Local context templates for Chinese constituent boundary prediction

Qiang Zhou
The State Key Laboratory of Intelligent Technology and Systems
Dept. of Computer Science and technology,
Tsinghua University, Beijing 100084
zhouq@s1000e.cs.tsinghua.edu.cn

Abstract:

In this paper, we proposed a shallow syntactic knowledge description: constituent boundary representation and its simple and efficient prediction algorithm, based on different local context templates learned from the annotated corpus. An open test on 2780 Chinese real text sentences showed the satisfying results: 94%(92%) precision for the words with multiple (single) boundary tag output.

1. Introduction

Research on syntactic parsing has been a focus in natural language processing for a long time. As the development of corpus linguistics, many statistics-based parsers were proposed, such as Magerman(1995)'s statistical decision tree parser, Collins(1996)'s bigram dependency model parser, Ratnaparkhi(1997)'s maximum entropy model parser. All of them tried to get the complete parse trees of the input sentences, based on the statistical data extracted from an annotated corpus. The best parsing accuracy of these parsers was about 87%.

Realizing the difficulties of complete parsing, many researches turned to explore the partial parsing techniques. Church(1988) proposed a simple stochastic technique for recognizing the non-recursive base noun phrases in English. Voutilaimen(1993) designed an English noun phrase recognition tool --- *NPTool*. Abney(1997) applied both rule-based and statistics-based approaches for parsing chunks in English. Due to the advantages of simplicity and robustness, these systems can be acted as good preprocessors for the further complete parsing.

In this paper, we will introduce our partial parsing approach for the Chinese language. We first proposed a shallow syntactic knowledge description: constituent boundary representation. It simplified the complex constituent levels in parse trees and only kept the boundary information of every word in different constituents. Then, we developed a simple and efficient constituent boundary prediction algorithm, based on different local context templates learned from the annotated corpus. An open test on 2780 Chinese real text sentences showed the satisfying results: 94%(92%) precision for the words with multiple (single) boundary tag output.

2. Constituent boundary description

The constituent boundary representation comes from the simplification of the complete parse trees of the sentences. It omits the constituent[1] levels in parse trees and only keeps the boundary information of every word in different constituents, i.e. it is at the left boundary, right boundary or middle position of a constituent.

Evidently, if the input sentence has only one parse tree, i.e. without syntactic ambiguity, the constituent boundary position of every word in the sentence is clear and definite. In the sense, the constituent boundary tag indicates the basic syntactic structure information in the sentence. Separating them from the constituent structure tree and assigning them to every word in the sentence, we can form a special syntactic unit: *word boundary block* (*WBB*).

Definition: A *word boundary block* is the combination of the word(including part-of-speech information) and its constituent boundary tag, i.e. wbb_i=<w_i, b_i>, where w_i is the *i*th word in the sentence, b_i can value 0,1,2, which means w_i is at

[1] Hereafter, 'constituent' represents all internal or root nodes in a parse tree, i.e. phrase or sentence tags. In our system, each constituent must consist of two or more words(leaf node in parser tree).

the middle, left-most, or right-most position of a constituent respectively.

In the view of syntactic description capability, the WBBs defined above, the chunks defined by Abney(1991) and the phrases(i.e. constituents) defined in a parse tree have the following realtions: WBBs < chunks < phrases

Here is an example:

- The input sentence (10 words): 我 的 弟弟 给 了 他 一 本 书 。 (My brother gives him a book.)
- Its parse tree representation (7 phrases): [$_{P1}$ [$_{P2}$ [$_{P3}$ 我 的 弟弟] [$_{P4}$ [$_{P5}$ 给 了] 他 [$_{P6}$ [$_{P7}$ 一 本] 书]]] 。]
- Its chunk representation (5 chunks): [$_{C1}$ 我 的 弟弟] [$_{C2}$ 给 了] [$_{C3}$ 他] [$_{C4}$ 一 本 书] [$_{C5}$ 。]
- Its constituent boundary representation (10 WBBs): <我,1> <的,0> <弟弟,2> <给,1> <了,2> <他,0> <一,1> <本,2> <书,2> <。,2>

The goal of the constituent boundary prediction is to assign a suitable boundary tag for every word in the sentence. It can provide basic information for further syntactic parsing research. The following lists some application examples:

- To develop a statistics-based Chinese parser(Zhou 1997) based on the bracket matching principle(Zhou and Huang,1997).
- To develop a Chinese maximum noun phrase identifier(Zhou,Sun and Huang, 1999).
- The automatic inference of Chinese probabilistic context-free grammar(PCFG) (Zhou and Huang 1998).

