A Maximum-Entropy Chinese Parser Augmented by Transformation-Based Learning

PASCALE FUNG[1],GRACE NGAI[2],YONGSHENG YANG[1], AND BENFENG CHEN[1]
[1]Hong Kong University of Science and Technology
[2]Hong Kong Polytechnic University

Parsing, the task of identifying syntactic components, e.g., noun and verb phrases, in a sentence, is one of the fundamental tasks in natural language processing. Many natural language applications such as spoken-language understanding, machine translation, and information extraction, would benefit from, or even require, high accuracy parsing as a preprocessing step.

Even though most state-of-the-art statistical parsers were initially constructed for parsing in English, most of them are not language-specific, in that they do not rely on properties of the language that are specific to English. Therefore, construction of a parser in a given language becomes a matter of retraining the statistical parameters with a Treebank in the corresponding language.

The development of the Chinese treebank [Xia et al. 2000] spurred the construction of parsers for Chinese. However, Chinese as a language poses some unique problems for the development of a statistical parser, the most apparent being word segmentation. Since words in written Chinese are not delimited in the same way as in Western languages, the first problem that needs to be solved before an existing statistical method can be applied to Chinese is to identify the word boundaries. This is a step that is neglected by most pre-existing Chinese parsers, which assume that the input data has already been pre-segmented.

This article describes a character-based statistical parser, which gives the best performance to-date on the Chinese treebank data. We augment an existing maximum entropy parser with transformation-based learning, creating a parser that can operate at the character level. We present experiments that show that our parser achieves results that are close to those achievable under perfect word segmentation conditions.

Categories and Subject Descriptors: Artificial intelligence, natural language processing, language parsing and understanding
General Terms: Languages, learning, parsing
Additional Key Words and Phrases: Parsing for Chinese, maximum entropy, transformation-based learning, POS tagging, chunking and parsing for Chinese

1. INTRODUCTION

The goal of computational linguistics is to let computers understand human language, so that we can build intelligent computer systems, e.g., machine-translation systems, speech-recognition systems, and automatic dialog systems. The first step towards this goal is to represent the inherent structures in a language in a format that is suitable for computer manipulation, computation, and analysis.

This research was partially supported by CERG no. HKUST6213/02E and HKUST6206/03E from the Hong Kong Government's Research Grants Council, and grants A-PE37 and 4-Z03S from the Hong Kong Polytechnic University.

Authors' addresses: Pascale Fung, Yongsheng Yang and Benfeng Chen, Human Language Technology Center, Department of Electrical and Electronic Engineering, Hong Kong University of Science & Technology, Clear Water Bay, Hong Kong({pascale,ysyang,bfchen}@ee.ust.hk). Grace Ngai, Department of Computing, Hong Kong Polytechnic University, Hong Kong(csgngai@polyu.edu.hk).

Parsing, the task of reconstructing the syntactic-phrase-structure tree that lies behind each sentence, is one of the most popular of the basic NLP problems. A parser takes a sequence of words as input and outputs the parse tree that best fits the syntactic structure of the sentence.

The resource that almost all statistical syntactic parsers depend on, to some degree or other, is the availability of treebanks, which are collections of sentences and their corresponding hand-annotated parse trees. The language-independent nature of statistical methods, together with the recent release of treebanks in various languages, has led to a flurry of research in parsing in languages other than English. However, even though statistical parsers may have been designed to be language-independent, languages usually have their own idiosyncrasies that complicate the adaptation of pre-existing parsers. Chinese, in particular, with its lack of well-defined words, poses a unique and serious problem, as most NLP tools, parsers included, assume the existence of words.

In this article, we present a method that extends a pre-existing, state-of-the-art maximum entropy English parser to Chinese. The maximum entropy parser is augmented with transformation-based learning for segmentation and POS tagging, creating a parser that accepts a character sequence as input. We present experiments that show that our parser outputs parse trees that are close to what can be expected under perfect word-segmentation conditions. Our system also outperforms a parser that is based on maximum entropy only.

