Zhou GD, Li PF. Improving syntactic parsing of Chinese with empty element recovery. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(6): 1106–1116 Nov. 2013. DOI 10.1007/s11390-013-1401-x

Improving Syntactic Parsing of Chinese with Empty Element Recovery

Guo-Dong Zhou (周国栋), *Senior Member*, *CCF*, *Member*, *ACM*, *IEEE*, and Pei-Feng Li (李培峰), *Member*, *CCF*

Natural Language Processing Lab, School of Computer Science and Technology, Soochow University, Suzhou 215006, China

E-mail: {gdzhou, pfli}@suda.edu.cn

Received June 20, 2012; revised September 18, 2013.

Abstract This paper puts forward and explores the problem of empty element (EE) recovery in Chinese from the syntactic parsing perspective, which has been largely ignored in the literature. First, we demonstrate why EEs play a critical role in syntactic parsing of Chinese and how EEs can better benefit syntactic parsing of Chinese via re-categorization from the syntactic perspective. Then, we propose two ways to automatically recover EEs: a joint constituent parsing approach and a chunk-based dependency parsing approach. Evaluation on the Chinese TreeBank (CTB) 5.1 corpus shows that integrating EE recovery into the Charniak parser achieves a significant performance improvement of 1.29 in $F1$-measure. To the best of our knowledge, this is the first close examination of EEs in syntactic parsing of Chinese, which deserves more attention in the future with regard to its specific importance.

Keywords Chinese syntactic parsing, empty element recovery, joint constituent parsing, chunk-based dependency parsing

1 Introduction

The UPenn series of treebanks (by Marcus *et al.*[1], e.g., Penn TreeBank for English①, Chinese TreeBank②, and Arabic TreeBank③) annotate a great deal of useful information for further semantic interpretation, including both functional tags and markers of empty elements (EEs) that indicate non-local dependencies, discontinuous constituents, and certain missing elements. In principle, EEs are co-indexed with their antecedents in the same sentence, if any. For example, in the sentence fragment $S1$, the phrase "who" is co-indexed with an EE of type *T* in the embedded S and the functional tag SBJ indicates that this EE is the subject of that S:

$S1$:[WHNP-1 who] NP want
[S [NP-SBJ-1 *T*] to VP].

Accordingly, EEs with co-indexation and functional tags provide much useful information in further determining the semantic interpretation of a sentence in the form of predicate-argument structures or deep dependencies. However, most of the state-of-the-art syntactic parsers (e.g., proposed by Collins[2]; Charniak[3]; Petrov and Klein[4]) ignore such useful information.

Furthermore, EEs are closely related with zero anaphora, particularly in Chinese[5-6]. For example, Kong *et al.*[6] proposed a tree kernel based framework for zero anaphora resolution in Chinese. This is largely due to the fact that Chinese is a pro-drop language and zero anaphors occur much more frequently in Chinese than in English. As shown by Kim[7], the use of overt subjects in English is over 96%, while this percentage is only 64% in Chinese, indicating that zero anaphors due to the lack of overt subjects are much more prevalent in Chinese. Recently, Chung *et al.*[8] studied the effect of EEs in machine translation and showed that proper handling of EEs in Chinese sentences improves the BLEU (bilingual evaluation understudy) score. In this paper, we focus on the interaction between syntactic parsing and EE recovery, and leave the identification of their antecedents to the future.

Previous studies in the literature focus on the recovery of EEs and identification of their antecedents by assuming the availability of constituent parse trees and ignoring the interaction between syntactic parsing and EE recovery. It is not surprising to observe that the performance of EE recovery relies heavily on the quality of constituent parse trees. For example, Campbell[9] reported that English EE recovery achieved the performa-

Regular Paper

Supported by the National Natural Science Foundation of China under Grant Nos. 61273320, 61331011, 61070123, and the National High Technology Research and Development 863 Program of China under Grant No. 2012AA011102.

①http://www.cis.upenn.edu/∼treebank/, Sept. 2013.

②http://www.cis.upenn.edu/∼chinese/ctb.html, Sept. 2013.

③http://www.ircs.upenn.edu/arabic/, Sept. 2013.

nce of 93.7 in $F1$-measure on gold-standard constituent parse trees, while the performance dropped to 76.7 in $F1$-measure on automatic constituent parse trees using the Charniak parser (Charniak[3]). Similarly, Guo *et al.*[10] achieved the performance of 92.2 in $F1$-measure for trace insertion (a task similar to EE recovery) in Chinese on gold-standard constituent parse trees, while the performance dropped to 64.7 in $F1$-measure on automatic constituent parse trees using the Bikel parser[11].

It is well known that besides word segmentation, POS tagging (with the accuracy of about 95%) is another major reason why syntactic parsing of Chinese obtains a much lower performance than that of English, with a performance gap of about 10 in $F1$-measure. EEs could be another major reason for such a performance gap, though it has not been well examined.

