-EW	NVEF-AC + NVEF-EW
Technique Report	97.12%(236/243)	96.61%(228/236)	96.86% (464/479)
Classic novel	98.64%(218/221)	93.55%(261/279)	95.80% (479/500)
Modern novel	98.18%(377/384)	95.42%(562/589)	96.51% (939/973)
Total Average	98.00%(831/848)	95.20%(1,051/1,104)	96.41% (1,882/1,952)

When we applied AUTO-NVEF to the entire 2001 ***UDN*** corpus, it auto-generated 173,744 NVEF sense-pairs (8.8M) and 430,707 NVEF word-pairs (14.1M). Within this data, 51% of the NVEF knowledge were generated based on NVEF accepting conditions (human-editing knowledge), and 49% were generated based on NVEF-enclosed word templates (machine-learning knowledge). Tables 5a and 5b show that the average accuracy of NVEF knowledge generated by NVEF-AC and NVEF-EW for news and specific texts reached 98.71% and 97.00%, respectively. These results indicate that our AUTO-NVEF has the ability to simultaneously maintain high precision and extend NVEF-EW knowledge, similar to the snowball effect, and to generate a large amount of NVEF knowledge without human intervention. The results also suggest that the best method to overcome the *Precision-Recall Tradeoff* problem for NLP is based on linguistic knowledge and statistical constraints, i.e., hybrid approach [Huang *et al.* 1996; Tsai *et al*. 2003].

3.3.1 Analysis and Classification of NVEF Knowledge

From the noun and verb positions of NVEF word-pairs in Chinese sentences, NVEF knowledge can be classified into four NV-position types: **N:V**, **N-V**, **V:N** and **V-N**, where **:** means next to and - means nearby. Table 6a shows examples and the percentages of the four NV-position types of generated NVEF knowledge. The ratios (percentages) of the collections of **N:V**, **N-V**, **V:N** and **V-N** are 12.41%, 43.83% 19.61% and 24.15%, respectively. Table 6a shows that an NVEF word-pair, such as *工程-完成*(**Construction**, **Complete**), can be an **N:V**, **N-V**, **V:N** or **V-N** in sentences. For our generated NVEF knowledge, the maximum and average number of characters between nouns and verbs in generated NVEF knowledge are 27 and 3, respectively.

Based on the numbers of noun and verb characters in NVEF word-pairs, we classify NVEF knowledge into four NV-word-length types: **N1V1**, **N1V2+**, **N2+V1** and **N2+V2+**, where N1 and V1 mean single-character nouns and verbs, respectively; N2+ and V2+ mean multi-character nouns and verbs. Table 6b shows examples and the percentages of the four NV-word-length

types of manually created NVEF knowledge for 1,000 randomly selected ***ASBC*** sentences. From the manually created NVEF knowledge, we estimate that the percentages of the collections of **N1V1**, **N1V2+**, **N2+V1** and **N2+V2+** NVEF word-pairs are 6.4%, 6.8%, 22.2% and 64.6%, respectively. According to this NVEF knowledge, we estimate that the auto-generated NVEF Knowledge (for 2001 ***UDN***) in conjunction with the NVEF word-pair identifier [Tsai *et al.* 2002] can be used to identify 54% of the NVEF-sentences in ***ASBC***.

Table 6a. An illustration of four NV-position types of NVEF knowledge and their ratios. The English words in parentheses are provided for explanatory purposes only. [ ] indicate nouns and <> indicate verbs.

Type	Example Sentence	Noun / DEF	Verb / DEF	Percentage
N:V	**[*工程*]<*完成*>** (The construction is now completed)	工程 (construction) affairs|事務,industrial|工	完成 (complete) fulfill|實現	24.15%
N-V	全部**[*工程*]**預定年底**<*完成*>** (All of constructions will be completed by the end of year)	工程 (construction) affairs|事務,industrial|工	完成 (complete) fulfill|實現	43.83%
V:N	**<*完成*>[*工程*]** (to complete a construction)	工程 (construction) affairs|事務,industrial|工	完成 (complete) fulfill|實現	19.61%
V-N	建商承諾在年底前**<*完成*>**鐵路**[*工程*]** (The building contractor promise to complete railway construction before the end of this year)	工程 (construction) affairs|事務,industrial|工	完成 (complete) fulfill|實現	12.41%

Table 6b. Four NV-word-length types of manually-edited NVEF knowledge from 1,000 randomly selected ASBC sentences and their percentages. The English words in parentheses are provided for explanatory purposes only. [ ] indicate nouns and <> indicate verbs.

Type	Example Sentence	Noun	Verb	Percentage
N1V1	然後就**<*棄*>[*我*]**而去	我(I)	棄(give up)	6.4%
N1V2+	**<*覺得*>[*他*]**很孝順	他(he)	覺得(feel)	6.8%
N2+V1	**<*買*>**了**[*可樂*]**來喝	可樂(cola)	買(buy)	22.2%
N2+V2+	**<*引爆*>**另一場美西**[*戰爭*]**	戰爭(war)	引爆(cause)	64.6%

Table 6c shows the Top 5 single-character verbs in N1V1 and N2+V1 NVEF word-pairs and their percentages. Table 6d shows the Top 5 multi-character verbs in N1V2+ and N2+V2+ NVEF word-pairs and their percentages. From Table 6c, the percentages of N2+是 and N2+有 NVEF word-pairs are both greater than those of other single-character verbs. Thus, the N2+是 and N2+有 NVEF knowledge was worthy to being considered in our AUTO-NVEF. On the other hand, we found that 3.2% of the NVEF-sentences (or 2.3% of the *ASBC* sentences) were N1V1-only sentences, where an N1V1-only sentence is a sentence that only has one N1V1-NVEF word-pair. For example, the Chinese sentence 他(he)說(say)過了(already) is an N1V1-only sentence because it has only one N1V1-NVEF word-pair: ***他-說*(he**, **say**). Since (1) N1V1-NVEF knowledge is not critical for our NVEF-based applications and (2) auto-generating N1V1 NVEF knowledge is very difficult, the auto-generation of N1V1-NVEF knowledge was not considered in our AUTO-NVEF. In fact, according to the system dictionary, the maximum and average word-sense numbers of single-character were 27 and 2.2, respectively, and those of multi-character words were 14 and 1.1, respectively.

Table 6c. The Top 5 single-character verbs in N1V1 and N2+V1 word-pairs in manually-edited NVEF knowledge for 1,000 randomly selected ASBC sentences and their percentages. The English words in parentheses are provided for explanatory purposes only. [ ] indicate nouns and <> indicate verbs.

Top	Verb of N1V1 / Example Sentence	Percentage of N1V1	Verb of N2+V1 / Example Sentence	Percentage of N2+V1
1	有(have) / [*我*]<*有*>九項獲參賽資格	16.5%	是(be) / 再來就<*是*>一間陳列樂器的[*房子*]	20.5%
2	是(be) / [*它*]<*是*>做人的根本	8.8%	有(have) / 是不是<*有*>[*問題*]了	15.5%
3	說(speak) / [*他*]<*說*>	7.7%	說(speak) / 而談到成功的秘訣[*妮娜*]<*說*>	3.9%
4	看(see) / <*看*>著[*它*]被卡車載走	4.4%	到(arrive) / 一[*到*]<*陰天*>	3.6%
5	買(buy) / 美國本土的人極少到那兒<*買*>[*地*]	3.3%	讓(let) / <*讓*>現職[*人員*]無處棲身	2.5%

Table 6d. The Top 5 multi-character verbs in N1V2+ and N2+V2+ word-pairs in manually-edited NVEF knowledge for 1,000 randomly selected ASBC sentences and their percentages. The English words in parentheses are provided for explanatory purposes only. [ ] indicate nouns and <> indicate verbs.

Top	Verb of N1V2+ / Example Sentence	Percentage of N1V2+	Verb of N2+V2+ / Example Sentence	Percentage of N2+V2+
1	吃到(eat) / 你也可能<*吃到*>毒[*魚*]	2.06%	表示(express) / 這位[*官員*]<*表示*>	1.2%
2	知道(know) / [*我*]<*知道*>哦	2.06%	使用(use) / 歌詞<*使用*>日常生活[*語言*]	1.1%
3	喜歡(like) / 至少還有人<*喜歡*>[*他*]	2.06%	沒有(not have) / 我們就<*沒有*>什麼[*利潤*]了	0.9%
4	充滿(fill) / [*心*]裡就<*充滿*>了感動與感恩	2.06%	包括(include) / <*包括*>被監禁的民運[*人士*]	0.8%
5	打算(plan) / [*你*]<*打算*>怎麼試	2.06%	成為(become) / 這種與上司<*成為*>知心[*朋友*]的作法	0.7%

3.3.2 Error Analysis - Non-Meaningful NVEF Knowledge Generated by AUTO-NVEF

One hundred collections of manually confirmed non-meaningful NVEF (NM-NVEF) knowledge from the experiment results were analyzed. We classified them according to eleven error types, as shown in Table 7, which lists the NM-NVEF confirmation principles and the percentages for the eleven error types. The first three types comprised 52% of the NM-NVEF cases that did not satisfy NVEF confirmation principles 1, 2 and 3. The fourth type was rare, representing only 1% of the NM-NVEF cases. Type 5, 6 and 7 errors comprised 11% of the NM-NVEF cases and were caused by HowNet lexicon errors, such as the incorrect DEF (word-sense) ***exist|存在*** for the Chinese word 盈盈 (an adjective, normally used to describe someone's beautiful smile). Type 8, 9, 10 and 11 errors are referred to as *four NLP errors* and comprised 36% of the NM-NVEF cases. Type 8 errors were caused by the different word-senses used in Old and Modern Chinese; Type 9 errors were caused by errors in WSD; Type 10 errors were caused by the unknown word problem; and Type 11 errors were caused by incorrect word segmentation.

Table 8 gives examples for each type of NP-NVEF knowledge. From Table 7, 11% of the NM-NVEF cases could be resolved by correcting the lexicon errors in HowNet [Dong 1999]. The four types of NLP errors that caused 36% of the NM-NVEF cases could be eliminated by using other techniques such as WSD ([Resnik *et al*. 2000; Yang *et al*. 2002]), unknown word identification ([Chang *et al*. 1997; Lai *et al*. 2000; Chen *et al*. 2002; Sun *et al*. 2002; and Tsai *et*

al. 2003]) or word segmentation ([Sproat *et al*. 1996; Teahan *et al*. 2000]).

Table 7. Eleven error types and their confirmation principles for non-meaningful NVEF knowledge generated by AUTO-NVEF.

Type	**Confirmation Principle for Non-Meaningful NVEF Knowledge**	Percentage
1*	NV Word-pair that cannot make a correct or sensible POS tag for the Chinese sentence	33% (33/100)
2*	The combination of an NV sense-pair (DEF) and an NV word-pair that cannot be an NVEF knowledge collection	17% (17/100)
3*	One word sense in an NV word-pair that does not inherit its corresponding noun sense or verb sense	2% (2/100)
4	The NV word-pair is not an NVEF word-pair for the sentence although it satisfies all the confirmiation principles	1% (1/100)
5	Incorrect word POS in HowNet	1% (1/100)
6	Incorrect word sense in HowNet	3% (3/100)
7	No proper definition in HowNet Ex:暫居(temporary residence) has two meanings: one is <reside|住下>（緊急暫居服務(emergency temporary residence service)）and another is <situated|處,Timeshort|暫>（SARS 帶來暫時性的經濟震盪(SARS will produce only a temporary economic shock))	7% (7/100)
8	Noun senses or verb senses that are used in Old Chinese	3% (3/100)
9	Word sense disambiguation failure (1) Polysemous words (2) Proper nouns identified as common words Ex: 公牛隊(**Chicago Bulls**) ⇨公牛(**bull**) <livestock|牲畜>；太陽隊(**Phoenix Suns**) ⇨太陽(**Sun**) <celestial|天體>；花木蘭(**HwaMulan**)⇨木蘭(**magnolia**)< FlowerGrass|花草>	27% (27/100)
10	Unknown word problem	4% (4/100)
11	Word segmentation error	2% (2/100)

* Type 1,2 and 3 errors are the failed results from the three confirmation principles for meaningful NVEF knowledge mentioned in section 3.2, respectively.

Table 8. Examples of eleven types of non-meaningful NVEF knowledge. The English words in parentheses are provided for explanatory purposes only. [ ] indicate nouns and <> indicate verbs.

NP type	Test Sentence	Noun / DEF	Verb / DEF
1	警方維護地方[治安]<辛勞> (Police work hard to safeguard local security.)	治安 (public security) attribute|屬性,circumstances|境況,safe|安,politics|政,&organization|組織	辛勞 (work hard) endeavour|賣力
2	<模糊>的[白宮]景象 (The White House looked vague in the heavy fog.)	白宮 (White House) house|房屋,institution|機構,#politics|政,(US|美國)	模糊 (vague) PolysemousWord|多義詞,CauseToDo|使動,mix|混合
3	<生活>條件[不足] (Lack of living conditions)	不足 (lack) attribute|屬性,fullness|空滿,incomplete|缺,&entity|實體	生活 (life) alive|活著
4	網路帶給[企業]許多<便利> (The Internet brings numerous benefits to industries.)	企業 (Industry) InstitutePlace|場所,*produce|製造,*sell|賣,industrial|工,commercial|商	便利 (benefit) benefit|便利
5	<盈盈>[笑靨] (smile radiantly)	笑靨 (a smiling face) part|部件,%human|人,skin|皮	盈盈 (an adjective normally used to describe someone's beautiful smile) exist|存在
6	保費較貴的<壽險>[保單] (higher cost life insurance policy)	保單 (insurance policy) bill|票據,*guarantee|保證	壽險 (life insurance) guarantee|保證,scope=die|死,commercial|商
7	債券型基金吸金[存款]<失血> Bond foundation makes profit but savings are lost	存款 (bank savings) money|貨幣,$SetAside|留存	失血 (bleed or lose(only used in finance diction)) bleed|出血
8	華南[銀行] 中山<分行> (Hwa-Nan Bank, Jung-San Branch)	銀行 (bank) InstitutePlace|場所,@Set Aside|留存,@TakeBack|取回,@lend|借出,#wealth|錢財,commercial|商	分行 (branch) separate|分離
9	[根據]<調查> (according to the investigation)	根據 (evidence) information|信息	調查 (investigate) investigate|調查
10	<零售>[通路] (retailer)	通路 (route) facilities|設施,route|路	零售 (retail sales) sell|賣
11	從今日<起到> 5[月底] (from today to the end of May)	月底 (the end of the month) time|時間,ending|末,month|月	起到 (to elaborate) do|做

4. Conclusions and Directions for Future Research

In this paper, we have presented an auto-generation system for NVEF knowledge (AUTO-NVEF) that fully and automatically discovers and constructs a large amount of NVEF knowledge for NLP and NLU systems. AUTO-NVEF uses both human-editing knowledge (HowNet conceptual constraints) and machine-learning knowledge (word-context patterns). Experimental results show that AUTO-NVEF achieves 98.52% accuracy for news and 96.41% accuracy for specific text types. The average number of characters between nouns and verbs in NVEF knowledge is 3. Since only 2.3% of the sentences in ***ASBC*** are N1V1-only sentences, N1V1 NVEF knowledge should not be a critical issue for NVEF-based applications. From our experimental results, neither word-segmentation nor POS tagging are critical issues for our AUTO-NVEF. The critical problems, about 60% of the error cases, were caused by failed word-sense disambiguation (WSD) and HowNet lexicon errors. Therefore, AUTO-NVEF using conventional maximum matching word-segmentation and bi-grams like POS tagging algorithms was able to achieve more than 98% accuracy for news. By applying AUTO-NVEF to the 2001 ***UDN*** corpus, we created 173,744 NVEF sense-pairs (8.8M) and 430,707 NVEF word-pairs (14.1M) in an NVEF-KR tree. Using this collection of NVEF knowledge and an NVEF word-pair identifier [Tsai *et al*. 2002], we achieved a WSD accuracy rate of 93.7% and a STW accuracy rate of 99.66% for the NVEF related portions of Chinese sentences. To sum up of the experimental results in [Tsai *et al*. 2002] and [Wu *et al*. 2003a; Wu *et al*. 2003b], NVEF knowledge was investigated and shown to be useful for WSD, STW, domain event extraction, domain ontology generation and text categorization.

According to our estimation, the auto-acquired NVEF knowledge from the 2001 ***UDN*** corpus combined with the NVEF word-pair identifier [Tsai *et al*. 2002] could be used to identify 54% and 60% of the NVEF-sentences in ***ASBC*** and in the 2001 ***UDN*** corpus, respectively. Since 94.73% (9,345/9,865) of the nouns in the most frequent 60,000 CKIP lexicon are contained in NVEF knowledge constructions, the auto-generated NVEF knowledge can be an acceptably large amount of NVEF knowledge for NLP/NLU systems. We found that the remaining 51.16% (5,122/10,011) of the noun-senses in HowNet were caused by two problems. One was that words with multiple noun-senses or multiple verb-senses, which are not easily resolved by WSD (for example, fully-automatic machine learning techniques), especially for single-character words. In our system dictionary, the maximum and average word-sense numbers of single-character words are 27 and 2.2, respectively. The other problem was corpus sparsness. We will continue expanding our NVEF knowledge through other corpora so that we can identify more than 75% of the NVEF-sentences in ***ASBC***. AUTO-NVEF will be extended to auto-generate other meaningful content word constructions, in particular, meaningful noun-noun, noun-adjective and verb-adverb word-pairs. In addition, we will investigate the effectiveness of NVEF knowledge in other NLP and NLU applications, such as syllable and speech understanding as well as full

and shallow parsing. In [董振東 1998; Jian 2003; Dong 2004], it was shown that the knowledge in bilingual Verb-Noun (VN) grammatical collections, i.e., NVEF word-pairs, is critically important for machine translation (MT). This motivates further work on the auto-generation of bilingual, especially Chinese-English, NVEF knowledge to support MT research.

Acknowledgements

We are grateful to our colleagues in the Intelligent Agent Systems Laboratory (IASL): Li-Yeng Chiu, Mark Shia, Gladys Hsieh, Masia Yu, Yi-Fan Chang, Jeng-Woei Su and Win-wei Mai, who helped us create and verify all the NVEF knowledge and tools for this study. We would also like to thank Professor Zhen-Dong Dong for providing the HowNet dictionary.

Reference

Benson, M., E. Benson, and R. Ilson, *The BBI Combination Dictionary of English: A Guide to Word Combination*, John Benjamins, Amsterdam, Netherlands, 1986.

Carey, S., "The origin and evolution of everyday concepts (In R. N. Giere, ed.)," *Cognitive Models of Science*, Minneapolis: University of Minnesota Press, 1992.

Chang, J. S. and K. Y. Su, "An Unsupervised Iterative Method for Chinese New Lexicon Extraction," *International Journal of Computational Linguistics & Chinese language Processing*, 1997Choueka, Y. and S. Lusignan, "A Connectionist Scheme for Modeling Word Sense Disambiguation," *Cognition and Brain Theory*, 6(1), 1983, pp.89-120.

Chen, C.G., K.J. Chen and L.S. Lee, "A Model for Lexical Analysis and Parsing of Chinese Sentences," *Proceedings of 1986 International Conference on Chinese Computing, Singapore*, 1986, pp.33-40.

Chen, K. J. and W. Y. Ma, "Unknown Word Extraction for Chinese Documents," *Proceedings of 19th COLING 2002*, Taipei, 2002, pp.169-175.

Chu, S. C. R., *Chinese Grammar and English Grammar: a Comparative Study*, The Commerical Press, Ltd. The Republic of China, 1982.

Chung, S. F., Ahrens, K., and Huang C. "ECONOMY IS A PERSON: A Chinese-English Corpora and Ontological-based Comparison Using the Conceptual Mapping Model," *In Proceedings of the 15th ROCLING Conference for the Association for Computational Linguistics and Chinese Language Processing*, National Tsing-Hwa University, Taiwan, 2003, pp.87-110.

Church, K. W. and P. Hanks, "Word Association Norms, Mutual Information, and Lexicongraphy," *Computational Linguistics*, 16(1), 1990, pp.22-29.

CKIP(Chinese Knowledge Information processing Group), *Technical Report no. 95-02, the content and illustration of Sinica corpus of Academia Sinica*. Institute of Information Science, Academia Sinica, 1995. http://godel.iis.sinica.edu.tw/CKIP/r_content.html

CKIP(Chinese Knowledge Information processing Group), *A study of Chinese Word Boundaries and Segmentation Standard for Information processing (in Chinese)*. Technical Report, Taiwan, Taipei, Academia Sinica, 1996.

Dang, H. T., K. Kipper and M. Palmer, "Integrating compositional semantics into a verb lexicon," *COLING-2000 Eighteenth International Conference on Computational Linguistics*, Saarbrücken, Germany, July 31 - August 4, 2000.

Dong, Z. and Q. Dong, *HowNet*, http://www.keenage.com/, 1999.

Dong, Z., Tutorials of HowNet, *The First International Joint Conference on Natural Language Processing (IJCNLP-04)*, 2004.

Fellbaum, C., *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

Fromkin, V. and R. Rodman, *An Introduction to Language*, Sixth Edition, Holt, Rinehart and Winston, 1998.

Huang, C. R., K. J. Chen, Y. Y. Yang, "Character-based Collection for Mandarin Chinese," *In ACL 2000*, 2000, pp.540-543.

Huang, C. R., K. J. Chen, "Issues and Topics in Chinese Natural Language Processing," *Journal of Chinese Linguistics*, Monograph series number 9, 1996, pp.1-22.