2. SYNTACTIC PARSING

Parsing is the task of identifying phrases of a sentence and describing the syntactic relations among them. A parser takes a raw sentence as input and outputs a parse tree, or the set of syntactic relations, that best fits the input sentence. Basically, the parse tree gives information on how a sentence is composed of several sub-sentence segments and how these segments in turn consist of smaller segments. For example, Figure 1 shows the parse tree for the sentence “第七届世界游泳锦标赛在罗马开幕”, whose equivalent bracket structure is

```
(IP (NP (QP ( OD 第七) ( CLP ( M 届)))
     ( NP (NN 世界) (NN 游泳) ( NN 锦标赛)))
           (VP (PP (P 在) (NP (NR 罗马)))
                (VP ( VV 开幕))))
```

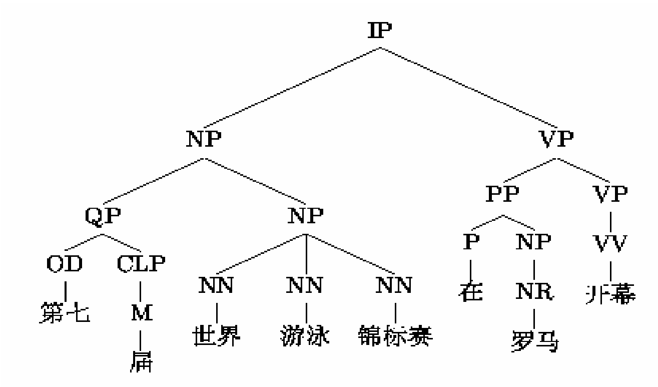

Fig. 1. Example of a parsed sentence (tree format).

Statistical syntactic parsing is an area that has attracted a lot of recent attention, and arguably, the resource that has contributed the most to making this a popular research topic is the construction of *treebanks*, a collection of sentences and their corresponding hand-annotated parse trees. The American English Penn Treebank [Marcus et al. 1993] is perhaps one of the best-known and widely-used resources in all of natural language processing.

The Chinese Penn Treebank [Xia et al. 2000], or the CTB, was released in 2000 and consists of 100,000 words (4185 sentences) from Xinhua newswire articles between 1994-98. The parse structures of the CTB follow roughly the same annotations and guidelines as the original Penn Treebank. As statistical syntactic parsers are often language-independent, and much work has been done on parsing on the English Penn Treebank; it suggests that given the release of the CTB, the development of a Chinese parser would simply involve retraining one of the pre-existing parsers on Chinese data. This would be a reasonable assumption for many languages, but the lack of well-formed words in Chinese poses a unique problem for parsers designed for the Indo-European languages. Hence, most earlier attempts at Chinese parsing [Bikel and Chiang 2000; Xu et al. 2002] have ignored the problem of word segmentation and assumed gold-standard (i.e., hand-annotated CTB) word boundaries, which is not realistic.

The problem of Chinese word segmentation is fairly severe, due to the lack of a good definition for what constitutes a word in Chinese. Previous experiments involving native speakers achieved an agreement rate of only around 75% [Wu and Fung 1994; Sproat et al. 1996]. Most Chinese word segmenters make use of large lexicons of manually-defined words; but the limitation of this method is that the word lists have to be constructed manually, which is a tedious and time-consuming process. In addition, the word lists constructed are heavily dependent on the domain at hand, as words vary from domain to domain.

This article presents a method in which a maximum entropy parser, which has state-of-the-art results for English syntactic parsing, is augmented by a transformation-based learner. The combined parser, which is purely corpus-driven, takes a raw, unsegmented Chinese sentence as input and outputs the parse tree that best fits the sentence.

2. THE MAXIMUM ENTROPY FRAMEWORK

The maximum entropy framework is a powerful learning model, which has been successfully employed for many natural language processing tasks, from part-of-speech tagging to word-sense disambiguation [Ratnaparkhi 1998].

The philosophy that maximum entropy works on is this: Given any set of training data, we will usually have some prior information from which to estimate the probability distribution, i.e., some constraints that the data is known to follow. However, this information is not complete, and the remaining uncertainty can be measured through entropy. Thus the maximum entropy principle seeks the distribution that maximizes the entropy of the distribution subject to the known constraints. The idea is to be "maximally noncommittal" about what we do not know, while still agreeing with what we do know [Jaynes 1968]. The advantage of maximum entropy is that it is robust and statistically efficient, while still allowing for easy representation and incorporation of different features [Ratnaparkhi 1998].