In this paper, we explore EE recovery in Chinese language from the syntactic parsing perspective. First, we demonstrate why EE recovery plays a critical role in syntactic parsing of Chinese and how EEs can better benefit syntactic parsing of Chinese via re-categorization from the syntactic perspective. Then, we propose two ways to recover EEs: a joint constituent parsing approach, and a chunk-based dependency parsing approach. For simplicity, we assume gold-standard word segmentation in this paper. Evaluation on the Chinese TreeBank (CTB) 5.1 corpus shows the impact of automatic EE recovery on syntactic parsing of Chinese and the superiority of the chunk-based dependency parsing approach over the joint constituent parsing approach.

The rest of this paper is organized as follows. Section 2 reviews the related work on EE recovery. Section 3 introduces EEs in Chinese language and the impact of gold-standard EEs on syntactic parsing of Chinese, which motivates our re-categorization of EEs from the syntactic perspective. Section 4 describes our EE re-categorization scheme and shows the impact of re-categorized gold-standard EEs on syntactic parsing of Chinese. Section 5 proposes the approaches to EE recovery. Section 6 presents the experimental results. Finally, Section 7 draws conclusions.

2 Related Work

While there is a certain amount of related work on EE recovery for English[9,12-14], there are only a few studies on EE recovery for Chinese[8,10,15-17].

2.1 EE Recovery in English

Johnson[12] pioneered the research on EE recovery for English by proposing a simple pattern-matching algorithm, which recovers EEs and identifies their co-indexed antecedents in constituent parse trees. He also proposed an evaluation procedure for EE recovery which is independent of most of the details in the constituent structure, and is thus widely adopted in the literature.

Dienes and Dubey[13-14] made further exploitation of recovering EEs and identifying their antecedents. As alternatives to the post-processing approach as adopted in [12], they explored two other approaches: pre-processing which recovers EEs before parsing, and in-processing which integrates EE recovery and antecedent identification into a constituent parser. In particular, the pre-processing approach first recovers EEs from POS-tagged sentences, then feeds EEs as overt "words" into a syntactic parsing model, and finally recovers their antecedents from the syntactic parsing output. Evaluation shows that the pre-processing approach achieves better performance on EE recovery than both post-processing and in-processing approaches.

Campbell[9] turned back to post-processing using a rule-based approach from the linguistic perspective. He proposed that the annotation of EEs in a treebank largely depends on linguistic principles, such as the ones described in the Government Binding theory. Evaluation shows that his approach outperforms previous work on both EE recovery and antecedent identification.

Furthermore, there are some studies on recovering non-local dependencies in English sentences, a similar task to antecedent identification of EEs[18-19].

2.2 EE Recovery in Chinese

Where Chinese is concerned, Guo *et al.*[10] explored the recovery of non-local dependencies in Lexical-Function Grammar.

Yang and Xue[15] did a preliminary study on recovering EEs. They simplified the problem as a binary classification task by predicting whether there is an EE before each word in a sentence, and did not further differentiate EE types. Recently, they[16] also examined EE detection from the dependency parse tree perspective by first converting the Chinese TreeBank from a phrase structure representation to a dependency representation with the ECs preserved.

Chung and Gildea[8] recognized two major types of EEs (i.e., *PRO* and *pro*, as described later in Table 1) using various kinds of approaches such as pattern matching, conditional random fields, and parsing.

2.3 Integration Between EE Recovery and Syntactic Parsing

Among the related work, Dienes and Dubey[13-14] linked syntactic parsing and EE recovery. However,

they mainly employed syntactic parsing for EE recovery in English. In particular, they discussed it from the perspective of EE recovery and antecedent identification, rather than from the syntactic parsing perspective.

As for Chinese, Cai *et al.*[17] presented a simple language-independent method for integrating EE recovery into syntactic parsing. They took a state-of-the-art parsing model, the Berkeley parser[4], trained it on data with explicit EEs, and tested it on word lattices that can non-deterministically insert EEs anywhere. However, they did not report the impact of EE recovery on syntactic parsing of Chinese.

3 EEs in Chinese Language

To better understand this paper, we first introduce some background knowledge on EEs in Chinese language (especially in the Chinese TreeBank 5.1 corpus) and then empirically show its potential impact on syntactic parsing of Chinese with gold-standard EEs.

3.1 EEs in Chinese TreeBank

In CTB, EEs are marked in a constituent parse tree which represents the hierarchical structure of a sentence. Table 1 shows the major types of EEs defined in CTB 5.1, along with their distribution. As we can see from Table 1, the distribution of these EE types is very uneven. Among them, type *T* occupies about 31%, type *PRO* occupies about 20%, type *pro* occupies about 16%, type *OP* occupies about 31%, while the other two types occupy only about 3% totally. More details about the annotation scheme of EEs in CTB can be found in [20].

Table 1. EEs in CTB 5.1 (chtb 001∼chtb 931)

EE Type	Number of Instances (Percentage)	Description
T	5 677 (31.19)	Trace of A'-movement such as topicalization
*	180 (0.99)	Trace of A-movement
PRO	3 605 (19.81)	Used in control structures
pro	2 886 (15.86)	Used to indicate a pro-drop
OP	5 571 (30.61)	Null operator used in relative constructions
RNR	283 (1.55)	Used for right node raising

Following are some examples illustrating different EE types in CTB 5.1.

