Unified Dependency Parsing of Chinese Morphological and Syntactic Structures

Zhongguo Li **Guodong Zhou**
Natural Language Processing Laboratory
School of Computer Science and Technology
Soochow University, Suzhou, Jiangsu Province 215006, China
{lzg, gdzhou}@suda.edu.cn

Abstract

Most previous approaches to syntactic parsing of Chinese rely on a preprocessing step of word segmentation, thereby assuming there was a clearly defined boundary between morphology and syntax in Chinese. We show how this assumption can fail badly, leading to many out-of-vocabulary words and incompatible annotations. Hence in practice the strict separation of morphology and syntax in the Chinese language proves to be untenable. We present a unified dependency parsing approach for Chinese which takes unsegmented sentences as input and outputs both morphological and syntactic structures with a single model and algorithm. By removing the intermediate word segmentation, the unified parser no longer needs separate notions for words and phrases. Evaluation proves the effectiveness of the unified model and algorithm in parsing structures of words, phrases and sentences simultaneously. [1]

1 Introduction

The formulation of the concept of words has baffled linguists from ancient to modern times (Hockett, 1969). Things are even worse for Chinese, partly due to the fact that its written form does not delimit words explicitly. While we have no doubt that there are linguistic units which are definitely words (or phrases, for that matter), it's a sad truth that in many cases we cannot manage to draw such a clear boundary between morphology and syntax, for which we now give two arguments.

The first argument is that many sub-word linguistic units (such as suffixes and prefixes) are so productive that they can lead to a huge number of out-of-vocabulary words for natural language processing systems. This phenomenon brings us into an awkward situation if we adhere to a rigid separation of morphology and syntax. Consider character 者 'someone' as an example. On the one hand, there is strong evidence that it's not a word as it can never be used alone. On the other hand, taking it as a mere suffix leads to many out-of-vocabulary words because of the productivity of such characters. For instance, Penn Chinese Treebank (CTB6) contains 失败者 'one that fails' as a word but not 成功者 'one that succeeds', even with the word 成功 'succeed' appearing 207 times. We call words like 成功者 'one that succeeds' *pseudo* OOVs. By definition, pseudo OOVs are OOVs since they do not occur in the training corpus, though their components are frequently-seen words. Our estimation is that over 60% of OOVs in Chinese are of this kind (Section 2).

Of course, the way out of this dilemma is to parse the internal structures of these words. That is to say, we can still regard characters like 者 as suffixes, taking into account the fact that they cannot be used alone. Meanwhile, pseudo OOVs can be largely eliminated through analyzing their structures, thus greatly facilitating syntactic and semantic analysis of sentences. In fact, previous studies have revealed other good reasons for parsing internal structures of words (Zhao, 2009; Li, 2011).

The second argument is that in Chinese many linguistic units can form both words and phrases with exactly the same meaning and part-of-speech, which

[1] Corresponding author is Guodong Zhou.

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1445–1454, Jeju Island, Korea, 12–14 July 2012.

Figure 1: Unified parsing of words and phrases.

causes lots of *incompatible annotations* in currently available corpora. Take character 法 'law' as an example. It is head of both 刑法 'criminal law' and 环境保护法 'environmental protection law', but CTB treat it as a suffix in the former (with the annotation being 刑法_NN) and a word in the later (the annotation is 环境_NN 保护_NN 法_NN). These annotations are incompatible since in both cases the character 法 'law' bears exactly the same meaning and usage (e.g. part-of-speech). We examined several widely used corpora and found that about 90% of affixes were annotated incompatibly (Section 2). Incompatibility can be avoided through parsing structures of both words and phrases. Figure 1 conveys this idea. A further benefit of unified parsing is to reduce data sparseness. As an example, in CTB6 器 'machine' appears twice in phrases but 377 times in words (e.g. 加速器 'accelerator'). Word structures in Chinese can be excellent guide for parsing phrase structures, and vice versa, due to their similarity.