Jian, J. Y., "Extracting Verb-Noun Collections from Text," *In Proceedings of the 15th ROCLING Conference for the Association for Computational Linguistics and Chinese Language Processing*, National Tsing-Hwa University, Taiwan, 2003, pp.295-302.

Kipper K., H. T. Dang and M. Palmer, "Class-Based Construction of a Verb Lexicon," *AAAI-2000 Seventeenth National Conference on Artificial Intelligence*, Austin, TX, July 30 - August 3, 2000.

Krovetz, R. and W. B. Croft, "Lexical Ambiguity and Information Retrieval," *ACM Transactions on Information Systems*, 10(2), 1992, pp.115-141.

Lai, Y. S. and Wu, C. H., "Unknown Word and Phrase Extraction Using a Phrase-Like-Unit-based Likelihood Ratio," *International Journal of Computer Processing Oriental Language*, 13(1), 2000, pp.83-95.

Li, N. C. and S. A. Thompson, *Mandarin Chinese: a Functional Reference Grammar*, The Crane Publishing Co., Ltd. Taipei, Taiwan, 1997.

Lin, D., "*Using Collection Statistics in Information Extraction*," In Proc. of the Seventh Message Understanding Conference (MUC-7), 1998.

Miller G., "WordNet: An On-Line Lexical Database," *International Journal of Lexicography*, 3(4), 1990.

Niles, I., and Pease, A, "Origins of the Standard Upper Merged Ontology: A Proposal for the IEEE Standard Upper Ontology," *In Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*, Seattle, Washington, August 6, 2001.

On-Line United Daily News, http://udnnews.com/NEWS/

Resnik, P. and D. Yarowsky, "Distinguishing Systems and Distinguishing Senses: New Evaluation Methods for Word Sense Disambiguation," *Natural Language Engineering*, 5(3), 2000, pp.113-133.

Smadjia, F., "Retrieving Collections from Text: Xtract," *Computational Linguistics*, 19(1), pp.143-177

Smadjia, F., K. R. McKeown, and V. Hatzivassiloglou, "Translating Collections for Bilingual Lexicons: A Statistical Approach," *Computational Linguistics*, 22(1) 1996, pp.1-38.

Small, S., and G. Cottrell, and M. E. Tannenhaus, *Lexical Ambiguity Resolution*, Morgan Kaufmann, Palo Alto, Calif., 1988.

Subrata D., Shuster K., and Wu, C., "Ontologies for Agent-Based Information Retrieval and Sequence Mining," *In Proceedings of the Workshop on Ontologies in Agent Systems (OAS02)*, held at the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems Bologna, Italy, July, 2002, pp.15-19.

Sun, J., J. Gao, L. Zhang, M. Zhou and C. Huang, "Chinese Named Entity Identification Using Class-based Language Model," *In the Proceedings of 19th COLING 2002*, Taipei, 2000, pp.967-973.

Sproat, R. and C. Shih, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese," *Computational Linguistics*, 22(3), 1996, pp.377-404.

Teahan, W.J., Wen, Y., McNab, R.J., Witten, I.H., "A compression-based algorithm for chinese word segmentation," *Computational Linguistics*, 26, 2000, pp.375-393.

Tsai, J. L, W. L. Hsu and J. W. Su, "Word sense disambiguation and sense-based NV event-frame identifier," *Computational Linguistics and Chinese Language Processing,* Vol. 7, No. 1, February 2002, pp.29-46.

Tsai, J. L, W. L. Hsu, "Applying NVEF Word-Pair Identifier to the Chinese Syllable-to-Word Conversion Problem," *Proceedings of 19th COLING 2002*, Taipei, 2002, pp.1016-1022.

Tsai, J. L, C. L. Sung and W. L. Hsu, "Chinese Word Auto-Confirmation Agent," *In Proceedings of the 15th ROCLING Conference for the Association for Computational Linguistics and Chinese Language Processing*, National Tsing-Hwa University, Taiwan, 2003, pp.175-192.

Wu, S. H., T. H. Tsai, and W. L. Hsu, "Text Categorization Using Automatically Acquired Domain Ontology," *In proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages (IRAL-03)*, Sapporo, Japan, 2003, pp.138-145.

Wu, S. H., T. H. Tsai, and W. L. Hsu, "Domain Event Extraction and Representation with Domain Ontology," *In proceedings of the IJCAI-03 Workshop on Information Integration on the Web*, Acapulco, Mexico, 2003, pp.33-38.

Yang, X. and Li T., "A study of Semantic Disambiguation Based on HowNet," *Computational Linguistics and Chinese Language Processing,* Vol. 7, No. 1, February 2002, pp.47-78.

朱曉亞，*現代漢語句模研究(Studies on Semantic Structure Patterns of Sentence in Modren Chinese)*，北京大學出版社，2001.

胡裕樹，范曉，*動詞研究*，河南大學出版社，1995.

董振東，語義關係的表達和知識系統的建造，*語言文字應用*，第 3 期，1998，頁 76-82.

陳克健，洪偉美，中文裏「動—名」述賓結構與「動—名」偏正結構的分析，*Communication of COLIPS*, 6(2), 1996, 頁 73-79.

陳昌來，*現代漢語動詞的句法語義屬性研究(XIANDAI HANYU DONGCI DE JUFA YUYI SHUXING YANJIU)*，學林出版社，2002.

劉順，*現代漢語名詞的多視角研究(XIANDAI HANYU DONGCI DE JUFA YUYI SHUXING YANJIU)*，學林出版社，2003.

Appendix A. Sample Table of Main Noun Features and Noun-Sense Classes

Main noun features	Noun-sense classes
bacteria|微生物	微生物(bacteria)
AnimalHuman|動物	動物類(animal)
human|人	人物類(human)
plant|植物	植物類(plant)
artifact|人工物	人工物(artifact)
natural|天然物	天然物(natural)
fact|事情	事件類(event)
mental|精神	精神類(mental)
phenomena|現象	現象類(phenomena)
shape|物形	物形類(shape)
InstitutePlace|場所	地點類(place)
location|位置	位置類(location)
attribute|屬性	抽象類(abstract)
quantity|數量	數量類(quantity)

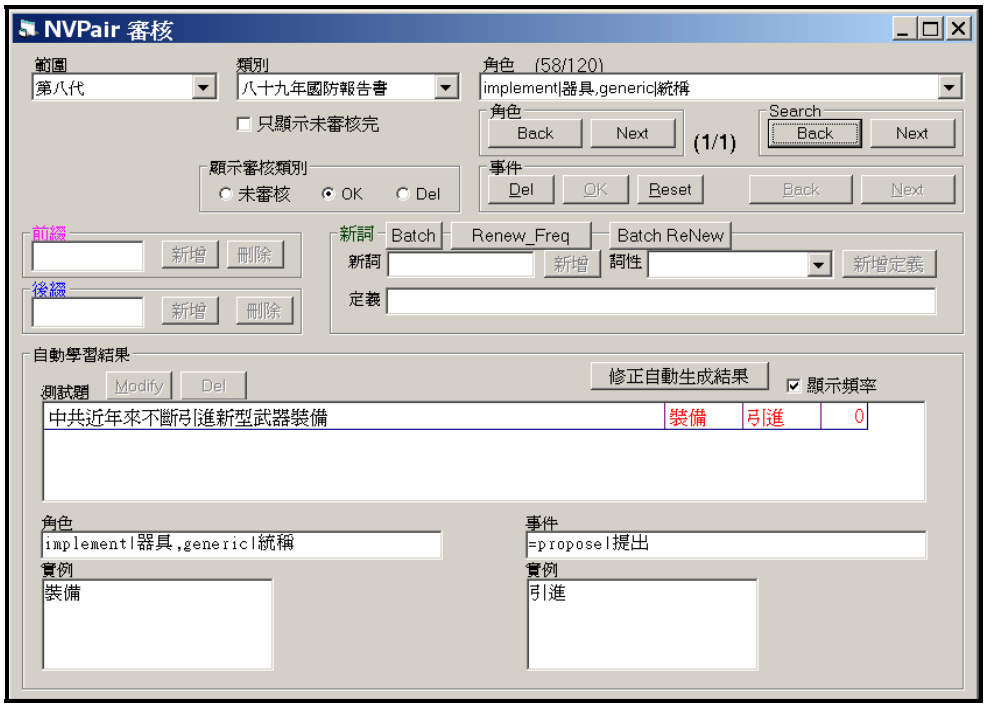
Appendix B. Example Mappings of FPOS and NV Word-Pairs

FPOS	NV word-pairs	Example, [] indicates nouns and <> indicates verbs
$N_1V_2ADJ_3N_4$	N_1V_2 & N_4V_2	[學生]<購買>許多[筆記本]
N_1V_2	N_1V_2	[雜草]<枯萎>
N_1 ADJ_2 ADV_3V_4	N_1V_4	[意願]遲未<回升>

Appendix C. Ten Examples of NVEF accepting Conditions

Noun-sense clas	Verb DEF	Example, [ ] indicates nouns and <> indicates verbs
微生物(bacteria)	own|有	已經使[細菌]<具有>高度抗藥性
位置類(location)	arrive|到達	若正好<蒞臨>[西班牙]
植物類(plant)	decline|衰敗	田中[雜草]<枯萎>
人工物(artifact)	buy|買	民眾不需要急著<購買>[米酒]
天然物(natural)	LeaveFor|前往	立刻驅船<前往>蘭嶼[海域]試竿
事件類(event)	alter|改變	批評這會<扭曲>[貿易]
精神類(mental)	BecomeMore|增多	民間投資[意願]遲未<回升>
現象類(phenomena)	announce|發表	做任何<公開>[承諾]
物形類(Shape)	be|是,all|全	由於從腰部以下<都是>合身[線條]
地點類(place)	from|相距	<距離>[小學]七百公尺

Appendix D. User Interface for Manually Confirming NVEF Knowledge



Computational Linguistics and Chinese Language Processing
Vol. 9, No.1 , February 2004, pp. 65-82

Mencius: A Chinese Named Entity Recognizer Using the Maximum Entropy-based Hybrid Model

Tzong-Han Tsai[*+], Shih-Hung Wu[+], Cheng-Wei Lee[+],

Cheng-Wei Shih[+], and Wen-Lian Hsu[+]

Abstract

This paper presents a Chinese named entity recognizer (NER): Mencius. It aims to address Chinese NER problems by combining the advantages of rule-based and machine learning (ML) based NER systems. Rule-based NER systems can explicitly encode human comprehension and can be tuned conveniently, while ML-based systems are robust, portable and inexpensive to develop. Our hybrid system incorporates a rule-based knowledge representation and template-matching tool, called InfoMap [Wu *et al*. 2002], into a maximum entropy (ME) framework. Named entities are represented in InfoMap as templates, which serve as ME features in Mencius. These features are edited manually, and their weights are estimated by the ME framework according to the training data. To understand how word segmentation might influence Chinese NER and the differences between a pure template-based method and our hybrid method, we configure Mencius using four distinct settings. The F-Measures of person names (PER), location names (LOC) and organization names (ORG) of the best configuration in our experiment were respectively 94.3%, 77.8% and 75.3%. From comparing the experiment results obtained using these configurations reveals that hybrid NER Systems always perform better performance in identifying person names. On the other hand, they have a little difficulty identifying location and organization names. Furthermore, using a word segmentation module improves the performance of pure Template-based NER Systems, but, it has little effect on hybrid NER systems.

[*] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
E-mail: d90013@csie.ntu.edu.tw
[+] Institute of Information Science, Academia Sinica., Taipei, Taiwan, R.O.C.
E-mail: {thtsai, shwu, aska, dapi, hsu}@iis.sinica.edu.tw

1. Introduction

Information Extraction (IE) is the task of extracting information of interest from unconstrained text. IE involves two main tasks: the recognition of named entities, and the recognition of the relationships among these named entities. Named Entity Recognition (NER) involves the identification of proper names in text and classification of them into different types of named entities (e.g., persons, organizations, locations). NER is important not only in IE [Grishman 2002] but also in lexical acquisition for the development of robust NLP systems [Coates-Stephens 1992]. Moreover, NER has proven useful for tasks such as document indexing and the maintenance of databases containing identified named entities.

During the last decade, NER has drawn much attention at Message Understanding Conferences (MUC) [Chinchor 1995a][Chinchor 1998a]. Both rule-based and machine learning NER systems have had some success. Traditional rule-based approaches have used manually constructed finite state patterns, which match text against a sequence of words. Such systems (like the University of Edinburgh's LTG [Mikheev *et al*. 1998]) do not need very much training data and can encode expert human knowledge. However, rule-based approaches lack robustness and portability. Each new source of text requires significant tweaking of the rules to maintain optimal performance, and the maintenance costs can be quite steep.

Another popular approach in NER is machine-learning (ML). ML is attractive in that it is more portable and less expensive to maintain. Representative ML approaches used in NER are HMM (BBN's IdentiFinder in [Miller *et al*. 1998][Bikel *et al*. 1999] and Maximum Entropy (ME) (New York University's MEME in [Borthwick *et al*. 1998][Borthwick 1999]). However, ML systems are relatively inexpensive to develop, and the outputs of these systems are difficult to interpret. In addition, it is difficult to improve the system performance through error analysis. The performance of an ML system can be very poor when the amount of training data is insufficient. Furthermore, the performance of ML systems is worse than that of rule-based ones by about 2%, as revealed at MUC-6 [Chinchor 1995b] and MUC-7 [Chinchor 1998b]. This might be due to the fact that current ML approaches can not capture non-parametric factors as effectively as human experts who handcraft the rules. Nonetheless, ML approaches do provide important statistical information that is unattainable by human experts. Currently, the F-measures of English rule-based and ML NER systems are in the range of 85% ~ 94%, based on MUC-7 data [Chinchor 1998c]. This is higher than the average performance of Chinese NER systems, which ranges from 79% to 86% [Chinchor 1998].

In this paper, we address the problem of Chinese NER. In Chinese sentences, there are no spaces between words, no capital letters to denote proper names, no sentence breaks, and, worst of all, no standard definition of “words.” As a result, word boundaries cannot, at times, be discerned without a context. In addition, the length of a named entity is longer on average than

that of an English one; thus, the complexity of a Chinese NER system is greater.

Previous works [Chen *et al*. 1998] [Yu *et al*. 1998] [Sun *et al*., 2002] on Chinese NER have relied on the word segmentation module. However, an error in the word segmentation step might lead to errors in NER results. Therefore, we want to compare the results of NER with/without performing word segmentation. Without word segmentation, a character-based tagger is used, which treats each character as a token and combines the tagged outcomes of contiguous characters to form an NER output. With word segmentation, we treat each word or character as a token, and combine the tagged outcomes of contiguous tokens to form an NER output.

Borthwick [1999] used an ME framework to integrate many NLP resources, including previous systems such as Proteus, a POS tagger. Mencius, the Chinese named entity recognizer presented here, incorporates a rule-based knowledge representation and a template-matching tool, called InfoMap [Wu *et al.* 2002], into a maximum entropy (ME) framework. Named entities are represented in InfoMap as templates, which serve as ME features in Mencius. These features are edited manually, and their weights are estimated by means of the ME framework according to the training data.

This paper is organized as follows. Section 2 provides the ME-based framework for NER. Section 3 describes features and how they are represented in our knowledge representation system, InfoMap. The data set and experimental results are discussed in section 4. Section 5 gives our conclusions and possible extensions of the current work.

2. Maximum Entropy-Based NER Framework

For our purpose, we regard each character as a token. Consider a test corpus and a set of n named entity categories. Since a named entity can have more than one token, we associate the following two tags with each category x: *x_begin* and *x_continue*. In addition, we use the tag *unknown* to indicate that a token is not part of a named entity. The NER problem can then be rephrased as the problem of assigning one of $2n + 1$ tags to each token. In Mencius, there are 3 named entity categories and 7 tags: *person_begin*, *person_continue*, *location_begin*, *location_continue*, *organization_begin*, *organization_continue* and *unknown*. For example, the phrase [李 遠 哲 在 高 雄 市] (Lee, Yuan Tseh in Kaohsiung City) could be tagged as *_begin*, [*person person_continue*, *person_continue*, *unknown*, *location_begin*, *location_continue*, *location_continue*].

2.1 Maximum Entropy

ME is a flexible statistical model which assigns an *outcome* for each token based on its *history*

and *features*. Outcome space is comprised of the seven Mencius tags for an ME formulation of NER. ME computes the probability $p(o|h)$ for any o from the space of all possible outcomes O, and for every h from the space of all possible histories H. A *history* is composed of all the conditioning data that enable one to assign probabilities to the space of outcomes. In NER, *history* can be viewed as consisting of the all information derivable from the test corpus relavant to the current token.

The computation of $p(o|h)$ in ME depends on a set of binary-valued *features*, which are helpful in making a prediction about the outcome. For instance, one of our features is as follows: when the current character is a known surname, it is likely to be the leading character of a person name. More formally, we can represent this feature as

$$f(h,o) = \begin{cases} 1 : \text{if Current - Char - Surname(h)} = \text{true and } o = person\_begin \\ 0 : \text{else} \end{cases} \quad (1)$$

Here, *Current-Char-Surname(h)* is a binary function that returns the value *true* if the *current character* of the history h is in the surname list.

Given a set of features and a training corpus, the ME estimation process produces a model in which every feature f_i has a weight α_i. This allows us to compute the conditional probability as follows [Berger *et al*. 1996]:

$$p(o \mid h) = \frac{1}{Z(h)} \prod_i \alpha_i^{f_i(h,o)} . \quad (2)$$

Intuitively, the probability is the multiplication of the weights of active features (i.e., those $f_i(h,o) = 1$). The weight α_i is estimated by means of a procedure called Generalized Iterative Scaling (GIS) [Darroch *et al*. 1972]. This is an iterative method that improves estimation of the weights at each iteration. The ME estimation technique guarantees that for every feature f_i, the expected value of α_i equals the empirical expectation of α_i in the training corpus.

As Borthwick [1999] remarked, ME allows the modeler to concentrate on finding the features that characterize the problem while letting the ME estimation routine deal with assigning relative weights to the features.

2.2 Decoding

After an ME model has been trained and the proper weight α_i has been assigned to each feature f_i, decoding (i.e., *marking up*) a new piece of text becomes a simple task. First, Mencius tokenizes the text and preprocesses the testing sentence. Then for each token, it checks which

features are active and combines the α_i of the active features according to equation 2. Finally, a Viterbi search is run to find the highest probability path through the lattice of conditional probabilities that does not produce any invalid tag sequences (for instance, the sequence [*person_begin*, *location_continue*] is invalid). Further details on Viterbi search can be found in [Viterbi 1967].

3. Features

We divide features that can be used to recognize named entities into four categories according to whether they are external or not and whether they are category dependent or not. McDonald defined internal and external features in [McDonald 1996]. Internal evidence is found within the entity, while external evidence is gathered from its context. We use category-independent features to distinguish named entities from non-named entities (e.g., first-character-of--a-sentence, capital-letter, out-of-vocabulary), and use category-dependent features to distinguish between different named entity categories (for example, surname and given name lists are used to recognize person names). However, to simplify our design, we only use internal features that are category-dependent in this paper.

3.1 InfoMap – Our Knowledge Representation System

To the calculate values of location features and organization features, Mencius uses InfoMap. InfoMap is our knowledge representation and template matching tool, which represents location or organization names as templates. An input string (sentence) is first matched to one or more location or organization templates by InfoMap and then passed to Mencius; there, it is assigned feature values which further distinguish which named entity category it falls into.

3.1.1 Knowledge Representation Scheme in InfoMap

InfoMap is a hierarchical knowledge representation scheme, consisting of several domains, each with a tree-like taxonomy. The basic units of information in InfoMap are called generic nodes, which represent concepts, and function nodes, which represent the relationships among the generic nodes of one specific domain. In addition, generic nodes can also contain cross references to other nodes to avoid needless repetition.

In Mencius, we apply the geographical taxonomy of InfoMap called GeoMap. Our location and organization templates refer to generic nodes in Geomap. As shown in Figure 1, GeoMap has three sub-domains: World, Mainland China, and Taiwan. Under the sub-domain Taiwan, there are four attributes: Cities, Parks, Counties and City Districts. Moreover, these attributes can be further divided; for example, Counties can be divided into individual counties:

Taipei County, Taoyuan County, etc. In InfoMap, we refer to generic nodes (or concept node) by means of paths. A path of generic nodes consists of all the node names from the root of the domain to the specific generic node, where function nodes are omitted. The node names are separated by periods. For example, the path for the "Taipei County" node is "GeoMap.Counties.Taipei County."

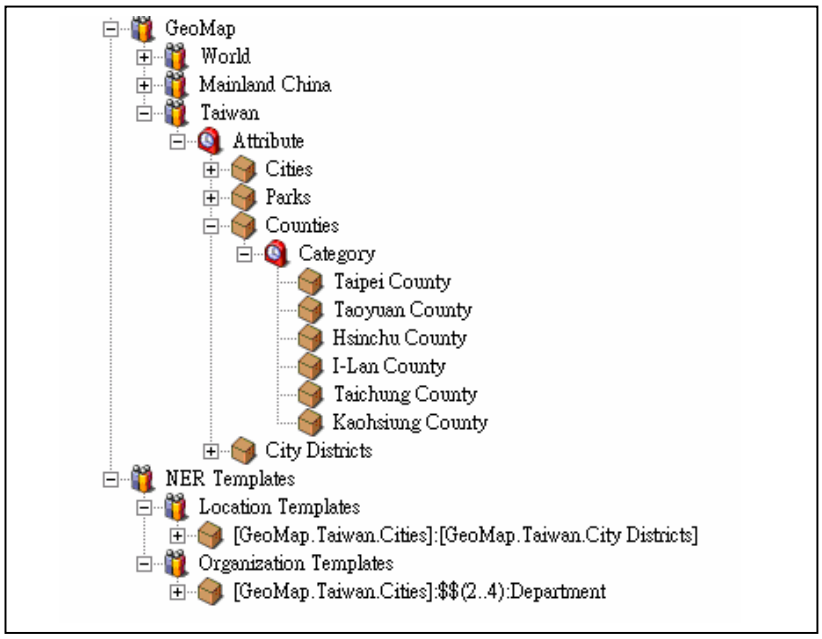


Figure 1. A partial view of GeoMap.

