Chinese Chunking with another Type of Spec

Hongqiao Li Beijing Institute of Technology Beijing 100081 China lhqtxm@bit.edu.cn **Chang-Ning Huang** Microsoft Research Asia Beijing 100080 China cnhuang@msrchina. research.microsoft.com

Abstract

Spec is a critical issue for automatic chunking. This paper proposes a solution of Chinese chunking with another type of spec, which is not derived from a complete syntactic tree but only based on the un-bracketed, POS tagged corpus. With this spec, a chunked data is built and HMM is used to build the chunker. TBLbased error correction is used to further improve chunking performance. The average chunk length is about 1.38 tokens, F measure of chunking achieves 91.13%, labeling accuracy alone achieves 99.80% and the ratio of crossing brackets is 2.87%. We also find that the hardest point of Chinese chunking is to identify the chunking boundary inside noun-noun sequences¹.

1 Introduction

Abney (1991) has proposed chunking as a useful and relative tractable median stage that is to divide sentences into non-overlapping segments only based on superficial analysis and local information. (Ramshaw and Marcus, 1995) represent chunking as tagging problem and the CoNLL2000 shared task (Kim Sang and Buchholz, 2000) is now the standard evaluation task for chunking English. Their work has inspired many others to study chunking for other human languages.

Besides the chunking algorithm, spec (the detailed definitions of all chunk types) is another critical issue for automatic chunking development. The well-defined spec can induce the chunker to perform well. Currently chunking specs are defined as some rules or one program to extract phrases from Treebank such as (Li, 2003) and (Li, 2004) in order to save the cost of manual annotation. We name it as Treebank-derived spec. However, we find that it is more valuable to compile another type of chunking spec according to the observation from un-bracketed corpus instead of Treebank.

Jianfeng Gao Microsoft Research Asia Beijing 100080 China jfgao@microsoft.com

Xiaozhong Fan Beijing Institute of Technology Beijing 100081 China fxz@bit.edu.cn

Based on the problems of chunking Chinese that are found with our observation, we explain the reason why another type of spec is needed and then propose our spec in which the shortening and extending strategies are used to resolve these problems. We also compare our spec with a Treebank-derived spec which is derived from Chinese Treebank (CTB) (Xue and Xia, 2000). An annotated chunking corpus is built with the spec and then a chunker is also constructed accordingly. For annotation, we adopt a two-stage processing, in which text is first chunked manually and then the potential inconsistent annotations are checked semi-automatically with a tool. For the chunker, we use HMM model and TBL (Transform-based Learning) (Brill, 1995) based error correction to further improve chunking performance. With our spec the overall average length of chunks arrives 1.38 tokens, in open test, the chunking F measure achieves 91.13% and 95.45% if under-combining errors are not counted. We also find the hardest point of Chinese chunking is to identify the chunking boundary inside a noun-noun sequence.

In the remainder of this paper section 2 describes some problems in chunking Chinese text, section 3 discusses the reason why another type of spec is needed and proposes our chunking spec, section 4 discusses the annotation of our chunking corpus, section 5 describes chunking model, section 6 gives experiment results, section 7, 8 recall some related work and give our conclusions respectively.

2 Problems of Chunking Chinese Text

The purpose of Chinese chunking is to divide sentence into syntactically correlated parts of words after word segmentation and part-of-speech (POS) tagging. For example:

[NP 珠海/ns 'Zhuhai'] 的/u 'of' [NP 立体/a 'solid' 交通/n 'traffic' 框架/n 'frame'] [VP 已/d 'already' 初具规模/v 'achieve considerable scale '了/u] 'Zhuhai has achieved considerable scale in solid traffic frame.'

According to Abney's definition, most chunks are modifier-head structures and non-overlapping. However, some syntactic structures in Chinese are very hard to be chunked correctly due to

¹ This work was done while Hongqiao Li was visiting Microsoft Research Asia.

characteristics of Chinese language, for example, less using of function words and less inflection formats. Table 1 shows the most common structural ambiguities occurred during Chinese chunking. Their occurrences and distributions of each possible structure are also reported. As can be seen in Table 1, only 77% neighboring nouns can be grouped inside one chunk; if the left word is '的 /of' or a verb, this figure will ascend to 80% and 94% respectively; but if the left word is an adjective or a numeral, it will descend to 70% and 59% respectively; for 'n_c_n', only 52% are word level coordination. In contrast with English chunking, several hard problems are described in detail as following.

