A Study of Applying BTM Model on the Chinese Chunk Bracketing

Jia-Lin Tsai
Tung Nan Institute of Technology, Department of Information Management
Taipei 222, Taiwan, R.O.C.
tsaijl@mail.tnit.edu.tw

Abstract

The purpose of this paper is to automatically generate Chinese chunk bracketing by a bottom-to-top mapping (BTM) model with a BTM dataset. The BTM model is designed as a supporting model with parsers. We define a word-layer matrix to generate the BTM dataset from Chinese Treebank. Our model matches auto-learned patterns and templates against segmented and POS-tagged Chinese sentences. A sentence that can be matched with some patterns or templates is called a matching sentence. The experimental results have shown that the chunk bracketing of the BTM model on the matching sentences is high and stable. By applying the BTM model to the matching sentences and the N-gram model to the non-matching sentences, the experiment results show the F-measure of an N-gram model can be improved.

1 Introduction

The definition of chunk, which has been represented as groups of words between square brackets, was first raised by (Abney, 1991). A chunker is to divide sentences into non-overlapping phrases by starting with finding correlated chunks of words. Text chunking has been shown a useful pre-processing step for language parsing (Sang and Buchholz, 2000). Among the chunk types, NP chunking is the first to receive the attention (Ramshaw and Marcus, 1995), than other chunk types, such as VP and PP chunking (Veenstra, 1999). For English (Sang and Buchholz, 2000) and Chinese (Li et al., 2004) languages, the top 3 most frequent chunk types are NP, VP and PP chunks. Meanwhile, the three chunk types cover about 80% of chunking problems. In many natural language processing (NLP) applications, such as information retrieval, knowledge discovery, example-based machine translation (EBMT) and text summarization, can benefit with chunks (Le et al., 2003; Munoz et al., 1999; Oliver, 2001; Zhou and Su, 2003).

As per the reports (Menzel, 1995; Sang and Buchholz, 2000; Basili and Zanzotto, 2002; Knutsson et al., 2003; Li et al., 2004; Xu et al., 2004; Johnny et al., 2005), there are three important trends in the study of Chinese text chunking and parsing. These important trends are: (1) ***Treebank-Derived Approaches*** for auto-constructing useful patterns and templates from Treebank (TB) as rules combined with statistical language models (SLM), such as N-gram models and support vector machines (SVMs), etc.; (2) ***Robust Chunkers*** against Treebank sparseness and perfect/actual input. Here the *perfect input* means the word-segmentation and Part-of-Speech (POS) tags all are correct. The *actual input* means the word-segmentation and POS tags all are generated by a selected segmenter and a POS tagger; and (3) ***High Performance Chunk Bracketing*** has been reported that the key issue of Chinese parsing (Li et al., 2004). To sum up these trends, one of critical issues for developing a high performance Chinese chunker is to find methods to achieve high performance of chunk bracketing against training size, perfect and actual input.

Following these trends of Chinese chunking and parsing, the goals of this paper are:

(1) *Define a* ***Word-Layer Matrix*** *and generate the* ***Bottom-to-Top Mapping (BTM)*** *dataset* to auto-derive useful patterns and templates with probabilities from Chinese Treebank

(CTB) as rules for chunking;

(2) *Develop a **BTM model*** with the BTM dataset to identify the chunks (i.e. phrase boundaries) for a given segmented and POS-tagged Chinese sentence;

(3) *Show the chunk bracketing performance of the BTM model is high and stable* against training corpus size, perfect and actual input;

(4) *Show the BTM model can improve* the performance (F-measure) of N-gram models on chunk bracketing.

The remainder of this paper is arranged as follows. In Section 2, we present the BTM model for identifying chunks for each segmented and POS-tagged Chinese sentence. Experimental results and analyses of the BTM model are presented in Section 3. Finally, in Section 4, we present our conclusions and discuss the direction of future research.

2 Development of the BTM model

2.1 Introduction of Chinese Treebank

A Chinese Treebank (CTB) is a segmented, POS-tagged and fully bracketed Chinese corpus with morphological, syntactic, semantic and discourse structures. The CKIP (Chinese Knowledge Information Processing) Chinese-Treebank (CCTB) and the Penn Chinese Treebank (PCTB) are two of most important Chinese Treebank resources for Treebank-derived NLP tasks in Chinese (CKIP, 1995; Xia et al., 2000; Xu et al., 2000; Li et al., 2004). The brief introductions of the CCTB and the PCTB are given as below (Table 1 is a brief comparison between the CCTB and the PCTB):

(1) ***CCTB***: the CCTB is developed in traditional Chinese texts (BIG5 encoded) taken from the Academia Sinica Balanced Corpus 3.0 (ASBC3) at the Academia Sinica, Taiwan (Chen et al., 1996; Chen et al., 1999; Huang et al., 2000, Chen et al., 2003; Chen et al., 2004). The CCTB uses Information-based Case Grammar (ICG) as the language framework to express both syntactic and semantic descriptions (Chen and Huang, 1996). The structural frame of CCTB is based on the Head-Driven Principle: it means a sentence or phrase is composed of a core Head and its arguments, or adjuncts (Chen and Hsieh, 2004). The Head defines its phrasal category and relations with other constituents. The present version CCTB2.1 (CCTB Version 2.1) includes 54,902 sentences (i.e. trees) and 290,144 words that are bracketed and post-edited by humans, based on the computer parsed results (CKIP, 1995). There are 1,000 CCTB trees open to the public for researchers to download on the CCTB portal. The details of supplementary principles, symbol illustrations, semantic roles, phrasal structures and applications of the CCTB can be found in (CCTB portal; Chen et al., 2003; Chen and Hsieh, 2004; You and Chen, 2004).