The present paper makes two contributions in light of these issues. Firstly, in order to get rid of pseudo OOVs and incompatible annotations, we have annotated structures of words in CTB6, after which statistical models can learn structures of words as well as phrases from the augmented treebank (Section 4). Although previous authors have noticed the importance of word-structure parsing (Li, 2011; Zhao, 2009), no detailed description about annotation of word structures has been provided in the literature. Secondly, we designed a unified dependency parser whose input is unsegmented sentences and its output incorporates both morphological and syntactic structures with a single model and algorithm (Section 5). By removing the intermediate step of word segmentation, our unified parser no longer depends on the unsound notion that there is a clear boundary between words and phrases. Evaluation (Section 6) shows that our unified parser achieves satisfactory accuracies in parsing both morphological and syntactic structures.

corpus	OOV	pseudo	percent
CTB6	158	112	70.9
MSR	1,783	1,307	73.3
PKU	2,860	1,836	64.2
AS	3,020	2,143	71.0
CITYU	1,665	1,100	66.0

Table 1: Statistics of pseudo OOVs for five corpora.

2 Pseudo OOVs and Incompatible Annotations

In this section we show the surprisingly pervasive nature of pseudo OOVs and incompatible annotations through analysis of five segmented corpora, which are CTB6 and corpus by MSR, PKU, AS and CITYU provided in SIGHAN word segmentation Bakeoffs [2].

First we use the standard split of training and testing data and extract all OOVs for each corpus, then count the number of pseudo OOVs. Table 1 gives the result. It's amazing that for every corpus, over 60% of OOVs are pseudo, meaning they can be avoided if their internal structures were parsed. Reduction of OOVs at such a large scale can benefit greatly downstream natural language processing systems.

We then sample 200 word types containing a productive affix from each corpus, and check whether the affix also occurs somewhere else in a phrase, i.e, the affix is annotated as a word in the phrase. The results are in Table 2. It's clear and somewhat shocking that most affixes are annotated incompatibly. We believe it is not the annotators to blame, rather the root cause lies deeply in the unique characteristics of the Chinese language. This becomes obvious in comparison with English, where suffix like '-ism' in 'capitalism' cannot be used alone as a word in phrases. [3] Incompatible annotations can be removed only through unified parsing of word and phrase structures, as mentioned earlier and illustrated in Figure 1.

[2] http://www.sighan.org/bakeoff2005/

[3] Actually English allows examples like "pre- and post-war imperialism" where a prefix like "pre" can appear on its own as long as the hyphen is present and it is in a coordination structure. Note that such examples are much rarer than what we discuss in this paper for Chinese. We thank the reviewer very much for pointing this out and providing this example for us.

corpus	incompatible	percent
CTB6	190	95
MSR	178	89
PKU	192	96
AS	182	91
CITYU	194	97

Table 2: Statistics of incompatibly annotated affixes in 200 sampled words for five segmented corpora.

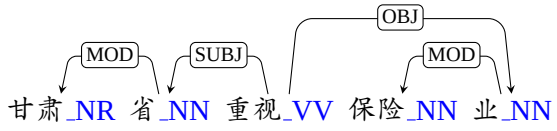

Figure 2: Example output of unified dependency parsing of Chinese morphological and syntactic structures.

3 Unified Parsing Defined

Given an unsegmented sentence 甘肃省重视保险业 'Gansu province attaches great importance to insurance industry', the output of unified dependency parser is shown in Figure 2. As can be seen, this output contains information about word (such as 重视_VV) as well as phrase structures (such as 重视_VV 保险_NN 业_NN), which is what we mean by 'unified' parsing. Now, it's no longer vital to differentiate between morphology and syntax for Chinese. People could regard 保险业 'insurance industry' as a word or phrase, but either way, there will be no disagreements about its internal structure. From the perspective of the unified parser, linguistic units are given the same labels as long as they function similarly (e.g, they have the same parts-of-speech).