*S*2: 据 *pro* 认为, 此 次 访问 的 目的 是 为了 *PRO* 改善 *RNR* 和 发展 两 国 关系, 加强 双边 经贸 合作, 扩大 泰国 在 缅甸 的 投资。(It is thought that the purpose of this visit is to improve and develop the relationship between the two countries, to strengthen bilateral economic and trade cooperation, and to expand Thailand's investment in Myanmar.)

*S*3: 到 目前 为止 , 全 区 已 有 四百一十 家 企业 , *OP* *T* 被 认定 * 为 高 新 技术 产业 的 有 二百二十三 家。(So far, there are already 410 enterprises in the whole zone, among which 223 have been identified as new, high level technology enterprises.)

3.2 Impact of Gold-Standard EEs on Syntactic Parsing of Chinese

To illustrate the importance of Chinese EE recovery, we empirically investigate the impact of gold-standard EEs on syntactic parsing of Chinese. To this end, we examine three experimental settings: 1) with no EEs involved; 2) with gold-standard EE positions known in advance; 3) with gold-standard EE positions and types known in advance.

We train the Charniak parser on CTB 5.1 using the widely adopted splitting, i.e., 648 files (chtb 0081»0899.fid) for training, 40 files (chtb 0041»0080.fid) for development, and 72 files (chtb 0001»0040.fid and chtb 0900»0931.fid) for testing. Table 2 shows the impact of gold-standard EEs on syntactic parsing of Chinese under different experimental settings. It is worth noting that EEs in the outputs of the last two experiments are stripped off in performance evaluation, so as to have a fair comparison with traditional syntactic parsing. From the results we can find that:

- Given gold-standard EE positions, the syntactic parser performance improves by 2.90 in $F1$-measure. This indicates the potential of EE recovery in syntactic parsing of Chinese.
- It is surprising to notice that further consideration of gold-standard EE types has even negative impact on syntactic parsing of Chinese.

Table 2. Contribution of Gold-Standard EEs on Syntactic Parsing of Chinese

Settings	Recall (%)	Precision (%)	$F1$
Without EEs	79.74	81.95	80.83
With gold-standard EE positions	83.86	84.62	83.73
With gold-standard EE positions and types	83.62	84.42	83.51

3.3 Impact of Gold-Standard EEs on Syntactic Parsing of English: A Comparison

As shown above, EEs play a critical role in syntactic parsing of Chinese, given gold-standard EEs. Thus, it might be interesting to investigate the impact of EEs in syntactic parsing of English. To this end, we test the Charniak parser on the Penn TreeBank (PTB) 2.0 corpus, using the widely adopted splitting, i.e., sections 2»21 for training, section 24 for development, and section 23 for testing. Table 3 illustrates the contribution

of gold-standard EEs on syntactic parsing of English. Although EEs in PTB 2.0 account for 6.32% of words, comparable to 6.76% in CTB 5.1, it shows that the performance improvement by integrating gold-standard EEs in syntactic parsing of English is not so prominent as that of Chinese. This may be largely due to the subject pro-drop nature of Chinese, which seals the critical role of EE recovery on syntactic parsing of Chinese, while the probability of dropping subjects in English is significantly less frequent. This also motivates us to re-categorize EEs in CTB from the syntactic perspective, e.g., whether an EE plays a subject role in a sentence.

Table 3. Contribution of Gold-Standard EEs on Syntactic Parsing of English

Settings	Recall (%)	Precision (%)	$F1$
Without EEs	89.60	89.87	89.73
With gold-standard EE positions	90.40	90.47	90.43
With gold-standard EE positions and types	90.48	90.68	90.58

4 Re-Categorization of EEs in Chinese TreeBank

Just as stated above, further consideration of gold-standard EE types has even negative impact on syntactic parsing of Chinese on CTB 5.1. This may be due to that EE types defined in CTB are a reflection of different kinds of discourse-level cohesion phenomena from the semantic perspective and thus may not be suitable for syntactic parsing from the syntactic perspective. Given the fact that such EE types definitely contain some useful information, it may be worthwhile to re-categorize them from the syntactic perspective and thus make better use of such information.

4.1 Re-Grouping EEs

Take the three fragments $S4$, $S5$ and $S6$ shown below as examples. *T*, *PRO* and *pro* all represent an EE in subject position, with subtle difference from the syntactic perspective, in that type *T* requires an antecedent in the same sentence (e.g., WHNP-1), type *pro* indicates a dropped pronoun and thus can be replaced by an overt NP (e.g., NP (政府/government)), and type *PRO* marks control structures and thus cannot be replaced by an overt NP. Chung *et al.*[8] compared the matching patterns of type *PRO* and type *pro* and discovered their similarity from the syntactic perspective. For example, pattern (IP (NP (-NONE- *PRO*)) VP) accounts for about 95% of type *PRO* while (IP (NP (-NONE- *pro*)) VP) accounts for about 85% of type *pro*. As a result, this makes the performance of recovering type *PRO* and type *pro* very low (about 63 in $F1$-measure for type *PRO* and about 44 in $F1$-measure for type *pro*), largely due to the difficulty in differentiating these two types.