3. Local context templates

The linguistic intuitions tell us that many local contexts may be useful for constituent boundary prediction. For example, many function words in Chinese have their certain constituent boundary position in the sentences, such as, most prepositions are at the left boundaries, and the aspectual particles (“le”, “zhe”, “guo”) are at the right boundaries. Moreover, some content words also show their preferential constituent boundary positions in a special local context, such as most adjectives are at the right boundary in local context: “adverb + adjective”.

A tentative idea is how to use such simple local context information(including the part-of-speech(POS) tags and the number of Chinese characters(CN)) to develop an efficient automatic boundary prediction algorithm. Therefore, we defined the following local context templates (LCTs):

1) Unigram POS template: t_i, $BPFL_i$

2) Bigram POS templates:

- Left restriction: $t_{i-1}\ t_i$, $BPFL_i$
- Right restriction: $t_i\ t_{i+1}$, $BPFL_i$

3) Trigram POS template: $t_{i-1} t_i t_{i+1}$, $BPFL_i$

4) Trigram POS+CN template: $t_{i-1}+cn_{i-1}$ t_i+cn_i $t_{i+1}+cn_{i+1}$, $BPFL_i$

In the above LCTs, t_i is the POS tag of the *i*th word in the sentence, cn_i is its character number, and $BPFL_i$ is the frequency distribution list of its different BP(boundary prediction) value(0,1,2) under the local context restrictions(LCR)(the left and right word).

Table 1 Some examples of the local context templates

Type	Token	Meaning
Unigram	p, 39 849 476	A preposition is prior to at the constituent left boundary in Chinese.
bigram (left)	a n, 5 164 2007	A noun is prior to at the right boundary if its previous word is an adjective.
bigram (right)	a n, 4 2012 160	An adjective is prior to at the left boundary if its next word is a noun.
Trigram POS	n n uJDE, 1 18 1496	A noun is prior to at the right boundary if its previous word is a noun and its next one is a partial(De).

Table 1 shows some examples of LCTs. All these templates can be easily acquired from the Chinese treebanks or Chinese corpus annotated with constituent boundary tags.

Among these templates, some special ones have the following properties:

a) $TF_i = \sum BPFL_i\ [bp_i] > \alpha$,

b) $\exists bp_i \in [0,2]$, $P(bp_i|LCR_i)=BPFL_i\ [bp_i]\ /\ TF_i > \beta$

where the total frequency threshold α and the BP probability threshold β are set to 3 and 0.95, respectively. They are called the projected templates (PTs) (i.e. the local context template with a projecting BP value).

Based on the different PTs, we can design a three-stage training procedure to overcome the problem of data sparseness:

Stage 1 : Learn the unigram and bigram templates on the whole instances in annotated corpus.

Stage 2 : Learn the trigram POS templates on the non-projected unigram and bigram instances (see next section for more detailed).

Stage 3 : Learn the trigram POS+CN templates on the non-projected trigram POS instances.

Therefore, only the useful trigram templates can be learned.

4. Automatic prediction algorithm

After getting the LCTs, the automatic prediction algorithm becomes very simple: 1) to set the projecting BPs based on the projected LCTs, 2) to select the best BPs based on the non-projected LCTs. Some detailed information will be discussed in the following sections.

4.1 Set the projecting BPs

In this stage, the reference sequence to the LCTs is : unigram → bigram → trigram POS → trigram POS+CN, i.e. from the rough restriction LCTs to the tight restriction LCTs. This sequence is same with the LCT training procedure.

The detailed algorithm is as follows:

```
Input: the position of the ith word in the sentence.
Background: the LCTs learned from corpus.
Output: the projecting BP of the word – if found;
        -1 – otherwise.
Procedure:
  ● Get the local context of the ith word.
  ● If its unigram template is a PT, then return its projecting BP.
  ● If its left and right bigram template satisfy the following conditions:
      ➢ $TF_L + TF_R = \sum BPFL_L [j] + \sum BPFL_L [j] > \alpha$
      ➢ $P(bp_j \mid LCR_i) = (BPFL_L [j] + BPFL_R [j]) / (TF_L + TF_R ) > \beta$
    then return this combined projecting BP($bp_j$).
  ● If its trigram POS template is a PT, then return its projecting BP.
  ● If its trigram POS+CN template is a PT, then return its projecting BP.
```