3.1.2 InfoMap Templates

In InfoMap, text templates are stored in generic nodes. Templates can consist of character strings, wildcards (see $$ in Table 1), and references to other generic nodes in InfoMap. For example, the template [通用地理.台灣.縣]:$$(2..4):局 ([GeoMap. Taiwan. Counties]: $$(2..4):Department) can be used to recognize county level governmental departments in Taiwan. The syntax used in InfoMap templates are shown in Table 1. The first part of our sample template shown above (enclosed by "[]") is a path that refers to the generic node "Counties." The second element is a wildcard, ($$) which must be 2 to 4 characters in length. The third element is a specified character "局" (Department).

Table 1. InfoMap template syntax.

Symbol	Semantics	Example Template	Sample Matching String
:	Concatenate two strings	A:B	AB
$$(m..n)	Wildcards (the number of characters can be from m to n; both m and n have to be non-negative integers)	A:$$(1..2):B	ACB, ADDB, ACDB
[p]	A path to a generic node	[GeoMap.Taiwan.Counties]	Taipei County, Taoyuan County, Hsinchu County, etc.

3.2 Category-Dependent Internal Features

Recall that category-dependent features are used to distinguish among different named entity categories.

3.2.1 Features for Recognizing Person Names

Mencius only deals with a surname plus a first name (usually composed of two characters), for example, 陳水扁 (Chen Shui-bian). There are various other ways to identify a person in a sentence, such as 陳先生 (Mr. Chen) and老陳 (Old Chen), which have not been incorporated into the current system. Furthermore, we do not target transliterated names, such as 布希 (Bush), since they do not follow Chinese name composition rules. We use a table of frequently occurring names to process our candidate test data. If a character and its context (history) correspond to a feature condition, the value of the current character for that feature will be set to 1. Feature conditions, examples and explanations for each feature are shown in Table 2. In the feature condition column, c_{-1}, c_0, and c_1 represent the preceding character, the current character, and the following character, respectively.

Current-Char-Person-Surname: This feature is set to 1 if $c_0c_1c_2$ or c_0c_1 is in the person name database. For example, in the case of $c_0c_1c_2$ = 陳水扁, the feature Current-Char-Person-Surname for 陳 is active since c_0 and its following characters c_1c_2 satisfy the feature condition.

Current-Char-Person-Given-Name: This feature is set to 1 if $c_{-2}c_{-1}c_0$, $c_{-1}c_0$, or $c_{-1}c_0c_1$ is in the person name database.

Current-Char-Surname: This feature is set to 1 if c_0 is in the top 300 popular surname list.

Table 2. Person features.

Feature	Feature Conditions	Example	Explanation
Current-Char-Person-Surname	$c_0c_1c_2$ or c_0c_1 is in the name list	"陳"水扁, "連"戰	Probably the first character of a person name
Current-Char-Person-Given-Name	$c_{-2}c_{-1}c_0$ or $c_{-1}c_0$ or $c_{-1}c_0c_1$ is in the name list	陳"水"扁, 陳水"扁", 連"戰"	Probably the second or third character of a person name
Current-Char-Surname	c_0 is in the surname list	"陳", "林", "李"	Probably a surname
Current-Char-Given-Name	c_0c_1 or $c_{-1}c_0$ is in the given name list	黃"其"聖, 黃其"聖"	Probably part of a popular given name
Current-Char-Freq-Given-Name-Character	Both c_0, c_1 or c_{-1}, c_1 is in the frequent given name character list	羅"方"全, 羅方"全"	Probably a given name character
Current-Char-Speaking-Verb	c_0 or c_0c_1 or $c_{-1}c_0$ is in the list of verbs indicating speech	"說", "表"示, 表"示"	Probably part of a verb indicating speech (ex: John said he was tired)
Current-Char-Title	c_0 or c_0c_1 or $c_{-1}c_0$ is in the title list	"先"生, 先"生"	Probably part of a title

Current-Char-Given-Name: This feature is set to 1 if c_0c_1 or $c_{-1}c_0$ is in the given name database.

Current-Char-Freq-Given-Name-Character: (c_0 and c_1) or (c_{-1} and c_0) is in the frequently given name character list

Current-Char-Speaking-Verb: c_0 or c_0c_1 or $c_{-1}c_0$ is in the speaking verb list. This feature distinguishes a trigram containing a speaking verb, such as 陳沖說 (Chen Chong said), from a real person name.

Current-Char-Title: c_0 or c_0c_1 or $c_{-1}c_0$ is in the title list. This feature distinguishes a trigram containing a title, such as 陳先生 (Mr. Chen), from a real person name.

3.2.2 Features for Recognizing Location Names

In general, locations are divided into four types: administrative division, public area (park, airport, or port), landmark (road, road section, cross section or address), and landform (mountain, river, sea, or ocean). An administrative division name usually contains one or more

location names in a hierarchical order, such as 安大略省多倫多市 (Toronto, Ontario). A public area name is composed of a Region-Name and a Place-Name. However, the Region-Name is usually omitted from news content if it was previously mentioned. For example, 倫敦海德公園 (Hyde Park, London) contains the Region-Name 倫敦 (London) and the Place-Name 海德公園 (Hyde Park). But "Hyde Park, London" is usually abbreviated as "Hyde Park" within a report. The same rule can be applied to landmark names. A landmark name includes a Region-Name and a Position-Name. In a news article, the Region-Name can be omitted if the Place-Name has been mentioned previously. For example, 溫哥華市羅伯遜街五號 (No. 5, Robson St., Vancouver City) will be stated as 羅伯遜街五號 (No. 5, Robson St.) later in the report.

In Mencius, we build templates to recognize three types of location names. Our administrative division templates contain more than one set of location names in a hierarchical order. For example, the template, [通用地理.台灣.市]:[通用地理.台灣.各市行政區] ([GeoMap.Taiwan.Cities]:[GeoMap.Taiwan.City Districts]) can be used to recognize all city districts in Taiwan. In addition, public area templates contain one set of location names and a set of Place-Name. For example, [通用地理.台灣.市]:[通用地理.台灣.公園] ([GeoMap.Taiwan.Cities]:[GeoMap.Taiwan.Parks]) can be used to recognize all city parks in Taiwan. Landmark templates are built in the same way.For example, [通用地理.台灣.市]:$$(2..4):路 ([GeoMap.Taiwan.Cities]:$$(2..4):Road) can be used to recognize roads in Taiwan.

Two features are associated with each InfoMap template category x (e.g., location and organization). The first is Current-Char-InfoMap-x-Begin, which is set to 1 for the first character of a matched string and set to 0 for the remaining characters. The other is Current-Char-InfoMap-x-Continue, which is set to 1 for all the characters of matched string except for the first character and set to 0 for the first character. The intuition behind this is as follows: InfoMap can be used to help ME detect which character in a sentence is the first character of the location name and which characters are the remaining characters of a location name. That is, Current-Char-InfoMap-x-Begin is helpful for determining which character should be tagged as *x_begin*, while Current-Char-InfoMap-x-Continue is helpful for determining which character should be tagged as *x_continue* if we build an InfoMap template for that category x. The two features associated with x category are shown below:

$$f(h,o)=\begin{cases}1:\text{if Current-Char-InfoMap-x-Begin = true and } o = x\_begin\\ 0:\text{else}\end{cases} \tag{3}$$

$$f(h,o)=\begin{cases}1:\text{if Current-Char-InfoMap-x-Continue= true and } o = x\_continue\\ 0:\text{else}\end{cases} \tag{4}$$

When recognizing a location name in a sentence, we test if any location templates match the sentence. If several matched templates overlap, we select the longest matched one. As mentioned above, the feature Current-Character-InfoMap-Location-Begin of the first character of the matched string is set to 1 while the feature Current-Character-InfoMap-Location--Continue of the remaining characters of the matched string is set to 1. Table 3 shows the necessary conditions for each organization feature and gives examples of matched data.

Table 3. Location features.

Feature	Feature Conditions	Example	Explanations
Current-Char-InfoMap-Location-Begin	c_0~c_{n-1} matches an InfoMap location template, where the character length of the template is n	“台”北縣板橋市	Probably the leading character of a location name.
Current-Char-InfoMap-Location-Continue	c_a…c_0….c_b matches an InfoMap location template, where a is a negative integer and b is a non-negative integer	台”北”縣板橋市	Probably a continuing character of a location name.

3.2.3 Features for Recognizing Organization Names

Organizations include named corporate, governmental, or other organizational entities. The difficulty in recognizing an organization name is that it usually begins with a location name, such as 台北市地檢署 (Taipei District Public Prosecutors Office). Therefore, traditional machine learning NER systems can only identify the location part rather than the full organization name. For example, the system only extracts 台北市 (Taipei City) from 台北市 SOGO 百貨週年慶 (Taipei SOGO Department Store Anniversary) rather than 台北市 SOGO 百貨 (Taipei SOGO Department Store). According to our analysis of the structure of Chinese organization names, they mostly end with a specific keyword or begin with a location name. Therefore, we use those keywords and location names as the boundary markers of organization names. Based on our observation, we categorize organization names into four types according to their boundary markers.

Type I: With left and right boundary markers

The organization names in this category begin with by one or more geographical names and

ended by an organization keyword. For example, 台北市 (Taipei City) is the left boundary marker of 台北市捷運公司 (Taipei City Rapid Transit Corporation), while an organization keyword, 公司 (Corporation), is the right boundary marker.

Type II: With a left boundary marker

The organization names in this category begin with by one or more than one geographical names, but the organization keyword (e.g., 公司 (Corporation)) is omitted. For example, 台灣捷安特 (Giant Taiwan) only contains the left boundary 台灣 (Taiwan).

Type III: With a right boundary marker

The organization names in this category end with an organization keyword. For example, 捷安特公司 (Giant Corporation) only contains the right boundary 公司 (Corporation).

Table 4. Organization features.

Feature	Feature Conditions	Example	Explanations
Current-Char-InfoMap-Organization-Begin	c_0~c_{n-1} matches an InfoMap organization template, where the character length of the template is n	"台"北市捷運公司	Probably the leading character of an organization name
Current-Char-InfoMap-Organization-Continue	$c_a \ldots c_0 \ldots c_b$ matches an InfoMap organization template, where a is a negative integer and b is a non-negative integer	台"北"市捷運公司	Probably the leading character of an organization name
Current-Char-Organization-Keyword	c_0 or c_0c_1 or $c_{-1}c_0$ is in the organization keyword list	"公"司, 公 "司"	Probably part of an organization keyword

Type IV: No boundary marker

In this category, both left and right boundaries as above mentioned are omitted, for example, 捷安特 (Giant). The organization names in this category are usually in abbreviated form.

In Mencius, we build templates for recognizing Type I organization names. Each organization template begins with a location name in GeoMap and ends with an organization keyword. For example, we can build [通用地理.台灣.市]:$$(2..4):局([GeoMap.Taiwan.Cities]:

$$(2..4):Department) to recognize county level government departments in Taiwan. However, in Types II, III, and IV, organization names cannot be recognized by templates. Therefore, the maximum entropy model uses features of characters (from c_{-2} to c_2), tags (from t_{-2} to t_2), and organization keywords, e.g., 公司 (Corporation), to find the most likely tag sequences and recognize them.

When a string matches an organization template, the feature Current-Character- InfoMap--Organization-Start of the first character is set to 1. In addition, the feature Current--Character-InfoMap-Organization-Continue of the remaining characters is set to 1. The necessary conditions for each organization feature and examples of matched data are shown in Table 4. These features are helpful for recognizing organization names.

4. Experiments

4.1 Data Sets

For Chinese NER, the most famous corpus is MET-2 [6]. There are two main differences between our corpus and MET-2: the number of domains and the amount of data. First, MET-2 contains only one domain (Accident), while our corpus, which was collected from the online United Daily News in December 2002 (http://www.udn.com.tw), contains six domains: Local News, Social Affairs, Investment, Politics, Headline News and Business, which provide a greater variety of organization names than a single domain corpus can. The full location names and organization names are comparatively longer, and our corpus contains more location names and addresses at the county level. Therefore, the patterns of location names and organization names are more complex in our corpus.

Secondly, our corpus is much larger than MET2, which contains 174 Chinese PER, 750 LOC, and 377 ORG. Our corpus contains 1,242 Chinese PER, 954 LOC, and 1,147 ORG in 10,000 sentences (about 126,872 Chinese characters). The statistics of our data are shown in Table 5.

Table 5. Statistics of the data Set

Domain	Number of Named Entities			Size (in characters)
	PER	LOC	ORG	
Local News	84	139	97	11835
Social Affairs	310	287	354	37719
Investment	20	63	33	14397
Politics	419	209	233	17168
Headline News	267	70	243	19938
Business	142	186	187	25815
Total	1242	954	1147	126872

4.2 Experimental Results

To understand how word segmentation might influence Chinese NER and the differences between a pure template-based method and our hybrid method, we configure Mencius using the following four settings: (1) Template-based with Char-based Tokenization (TC), (2) Template-based with Word-based Tokenization (TW), (3) Hybrid with Char-based Tokenization (HC), and (4) Hybrid with Word-based Tokenization (HW). Following the standard 10-fold cross-validation method, we tested Mencius with each configuration using the data set mentioned in section 4.1. The following subsections provide details about each configuration and the results obtained.

4.2.1 Template-based with Char-based Tokenization (TC)

In this experiment, we regarded each character as a token, and used a person name list and InfoMap templates to recognize all named entities. The number of lexicons in the person name lists and gazetteers was 32000. As shown in Table 6, the obtained F-Measures of PER, LOC and ORG were 76.2%, 75.4% and 75.1%, respectively.

Table 6. Performance of the Template-based System with Char-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	64.77	92.59	76.22
LOC	76.41	74.42	75.40
ORG	85.60	66.93	75.12
Total	72.95	78.62	75.67

4.2.2 Template-based with Word-based Tokenization (TW)

In this experiment, we used a word segmentation module based on the 100,000-word CKIP Traditional Chinese dictionary to split sentences into tokens. This module combines forward and backward longest matching algorithms in the following way: if the segmentation results of the two algorithms agree in certain substrings, this module outputs tokens in those substrings. While in the part which the segmentation results of the two algorithms differ, this module skips word tokens and only outputs character tokens. In the previous test, 98% of the word tokens were valid words. Then, we used person name lists and InfoMap templates to recognize all the named entities. The number of lexicons in the person name lists and gazetteers was 32,000. As shown in Table 6, the obtained F-Measures of PER, LOC and ORG were 89.0%, 74.1% and 71.6%, respectively.

Table 7. Performance of the Template-based System with Word-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	88.69	89.32	89.00
LOC	76.92	71.44	74.08
ORG	85.66	61.44	71.55
Total	84.14	74.70	79.14

4.2.3 Hybrid with Char-based Tokenization (HC)

In this experiment, we regarded each character as a token without performing any word segmentation. We then integrated person name lists, location templates, and organization templates into a Maximum-Entropy-Based framework. As shown in Table 8, the obtained F-Measures of PER, LOC and ORG were 94.3%, 77.8% and 75.3%, respectively.

Table 8. Performance of the Hybrid System with Char-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	96.97	91.71	94.27
LOC	80.96	74.81	77.76
ORG	87.16	66.22	75.26
Total	89.05	78.18	83.26

4.2.4 Hybrid System with Word-based Tokenization (HW)

In this experiment, we used the same word segmentation module described in section 4.2.2 to split sentences into tokens. Then, we integrated person name lists, location templates, and organization templates into a Maximum-Entropy-Based framework. As shown in Table 9, the obtained F-Measures of PER, LOC and ORG were 95.9%, 73.4% and 76.1%, respectively.

Table 9. Performance of the Hybrid System with Word-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	98.74	93.31	95.94
LOC	81.46	66.73	73.36
ORG	87.54	67.29	76.09
Total	90.33	76.66	82.93

4.2.5 Comparisons

TC versus TW

We observed that TW achieved much higher precision than TC in PER. When word segmentation is not performed, some trigrams and quadgrams may falsely appear to be person names. Take the sentence “新古典主義” for example. TC would extract “古典主” as a person

name since "古典主" matches our family-name trigram template. However, in TW, thanks to word segmentation, "古典" and "主義" would be marked as tokens first and would not match the family-name trigram template.

HC versus HW

We observed that HW achieved similar precision to that of HC in all three NE categories. HW also achieved recall rates similar to those achieved by HC with PER and ORG NEs. In the case of PER NEs, this is because the length of person names is 2 to 4 characters. Therefore, a five-character long window (-2 to +2) is sufficient to recognize a person name. As far as recognizing LOC NEs is concerned, HW's recall rate was worse than HC's. This is because the word segmentation module marks occupational titles as tokens, for example: "台北市長". HW cannot extract the LOC NE "台北市" from "台北市長" because it has already been defined as a token. To recognize LOC and ORG NEs, we need higher-level features and more external features. Since Mencius lacks these kinds of features, HW doesn't achieve significantly better performance than HC.

TC versus HC

We observed that in PER, HC achieved much higher precision than TC, while in LOC and ORG, HC performed slightly better than TC. This is because most of the key features for identifying a person name are close to the person name, or inside the personal name. Take the sentence "立即連絡海鷗直升機" as an example; when we wish to determine whether "連絡海" is a person name, we can see that "立即" seldom appears before a person name, and that "鷗" seldom appears after a person name. In HC, ME can use this information to determine that "連絡海" is not a person name, but to recognize a location name and an organization name, we need wider context and features, such as sentence analysis or shallow parsing. Take "如馬公、七美、望安、蘭嶼、綠島、馬祖和金門等離島爲管制航線" as an example; the two preceding characters are "美"and "、", and the two following characters are "、"and "蘭". ME cannot use this information to identify a location name.

TW versus HW

We observed that HW achieved better precision than TW in identifying personal names. This is because in HW, ME can use context information to filter some trigrams and 4 grams, which are not personal names. Take "王 金 平 和 其他 委員" as an example; it matches the double-family-name quadgram template because "王" and "金" are both family names. However, "王金平" is the correct person name. In HW, ME can use the information that "王金平" has appeared in the training corpus and been tagged as a PER NE to identify the person name "王金平" in a sentence. We also observed that HW achieved better recall than TW in identifying person names. This is because in HW, ME can use the information that bigram

personal names are tagged as PER NEs from the training data, but TW cannot because we don't have bigram-person-name templates. In addition, some person names are in the dictionary, so some tokens are person names. Take "陳建仁 的 作爲" as an example. Although the token "陳建仁" cannot match any person name template, in HW, ME can use context information and training data to recognize "陳建仁". To identify location names, ME needs a wider context to detect location names, so HW's recall is worse than TW's. However, ME can filter out some unreasonable trigrams, such as "黃榮村", because it matches a location name template $$(2..3): 村, which represents a village in Taiwan. Therefore, ME achieves bigger precision in identifying location names.

5. Conclusions

In this paper, we have presented a Chinese NER system, called Mencius. We configured Mencius according to the following settings of to analyze the effects of using a Maximum Entropy-based Framework and a word segmentation module: (1) Template-based with Char-based Tokenization (TC), (2) Template-based with Word-based Tokenization (TW), (3) Hybrid with Char-based Tokenization (HC), and (4) Hybrid with Word-based Tokenization (HW). The experimental results showed that whether a character or a word was taken as a token, the hybrid NER System always performed better in identifying person names. However,this had little effect on the identification of location and organization names. This is because the context information around a location name or an organization name is more complex than that around a person name. In addition, using a word segmentation module improved the performance of the pure Template-based NER System. However, it had little effect with the hybrid NER systems. The current version of Mencius lacks sentence parsing templates and shallow parsing tools to handle such complex information. We will add these functions in the future.

References

Berger, A., Della Pietra, S. A., and Della Pietra, V. J., "A maximum entropy approach to natural language processing," Computational Linguistics, vol. 22, pp. 39-71, 1996.

Bikel, D., Schwartz, R., and Weischedel, R., "An Algorithm that Learns What's in a Name," Machine Learning, 1999.

Borthwick, A., Sterling J., Agichtein, E., and Grishman, R., "NYU: Description of the MENE Named Entity System as Used in MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Borthwick, A., "A Maximum Entropy Approach to Named Entity Recognition," New York University, 1999.

Chinchor, N., "MUC-6 Named Entity Task Definition (Version 2.1)," presented at the 6th Message Understanding Conference, Columbia, Maryland, 1995.

Chinchor, N., "Statistical Significance of MUC-6 Results," presented at the 6th Message Understanding Conference, Columbia, Maryland, 1995.

Chinchor, N., "MUC-7 Named Entity Task Definition (Version 3.5)," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Chinchor, N., "Statistical Significance of MUC-7 Results," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Chinchor, N., "MUC-7 Test Score Reports for all Participants and all Tasks," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Chen, H. H., Ding, Y. W., Tsai, S. C., and Bian, G. W., "Description of the NTU System Used for MET2," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Coates-Stephens, S., "The Analysis and Acquisition of Proper Names for Robust Text Understanding," in Dept. of Computer Science. London: City University, 1992.