(1) Noun-noun compounds

Compounds formed by more than two neighboring nouns are very common in Chinese and not always all the left nouns modify the head of the compound. Some compounds consist of several shorter sub-compounds. For example:

(青年/younger 志愿者/volunteer 科技/science and technology 服务队/service team) 'young volunteer service team of science and technology'

'青年 志愿者' and '科技 服务队' are two subcompounds and the former modifies the latter.

But sometimes it is impossible to distinguish the inner structures, for example:

世界/world 和平/peace 事业/career

It is impossible to distinguish whether it is {{世界和平}事业} or {世界 {和平 事业}}.

English chunking also shows such problem, and the common solution for English is not to identify their inner structure and treat them as a flat noun phrase. Following is an example in CoNLL2000 shared task:

[NP employee assistance program directors]

(2) Coordination

Coordination in all cases can be divided into two types: with conjunctions and without conjunctions. The former can be further divided into two subcategories: word-level and phrase-level coordinations. For example:

{政策性/policy 银行/bank 与/and 商业/commercial 银行/bank}的/of {联系/relationship 与/and 合作/cooperation} 'the relationship and cooperation between policy banks and commercial banks'.

The former coordination is phrase-level and the latter is word-level. Unfortunately, sometimes it is difficult or even impossible to distinguish whether it is word-level or phrase-level at all, for example:

最低/least 工资/salary 和/and 生活费/living maintenance 'the least salary and living maintenance'

It is impossible to distinguish '最低' is a shared modifier or not. English chunking also has such kind of problems. The solution of CoNLL2000 is to leave the conjunctions outside chunks for phrase-level coordinations and to group the conjunction inside a chunk when it is word-level or impossibly distinguished phrase-level. For example:

[NP enough food and water]

In Chinese, some coordinate construction has no conjunction or punctuation inside, and also could not be distinguished from a modifier-head construction with syntactic knowledge only. For example:

整顿/order (警车/police wagon 警灯/caution light 警报器/alarm whistle) 'Order the police

Pattern ¹	$No.^2$	Distributions	Examples
		77% (modifier head)	(社会/society 现象/phenomenon) 'social phenomena'
n_n	951	7% (coordination) 16% (others)	(语言/language 文字/wordage) 'language and wordage'
			(首都/capital 文艺/art <i>舞台/stage</i>) ' the stage of capital art'
u n n	154	6% (v_n modify the last noun)	进/enter 厂/factory 工人/worker
v_n_n 154	134	⁴ 94 % (others)	逃避/avoid 法律/law 责任/duty 'avoid legal duties'
的_n_n 9	00	80% (n_n is modifier_head)	执勤/watch 的/of 交通/traffic 警察/cop 'a orderly traffic cop'
	90	20% (others)	<i>瘫痪/paralytic</i> 的/of肢体/body功能/function
a_n_n 27		70% (a modify the first n)	高/high 科技/technology 企业/company 'high-tech company'
		30% (others)	老/old 新闻/news 工作者/worker 'old news worker'
17		41% (m modify the first n)	两/two 国/nation 人民/people 'our two peoples'
m_n_n	17	59% (others)	一些/some 农村/country 地区/area 'some rural areas'
n_c_n	00	52% (word level coordination)	经济/economy 和/and 社会/society 'economy and society'
	00	48%(others)	质量/quality 和/and 技术/technology 要求/requirement

¹ n, v, a, d, m, q, p, f, c are the POS tags of noun, verb, adjective, adverb, number, measure, preposition, localizer, conjunction respectively, '_' means neighboring, ' \square /of' is a common auxiliary word in Chinese.

²This statistical work is done on our test corpus whose setting is shown in Table 3.

Table 1: The observation of several common structural ambiguities during Chinese chunking

wagons, caution lights and alarm whistles'

Such problem does not exist in English because almost all coordinations have certain conjunctions or punctuations between words or phrases of the same syntactic categories in formal English.