As a bonus, output of unified parsing incorporates Chinese word segmentation, part-of-speech tagging and dependency parsing. To achieve these goals, previous systems usually used a pipelined approach by combining several statistical models, which was further complicated by different decoding algorithms for each of these models. The present paper shows that a single model does all these jobs. Besides being much simpler in engineering such a parser, this approach is also a lot more plausible for modeling human language understanding.

4 Annotation of Word Structures

Unified parsing requires a corpus annotated with both morphological and syntactic structures. Such a corpus can be built with the least effort if we begin with an existing treebank such as CTB6 already annotated with syntactic structures. It only remains for us to annotate internal structures of words in this treebank.

4.1 Scope of Annotation

In order to get rid of pseudo OOVs and incompatible annotations, internal structures are annotated for two kinds of words. The first kind contains words with a productive component such as suffix or prefix. One example is 陈述人 'speaker' whose suffix is the very productive 人 'person' (e.g, in CTB6 there are about 400 words having this suffix). The second kind includes words with compositional semantics. Examples are 星期一 'Monday' and 星期天 'Sunday'. Though 星期 'week' is not very productive, the meaning of words with this prefix is deducible from semantics of their components.

Other compound words such as 研究 'research' have no productive components and are not a cause of pseudo OOVs. They are universally considered as words instead of phrases due to their non-compositional semantics. Hence their structures are not annotated in the present research. Meanwhile, for single-morpheme words with no structures whatsoever, like 伊拉克 'Iraq' and 蝙蝠 'bat', annotation of internal structures is of course unnecessary either.

Of all the 54, 214 word types in CTB6, 35% are annotated, while the percentage is 24% for the 782, 901 word tokens. Around 80% of sentences contain words whose structures need annotation. Our annotations will be made publicly available for research purposes.

4.2 From Part-of-speeches to Constituents

Of all 33 part-of-speech tags in CTB, annotation of word structures is needed for nine tags: NN, VV, JJ, CD, NT, NR, AD, VA and OD. Since part-of-speech tags are preterminals and can only have one terminal word as its child, POS tags of words become constituent labels after annotation of word structures. The mapping rules from POS tags to constituent labels are listed in Table 3. Readers should note that

POS tags	constituent label
NR, NN, NT	NP
JJ	ADJP
AD	ADVP
CD, OD	QP
VV, VA	VP

Table 3: Correspondence between POS tags and constituent labels after annotation.

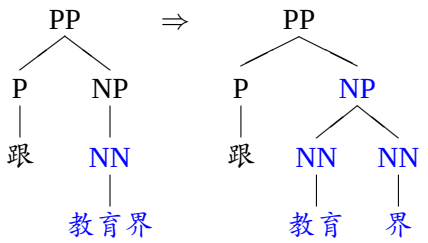

Figure 3: Example annotation for the word 教育界_NN in CTB6: POS tag NN changes to constituent label NP after annotation.

such mapping is not arbitrary. The constraint is that in the treebank the POS tag must somewhere be the unique child of the constituent label. Figure 3 depicts an example annotation, in which we also have an example of NP having a tag NN as its only child.

4.3 Recursive Annotation

Some words in CTB have very complex structures. Examples include 原子核物理学家 'physicist majoring in nuclear physics', 反托拉斯法 'anti-trust laws' etc. Structures of these words are annotated to their full possible depth. Existence of such words are characteristic of the Chinese language, since they are further demonstrations of the blurred boundary between morphology and syntax. A full-fledged parser is needed to analyze structures of these words, which incidentally provides us with another motivation for unified morphological and syntactic parsing of Chinese.

5 Unified Dependency Parsing

All previous dependency parsers for Chinese take it for granted that the input sentence is already segmented into words (Li et al., 2011). Most systems even require words to be tagged with their part-of-speeches (Zhang and Nivre, 2011). Hence current off-the-shelf algorithms are inadequate for parsing unsegmented sentences. Instead, a new unified parsing algorithm is given in this section.