The Maximum Entropy Framework for Parsing

Maximum entropy has been applied to many NLP tasks, part-of-speech tagging and parsing among them, achieving state-of-the-art results. Our maximum entropy parser is a reimplementation of Ratnaparkhi [1998], which breaks down the parsing process into three steps: tagging, chunking, and tree-building. The output of each component is piped into the next as input. Since we are working at the word level, a fourth component, the word segmenter, is added as an initial (pre-processing) step. Since maximum entropy models are inherently classifiers, the various subtasks are mapped to classification tasks.

The probabilistic models for all the subtask components follow the form:

$$P(T \mid W) = \prod_{i=0}^{n} P(y_i \mid x_i)$$

$$P(y_i \mid x_i) = \frac{1}{Z(x_i)} \exp\left(\sum_{j=0}^{m} \lambda_j \bullet f_j(x_i, y_i) \right)$$

where $W=\{w_0, w_1, \ldots, w_n\}$ is the input character sequence, and T is the most likely output tag sequence for the corresponding component model. The output classification for the ith sample is denoted y_i, which are determined as a probability of the given contextual features x_i.

Word Segmentation. Unlike English and many other Western languages, Chinese (and Japanese) is written with no spaces between words. The task of word segmentation, therefore, attempts to word-delimit a text by inserting indicators that mark the boundaries between pre-defined words. The difficulty of Chinese word segmentation lies in the ambiguity of the task, as well as the fact that for any given sentence, there may be more than one valid word-segmentation sequence.

Since words follow one another and do not overlap, the task of word segmentation can be easily mapped to a tagging problem in a similar way to that pioneered by Ramshaw and Marcus [1995[for English text-chunking. The character that *begins* a word segment is tagged with a "B" while all other words are tagged with "I" to denote that they are *inside* a word segment. Each sample is therefore a *character*, and the output classification is the word-segment tag {B,I} that best fits the character in that context.

Part-of-Speech Tagging. Part-of-speech (POS) tagging, or simply tagging, is one of the most basic tasks in natural language processing. The task involves labeling each word in a sentence with a tag indicating its part-of-speech function (e.g., noun, verb, adjective, etc). Since many words have more than one POS tag, the task of the tagger is to use lexical and syntactic features of the word to determine the most likely tag for that particular use of the word in the given sentence. The problem of part-of-speech ambiguity is especially severe for Chinese, since Chinese words lack morphological information, which is an important indicator for syntactic function.

Since POS tagging is already a classification task, no extra steps are needed to map it for classification algorithms. Each sample is naturally a *word*, and the output classification, the POS tag (e.g., noun, verb, adjective) is the most appropriate for that particular word instance.

Text-Chunking. An intermediate step between POS tagging and full parsing is text-chunking, which is dividing a sentence into syntactically correlated segments called *chunks*, or base phrases. Unlike parse constituents, chunks are non-recursive and are usually based on superficial syntactic analysis. For example, the sentence "第七届世界游泳锦标赛在罗马开幕" can be chunked as

［$_{QP}$ 第七］［$_{CLP}$ 届］［$_{NP}$ 世界游泳锦标赛］［$_{PP}$ 在］［$_{NP}$ 罗马］［$_{VP}$ 开幕］

where each text chunk is delimited with brackets ([…]) that are annotated with chunk type.

Since text chunks are non-recursive, the task of text-chunking can easily be mapped to that of word classification [Ramshaw and Marcus 1995]. Each word is tagged with information that denotes the chunk that the word is in (e.g., NP, VP, PP), as well as the position of the word within the chunk (i.e., begin or inside). Words that did not fit inside any chunk were classified as *outside*. The fact that text chunks are non-overlapping makes it possible to deterministically map all possible sequences of text chunks to some sequence of *chunk tags*. A sample for the chunker component would then be a word and the classification of its corresponding chunk tag.

As an example of the various subtasks, Figure 2 shows an example sentence tagged with word segment, POS, and chunk tags.