(3) Structural ambiguities

In Chinese, some structural ambiguities in phrase level are impossible or unnecessary to be distinguished during chunking. There is an example of 'a_n_n':

现代/a 'modern' 企业/n 'industry' 制度/n 'system'

{现代 {企业 制度}} or {{现代 企业} 制度} are identically acceptable. English also has such problem. The solution of CoNLL2000 is not to distinguish inner structure and group the given sequence as a single chunk. For example, the inner structure of '[NP heavy truck production]' is '{{heavy truck} production}', whereas one reading of '[NP heavy quake damage]' is '{heavy {quake damage}}'.

Besides, 'a_n_n', 'm_n_n' and 'm_q_n_n' also have the similar problem.

3 Chinese Chunking Spec

As a kind of shallow parsing, the principles of chunking are to make chunking much more efficient and precise than full parsing. Obviously, one can shorten the length of chunks to leave ambiguities outside of chunks. For example, if we let noun-noun sequences always chunk into single word, those ambiguities listed in Table 1 would not be encountered and the performance would be greatly improved. In fact, there is an implicit requirement in chunking, no matter which language it is, the average length of chunks is as longer as possible without violating the general principle of chunking. So a trade-off between the average chunk length and the chunking performance exists.

3.1 Why another type of spec is needed

A convenient spec is to extract the lowest nonterminal nodes from a Treebank (e.g. CTB) as Chinese chunked data. But there are some problems. The trees are designed for full parsing instead of shallow parsing, thus some of these problems listed in section 2 could not be resolved well in chunking. Maybe we can compile some rules to prune the tree or break some non-terminal nodes in order to properly resolve these problems just like CoNLL2000. However, just as (Kim Sang and Buchholz, 2000) noted: "some trees are very complex and some annotations are inconsistent". So these rules are complex, the extracted data are inconsistent and manual check is also needed. In addition, the resource of Chinese Treebank is limited and the extracted data is not enough for chunking.

So we compile another type of chunking spec according to the observation from un-bracket corpus instead of Treebank. The only shortcoming is the cost of annotation, but there are some advantages for us to explore.

1) It coincides with auto chunking procedure, and we can select proper solutions to these problems without constraints of the exist Treebank. The purpose of drafting another type of chunking spec is to keep chunking consistency as high as possible without hurting the performance of autochunking in whole.

2) Through spec drafting and text annotating most frequent and significant syntactic ambiguities could be studied, and those observations are in turn described in the spec carefully.

3) With a proper spec and certain mechanical approaches, a large-scale chunked data could be produced without supporting from the Treebank.

3.2 Our spec

Our spec and chunking annotation are based on PK corpus² (Yu et al. 1996). The PK corpus is unbracketed, but in which all words are segmented and only one POS tag is assigned to each word. We define 11 chunk types that are similar with CoNLL2000. They are NP (noun chunk), VP (verb chunk), ADJP (adjective chunk), ADVP (adverb chunk), PP (prepositional chunk), CONJP (conjunction), MP (numerical chunk), TP (temporal chunk), SP (spatial chunk), INTJP (interjection) and INDP (independent chunk).

During spec drafting we try to find a proper chunk spec to solve these problems by two ways: either merging neighboring chunks into one chunk or shortening them. Besides those structural ambiguities, we also extend boundary of the chunks with minor structural ambiguities in order to make the chunks close to the constituents.

3.2.1 Shortening

The auxiliary '的/of' is one of the most frequent words in Chinese and used to connect a premodifier with its nominal head. However the left boundary of such a 的 -construction is quite complicated: almost all kinds of preceding clauses, phrases and words can be combined with it to form such a pre-modifier, and even one 的-construction can embed into another. So we definitely leave it outside any chunk. Similarly, conjunctions, '和 /and', '或/or' and '与/and' et al., are also left outside any chunk no matter they are word-level or

² Can be downloaded from <u>www.icl.pku.edu.cn</u>

phrase-level coordinations. For instances, the examples in Section 2 are chunked as '[NP 政策性 银行] 与 [NP 商业 银行] 的 [NP 联系] 与 [NP 合 作]' and '[ADJP 最低] [NP 工资] 和 [NP 生活 费]'

3.2.2 Extending

(1) NP

Similar with the shared task of CoNLL2000, we define noun compound that is formed by a noun-sequence: 'a_n_n', 'm_n_n' or 'm_q_n_n', as one chunk, even if there are sub-compounds, sub-phrase or coordination relations inside it. For instances, '[NP 青年 志愿者 科技 服务队]', '[NP 世界 和平 事业]', '[VP 整顿] [NP 警车 警 灯 警报器]', '[NP 现代 企业 制度] and '[NP 立 体 交通 框架]' are grouped into single chunks respectively.