5.1 Transitions

To map a raw sentence directly to output shown in Figure 2, we define four transitions for the unified dependency parser. They act on a stack containing the incremental parsing results, and a queue holding the incoming Chinese characters of the sentence:

SHIFT: the first character in the queue is shifted into the stack as the start of a new word. The queue should not be empty.

LEFT: the top two words of the stack are connected with an arc, with the top one being the head. There should be at least two elements on the stack.

RIGHT: the top two words of the stack are connected, but with the top word being the child. The precondition is the same as that of LEFT.

APPEND: the first character in the queue is appended to the word at the top of the stack. There are two preconditions. First, the queue should not be empty. Second, the top of the stack must be a word with no arcs connected to other words (i.e, up to now it has got neither children nor parent).

We see that these transitions mimic the general arc-standard dependency parsing models. The first three of them were used, for example, by Yamada and Matsumoto (2003) to parse English sentences. The only novel addition is APPEND, which is necessary because we are dealing with raw sentences. Its sole purpose is to assemble characters into words with no internal structures, such as 西雅图 'Seattle'. Thus this transition is the key for removing the need of Chinese word segmentation and parsing unsegmented sentences directly.

To also output part-of-speech tags and dependency labels, the transitions above can be augmented accordingly. Hence we can change SHIFT to SHIFT·X where X represents a certain POS tag. Also, LEFT and RIGHT should be augmented with appropriate dependency relations, such as LEFT·SUBJ for a dependency between verb and subject.

As a demonstration of the usage of these transitions, consider sentence 我喜欢西雅图 "I love Seattle". Table 4 lists all steps of the parsing process. Readers interested in implementing their own

step	stack	queue	action
1		我喜欢西雅图	SHIFT·PN
2	我_PN	喜欢西雅图	SHIFT·VV
3	我_PN 喜_VV	欢西雅图	APPEND
4	我_PN 喜欢_VV	西雅图	LEFT·SUBJ
5	我_PN $\xleftarrow{\text{SUBJ}}$ 喜欢_VV	西雅图	SHIFT·NR
6	我_PN $\xleftarrow{\text{SUBJ}}$ 喜欢_VV 西_NR	雅图	APPEND
7	我_PN $\xleftarrow{\text{SUBJ}}$ 喜欢_VV 西雅_NR	图	APPEND
8	我_PN $\xleftarrow{\text{SUBJ}}$ 喜欢_VV 西雅图_NR		RIGHT·OBJ
9	我_PN $\xleftarrow{\text{SUBJ}}$ 喜欢_VV $\xrightarrow{\text{OBJ}}$ 西雅图_NR		STOP

Table 4: Parsing process of a short sentence with the four transitions defined above.

unified dependency parsers are invited to study this example carefully.

5.2 Model

Due to structural ambiguity, there might be quite a lot of possibilities for parsing a given raw sentence. Hence at each step in the parsing process, all four transitions defined above may be applicable. To resolve ambiguities, each candidate parse is scored with a global linear model defined as follows.

For an input sentence x, the parsing result $F(x)$ is the one with the highest score in all possible structures for this x:

$$F(x) = \arg\max_{y \in \mathrm{GEN}(x)} \mathrm{Score}(y) \quad (1)$$

Here $\mathrm{GEN}(x)$ is a set of all possible parses for sentence x, and $\mathrm{Score}(y)$ is a real-valued linear function:

$$\mathrm{Score}(y) = \Phi(y) \cdot \vec{w} \quad (2)$$

where $\Phi(y)$ is a global feature vector extracted from parsing result y, and $\vec{w}$ is a vector of weighting parameters. Because of its linearity, $\mathrm{Score}(y)$ can be computed incrementally, following the transition of each parsing step. Parameter vector $\vec{w}$ is trained with the generalized perceptron algorithm of Collins (2002). The early-update strategy of Collins and Roark (2004) is used so as to improve accuracy and speed up the training.

