Chinese Parsing Exploiting Characters

Meishan Zhang[†], **Yue Zhang**[‡*], **Wanxiang Che**[†], **Ting Liu**[†]
[†]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{mszhang, car, tliu}@ir.hit.edu.cn
[‡]Singapore University of Technology and Design
yue_zhang@sutd.edu.sg

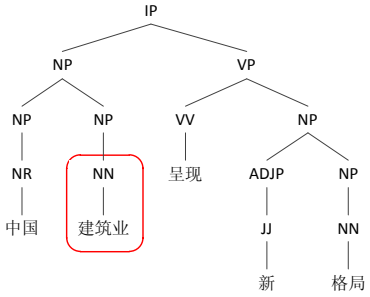
Abstract

Characters play an important role in the Chinese language, yet computational processing of Chinese has been dominated by word-based approaches, with leaves in syntax trees being words. We investigate Chinese parsing from the character-level, extending the notion of phrase-structure trees by annotating internal structures of words. We demonstrate the importance of character-level information to Chinese processing by building a joint segmentation, part-of-speech (POS) tagging and phrase-structure parsing system that integrates character-structure features. Our joint system significantly outperforms a state-of-the-art word-based baseline on the standard CTB5 test, and gives the best published results for Chinese parsing.

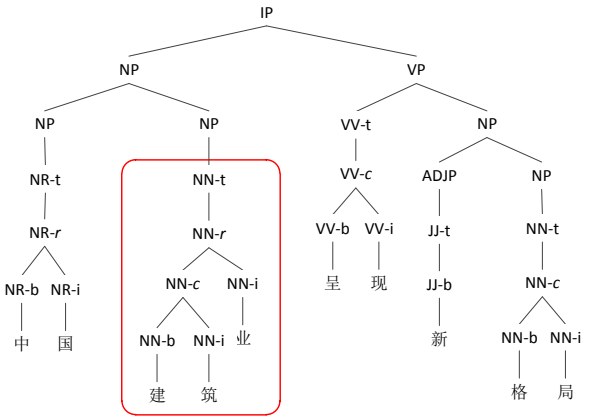
1 Introduction

Characters play an important role in the Chinese language. They act as basic phonetic, morpho-syntactic and semantic units in a Chinese sentence. Frequently-occurring character sequences that express certain meanings can be treated as words, while most Chinese words have syntactic structures. For example, Figure 1(b) shows the structure of the word "建筑业 (construction and building industry)", where the characters "建 (construction)" and "筑 (building)" form a coordination, and modify the character "业 (industry)".

However, computational processing of Chinese is typically based on words. Words are treated as the atomic units in syntactic parsing, machine translation, question answering and other NLP tasks. Manually annotated corpora, such as the Chinese Treebank (CTB) (Xue et al., 2005), usually have words as the basic syntactic elements



(a) CTB-style word-based syntax tree for "中国 (China) 建筑业 (architecture industry) 呈现 (show) 新 (new) 格局 (pattern)".



(b) character-level syntax tree with hierarchal word structures for "中 (middle) 国 (nation) 建 (construction) 筑 (building) 业 (industry) 呈 (present) 现 (show) 新 (new) 格 (style) 局 (situation)".

Figure 1: Word-based and character-level phrase-structure trees for the sentence "中国建筑业呈现新格局 (China's architecture industry shows new patterns)", where "l", "r", "c" denote the directions of head characters (see section 2).

(Figure 1(a)). This form of annotation does not give character-level syntactic structures for words, a source of linguistic information that is more fundamental and less sparse than atomic words.

In this paper, we investigate Chinese syntactic parsing with character-level information by extending the notation of phrase-structure

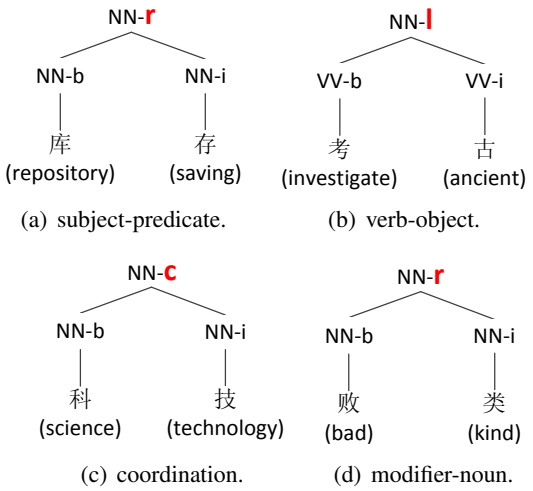
*Email correspondence.