Table 1. A brief comparison between CCTB2.1 and PCTB4 (The number in () is the word frequency and the English word in [] is the English Translation for the corresponding Chinese word)

	CCTB2.1	PCTB4
Developer	CKIP	UPenn
Content type	Balanced corpus	Newswire sources
Language framework	ICG	HPSG
Word standard	Taiwan (CKIP, 1996)	China (Liu et al.,1993)
POS-tagging system type	hierarchical (5 layer)	non-hierarchical
Structure frame	Head-driven	Head-driven
Code	BIG5	GB
No. of sentences	54,902	15,162
No. of distinct POS tags	302	47
No. of words in CTB	290,144	404,156
Top 3 one-char words	的(19,212) [of]	的(15,080) [of]
	是(4,608) [is/are]	在(4,055) [at]
	在(4,235) [at]	是(2,965) [is/are]
Top 3 two-char words	我們(1,057) [we]	中国(2,097) [China]
	一個(675) [a/an/one]	经济(1,015) [Economy]
	他們(564) [they]	企业(989) [business]

(2) ***PCTB***: the PCTB is developed in simplified Chinese texts (GB encoded) taken from the newswire sources (consists of Xinhua newswire, Hong Kong news and Sinorama news magazine, Taiwan) at the Department of Computer and Information Science, University of Pennsylvania (UPenn). The PCTB uses Head-driven Phrase Structure Grammar (HPSG) to create Chinese texts with syntactic bracketing (Xia et al, 2000; Xue et al, 2002). Meanwhile, the semantic annotation of PCTB mainly deals with the predicate-argument structure of Chinese verbs in Penn Chinese Proposition Bank (Xue and Palmer, 2003;

Xue and Palmer, 2005). The present version PCTB5 (PCTB Version 5), contains 18,782 sentences, 507,222 words, 824,983 Hanzi and 890 data files. The PCTB was created by two pass approach. The first pass was done by one annotator, and the resulting files were checked by a second annotator (the second pass). The details and applications of PCTB can be found in (PCTB portal; Xia et al, 2000; Chiou et al, 2001; Xue et al, 2002; Xue et al, 2005).

Overall, from Table 1, the four major differences between the CCTB and the PCTB are content type, language framework, word standard and POS-tagging system type. The CCTB is natural to be a balanced CTB because its content is taken from the Academia Sinica Balanced Corpus (CKIP, 1995). On the other hand, since the content type of PCTB is newswire sources, it is natural to be a newswire-based CTB and not a balanced CTB.

2.2 Generating the BTM Dataset

Firstly, we use CCTB2.1 as an example to describe how to generate a BTM dataset from the CCTB with the word-layer matrix. Then, we define two types of conditional probabilities used in this study for constructing the BTM model. Finally, the algorithm of our BTM model is given in Section 2.3.

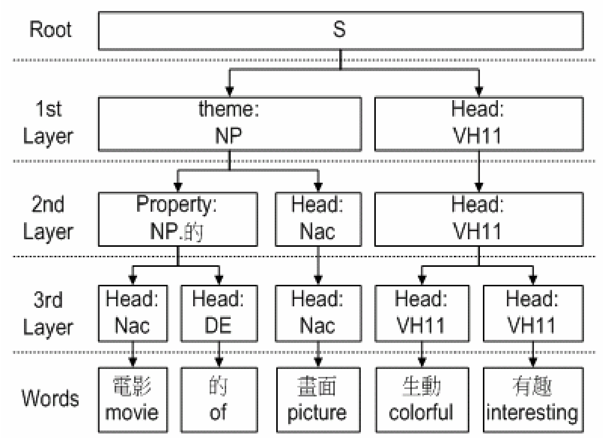

Figure 1. The tree structure of CCTB2.1 for the Chinese sentence “電影(movie)的(of)畫面(picture)生動(colorful)有趣(interesting)” (Note that the content of the nodes between the root and the words is [Thematic role : Syntactic category])

(1) ***Generation of BTM dataset from CCTB2.1***: Figure 1 shows the tree structure of CCTB2.1 for the Chinese sentence “電影(movie)的(of)畫面 (picture) 生 動 (colorful) 有 趣 (interesting).” The content of the nodes between the root layer and the words layer (leaves) is comprised of thematic roles and syntactic categories. The thematic roles can be annotated as a **Theme**, **Property**, etc., while the syntactic categories can be annotated as a POS-tag (such as **Nac**) or a phrasal category (such as **NP**). The details of CCTB syntactic and thematic annotations can be found in (Chen et al., 2003).