5.3 Feature Templates

For a particular parse y, we now describe the way of computing its feature vector $\Phi(y)$ in the linear model of Equation (2). If S denotes the stack holding the partial results, and Q the queue storing the incoming Chinese characters of a raw sentence, then transition-based parsing features are extracted from S and Q according to those feature templates in Table 5.

	Description	Feature Templates
1	top of S	S0wt; S0w; S0t
2	next top of S	S1wt; S1w; S1t
3	S0 and S1	S1wtS0wt; S1wtS0w S1wS0wt; S1wtS0t S1tS0wt; S1wS0w; S1tS0t
4	char unigrams	Q0; Q1; Q2; Q3
5	char bigrams	Q0Q1; Q1Q2; Q2Q3
6	char trigrams	Q0Q1Q2; Q1Q2Q3
7	ST+unigrams	STwtQ0; STwQ0; STtQ0
8	ST+bigrams	STwtQ0Q1; STwQ0Q1 STtQ0Q1
9	ST+trigrams	STwtQ0Q1Q2 STwQ0Q1Q2; STtQ0Q1Q2
10	parent P of ST	PtSTtQ0; PtSTtQ0Q1 PtSTtQ0Q1Q2
11	leftmost child LC and rightmost child RC	STtLCtQ0; STtLCtQ0Q1 STtLCtQ0Q1Q2 STtRCtQ0; STtRCtQ0Q1 STtRCtQ0Q1Q2

Table 5: Transition-based feature templates. Q0 is the first character in Q, etc. w = word, t = POS tag.

Although we employ transition-based parsing, nothing prevents us from using graph-based features. As shown by Zhang and Clark (2011), depen-

	Description	Feature Templates
1	parent word	Pwt; Pw; Pt
2	child word	Cwt; Cw; Ct
3	P and C	PwtCwt; PwtCw; PwCwt PtCwt; PwCw; PtCt PwtCt
4	neighbor word of P and C left (L) or right (R)	PtPLtCtCLt; PtPLtCtCRt PtPRtCtCLt; PtPRtCtCRt PtPLtCLt; PtPLtCRt PtPRtCLt; PtPRtCRt PLtCtCLt; PLtCtCRt PRtCtCLt; PRtCtCRt PtCtCLt; PtCtCRt PtPLtCt; PtPRtCt
5	sibling(S) of C	CwSw;CtSt; CwSt CtSw; PtCtSt
6	leftmost and rightmost child	PtCtCLCt PtCtCRCt
7	left (la) and right (ra) arity of P	Ptla; Ptra Pwtla; Pwtra Pwla; Pwra

Table 6: Graph-based feature templates for the unified parser. Most of these templates are adapted from those used by Zhang and Clark (2011). w = word; t = POS tag.

dency parsers using both transition-based and graph-based features tend to achieve higher accuracy than parsers which only make use of one kind of features. Table 6 gives the graph-based feature templates used in our parser. All such templates are instantiated at the earliest possible time, in order to reduce as much as possible situations where correct parses fall out of the beam during decoding.

5.4 Decoding Algorithm

We use beam-search to find the best parse for a given raw sentence (Algorithm 1). This algorithm uses double beams. The first beam contains unfinished parsing results, while the second holds completed parses. Double beams are necessary because the number of transitions might well be different for different parses, and those parses that finished earlier are not necessarily better parses. During the searching process, correct parse could fall off the beams, resulting in a search error. However, in practice beam search decoding algorithm works quite well. In addition, it's not feasible to use dynamic programming because of the complicated features used in the model.