Darroch, J. N. and Ratcliff, D., "Generalized iterative scaling for log-linear models," Annals of Mathematicl Statistics, vol. 43, pp. 1470-1480, 1972.

Grishman, R., "Information Extraction: Techniques and Challenges," in Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, J. G. Carbonell, Ed. Frascati, Italy: Springer, 1997, pp. 10-26.

McDonald, D., "Internal and External Evidence in the Identification and Semantic Categorization of Proper Names," in Corpus Processing for Lexical Acquisition, J. Pustejovsky, Ed. Cambridge, MA: MIT Press, 1996, pp. 21-39.

Mikheev, A., Grover, C., and Moensk, M., "Description of the LTG System Used for MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., and Weischedel, R., "BBN: Description of the SIFT System as Used for MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Sun, J., Gao, J. F., Zhang, L., Zhou, M., and Huang, C. N., "Chinese Named Entity Identification Using Class-based Language Model," presented at the 19th International Conference on Computational Linguistics,, 2002.

Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, vol. IT, pp. 260-269, 1967.

Wu, S. H., Day, M. Y., Tsai, T. H., and Hsu, W. L., "FAQ-centered Organizational Memory," in Knowledge Management and Organizational Memories, R. Dieng-Kuntz, Ed. Boston: Kluwer Academic Publishers, 2002.

Yu, S. H., Bai, S. H., and Wu, P., "Description of the Kent Ridge Digital Labs System Used for MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 1 , February 2004, pp. 83-96

Reliable and Cost-Effective Pos-Tagging

Yu-Fang Tsai*, and Keh-Jiann Chen*

Abstract

In order to achieve fast, high quality Part-of-speech (pos) tagging, algorithms should achieve high accuracy and require less manually proofreading. This study aimed to achieve these goals by defining a new criterion of tagging reliability, the estimated final accuracy of the tagging under a fixed amount of proofreading, to be used to judge how cost-effective a tagging algorithm is. In this paper, we also propose a new tagging algorithm, called the context-rule model, to achieve cost-effective tagging. The context rule model utilizes broad context information to improve tagging accuracy. In experiments, we compared the tagging accuracy and reliability of the context-rule model, Markov bi-gram model and word-dependent Markov bi-gram model. The result showed that the context-rule model outperformed both Markov models. Comparing the models based on tagging accuracy, the context-rule model reduced the number of errors 20% more than the other two Markov models did. For the best cost-effective tagging algorithm to achieve 99% tagging accuracy, it was estimated that, on average, 20% of the samples of ambiguous words needed to be rechecked. We also compared tradeoff between the amount of proofreading needed and final accuracy for the different algorithms. It turns out that an algorithm with the highest accuracy may not always be the most reliable algorithm.

Keywords: part-of-speech tagging, corpus, reliability, ambiguous resolution

1. Introduction

Part-of-speech (pos) tagging for a large corpus is a labor intensive and time-consuming task. Most tagging algorithms try to achieve high accuracy, but 100% accuracy is an impossible goal. Even after tremendous amounts of time and labor are spent on the post-process of proofreading, many errors still exist in publicly available tagged corpora. Therefore, in order to achieve fast, high quality pos tagging, tagging algorithms should not only achieve high accuracy but also require less manually proofreading. In this paper, we propose a context-rule

* Institute of Information Science, Academia Sinica
128 Academia Rd. Sec.2, Nankang, Taipei, Taiwan E-mail: {eddie,kchen}@iis.sinica.edu.tw

model to achieve both goals.

The first goal is to improve tagging accuracy. According to our observation, the pos tagging of a word depends on its context but not simply on its context category. Therefore, the proposed context-rule model utilizes a broad scope of context information to perform pos tagging of a word. Rich context information helps to improve the model coverage rate and tagging accuracy. The context-rule model will be described in more detail later in this paper. Our second goal is to reduce the manual editing effort. A new concept of reliable tagging is proposed. The idea is as follows. An evaluation score is assigned to each tagging decision as an indicator of tagging confidence. If a high confidence value is achieved, it indicates that the tagging decision is very likely correct. On the other hand, a low confidence value means that the tagging decision requires manual checking. If a tagging algorithm can achieve a high degree of reliability in evaluation, this means that most of the high confidence tagging results need not manually rechecked. As a result, the time and manual efforts required in the tagging process can be drastically reduced. The reliability of a tagging algorithm is defined as follows:

Reliability = The estimated final accuracy achieved by the tagging model under the constraint that only a fixed number of target words with the lowest confidence values are manually proofread.

The notion of tagging reliability is slightly different from the notion of tagging accuracy since high accurate algorithm may require more manual proofreading than a reliable algorithm that achieves lower accuracy.

The rest of this paper is organized as follows. In section 2, the relation between reliability and accuracy is discussed. In section 3, three different tagging algorithms, the Markov pos bi-gram model, word-dependent Markov bi-gram model, and context-rule model, are discussed. In section 4, the three algorithms are compared based on tagging accuracy. In addition, confidence measures of tagging results are defined, and the most cost-effective algorithm is determined. Conclusions are drawn on section 5.

2. Reliability vs. Accuracy

The reported accuracy of automatic tagging algorithms ranges from about 95% to 96% [Chang *et al.*, 1993; Lua, 1996; Liu *et al.*, 1995]. If we can pinpoint errors, then only 4~5% of the target corpus has to be revised to achieve 100% accuracy. However, since the errors are not identified, conventionally, the whole corpus has to be re-examined. This is most tedious and time consuming since a practically useful tagged corpus is at least several million words in size. In order to reduce the amount manual editing required and speed up the process of constructing a large tagged corpus, only potential tagging errors should be rechecked manually [Kveton *et al.*, 2002; Nakagawa *et al.*, 2002]. The problem is how to find the

potential errors.

Suppose that a probabilistic-based tagging method assigns a probability to each pos of a target word by investigating the context of this target word *w*. The hypothesis is that if the probability $P(c_1 \mid w, context)$ of the top choice candidate c_1 is much higher than the probability $P(c_2 \mid w, context)$ of the second choice candidate c_2, then the confidence value assigned to c_1 will also be higher. (Hereafter, for the purpose of simplification, we will use $P(c)$ to stand for $P(c \mid w, context)$, if without confusing.) Likewise, if the probability $P(c_1)$ is close to the probability $P(c_2)$, then the confidence value assigned to c_1 will also be lower. We aim to prove the above hypothesis by using empirical methods. For each different tagging method, we define its confidence measure according to the above hypothesis and examine whether tagging errors are likely to occur for words with low tagging confidence. If the hypothesis is true, we can proofread among the auto-tagged results only those words with low confidence values. Furthermore, the final accuracy of the tagging process after partial proofreading is done can also be estimated based on the accuracy of the tagging algorithm and the number of errors contained in the proofread data. For instance, suppose that a system has a tagging accuracy of 94%, and that K% of the target words with the lowest confidence scores covers 80% of the errors. After those K% of tagged words are proofread, 80% of the errors are fixed. Therefore, the reliability score of this tagging system of K% proofread words will be 1 - (error rate) * (reduced error rate) = 1 - ((1 - accuracy rate) * 20%) = 1 - ((1 - 94%) * 20%) = 98.8%. On the other hand, suppose that another tagging system has a higher tagging accuracy of 96%, but that its confidence measure is not very high, such that K% of the words with the lowest confidence scores contains only 50% of the errors. Then the reliability of this system is 1 - ((1 - 96%) * 50%) = 98%, which is lower than that of the first system. That is to say, after expending the same amount of effort on manual proofreading, the first system achieves better results even though it has lower tagging accuracy. In other words, a reliable system is more cost-effective.

3. Tagging Algorithms and Confidence Measures

In this paper, we will evaluate three different tagging algorithms based on the same training and testing data, compare them based on tgging accuracy, and determine the most reliable tagging algorithm among them. The three tagging algorithms are the Markov bi-gram model, word-dependent Markov model, and context-rule model. The training data and testing data were extracted from the Sinica corpus, a 5 million word balanced Chinese corpus with pos tagging [Chen *et al.*, 1996]. The confidence measure was defined for each algorithm, and the final accuracy was estimated with the constraint that only a fixed amount of testing data needed to be proofread.

Table 1. Sample keyword-in-context file of the words '研究' sorted according to its left/right context.

的(DE)	重要(VH)	**研究(Nv)**	機構(Na)	之(DE)
相當(Dfa)	重視(VJ)	**研究(Nv)**	開發(Nv)	,(COMMACATEGORY)
內(Ncd)	重點(Na)	**研究(Nv)**	需求(Na)	。(PERIODCATEGORY)
仍(D)	限於(VJ)	**研究(Nv)**	階段(Na)	。(PERIODCATEGORY)
民族(Na)	音樂(Na)	**研究(VE)**	者(Na)	明立國(Nb)
赴(VCL)	香港(Nc)	**研究(VE)**	該(Nes)	地(Na)
亦(D)	值得(VH)	**研究(VE)**	。(PERIODCATEGORY)	
合宜性(Na)	值得(VH)	**研究(VE)**	。(PERIODCATEGORY)	
更(D)	值得(VH)	**研究(Nv)**	。(PERIODCATEGORY)	

It is easier to proofread and obtain consistent tagging results if proofreading is done by checking each ambiguous word in its keyword-in-context file. For instance, in Table 1, the keyword-in-context file of the word '研究' (research), which has pos of verb type *VE* and noun type *Nv*, is sorted according to its left/right context. Proofreaders can take the other examples as references to determine whether tagging results are correct. If all of the occurrences of ambiguous words had to be rechecked, this would require too much work. Therefore, only words with low confidence scores will be rechecked.

A general confidence measure can be defined as $\frac{P(c_1)}{P(c_1) + P(c_2)}$, where $P(c_1)$ is the the probability of the top choice pos c_1 assigned by the tagging algorithm and $P(c_2)$ is the probability of the second choice pos c_2 [1]. The common terms used in the following tagging algorithms discussed below are defined as follows:

w_k	the k-th word in a sequence;
c_k	the pos associated with the k-th word w_k;
$w_1c_1,...,w_nc_n$	a word sequence containing n words with their associated categories.

3.1 Markov Bi-gram Model

The most widely used tagging models are the part-of-speech n-gram models, in particular, the

[1] The log-likelihood ratio of $\log(P(c_1)/P(c_2))$ is an alternative confidence measure. However, some tagging algorithms, such as context-rule model, may not necessary produce a real probability estimation for each pos. Scaling control for the log-likelihood ratio will be hard for those algorithms to achieve. In addition, the range of our confidence score is 0.5 ~ 1.0 and it is thus easier to evaluate different tagging algorithms. Therefore, the above confidence value is adopted.

bi-gram and tri-gram models. A bi-gram model looks at pairs of categories (or words) and uses the conditional probability of $P(c_k \mid c_{k-1})$. The Markov assumption is that the probability of a pos occurring depends only on the pos before it.

Given a word sequence $w_1,...w_n$, the Markov bi-gram model searches for the pos sequence $c_1,...c_n$ such that argmax $\Pi P(w_k \mid c_k) * P(c_k \mid c_{k-1})$ is achieved. In our experiment, since we were only focusing on the resolution of ambiguous words, a twisted Markov bi-gram model was applied. For each ambiguous target word, its pos with the highest model probability was tagged. The probability of each candidate pos c_k for a target word w_k was estimated as $P(c_k \mid c_{k-1}) * P(c_{k+1} \mid c_k) * P(w_k \mid c_k)$. We call this model the general Markov bi-gram model.

3.2 Word-Dependent Markov Bi-gram Model

The difference between the general Markov bi-gram model and the word-dependent Markov bi-gram model lies in the way in which the statistical data for $P(c_k \mid c_{k-1})$ and $P(c_{k+1} \mid c_k)$ is estimated. There are two approaches to estimating the probability. One is to count all the occurrences in the training data, and the other is to count only the occurrences in which each w_k occurs. In other words, the algorithm tags the pos c_k for w_k, such that c_k maximizes the probability of $P(c_k \mid w_k, c_{k-1}) * P(c_{k+1} \mid w_k, c_k) * P(w_k \mid c_k)$ instead of maximizing the probability of $P(c_k \mid c_{k-1}) * P(c_{k+1} \mid c_k) * P(w_k \mid c_k)$. We call this model the word-dependent Markov bi-gram model.

3.3 Context-Rule Model

The dependency features utilized to determine the best pos-tag in Markov models are the categories of context words. In fact, in some cases, the best pos-tags might be determined by using other context features, such as context words [Brill, 1992]. In the context-rule model, broad context information is utilized to determine the best pos-tag. We extend the scope of the dependency context of a target word to its 2 by 2 context windows. Therefore, the context features of a word can be represented by the vector of $[w_{-2}, c_{-2}, w_{-1}, c_{-1}, w_1, c_1, w_2, c_2]$. Each feature vector may be associated with a unique pos-tag or many ambiguous pos-tags. The association probability of a possible pos c_0' is $P(c_0' \mid w_0$, *feature vector*). If for some (w_0, c_0'), the value of $P(c_0' \mid w_0$, *feature vector*) is not 1, then this means that the c_0 of w_0 cannot be uniquely determined by its context vector. Some additional features have to be incorporated to resolve the ambiguity. If the full scope of the context feature vector is used, data sparseness problem will seriously degrade the system performance. Therefore, partial feature vectors are used instead of full feature vectors. The partial feature vectors applied in our context-rule model are w_{-1}, w_1, $c_{-2}c_{-1}$, c_1c_2, $c_{-1}c_1, w_{-2}c_{-1}$, $w_{-1}c_{-1}$, and c_1w_2.

In the training stage, for each feature vector type, many rule instances are generated, and their probabilities associated with the pos of the target word are calculated. For instance, with the feature vector types w_{-1}, w_1, $c_{-2}c_{-1}$, c_1c_2,..., we can extract the rule patterns of w_{-1}(先生), w_1(之餘), $c_{-2}c_{-1}$(*Nb*, *Na*), c_1c_2(*Ng*, *COMMA*), ...etc. associated with the pos *VE* of the target word from the following sentence while the target word is '研究 research':

周 Tsou (Nb) 先生 Mr (Na) 研究 research (VE) 之餘 after (Ng) ，(COMMA)

"After Mr. Tsou has done his research,"

Through the investigation of all training data, various different rule patterns (associated with a candidate pos of a target word) are generated and their association probabilities of $P(c_0' \mid w_0, \textit{feature vector})$ derived. For instance, if we take those word sequences listed in 0 as training data and take $c_{-1}c_1$ as a feature pattern, and if we let '研究' be the target word, then the rule pattern $c_{-1}c_1$(*VH*, *PERIOD*) will be extracted, and we will derive the probabilities *P*(*VE* | '研究', (*VH*, *PERIOD*)) = 2/3 and *P*(*NV* | '研究', (*VH*, *PERIOD*)) = 1/3. The rule patterns and their association probability are used to determine the probability of each candidate pos of a target word in a testing sentence. Suppose that the target word w_0 has ambiguous categories $c_1, c_2, ..., c_n$, and context patterns *pattern*$_1$, *pattern*$_2$, …, *pattern*$_m$; then, the probability of assigning tag c_i to the target word w_0 is defined as follows:

$$P(c_i) \cong \frac{\sum_{y=1}^{m} P(c_i \mid w, \textit{pattern}_y)}{\sum_{x=1}^{n}\sum_{y=1}^{m} P(c_x \mid w, \textit{pattern}_y)}.$$

In other words, the probabilities of different patterns with the same candidate pos are accumulated and normalized by means of the total probability distributed to all the candidates as the probability of the candidate pos. The algorithm tags the pos of the highest probability.

4. Experiments and Results

For our experiments, the Sinica corpus was divided into two parts. The training data contained 90% of the corpus, while the testing data contained the remaining 10%. Only the target words with ambiguous pos were evaluated. We evaluated only on the ambiguous words with frequencies higher than or equal to 10 for sufficiency of the training data and testing data. Furthermore, the total token count of the words with frequencies less than 10 occupied only 0.4335% of all the ambiguous word tokens. Since those words had much less effect on the overall performance, we just ignored them to simplify the designs of the evaluated tagging systems in the experiments. Another important reason was that for those words with low frequencies, all their tagging results had to be rechecked anyway, since our experiments

showed that low tagging accuracies were inevitable due to the lack of training data. We also examined the effects on the tagging accuracy and reliability on the words with variations on pos ambiguities and the amount of training data. Six ambiguous words with different frequencies, listed in Table 2, were selected as our target words for detail examinations.

Table 2. Target words used in the experiments tagging accuracy.

Word	Frequency	Ambiguity (Pos-Count)				
了	47607	Di-36063	T-11504	VJ-25	VC-11	
將	13188	D-7599	P-5547	Na-27	Di-8	VC-5
研究	4734	Nv-3695	VE-1032	VC-6	VA-1	
改變	1298	VC-953	Na-345			
演出	723	VC-392	Na-331			
採訪	121	VC-70	Nv-45	Na-6		

Table 3. Accuracy rates of the evaluated tagging algorithms.

Word	General Markov	Word-Depend. Markov	Context-Rule
了	96.95 %	97.92 %	98.87 %
將	93.47 %	93.17 %	95.52 %
研究	80.76 %	79.28 %	81.40 %
改變	87.60 %	89.92 %	93.02 %
採訪	68.06 %	63.89 %	77.78 %
演出	41.67 %	66.67 %	66.67 %
Average of 6 words	94.56 %	95.12 %	96.60 %
Average of all ambiguous words	91.07 %	94.07 %	95.08 %

The frequencies of some words were too low to provide enough training data, such as the words '採訪 interview' and '演出 perform' listed in 0. To solve the problem of data sparseness, the Jeffreys-Perks law, or Expected Likehood Estimation (ELE) [Manning *et al.*, 1999], was used as a smoothing method for all the tagging algorithms evaluated in the experiments. The probability $P(w_1,...,w_n)$ was defined as $\frac{C(w_1,...,w_n)}{N}$, where $C(w_1,...,w_n)$ is the number of times that pattern $w_1,...,w_n$ occurs in the training data, and N is the total number of training patterns. To smooth for an unseen event, the probability of $P(w_1,...,w_n)$ was redefined as $\frac{C(w_1,...,w_n)+\lambda}{N+B\lambda}$, where B denotes the number of all

pattern types in the training data and λ denotes the default occurrence count for an unseen event. That is to say, we took a value λ for an unseen event as its occurrence count. If the value of λ was 0, this means that there was no smoothing process for the unseen event. The most widely used value for λ is 0.5, which was also applied in our experiments.

4.1 Tagging Accuracy

In the experiments, we compared the tagging accuracy of the three tagging algorithms as described in section 3. The experiment results are shown in Table 3. It is obvious that the word-dependent Markov bi-gram model outperformed the general Markov bi-gram model. It reduced almost 30% the number of errors compared to the general Markov bi-gram model. As expected, the context-rule model performed the best for each selected word and the overall tagging accuracy. The tagging accuracy results for selected words show inconsistency. This is exemplified by the lower accuracy for the word '研究 research'. It is believed that the flexible usage of '研究 research' degraded the performances of the tagging algorithms. The lack of training data also hurt the performance of the tagging algorithms. The words with fewer training data, such as '採訪 interview' and '演出 perform', were also associated with poor tagging accuracy. Therefore, words with low frequencies should be handled using some general tagging algorithms to improve the overall performance of a tagging system. Furthermore, in future, word-dependent reliability criteria need to be studied.

4.2 Tagging Reliability

In the experiments on reliability, the confidence measure of the ratio of the probability gap between the top choice candidate and the second choice candidate $\frac{P(c_1)}{P(c_1)+P(c_2)}$ was adopted for all three models. The tagging results with confidence scores lower than a pre-defined threshold were re-checked. Some tagging results were assigned the default pos (in general, the one with the highest frequency of the word) since there were no training patterns applicable to the tagging process. Those tagging results that were not covered by the training patterns also needed to be re-checked. With the increased pre-defined threshold, the amount of partial corpus that needed to be re-checked could be estimated automatically since the Sinica corpus provides the correct pos-tag for each target word. Furthermore, the final accuracy could be estimated if the corresponding amount of partial corpus was proofread.

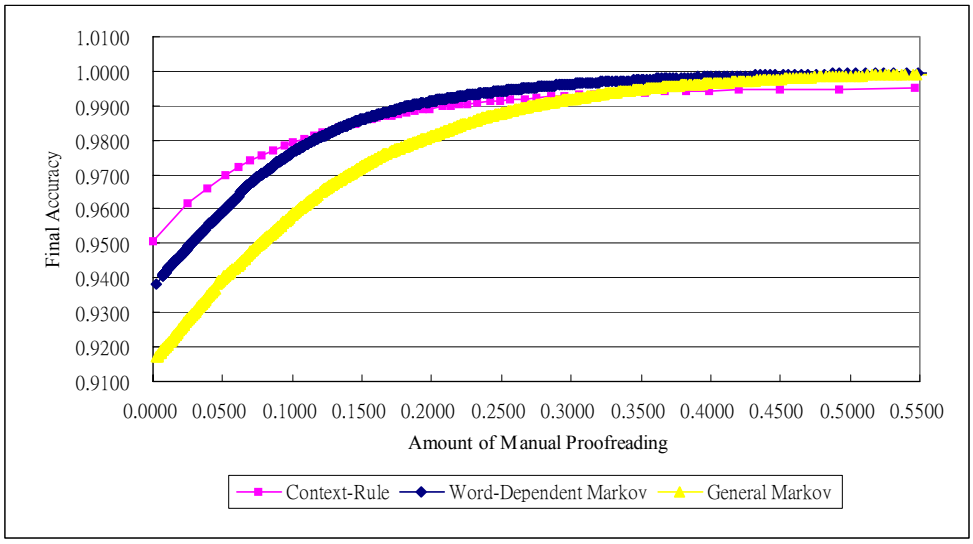


Figure 1. Tradeoff between the amount of manual proofreading and the final accuracy.

