Chinese sentence segmentation as comma classification

Nianwen Xue and Yaqin Yang
Brandeis University, Computer Science Department
Waltham, MA, 02453
{xuen,yaqin}@brandeis.edu

Abstract

We describe a method for disambiguating Chinese commas that is central to Chinese sentence segmentation. Chinese sentence segmentation is viewed as the detection of loosely coordinated clauses separated by commas. Trained and tested on data derived from the Chinese Treebank, our model achieves a classification accuracy of close to 90% overall, which translates to an F1 score of 70% for detecting commas that signal sentence boundaries.

1 Introduction

Sentence segmentation, or the detection of sentence boundaries, is very much a solved problem for English. Sentence boundaries can be determined by looking for periods, exclamation marks and question marks. Although the symbol (dot) that is used to represent period is ambiguous because it is also used as the decimal point or in abbreviations, its resolution only requires local context. It can be resolved fairly easily with rules in the form of regular expressions or in a machine-learning framework (Reynar and Ratnaparkhi, 1997).

Chinese also uses periods (albeit with a different symbol), question marks, and exclamation marks to indicate sentence boundaries. Where these punctuation marks exist, sentence boundaries can be unambiguously detected. The difference is that the Chinese comma also functions similarly as the English period in some context and signals the boundary of a sentence. As a result, if the commas are not disambiguated, Chinese would have these "run-on" sentences that can only be plausibly translated into multiple English sentences. An example is given in (1), where one Chinese sentence is plausibly translated into three English sentences.

(1) 这 段 时间一直在 留意 这
this period time AS AS pay attention to this
款 nano 3 ， [1] 还 专门 跑 了
CL Nano 3 , even in person visit AS
几 家 电脑 市场 , [2] 相比较
a few AS computer market , comparatively
而言 , [3] 卓越 的 价格 算
speaking , Zhuoyue ' s price relatively
低 的 , [4] 而且能 保证 是 行货
low DE , and can guarantee be genuine
， [5] 所以就 下 了 单 。
, therefore place [AS] order .

"I have been paying attention to this Nano 3 recently, [1] and I even visited a few computer stores in person. [2] Comparatively speaking, [3] Zhuoyue' s prices are relatively low, [4] and they can also guarantee that their products are genuine. [5] Therefore I placed the order."

In this paper, we formulate Chinese sentence segmentation as a comma disambiguation problem. The problem is basically one of separating commas that mark sentence boundaries (such as [2] and [5] in (1)) from those that do not (such as [1], [3] and [4]). Sentences that can be split on commas are generally loosely coordinated structures that are syntactically and semantically complete on their own, and they do not have a close syntactic relation with one another. We believe that a sentence boundary detection task that disambiguates commas, if successfully

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 631–635,
Portland, Oregon, June 19-24, 2011.

solved, simplifies downstream tasks such as parsing and Machine Translation.

The rest of the paper is organized as follows. In Section 2, we describe our procedure for deriving training and test data from the Chinese Treebank (Xue et al., 2005). In Section 3, we present our learning procedure. In Section 4 we report our results. Section 5 discusses related work. Section 6 concludes our paper.

2 Obtaining data

To our knowledge, there is no data in the public domain with commas explicitly annotated based on whether they mark sentence boundaries. One could imagine using parallel data where a Chinese sentence is word-aligned with multiple English sentences, but such data is generally noisy and commas are not disambiguated based on a uniform standard. We instead pursued a different path and derived our training and test data from the Chinese Treebank (CTB). The CTB does not disambiguate commas explicitly, and just like the Penn English Treebank (Marcus et al., 1993), the sentence boundaries in the CTB are identified by periods, exclamation and question marks. However, there are clear syntactic patterns that can be used to disambiguate the two types of commas. Commas that mark sentence boundaries delimit loosely coordinated top-level IPs, as illustrated in Figure 1, and commas that don't cover all other cases. One such example is Figure 2, where a PP is separated from the rest of the sentence with a comma. We devised a heuristic algorithm to detect loosely coordinated structures in the Chinese Treebank, and labeled each comma with either EOS (end of a sentence) or Non-EOS (not the end of a sentence).

3 Learning

After the commas are labeled, we have basically turned comma disambiguation into a binary classification problem. The syntactic structures are an obvious source of information for this classification task, so we parsed the entire CTB 6.0 in a round-robin fashion. We divided CTB 6.0 into 10 portions, and parsed each portion with a model trained on other portions, using the Berkeley parser (Petrov and Klein, 2007). The labels for the commas are derived