(constituent) trees, adding recursive structures of characters for words. We manually annotate the structures of 37,382 words, which cover the entire CTB5. Using these annotations, we transform CTB-style constituent trees into character-level trees (Figure 1(b)). Our word structure corpus, together with a set of tools to transform CTB-style trees into character-level trees, is released at https://github.com/zhangmeishan/wordstructures. Our annotation work is in line with the work of Vadas and Curran (2007) and Li (2011), which provide extended annotations of Penn Treebank (PTB) noun phrases and CTB words (on the morphological level), respectively.

We build a character-based Chinese parsing model to parse the character-level syntax trees. Given an input Chinese sentence, our parser produces its character-level syntax trees (Figure 1(b)). With richer information than word-level trees, this form of parse trees can be useful for all the aforementioned Chinese NLP applications.

With regard to task of parsing itself, an important advantage of the character-level syntax trees is that they allow word segmentation, part-of-speech (POS) tagging and parsing to be performed jointly, using an efficient CKY-style or shift-reduce algorithm. Luo (2003) exploited this advantage by adding flat word structures without manually annotation to CTB trees, and building a generative character-based parser. Compared to a pipeline system, the advantages of a joint system include reduction of error propagation, and the integration of segmentation, POS tagging and syntax features. With hierarchical structures and head character information, our annotated words are more informative than flat word structures, and hence can bring further improvements to phrase-structure parsing.

To analyze word structures in addition to phrase structures, our character-based parser naturally performs joint word segmentation, POS tagging and parsing jointly. Our model is based on the discriminative shift-reduce parser of Zhang and Clark (2009; 2011), which is a state-of-the-art word-based phrase-structure parser for Chinese. We extend their shift-reduce framework, adding more transition actions for word segmentation and POS tagging, and defining novel features that capture character information. Even when trained using character-level syntax trees with flat word structures, our joint parser outperforms a strong pipelined baseline that consists of a state-of-the-art joint segmenter and POS tagger, and our baseline word-based parser. Our word annotations lead to further improvements to the joint system, especially for phrase-structure parsing accuracy.



(a) subject-predicate. (b) verb-object.

(c) coordination. (d) modifier-noun.

Figure 2: Inner word structures of "库存 (repertory)","考古 (archaeology)", "科技 (science and technology)" and "败类 (degenerate)".

Our parser work falls in line with recent work of joint segmentation, POS tagging and parsing (Hatori et al., 2012; Li and Zhou, 2012; Qian and Liu, 2012). Compared with related work, our model gives the best published results for joint segmentation and POS tagging, as well as joint phrase-structure parsing on standard CTB5 evaluations. With linear-time complexity, our parser is highly efficient, processing over 30 sentences per second with a beam size of 16. An open release of the parser is freely available at http://sourceforge.net/projects/zpar/, version 0.6.

2 Word Structures and Syntax Trees

The Chinese language is a character-based language. Unlike alphabetical languages, Chinese characters convey meanings, and the meaning of most Chinese words takes roots in their character. For example, the word "计算机 (computer)" is composed of the characters "计 (count)", "算 (calculate)" and "机 (machine)". An informal name of "computer" is "电脑", which is composed of "电 (electronic)" and "脑 (brain)".

Chinese words have internal structures (Xue, 2001; Ma et al., 2012). The way characters interact within words can be similar to the way words interact within phrases. Figure 2 shows the structures of the four words "库存 (repertory)", " 考古

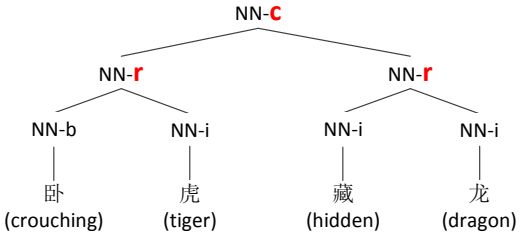


Figure 3: Character-level word structure of "卧虎藏龙 (crouching tiger hidden dragon)".

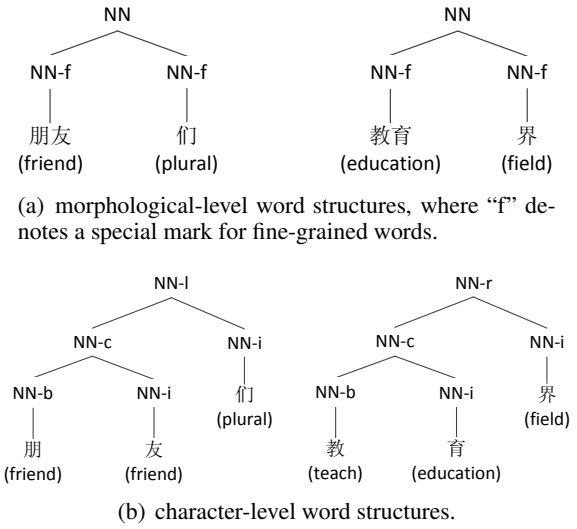
(archaeology)", "科技 (science and technology)" and "败类 (degenerate)", which demonstrate four typical syntactic structures of two-character words, including subject-predicate, verb-object, coordination and modifier-noun structures. Multi-character words can also have recursive syntactic structures. Figure 3 illustrates the structure of the word "卧虎藏龙 (crouching tiger hidden dragon)", which is composed of two subwords "卧虎 (crouching tiger)" and "藏龙 (hidden dragon)", both having a modifier-noun structure.