However, it does not mean that we blindly bind all neighboring nouns into a flat NP. If those neighboring nouns are not in one constituent or cross the phrase boundary, they will be chunked separately, such as following two examples in Table 1: '[VP 进] [NP \Box] [NP \Box Å]' and '[ADJP 瘫痪] 的/u [NP 肢体] [NP 功能]'. So our solution does not break the grammatical phrase structure in a given sentence.

With this chunking strategy, we not only properly resolved these problems, but also get longer chunks. Longer chunks can make successive parsing easier based on chunking. For example, if we chunked the sentence as:

[NP 珠海] 的 [NP 立体 交通] [NP 框架] [VP 已 初具规模 了]。/w

There would be three possible syntactic trees which are difficult to be distinguished:

1a) {{ [NP 珠海] 的 { [NP 立体 交通] [NP 框 架]}} [VP 己 初具规模]}

1b) {{{ [NP 珠海] 的 [NP 立体 交通]} [NP 框 架]} [VP 已 初具规模]}

1c) {{ [NP 珠海] 的 [NP 立体 交通]} { [NP 框 架] [VP 已 初具规模]}}

Whereas with above chunking strategy of our spec, there is only one syntactic tree remained:

{{[NP 珠海] 的 [NP 立体 交通 框架]} [VP 已 初具规模 了]} 。/w

Another reason of the chunking strategy is that for some NLP applications such as IR, IE or QA, it is unnecessary to analyze these ambiguities at the early stage of text analysis.

(2) PP

Most PP consists of only the preposition itself because the right boundary of a preposition phrase is hard to identify or far from the preposition. But certain prepositional phrases in Chinese are formed with a frame-like construction, such as [PP \pm/p 'at' ... 中/f 'middle'], [PP \pm/p ... \pm/f 'top'], etc. Statistics shows that more than 90% of those frame-like PPs are un-ambiguous, and others commonly have certain formal features such as an auxiliary 的 or a conjunction immediately following the localizer. Table 2 shows the statistic result. Thus with those observations, those frame-like constructions could be chunked as PP. The length of such kind of PP frames is restricted to be at most two words inside in order to keep the distribution of chunk length more even and the chunking annotation more consistent.

Pattern ¹	No.of occurrence	Ratio as a chunk
p_*_f	45	93.33%
P_*_*_f	36	97.22%
*_f	40	92.50%
*_*_f	9	77.78%

¹ This statistical work is also done on our test corpus and '*' means a wildcard for a POS tag.

Table 2: The ration of grouping these patterns

as a chunk without any ambiguity

(3) SP

Most spatial chunks consist of only the localizer(with POS tag '/s' or '/f'). But if the spatial phrase is in the beginning of a sentence, or there is a punctuation (except "、") in front of it, then the localizer and its preceding words could be chunked as a SP. And the number of words in front of the localizer is also restricted to at most two for the same reason.

(4) VP

Commonly, a verb chunk VP is a pre-modifier verb construction, or a head-verb with its following verb-particles which form a morphologically derived word sequence. The pre-modifier is formed by adverbial phrases and/or auxiliary verbs. In order to keep the annotation consistent those verb particles and auxiliary verbs could be found in a closed list respectively only. Post-modifiers of a verb such as object and complement should be excluded in a verb chunk.