As an alternative, we keep type *OP* due to its special structures, and mix up all other EEs and regroup them into three types according to their syntactic functions. As syntactic parsing is syntax-driven, it is reasonable to believe that the regrouped EE types are more suitable for syntactic parsing. In particular, we re-label all EEs in subject position with pattern "(IP ...(-NONE- EE) VP..." as type *SBJ*, while all EEs in object position with pattern "(VP ...VV/VA/VE/VC (-NONE- EE)...)" as type *OBJ*. Table 4 lists the four types of regrouped EEs. Accordingly, type *T* in $S4$, type *pro* in $S5$, and type *PRO* in $S6$ will be re-labeled as type *SBJ*, while type *T* in $S7$ re-labeled as type *OBJ*.

$S4$: [WHNP-1 *OP*] [IP [NP-1 *T*] [NP 一九九四年/in 1994] [VP 成立/establish]] 的/DEC [NP 政策性银行/a policy bank].

$S5$: [IP [NP *pro*] [NP-TMP 一九九四年/in 1994] [VP 成立 政策性银行/establish a policy bank]].

$S6$: [NP 政府/government] 同意/agree [IP [NP *PRO*] [NP 一九九四年/in 1994] [VP 成立 政策性银行/establish a policy bank]].

$S7$: [WHNP-1 *OP*] [IP [NP 政府/government] [NP 一九九四年/in 1994] [VP 并购/merge [NP-1 *T*]]] 的/DEC [NP 政策性银行/a policy bank].

Table 4. Regrouped EEs in CTB 5.1 (chtb 001∼chtb 931)

EE Type	Number of Instances (Percentage)	Description
OP	5 571 (30.61)	Kept same
SBJ	10 082 (55.39)	EEs in subject position
OBJ	1 653 (9.08)	EEs in object position
OTHER	896 (4.92)	EEs in other cases

4.2 Impact of Re-Grouped Gold-Standard EEs on Syntactic Parsing of Chinese

According to the annotation principles in CTB, every verb phrase (VP) should have a subject to satisfy its grammatical requirements. Moreover, a relative clause should be adjoined to its head noun phrase. For example, "的/DEC" is considered to be a complementizer and should be put inside the CP co-indexed with the trace in the clause. Here, CP is a clause label in CTB introduced by a (possibly empty) complementizer. Note that in Chinese, the relative operator is always omitted. Actually, further examination of automatic parse trees on the development data of CTB 5.1 (Table 2: Without EEs) shows that about 20% of sentences have subject attachment errors, while about 40% of sentences have VP/IP coordinate errors.

Table 5 illustrates the impact of re-grouped gold-standard EEs on syntactic parsing of Chinese under the same experimental settings as in Subsection 3.2. It shows that re-grouped gold-standard EEs contribute much to syntactic parsing by 4.13 in $F1$-measure. This justifies re-categorization of EEs from the syntactic perspective, which further improves the performance over EE positions only by 1.23 (4.13 ¡ 2.90) in $F1$-measure, compared with the negative impact of original EE types (please see Table 2). It also shows that the contribution of different re-grouped EEs varies, with type *SBJ* contributing most, followed by type *OP*, while the contributions of type *OBJ* and type *OTHER* are very limited and can be almost ignored (much due to the limited role of type *OBJ* in determining the overall constituent structure and the small scale of type *OTHER*). Hereafter, we focus on recovering EEs of type *SBJ* and type *OP* and simply ignore EEs of type *OBJ* and type *OTHER*.

Table 5. Contribution of Re-Grouped Gold-Standard EEs on Syntactic Parsing of Chinese

Settings	Recall (%)	Precision (%)	$F1$
Without EEs	79.74	81.95	80.83
+*SBJ*	82.57	84.50	83.53
+*OP*	82.61	84.22	83.41
+*OBJ*	80.26	82.16	81.20
+*OTHER*	80.11	82.24	81.16
+*SBJ*+*OP*	83.98	85.72	84.84
+all EEs	84.08	85.86	84.96

5 EE Recovery

In this section, we explore EE recovery in three scenarios. As a baseline, we first recover EEs on constituent parse trees to explore the dependency of EE recovery on syntactic parsing, similar to the post-processing approach[12,15]. Then, we propose two ways to explore EE recovery for syntactic parsing: a joint constituent parsing approach, and a chunk-based dependency parsing approach. In all the three scenarios, we explicitly (the first and third approaches) or implicitly (the second approach) judge whether a VP should have an EE of type *SBJ*, and an EE of type *OP*, respectively. The obvious advantage of this solution is that it can naturally handle the situation where two or more EEs are consecutive. This is important since about 30% of EEs have another EE to either its left or right side.

5.1 EE Recovery on Constituent Parse Trees

The structural annotation of constituent parse trees provides a great deal of useful information to recover EEs of type *SBJ* and type *OP*. Our statistics on CTB 5.1, for example, shows that all EEs of type *SBJ* occur in the production pattern of IP ! ¢¢¢*SBJ*VP¢¢¢, while all EEs of type *OP* occur in the production pattern of CP ! *OP*¢¢¢. The distinction between type *SBJ* and type *OP* over the structural context makes it natural to recover them separately.