The meaning of characters can be a useful source of information for computational processing of Chinese, and some recent work has started to exploit this information. Zhang and Clark (2010) found that the first character in a Chinese word is a useful indicator of the word's POS. They made use of this information to help joint word segmentation and POS tagging.

Li (2011) studied the morphological structures of Chinese words, showing that 35% percent of the words in CTB5 can be treated as having morphemes. Figure 4(a) illustrates the morphological structures of the words " 朋友们 (friends)" and "教育界 (educational world)", in which the characters "们 (plural)" and "界 (field)" can be treated as suffix morphemes. They studied the influence of such morphology to Chinese dependency parsing (Li and Zhou, 2012).

The aforementioned work explores the influence of particular types of characters to Chinese processing, yet not the full potentials of complete word structures. We take one step further in this line of work, annotating the full syntactic structures of 37,382 Chinese words in the form of Figure 2 and Figure 3. Our annotation covers the entire vocabulary of CTB5. In addition to difference in coverage (100% vs 35%), our annotation is structurally more informative than that of Li (2011), as illustrated in Figure 4(b).

Our annotations are binarized recursive word



(a) morphological-level word structures, where "f" denotes a special mark for fine-grained words.

(b) character-level word structures.

Figure 4: Comparison between character-level and morphological-level word structures.

structures. For each word or subword, we specify its POS and head direction. We use "l", "r" and "c" to indicate the "left", "right" and "coordination" head directions, respectively. The "coordination" direction is mostly used in coordination structures, while a very small number of transliteration words, such as "奥巴马 (Obama)" and "洛杉矶 (Los Angeles)", have flat structures, and we use "coordination" for their left binarization. For leaf characters, we follow previous work on word segmentation (Xue, 2003; Ng and Low, 2004), and use "b" and "i" to indicate the beginning and non-beginning characters of a word, respectively.

The vast majority of words do not have structural ambiguities. However, the structures of some words may vary according to different POS. For example, "制服" means "dominate" when it is tagged as a verb, of which the head is the left character; the same word means "uniform dress" when tagged as a noun, of which the head is the right character. Thus the input of the word structure annotation is a word together with its POS. The annotation work was conducted by three persons, with one person annotating the entire corpus, and the other two checking the annotations.

Using our annotations, we can extend CTB-style syntax trees (Figure 1(a)) into *character-level* trees (Figure 1(b)). In particular, we mark the original nodes that represent POS tags in CTB-style trees with "-t", and insert our word structures as unary subnodes of the "-t" nodes. For the rest of the paper, we refer to the "-t" nodes as *full-word nodes*, all nodes above full-word nodes as *phrase*

nodes, and all nodes below full-word nodes as *subword nodes*.

Our character-level trees contain additional syntactic information, which are potentially useful to Chinese processing. For example, the head characters of words can be populated up to phrase-level nodes, and serve as an additional source of information that is less sparse than head words. In this paper, we build a parser that yields character-level trees from raw character sequences. In addition, we use this parser to study the effects of our annotations to character-based statistical Chinese parsing, showing that they are useful in improving parsing accuracies.

3 Character-based Chinese Parsing

To produce character-level trees for Chinese NLP tasks, we develop a character-based parsing model, which can jointly perform word segmentation, POS tagging and *phrase-structure* parsing. To our knowledge, this is the first work to develop a transition-based system that jointly performs the above three tasks. Trained using annotated word structures, our parser also analyzes the internal structures of Chinese words.

Our character-based Chinese parsing model is based on the work of Zhang and Clark (2009), which is a transition-based model for lexicalized constituent parsing. They use a beam-search decoder so that the transition action sequence can be globally optimized. The averaged perceptron with early-update (Collins and Roark, 2004) is used to train the model parameters. Their transition system contains four kinds of actions: (1) `SHIFT`, (2) `REDUCE-UNARY`, (3) `REDUCE-BINARY` and (4) `TERMINATE`. The system can provide binarzied CFG trees in Chomsky Norm Form, and they present a reversible conversion procedure to map arbitrary CFG trees into binarized trees.

In this work, we remain consistent with their work, using the head-finding rules of Zhang and Clark (2008), and the same binarization algorithm.[1] We apply the same beam-search algorithm for decoding, and employ the averaged perceptron with early-update to train our model.

We make two extensions to their work to enable joint segmentation, POS tagging and phrase-structure parsing from the character level. First, we modify the actions of the transition system for parsing the inner structures of words. Second, we extend the feature set for our parsing problem.