We find that although a head verb groups more than one preceding adverbial phrases, auxiliary verbs and following verb-particles into one VP, its chunking performance is still high. For example:

[CONJP 如果/c 'if'] [VP 迟迟/d 'lately'

不/d 'not' 能/v 'can' 建立/v 'build' 起/v

'up'] [NP 外交/n 'diplomat 关系/n 'relation']

'If we could not build up the foreign relations soon'

3.3 Spec Comparison

We compare our spec with the Treebank-derived spec, named as S1, which is to extract the lowest non-terminal nodes from CTB as chunks from the aspect of the solutions of these problems in section 2. Noun-noun compound and the coordination which has no conjunction are chunked identically in both specs. But for others, there are different. In S1, the conjunctions of phrase-level coordination are outside of chunks and the ones of word-level are inside a chunk, all adjective or numerical modifiers are separate from noun head. According to S1, the example in 3.2.1 should be chunked as following.

[ADJP 政策性] [NP 银行] 与 [NP 商业] [NP 银行] 的 [NP 联系 与 合作]

But these phrases that are impossible to distinguish inner structures during the early stage of text analysis are hard to be chunked and would cause some inconsistency. '[ADJP 最低] [NP 工资]和 [NP 生活费]' or '[ADJP 最低] [NP 工资和生活费]', '[ADJP 现代] [NP 企业] [NP 制度]' or '[ADJP 现代] [NP 企业制度]', are hard to make decisions with S1.

In addition, with our spec outside words are only punctuations, structural auxiliary ' \oint /of', or conjunctions, whereas with S1, outside words are defined as all left words after lowest non-terminal extraction.

4 Chunking Annotation

Four graduate students of linguistics were assigned to annotate manually the PK corpus with the proposed chunking spec. Many discussions between authors and those annotators were conducted in order to define a better chunking spec for Chinese. Through the spec drafting and text annotating most significant syntactic ambiguities in Chinese, such as those structural ambiguities discussed in section 2 and 3, have been studied, and those observations are carefully described in the spec in turn.

Consistency control is another important issue during annotation. Besides the common methods: manual checking, double annotation, post annotation checking, we explored a new consistency measure to help us find the potential inconsistent annotations, which is hinted by (Kenneth and Ryszard. 2000), who defined consistency gain as a measure of a rule in learning from noisy data.

The consistency of an annotated corpus in whole could be divided down into consistency of each chunk. If the same chunks appear in the same context, they should be identically annotated. So we define the consistency of one special chunk as the ratio of identical annotation in the same context. cons(P, context(P))

$$= \frac{\text{No. of same annotation in context}(P)}{\text{No. of } (P, \text{context}(P)) \text{ in corpus}}$$
(1)
$$\cosh(S) = \frac{1}{N} \sum_{n=1}^{N} \cosh(P, \text{context}(P))$$

 $\operatorname{cons}(S) = \frac{1}{N} \sum_{i=1}^{N} \operatorname{cons} (P_i, \operatorname{context} (P_i))$ (2) Where P represents a pattern of the chunk (POS)

Where P represents a pattern of the chunk (POS or/and lexical sequence), context(P) represents the needed context to annotate this chunk, N represents the number of chunks in the whole corpus S.

In order to improve the efficiency we also develop a semi-automatic tool that not only check mechanical errors but also detect those potential inconsistent annotations. For example, one inputs a POS pattern: 'a_n_n', and an expected annotation result: 'B-NP_I-NP_E-NP³', the tool will list all the consistent and inconsistent sentences in the annotated text respectively. Based on the output one can revise those inconsistent results one by one, and finally the consistency of the chunked text will be improved step by step.

5 Chunking Model

After annotating the corpus, we could use various learning algorithms to build the chunking model. In this paper, HMM is selected because not only its training speed is fast, but also it has comparable performance (Xun and Huang, 2000). Automatic chunking with HMM should conduct the following two steps. 1) Identify boundaries of each chunk. It is to assign each word a chunk mark, named M, which contains 5 classes: B, I, E, S (a single word chunk) and O (outside all chunks). 2) Tag the chunk type, named X, which contains 11 types defined in Section 3.