Table 2. The word-layer matrix extracted from CCTB2.1 for the Chinese sentence “電影(movie)的(of)畫面(picture)生動(colorful)有趣(interesting)”

Word	1st layer (Top)	2nd layer	3rd layer (Bottom)
電影	Theme:NP	Property:NP.的	Head:Nac
的			Head:DE
畫面		Head:Nac	Head:Nac
生動	Head:VH11	Head:VH11	Head:H11
有趣			Head:H11

Table 3. The BTM dataset for the CCTB2.1 tree of the Chinese sentence “電影(movie)的(of)畫面(picture)生動(colorful)有趣(interesting)”

Type	Content
BL Word pattern	<電影\的\畫面\生動\有趣>
TL Word pattern	<電影:的:畫面+生動:有趣>
BL POS pattern	<Nac\DE\Nac\VH11\VH11>
TL POS pattern	<Nac:DE:Nac+VH11:VH11>
TL POS template	<Na%Na+VH11:VH11>
PC pattern	<NP+VH11>

For each tree structure of CCTB2.1 (as shown in Fig.1), we first translate it into a word-layer matrix as shown in Table 2. In a word-layer matrix, the left, first row is the word layer (with words) and the other rows are the first layer to the last layer (with thematic roles and syntactic categories). For each word-layer matrix, the first layer and last layer are called the **Top Layer** (TL) and the **Bottom Layer** (BL), respectively. According to the TL and the BL of a word-layer matrix (see Table 2), we can translate a CCTB tree into a BTM dataset as shown in Table 3. Each BTM dataset includes two types of BTM content. One is the BL and TL word patterns expressed by Chinese words. The other one is the BL and TL POS patterns expressed by POS tags. Furthermore, for each TL

POS pattern, we also generate its corresponding TL POS template (with POS tags) and phrasal category (PC) pattern (with POS tags and phrasal categories).

In the Table 3:

BL stands for the bottom layer (the last layer of a word-layer matrix);
TL stands for the top layer (the first layer of a word-layer matrix);
PC stands for the phrasal category in a TL;
"\" indicates the word boundary in a BL;
"+" indicates word/phrase boundaries in a TL;
"**:**" indicates next to; for example,
"Nac:DE" means "**Nac**" next to "**DE**";
"**%**" indicates near by; for example,
"Nac%Nac" means "**Nac**"near by "**Nac**";
"<" indicates the begin of a sentence; and
">" indicates the end of a sentence.

The CKIP POS tagging is a hierarchical system. The first layer of CKIP POS tagging include eight main syntactic categories, i.e. N (noun), V (verb), D (adverb), A (adjective), C (conjunction), I (interjection), T (particles) and P (preposition). As per the CKIP technical reports (CKIP, 1995; CKIP, 1996), the maximum layer number of CKIP POS tagging is 5. Take the CKIP POS tag "**Ndabe**" as an example, we define its POS tags with POS layer numbers 1, 2, 3, 4 and 5 as "**N**", "**Nd**", "**Nda**", "**Ndab**" and "**Ndabe**", respectively. Thus, if the POS layer of BTM model is set to 2 (called 2 POS-layer mode), the BL POS pattern "<Nac\DE\Nac\VH11\VH11>" in Table 3 will become "<Na\DE\Na\VH\VH>", and so forth.

Table 4. The BTM dataset for the PCTB4 tree of the Chinese sentence "双方(both)主要(major)代表(agent)出场(appear)"

Type	Content
BL Word pattern	<双方\主要\代表\出场>
TL Word pattern	<双方:主要:代表+出场>
BL POS pattern	<PN\JJ\NN\VV>
TL POS pattern	<PN:JJ:NN+VV>
TL POS template	<PN%NN+VV>
PC pattern	<NP-SBJ+VP>

By the word-layer matrix, the BTM dataset of PCTB can also be generated. Table 4 shows an example BTM dataset for the PCTB4 tree of the Chinese sentence "双方(both)主要(major)代表(agent)出场(appear)." Since the POS tagging of PCTB is not a hierarchical system, there is no POS layer mode can be set to the BTM dataset of PCTB.

(2) ***Definitions of Two Types of Probabilities***: In this study, two conditional probabilities were used in the BTM model. The **Type I** conditional probability is used to perform full TL POS pattern matching. The **Type II** conditional probability is used to perform full TL POS template matching. Details of these probabilities are given below.

Type I. Pr(a given TL POS pattern | the BL POS pattern of the given TL POS pattern) =
(# of the given TL POS pattern found in the training BTM dataset) /
(# of the BL POS patterns of the given TL POS pattern found in the training BTM dataset).

Take the BL POS pattern "Cb\Nc\DE\Na" as an example. There are:
one TL POS pattern "Cb+Nc:DE:Na"
four TL POS pattern "Cb+Nc:DE+Na" and
five BL POS pattern "Cb\Nc\DE\Na" in the CCTB2.1 BTM dataset. Thus,
the Pr(Cb+Nc:DE:Na|Cb+Nc+DE+Na) = 1/5 = 0.2; and
the Pr(Cb+Nc:DE+Na|Cb+Nc+DE+Na) = 4/5 = 0.8.