The B in Algorithm 1 is the width of the two beams. In our experiments we set B to 64. This value of B was determined empirically by using the standard development set of the data, with the goal of achieving the highest possible accuracy within reasonable time. Note that in line 20 of the algorithm, the beam for completed parsers are pruned at each iteration of the parsing process. The purpose of this action is to keep this beam from growing too big, resulting in a waste of memory space.

Algorithm 1 Beam Search Decoding

```
1:  candidates ← {STARTITEM()}
2:  agenda ← φ
3:  completed ← φ
4:  loop
5:    for all candidate in candidates do
6:      for all legal action of candidate do
7:        newc ← EXPAND(candidate, action)
8:        if COMPLETED(newc) then
9:          completed.INSERT(newc)
10:       else
11:         agenda.INSERT(newc)
12:       end if
13:     end for
14:   end for
15:   if EMPTY(agenda) then
16:     return TOP(completed)
17:   end if
18:   candidates ← TOPB(agenda, B)
19:   agenda ← φ
20:   completed ← TOPB(completed, B)
21: end loop
```

6 Experiments and Evaluation

We describe the experiments carried out and our method of evaluation of the unified dependency parser. We used Penn2Malt [4] to convert constituent trees of CTB to dependency relations. The head rules for this conversion was given by Zhang and Clark (2008). In all experiments, we followed the stan-

[4] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

	P	R	F
our method, labeled	78.54	80.93	79.72
our method, unlabeled	81.01	83.77	82.37
ZC2011, unlabeled	N/A	N/A	75.09

Table 7: Evaluation results on the original CTB5. N/A means the value is not available to us. ZC2011 is Zhang and Clark (2011).

dard split of the data into training, testing and development data (Zhang and Clark, 2011). Though we annotated structures of words in CTB6, most previously results were on CTB5, a subset of the former treebank. Hence we report our results of evaluation on CTB5 for better comparability.

6.1 Dependency Parsing of Morphological and Syntactic Structures

If we look back at the Figure 2, it's clear that a dependency relation is correctly parsed if and only if three conditions are met: Firstly, words at both ends of the dependency are correctly segmented. Secondly, part-of-speech tags are correct for both words. Thirdly, the direction of the dependency relation are correct. Of course, if labeled precision and recall is to be measured, the label of the dependency relation should also be correctly recovered. Let nc be the number of dependencies correctly parsed with respect to these criterion, no be the total number of dependencies in the output, and nr the number of dependencies in the reference. Then precision is defined to be $p = nc/no$ and recall is defined to be $r = nc/nr$.

6.1.1 Results on the Original CTB5

We first train our unified dependency parser with the original treebank CTB5. In this case, all words are considered to be flat, with no internal structures. The result are shown in Table 7. Note that on exactly the same testing data, i.e, the original CTB5, unified parser performs much better than the result of a pipelined approach reported by Zhang and Clark (2011). There are about 30% of relative error reduction for the unlabeled dependency parsing results. This is yet another evidence of the advantage of joint modeling in natural language processing, details of which will be discussed in Section 7.

	P	R	F
original dependencies in CTB5	82.13	84.49	83.29
ZN2011 with Gold segmentation & POS	N/A	N/A	84.40
original dependencies plus word structures	85.71	87.18	86.44

Table 8: Evaluation results on CTB5 with word structures annotated. All results are labeled scores.

6.1.2 Results on CTB with Structures of Words Annotated

Then we train the parser with CTB5 augmented with our annotations of internal structures of words. For purpose of better comparability, we report results on both the original dependencies of CTB5 and on the dependencies of CTB5 plus those of the internal structures of words. The results are shown in Table 8. First, note that compared to another result by Zhang and Nivre (2011), whose input were sentences with gold standard word segmentation and POS tags, our F-score is only slightly lower even with input of unsegmented sentences. This is understandable since gold-standard segmentation and POS tags greatly reduced the uncertainty of parsing results.