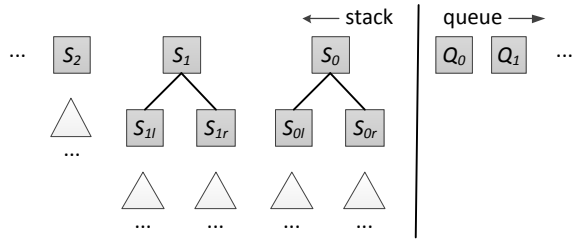


Figure 5: A state in a transition-based model.

3.1 The Transition System

In a transition-based system, an input sentence is processed in a linear left-to-right pass, and the output is constructed by a state-transition process. We learn a model for scoring the transition A_i from one state ST_i to the next ST_{i+1}. As shown in Figure 5, a state ST consists of a stack S and a queue Q, where $S = (\cdots, S_1, S_0)$ contains partially constructed parse trees, and $Q = (Q_0, Q_1, \cdots, Q_{n-j}) = (c_j, c_{j+1}, \cdots, c_n)$ is the sequence of input characters that have not been processed. The candidate transition action A at each step is defined as follows:

- `SHIFT-SEPARATE`(t): remove the head character c_j from Q, pushing a subword node $\frac{S'}{c_j}$[2] onto S, assigning $S'.t = t$. Note that the parse tree S_0 must correspond to a full-word or a phrase node, and the character c_j is the first character of the next word. The argument t denotes the POS of S'.

- `SHIFT-APPEND`: remove the head character c_j from Q, pushing a subword node $\frac{S'}{c_j}$ onto S. c_j will eventually be combined with all the subword nodes on top of S to form a word, and thus we must have $S'.t = S_0.t$.

- `REDUCE-SUBWORD`(d): pop the top two nodes S_0 and S_1 off S, pushing a new subword node $\frac{S'}{S_1\ S_0}$ onto S. The argument d denotes the head direction of S', of which the value can be "left", "right" or "coordination".[3] Both S_0 and S_1 must be subword nodes and $S'.t = S_0.t = S_1.t$.

[1] We use a left-binarization process for flat word structures that contain more than two characters.

[2] We use this notation for a compact representation of a tree node, where the numerator represents a father node, and the denominator represents the children.

[3] For the head direction "coordination", we extract the head character from the left node.

Category	Feature templates	When to Apply
Structure features	S_0ntl S_0nwl S_1ntl S_1nwl S_2ntl S_2nwl S_3ntl S_3nwl, Q_0c Q_1c Q_2c Q_3c $Q_0c \cdot Q_1c$ $Q_1c \cdot Q_2c$ $Q_2c \cdot Q_3c$, $S_{0l}twl$ $S_{0r}twl$ $S_{0u}twl$ $S_{1l}twl$ $S_{1r}twl$ $S_{1u}twl$, $S_0nw \cdot S_1nw$ $S_0nw \cdot S_1nl$ $S_0nl \cdot S_1nw$ $S_0nl \cdot S_1nl$, $S_0nw \cdot Q_0c$ $S_0nl \cdot Q_0c$ $S_1nw \cdot Q_0c$ S_1nlQ_0c, $S_0nl \cdot S_1nl \cdot S_2nl$ $S_0nw \cdot S_1nl \cdot S_2nl$ $S_0nl \cdot S_1nw \cdot S_2nl$ $S_0nl \cdot S_1nl \cdot S_2nw$, $S_0nw \cdot S_1nl \cdot Q_0c$ $S_0nl \cdot S_1nw \cdot Q_0c$ $S_0nl \cdot S_1nl \cdot Q_0c$, $\boldsymbol{S_0ncl}$ $\boldsymbol{S_0nct}$ $\boldsymbol{S_0nctl}$ $\boldsymbol{S_1ncl}$ $\boldsymbol{S_1nct}$ $\boldsymbol{S_1nctl}$, $\boldsymbol{S_2ncl}$ $\boldsymbol{S_2nct}$ $\boldsymbol{S_2nctl}$ $\boldsymbol{S_3ncl}$ $\boldsymbol{S_3nct}$ $\boldsymbol{S_3nctl}$, $\boldsymbol{S_0nc \cdot S_1nc}$ $\boldsymbol{S_0ncl \cdot S_1nl}$ $\boldsymbol{S_0nl \cdot S_1ncl}$ $\boldsymbol{S_0ncl \cdot S_1ncl}$, $\boldsymbol{S_0nc \cdot Q_0c}$ $\boldsymbol{S_0nl \cdot Q_0c}$ $\boldsymbol{S_1nc \cdot Q_0c}$ $\boldsymbol{S_1nl \cdot Q_0c}$, $\boldsymbol{S_0nc \cdot S_1nc \cdot Q_0c}$ $\boldsymbol{S_0nc \cdot S_1nc \cdot Q_0c \cdot Q_1c}$	All
	start(S_0w) $\cdot$ start(S_1w) start(S_0w) $\cdot$ end(S_1w), indict(S_1wS_0w) $\cdot$ len(S_1wS_0w) indict(S_1wS_0w, S_0t) $\cdot$ len(S_1wS_0w)	REDUCE-SUBWORD
String features	$t_{-1} \cdot t_0$ $t_{-2} \cdot t_{-1}t_0$ $w_{-1} \cdot t_0$ $c_0 \cdot t_0$ start(w_{-1}) $\cdot$ t_0 $c_{-1} \cdot c_0 \cdot t_{-1} \cdot t_0$,	SHIFT-SEPARATE
	w_{-1} $w_{-2} \cdot w_{-1}$ w_{-1}, where len(w_{-1}) $= 1$ end(w_{-1}) $\cdot$ c_0, start(w_{-1}) $\cdot$ len(w_{-1}) end(w_{-1}) $\cdot$ len(w_{-1}) start(w_{-1}) $\cdot$ end(w_{-1}), $w_{-1} \cdot c_0$ end(w_{-2}) $\cdot$ w_{-1} start(w_{-1}) $\cdot$ c_0 end(w_{-2}) $\cdot$ end(w_{-1}), $w_{-1} \cdot$ len(w_{-2}) $w_{-2} \cdot$ len(w_{-1}) $w_{-1} \cdot t_{-1}$ $w_{-1} \cdot t_{-2}$ $w_{-1} \cdot t_{-1} \cdot c_0$, $w_{-1} \cdot t_{-1} \cdot$ end(w_{-2}) $c_{-2} \cdot c_{-1} \cdot c_0 \cdot t_{-1}$, where len($w_{-1}$) $= 1$ end(w_{-1}) $\cdot$ t_{-1}, $c \cdot t_{-1} \cdot$ end(w_{-1}), where $c \in w_{-1}$ *and* $c \neq$ end(w_{-1})	REDUCE-WORD
	$c_0 \cdot t_{-1}$ $c_{-1} \cdot c_0$ start(w_{-1}) $\cdot$ c_0t_{-1} $c_{-1} \cdot c_0 \cdot t_{-1}$	SHIFT-APPEND