So each word will be tagged with two tags: M and X (the words excluding from any chunk only have M). So the result after chunking is a sequence of triples (t, m, x), where t, m, x represent POS tag, chunk mark and chunk type respectively. All the triples of a chunk are combined as an item n_i , which also could be named as a chunk rule. Let W as the word segmentation result of a given sentence, T as POS tagging result and C ($C = n_1 n_2 ... n_j$) as the chunking result. The statistical chunking model could be described as following: $C^* = \arg \max P(C | W, T)$

$$= \arg \max_{C} P(C \mid W, T)$$

$$= \arg \max_{C} P(W \mid C, T) P(C, T) / P(W, T)$$

$$= \arg \max_{C} P(W \mid C, T) P(C, T)$$

$$(3)$$

Independent assumption is used to approximate P(W/C,T), that is:

³ B, E, I represent the left/right boundary of a chunk and inside a chunk respectively, B-NP means this word is the beginning of NP.

$$P(W \mid C, T) \approx \prod_{i=1}^{m} P(w_i \mid t_i, m_i, x_i)$$
(4)

If the triple is unseen, formula 5 is used.

$$P(w_{i} | t_{i}, m_{i}, x_{i}) = \frac{coun(t_{i}, m_{i}, x_{i})}{\max_{j,k}(coun(t_{i}, m_{j}, x_{k}))^{2}}$$
(5)

For P(C, T), tri-grams among chunks and outside words are used to approximate, that is:

$$P(C,T) \approx P(n_1)P(n_2 \mid n_1) \prod_{i=3}^{k} P(n_i \mid n_{i-2}n_{i-1})$$
(6)

Smoothing follows the method of (Gao et al., 2002).

In order to improve the performance we use N-fold error correction (Wu, 2004) technique to reduce the error rate and TBL is used to learn the error correction rules based on the output of HMM.

6 Data and Evaluation

The performance of chunking is commonly measured with three figures: precision (P), recall (R) and F measure that are defined in CoNLL2000. Besides these, we also use two other measurements to evaluate the performance of bracketing and labeling respectively: RCB(ratio of crossing brackets), that is the percentage of the found brackets which cross the correct brackets; LA(labeling accuracy), that is the percentage of the found chunks which have the correct labels.

RCB =

No. of the chunks crossed chunk boundaries No. of chunks in test data

LA =

No. of correct chunks

No. of the chunks with correct boundaries

The average length (ALen) of chunks for each type is the average number of tokens in each chunk of given type. The overall average length is the average number of tokens in each chunk. To be more disinterested, outside tokens (including outside punctuations) are also concerned and each of them is counted as one chunk.

6.1 Chunking performance with our spec

Training and test was done on the PK corpus. Table 3 shows the detail information. We use the uni-gram of chunk POS rules as the baseline.

Data	No. of	No. of	No. of	ALen
Data	tokens	chunks	outside	(include O)
Train	444,777	229,989	92,839	1.377
Test	28,382	13,879	5,493	1.363

Table 3: The information of data set

Table 4 shows the chunking performance of close test and open test when HMM and ten folds

TBL based error correction (EC) are done respectively.

	Close Test (%)			Open Test (%)		
	F	RCB	LA	F	RCB	LA
Baseline	81.95	6.55	99.46	81.44	6.58	99.47
HMM	94.79	2.62	99.78	88.39	3.18	99.65
HMM+EC	95.11	2.38	99.91	91.13	2.87	99.80

Table 4: The overall performance of chunking

As can be seen, the performance of open test doesn't drop much. For open test, HMM achieves 6.9% F improvement, 3.4% RCB reduction on baseline; error correction gets another 2.7% F improvement, 0.3% RCB reduction. Labeling accuracy is so high even with the baseline, which indicates that the hard point of chunking is to identify the boundaries of each chunk.

Table 5 shows the performance of each type of chunks respectively. NP and VP amount to approximately 76% of all chunks, so their chunking performance dominates the overall performance. Although we extend VP and PP, their performances are much better than overall. The performance of INDP can arrive 99% although it is much longer than other types. Because its surface evidences are clear and complete owing to its definition: the meta-data of a document, all the descriptions inside a pair of parenthesis, and also certain fixed phrases which do not act as a syntactic constituent in a sentence. From the relative lower performance of NP, but the most part of all chunks, we can conclude that the hardest issue of Chinese chunking is to identify boundaries of NPs.