4.2 Select the best BPs

In this stage, the reference sequence to the LCTs is : trigram POS+CN → trigram POS → bigram → unigram. It's a backing-off model (Katz,1987), just like the approach of Collins and Brooks(1995) for the prepositional phrase attachment problem in English. The detailed algorithm is as follows:

```
Input: the position of the ith word in the sentence.
Background: the LCTs learned from corpus.
Output: the best BP of the word.
Procedure:
  ● Get the local context of the ith word.
  ● For the kth matched trigram POS+CN templates, if $TF_k > \alpha$, then return SelectBestBP ($BPFL_k$).
  ● For the mth matched left bigram and nth matched right bigram,
      ➢ Get the Combined BPFL = $BPFL_m + BPFL_n$
      ➢ If $TF_{Combined\_template} > 0$, then return SelectBestBP(Combined BPFL).
  ● For the kth matched unigram templates, if $TF_k > 0$, then return SelectBestBP($BPFL_k$).
  ● Return 1(default is at the left boundary).
```

The internal function *SelectBestBP()* tries to select the best BP based on the frequency distribution list of different BP value in LCTs. It has two output modes: 1) single-output mode: only output the best BP with the highest frequency in the LCT; 2) multiple-output mode: output the BPs satisfying the conditions:

$|P_{bpi}-P_{best}| < \gamma$, where $\gamma = 0.2$

5. Experimental results

5.1 Training and test data

The training data were extracted from two different parts of annotated Chinese corpus:

1) The small Chinese treebank developed in Peking University(Zhou, 1996b), which consists of the sentences extracted from two parts of Chinese texts: (a) test set for Chinese-English machine translation systems, (b) Singapore primary school textbooks.

2) The test suite treebank being developed in Tsinghua University(Zhou and Sun,1999), which consists of about 10,000 representative Chinese sentences extracted from a large-scale Chinese balanced corpus with about 2,000,000 Chinese characters.

The test data were extracted from the articles of People's Daily and manually annotated with correct constituent boundary tags. It was also divided into two parts:

1) The ordinary sentences.

2) The sentences with keywords for conjunction structures (such as the conjunctions or special punctuation ‘DunHao’). They can be used to test the performance of our prediction algorithm on complex conjunction structures.

Table 2 shows some basic statistics of these training and test data. Only the sentences with more than one word were used for training and testing.

Table 2 The basic statistics of training and test data. (ASL = Average sentence length)

	Sent. Num.	Word Num.	Char. Num.	ASL (w/s)
Train1	5573	64426	89492	11.56
Train2	7774	108542	173334	13.96
Test1	2780	68986	108218	24.82
Test2	1071	32358	51169	30.21

5.2 The learned templates

After the three-stage learning procedure, we got four kinds of local context templates. Table 3 shows their different distribution data, where the section ‘Type’ lists the distribution of different kinds of LCTs and the section ‘Token’ lists the distribution of total words(i.e. tokens) covered by the LCTs. In the column ‘PTs’ and ‘Ratio’, the slash ‘/’ was used to separate the PTs with total frequency threshold 0 and 3.

More than 66% words in the training corpus can be covered by the unigram and bigram POS projected templates. Then only about 1/3 tokens will be used for training the trigram templates. Although the type distribution of the trigram templates shows the tendency of data sparseness (more than 70% trigram projected templates with total frequency less than 3), the useful trigram templates (TF>3) still covers about 70% tokens learned. Therefore, we can expect that them can play an important role during constituent boundary prediction in open test set.

5.3 Prediction results

In order to evaluate the performance of the constituent boundary prediction algorithm, the following measures were used:

1) The cost time(CT) of the kernal functions(CPU: Celeron$^{\text{TM}}$ 366, RAM: 64M).

2) Prediction precision(PP) =

$$\frac{\text{number of words with correct BPs(CortBP)}}{\text{total word number (TWN)}}$$

For the words with single BP output, the correct condition is:

Annotated BP = Predicted BP

For the words with multiple BP outputs, the correct condition is:

Annotated BP $\in$ Predicted BP set

The prediction results of the two test sets were shown in Table 4 and Table 5, whose first columns list the different template combinations using in the algorithm. In the columns ‘CortBP’ and ‘PP’, the slash ‘/’ was used to list the different results of the single and multiple BP outputs.

After analyzing the experimental results, we found:

1) The POS information in local context is very important for constituent boundary prediction. After using the bigram and trigram POS templates, the prediction accuracy was increased by about 9% and 3% respectively. But the character number information shows lower boundary restriction capability. Their application only results in a slight increase of precision in single-output mode but a slight decrease in multiple-output mode.