*Recovering EEs of Type *SBJ**. Since EEs of type *SBJ* represent implicit subjects in subject positions, we can easily insert EEs of type *SBJ* in appropriate subject positions in a constituent parse tree. Specifically, given a production IP ! X_1 ¢¢¢ X_nVP¢¢¢, where IP (a bracket label in CTB, similar to S in PTB for English) indicates a simple clause, and X_i represents a constituent, we perform a binary classification to predict whether an EE of type *SBJ* is needed in this occasion. If yes, an EE of type *SBJ* is inserted before VP. The features adopted here focus on whether there is an overt subject among X_1 to X_n. Therefore, they are local and can be easily derived from the context of the production.

*Recovering EEs of Type *OP**. EEs of type *OP* are assumed to be in the scope of CP, which indicates a relative clause adjoined to the head noun phrase. Our statistics on CTB 5.1 shows that with few exceptions, there is an EE of type *OP* before each CP. Therefore, given a parse tree, we simply locate CPs and insert an EE of type *OP* before each CP.

5.2 Joint Constituent Parsing Approach

The joint constituent parsing approach is largely motivated by Finkel *et al.*[21], who augmented constituent labels with named entity recognition information. The idea behind is that joint learning of two related tasks allows the information from each type of annotation to improve the performance of the other.

In the training phase, we re-label gold-standard constituent parse trees by augmenting syntactic constituent labels with EE information as follows. Given a production of IP ! ¢¢¢*SBJ*VP¢¢¢, the algorithm starts from *SBJ*'s right-hand VP node and then iteratively moves one level down to the leftmost child of the current node till it reaches a non-VP node. At each level, the algorithm augments the current node with the sub-tag of SBJ, as shown in Fig.1. Similarly, the algorithm augments nodes with the sub-tag of OP from the production of CP ! *OP*¢¢¢ and ends when it reaches a non-CP node, as shown in Fig.1.

In the testing phase, we feed the syntactic parsing model a sentence with EEs stripped off and recover EEs from the complex constituent labels in the output parse tree.

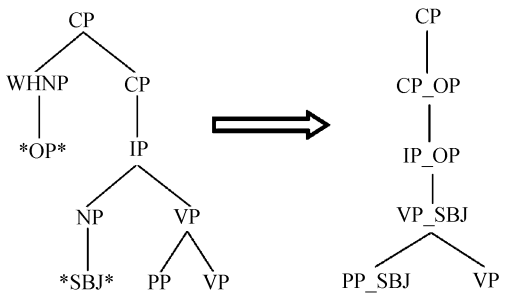

Fig.1. Example of re-labeling syntactic parse tree for joint learning.

5.3 Chunk-Based Dependency Parsing Approach

Dienes *et al.*[14] proposed an approach to recover EEs from POS-tagged sentences and achieved the $F1$-measure of 79.1 for English. However, this approach does not perform well on Chinese, due to the inherent differences between the two languages. A preliminary result shows that it only achieves the $F1$-measure of 61.4 in Chinese, even with gold-standard POS tags. This is consistent with the low performance achieved by Yang *et al.*[15] The reason for the low performance lies in the fact that the input unit is too fine-grained. This makes it hard to cover an EE and its corresponding verb within a window of a reasonable size (e.g., 5 words). Alternatively, we propose an EE recovery approach on chunks. Moreover, instead of examining each chunk and deciding whether to leave it as it is, insert type *SBJ*, or insert type *OP*, we recover the EE using a chunk-based dependency parsing model. This is because the former approach fails to handle such cases where two or more EEs are consecutive, while the latter approach can handle them quite naturally.

Since both EEs of type *SBJ* and type *OP* are verb phrase driven, we refer to m-chunks in this paper as the proper maximal constituents that do not contain a VP node. Given the m-chunk sequence, we first perform dependency parsing on m-chunks and then recover EEs of type *SBJ* and type *OP* separately according to the output of dependency parsing. For the effect of EE recovery on syntactic parsing, the word sequence with automatically recovered EEs inserted explicitly is fed to a syntactic parsing model, which is trained on the gold-standard constituent parse trees with EEs kept.

5.3.1 Chunk-Based Dependency Parsing Model

Our model is mainly a shift-reduce history-based one, as described in [22], which maintains a stack to store processed tokens and a queue to store remaining input tokens. It performs dependency parsing in $O(n)$ time, where n is the number of tokens in the input sentence. As a shift-reduce model, it defines four types of parsing behaviors: Shift, Reduce, Left-Arc, and Right-Arc.

In this paper, we extend the above model to address dependency parsing on m-chunks in the following two dimensions:

- The basic input unit is an m-chunk instead of a token. In the training phase, we extract gold-standard m-chunk sequences from gold-standard constituent parse trees directly; while in the testing phase, we extract automatic m-chunk sequences from automatic constituent parse trees directly.
- We define four types of dependency relations between two verb m-chunks: VPCOOR, denoting that the two verb m-chunks are coordinated; IPCOOR, denoting that the two verb m-chunks have different respective subjects first and are then coordinated; IPMOD, denoting that the former verb m-chunk together with its subject forms the subject of the latter VP; and VPOBJ, denoting that the latter verb m-chunk is the object of the former verb m-chunk. Fig.2 demonstrates the four types of dependency relations from a constituent parsing perspective. Discovering the dependency relations between two verb m-chunks is helpful in recovering EEs. For example in Figs. 2(a), 2(b), and 2(c), corresponding to the cases of VPCOOR, IPMOD and VPOBJ, respectively, there is a need to insert an EE in front of VP2 (i.e., the 2nd VP/VV on the leave).