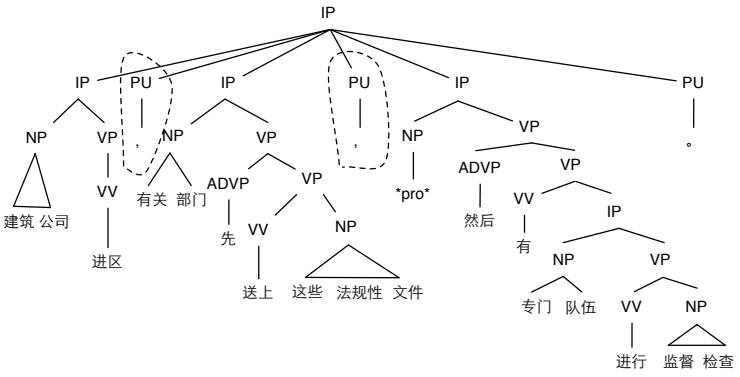

Figure 1: Sentence-boundary denoting comma

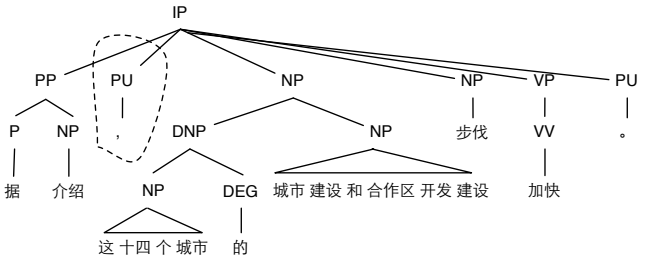

Figure 2: Non-sentence boundary denoting comma

from the gold-standard parses using the heuristics described in Section 2, as they obviously should be. We first established a baseline by applying the same heuristic algorithm to the automatic parses. This will give us a sense of how accurately commas can be disambiguated given imperfect parses. The research question we're trying to address here basically is: can we improve on the baseline accuracy with a machine learning model?

We conducted our experiments with a Maximum Entropy classifier trained with the Mallet package (McCallum, 2002). The following are the features we used to train our classifier. All features are described relative to the comma being classified and the context is the sentence that the comma is in. The actual feature values for the first comma in Figure 1 are given as examples:

1. Part-of-speech tag of the previous word, and the string representation of the previous word if it has a frequency of greater than 20 in the training corpus, e.g., *f1=VV*, *f2=*进区.
2. Part-of-speech of the following word and the

string representation of the following word if it has a frequency of greater than 20 in the training corpus, e.g., *f3=JJ, f4=*有关

3. The string representation of the following word if it occurs more than 12,000 times in sentence-initial positions in a large corpus external to our training and test data.[1]
4. The phrase label of the left sibling and the phrase label of their right sibling in the syntactic parse tree, as well as their conjunction, e.g, *f6=IP, f7=IP, f8=IP+IP*
5. The conjunction of the ancestors, the phrase label of the left sibling, and the phrase label of the right sibling. The ancestor is defined as the path from the parent of the comma to the root node of the parse tree, e.g., *f9=IP+IP+IP*.
6. Whether there is a subordinating conjunction (e.g., "if", "because") to the left of the comma. The search starts at the comma and stops at the previous punctuation mark or the beginning of the sentence, e.g., *f10=noCS*.
7. Whether the parent of the comma is a coordinating IP construction. A coordinating IP construction is an IP that dominates a list of coordinated IPs, e.g., *f11=CoordIP*.
8. Whether the comma is a top-level child, defined as the child of the root node of the syntactic tree, e.g., *f12=top*.
9. Whether the parent of the comma is a top-level coordinating IP construction, e.g., *f13=top+coordIP*.
10. The punctuation mark template for this sentence, e.g., *f14=*,+,+。
11. whether the length difference between the left and right segments of the comma is smaller than 7. The left (right) segment spans from the previous (next) punctuation mark or the beginning (end) of the sentence to the comma, e.g., *f15=>7*

4 Results and discussion

Our comma disambiguation models are trained and evaluated on a subset of the Chinese TreeBank (CTB) 6.0, released by the LDC. The unused portion of CTB 6.0 consists of broadcast news data that contains disfluencies, different from the rest of the CTB 6.0. We used the training/test data split recommended in the Chinese Treebank documentation. The CTB file IDs used in our experiments are listed in Table 1. The automatic parses in each test set are produced by retraining the Berkeley parser on its corresponding training set, plus the unused portion of the CTB 6.0. Measured by the ParsEval metric (Black et al., 1991), the parsing accuracy on the CTB test set stands at 83.63% (F-score), with a precision of 85.66% and a recall of 81.69%.

Data	Train	Test
CTB	41-325, 400-454, 500-554 590-596, 600-885, 900 1001-1078, 1100-1151	1-40 901-931

Table 1: Data set division.

There are 1,510 commas in the test set, and our heuristic baseline algorithm is able to correctly label 1,321 or 87.5% of the commas. Among these, 250 or 16.6% of them are EOS commas that mark sentence boundaries and 1,260 of them are Non-EOS commas. The results of our experiments are presented in Table 2. The baseline precision and recall for the EOS commas are 59.1% and 79.6% respectively with an F1 score of 67.8% . For Non-EOS commas, the baseline precision and recall are 95.7% and 89.0% respectively, amounting to an F1 score of 70.1%. The learned maximum classifier achieved a modest improvement over the baseline. The overall accuracy of the learned model is 89.2%, just shy of 90%. The precision and recall for EOS commas are 64.7% and 76.4% respectively and the combined F1 score is 70.1%. For Non-EOS commas, the precision and recall are 95.1% and 91.7% respectively, with the F1 score being 93.4%. Other than a list of most frequent words that start a sentence, all the features are extracted from the sentence the comma occurs in. Given that the heuristic algorithm and the learned model use essentially the same source of information, we attribute the improvement to the use of lexical features that the heuristic algorithm cannot easily take advantage of.