Table 3 Distribution data of different learned LCTs

LCTs	Type			Token		
	Total	PTs(α=0/3)	Ratio(α=0/3)	Total	PTs(α=0/3)	Ratio(α=0/3)
1-gram	59	24	40.68	171705	53932	31.41
2-gram(Left)	1448	1030 / 591	71.13 / 40.81	171705	87027 / 86339	50.68 / 50.28
2-gram(Right)	1440	1008 / 567	70.00 / 39.38	171705	99443 / 98754	57.92 / 57.51
3-gram (POS)	3105	2324 / 713	74.85 / 22.96	50333	24280 / 21982	48.24 / 43.07
3-gram(P+CN)	2553	1677 / 287	65.69 / 11.24	19098	5978 / 4079	31.30 / 21.36

Table 4 Experimental results of the test set 1

templates used	Set the Projecting BPs			Select the best BPs			Total			CT
	TWN	CortBP	PP(%)	TWN	CortBP	PP(%)	TWN	CortBP	PP(%)	
1-gram	22408	22143	98.82	46555	32876/24374	70.62/73.84	68963	55019/56517	79.78/81.95	14/16
+2-gram	46167	45285	98.09	22796	16188/17678	71.01/77.55	68963	61473/62963	89.14/91.30	11/15
+3-gram POS	55321	53969	97.56	13642	9946/10986	72.90/80.53	68963	63915/64955	92.68/94.19	13/11
+3-gram P+CN	57360	55866	97.40	11603	8168/8955	70.40/77.18	68963	64034/64821	92.85/93.99	11/14

2) Most of the prediction errors can be attributed to the special structures in the sentences, such as conjunction structures (CSs) or collocation structures. Due to the long distance dependencies among them, it's very difficult to assign the correct boundary tags to the words in these structures only according to the local context templates. The lower overall precision of the test set 2 (about 2% lower than test set 1) also indicates the boundary prediction difficulties of the conjunction structures, because there are more CSs in test set 2 than in test set 1.

3) The accuracy of the multiple output results is about 2% better than the single output results. But the words with multiple boundary tags constitute only about 10% of the total words predicted. Therefore, the multiple-output mode shows a good trade-off between precision and redundancy. It can be used as the best preprocessing data for the further syntactic parser.

4) The maximal ratio of the words set by projected templates can reach 80%. It guarantees the higher overall precision.

5) The algorithm shows high efficiency. It can process about 6,000 words per second (CPU: Celeron™ 366, RAM: 64M).

5.4 Compare with other work

Zhou(1996) proposed a constituent boundary prediction algorithm based on hidden Marcov model(HMM). The Viterbi algorithm was used to find the best boundary path *B'*:

$$B' = \arg\max P(W,T \mid B)P(B)$$

$$= \arg\max \prod_{i=1}^{n} P(CT_i \mid b_i)P(bi \mid b_{i-1})$$

where the local POS probability $P(CT_i \mid b_i)$ was computed by backing-off model and the bigram parameters: $f(t_{i-1}, t_i, b_i)$ and $f(b_i, t_i, t_{i+1})$.

To compare its performance with our algorithm, the trigram (POS and POS+CN) information was added up to its backing-off model. Table 6 and Table 7 show the prediction results of the HMM-based algorithm, based on the same parameters learned from training set 1 and 2.

Table 6. Prediction results of the HMM-based

Table 5 Experimental results of the test set 2

Templates Used	Set the Projecting BPs			Select the best BPs			Total			CT
	TWN	CortBP	PP(%)	TWN	CortBP	PP(%)	TWN	CortBP	PP(%)	
1-gram	10016	9873	98.57	22342	15737/16593	70.44/74.27	32358	25610/26466	79.15/81.79	6/5
+2-gram	21085	20454	97.00	11273	7856/8607	70.44/74.27	32358	28310/29061	87.49/89.81	4/4
+3-gram POS	24974	24079	96.42	7384	5225/5777	70.76/78.24	32358	29304/29856	90.56/92.27	3/6
+3-gram P+CN	25839	24866	96.23	6519	4525/4958	69.40/76.05	32358	29390/29824	90.83/92.17	8/6

algorithm(test set 1)

	TWN	CortBP	PP(%)	Ctime
2-gram	68963	60908	88.32	144
+ 3g-POS	68963	63397	91.93	138
+3g-P+CN	68963	63649	92.29	139

Table 7. Prediction results of the HMM-based algorithm(test set 2)

	TWN	CortBP	PP(%)	Ctime
+2-gram	32358	27792	85.89	68
+3g-POS	32358	28918	89.37	70
+3g-P+CN	32358	29030	89.72	68

The performance of the LCT-based algorithm surpassed the HMM-based algorithm in accuracy(about 1%) and efficiency (about 10 times).