Table 1: Feature templates for the character-level parser. The function start($\cdot$), end($\cdot$) and len($\cdot$) denote the first character, the last character and the length of a word, respectively.

- REDUCE-WORD: pop the top node S_0 off S, pushing a full-word node $\frac{S'}{S_0}$ onto S. This reduce action generates a full-word node from S_0, which must be a subword node.
- REDUCE-BINARY(d, l): pop the top two nodes S_0 and S_1 off S, pushing a binary phrase node $\frac{S'}{S_1\ S_0}$ onto S. The argument l denotes the constituent label of S', and the argument d specifies the lexical head direction of S', which can be either "left" or "right". Both S_0 and S_1 must be a full-word node or a phrase node.
- REDUCE-UNARY(l): pop the top node S_0 off S, pushing a unary phrase node $\frac{S'}{S_0}$ onto S. l denotes the constituent label of S'.
- TERMINATE: mark parsing complete.

Compared to set of actions in our baseline transition-based phrase-structure parser, we have made three major changes. First, we split the original SHIFT action into SHIFT-SEPARATE(t) and SHIFT-APPEND, which jointly perform the word segmentation and POS tagging tasks. Second, we add an extra REDUCE-SUBWORD(d) operation, which is used for parsing the inner structures of words. Third, we add REDUCE-WORD, which applies a unary rule to mark a completed subword node as a full-word node. The new node corresponds to a unary "-t" node in Figure 1(b).

3.2 Features

Table 1 shows the feature templates of our model. The feature set consists of two categories: (1) structure features, which encode the structural information of subwords, full-words and phrases. (2) string features, which encode the information of neighboring characters and words.

For the structure features, the symbols S_0, S_1, S_2, S_3 represent the top four nodes on the stack; Q_0, Q_1, Q_2, Q_3 denote the first four characters in the queue; S_{0l}, S_{0r}, S_{0u} represent the left, right child for a binary branching S_0, and the single child for a unary branching S_0, respectively; S_{1l}, S_{1r}, S_{1u} represent the left, right child for a binary branching S_1, and the single child for a unary branching S_1, respectively; n represents the type for a node; it is a binary value that indicates whether the node is a subword node; c, w, t and l represent the head character, word (or subword), POS tag and constituent label of a node, respectively. The structure features are mostly taken

from the work of Zhang and Clark (2009). The feature templates in bold are novel, are designed to encode head character information. In particular, the **indict** function denotes whether a word is in a tag dictionary, which is collected by extracting all multi-character subwords that occur more than five times in the training corpus.