Table 3 shows the contribution of individual feature groups. The numbers reflect the accuracy when each feature group is taken out of the model. While all the features have made a contribution to the over-

[1]This feature is not instantiated here because the following word in this example does not occur with sufficient accuracy.

	Baseline			Learning		
(%)	p	r	f1	p	r	f1
Overall			87.5			89.2
EOS	59.1	79.6	67.8	64.7	76.4	70.1
Non-EOS	95.7	89.0	92.2	95.1	91.7	93.4

Table 2: Accuracy for the baseline heuristic algorithm and the learned model

all accuracy on the development set, some of the features (3 and 8) actually hurt the overall performance slightly on the test set. What's interesting is while the heuristic algorithm that is based entirely on syntactic structure produced a strong baseline, when formulated as features they are not at all effective. In particular, feature groups 7, 8, 9 are explicit reformulations of the heuristic algorithm, but they all contributed very little to or even slightly hurt the overall performance. The more effective features are the lexical features (1, 2, 10, 11) probably because they are more robust. What this suggests is that we can get reasonable sentence segmentation accuracy without having to parse the sentence (or rather, the multi-sentence group) first. The sentence segmentation can thus come before parsing in the processing pipeline even in a language like Chinese where sentences are not unambiguously marked.

	overall	f1 (EOS)	f1 (non-EOS)
all	89.2	70.1	93.4
- (1,2)	87.5	67.7	92.3
-10	87.8	67.5	92.5
-11	88.6	68.6	93.1
-4	89.0	69.6	93.3
-5	89.1	69.5	93.3
-6	89.1	69.9	93.4
-7	89.1	70.1	93.4
-9	89.1	69.7	93.3
-8	89.2	70.5	93.4
- 3	89.4	70.5	93.5

Table 3: Feature effectiveness

5 Related work

There has been a fair amount of research on punctuation prediction or generation in the context of spoken language processing (Lu and Ng, 2010; Guo et al., 2010). The task presented here is different in that the punctuation marks are already present in the text and we are only concerned with punctuation marks that are semantically ambiguous. Our specific focus is on the Chinese comma, which sometimes signals a sentence boundary and sometimes doesn't. The Chinese comma has also been studied in the context of syntactic parsing for long sentences (Jin et al., 2004; Li et al., 2005), where the study of comma is seen as part of a "divide-and-conquer" strategy to syntactic parsing. Long sentences are split into shorter sentence segments on commas before they are parsed, and the syntactic parses for the shorter sentence segments are then assembled into the syntactic parse for the original sentence. We study comma disambiguation in its own right aimed at helping a wide range of NLP applications that include parsing and Machine Translation.

6 Conclusion

The main goal of this short paper is to bring to the attention of the field a problem that has largely been taken for granted. We show that while sentence boundary detection in Chinese is a relatively easy task if formulated based on purely orthographic grounds, the problem becomes much more challenging if we delve deeper and consider the semantic and possibly the discourse basis on which sentences are segmented. Seen in this light, the central problem to Chinese sentence segmentation is comma disambiguation. We trained a statistical model using data derived from the Chinese Treebank and reported promising preliminary results. Much remains to be done regarding how sentences in Chinese should be segmented and how this problem should be modeled in a statistical learning framework.

Acknowledgments

This work is supported by the National Science Foundation via Grant No. 0910532 entitled "Richer Representations for Machine Translation". All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

References

E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 306–311.

Yuqing Guo, Haifeng Wang, and Josef Van Genabith. 2010. A Linguistically Inspired Statistical Model for Chinese Punctuation Generation. *ACM Transactions on Asian Language Processing*, 9(2).

Meixun Jin, Mi-Young Kim, Dong-Il Kim, and Jong-Hyeok Lee. 2004. Segmentation of Chinese Long Sentences Using Commas. In *Proceedings of the SIGHANN Workshop on Chinese Language Processing*.

Xing Li, Chengqing Zong, and Rile Hu. 2005. A Hierarchical Parsing Approach with Punctuation Processing for Long Sentence Sentences. In *Proceedings of the Second International Joint Conference on Natural Language Processing: Companion Volume including Posters/Demos and Tutorial Abstracts*.

We Lu and Hwee Tou Ng. 2010. Better Punctuation Prediction with Dynamic Conditional Random Fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, MIT, Massachusetts.

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Slav Petrov and Dan Klein. 2007. Improved Inferencing for Unlexicalized Parsing. In *Proc of HLT-NAACL*.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C.

Nianwen Xue, Fei Xia, Fu dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2):207–238.