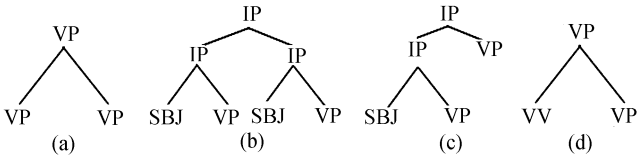

Fig.2. Four types of dependency relations between two verb m-chunks (e.g., the two VPs in the leaf nodes of (a), (b), and (c), VV and VP in the leaf nodes of (d)). (a) VPCOOR. (b) IPCOOR. (c) IPMOD. (d) VPOBJ.

For simplicity, we derive the dependency relations among m-chunks according to the head rules, i.e., according to the dependency relationship between a modifier and its head m-chunk. Fig.3 illustrates an example sentence of dependency relations among m-chunks in chunk-based dependency parsing. Basically, our chunk-based dependency parser is a classifier-driven parsing model, which relies on features derived from the stack and the queue to predict subsequent parsing behavior. Supposing TOP is the token on top of the stack and NEXT is the next token in the queue, the features used in our model are listed as follows:

- TOP-related features: the chunk tag, headword and its POS tag of TOP, TOP's parent chunk, TOP's last child, the chunk tag of the m-chunk immediately

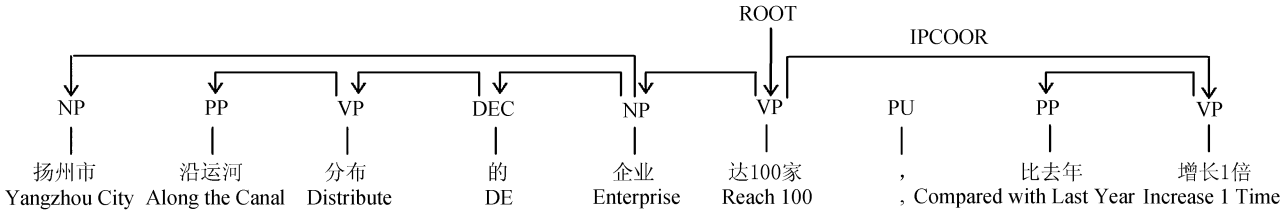

Fig.3. Example of dependency-parsed m-chunks.

before TOP in the stack; combined chunk tags of TOP's children to the left and TOP's children to the right.

² NEXT-related features: the chunk tag, headword and its POS tag of NEXT, the m-chunk immediately after NEXT in the queue, and the last child of NEXT.

² features related to both TOP and NEXT: whether TOP and NEXT are adjacent; whether there is punctuation between TOP and NEXT; whether there is a CC between TOP and NEXT.

5.3.2 Recovering EEs of Type *SBJ*

Given dependency relations returned from the chunk-based dependency parsing model, it is natural to predict EEs of type *SBJ* by casting it as a classification problem. For example, given a verb m-chunk VP with n modifiers, as shown in Fig.4, it has $n+2$ possibilities, namely, inserting a *SBJ* in front of M_i $(1 \leqslant i \leqslant n)$, inserting a *SBJ* in front of VP itself, or inserting no *SBJ* at all. In particular, we make $n+1$ binary classifications and every m-chunk (M_i and VP in Fig.4) receives a probability distribution of inserting a *SBJ* in front of it. If all probabilities are less than 0.5, then no *SBJ* is inserted. Otherwise, the m-chunk with the highest probability has a *SBJ* inserted.

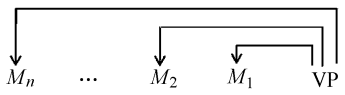

Fig.4. Verb m-chunk and its n modifiers.

Assuming that the current verb m-chunk and modifying m-chunk are VERB and MOD respectively, our model uses the following features to predict an EE of type *SBJ* in front of MOD:

² features on MOD: the chunk tag, headword and its POS tag of MOD; the first and last words of MOD;

² features on VERB: the chunk tag, headword and its POS tag of VERB; the category of VERB; the chunk tag and headword of VERB's parent;

² features on MOD and VERB: the dependency relation between MOD and VERB; whether MOD and VERB are the same node; whether MOD is the first child of VERB; whether MOD is the last child of VERB; whether there is punctuation between MOD and VERB; the path from MOD to VERB;

² features on modifying m-chunks and VERB: whether there is the dependency relation of VP-COOR/IPCOOR/IPMOD/VPOBJ between VERB and all the other modifying m-chunks/the neighboring modifying m-chunks of M_{i-1} and M_{i+1} (assuming MOD is M_i). This is to include competitive information from the dependency relationship between VERB and other modifying m-chunks.

5.3.3 Recovering EEs of Type *OP*

Similar to EE recovery of type *SBJ*, EE recovery of type *OP* can also be recast as a classification problem. Moreover, due to the special structure of relative clauses in CTB, it is found that an EE of type *OP* is always located to the left of the leftmost m-chunk dominated by its verb m-chunk. For example, if the verb m-chunk VP in Fig.4 has an EE of type *OP*, then such an EE must locates ahead of the leftmost m-chunk dominated by the governing VP. Therefore, predicting an EE of type *OP* equates to predicting whether a verb m-chunk forms a relative clause, which indicates the necessity of an EE of type *OP*. To this end, given a verb m-chunk VERB, we employ the following features in our model:

² VERB-related features: the chunk tag, headword and its POS tag of VERB, the subcategory of VERB;

² features related with VERB's parental chunk PAR: the chunk tag and POS of its headword of PAR; the position of PAR (left or right of VERB).