For string features, c_0, c_{-1} and c_{-2} represent the current character and its previous two characters, respectively; w_{-1} and w_{-2} represent the previous two words to the current character, respectively; t_0, t_{-1} and t_{-2} represent the POS tags of the current word and the previous two words, respectively. The string features are used for word segmentation and POS tagging, and are adapted from a state-of-the-art joint segmentation and tagging model (Zhang and Clark, 2010).

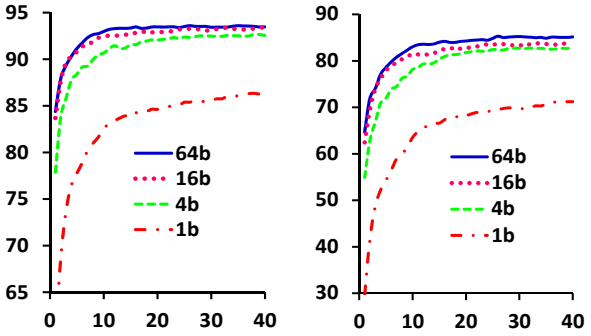
In summary, our character-based parser contains the word-based features of constituent parser presented in Zhang and Clark (2009), the word-based and shallow character-based features of joint word segmentation and POS tagging presented in Zhang and Clark (2010), and additionally the deep character-based features that encode word structure information, which are the first presented by this paper.

4 Experiments

4.1 Setting

We conduct our experiments on the CTB5 corpus, using the standard split of data, with sections 1–270,400–931 and 1001–1151 for training, sections 301–325 for system development, and sections 271–300 for testing. We apply the same preprocessing step as Harper and Huang (2011), so that the non-terminal yield unary chains are collapsed to single unary rules.

Since our model can jointly process word segmentation, POS tagging and phrase-structure parsing, we evaluate our model for the three tasks, respectively. For word segmentation and POS tagging, standard metrics of word precision, recall and F-score are used, where the tagging accuracy is the joint accuracy of word segmentation and POS tagging. For phrase-structure parsing, we use the standard `parseval` evaluation metrics on bracketing precision, recall and F-score. As our constituent trees are based on characters, we follow previous work and redefine the boundary of a constituent span by its start and end characters. In addition, we evaluate the performance of word structures, using the word precision, recall and F-score metrics. A word structure is correct only if the word and its internal structure are both correct.



(a) Joint segmentation and POS tagging F-scores.

(b) Joint constituent parsing F-scores.

Figure 6: Accuracies against the training epoch for joint segmentation and tagging as well as joint phrase-structure parsing using beam sizes 1, 4, 16 and 64, respectively.

4.2 Development Results

Figure 6 shows the accuracies of our model using different beam sizes with respect to the training epoch. The performance of our model increases as the beam size increases. The amount of increases becomes smaller as the size of the beam grows larger. Tested using gcc 4.7.2 and Fedora 17 on an Intel Core i5-3470 CPU (3.20GHz), the decoding speeds are 318.2, 98.0, 30.3 and 7.9 sentences per second with beam size 1, 4, 16 and 64, respectively. Based on this experiment, we set the beam size 64 for the rest of our experiments.

The character-level parsing model has the advantage that deep character information can be extracted as features for parsing. For example, the head character of a word is exploited in our model. We conduct feature ablation experiments to evaluate the effectiveness of these features. We find that the parsing accuracy decreases about 0.6% when the head character related features (the bold feature templates in Table 1) are removed, which demonstrates the usefulness of these features.

4.3 Final Results

In this section, we present the final results of our model, and compare it to two baseline systems, a pipelined system and a joint system that is trained with automatically generated flat words structures.

The baseline pipelined system consists of the joint segmentation and tagging model proposed by

	Task	P	R	F
Pipeline	Seg	97.35	98.02	97.69
	Tag	93.51	94.15	93.83
	Parse	81.58	82.95	82.26
Flat word structures	Seg	97.32	98.13	97.73
	Tag	94.09	94.88	94.48
	Parse	83.39	83.84	83.61
Annotated word structures	Seg	97.49	98.18	**97.84**
	Tag	94.46	95.14	**94.80**
	Parse	84.42	84.43	**84.43**
	WS	94.02	94.69	94.35

Table 2: Final results on test corpus.

Zhang and Clark (2010), and the phrase-structure parsing model of Zhang and Clark (2009). Both models give state-of-the-art performances, and are freely available.[4] The model for joint segmentation and POS tagging is trained with a 16-beam, since it achieves the best performance. The phrase-structure parsing model is trained with a 64-beam. We train the parsing model using the automatically generated POS tags by 10-way jackknifing, which gives about 1.5% increases in parsing accuracy when tested on automatic segmented and POS tagged inputs.

The joint system trained with flat word structures serves to test the effectiveness of our joint parsing system over the pipelined baseline, since flat word structures do not contain additional sources of information over the baseline. It is also used to test the usefulness of our annotation in improving parsing accuracy.