Table 5a. Top 5 most frequent TL POS patterns whose number of POS tags is 5 for 2 POS-layer mode (training size is 45,000 CCTB2.1 trees)

TL POS pattern (Type I pro.)
V_+DM:VH:DE:Na (19/19 = 100%) (Eg. 是 [is]+ 一 個 [a]: 小 小 [small]: 的 [of]: 村 子 [village])
Nb:A:Na:Nb+VE (11/11 = 100%) (Eg. 泰 源 [taiyuan]: 投 顧 [inference reader]: 經 理 [manager]:高子能[gao-zi-neng]+指出[point out])
Nc:Nb+VH+Nc:Nb (10/10 = 100%) (Eg. 費城[Philadelphia]:七六人隊[76-people team]+勝[win]+華盛頓[Washington]:子彈隊[bullet team]))
VC+Di+Na:DE:Na (9/9 = 100%) (Eg. 搭 [attach]+ 起 [to]+ 友 誼 : 的 : 橋 樑 [bridge of friendship])
Nh+VA+P2:Na:Nc (8/8 = 100%) (Eg. 他們[they]+坐[sit]+在:太空船:裡[at the aerospace plane])

Table 5a gives the Top 5 most frequent TL POS patterns whose number of POS tags is 5 while the POS layer number is 2.

Type II. Pr(matching patterns | a given TL POS template) =
(# of matching TL POS patterns of the given TL POS template found in the training BTM dataset) / (# of matching BL POS pattern of the given TL POS template found in the training BTM dataset).

Take the TL POS template “P3%Na+VA” as an example. In the CCTB2.1 BTM dataset, there are four matching BL POS patterns and two matching TL POS pattern for the template “P3%Na+VA”, namely: (Note that “%” means “near by”)

“P3\Dd\VA\DE\Na\VA”
“P3:Dd:VA:DE:Na+VA”(matching)

“P3\Na\P2\VC\Nb\Nc\DE\Na\VA”
“P3:Na+P2:VC:Nb:Nc:DE:Na+VA”(no matching)

“P3\Na\VA”
“P3:Na+VA”(matching)

“P3\Na\VC\Na\VA”
“P3:Na+VC+Na+VA”(no matching)

Thus, the Pr(matching pattern|P3%Na+VA) = 2/4 = 0.5.

Table 5b gives 5 randomly selected TL POS templates where their POS number is 5 while the POS layer number is 2.

Table 5b. Five randomly selected TL POS templates where their POS number is 5 for 2 POS-layer mode (training size = 45,000)

TL POS template	Type II pro.
Na+VF+Nh+VA%Na	100% (1/1)
P1%Nc+VC+Nc%Na	50% (1/2)
Ne+Dd+V_+VH%Na	100% (3/3)
DM%Na+VE+Nc%Na	100% (1/1)
DM%Na+VK+VC%Na	100% (1/1)

2.3 Algorithm of the BTM Model

Following is the algorithm of our BTM model uses Types I (full TL POS pattern matching) and Type II (full TL POS template matching) conditional probabilities to determine the chunks for a given segmented and POS-tagged Chinese sentence. We use BTM (*value1, value2, value3*) to express the function of our BTM model, where *value1* is the BTM threshold value, *value2* is the POS layer number and *value3* is the BTM training size. Table 6 is a step by step example to demonstrate the detailed processes and outputs of our BTM model.

Table 6. A step by step example of the application of BTM (0.5; 2; 45,000) for the given BL POS pattern “Na\Na\DE\Nb(少年[boy]\時代[age]\的[of]\史懷哲[schweitzer])”

Step	Output
1	Na\Na\DE\Nb (少年[boy]\時代[age]\的[of]\史懷哲[schweitzer])
2	NULL; Goto Step 4
3	-
4	Pr(Na%DE+Nb) = 66.7% (Type II); and use the selected TL POS template “Na%De+Nb” to translate “Na\Na\DE\Nb” into “Na:Na:DE+Nb”
5	TL POS pattern = Na:Na:DE+Nb, and Matching sentence = 少年:時代:的+史懷哲 Chunks = “少年時代的” and “史懷哲”

Step 1. Give the *value1* (BTM threshold value), *value2* (POS layer number) and *value3* (training size), as well as the segmented and POS-tagged sentence. In the following steps, the POS tagging sequence of the given sentence is called the BL POS pattern, such as the “Na\Na\DE\Nb” in Table 6.

Step 2. According to the BL POS pattern in Step 1, find all matched TL POS patterns whose corresponding **Type I** probabilities are greater than or equal to the BTM threshold value. If the number of matched TL POS patterns is zero, then go to Step 4.

Step 3. Using the matched TL POS patterns from Step 2, select the TL POS pattern that has the maximum **Type I** probability as the output. If there are two or more TL POS patterns with the same maximum **Type I** probability, randomly select one as the output. Go to Step 5.

Step 4. According to the BL POS pattern, find all matched TL POS templates whose corresponding **Type II** probabilities are greater than or equal to the BTM threshold value. Select the TL POS template that has the maximum **Type II** probability to generate the output (see Table 6, Step 4). If there are two or more TL POS templates with the same maximum **Type II** probability, randomly select one to generate the output. If the number of matched TL POS patterns is zero, then a NULL output will be given.