Figure 1 shows the results for the tradeoff between the amount of proofreading and the estimated final accuracy for the three algorithms. The x-coordinate indicates the portion of the partial corpus that needed to be manually proofread under a pre-defined threshold. The y-coordinate indicates the final accuracy after the corresponding portion of the corpus was proofread. Without any manual proofreading, the accuracy of the context-rule algorithm was about 1.4% higher than that of the word-dependent Markov bi-gram model. As the percentage of manual proofreading increased, the accuracy of each algorithm also increased. It is obvious that the accuracy of the context-rule model increased more slowly than did that of the two Markov models, as the amount of manual proofreading increased.

The final accuracy results of the context-rule model and the two Markov models coincided at approximately 98.5% and 99.4%, with around 13% and 35% manual proofreading. After that, both Markov models achieved higher final accuracy than the context-rule model did when the amount of manual proofreading increased more. The results indicate that if the required tagging accuracy is over 98.5%, then the two Markov models will be better choices since in our experiments, they achieved higher final accuracy than the context-rule model did. It can also be concluded that an algorithm with higher accuracy may not always be an accurate algorithm.

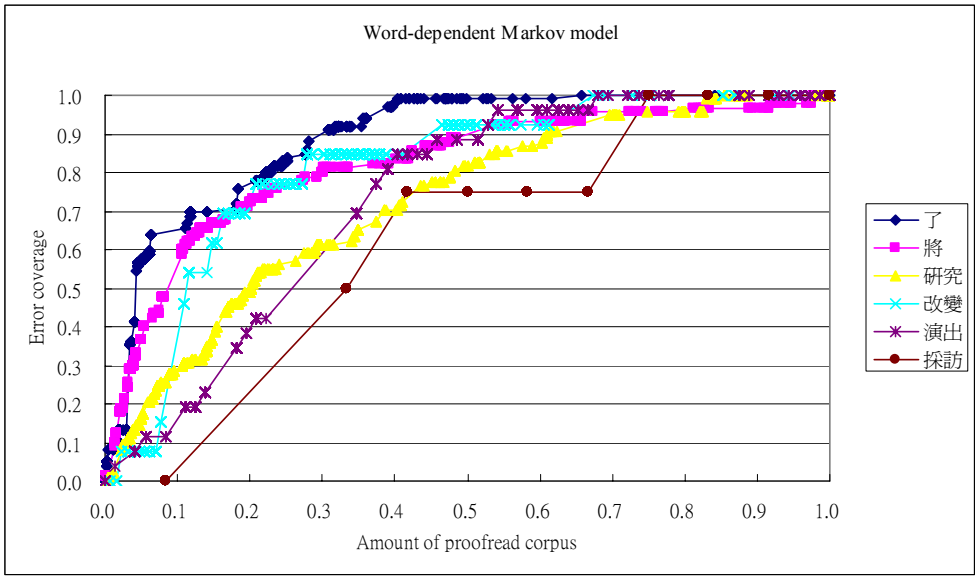


Figure 2. Error coverage of word-dependent Markov model after amount of corpus is proofread.

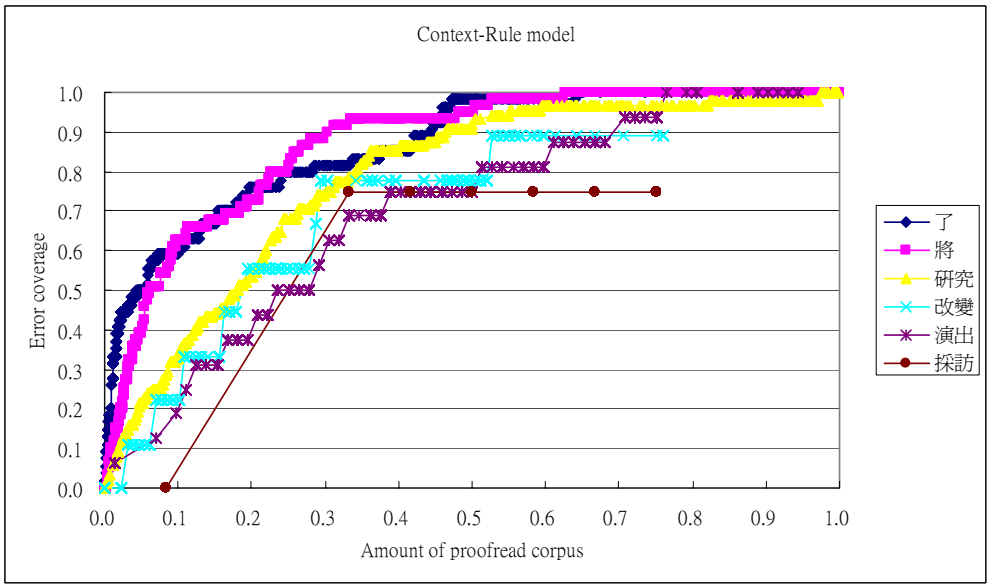


Figure 3. Error coverage of context-rule model after amount of corpus is proofread.

Figure 2 and Figure 3 show the error coverage of the six ambiguous target words after different portions of corpus are proofread respectively. It shows that not only tagging accuracy but also reliability were degraded due to the lack of sufficient training data. Tagging algorithms achieve better error coverage for target words with more training data.

4.3 The Tradeoff between the Amount of Manual Proofreading and the Final accuracy

There is a tradeoff between amount of manual proofreading and the final accuracy. If the goal of tagging is to achieve 99% accuracy, then an estimated threshold value of the confidence score needed to achieve the target accuracy rate will be given, and a tagged word with a confidence score less than this designated threshold value will be checked. On the other hand, if the requirement is to finish the tagging process in a a limited amount of time and with limited amount of manual labor, then in order to achieve the desired final accuracy, we will first need to estimate the portion of the corpus which will have to be proofread, and then determine the threshold value of the confidence score. Figure 4 shows the error coverage of each different portions of corpus with the lowest confidence score. By proofreading the initial 10% low confidence tagging data we achieve the most improvement in accuracy. As the amount of proofread corpus increased, the level of accuracy decreased rapidly. The experimental results of tagging reliability can help us decide which is the most cost-effective tagging algorithm and how to proofread tagging results under constraints on the available human resources and time.

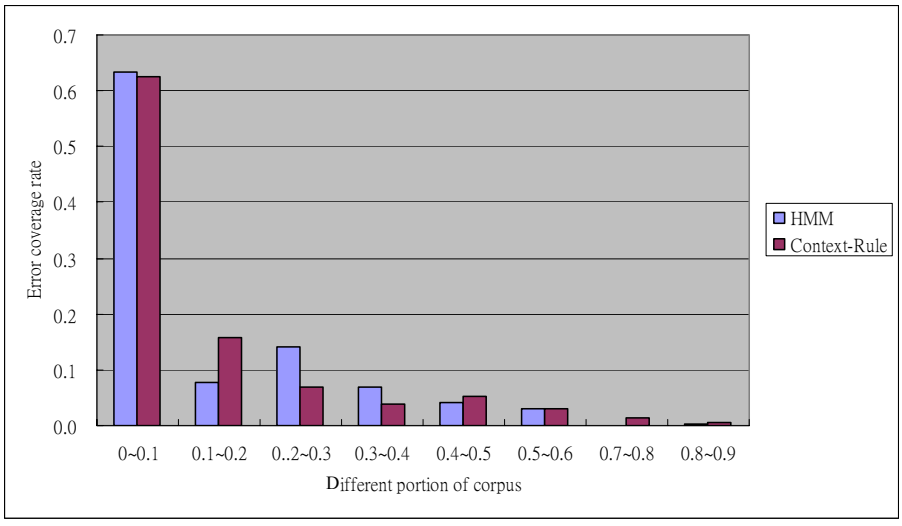


Figure 4. Error coverage rate of different portion of corpus to be proofread.

5. Conclusion

In this paper, we have proposed a context-rule model for pos tagging. We have also proposed a new way of finding the most cost-effective tagging algorithm. Cost-effectiveness is defined based on a criterion of reliability. The reliability of the system is measured in terms of the confidence score for ambiguity resolution of each tagging. The basic observation of confidence tagging is as follows: the larger the gap between the candidate pos with the highest probability and other (the second, for example) candidate pos with lower probability, the more confidence can be placed in the tagging result. It is believed that the ability to resolve pos ambiguity plays a more important part than the confidence measurement in the tagging system, since a larger gap between the first candidate pos and the second candidate pos can result in a high confidence score. Therefore, another reasonable measurement of the confidence score will work as well as the one used in our experiments if the tagging algorithms have good ability to resolve pos ambiguity.

For the best cost-effective tagging algorithm, on average, 20% of the samples of ambiguous words need to be rechecked to achieve 99% accuracy. In other words, the manual labor of proofreading is reduced by more than 80%. Our study on tagging reliability, in fact, provides a way to determine the optimal tagging strategy under different constraints. The constraints might be to achieve the best tagging accuracy under time and labor constraints or to achieve a certain accuracy with the least effort possible expended on proofreading. For instance, if the goal of tagging is to achieve 99% accuracy, then a threshold value of the confidence score needed to achieve the target accuracy will be estimated, and a tagged word with a confidence score less than this designated threshold value will be checked. On the other hand, if the constraint is to finish the tagging process under time and manual labor constraints, then in order to achieve the desired final accuracy, we will first estimate the portion of the corpus that will have to be proofread, and then determine the threshold value of the confidence score.

In future, we will extend the coverage of confidence checking for all words, including words with single pos, to detect flexible word usages. The confidence measure for words with single pos can be obtained by comparing the tagging probability of the pos of the words with the probabilities of the other categories. Furthermore, since tagging accuracy and reliability are degrading due to the intrinsic complexity of word usage and the less amount of training data, we will study word-dependent reliability to overcome the degrading problems. There are many possible confidence measures. For instance $\log(p(c_1)/p(c_2))$ is a reasonable alternative. We will study different alternatives in the future to obtain a more reliable confidence measure.

Acknowledgement:

The work was partially supported under NSC grant 92-2213-E-001-016.

References

C. H. Chang & C. D. Chen, 1993, "HMM-based Part-of-Speech Tagging for Chinese Corpora," in Proceedings of the Workshop on Very Large Corpora, Columbus, Ohio, pp. 40-47.

C. J. Chen, M. H. Bai, & K. J. Chen, 1997, "Category Guessing for Chinese Unknown Words," in Proceedings of NLPRS97, Phuket, Thailand, pp. 35-40.

Christopher D. Manning & Hinrich Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999, pp. 43-45, pp. 202-204.

E. Brill, "A Simple Rule-Based Part-of-Speech Taggers," in Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing 1992, pp. 152–155.

K. J. Chen, C. R. Huang, L. P. Chang, & H. L. Hsu, 1996, "Sinica Corpus: Design Methodology for Balanced Corpora," in Proceedings of PACLIC II, Seoul, Korea, pp. 167-176.

K. T. Lua, 1996, "Part of Speech Tagging of Chinese Sentences Using Genetic Algorithm," in Proceedings of ICCC96, National University of Singapore, pp. 45-49.

P. Kveton & K. Oliva, 2002, "(Semi-) Automatic Detection of Errors in Pos-Tagged Corpora," in Proceedings of Coling 2002, Taipei, Taiwan, pp. 509-515.

S. H. Liu, K. J. Chen, L. P. Chang, & Y. H. Chin, 1995, "Automatic Part-of-Speech Tagging for Chinese Corpora," on Computer Proceeding of Oriental Languages, Hawaii, Vol. 9, pp.31-48.

T. Nakagawa & Y. Matsumoto, 2002, "Detecting Errors in Corpora Using Support Vector Machines," in Proceedings of Coling 2002, Taipei, Taiwan, pp.709-715.

Computational Linguistics and Chinese Language Processing
Vol. 9, No.1, February 2004, pp.97-112

基於術語抽取與術語叢集技術的主題抽取

Topic Extraction Based on Techniques of Term Extraction and Term Clustering

林頌堅*

Sung-Chen Lin*

摘要

本論文針對主題抽取的問題，提出一系列以自然語言處理爲基礎的技術，應用這些技術可以從學術論文抽取重要的術語，並將這些術語依據彼此間的共現關係進行叢集，以叢集所得到的術語集合表示領域中重要的主題，提供研究人員學術領域的梗概並釐清他們的資訊需求。我們將所提出的方法應用到 ROCLING 研討會的論文資料上，結果顯示這個方法可以同時抽取出計算語言學領域的中文和英文術語，所得到的術語叢集結果也可以表示領域中重要的主題。這個初步的研究驗證了本論文所提出方法的可行性。重要的主題包括機器翻譯、語音處理、資訊檢索、語法模式與剖析、斷詞和統計式語言模型等等。從研究結果中，我們也發現計算語言學研究與實務應用有密切的關係。

關鍵詞: 主題抽取、術語抽取、術語叢集

Abstract

In this paper, we propose a series of natural language processing techniques to be used to extract important topics in a given research field. Topics as defined in this paper are important research problems, theories, and technical methods of the examined field, and we can represent them with groups of relevant terms. The terms are extracted from the texts of papers published in the field, including titles, abstracts, and bibliographies, because they convey important research information and are relevant to knowledge in that field. The topics can provide a clear outline of the field for researchers and are also useful for identifying users' information

*世新大學資訊傳播學系 Department of Information and Communications, Shih-Hsin University, Taipei, Taiwan, R.O.C.
Email: scl@cc.shu.edu.tw

needs when they are applied to information retrieval. To facilitate topic extraction, key terms in both Chinese and English are extracted from papers and are clustered into groups consisting of terms that frequently co-occur with each other. First, a PAT-tree is generated that stores all possible character strings appearing in the texts of papers. Character strings are retrieved from the PAT-tree as candidates of extracted terms and are tested using the statistical information of the string to filter out impossible candidates. The statistical information for a string includes (1) the total frequency count of the string in all the input papers, (2) the sum of the average frequency and the standard deviation of the string in each paper, and (3) the complexity of the front and rear adjacent character of the string. The total frequency count of the string and the sum of its average frequency and standard deviation are used to measure the importance of the corresponding term to the field. The complexity of adjacent characters is a criterion used to determine whether the string is a complete token of a term. The less complexity the adjacent characters, the more likely the string is a partial token of other terms. Finally, if the leftmost or rightmost part of a string is a stop word, the string is also filtered out. The extracted results are clustered to generate term groups according to their co-occurrences. Several techniques are used in the clustering algorithm to obtain multiple clustering results, including the clique algorithm and a group merging procedure. When the clique algorithm is performed, the latent semantic indexing technique is used to estimate the relevance between two terms to improve the deficiency of term co-occurrences in the papers. Two term groups are further merged into a new one when their members are similar because it is possible that the clusters represent the same topic. The above techniques were applied to the proceedings of ROCLING to uncover topics in the field of computational linguistics. The results show that the key terms in both Chinese and English were extracted successfully, and that the clustered groups represented the topics of computational linguistics. Therefore, the initial study proved the feasibility of the proposed techniques. The extracted topics included “machine translation,” “speech processing,” “information retrieval,” “grammars and parsers,” “Chinese word segmentation,” and “statistical language models.” From the results, we can observe that there is a close relation between basic research and applications in computational linguistics.

Keywords: Topic extraction, term extraction, term clustering

1. 緒論

本論文提出一個自動化的主題抽取方法，利用論文中的詞彙訊息來抽取學術領域的主題。論文的題名、摘要、本文，甚至所引用的參考文獻題名等文字資料表達了研究的問

題、方法與結果，因此這些論文資料中的術語與研究主題非常相關。以本論文做一例子，在題名、摘要和本文出現許多『學術領域』、『主題』、『論文』、『抽取』等等術語，可以了解這個研究與從學術論文中抽取主題相關。所以抽取論文中的術語可以了解論文的主題。在一個學術領域中，受到重視的主題的相關術語會在許多論文中出現。以計算語言學領域爲例，許多論文包含了諸如『語料庫』、『剖析』、『資訊檢索』等等術語，因爲它們與這個領域的重要主題相關。而且進一步地，主題相關的術語會經常一起出現，具有較強的共現(co-occurrence)關係。因此，如果對學術領域相關的論文進行分析，選取具有高頻而代表主題意義的術語，利用共現資訊將相關的術語叢集成一個集合，所形成的術語集合便可以視爲是領域中重要的主題。在分析論文的主題時，便可以透過論文對各術語集合的相關性來進行評估。

因應學術論文較多獨特術語的特性，本研究在術語抽取(term extraction)的技術上，參考[Chien, 1997]、[Chien, *et. al.*, 1999]和[Zhang, *et. al.*, 2000]等統計方法，利用字串的頻次爲基礎的統計訊息，從論文中抽取多語的術語。在術語叢集(term clustering)上，則考慮同義詞和一詞多義的現象，利用 LSI (latent semantics indexing) [Deerwester, *et. al.*, 1990] 和 clique 叢集演算法[Kowalski and Maybury, 2000]等技術，將經常共現的術語叢集起來。在應用上，我們使用 ROCLING 一到十四屆學術研討會的論文資料，進行術語抽取與術語叢集。研究結果初步驗證了這些技術用於主題抽取的可行性。

本論文其餘的章節架構如下：在第二節中扼要說明相關研究及所提出一系列之技術。接著在第三節和第四節中分述這個研究的核心技術：術語抽取和術語叢集。第四節中並且說明主題與論文之間相關程度的計算方式。第五節是應用這些技術到國內計算語言學領域的研究與結果。第六節則是本論文的結論。

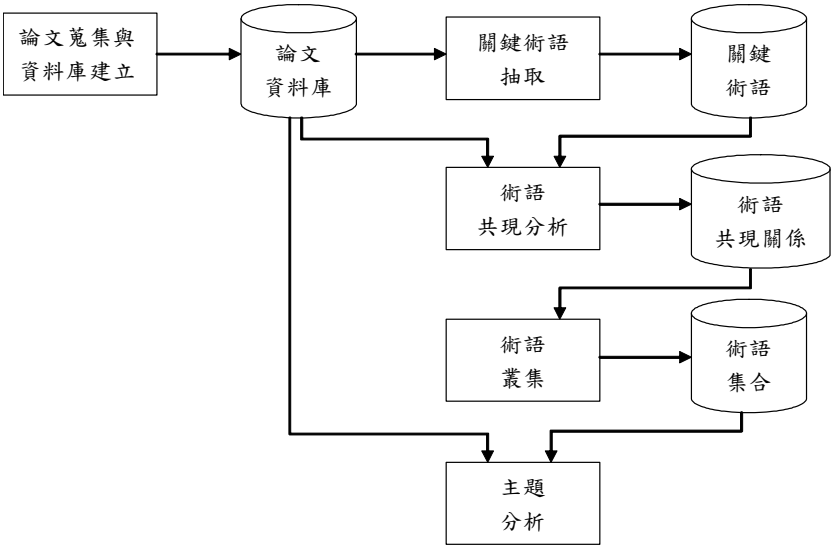
2. 本論文提出的主題分析方法

本論文希望發展主題抽取的技術，從相關論文抽取重要的主題。在資訊檢索研究的範疇中類似的研究有主題偵測(topic detection)。主題偵測希望從一序列來源各不相同的新聞中，偵測出某些『事件』(events)相關的連續報導[Wayne, 2000]。目前許多的研究利用『叢集假說』(cluster hypothesis)來解決這個問題[Yang, Pierce and Carbonell, 1998][Hatzivassiloglou, Gravano and Maganti, 2000]，以文件叢集(document clustering)技術，利用相關文件具有相似的術語分布，統計新進文件的術語分布情形，將文件歸入相關事件的集合中。因此，本論文也嘗試應用叢集假說發展相關技術。再者，主題偵測研究已應用專有名詞(proper nouns)等術語作爲區隔不同新聞事件的重要訊息[Hatzivassiloglou, Gravano and Maganti, 2000]，因此本論文也將嘗試利用論文中的相關術語。此外，主題偵測應用所謂的『新聞熱潮』(news bursts)現象，將時間訊息加入叢集演算法，提昇偵測的結果[Yang, Pierce and Carbonell, 1998]。雖然學術論文有所謂『資訊流行』(information epidemics)的說法，然而在實證研究中卻發現此一現象雖然存在，但並不常見[Tabah, 1996]，所以在本論文並不考慮加入時間訊息。

在本論文中，我們利用術語在論文中的共現關係，找出術語的叢集情形來代表主題。

以論文中出現的術語取代整篇論文作爲分析對象的主要原因是希望能獲得較可信賴的統計訊息。由於較小的學術領域所出版的論文數量較爲不足，統計上不易得到滿意的分析結果。以術語作爲分析對象，因爲數量較多，可以獲得充足的統計訊息，克服文件數量較少的問題。具有多個主題的論文也可藉由術語的叢集，找出所有的相關主題，並且進而探索主題間的關係。此外，文件叢集不易直接詮釋結果所代表的主題，術語叢集則可以由成員的語意進行解釋。