Table 2 shows the final results of our model and the two baseline systems on the test data. We can see that both character-level joint models outperform the pipelined system; our model with annotated word structures gives an improvement of 0.97% in tagging accuracy and 2.17% in phrase-structure parsing accuracy. The results also demonstrate that the annotated word structures are highly effective for syntactic parsing, giving an absolute improvement of 0.82% in phrase-structure parsing accuracy over the joint model with flat word structures.

Row "WS" in Table 2 shows the accuracy of hierarchical word-structure recovery of our joint system. This figure can be useful for high-level applications that make use of character-level trees by our parser, as it reflects the capability of our parser in analyzing word structures. In particular, the performance of parsing OOV word structure is an important metric of our parser. The recall of OOV word structures is 60.43%, while if we do not consider the influences of segmentation and tagging errors, counting only the correctly segmented and tagged words, the recall is 87.96%.

[4]http://sourceforge.net/projects/zpar/, version 0.5.

4.4 Comparison with Previous Work

In this section, we compare our model to previous systems on the performance of joint word segmentation and POS tagging, and the performance of joint phrase-structure parsing.

Table 3 shows the results. Kruengkrai+ '09 denotes the results of Kruengkrai et al. (2009), which is a lattice-based joint word segmentation and POS tagging model; Sun '11 denotes a subword based stacking model for joint segmentation and POS tagging (Sun, 2011), which uses a dictionary of idioms; Wang+ '11 denotes a semi-supervised model proposed by Wang et al. (2011), which additionally uses the Chinese Gigaword Corpus; Li '11 denotes a generative model that can perform word segmentation, POS tagging and phrase-structure parsing jointly (Li, 2011); Li+ '12 denotes a unified dependency parsing model that can perform joint word segmentation, POS tagging and dependency parsing (Li and Zhou, 2012); Li '11 and Li+ '12 exploited annotated morphological-level word structures for Chinese; Hatori+ '12 denotes an incremental joint model for word segmentation, POS tagging and dependency parsing (Hatori et al., 2012); they use external dictionary resources including HowNet Word List and page names from the Chinese Wikipedia; Qian+ '12 denotes a joint segmentation, POS tagging and parsing system using a unified framework for decoding, incorporating a word segmentation model, a POS tagging model and a phrase-structure parsing model together (Qian and Liu, 2012); their word segmentation model is a combination of character-based model and word-based model. Our model achieved the best performance on both joint segmentation and tagging as well as joint phrase-structure parsing.

Our final performance on constituent parsing is by far the best that we are aware of for the Chinese data, and even better than some state-of-the-art models with gold segmentation. For example, the un-lexicalized PCFG model of Petrov and Klein

System	Seg	Tag	Parse
Kruengkrai+ '09	97.87	93.67	–
Sun '11	98.17*	94.02*	–
Wang+ '11	98.11*	94.18*	–
Li '11	97.3	93.5	79.7
Li+ '12	97.50	93.31	–
Hatori+ '12	**98.26***	94.64*	–
Qian+ '12	97.96	93.81	82.85
Ours pipeline	97.69	93.83	82.26
Ours joint flat	97.73	94.48	83.61
Ours joint annotated	97.84	**94.80**	**84.43**

Table 3: Comparisons of our final model with state-of-the-art systems, where "*" denotes that external dictionary or corpus has been used.

(2007) achieves 83.45%[5] in parsing accuracy on the test corpus, and our pipeline constituent parsing model achieves 83.55% with gold segmentation. They are lower than the performance of our character-level model, which is 84.43% without gold segmentation. The main differences between word-based and character-level parsing models are that character-level model can exploit character features. This further demonstrates the effectiveness of characters in Chinese parsing.

5 Related Work

Recent work on using the internal structure of words to help Chinese processing gives important motivations to our work. Zhao (2009) studied character-level dependencies for Chinese word segmentation by formalizing segmentsion task in a dependency parsing framework. Their results demonstrate that annotated word dependencies can be helpful for word segmentation. Li (2011) pointed out that the word's internal structure is very important for Chinese NLP. They annotated morphological-level word structures, and a unified generative model was proposed to parse the Chinese morphological and phrase-structures. Li and Zhou (2012) also exploited the morphological-level word structures for Chinese dependency parsing. They proposed a unified transition-based model to parse the morphological and dependency structures of a Chinese sentence in a unified framework. The morphological-level word structures concern only prefixes and suffixes, which cover only 35% of entire words in CTB. According to their results, the final performances of their model on word segmentation and POS tagging are below the state-of-the-art joint segmentation and POS tagging models. Compared to their work, we consider the character-level word structures for Chinese parsing, presenting a unified framework for segmentation, POS tagging and phrase-structure parsing. We can achieve improved segmentation and tagging performance.