Step 5. Stop. If a NULL TL POS pattern output is given, this input sentence is a *non-matching sentence*. Otherwise, it is a *matching sentence*.

3 Experiment Results

To conduct the following experiments in ten-folds, we randomly select 50,000 trees of CCTB2.1 and separate them into the following two sets:

(1) **Training Set** consists of 45,000 CCTB2.1 trees; and

(2) **Open Testing Set** consists of the other 5,000 CCTB2.1 trees.

In our computation, 66% of CCTB2.1 BL POS patterns in the open testing set are not found in the training set. This means the ratio of unseen CCTB2.1 BL POS patterns in the open testing set is 66%. The PCTB4 BTM dataset was not used in this study by two reasons: the first one is that the PCTB is not a balanced CTB; the second one is that the POS tagging system of PCTB is not a hierarchical system.

We conducted four experiments in this study. The first three experiments are designed to show the relationships between the *chunk bracketing performance of the BTM model* on the matching sentences and the three *BTM parameters*: POS layer number; BTM threshold value; and BTM training size. To avoid the error propagation of word segmentation and POS tagging, the first three experiments only consider open testing sentences with correct word segmentations and POS tags provided in CCTB2.1 as ***perfect input***. The fourth experiment is to show the BTM model is able to improve the performance (F-measure) of N-gram models on Chinese chunk bracketing for both perfect input and actual input. Here, the ***actual input*** means the word segmentations and POS tags of the testing sentences were all generated by a forward maximum matching (FMM) segmenter and a bigram-based POS tagger, respectively.

To evaluate the performance of our BTM model, we use recall (R), precision (P), and F-measure (F) (Manning and Schuetze, 1999), which are defined as follows:

Recall (R) = (# of correctly identified chunk brackets) / (# of chunk brackets) (1)

Precision (P) = (# of correctly identified chunk brackets) / (# of identified chunk brackets) (2)

F-measure (F) = (2 × recall × precision) / (recall + precision) (3)

In addition, we use coverage ratio (CR) to represent the size of matching sentences (or say, matching set) of our BTM model. The CR is defined as:

Coverage Ratio (CR) = (# of not NULL output sentences) / (# of total testing sentences) (4)

3.1 Relationship between POS layer number and BTM performance

In the 1_{st} experiment, the BTM threshold value is set to 1 and the BTM training size is set to 45,000. Table 7 is the first experimental results of BTM performance (P, R, F) and CR for the POS layer numbers are 1, 2, 3, 4 or 5. From Table 7, it shows the POS layer number is positively related to the F-measure. Since the BTM model with POS layer number 2 is able to achieve more than 96% F-measure, we use POS layer number 2 to conduct the following experiments. This experimental result seems to indicate that the CCTB2.1 dataset with POS layer number 2 (including 57 distinct POS tags) can provide sufficient information for the BTM model to achieve an F-measure of more than 96% and a maximum CR of 46.88%.

Table 7. The first experimental results of BTM (1; 1/2/3/4/5; 45,000)

POS Layer #	P(%)	R(%)	F(%)	CR(%)
1	86.32	85.57	85.94	33.43
2	97.03	95.82	96.42	46.88
3	99.04	98.86	98.95	34.07
4	99.07	98.88	98.97	31.92
5	99.07	98.87	98.97	31.84

3.2 Relationship between BTM threshold value and BTM performance

In the 2_{nd} experiment, the POS layer number is set to 2 and the BTM training size is set to

45,000. Table 8 is the second experimental results of BTM performance and CR when the BTM threshold value is 1.0, 0.9, 0.8, 0.7, 0.6 or 0.5. From Table 8, it shows the BTM threshold value is positively related to the F-measure. Besides, the F-measure difference between threshold values 1.0 and 0.5 is only 1.37%. This result indicates that the BTM model can robustly maintain an F-measure of more than 95% and a CR of more than 46% while the POS layer number is set to 2, BTM training size is set to 45,000 and the BTM threshold value is ≥ 0.5.

Table 8. The second experimental results of BTM (1/0.9/0.8/0.7/0.6/0.5; 2; 45,000)

Threshold Value	P(%)	R(%)	F(%)	| CR(%)
1.0	97.03	95.82	96.42	| 46.88
0.9	96.99	95.71	96.35	| 47.84
0.8	96.95	95.54	96.24	| 49.42
0.7	96.94	95.49	96.21	| 50.66
0.6	96.86	95.26	96.05	| 51.92
0.5	96.34	93.80	95.05	| 53.72

3.3 Relationship between BTM training size and BTM performance

In the 3_{rd} experiment, the BTM threshold value is set to 0.5 and POS layer number is set to 2. Table 9 is the third experimental results of BTM performance and CR when the BTM training size is 5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000 or 45000. From Table 9, it seems to indicate that the F-measure of the BTM model is independent of the training size because the maximum difference between these respective F-measures is only 0.88%.