6 Experimentation

We have evaluated our approaches on CTB 5.1. The data splits for training, test, and development follow previous research in syntactic parsing and semantic role labeling of Chinese[23-25], as described in Subsection 3.2.

For the evaluation measurement on syntactic parsing, we adopt the Evalb script④ and report labeled recall, labeled precision, and $F1$-measure. All EE-related

④http://nlp.cs.nyu.edu/evalb/, Sept. 2013.

nodes or sub-labels are stripped off before evaluation. We also report recall, precision, and $F1$-measure for EE recovery following Johnson[12]. To see whether an improvement in $F1$-measure is statistically significant, we also conduct significance tests using a type of stratified shuffling[26] which is a type of computation-intensive randomized tests. In this paper, "$\ggg$", "$\grave{A}$", and ">" denote p-values less than or equal to 0.01, in-between (0.01, 0.05), and bigger than 0.05, which mean significantly better, moderately better and slightly better, respectively.

In addition, SVMLight⑤ is selected as our classifier. In order to handle multi-classification in dependency parsing on m-chunks, we apply the *one-vs-others* strategy, which builds multiple classifiers so as to separate one class from others.

6.1 Performance of EE Recovery

Table 6 shows the performance of EE recovery in the different scenarios. The first two rows show the performance of the EE recovery approach on constituent parse trees, as described in Subsection 4.1. The third row shows the performance of our joint constituent parsing approach, as described in Subsection 4.2. The last three rows demonstrate the performance of our chunk-based dependency parsing approach with different settings. In particular, dependency parsing on gold-standard m-chunks achieves unlabeled attachment score (UAS) of 90.67% and labeled attachment score (LAS) of 89.89%, respectively. For simplicity, we extract the gold-standard/automatic m-chunk sequences from the gold-standard/automatic constituent parse trees directly. Since an m-chunk is defined as the proper maximal constituents that do not contain a VP node, the performance of m-chunk recognition is decided by that of the syntactic parser, i.e., the Charniack parser as employed in this paper. As a result, m-chunk recognition achieves the performance of 91.71%, 92.13% and 91.92 in recall, precision and $F1$-measure, respectively.

Table 6 shows that:

1) The EE recovery approach on constituent parse trees achieves the performance of 96.83 in $F1$-measure on gold-standard constituent parse trees, suggesting the critical role of structured information in EE recovery. However, the apparent advantage of such information is diminished with a decrease of 32.88 (96.83 ¡ 63.95) in $F1$-measure when automatic constituent parse trees are adopted, suggesting the heavy dependence of this approach on the syntactic parsing performance. In this paper, the automatic constituent parse trees are from the Charniak parser which achieves 80.83 in $F1$-measure.

2) Compared with the EE recovery approach on constituent parse trees, the joint constituent parsing approach improves the performance by 1.41 (65.36 ¡ 63.95, $\grave{A}$) in $F1$-measure.

3) Given gold-standard m-chunks and gold-standard dependency relations among them, the chunk-based dependency parsing approach achieves the performance of 92.11 in $F1$-measure, much comparable to the performance of 96.83 achieved on gold-standard constituent parse trees. However, dependency parsing on automatic dependency relations decreases the performance of EE recovery greatly by 11.64 (92.11 ¡ 80.47, $\ggg$) in $F1$-measure and a further reduction of 10.27 (80.47 ¡ 70.20, $\ggg$) in $F1$-measure on automatic m-chunks .

4) The chunk-based dependency parsing approach outperforms the joint constituent parsing approach by 4.84 (70.20 ¡ 65.36, $\ggg$) in $F1$-measure on automatic constituent parse trees, due to a big gain in recovering EEs of type *SBJ*.

Table 6. Performance of EE Recovery

Settings	EEs of *SBJ* + *OP*			EEs of *SBJ*			EEs of *OP*		
	Recall (%)	Precision (%)	$F1$	Recall (%)	Precision (%)	$F1$	Recall (%)	Precision (%)	$F1$
EE recovery on gold-standard constituent parse trees	99.23	94.55	96.83	98.87	99.49	99.18	99.83	87.33	93.16
EE recovery on automatic constituent parse trees	62.56	65.41	63.95	62.03	67.56	64.67	63.45	62.16	62.80
Joint constituent parsing	62.17	68.90	65.36	61.31	68.15	64.55	63.62	70.15	66.73
Gold-standard m-chunk + gold-standard DP	93.77	90.51	92.11	90.79	95.89	93.27	98.79	83.28	90.38
Gold-standard m-chunk + automatic DP	80.86	80.09	80.47	77.89	85.79	81.65	85.86	72.70	78.74
Automatic m-chunk + automatic DP	68.62	71.86	70.20	67.17	78.53	72.40	70.96	63.28	66.87

⑤http://svmlight.joachims.org/, Sept. 2013.