Char	第	七	届	世	界	游	泳	锦	标	赛	在	罗	马	开	幕
Word	B	I	B	B	I	B	I	B	I	I	B	B	I	B	I
POS	OD		M	NN		NN		NN			P	NP		VV	
Chunk	B-QP		B-CLP	B-NP		I-NP		I-NP			B-PP	B-NP		B-VP	

Fig. 2. Marking word segment, POS, and chunk tags for a sentence.

Building Parse Trees. Due to the recursive structure of a parse tree, the task of the tree builder is substantially more complicated than that of the tagger or chunker, and samples cannot be mapped easily to individual words. Rather, the build component focuses on the *nodes* within the parse tree. The task of the builder is then to join nodes together to form new nodes, and thus build the tree from the bottom up.

Given a tagged and chunked sentence, we define our initial set of nodes to be the set of chunk segments. The builder then attempts to classify each node with an event tag *(n,b,c)*, where

- *n* denotes the current node (e.g., NP, VP);
- *b* denotes the build state, which specifies whether the current node should start or join a parent node. If the current node is the first child node of its parent, the node is assigned a state of "Start"+<*parent node*>. Otherwise, the build state of the node would be "Join"+<*parent node*>.
- *c* denotes the check state. This has two possible outcomes, YES and NO, and specifies whether the current node is the last child node of its parent.

As new nodes are formed by the build state, the tree is constructed recursively in a bottom-up fashion from left to right, until the root has been reached. A sample for the build component is therefore a given node, and the classification would be the event that is associated with the node. The algorithm is as follows:

```
Input:    a sequence of m initial nodes W={w_1, w_2, …, w_m}
          a sequence of k events E={e_1, e_2, …, e_k}
Output:  a tree
Algorithm:
begin
  i = 1;
  set current node w =w_i
  for j =1 to k do
     (n, b, c) = e_j
     if b is a "start" action do
        create a parent node p for w;
     else if b is a "join" action
        add w to be a son of p
     end
     if c is "yes" do
       w = p;
     else
       i = i +1
       w = w_i;
     end
  end
  output the tree T generated by events
end
```

Figure 3 shows the features used for the various parser components. As can be seen, with the exception of the word segmenter, all components use both lexical and syntactic/syntagmatic contextual information.

Component	Pattern	Meaning
Word segmenter	char[i], i ∈ {-2,-1,0,1,2}	Lexical identity of target and context characters
Tagger/ Chunker	word[i],i ∈ {-2,-1,0,1,2}	Lexical identity of target and context words
	pos[i],i ∈ {-2,-1}	Assigned POS/chunk tag of previous words
	(pos)[-2],[-1]	Bigram of assigned POS/chunk tags of previous words
	prefix	The first 1 and 2 characters of current word
	suffix	The last 1 and 2 characters of current word
	numeric	The word contains a Chinese number
Builder	cons(n)*, n ∈ [−1,2]	The head word, constituent (or POS) label, and Start/Join state of the *n*th node.
	cons(n-1, n)*, n ∈ [−1,0]	Bigram of cons(n)
	cons(n-2,n-1,n)*, n ∈ [−1,0]	Trigram of cons(n)
	bracketsmatch	The constituent we could join consists of a “|” and the current node is “]”
	iscomma	The constituent we could join consists of an “,”, and the current node is “.”
	endofsentence	The constituent we could join spans the entire sentence and current node is “.”

Fig. 3. Features for maximum entropy parser components.

Maximum Entropy Word Segmentation

As can be seen from the components, the maximum entropy parser is capable of performing word segmentation by itself. However, one major disadvantage of the maximum entropy word segmenter is its reliance on lexical features in the training set. As a result, its generalization ability is much reduced, and thus leads to poor word-segmentation accuracy: training on 80% of the training set achieved a precision/recall performance of only 74/72%.

3. TRANSFORMATION-BASED WORD SEGMENTATION AND POS TAGGING

It is well known that in piped systems, where later components work on the results of earlier ones, that mistakes made by the earlier systems will be propagated down the chain. Since the word segmenter is the first component in the parser, its poor performance creates a big problem for our parser.