For the unified parser, the improvement of F-score from 79.72% to 83.29% is attributed to the fact that with internal structures of words annotated, parsing of syntactic structures is also improved due to the similarity of word and phrase structures mentioned in Section 1, and also due to the fact that many phrase level dependencies are now facing a much less severe problem of data sparsity. The improvement of F-score from 83.29% to 86.44% is attributed to the annotation of word structures. Internal structures of words are be mostly local in comparison with phrase and sentence structures. Therefore, with the addition of word structures, the overall dependency parsing accuracy naturally can be improved.

6.2 Chinese Word Segmentation

From the example in Figure 2, it is clear that output of unified parser contains Chinese word segmentation information. Therefore, we can get results of word segmentation for each sentence in the test sets,

	P	R	F
K2009	N/A	N/A	97.87
This Paper	97.63	97.38	97.50

Table 9: Word segmentation results of our parser and the best performance reported in literature on the same dataset. K2009 is the result of Kruengkrai et al. (2009).

	P	R	F
K2009	N/A	N/A	93.67
ZC2011	N/A	N/A	93.67
This Paper	93.42	93.20	93.31

Table 10: Joint word segmentation and POS tagging scores. K2009 is result of Kruengkrai et al. (2009). ZC2011 is result of Zhang and Clark (2011).

and evaluate their accuracies. For maximal comparability, we train the unified parser on the original CTB5 data used by previous studies. The result is in Table 9. Despite the fact that the performance of our unified parser does not exceed the best reported result so far, which probably might be caused by some minute implementation specific details, it's fair to say that our parser performs at the level of state-of-the-art in Chinese word segmentation.

6.3 Joint Word Segmentation and POS Tagging

From Figure 2 we see that besides word segmentation, output of the unified parser also includes part-of-speech tags. Therefore, it's natural that we evaluate the accuracy of joint Chinese word segmentation and part of speech tagging, as reported in previous literature (Kruengkrai et al., 2009). The results are in Table 10, in which for ease of comparison, again we train the unified parser with the vanilla version of CTB5. We can see that unified parser performs at virtually the same level of accuracy compared with previous best systems.

7 Related Work

Researchers have noticed the necessity of parsing the internal structures of words in Chinese. Li (2011) gave an method that could take raw sentences as input and output phrase structures and internal structures of words. This paper assumes that the input are unsegmented, too, and our output also includes both word and phrase structures. There are

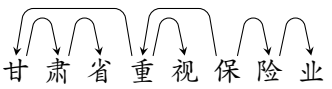

Figure 4: Example output of Zhao's parser.

two key differences, though. The first is we output dependency relations instead of constituent structures. Although dependencies can be extracted from the constituent trees of Li (2011), the time complexity of their algorithm is $O(n^5)$ while our parser runs in linear time. Secondly, we specify the details of annotating structures of words, with the annotations being made publicly available.

Zhao (2009) presented a dependency parser which regards each Chinese character as a word and then analyzes the dependency relations between characters, using ordinary dependency parsing algorithms. Our parser is different in two important ways. The first is we output both part-of-speech tags and labeled dependency relations, both of which were absent in Zhao's parser. More importantly, the APPEND transition for handling flat words were unseen in previous studies as far as we know. The difference can best be described with an example: For the sentence in Section 3, Zhao's parser output the result in Figure 4 while in contrast our output is Figure 2.

In recent years, considerable efforts have been made in joint modeling and learning in natural language processing (Lee et al., 2011; Sun, 2011; Li et al., 2011; Finkel and Manning, 2009; Kruengkrai et al., 2009; Jiang et al., 2008; Goldberg and Tsarfaty, 2008). Joint modeling can improve the performance of NLP systems due to the obvious reason of being able to make use of various levels of information simultaneously. However, the thesis of this paper, i.e, unified parsing of Chinese word and phrase structures, bears a deeper meaning. As demonstrated in Section 1 and by Li (2011), structures of words and phrases usually have significant similarity, and the distinction between them is very difficult to define, even for expert linguists. But for real world applications, such subtle matters can safely be ignored if we could analyzed morphological and syntactic structures in a unified framework. What applications really cares is structures instead of whether a linguistic unit is a word or phrase.