Our character-level parsing model is inspired by the work of Zhang and Clark (2009), which is a transition-based model with a beam-search decoder for word-based constituent parsing. Our work is based on the shift-reduce operations of their work, while we introduce additional operations for segmentation and POS tagging. By such an extension, our model can include all the features in their work, together with the features for segmentation and POS tagging. In addition, we propose novel features related to word structures and interactions between word segmentation, POS tagging and word-based constituent parsing.

Luo (2003) was the first work to introduce the character-based syntax parsing. They use it as a joint framework to perform Chinese word segmentation, POS tagging and syntax parsing. They exploit a generative maximum entropy model for character-based constituent parsing, and find that POS information is very useful for Chinese word segmentation, but high-level syntactic information seems to have little effect on segmentation. Compared to their work, we use a transition-based discriminative model, which can benefit from large amounts of flexible features. In addition, instead of using flat structures, we manually annotate hierarchal tree structures of Chinese words for converting word-based constituent trees into character-based constituent trees.

Hatori et al. (2012) proposed the first joint work for the word segmentation, POS tagging and dependency parsing. They used a single transition-based model to perform the three tasks. Their work demonstrates that a joint model can improve the performance of the three tasks, particularly for POS tagging and dependency parsing. Qian and Liu (2012) proposed a joint decoder for word segmentation, POS tagging and word-based constituent parsing, although they trained models for the three tasks separately. They reported better

[5]We rerun the parser and evaluate it using the publicly-available code on http://code.google.com/p/berkeleyparser by ourselves, since we have a preprocessing step for the CTB5 corpus.

performances when using a joint decoder. In our work, we employ a single character-based discriminative model to perform segmentation, POS tagging and phrase-structure parsing jointly, and study the influence of annotated word structures.

6 Conclusions and Future Work

We studied the internal structures of more than 37,382 Chinese words, analyzing their structures as the recursive combinations of characters. Using these word structures, we extended the CTB into character-level trees, and developed a character-based parser that builds such trees from raw character sequences. Our character-based parser performs segmentation, POS tagging and parsing simultaneously, and significantly outperforms a pipelined baseline. We make both our annotations and our parser available online.

In summary, our contributions include:

- We annotated the internal structures of Chinese words, which are potentially useful to character-based studies of Chinese NLP. We extend CTB-style constituent trees into character-level trees using our annotations.
- We developed a character-based parsing model that can produce our character-level constituent trees. Our parser jointly performs word segmentation, POS tagging and syntactic parsing.
- We investigated the effectiveness of our joint parser over pipelined baseline, and the effectiveness of our annotated word structures in improving parsing accuracies.

Future work includes investigations of our parser and annotations on Chinese NLP tasks.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 61133012, the National "863" Major Projects via grant 2011AA01A207, the National "863" Leading Technology Research Project via grant 2012AA011102, and SRG ISTD 2012 038 from Singapore University of Technology and Design.

References

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 111–118, Barcelona, Spain, July.

Mary Harper and Zhongqiang Huang. 2011. Chinese statistical parsing. *Handbook of Natural Language Processing and Machine Translation*.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1045–1053, Jeju Island, Korea, July. Association for Computational Linguistics.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 513–521, Suntec, Singapore, August. Association for Computational Linguistics.

Zhongguo Li and Guodong Zhou. 2012. Unified dependency parsing of chinese morphological and syntactic structures. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1445–1454, Jeju Island, Korea, July. Association for Computational Linguistics.

Zhongguo Li. 2011. Parsing the internal structure of words: A new paradigm for chinese word segmentation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1414, Portland, Oregon, USA, June. Association for Computational Linguistics.

Xiaoqiang Luo. 2003. A maximum entropy Chinese character-based parser. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 192–199.

Jianqiang Ma, Chunyu Kit, and Dale Gerdemann. 2012. Semi-automatic annotation of chinese word structure. In *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 9–17, Tianjin, China, December. Association for Computational Linguistics.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 277–284, Barcelona, Spain, July. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language*

Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Xian Qian and Yang Liu. 2012. Joint chinese word segmentation, pos tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 501–511, Jeju Island, Korea, July. Association for Computational Linguistics.

Weiwei Sun. 2011. A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1385–1394, Portland, Oregon, USA, June. Association for Computational Linguistics.

David Vadas and James Curran. 2007. Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague, Czech Republic, June. Association for Computational Linguistics.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 309–317, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

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.

Nianwen Xue. 2001. *Defining and Automatically Identifying Words in Chinese*. Ph.D. thesis, University of Delaware.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1).

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu, Hawaii, October. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 162–171, Paris, France, October. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 843–852, Cambridge, MA, October. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.

Hai Zhao. 2009. Character-level dependencies in chinese: Usefulness and learning. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 879–887, Athens, Greece, March. Association for Computational Linguistics.