本論文方法的架構如圖一所示。首先對需要進行分析的學術領域蒐集相關論文資料，建立論文資料庫。資料庫中收錄的資料包括論文的題名、摘要和參考文獻的題名等作爲術語抽取與叢集分析的資訊，論文作者和出版年等項目則可以用來作爲後續的分析工作上。特別值得一提的是，國內的學術論文基本上是中、英語雙語並行，許多領域皆接受論文以中文或英文發表，然而並非所有的論文都具有雙語的題名和摘要。若只針對以某一種語言發表的論文進行分析，而忽略另一種語言，有可能造成某些主題被遺漏的情形。若是分別處理各種語言的論文，缺乏分屬兩種語言的術語在論文中的共現訊息，無法分析出這些術語的相關性，在整合上有相當大的困難。因此需要考慮這個特殊的論文發表現象，提出可以同時獲得兩種語言的術語之方法。本論文所提出的解決之道是加入論文中參考文獻的題名進行分析，通常論文的主題與其他的文獻相關時會加以引用，因此參考文獻的題名與主題間也有密切的關係，加入參考文獻的題名可以增加分析的資訊，而且引用的參考文獻可能來自中英文兩種語言，若能利用適當的多語術語抽取技術，便可以統計分屬兩種語言的相關術語的共現現象，整合兩種語言的術語訊息，得到較佳的結果。



圖一 本論文的主題抽取方法

在建立好論文資料庫後，便利用第 3 節所描述的多語術語抽取方法從論文資料中自動抽取領域中具有意義的術語。接著以第 4 節的術語叢集技術統計術語在論文中的共現關係，將相關的術語叢集成集合，用來代表特定的主題。進行主題分析時，對於某一主題，可以根據術語集合與論文的相關程度，取出具有主題的論文。

3. 多語環境下的術語抽取

爲了抽取主題相關的術語，我們首先確認重要的中英文詞組(phrases)以及中文的多字詞，再選擇具有代表意義的術語，作爲這一階段的結果。在學術論文中，常以詞組的形式表達重要的主題，比方在計算語言學領域中，可以發現如英文的"language model"、"machine translation"或是中文的"語言模型"、 "機器翻譯"等等都是重要術語。此外，中文的文本裡，詞與詞之間沒有明顯的界限，進行自然語言處理前，需要先斷詞。然而學術論文中經常有許多新的術語出現，來代表新的概念、方法和技術，我們無法事先收錄各個領域裡所有可能的術語來製作十分完整的詞典，進行斷詞。而且利用構詞律的規則式斷詞方法，需要處理同時中文和英文兩種語言，難以整合應用。所以本論文採用統計式的處理方法[Chien, et. al., 1999] ，以便同時解決中文的多字詞及中英文的詞組問題。

在過去對於術語抽取的相關研究中，曾利用字串的『相對頻率』(relative frequency)、『互見資訊』(mutual information)和『上下文依附』(context dependency)等各種統計訊息[Su, *et. al*., 1994][Chien, 1997][Zhang, *et. al*., 2000]。字串的『相對頻率』是指該字串的出現頻次與語料中所有長度相同字串平均頻次的比值，可以測量字串的重要性，相對頻率愈大的字串愈重要，愈可能是一個術語[Su, *et. al*., 1994]。『互見資訊』雖然有不同計算公式，但都是用來測量組成術語的字或詞彼此間的相互關係(association)，成員間『互見資訊』愈高的字串，愈有可能是一個術語[Su, *et. al*., 1994] [Zhang, *et. al*., 2000]。『上下文依附』則用來測量字串與上下文字詞間的依附程度，依附程度較大的字串可能是術語的一個部份，不應被抽取出來；反之，字串的依附程度較小，則可能代表是術語的邊界[Chien, 1997][Zhang, *et. al*., 2000]。

本論文所使用的方法如下：首先利用題名、摘要和參考文獻的題名等論文資料建立一個 PAT-tree 資料結構，儲存所有出現在論文資料中的字串及它們所在的論文資料[Chien, 1997]。接著在 PAT-tree 中擷取可能的字串作爲候選術語，以統計訊息及經驗法則(heuristic rules)作爲判斷是否爲術語的標準。在本論文中，所使用的統計訊息包括字串在所有資料中的出現總次數、字串的平均頻次和標準差(standard deviation)以及字串前後接字的複雜度。其中，字串的出現總次數、平均頻次和標準差等統計訊息和相關研究中的『相對頻率』作用相同，在於衡量字串的重要性。字串的出現總次數代表在領域中的重要性，總次數高表示這個字串在領域裡的論文經常出現而具有重要意義。字串對於出現論文的重要程度則用字串的平均頻次和標準差來表示，如式(1)

$$R_S \overset{def}{=} m_S + \sigma_S \tag{1}$$

m_S 和 σ_S 分別代表字串 S 的平均頻次和標準差。當字串 S 的平均頻次超過某一閾值

時，表示此字串極有可能在許多論文中出現多次，是這些論文的關鍵術語，應該被選取出來。或者雖然字串 S 的平均頻次較低，但在某些論文中出現相當多次，是這些論文的關鍵術語，也需要被選取出來，此時字串 S 會有一個較大的標準差 σ_S。因此，我們可以利用字串的平均頻次和標準差的總和 R_S 代表字串對出現論文的重要程度，R_S 值愈高的字串對出現論文愈重要。

前後接字的複雜度則和『上下文依附』的作用相同，可以判斷字串是否是一個完整的術語或是其他術語的部分，字串 S 的前後接字複雜度 C_{1S} 和 C_{2S} 分別如式(2a)和(2b)所示

$$C_{1S} \overset{def}{=} -\sum_{a} \frac{F_{aS}}{F_S} \log(\frac{F_{aS}}{F_S}) \tag{2a}$$

$$C_{2S} \overset{def}{=} -\sum_{b} \frac{F_{Sb}}{F_S} \log(\frac{F_{Sb}}{F_S}) \tag{2b}$$

式(2a)和(2b)中，a 和 b 代表字串 S 在論文資料中任一個可能的前接字和後接字，F_S、F_{aS} 和 F_{Sb} 分別是字串 S、aS 和 Sb 的出現總次數。以式(2a)前接字的情形來看，若是字串 S 有愈多種類的前接字，而且每一種前接字出現的次數越接近時，C_{1S} 的值愈大，反之，當字串前只有一種前接字時，C_{1S} 的值等於 0，或是有一個前接字出現的機會較其他大非常多時，則 C_{1S} 的值接近於 0，表示該字串再加上這個前接字可能才是一個術語，所以前接字複雜度愈大代表該字串愈有可能是獨立的術語。後接字的情形也是相同的道理。

通過上面條件的字串，再利用停用詞(stop words)不能出現在字串首尾的經驗法則，進一步過濾不完整的術語。在過去的經驗中，介詞、連接詞和補語等停用詞常出現在抽取出字串的首尾，如 “名詞+的”、“名詞+of”或“to+動詞”等詞組結構。但停用詞出現在字串的中間代表特定的詞組，例如“part of speech”，因此，將這種情形加以保留。

在確認論文資料中重要的中英文詞組以及中文的多字詞後，以這些術語建立斷詞處理所需的詞典。我們使用長詞優先法則與術語的出現總頻次將所有論文資料加以斷詞。論文資料經過斷詞處理後，將產生了一些中英文的詞組、詞和一些中文單字。在這一階段的目標是抽取論文中所有可能代表主題的術語，因此我們過濾具有以下情形的字串。首先是中文單字，多半是一些停用詞或是無法組成術語的片段。其次，出現總次數與式(1)之 R_S 值太小的術語，因爲對領域的重要性較低，也加以過濾。剩下的術語則是下一階段分析的對象。

4. 術語叢集

本論文依據術語在論文資料的共現關係，將術語進行叢集，以一組叢集的相關術語作爲一個主題。由於有些術語可能包含在不同的主題中，本節中提出一個可以對術語進行多重叢集的演算法。

首先，我們將上一階段抽取出來的術語，利用 cliques 叢集演算法[Kowalski and Maybury, 2000]進行術語叢集。cliques 叢集演算法在選定最小相關程度的情形下，可以得到若干個術語集合，在集合中的術語，彼此間的相關程度都在所選定的最小相關程度之上，而且術語可以被歸類到多個集合，因此符合多重叢集的要求。本論文所使用的相關程度計算方式如下：我們先計算每一術語在每一筆論文資料中出現的頻次，作爲術語的特徵值。但因爲術語在論文資料裡出現頻次不高，爲了使低頻次的術語差異不會太大，以頻次的平方根作爲特徵值，並除以術語的總頻次進行正規化(normalization)。如此一來，對每一術語便有一組特徵向量(feature vector)，如式(3)表示某一術語 *A* 的特徵向量。

$$\vec{v}_A \stackrel{def}{=} \frac{[\sqrt{f_{A,1}}, \sqrt{f_{A,2}}, \ldots, \sqrt{f_{A,N}}]^T}{\sum_{i=1}^{N} f_{A,i}} \tag{3}$$

式(3)中，$f_{A,i}$代表術語*A*在第*i*篇論文資料中出現的頻次，分母的 $\sum_{i=1}^{N} f_{A,i}$ 是 *A* 的總頻次。術語間的相關程度便可以利用特徵向量的內積 (inner product)來估算。

經過 cliques 演算法與上述的相關程度計算方式所得到的結果是相當嚴格的，相關術語若要叢集在同一個集合中，所有術語彼此間的共現關係必須都很強。然而，在論文中相同或相近的概念可能以不同術語來表示，使得相關的術語不一定經常共同出現，利用上述的估算方法將會得到很小的相關程度，無法將這些術語叢集起來。爲此，本論文採用以下兩種技術來加以補救。

首先我們改用 LSI 技術估算術語間的相關程度。LSI 技術是利用奇異值分解 (SVD, singular value decomposition)對上述的特徵向量所形成的『術語-特徵』矩陣 *M* 進行分解[Deerwester, *et. al*., 1990]，產生新的矩陣 $\hat{M}$ ，假設 $\hat{M}$ 的秩(rank)爲 *k*，*k* 小於或等於原先矩陣 *M* 的秩，則 $\hat{M}$ 是所有秩爲 *k* 的矩陣中，與 *M* 的平方差最小的矩陣。以術語在新矩陣 $\hat{M}$ 所對應的行向量(row vector)取代原先之特徵向量，當進行術語的相關程度估算時，便可以 $\hat{M}\hat{M}^T$ 來估算原先以 MM^T 計算兩兩術語特徵向量間的內積值。利用 SVD 可以取得隱含語義結構(latent semantic structure)，使得原先因爲共現關係較弱或是不存在的兩個相關術語，獲得較大的估算值[Deerwester, *et. al*., 1990]。

其次，在進行 cliques 叢集演算法後，對於所得到的結果依據它們成員間重疊的情形進行合併。假設兩個集合之間有多個成員是相同的，這兩個集合很可能屬於同一主題，我們即將這兩個術語集合進行聯集，產生新集合。以數學式表示如下，C_1 和 C_2 爲兩個集合，如果$|C_1 \cap C_2| \geq c*Min(|C_1|, |C_2|)$，則合併成新的集合$C_3$，此處 $c*Min(|C_1|, |C_2|)$是兩個集合的最小相同成員數，*c* 是一個介於 1 與 0 間的實數，$Min(|C_1|, |C_2|)$是取出兩個數值中最小值的函數。

經過上述的叢集處理後，可以得到代表重要主題的術語集合。在確認主題的相關論

文方面，可以利用 LSI 的估計方式[Deerwester, *et*. *al*., 1990]，計算每一術語集合與論文間的相關程度。計算方式是將術語集合中每一個術語的特徵向量相加，再正規化成單位向量，即可求得主題與所有論文之間的相關程度估算值。最後依據將相關程度大的論文資料取出，作爲主題的相關論文。

5. 國內計算語言學的主題抽取之實驗結果

計算語言學研討會 ROCLING 是國內的計算語言學領域相當重要的學術活動。因此，ROCLING 的研討會論文集，可以說是歷年來國內計算語言學領域學者的心血結晶，所蘊含的主題也是他們所共同關心的主題。因此，本論文將以第一屆(1988)到第十四屆(2001) ROCLING 研討會的 235 篇論文資料做爲分析國內計算語言學主題的素材。

進行術語抽取時，本論文根據字串長度將字串出現總次數的閾值作不同的設定，較短的字串(2 或 3 字)設定爲 15 次，較長的字串(4~5 字)則設定爲 10 次，平均頻次和標準差的總和 R_S 和前後接字的複雜度分別設爲 2.5 與 0.5。接著利用抽取出來的多字詞或詞組對論文資料進行斷詞，過濾不是術語的字串，並進行統計。結果共得到 343 個術語，表一是出現總次數最高的前 50 個術語及它們的出現次數。表一中列出的術語大多屬於概念較廣泛的術語。這些術語出現在較多論文資料中，因此出現次數較高。表一中有些是其他領域也常見的術語，比方說『系統』、『方法』、『分析』等等，但許多術語和計算機科學及語言學相關，如『parsing』、『data』、『speech』、『lexical』等等，或是本身即是計算語言學特有的概念，如『speech recognition』、『machine translation』等等。

表一 術語抽取所得到的前 50 個出現總次數最高的術語

次序	詞名	出現次數	次序	詞名	出現次數	次序	詞名	出現次數
1	parsing	209	2	speech	184	3	系統	175
4	sentences	141	5	lexical	138	6	mandarin	134
7	speech recognition	132	8	方法	131	9	semantic	130
10	corpus	129	11	syntactic	107	12	recognition	106
13	data	105	14	分析	104	15	learning	102
16	mandarin chinese	97	17	sentence	97	18	machine translation	92
19	words	92	20	theory	87	21	rules	84
22	models	83	23	phrase	83	24	漢語	82
25	classification	80	26	parser	80	27	probabilistic	78
28	動詞	78	29	語音	78	30	knowledge	74
31	語法	74	32	chinese text	73	33	語言	73
34	semantics	72	35	corpora	71	36	used	71
37	國語	71	38	discourse	70	39	處理	70
40	dictionary	68	41	problem	65	42	分類	65
43	corpus based	64	44	design	62	45	information retrieval	62
46	syntax	61	47	generation	60	48	語料庫	60
49	應用	60	50	character	59			

接著進行術語叢集，首先利用 LSI 技術進行相關程度估算。由於無法以客觀而有系統的方式決定新矩陣之秩的大小[Deerwester, *et. al.*, 1990] ，因此在本論文中分別嘗試秩爲 30、60 及 120 的新矩陣，產生術語的特徵向量來估算相關程度。將 clique 叢集中相關程度的閾值與集合合併 *c* 值分別設爲 0.4 和 0.6，最後所得到三個術語以上的叢集的數目與未被叢集術語的數目，如表二所示，另外爲了檢驗應用 LSI 技術的優點，表二中也同時顯示未經 SVD 處理的特徵向量之叢集結果。

表二 進行術語叢集所得到的結果

	SVD Rank=120	SVD Rank=60	SVD Rank=30	Original feature vector
cliques 叢集後的集合數目	78	85	74	65
合併後的集合數目	44	34	32	32
未被叢集的術語數目	209	189	214	223

從表二中，可以觀察到經過 SVD 處理的特徵向量，不論秩的大小，其未被叢集的術語數目都較原先特徵向量者少，換言之，LSI 技術有助於捕捉術語不共現卻相關的隱含語義結構，因此較多的術語可以被叢集。其中秩值爲 60 的特徵向量所得到的結果，是未被叢集的術語數目最少者，因此我們將所得到的 34 個術語集合作爲進一步分析的對象，這 34 個術語集合列表於附錄一。

從術語集合的結果我們可以看到幾個現象。第一、若干集合同時包含中文術語與英文術語，甚至包含縮寫與相同概念但不同詞名的術語，比方說，集合 12 包含了‘machine translation’、‘mt’、‘機器翻譯’等術語；或是又如集合 18 包含了 ‘word identification’、‘word segmentation’、‘斷詞’等術語。可見將參考文獻的題名加入論文資料，可以獲得中文和英文兩種語言的詞彙訊息，而且利用術語的共現關係與 LSI 技術可以將相關的術語叢集起來。第二、大部分的術語集合都可以明顯地用來代表一個特定的主題。除了集合 3、11 與 29 由概念較廣泛的術語形成之外，其餘集合的術語間都具有相關性，可以用來代表計算語言學領域中的特定主題。比方說，集合 7 爲語音辨認的相關術語、集合 9 則爲文件分類的相關術語。此外，對於沒有被叢集的術語加以檢視，發現術語未被叢集的原因，一是該術語與其他術語間的相關程度較小，而這些往往都是一些主題較廣泛的術語，或者是該術語的主題相當特殊，僅有少數論文進行探討，因此僅與少數術語發生共現關係，無法形成術語集合。因此，本實驗所得到的術語集合大多具有概念明確，容易進一步詮釋結果，而本論文所提出來的主題抽取方法的可行性，便可以得到初步驗證。

由於篇幅的限制，本論文無法對所有抽取出的術語集合一一進行詳盡的報告，以下針對幾個主題較明確的術語集合進行說明。表三是與語言的計算模式相關的術語集合及相關論文的列表，論文前的數值是論文在 ROCLING 研討會中發表的年份。表三可以驗證早期的計算語言學多以規則式的語法模式與剖析爲主，近來則較多發展統計式語言模型，而斷詞則是一直以來國內計算語言學領域相當重視的獨特問題。

表三 與語言的計算模式相關的術語集合及相關論文

集合編號	術語	相關論文資料
23 語法模式與剖析	分析, 表達, 剖析, 格位, 訊息, 動詞, 結構, 詞類, 漢語, 語法, 語法模式, 語意, 模式, 關係	1989 "訊息爲本的格位語法--一個適用於表達中文的語法模式" 1991 "連接詞的語法表達模式-以中文訊息格位語法(ICG)爲本的表達形式" 1992 "漢語的動詞名物化初探--漢語中帶論元的名物化派生名詞"
18 斷詞	chinese text, chinese word segmentation, segmentation, unknown word, word identification, word segmentation, words, 斷詞	1994 "Chinese-Word Segmentation Based on Maximal-Matching and Bigram Techniques" 1995 "A Unifying Approach to Segmentation of Chinese and Its Application to Text Retrieval" 1997 "Unknown Word Detection for Chinese by a Corpus-based Learning Method" 1997 "Chinese Word Segmentation and Part-of-Speech Tagging in One Step" 1997 "A Simple Heuristic Approach for Word Segmentation"
22 統計式語言模型	bigram, class based, clustering, entropy, language model, language modeling, language models, n gram	1994 "An Estimation of the Entropy of Chinese - A New Approach to Constructing Class-based n-gram Models" 1997 "Truncation on Combined Word-Based and Class-Based Language Model Using Kullback-Leibler Distance Criterion" 2001 "使用關聯法則爲主之語言模型於擷取長距離中文文字關聯性"

此外從研究結果中，我們也發現計算語言學研究與實務應用有密切的關係，表四到表六分別列出與機器翻譯、語音處理和資訊檢索相關的集合。從表四的結果，說明機器翻譯是計算語言學最早的應用問題之一[Lenders, 2001]，而其發展從規則式的自動翻譯到統計式，近期的應用則是在跨語言檢索部分。

表四 與機器翻譯相關的術語集合及相關論文

集合編號	術語	相關論文資料
12 機器翻譯	'bilingual', 'machine translation', 'mt', 'transfer', '機器翻譯'	1991 "Lexicon-Driven Transfer In English-Chinese Machine Translation" 1992 "A Modular and Statistical Approach to Machine Translation" (只有與叢集 12 相關)
32 機器翻譯	'bilingual', 'machine translation', 'translation', '機器翻譯'	1995 "THE NEW GENERATION BEHAVIORTRAN: DESIGN PHILOSOPHY AND SYSTEM ARCHITECTURE" 1996 "介詞翻譯法則的自動擷取" 2001 "統計式片語翻譯模型"

表五 與語音處理相關的術語集合及相關論文

集合編號	術語	相關論文資料
13 語言模型	dictation, large vocabulary, 語言模型, 語音辨認	1993 "國語語音辨認中詞群雙連語言模型的解碼方法" 1994 "國語語音辨認中詞群語言模型之分群方法與應用" 1995 "應用於'音中仙'國語聽寫機之短語規則分析與建立" 1996 "國語語音辨認中多領域語言模型之訓練、偵測與調適" 1999 "國語電話語音辨認之強健性特徵參數及其調整方法" (只有與叢集 17 相關)
17 語言模型	國語, 語言模型, 語音辨認, 辨認	
7 聲學辨認	hidden markov, maximum, robust speech recognition, speech recognition	1998 "Speaker-Independent Continuous Mandarin Speech Recognition Under Telephone Environments" 1999 "國語電話語音辨認之強健性特徵參數及其調整方法" 2000 "具有累進學習能力之貝氏預測法則在汽車語音辨識之應用" 2000 "綜合麥克風陣列及模型調整技術之遠距離語音辨識系統"
30 語音合成	speech, synthesis, 文句翻語音, 合成, 系統, 音節, 國語, 連音, 語音, 輸入	1995 "以 CELP 爲基礎之文句翻語音中韻律訊息之產生與調整" 1996 "時間比例基週波形內差--一個國語音節信號合成之新方法" 1996 "中英文文句翻語音系統中連音處理之研究" 1999 "台語多聲調音節合成單元資料庫暨文字轉語音雛形系統之發展" (只有與叢集 30 相關) 1999 "國語文句翻台語語音系統之研究" (只有與叢集 30 相關) 2001 "Pitch Marking Based on an Adaptable Filter and a Peak-Valley Estimation Method", (只有與叢集 31 相關)
31 語音合成	mandarin text to speech, pitch, prosodic, speech, synthesis, 文句翻語音, 合成	

在過去計算語言學所處理的對象多爲書寫語言(orthographic languages)，近年來語音處理已經成爲計算語言學相當重視的主題。從 ROCLING 的論文資料中所得到的結果可以分析成語言模型、聲學辨認以及語音合成三個主題(表五)。國內計算語言學較早進行研究的主題是語言模型和語音合成，近年在聲學辨認研究上，也有許多研究人員進入這個領域發表相關論文。在表五，另外還可將語音合成研究分成系統製作(集合 30)與聲學訊息研究(集合 31)兩個部分。

表六 與資訊檢索相關的術語集合及相關論文

集合編號	術語	相關論文資料
25 資訊檢索	csmart, databases, document, indexing, information retrieval, retrieval, text retrieval, 檢索	1995 "適合大量中文文件全文檢索的索引及資料壓縮技術" 1996 "尋易(Csmart-II):智慧型網路中文資訊檢索系統" 1997 "An Assessment on Character-based Chinese News Filtering Using Latent Semantic Indexing" 1999 "A New Syllable-Based Approach for Retrieving Mandarin Spoken Documents Using Short Speech Queries"
9 文件分類	document, hierarchical, text categorization, 分類, 文件, 文件分類, 特徵	1993 "中文文件自動分類之研究" 1999 "階層式文件自動分類之特徵選取研究" 2001 "基於階層式神經網路之自動文件分類方法" 2001 "適應性文件分類系統"
28 文件分類	document, text categorization, 分類, 文件分類, 文件自動, 關鍵詞	

在計算語言學領域中，資訊檢索比起其他研究可說是一個較新的主題，然而由於網際網路與電子文件的發展使得這項應用成爲相當具有潛力的主題。我們可以從表六中發現國內計算語言學在這方面的重要研究包括資訊檢索和文件分類。

6. 結論

本論文針對主題分析的問題，提出一系列以自然語言處理爲基礎的技術，從學術領域中發表的論文資料中抽取重要的術語，並將這些術語依據彼此間共現關係進行叢集，以叢集所得到的術語集合表示領域中重要的主題。在本論文中，我們將所提出的方法應用到 ROCLING 研討會的論文資料上，結果初步驗證了本論文所提出方法的可行性。

在後續的研究上，將進一步改善目前所提出來的方法，比方說，目前術語叢集的效果還不十分理想，對於術語叢集技術的改進將是下一階段努力的目標。此外，在本研究中有許多參數需要設定，未來需要參考各種客觀的參數調整法來達到較佳的結果。最主要的工作在於深入探討各主題的起源、發展與演變之外，我們將探索各個主題之間的相關性，並嘗試將結果以圖形化的方式加以呈現。另外，對於不同學術領域間的相關主題的發掘和分析，比方說資訊檢索同樣是圖書資訊學所關心的主題，兩個領域間共通與相異的分析相當值得探討。

致謝

本研究爲國科會計畫 NSC91-2413-H-128-004-『國內計算語言學學術資訊交流之研究(I)』之研究成果。另外，作者亦對三位審查者的寶貴意見與建議深表感謝。

參考文獻

Lee-Feng Chien, "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval," *Proceedings of SIGIR'97*, 1997, pp. 50-58.