The solution to this problem is to leverage another successful machine-learning algorithm, transformation-based learning (TBL) [Brill 1995]. In TBL, the system is provided with a training set, upon which an initial class assignment has been made, often based on some simple statistics. In addition, a set of allowable rule templates are also provided: these templates control the types of rules which may be learned. The system then learns a set of rules that progressively improve upon the current state of the training set, with each rule evaluating on the result of the previous ones. (For a detailed description of the TBL algorithm; see Brill [1995].)

One useful advantage of TBL over maximum entropy for tackling the word-segmentation problem is its ability to handle multi-class problems [Florian and Ngai 2001]. One of the biggest problems facing the word segmenter was its inability to generalize over different character types; this led to a sparse-data problem, and resulted in a hit to performance. Many other tasks mitigate this problem through the use of syntactic features, but as syntactic features such as POS tags rely on the presence of words, which the word-segmentation process is trying to achieve, this leads to a chicken-and-egg problem.

Multi-class TBL, in which the system performs multiple classification tasks (e.g., word segmentation and POS tagging) at the same time, tackles this problem by training jointly and simultaneously on all fields. This allows the system to model all possible dependencies between the tasks without imposing any ordering upon them. Florian and Ngai [2001] performed experiments with the multi-class TBL on four problems: English POS-tagging and text-chunking, and Chinese word segmentation and POS-tagging. In all cases, the simultaneously trained classifiers outperformed their sequentially trained counterparts. In addition, on our training and testing set, the multi-class TBL word segmenter/POS tagger achieved a word-segmentation precision/recall performance of 93/94%, easily outperforming the maximum entropy word segmenter.

EXPERIMENTS

The experiments in this article were performed using version 1 of the Chinese Treebank [Xia et al. 2000], which has a total of around 100,000 words. To facilitate comparison with current work, Sections 001-270 (approximately 90% of the CTB) were used for training, and Sections 271-300 (approximately 10%) for evaluation. The remaining sections (301-325) were held for later development/tuning purposes. This is the same data split that was used in Bikel and Chiang [2000].

Evaluation Metrics

The evaluation metrics for parsing are the standard precision, recall, and f-measure, which are defined as: *F-measure = 2* precision* recall / (precision + recall)*

Table I. MaxEnt Parsing Augmented by TBL.

	Precision	Recall	F-Measure
TBL segmentation + ME tagging/chunk/build	76.05%	74.41%	75.22%
TBL segmentation/tagging + ME chunk/build	74.83%	75.34%	75.09%
ME segmentation/tagging/chunk/build	59.32%	56.99%	58.13%
Gold standard segmentation + ME tagging/chunk/build	80.85%	78.30%	79.56%

Table I shows various configurations for word-segmentation/POS-tagging/text-chunking/parse-tree-building. The result of maximum entropy tagging and parsing on top of *gold-standard* word-segment boundaries is provided as a reference. As expected, the pure maximum entropy system achieves the poorest performance. This is not surprising, given the poor word-segmentation results. The encouraging thing, however, is that by incorporating a corpus-based word segmenter (i.e., without the use of time-consuming word lists) we were able to achieve performances close to gold-standard word segmentations (if they were available).

One interesting fact about the maximum entropy parser and transformation-based word segmenter is that they were both built using language-independent methods. The architecture and features of the maximum entropy parser were essentially the same as those for parsing in English, while the multi-class transformation-based tagger used features that were largely adapted from those for simultaneous POS-tagging and text-chunking for English. This further illustrates the robustness and language-independence of statistical supervised NLP methods.

COMPARISONS TO RELATED WORK

The most directly comparable work to ours on Chinese syntactic parsing is that of Bikel and Chiang [2000], who constructed two parsers: a lexicalized PCFG and a statistical TAG on the first release of the treebank. They used the same training/test-set split, and achieved a precision/recall performance of 73.3/74.6%. However, this result is not directly comparable to our work for two reasons: their parser was built on top of gold-standard word segmentations and their evaluation metric did not consider POS-tagged words to be constituents. Likewise, Xu et al. [2002] constructed a PCFG parser on CTB v.1.0, which achieved 75.2 f-measure; but is not directly comparable for the same reasons.