Table 9. The third experimental results of BTM (0.5, 2, 5,000/ 10,000/ 15,000/ 20,000/ 25,000 /30,000 /35,000 / 40,000/ 45,000)

Training size	P(%)	R(%)	F(%)	| CR(%)
5,000	95.82	91.33	94.61	| 30.23
10,000	96.28	92.16	94.64	| 35.13
15,000	96.29	92.48	94.54	| 40.32
20,000	96.07	92.59	94.44	| 43.73
25,000	96.19	92.74	94.43	| 46.70
30,000	96.17	92.78	94.30	| 48.46
35,000	96.22	92.92	94.35	| 50.51
40,000	96.28	93.06	94.17	| 52.29
45,000	96.34	93.80	95.05	| 53.72

To sum up the above three experimental results (Tables 7-9), it shows that the F-measure (overall performance) of our BTM model with POS layer number (≥ 2) is apparently not sensitive to BTM threshold value (≥ 0.5) and BTM training size (≥ 5,000) on the matching set with perfect input. Since the CR of our BTM model is positively related to BTM training size, it indicates our BTM model should be able to maintain the high performance chunk bracketing (more than 95% F-measure on the matching set with perfect input) and increase the CR only by enlarging the BTM training size.

3.4 Comparative study of the N-gram model and the BTM model on perfect/actual input

To conduct the 4_{th} experiment, we develop N-gram models (NGM) by the SRILM (Stanford Research Institute Language Modeling) toolkit (Stolcke, 2002) as the baseline model. SRILM is a freely available collection of C++ libraries, executable programs, and helper scripts designed to allow both production of, and experimentation with, statistical language models for speech recognition and other NLP applications (Stolcke, 2002). In this experiment, the TL POS patterns (such as "<Na:DE:Na+VH:VH>") of training set were used as the data for SIRLM to build N-gram models. Then, use these N-gram models to determine the chunks for each BL POS pattern in the testing set. Note that these N-gram models were trained by the TL POS patterns only, not by each layer's POS patterns. Figure 2 shows the distribution of n-gram patterns of N-gram models (N is from 2 to 44) trained by the training set.

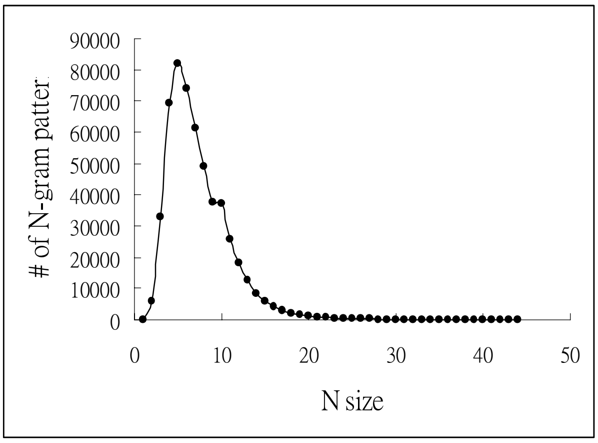

Fig.2 The n-gram distribution of N-gram models (N is 2 to 44) trained by the 45,000 CCTB2.1 TL POS patterns of training set

Tables 10, 11, 12 13 and 14 are the results of the fourth experiment. The explanations of the five tables are given below.

Table 10. The fourth experimental results of NGM (2/3/4/5/6/N; 45,000) for perfect input

N-gram	P(%)	R(%)	F(%)
2	77.62	78.98	78.30
3	80.16	81.83	80.99
4	80.36	81.91	81.13
5	79.58	81.38	80.47
6	78.98	80.35	79.91
N (=44)	78.51	80.44	79.46

Table 11. The comparative experimental results of P/R/F and CR between BTM (0.5, 2, 45,000) and a 4-gram model for perfect input

Set	BTM(0.5, 2, 45k)	4-gram	| CR(%)
matching	96.3/93.8/95.1	89.3/89.7/89.5	| 53.6
no matching	-	73.5/75.7/74.6	| 46.4

Table 12. The comparative experimental results of P/R/F and CR between BTM (0.5, 2, 45,000) and a 4-gram model for actual input

Set	BTM(0.5, 2, 45k)	4-gram	| CR(%)
matching	97.4/97.3/97.3	95.1/96.7/95.9	| 19.2
no matching	-	69.1/68.8/68.9	| 80.8

Table 13. The comparative experimental results of P/R/F of a 4-gram model and a 4-gram with BTM (0.5, 2, 45,000) model for perfect input and actual input

Model	Perfect(P/R/F)	Actual(P/R/F)
4-gram	80.4/81.9/81.1	72.63/72.23/72.43
BTM+4-gram	83.8/83.4/83.6	74.70/73.03/73.42

From Table 10, it shows the maximum precision, recall and F-measure of N-gram models all occur at the 4-gram model for perfect input. Thus, we use the 4-gram model as the baseline model in this experiment. Tables 11 and 12 are the comparative experimental results of the baseline model and the BTM model on the matching sets of perfect input and actual input, respectively. From Table 11, it shows the performance (95.1% F-measure) of a BTM (0.5, 2, 45,000) is 5.6% greater than that of a 4-gram model (89.5% F-measure) for the matching set with perfect input. From Table 12, it shows the performance (97.3% F-measure) of a BTM (0.5, 2, 45,000) is 1.4% greater than that of a 4-gram model (95.9% F-measure) for the matching set with actual input. Table 13 is the experimental results of applying the BTM model to the matching set and the 4-gram model to the non-matching set. From Table 13, it shows the F-measure of a 4-gram model can be improved by the BTM model for both perfect input (2.5% increasing) and actual input (1% increasing).