Lee-Feng Chien, Chun-Liang Chen, Wen-Hsiang Lu, and Yuan-Lu Chang, "Recent Results on Domain-Specific Term Extraction From Online Chinese Text Resources," *Proceedings of ROCLING XII*, 1999, pp. 203-218.

K. W. Church and R. L. Mercer, "Introduction to the Special Issue on Computational Linguistics Using Large Corpora," *Computational Linguistics*, 19(1), 1993, pp.1-24.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 41(6), pp. 391-407.

V. Hatzivassiloglou, L. Gravano and A. Maganti, "An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering," *Proceedings of SIGIR'2000*, 2000, pp. 224-231.

G. J. Kowalski and M. T. Maybury, "Document and Term Clustering," *Information Storage and Retrieval Systems: Theory and Implementation, 2nd ed.*, Chapter 6, 2000, pp.139-163.

W. Lenders, "Past and Future Goals of Computational Linguistics," *Proceedings of ROCLING XIV*, 2001, pp. 213-236.

Keh-Yih Su, Ming-Wen Wu, and Jing-Shin Chang, "A Corpus-based Approach to Automatic Compound Extraction," *Proceedings of ACL 94*, 1994, pp. 242-247.

A. N. Tabah, *Information Epidemics and the Growth of Physics*, Ph. D. Dissertation of McGill University, Canada, 1996.

C. L. Wayne, "Topic Detection and Tracking in English and Chinese," *Proceedings of IRAL 5*, 2000, pp.165-172.

Y. Yang, T. Pierce, and J. Carbonell, "A Study on Retrospective and On-Line Event Detection," *Proceedings of SIGIR'98*, 1998, pp. 28-36.

Jian Zhang, Jianfeng Gao, and Ming Zhou, "Extraction of Chinese Compound Words: An Experimental Study on a Very Large Corpus," *Proceedings of the Second Chinese Language Processing Workshop*, 2000, pp. 132-139.

附錄一 ROCLING研討會論文資料所得到的術語叢集

叢集編號	術語
1	generation, generator, systemic, text generation
2	acquisition, explanation, generalization, learning
3	方法, 系統, 問題, 處理
4	initial, min, taiwanese, 台語, 台灣, 資料庫
5	atn, attachment, pp, preference
6	complexity, computational, gpsg, morphology
7	hidden markov, maximum, robust speech recognition, speech recognition
8	aspect, logic, temporal, tense
9	document, hierarchical, text categorization, 分類, 文件, 文件分類, 特徵
10	classifiers, decision, non, symbols
11	分析, 系統, 處理, 語言
12	bilingual, machine translation, mt, transfer, 機器翻譯
13	dictation, large vocabulary, 語言模型, 語音辨認
14	adaptation, maximum, robust speech recognition, 語音辨識
15	attachment, pp, preference, score
16	系統, 設計, 輸入, 鍵盤
17	國語, 語言模型, 語音辨認, 辨認
18	chinese text, chinese word segmentation, segmentation, unknown word, word identification, word segmentation, words, 斷詞
19	attention, conversation, discourse, elicitation, interaction
20	continuous, hidden markov, maximum, speech recognition
21	統計, 詞彙, 語言, 語料
22	bigram, class based, clustering, entropy, language model, language modeling, language models, n gram
23	分析, 表達, 剖析, 格位, 訊息, 動詞, 結構, 詞類, 漢語, 語法, 語法模式, 語意, 模式, 關係
24	adaptive, compression, scheme, 英文, 資料, 調整, 壓縮
25	csmart, databases, document, indexing, information retrieval, retrieval, text retrieval, 檢索
26	grammars, parser, parsing, sentence
27	continuous, large vocabulary, mandarin, speaker, speech, speech recognition, telephone
28	document, text categorization, 分類, 文件分類, 文件自動, 關鍵詞
29	方法, 系統, 設計, 應用

叢集編號	術語
30	speech, synthesis, 文句翻語音, 合成, 系統, 音節, 國語, 連音, 語音, 輸入
31	mandarin text to speech, pitch, prosodic, speech, synthesis, 文句翻語音, 合成
32	bilingual, machine translation, translation, 機器翻譯
33	explanation, generalization, learning, parse
34	aspect, functional, lexical, lexical semantic, mandarin chinese, meaning, parsing, phrase, roles, semantic, semantics, syntactic, syntax, thematic, theory, verb, verbal, verbs

Computational Linguistics and Chinese Language Processing
Vol. 9, No. 1, February 2004 , pp.113-128

The Properties and Further Applications of Chinese Frequent Strings

Yih-Jeng Lin*, and Ming-Shing Yu+

Abstract

This paper reveals some important properties of CFSs and applications in Chinese natural language processing (NLP). We have previously proposed a method for extracting Chinese frequent strings that contain unknown words from a Chinese corpus [Lin and Yu 2001]. We found that CFSs contain many 4-character strings, 3-word strings, and longer n-grams. Such information can only be derived from an extremely large corpus using a traditional language model(LM). In contrast to using a traditional LM, we can achieve high precision and efficiency by using CFSs to solve Chinese toneless phoneme-to-character conversion and to correct Chinese spelling errors with a small training corpus. An accuracy rate of 92.86% was achieved for Chinese toneless phoneme-to-character conversion, and an accuracy rate of 87.32% was achieved for Chinese spelling error correction. We also attempted to assign syntactic categories to a CFS. The accuracy rate for assigning syntactic categories to the CFSs was 88.53% for outside testing when the syntactic categories of the highest level were used.

Keywords: Chinese frequent strings, unknown words, Chinese toneless phoneme-to-character, Chinese spelling error correction, language model.

1. Introduction

An increasing number of new or unknown words are being used on the Internet. Such new or unknown words are called "out of vocabulary (OOV) words" [Yang 1998], and they are not listed in traditional dictionaries. Many researchers have overcome problems caused by OOV words by using N-gram LMs along with smoothing methods. N-gram LMs have many useful applications in NLP [Yang 1998]. In Chinese NLP tasks, word-based bi-gram LMs are used by many researchers. To obtain useful probabilities for training, a corpus size proportional to 80000^2 (80000 is the approximate number of words in ASCED) = $6.4*10^9$ words is required.

* Department of Information Management, Chien Kuo Institute of Technology, Changhua, 500 Taiwan
E-mail: yclin@ckit.edu.tw Tel: 04-7111111 ext 3637 Fax:04-7111142
+ Department of Computer Science, National Chung-Hsing University, Taichung, 40227 Taiwan

However, it is not easy to find such a corpus at the present time.

A small-size corpus will lead too many unseen events when using N-gram LMs. Although we can apply some smoothing strategies, such as Witten-Bell interpolation or the Good-turing method [Wu and Zheng 2001] to estimate the probabilities of unseen events, this will be of no use when the size of training corpus is limited. From our observations, many the unseen events that occur when using N-gram LMs are unknown words or phrases. Such unknown words and phrases cannot be found in a dictionary. For example, the term "週休二日" (two days off per week) is presently popular in Taiwan. We cannot find this term in a traditional dictionary. The term "週休二日" is a 4-word string pattern which consists of four words: "週" (a week), "休" (to rest), "二" (two), and "日" (day). A word-based 4-gram LM and a large training corpus are required to record the data of such terms. Such a word-base 4-gram LM has not been applied to Chinese NLP in practice, and such a huge training corpus cannot be found at present. Alternatively, we can record the specifics of the term "週休二日" by using a CFS with relatively limited training data in which the specified term appear two or more times. Such training data could be recorded in one or two news articles containing hundreds of Chinese characters. Many researchers have shown that frequent strings can be used in many applications [Jelinek 1990; Suhm and Waibel 1994].

We have shown that adding Chinese frequent strings (CFSs), including unknown words, can improve performance in Chinese NLP tasks [Lin and Yu 2001]. A CFS defined based on our research is a Chinese string which appears two or more times by itself in the corpus. For example, consider the following fragment:

"國立中興大學，中興大學。"（National Chung-Hsing University, Chung-Hsing University.）

"中興大學" (Chung-Hsing University) is a CFS since it appears twice and its appearances are not brought out by other longer strings. The string "中興" (Chung-Hsing) appears twice, but it is not a CFS here since it is brought about by the longer string "中興大學".

In our previous research, we showed that adding CFSs to a traditional lexicon, such as ASCED, can reduce the normalized perplexity from 251.7 to 63.5 [Lin and Yu 2001]. We also employed CFSs combined with ASCED as a dictionary to solve some Chinese NLP problems using the word-based uni-gram language model. We achieved promising results in both Chinese CTP and PTC conversion. It is well known that using a word-based bi-gram LM with a traditional lexicon can also improve accuracy in these two cases, especially in Chinese PTC conversion.

The organization of this paper is as follows. Section 2 gives some properties and distributions of CFSs, and we also make a comparison between CFS and an n-gram LM. Section 3 shows that by using a CFS-based uni-gram LM, we can achieve higher accuracy

than we can by using a traditional lexicon with a word-based bi-gram LM. We demonstrate this by using two challenging examples of Chinese NLP. In section 4, we assign syntactic categories to CFSs. Finally, section 5 presents our conclusions.

2. The Properties of CFS

We used a training corpus of 59 MB (about 29.5M Chinese characters) in our experiments. In this section, we will present the properties of CFSs. Compared with language models and ASCED, CFSs have some important and distinctive features. We extracted 439,666 CFSs from a training corpus.

2.1 Extracting CFSs from a Training Corpus

The algorithm for extracting CFSs was proposed in our previous work[Lin and Yu 2001]. We extracted CFSs from a training corpus that contained 29.5M characters. The training corpus also included a portion of the Academia Sinica Balanced Corpus [Chen *et al.* 1996] and many Internet news texts.

The length distribution of the CFSs is shown in the second column of Table 1. The total number of CFSs that we extracted was 439,666. Our dictionary, which we call CFSD, is comprised of these 439,666 CFSs. In contrast to the second column of Table 1, we show the length distribution of the words in ASCED in the forth column of Table 1. We found that three-character CFSs were most numerous in our CFS lexicon, while two-character words were most numerous in ASCED. Many meaningful strings and unknown words are collected in our CFSs. These CFSs usually contain more than two characters. Some examples are "小企鵝" (a little penguin), "西醫師" (modern medicine), "佛教思想" (Buddhist thought), "樂透彩券" (lottery), and so on. The above examples cannot be found in ASCED, yet they frequently appear in our training corpus.

2.2 Comparing CFSs with Word-Based N-Gram LMs

Since CFSs are strings frequently used by people, a CFS like "大學教授" (professors of a university) may contain more characters than a word defined in ASCED does. That is, a CFS may contain two or more words. If a CFS contains two words, we say that this CFS is a 2-word CFS. If a CFS contains three words, we say that this CFS is a 3-word CFS and so on. Figure 1 shows the distributions of CFSs according to word-based n-grams. The words are defined in ASCED. We also found 31,275 CFSs(7.11% of the CFSs in CFSD) that are words in ASCED.

From Figure 1, it can be shown that a CFS may contain more than 3 words. Many researchers in Chinese NLP have used word-based bi-gram LMs [Yang 1998] as a basic LM to

solve problems. A very large corpus is required to train a word-based 3-gram LM, while our CFS-based uni-gram model does not need such a large corpus. We also found that a CFS contains 2.8 words on average in CFSD. This shows that a CFS contains more information than a word-based bi-gram LM. In our experiment, we also found that the average number of characters of a word-based bi-gram was 2.75, and that the average number of characters of a CFS was 4.07. This also shows that a CFS contains more information than a word-based bi-gram LM.

Table 1. The length distributions of CFSs in CFSD and words in ASCED.

Number of characters in a CFS or a word	Number of CFSs of that length in our CFS dictionary	Percentage	Number of words of that length in ASCED	Percentage
1	3,877	0.88%	7,745	9.57%
2	69,358	15.78%	49,908	61.67%
3	114,458	26.03%	11,663	14.41%
4	113,005	25.70%	10,518	13.00%
5	60,475	13.75%	587	0.73%
6	37,044	8.43%	292	0.36%
7	19,287	4.39%	135	0.17%
8	11,494	2.61%	66	0.08%
9	6,588	1.50%	3	0.004%
10	4,080	0.93%	8	0.006%

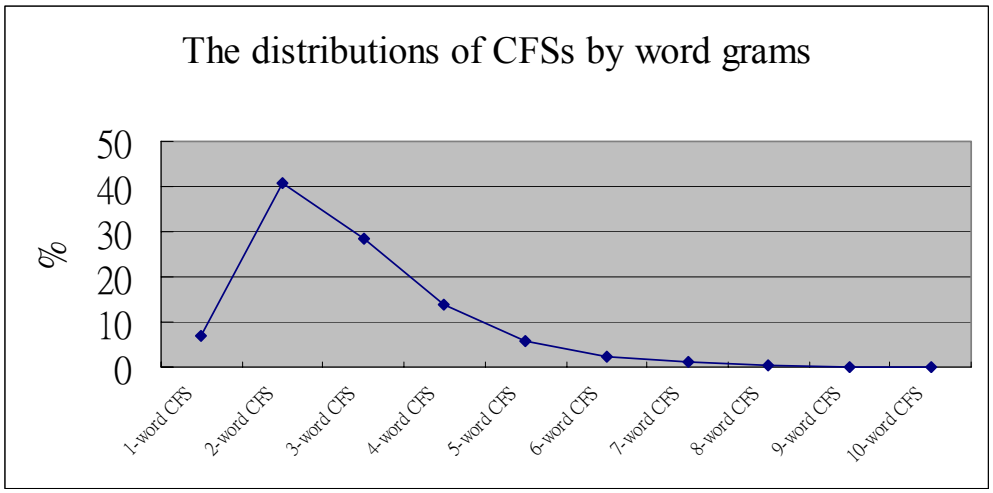


Figure 1. The distributions of CFSs by word-based grams

2.3 Compare the Distributions of CFSs and ASCED

In this subsection, we will make a comparison between our CFSs and ASCED. Table 1 and Figure 2 show the length distributions of our CFSs and ASCED. Comparing them, we find that the average number of characters in a word in ASCED is 2.36, while the average number of characters in a CFS is 4.07. Examining Figure 2, we notice that most of the words in ASCED are 2-character words, while the largest portion of CFSs are 2-character CFSs, 3-character CFSs, 4-character CFSs, and 5-character CFSs. This shows that our CFSs contain many 4-character and 5-character strings. To train character-based 4-gram and character-based 5-gram LMs requires a large training corpus. We also find that the number of one-character CFSs is fewer than that in ASCED. This shows that by using the CFSs, we can eliminate some ambiguities in Chinese PTC and Chinese CTP.

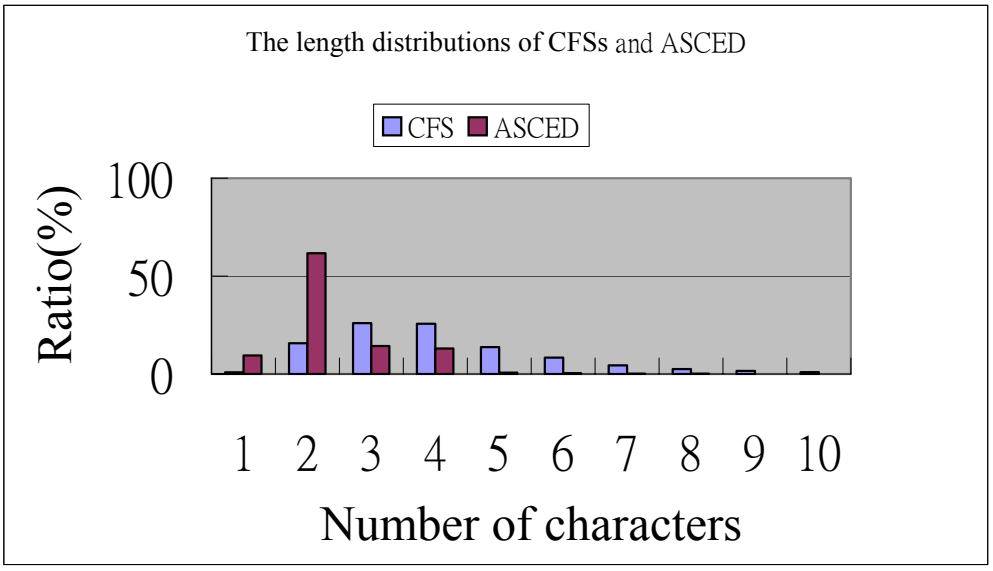


Figure 2. The length distributions of CFSs and ASCED.

We found 31,275 CFSs that were in ASCED. The length distribution of these 31,275 CFSs is shown in Table 2. We also compared the length distribution of these 31,275 CFSs with the length distribution in ASCED. Our comparison is shown in Figure 3. Note that the length distribution in ASCED is listed in the fifth column of Table 1. We find that the length distribution of these 31,275 CFSs is similar to the length distribution in ASCED. We conjecture that if the corpus is large enough, we can find most of the words in ASCED.

Table 2. The length distribution of 31,275 CFSs.

Number of characters in a CFS	Number of CFSs	Percentage
1	3,877	12.40%
2	21,411	68.46%
3	3,742	11.96%
4	2,089	6.68%
5	115	0.37%
6	33	0.105%
7	7	0.022%
8	1	0.003%
9	0	0%
10	0	0%

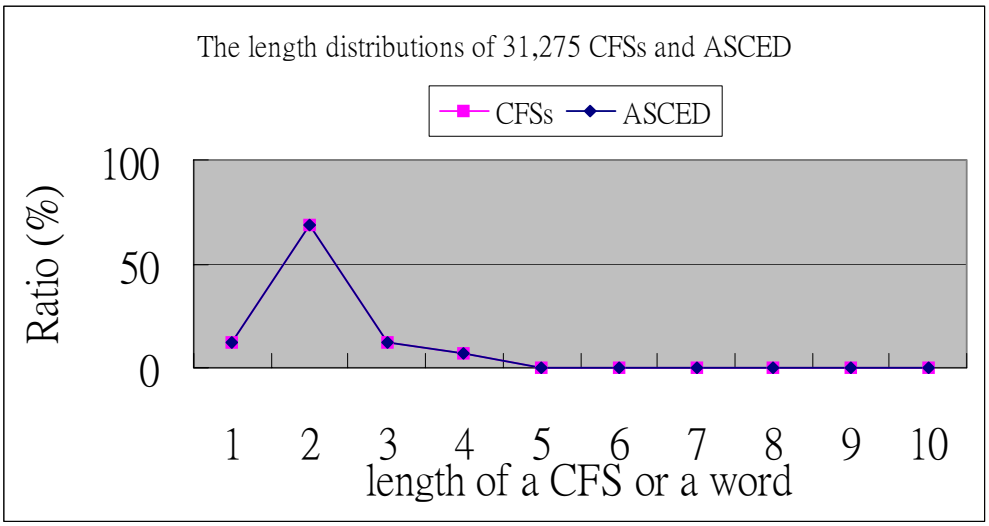


Figure 3. The length distributions of 31,275 CFSs and ASCED.