Another comparable work to the present work is that of Luo [2003], who constructed a character-based parser that achieved up to 81.4 f-measure, but unfortunately was trained and tested on the second version of the CTB, which is approximately twice as big as our version. Similarly, Levy and Manning [2003] achieved an 78.8 f-measure, but on CTB 2.0 and under a different training/test split.

Word segmentation has been widely studied, and many different approaches have been applied to the problem. However, due to the ambiguity of the *definition* of the problem [Wu and Fung 1994; Sproat et al. 1996], and the lack of a common training/test corpus, it is very difficult to compare systems objectively. Among the most relevant approaches are those of Palmer [1997] and Hockenmeier and Brew [1998], who both used corpus-based, supervised transformation-based approaches to tackle the problem. Hockenmeier and Brew achieved an f-measure performance of 87.8 after training on a corpus of 100,000 words, and Palmer's system achieved an f-measure of 87.7 on a corpus of 60,000 words.

CONCLUSION

The release of the Chinese Treebank has spurred much interest in statistical parsing for Chinese. However, this task is complicated by the fact that Chinese does not have well-defined word boundaries, something that most statistical NLP tools depend upon. We propose to augment a maximum entropy Chinese parser with transformation-based word segmentation and POS tagging, as TBL can take both features into account simultaneously. Our final system works at the character level and outperforms the state-of-the-art maximum entropy parser for Chinese. Our experiments show that the combined system achieves performances close to what could be obtained with gold-standard word segment annotations.

ACKNOWLEDGMENTS

The authors would also like to thank Radu Florian at IBM Research for providing the rule templates used by the transformation-based word segmenter and POS tagger.

REFERENCES

BIKEL, D. M. AND CHIANG, D. 2000. Two statistical parsing models applied to the Chinese treebank. In *Proceedings of the Second Chinese Language Processing Workshop* (Hong Kong, 2000). 1–6.

BRILL, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics 21*, 4 (1995), 543-565.

FLORIAN, R. AND NGAI, G. 2001. Multidimensional transformation-based learning. In *Proceedings of CONLL'01* (Toulouse, France, 2001). 1-8.

HOCKENMEIER, J. AND BREW, C. 1998. Error-driven segmentation of Chinese. *Communications of COLIPS 8,* 1 (1998), 69-84.

JAYNES, E. T. 1984. Monkeys, kangaroos and n. In *Maximum Entropy and Bayesian Methods in Applied Statistics*: *Proceedings of the 4th Maximum Entropy Workshop* (Univ. of Calgary, 1984). J. H. Justice (ed.). 26-58.

LEVY, R. AND MANNING, C. D. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of the 41st ACL* (Sapporo, Japan, 2003). 439-446.

LUO, X. 2003. A maximum entropy Chinese character-based parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (July 2003).

MARCUS, M., MARCINKIEWICZ, M., AND SANTORINI, B. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics 19*, 2 (1993), 313-330.

PALMER, D. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the 35th ACL* (Madrid 1997).

RAMSHAW, L. AND MARCUS, M. 1995. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL Workshop on Very Large Corpora* (Cambridge, MA, 1995).

RATNAPARKHI, A. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the EMNLP* (University of Pennsylvania, 1996).

RATNAPARKHI, A. 1998. Maximum entropy models for natural language ambiguity resolution. Ph.D. dissertation*,* Univ. of Pennsylvania, Philadelphia, 1998.

SPROAT, R., SHIH, C., GALE, W., AND CHANG, N. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics 22*, 3 (1996).

XU, J., MILLER, S., AND WEISCHEDEL, R. 2002. A statistical parser for Chinese. In *Proceedings of the 2002 Human Language Technology Workshop*.

XIA, F. AND PALMER, M., ET AL. 2000. Developing guidelines and ensuring consistency for Chinese text annotation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation* (LREC-2000, Athens, 2000).

Wu, D. and Fung, P. 1994. Improving Chinese word segmentation with linguistic filters on statistical lexical acquisition. In *Proceedings of the Fourth ACL Conference on Applied Natural Language Processing* (Stuttgart, Germany,1994).

Received August 2003; revised May 2004; accepted June 2004