	Percent	ALen	Р	R	F
	age(%)	(tokens)	(%)	(%)	(%)
NP	45.94	1.649	88.82	86.25	87.52
VP	29.82	1.416	96.60	96.49	96.55
PP	6.59	1.221	93.67	93.58	93.63
MP	3.69	1.818	89.51	86.33	87.89
ADJP	3.77	1.308	86.11	89.43	87.74
SP	2.71	1.167	84.70	84.03	84.36
TP	2.59	1.251	93.23	94.30	93.76
CONJP	2.22	1.000	97.20	98.73	97.96
INDP	1.41	4.297	99.06	99.06	99.06
ADVP	1.06	1.117	85.48	85.03	85.25
INTJP	0.23	1.016	68.75	95.65	80.00
ALL	100	1.507	91.70	90.55	91.13

Table 5: The result of each type with our spec

All the chunking errors could be classified into four types: wrong labeling, under-combining, overcombining and overlapping. Table 6 lists the number and percentage of each type of errors. Under-combining errors count about a half number of overall chunking errors, however it is not a problem in certain applications because they does not cross the brackets, thus there are still opportunities to combine them later with additional knowledge. If we evaluate the chunking result without counting those under-combining errors, the F score of the proposed chunker achieves 95.45%.

Error type	No.of the Errors	Percentage	
Wrong labeling	22	2.56%	
Under-combine	418	48.71%	
Over-combining	339	39.51%	
Overlapping	59	6.88%	

Table 6: The distribution of chunking errors

With comparison we also use some other learning methods, MBL(Bosch and Buchholz, 2002), SVM(Kudoh and Matsumoto, 2001) and TBL to build the chunker. The features for MBL and SVM are the POS of current, left two and right two words, lexical of current, left one and right one word. TiMBL⁴ and SVM-light⁵ are used as the tools. For SVM, we convert the chunk marks BIOES to BI and the binary class SVM is used to classifier the chunk boundary, then some rules are used to identify its label. For TBL, the rule templates are all the possible combinations of the features and the initial state is that each word is a chunk. Table 7 shows the result. As seen, without error correction all these models do not perform well and our HMM gets the best performance.

	MBL	SVM	TBL	HMM
F(%)	85.31	86.25	86.92	88.39

Table 7: Comparison with different algorithms

6.2 Further applications

The length of chunks with our spec (AoL is 1.38) is longer than other Treebank-derived specs (AoL of S1 is 1.239) and closer to the constituents of sentence. Thus there are several applications benefit from the fact, such as:

1) The longest/full noun phrase identification. According to our statistics, due to including nounnoun compounds, 'a_n_n' and 'm_n_n' inside NPs, 65% noun chunks are already the longest/full noun phrases and other 22% could become the longest /full noun phrases by only one next combining step.

2) The predicate-verb identification.

By extending the average length of VPs, the main verb (or predicate-verb, also called tensed verb in English) of a given sentence could be identified based on certain surface evidences with a relatively high accuracy. With certain definition our statistics based on our test set show that 84.88% of those main verbs are located in the first longest VPs among all VPs in a sentence.

7 Related Work

For chunking spec, the CoNLL2000 shared task defines a program chunklink to extract chunks from English Treebank. (Li, 2003) defines the similar Treebank-derived spec for Chinese and she reports manual check is also needed to make data consistent. Part of the Sparkle project has concentrates on a spec based on un-bracketed English, corpus of Italian, French and German(Carroll et al., 1997). (Zhou, 2002) defines base phrase which is similar as chunk for Chinese, but his annotation and experiment are on his own corpus.

For chunking algorithm, many machine learning (ML) methods have been applied and got promising results after chunking is represented as tagging problem, such as: SVM (Kudoh and Matsumoto, 2001), Memory-based (Bosch and Buchholz, 2002), SNoW (Li and Roth), et al... Some rule-base chunking (Kinyon, 2003) and combining rules with learning (Park and Zhang, 2003) are also reported.

For annotation, (Brants, 2000) reports the interannotator agreement of part-of-speech annotations is 98.57%, the one of structural annotations is 92.43% and some consistency measures. (Xue et al., 2002) also address some issues related to building a large-scale Chinese corpus.