Another similar work is Sun(1999). The difference lies in the definition of the constituent boundary tags: he defined them between word pair: w_i $\underline{b_i}$ w_{i+1}, not for the word. By using the HMM and Viterbi model, his algorithm showed the similar performance with Zhou(1996) (using bigram POS parameters):

- Training data : 3051 sentences extracted from People's Daily.
- Test data: 1000 sentences.
- Best precision: 86.3%

6. Conclusions

The paper proposed a constituent boundary prediction algorithm based on local context templates. Its characteristics can be summarized as follows:

- The simple definition of the local context templates made the training procedure very easy.
- The three-stage training procedure guarantees that only the useful trigram templates can be learned. Thus, the data sparseness problem was partially overcome.
- The high coverage of different types of projected templates assures a higher overall prediction accuracy.
- The multiple output mode provides the possibility to describe different boundary ambiguities.
- The algorithm runs very fast, surpasses the HMM-based algorithm in accuracy and efficiency.

There are a few possible improvement which may raise performance further. Firstly, some lexical-based templates, such as prepositions as left restriction, may improve performance further – this needs to be investigated. The introduction of the automatic identifiers for some special structures, such as conjunction structures or collocation structures, may reduce the prediction errors due to the long distance dependency problem. Finally, more training data is almost certain to improve results.

Acknowledgements

The research was supported by National Natural Science Foundation of China (NSFC) (Grant No. 69903007).

References

Abney S. (1991). "Parsing by Chunks", In Robert Berwick, Steven Abney and Carol Tenny (eds.) Principle-Based Parsing, Kluwer Academic Publishers.

Abney S. (1997). "Part-of-speech Tagging and Partial Parsing", In Young S. Bloothooft G. (eds.) Corpus-based methods in language and speech processings, 118-136.

Collins M. and Brooks J. (1995) "Prepositional Phrase Attachment through a Backing-Off Model", In David Yarowsky & Ken Church(eds.) Proceedings of the third workshop on very large corpora, MIT. 27-38.

Church K. (1988). "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text." In: Proceedings of Second Conference on Applied Natural Language Processing, Austin, Texas, 136-143.

Collins M. J. (1996). "A New Statistical Parser Based on Bigram Lexical Dependencies." In Proc. of ACL-34, 184-191.

Katz S. (1987). "Estimation of Probabilities from sparse data for the language model component of a speech recogniser". IEEE Transactions on ASSP, Vol .35, No. 3.

Magerman. D. M. (1995). "Statistical Decision-Tree Models for Parsing", In Proc. of ACL-95, 276-303.

Ratnaparkhi A.(1997). "A linear observed time statistical parser based on maximum entropy models". In Claire Cardie and Ralph Weischedel(eds.), Second Conference on Empirical Methods in Natural Language Processing(EMNLP-2), Somerset, New Jersey, ACL.

Sun H. L., Lu Q. and Yu S. W.(1999). "Two-level shallow parser for unrestricted Chinese text", In

Changning Huang and Zhendong Dong (eds.) Proceedings of Computational linguistics, Beijing: Tsinghua University press, 280-286.

Voutilamen A. (1993). “NPTool, a detector of English Noun Phrases.” In: Ken Church (ed.) Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives. Columbus, Ohio, USA, 48-57.

Zhou Q. (1996a). “A Model for Automatic Prediction of Chinese Phrase Boundary Location”,

Zhou Q. (1996b). Phrase Bracketing and Annotating on Chinese Language Corpus . Ph.D. Dissertation, Peking University.

Zhou Q. (1997) “A Statistics-Based Chinese Parser”, In Proc. of the Fifth Workshop on Very Large Corpora, 4-15.

Zhou Q. and Huang C.N. (1997) “A Chinese syntactic parser based on bracket matching principle”, Communication of COLIPS, 7(2), #97008.

Zhou Q. and Huang C.N. (1998). “An Inference Approach for Chinese Probabilistic Context-Free Grammar”, Chinese Journal of Computers, 21(5), 385-392.

Zhou Q. and Sun M.S. (1999). “Build a Chinese Treebank as the test suite for Chinese parser”, In Key-Sun Choi & Young-Soog Chae(eds.) Proceedings of the workshop MAL’99, Beijing. 32-37.

Zhou Q., Sun M.S. and Huang C.N.(1999) “Automatically Identify Chinese Maximal Noun Phrases”, Technical Report 99001, State Key Lab. of Intelligent Technology and Systems, Dept. of Computer Science and Technology, Tsinghua University.