According to all the four experimental results, we have: (1) the BTM model can achieve better F-measure performance than N-gram models on the matching sets for both perfect input and actual input; and (2) the chunk bracketing performance of the BTM model for the matching sets should be high and stable against training size, perfect and actual input while POS layer number ≥ 2 and BTM threshold value ≥ 0.5.

4 Conclusion and Future Directions

In this paper, we define a word-layer matrix that can be used to translate the CKIP Treebank and the Penn Chinese Treebank into corresponding BTM datasets. By the BTM dataset, we developed a BTM model, adopting two types of conditional probabilities and using full TL POS pattern matching and full TL POS template matching to identify the chunks for each segmented and POS-tagged Chinese sentence.

Our experiment results show that the BTM model can effectively achieve precision and recall optimization on the matching sets for both perfect input and actual input. The experimental results also demonstrate that:

(1) The BTM threshold value is positively related to the BTM F-measure;

(2) The POS layer number is positively related to the BTM F-measure;

(3) The F-measure of our BTM model for the matching set should be not sensitive to two BTM parameters: BTM threshold value and BTM training size;

(4) The chunk bracketing of our BTM model on the matching set should be high and stable (or say, robust) against training size, perfect and actual input while POS layer number is ≥ 2 and BTM threshold value is ≥ 0.5;

(5) The BTM model can provide a matching set with high and stable performance (more than 95% F-measure) for improving N-gram-like models without trial-and-error, or say, a tuning process. For most statistical language models, such N-gram models, need tuning to improve their performance and large-scale corpus to

overcome corpus sparseness problem (Manning et al., 1999; Gao et al., 2002; Le et al., 2003). Furthermore, it is difficult for them to identify their "matching set" with high and stable performance, whereas our BTM model has the ability to support chunkers and parsers for improving chunking performance. According to the fourth experiment results, when applying a BTM (0.5, 2, 45,000) model on the matching set and a 4-gram model on the non-matching set, the combined system can improve the F-measure of 4-gram model 2.5% for perfect input and 1.0% for actual input. Among the chunking and parsing models, Cascaded Markov Models should be the first one to construct the parse tree layer by layer with each layer's Markov Model. As per (Brants, 1999), each layer's chunk bracketing of Cascaded Markov Models is dependent because the output of a lower layer is passed as input to the next higher layer. On the contrast, our BTM model can independently generate the chunks for top layer without the results of lower layer chunk bracketing; and

(6) Since the F-measures of the BTM model for the matching sets of perfect and actual input both are greater than 95%, we believe our BTM model can be used not only to improve the F-measure of existing shallow parsing or chunking systems, but also to help select valuable sentences from the non-matching set for effectively extending the CR of our BTM model.

In the future, we shall study how to combine our BTM model with more conventional statistical approaches, such as Bayesian Networks, Maximum Entropy and Cascaded Markov Models, etc. Meanwhile, we will also apply our BTM model to the Penn English Treebank as a comparative study.

Acknowledgement

Some work of this paper was done while the author was a visitor at the Institute of Information Science (Academia Sinica in Taiwan). We thank Prof. Wei-Lian Hsu and Cheng-Wei Shih for their kind help. We also thank the Chinese Knowledge Information Processing group in Taiwan for providing us the CKIP Treebank.

References

Abney, S. Parsing by chunks. In Principle-Based Parsing. Kluwer Academic Publishers, Dordrecht: pp.257–278. 1991.

Basili, R. and Zanzotto, F. M. Parsing engineering and empirical robustness. Natural Language Engineering, 8(2–3):pp.97–120. 2002.

Brants, T. Cascaded Markov Models. Proceedings of EACL '99. pp.118 – 125. 1999.

Chen, K.-J. and Huang C.-R. Information-based Case Grammar: A Unification-based Formalism for Parsing Chinese. In Huang et al. (Eds.): pp.23-46. 1996.

Chen, K.-J., Huang C.-R., Chang, L.-P. and Hsu, H.-L. Sinica Corpus: Design Methodology for Balanced Corpra." Proceedings of the 11th Pacific Asia Conference on Language, Information, and Computation (PACLIC II), SeoulKorea, pp.167-176. 1996.

Chen, K.-J., Luo, C.-C., Gao, Z.-M., Chang, M.-C., Chen, F.-Y., and Chen, C.-J. The CKIP Chinese Treebank. In Journ ees ATALA sur les Corpus annot es pour la syntaxe, Talana, Paris VII: pp.85-96. 1999.

Chen, K.-J. et al. Building and Using Parsed Corpora. (Anne Abeillé ed. s) KLUWER, Dordrecht. 2003.