Another notable line of research closely related to the present work is to annotate and parse the flat structures of noun phrases (NP) (Vadas and Curran, 2007; Vadas and Curran, 2011). This paper differs from those previous work on parsing NPs in at least two significant ways. First, we aim to parse all kinds of words (e.g, nouns, verbs, adverbs, adjectives etc) whose structures are not annotated by CTB, and whose presence could cause lots of pseudo OOVs and incompatible annotations. Second, the problem we are trying to solve is a crucial observation specific to Chinese language, that is, in lots of cases forcing a separation of words and phrases leads to awkward situations for NLP systems. Remember that in Section 2 we demonstrated that all corpora we examined had the problem of pseudo OOVs and incompatible annotations. In comparison, the problem Vadas and Curran (2007) tried to solve is a lack of annotation for structures of NPs in currently available treebanks, or to put it in another way, a problem more closely related to treebanks rather than certain languages.

8 Discussion and Conclusion

Chinese word segmentation is an indispensable step for traditional approaches to syntactic parsing of Chinese. The purpose of word segmentation is to decide what goes to words, with the remaining processing (e.g, parsing) left to higher level structures of phrases and sentences. This paper shows that it could be very difficult to make such a distinction between words and phrases. This difficulty cannot be left unheeded, as we have shown quantitatively that in practice it causes lots of real troubles such as too many OOVs and incompatible annotations. We showed how these undesirable consequences can be resolved by annotation of the internal structures of words, and by unified parsing of morphological and syntactic structures in Chinese.

Unified parsing of morphological and syntactic structures of Chinese can also be implemented with a pipelined approach, in which we first segment input sentences into words or affixes (i.e, with the finest possible granularity), and then we do part-of-speech tagging followed by dependency (or constituent) parsing. However, a unified parsing approach using a single model as presented in this paper offers several advantages over pipelined approaches. The first one is that joint modeling tends to result in higher accuracy and suffer less from error propagation than do pipelined methods. Secondly, both the unified model and the algorithm are conceptually much more simpler than pipelined approaches. We only need one implementation of the model and algorithm, instead of several ones in pipelined approaches. Thirdly, our model and algorithm might comes closer to modeling the process of human language understanding, because human brain is more likely a parallel machine in understanding languages than an alternative pipelined processor. Hence this work, together with previous studies by other authors like Li (2011) and Zhao (2009), open up a possibly new direction for future research efforts in parsing the Chinese language.

Acknowledgments

Reviewers of this paper offered many detailed and highly valuable suggestions for improvement in presentation. The authors are supported by NSFC under Grant No. 90920004 and National 863 Program under Grant No. 2012AA011102.

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.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, July.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334, Boulder, Colorado, June. Association for Computational Linguistics.

Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *Proceedings of ACL-08: HLT*, pages 371–379, Columbus, Ohio, June. Association for Computational Linguistics.

C. F. Hockett. 1969. *A Course in Modern Linguistics*. Macmillan.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008. A cascaded linear model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL-08: HLT*, pages 897–904, Columbus, Ohio, June. 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.

John Lee, Jason Naradowsky, and David A. Smith. 2011. A discriminative model for joint morphological disambiguation and dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 885–894, Portland, Oregon, USA, June. Association for Computational Linguistics.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for Chinese POS tagging and dependency parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1180–1191, Edinburgh, Scotland, UK., 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.

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.

David Vadas and James R. Curran. 2011. Parsing noun phrases in the penn treebank. *Computational Linguistics*, 37(4):753–809.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceeding of the 8th International Workshop of Parsing Technologies (IWPT)*, pages 195–206, Nancy, France.

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

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June. Association for Computational Linguistics.

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.