2.4 Comparing the Normalized Perplexity

Perplexity [Rabiner and Juang 1993] is an important and commonly used measurement of language models. Formula (1) provides a definition of perplexity. Since *Nw*, which is the number of words in the test corpus, in (1) is uncertain for Chinese, we normalize the

perplexity into characters by means of (2) [Yang 1998], producing is called the normalized perplexity (or relative perplexity):

$$PP = \Pr\left(W_1^{Nw}\right)^{-\frac{1}{Nw}}, \tag{1}$$

where $\Pr\left(W_1^{Nw}\right) = \Pr(w_1) \bullet \Pr(w_2) \bullet \ldots \bullet \Pr(w_{Nw}),$

$$NP = PP^{\frac{Nw}{L(W)}}. \tag{2}$$

Here, $W_1^{Nw} = w_1 w_2 \ldots w_{Nw}$ is the test sequence of the corpus and $\Pr(W_1^{Nw})$ is the probability that W_1^{Nw} will be computed within a given language model. *L (W)* is the number of characters in *W*. *PP* is perplexity, and *NP* is the normalized perplexity.

We used a testing corpus to compute the normalized perplexities within the CFS-based uni-gram LMs and the word-based bi-gram LMs. The size of the testing corpus was 2.5M characters. We used the same training corpus mentioned in subsection 2.1 to extract CFSs and to train the word-based bi-gram LMs. Each word in the word-based bi-gram LM was defined in ASCED. We used the Good-Turing smoothing method to estimate the unseen bi-gram events. The normalized perplexity obtained using the word-based bi-gram LM was 78.6. The normalized perplexity became 32.5 when the CFS-based uni-gram LM was used. This shows that the CFS-based uni-gram LM has a lower normalized perplexity. That is to say, using the CFS-based uni-gram LM is better than using the traditional word-based bi-gram LM with a small-sized training corpus of 29.5M characters.

3. Application of CFS to Two Difficult Problems

In a previous study [Lin and Yu 2001], we showed that using CFSs and ASCED as the dictionary with the uni-gram language model can lead to good results in two Chinese NLP applications. These two applications are Chinese character-to-phoneme (CTP) conversion and Chinese phoneme-to-character (PTC) conversion. The achieved accuracy rates were 99.7% for CTP conversion and 96.4% for PTC conversion[Lin and Yu 2001]. The size of the training corpus in our previous research was 0.5M characters. There were 55,518 CFSs extracted from the training corpus. In this study, we solved two challenging Chinese NLP problems with a larger training corpus. The two problems were Chinese toneless phoneme-to-character (TPTC) conversion and Chinese spelling error correction (SEC).

The first task was Chinese TPTC conversion. Chinese TPTC tries to generate correct characters according to input syllables without tonal information. The second task was Chinese SEC (spelling error correction). In our study, we attempted to identify and correct the

possible errors in sentences with no more than one error that were input using the Cang-Jie (倉頡) input method.

3.1 Chinese Toneless Phoneme-to-Character Conversion

The first task was Chinese TPTC conversion. The lexicon we used was CFSD as mentioned in section 2.1. This task is more complex than traditional Chinese phoneme-to-character conversion. There are five tones in Mandarin. They are high-level (1^{st} tone), high-rising (2^{nd} tone), low-dipping (3^{rd} tone), high-falling (4^{th} tone), and the neutral tone [National Taiwan Normal University 1982]. There are a total of 1,244 possible syllables (combinations of phonetic symbols) in Mandarin, and there are a total of 408 possible toneless syllables. Therefore, each toneless syllable has about 1,244/408=3.05 times the number of characters of a tonal syllable. The average length of a sentence in our training corpus is 8 characters per sentence. The number of possibilities for Chinese TPTC conversion is about 3.05^8=7489 times that of Chinese PTC conversion. This shows that Chinese TPTC conversion is more difficult than Chinese PTC conversion.

The size of the outside testing data was 2.5M characters. In our TPTC module, we initially searched the system dictionary to access all the possible CFSs according to the input toneless phonemes. Such possible CFSs constitute a CFS lattice. We applied a dynamic programming methodology to find the best path in the CFS lattice, where the best path was the sequence of CFS-based uni-grams with the highest probability. The definition we employed of the probability $P(S)$ of each input sentence S was as follows:

$$S = CFS_1\ CFS_2\ \ldots\ CFS_n,$$
$$P(S) = P(CFS_1) \bullet P(CFS_2) \bullet \cdots \bullet P(CFS_n), \quad (3)$$

The achieved precision rate was 92.86%. The precision rate was obtained by using the formula (total number of correct characters) / (total number of characters). The processing time was 12 ms/character. We also applied the dictionary used in our previous research [Lin and Yu 2001] to test the data, which was 2.5M characters in size. The dictionary combines ASCDE with 55,518 CFSs. The achieved precision rate in solving the Chinese TPTC problem was 87.3%. This indicates that if we can collect more CFSs, we can obtain higher accuracy.

In this task, we also applied the word-based bi-gram LM with ASCED. The size of the training corpus was the same as that of the corpus mentioned in section 2.1, that is, 29.5M characters. The Good-Turing smoothing method was applied here to estimate the unseen events. The achieved precision rate was 66.9%, and the processing time was 510 ms/character. These results show that when the CFS-based uni-gram LM was used, the precision rate improved greatly (92.8 % vs. 66.9%) and the processing time was greatly reduced (12

ms/character vs. 510 ms/character) compared to the results obtained using the traditional word-based bi-gram LM.

3.2 The Chinese Spelling Error Correction Issue

We also applied the CFS-based uni-gram LM to the Chinese SEC problem [Chang 1994]. Chinese SEC is a challenging task in Chinese natural language. A Chinese SEC system should correct character errors in input sentences. To make the task meaningful in practice, we limited our Chinese SEC problem based on the following constraints: (1) the sentences were input using the Cang-Jie Chinese input method; (2) there was no more than one character error in an input sentence.

The reasons why we applied the above two constraints are as follows: (1) our Chinese SEC system is designed for practiced typists; (2) the Cang-Jie Chinese input method is a popular method widely used in Taiwan; (3) at most one character error is likely to be made in a sentence by a practiced typist; and (4) we can easily apply the methodology used this research to other Chinese input or processing systems. Our methodology for Chinese SEC is shown in Algorithm SEC.

Characters with similar Cang-Jie codes define a confusing set in Algorithm SEC. We constructed the confusing set for each Chinese character based on the five rules listed in Table 3. The longest common subsequence (LCS) algorithm is a well known algorithm that can be found in most computer algorithm textbooks, such as [Cormen *et al.* 1998].

Algorithm SEC.

Input: A sentence S with no more than one incorrect character.

Output: The corrected sentence for the input sentence S.

Algorithm:

Step 1: For each *i-th* character in S, find the characters whose Cang-Jie codes are similar to the code of the *i-th* character. Let C be the set consisting of such characters. C is called the 'confusing set'.

Step 2: Replace each character in C with the *i-th* character in S. There will be a 'maybe' sentence S_1. Find the probability of S_1 by using the CFS-based uni-gram LM. Record the maybe sentence with the highest probability.

Step 3: For each character in S, repeat Step 1 and Step 2.

Step 4: Output the 'maybe' sentence with the highest probability found in Steps 1, 2, and 3.

Table 3. Rules used to construct the confusing set based on the Cang-Jie Chinese input method.

Length of Cang-Jie code: to the target character *t*	Each character *s* satisfying the conditions listed below is a similar character of *t*.
1	The characters whose Cang-Jie codes are the same as that of the target character.
2	A. The length of the Cang-Jie code of *s* is 2, and the length of the LCS of *s* and *t* is 1. B. The length of the Cang-Jie code of *s* is 3, and the length of the LCS of *s* and *t* is 2.
3	The length of the Cang-Jie code of *s* is greater than 1, and the length of the LCS of *s* and *t* is 2.
4	The length of the Cang-Jie code of *s* is greater than 2, and the length of the LCS of *s* and *t* is 3.
5	The length of Cang-Jie code of *s* is 4 or 5, and the length of the LCS of *s* and *t* is 4.

The uni-gram language model was used to determine the probability of each sentence. We used CFSD as our dictionary. There were 485,272 sentences for the outside test. No more than one character in each sentence was replaced with a similar character. Both the location of the replaced character and that of the similar character were randomly selected. The achieved precision rate was 87.32% for the top one choice. The precision rate was defined as (the number of correct sentences) / (the number of tested sentences). The top 5 precision rates are listed in Table 4. The precision rate of the top 5 choices was about 95%, as shown in Table 4. This shows that our approach can provide five possible corrected sentences for users in practice. The achieved precision rate in determining the location of the replaced character with the top one choice was 97.03%.

Table 4. The precision rates achieved using our Chinese SEC and the CFS-based uni-gram LM.

Top n	Precision rate
1	87.32%
2	90.82%
3	92.66%
4	93.98%
5	94.98%

We also applied ASCDE with word-based bi-gram LMs to compute the probability for each possible sentence. The size of the training corpus was 29.5M characters, which was the same as that of the training corpus mentioned in section 2.1. We also used the Good-Turing

smoothing method to estimate the unseen bi-gram events. The achieved precision rates are shown in Table 5. The achieved precision rate for the top one choice was 80.95%.

Table 5. The precision rates achieved using the Chinese SEC and the word-based bi-gram LM.

Top n	Precision rate
1	80.95%
2	82.58%
3	83.31%
4	83.77%
5	84.09%

From Table 4 and Table 5, we can find that using CFS-based uni-gram LM is better than using ASCED with a word-based bi-gram LM. The advantages are the high achieved precision rate (87.32% vs. 80.95%) and short processing time (55 ms/character vs. 820 ms/character).

4. Assigning Syntactic Categories to CFSs

A CFS is a frequently used combination of Chinese characters. It may be a proper noun, like "網際網路" (the Internet), a verb phrase, like "全力動員投入" (try one's best to mobilize), and other word forms. If a CFS can be assigned to some syntactic categories, it can be used in more applications. The CYK algorithm is a well known method used to assign syntactic categories [Lin 1994]. In this study, we tried to assign syntactic categories to CFSs by a using dynamic programming strategy. If a CFS *s* is also a word *w*, we can assign the syntactic categories of *w* to *s*. When *s* is a combination of several words, we can attempt to find syntactic categories associated with it. We first find the probabilities of production rules. Then, we use these probabilities to determine the syntactic categories.

4.1 Extracting Production Rules from Sinica Treebank Version 1.0

We used the Sinica Treebank [Chen *et al.* 1994] as the training and testing data. The contents of the Sinica Treebank are composed of the structural trees of sentences. Structural trees contain the forms of words, the syntactic categories of each word, and the reductions of the syntactic categories of words. Figure 4 shows the structural tree of the sentence "你要不要這幅畫" (Do you want this picture?). The representation of this structural tree in the Sinica Treebank is as follows:

#S((agent:NP(Head:Nhaa:你))|(Head:VE2(Head:VE2:要)|(negation:Dc:不)|(Head:VE2:要))|(goal:NP(quantifier:DM:這幅)|(Head:Nab:畫)))#

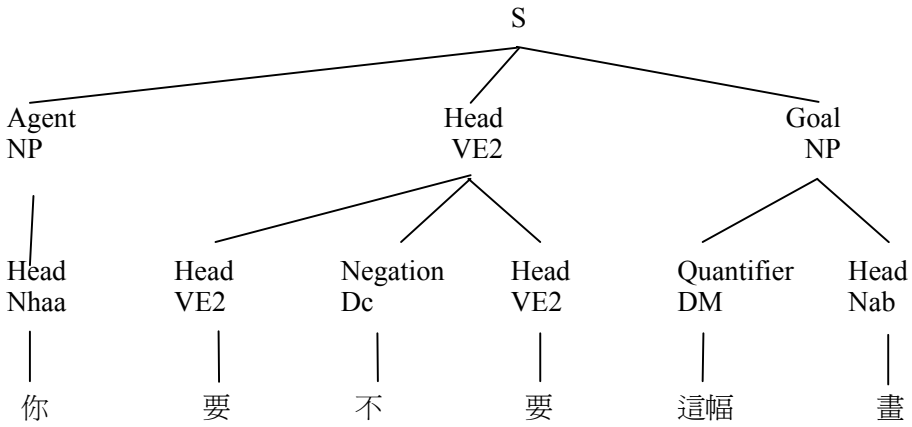


Figure 4. The structural tree of the sentence "你要不要這幅畫" (Do you want this picture?)

There are 38,725 structural trees in the Sinica Treebank version 1.0. They are stored in 9 files. We first used a portion of the 38,725 structural trees as the training data. We wanted to extract the production rules from each structural tree. These production rules were used to determine the syntactic categories of CFSs. Since each CFS could contain one or more words, the syntactic category of a CFS could be a portion of the structural tree. For example, four different production rules were extracted from the structural tree shown in Figure 4. They are "NP←Nhaa", "VE2←VE2+Dc+VE2"," NP←DM+Nab", and "S←NP+VE2+NP". The notations of syntactic categories are defined by the Chinese Knowledge Information Processing group (CKIP).

Examples of probabilities of production rules are listed in Table 6. We extracted 15,946 different production rules from 90% of the Sinica Treebank version 1.0. The other 10% of the structural trees are left for testing.

Table 6. Examples of production rules and their corresponding probabilities.

Rule			Count	Probability
ADV	←	A	1	1
ADV	←	Dbaa	4	1
S	←	Cbaa + S	15	0.9375
VP	←	Cbaa + S	1	0.0625
NP	←	NP + A + Nab	5	1
S	←	Cbba + NP + VJ3	1	0.5
VP	←	Cbba + NP + VJ3	1	0.5
NP	←	NP + VG2 + NP	1	0.008
S	←	NP + VG2 + NP	111	0.941
VP	←	NP + VG2 + NP	6	0.051

4.2 Determining the Syntactic Categories of a CFS

We used the 15,946 production rules to determine the syntactic categories of CFSs. To perform this task, a lexicon with syntactic categories was required for each word. We used ASCED, provided by Academia Sinica, Taiwan, as the dictionary. ASCED is a well-defined dictionary which contains about 80,000 words. For an input CFS, we first looked in ASCED to get the syntactic categories for each substring word of the input CFS. We also used these syntactic categories and the 15,946 production rules to determine the syntactic categories of the input CFS. We tried to find the syntactic categories of a CFS by using the syntactic categories of the substrings of that CFS. The method we used is a dynamic programming method. As an example, Figure 5 shows the syntactic categories of the CFS "林小姐" (Miss Lin).

	1(林)	2(小)	3(姐)
A(林)	Nab, 0.5 Nbc, 0.5	NP, 1	NP, 1
B(小)		VH13, 0.25 V3, 0.25 Nv4, 0.25 VH11, 0.25	Nab, 1
C(姐)			B, 1

Figure 5. The syntactic categories of the CFS "林小姐" (Miss Lin).

As shown in Figure 5, we first looked in ASCED to find the syntactic categories of each possible word which was a substring of "林小姐". Cell (A,1) contains the possible syntactic categories of the word "林", cell (B,2) contains the possible syntactic categories of "小", cell (C,3) contains the possible syntactic categories of "姐", and cell (B, 3) contains the possible syntactic categories of "小姐". The number following each syntactic category in a cell is the probability of that syntactic category.

Next, we tried to determine the syntactic categories of cell (A, 2) by using the production rules we extracted from the Sinica Treebank. The syntactic categories of cell (A, 2) could be derived using the information of cell (A, 1) and cell (B, 2). A total of 2 * 4 = 8 possible production rules were derived. Examining the production rules we extracted, we found that only one of the 8 possible combinations existed in the production rules. This combination was NP ← Nab + Nv4. The result of cell (A, 2) was NP. The probability was 1 because Nab + Nv4 could only derive NP. The contents of cell (B, 3) could also be derived from the contents of cells (B, 2) and (C, 3).

Finally, we determined the syntactic categories of cell (A, 3) in the same way as in the preceding step. The syntactic categories of cell (A, 3) could be derived from cells (A, 1) and (B, 3), or cells (A, 2) and (C, 3) or cells (A, 1) and (B, 2) and (C, 3). The result was NP, which was derived from cell (A,1) and (B,3) by using the rule NP ← Nbc + Nab. The syntactic category of the CFS "林小姐" was NP, which was the only syntactic category derived by inspecting the contents of cell (A, 3).

4.3 Experimental Results

Our goal was to determine the syntactic categories of CFSs. The testing data we chose were in the bottom layer of each structural tree. Each level of the testing data contained many words. For example, we determined the syntactic categories of "要不要" and "這幅畫" as described for the example shown in Figure 4. We found that the syntactic category of "要不要" was VE2, and that syntactic category of "這幅畫" was NP. We retrieved 1,309 patterns and their related syntactic categories from the testing corpus. Among the 1,309 patterns, 98 patterns were our CFSs.

The structure of the notations of the syntactic categories defined by CKIP is a hierarchical one. There are a total of 178 syntactic categories with five layers in the hierarchical tree [CKIP 1993]. There are 8 categories in the first layer: N (noun), C (conjunction), V (verb), A (adjective), D (adverb), P (preposition), I (interjection), and T (auxiliary). The second layer contains 103 syntactic categories. For example, there are two sub-categories, Ca and Cb, in the second layer of category C in the first layer. Seven syntactic categories are defined in the Sinica Treebank. They are S (sentence), VP (verb phrase), NP (noun phrase), GP (direction phrase), PP (preposition phrase), XP (conjunction phrase), and DM (determinate phrase). We also put these 7 syntactic categories in the first layer of the hierarchical tree.

The achieved accuracy rates for determining the syntactic categories of these 98 CFSs by using all of the syntactic categories are shown in Table 7. When we used the syntactic categories in the first layer, the accuracy rate for the top one choice was 70.35%.

Table 7. The accuracy rate for 98 CFSs obtained by using all five layers of syntactic categories.

TOP n	Accuracy
TOP 1	63.26%
TOP 2	78.57%
TOP 3	91.67%
TOP 4	97.62%
TOP 5	97.62%

Because the size of training corpus was small compared with the hundreds of available syntactic categories, we also reduced the tags in each production tree to the second layer of the hierarchical tree. For example, when we reduced the syntactic categories of the production rule "S ← Cbca + NP + Dbb + VK2 + NP" to the second layer, we got the reduced production rule "S ← Cb + NP + Db + VK + NP ". We also determined the syntactic categories of the 98 patterns. The results are shown in Table 8. When we used the syntactic categories in the first layer, the accuracy rate for the top 1 choice was 76.28%.

Table 8. The accuracy rate for 98 CFSs obtained by using the syntactic categories reduced to the 2nd layer.

TOP n	Accuracy
TOP 1	71.02%
TOP 2	84.53%
TOP 3	92.86%
TOP 4	96.43%
TOP 5	98.81%

5. Conclusions

In this paper, we have presented some important properties of Chinese frequent strings. We used CFSs in several applications. We found that the CFS-based uni-gram LM was superior to traditional N-gram LMs when the training data was sparse. While the size of a corpus using the CFS-based uni-gram LM can be far smaller than that needed when using traditional N-gram LMs, for the applications studied here, the results obtained using the CFS-based uni-gram LM are better than those obtained using an n-gram LM.

Acknowledgements

We would like to thank Academia Sinica for providing its ASBC corpus, ASCED dictionary, and Sinica Treebank. We also extend our thanks to the many news companies for distributing their files on the Internet.

References

C. H. Chang, "A Pilot Study on Automatic Chinese Spelling Error Correction," *Communication of COLIPS*, Vol. 4, No. 2, 1994, pp. 143-149.

K. J. Chen, C. R. Huang, L. P. Chang, and H. L. Hsu, "SINICA CORPUS: Design Methodology for Balanced Corpora," *Proceeding of PACLIC 11th Conference*, 1996, pp. 167-176.

F. Y. Chen, P. F. Tsai, K. J. Chen, and C. R. Huang, "Sinica Treebank," *Computational Linguistics and Chinese Language Processing,* Vol. 4, No. 2, 1994, pp. 75-85.

CKIP(Chinese Knowledge Information Processing Group, 詞庫小組) , "Analysis of Chinese Part-of-Speech (中文詞類分析), Technical Report of CKIP #93-05(中文詞知識庫小組技術報告 #93-05)," Academia Sinica, Taipei, Taiwan, 1993.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to Algorithms," The MIT Press, 1998.

F. Jelinek, "Self-organized Language Modeling for Speech Recognition," Readings in Speech Recognition, Ed. A. Wabel and K. F. Lee. Morgan Kaufmann Publishers Inc., San Mateo, California, 1990, pp. 450-506.

Y. C. Lin, "A Level Synchronous Approach to Ill-formed Sentence Parsing and Error Correction," Ph.D. Thesis, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, June 1994.

Y. J. Lin and M. S. Yu, "Extracting Chinese Frequent Strings Without a Dictionary From a Chinese Corpus And its Applications," *Journal of Information Science and Engineering*, Vol. 17, No. 5, 2001, pp. 805-824.

National Taiwan Normal University, "Mandarin Phonetics," National Taiwan Normal University Press, Taipei, Taiwan, 1982.

L. R. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition," Prentice Hall Co. Ltd., 1993.

B. Suhm and A. Waibel, "Toward Better Language Models for Spontaneous Speech," Proc. ICSLP, 1994, pp. 831-834.

Jian Wu and Fang Zheng, "On Enhancing Katz-Smoothing Based Back-Off Language Model," *International Conference on Spoken Language Processing,* 2001, pp. I-198-201.

K. C. Yang, "Further Studies for Practical Chinese Language Modeling," Master Thesis, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, June 1998.