Chen, K.-J and Hsieh Y.-M. Chinese Treebanks and Grammar Extraction. Proceedings of IJCNLP-2004: pp.560-565. 2004.

Chiou, F.-D., Chiang, D. and Palmer, M. Facilitating Treebank Annotation with a Statistical Parser. Proceedings of the Human Language Technology Conference (HLT 2001), San Diego, California, 2001.

CKIP portal. http://rocling.iis.sinica.edu.tw/CKIP/treebank.htm

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.

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.

Gao, J.-F., Goodman, J., Li, M.-J., and Lee, K.-F. Toward a Unified Approach to Statistical Language Modeling for Chinese, ACM Transactions on Asian language Information processing. 1(1): pp.23-33. 2002.

Huang, Chu-Ren, Chen, K.-J., Chen, F.-Y., Gao, Z.-M. and Chen, K.-Y. Sinica Treebank: Design Criteria, Annotation Guidelines, and On-line Interface. Proceedings of 2nd Chinese Language Processing Workshop (Held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, ACL-2000). October 7, 2000, Hong Kong: pp.29-37. 2000.

Johnny Bigert, Jonas Sjöbergh, Ola Knutsson and Magnus Sahlgren. Unsupervised Evaluation of Parser Robustness. In Proc. of CICLing 2005. Mexico City, Mexico. 2005.

Knutsson, O., Bigert, J. and Kann, V. A Robust Shallow Parser for Swedish. In: Proc. of Nodalida'03. Reykavik, Iceland. 2003.

Le, Zhang, Lu X.ue-qiang, Shen Yan-na and Yao Tian-shun. A Statistical Approach to Extract Chinese Chunk candidates from Large Corpora. In: Proc. of ICCPOL-2003. ShengYang: pp.109-117. 2003.

Li, Hongqiao, Huang, C.-N., Gao, Jianfeng and Fan, Xiaozhong. Chinese chunking with another type of spec. In SIGHAN-2004. Barcelona: pp. 41-48. 2004.

Liu Y., Q. Tan, and X. Shen. Segmentation Standard for Modern Chinese Information Processing and Automatic Segmentation Methodology. 1993.

Jorn Veenstra. Memory-Based Text Chunking. In: Nikos Fakotakis (ed), Machine learning in human language technology, workshop at ACAI 99. Chania, Greece. 1999.

Lance A. Ramshaw and Mitchell P. Marcus. Text Chunking Using Transformation-Based Learning. In: Proc. of the Third ACL Workshop on Very Large Corpora. Cambridge MA, USA: pp.82-94. 1995.

Manning, C. D. and Schuetze, H. Fundations of Statistical Natural Language Processing, MIT Press: pp.191-220., 1999.

Menzel, W. Robust processing of natural language. In Proc. 19th Annual German Conference on Artificial Intelligence, Berlin. Springer: pp.19–34. 1995.

Munoz, M., V. Punyakanok, D. Roth, and D. Zimak. A learning approach to shallow parsing. Technical Report UIUCDCS-R-99-2087, UIUC Computer Science Department. 1999.

Oliver Streiter. Memory-based Parsing: Enhancing Recursive Top-down Fuzzy Match with Bottom-up Chunking. ICCPOL 2001, Seoul. 2001.

PCTB portal. http://www.cis.upenn.edu/~chinese/ctb.html

Ruifeng Xu, Qin Lu, Yin Li and Wanyin Li. The Construction of a Chinese Shallow Treebank. In: Proc. of 3rd ACL SIGHAN Workshop. Barcelona: pp.94-101. 2004.

Sang, Erik F. Tjong Kim and Buchholz, Sabine. Introduction to the CoNLL-2000 Shared Task: Chunking. In: Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal: pp.127-132. 2000.

Stolcke, A. SRILM - An Extensible Lan-guage Modeling Toolkit, in Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado. 2003.

Xia, Fei, Palmer, M., Xue, N., Okurowski, M.E., Kovarik, J., Chiou, F.-D., Huang, S., Kroch, T. and Marcus, M. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In: Proc. of LREC-2000. Greece. 2000.

Xue, N., Chiou, F. and M. Palmer. Building a Large-Scale Annotated Chinese Corpus, In: Proc. of COLING-2002. Taipei, Taiwan. 2002.

Xue, N. and Palmer, M. Annotating Propositions in the Penn Chinese Treebank, In Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing, in conjunction with ACL'03. Sapporo, Japan. 2003.

Xue., N. and M. Palmer. Automatic Semantic Role Labeling for Chinese Verbs, in Proceedings of the 19th International Joint Conference on Artificial Intelligence. Edinburgh, Scotland. 2005

Xue, N., Xia, F., Chiou F.-D. and M. Palmer. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. Natural Language Engineering, 11(2)-207. 2005.

You Jia-Ming and Chen, K.-J. Automatic Semantic Role Assignment for a Tree Structure. Proceedings of SIGHAN workshop. pp.109-115. 2004.

Zhou, G.-D. and Su, J. A Chinese Efficient Analyser Integrating Word Segmentation, Part-Of-Speech Tagging, Partial Parsing and Full Parsing. ACL Second SIGHAN Workshop on Chinese Language Processing. pp.78-83. 2003.