8 Conclusion

We propose a solution of Chinese chunking with another type of spec that is based on un-bracketed corpus rather than derived from a Treebank. Through spec drafting and annotating, most significant syntactic ambiguous patterns have been studied, and those observations in turn have been described in the spec carefully. The proposed method of defining a chunking spec helps us find a proper solution for the hard problems of chunking Chinese. The experiments show that with our spec, the overall Chinese chunking F-measure achieves 91.13% and 95.45% if under-combining errors are not counted.

9 Acknowledgements

We would like to thank the members of the Natural Language Computing Group at Microsoft Research Asia. Especial acknowledgement is to the anonymous reviewers for their insightful comments and suggestions. Based on that we have revised the paper accordingly.

⁴ <u>http://ilk.kub.nl/software.html</u>

⁵ <u>http://svmlight.joachims.org/</u>

References

- S. Abney. 1991. Parsing by chunks. In *Principle-Based Parsing*. Kluwer Academic Publishers, Dordrecht: 257–278.
- L.A. Ramshaw and M.P. Marcus. 1995. Text chunkingusing transformation-based learning. In *Proceedings of the 3rd ACL/SIGDAT Workshop*, Cambridge, Massachusetts, USA: 82–94.
- E. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal: 127–132.
- Sujian Li, Qun Liu and Zhifeng Yang. 2003. Chunking based on maximum entropy. *Chinese Journal of Computer*, 25(12): 1734–1738.
- Heng Li, Jingbo Zhu and Tianshun Yao. 2004. SVM based Chinese text chunking. *Journal of Chinese Information Processing*, 18(2): 1–7.
- Nianwen Xue and Fei Xia. 2000. The Bracketing Guidelines for the Penn Chinese Treebank(3.0). *Technical report*, *University of Pennsylvania*, URL http:// www.cis.upenn.edu/~chinese/.
- Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*,21 (4):543–565.
- Shiwen Yu, Huiming Duan, Xuefeng Zhu, et al. 2002. The basic processing of contemporary Chinese corpus at Peking University. *Journal of Chinese Information Processing*, 16(6): 58–65.
- K.A. Kaufman and R.S. Michalski. 1999. Learning from inconsistent and noisy data: the AQ18 approach, *Proceedings of the Eleventh International Symposium on Methodologies for Intelligent Systems*, Warsaw: 411–419.
- Endong Xun and Changning Huang. 2000. A unified statistical model for the identification of English baseNP, In *Proceedings of the 38th ACL*: 109–117.
- Jianfeng Gao, Joshua Goodman, Mingjing Li, Kai-Fu Lee. Toward a unified approach to statistical language modeling for Chinese. *ACM Transactions on Asian Language Information Processing*, Vol. 1, No. 1, 2002: 3-33.
- Dekai WU, Grace NGAI, Marine CARPUAT. Nfold Templated Piped Correction. *Proceedings of the First International Joint Conference on Natural Language Processing*, SANYA: 632– 637.
- Antal van den Bosch and S. Buchholz. 2002. Shallow parsing on the basis of words only: a case study, In *Proceedings of the 40th ACL*: 433–440.

- Taku Kudo and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification, In *Proceedings of the 4th CoNLL*: 142–144.
- J. Carroll, T. Briscoe, G. Carroll et al. 1997. *Phrasal parsing software*. Sparkle Work Package 3, Deliverable D3.2.
- Yuqi Zhang and Qiang Zhou. 2002. Automatic identification of Chinese base phrases. *Journal of Chinese Information Processing*, 16(6):1–8.
- X. Li and D. Roth. 2001. Exploring evidence for shallow parsing. In *Proceedings of the* 5^{th} *CoNLL*.
- Alexandra Kinyon. 2003. A language-independent shallow-parser compiler. In *Proceedings of 10th EACL Conference*, Toulouse, France: 322-329.
- S.-B. Park, B.-T. Zhang. 2003. Text chunking by combining hand-crafted rules and memory-based, In *Proceedings of the 41th ACL*: 497–504.
- Thorsten Brants. 2000. Inter-annotator agreement for a German newspaper corpus, In Second International Conference on Language Resources and Evaluation LREC-2000, Athens, Greece: 69–76.
- Nianwen Xue, Fu-Dong Chiou and M. Palmer. 2002. Building a large-scale annotated Chinese corpus, In *Proceedings of COLING*.