6.2 Contribution of EE Recovery on Syntactic Parsing

In this subsection, we feed the syntactic parser with automatically recovered EEs. Experimental results are shown in Table 7. In particular, we train the Charniak parser on data with explicit gold-standard EEs, test with automatically recovered EEs and evaluate with EEs stripped off. The third row in Table 7 gives the syntactic parsing performance with EEs recovered from automatic constituent parse trees. The fourth row presents the syntactic parsing of our joint constituent parsing approach. Finally, the last three rows show the syntactic parsing performance with EEs recovered by our chunk-based dependency parsing approach.

Table 7. Contribution of EE Recovery on Syntactic Parsing of Chinese

Settings	Recall (%)	Precision (%)	$F1$
Without EEs	79.74	81.95	80.83
With gold-standard EEs	83.98	85.72	84.84
EE recovery on automatic constituent parse trees	79.62	81.78	80.69
Joint constituent parsing	79.87	82.07	80.95
Gold-standard m-chunk + gold-standard DP	83.95	85.48	84.71
Gold-standard m-chunk + automatic DP	82.36	83.87	83.10
Automatic m-chunk + automatic DP	81.12	83.15	82.12

Table 7 shows that:

1) The syntactic parsing performance improves by 4.01 (84.84 ¡ 80.83, ⋙) in $F1$-measure with gold-standard EEs, indicating the critical role of EEs in syntactic parsing of Chinese. It is interesting to note that feeding the syntactic parser with automatic EEs recovered from automatic constituent parse trees slightly hurts the performance by 0.14 (80.83 ¡ 80.69), indicating the limitation of the EE recovery approach directly on constituent parse trees.

2) The joint constituent approach achieves a slight improvement with only 0.12 (80.95 ¡ 80.83, >) in $F1$-measure over traditional syntactic parsing. This may be due to the failure of a traditional syntactic parsing model to incorporate effective features related with EEs during the joint learning process.

3) Using automatic EEs from gold-standard m-chunks and gold-standard dependency relations achieves comparable performance with gold-standard EEs (84.71 ¡ 84.84). In particular, our EE recovery approach on chunk-based dependency parsing improves the syntactic parsing performance by 1.17 (82.12 ¡ 80.95, ⋙) in $F1$-measure over the joint constituent parsing approach, and by 1.29 (82.12 ¡ 80.83, ⋙) over the traditional syntactic parsing approach.

Fig.5 demonstrates an example where automatically recognized EEs help syntactic parsing. In Fig.5(a), NP (扬州市/Yangzhou City) is mistakenly parsed as subject of VP (沿运河分布/distribute along the canal) due to the high frequency of pattern IP ! NP + VP. Such syntactic parsing errors could be avoidable if EEs are explicitly recovered and fed into the parsing model, as shown in Fig.5(b).

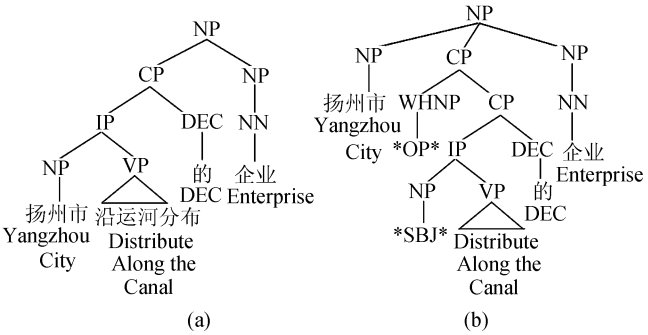

Fig.5. Example of parsing output with/without EEs. (a) Parsing output without EEs. (b) Parsing output with recognized EEs.

6.3 Discussion

In this part, we compare our work in methodology with the most related work by Cai *et al.*[17], an ACL 2011 short paper.

1) Cai *et al.*[17] encoded EE information in non-terminal nodes and relied on a syntactic parser to decode such EE information. Basically, the approach proposed in [17] is very similar to one of our adopted approaches, the joint constituent parsing approach. Our evaluation shows that such a language-independent approach can only slightly improve the performance of Chinese syntactic parsing (Table 7 in our paper: 80.95 vs 80.83).

2) Cai *et al.* provided little information on the impact of EEs in syntactic parsing of Chinese (Table 2 in [17]). They only provided the syntactic parsing performance after considering the EE information and no performance was given without considering the EE information.

3) The methods proposed in this paper are highly Chinese-specific since EEs in Chinese are quite different from EEs in English. We have discussed a lot throughout the paper on this issue. That is why a language-independent approach does not work well on employing EE information in syntactic parsing of Chinese.

4) In this paper, we regroup EEs in Chinese into four groups from the syntactic perspective. Our methods focus on *SBJ* (combined *T*, *PRO* and *pro*)

and *OP*(*op), which are closely related with syntactic parsing of Chinese and occupy 86% of all EEs (the majority instead of a small subset). We ignore the other two groups since they only occupy only 14% of all EEs and have little effect on syntactic parsing of Chinese with gold-standard ones. Please see Table 4 and Table 5.

5) It is worth noting that the reported numbers in [17] are not directly comparable to those in our paper, given different experimental settings in our paper and [17]. However, a quick comparison of Table 6 and Table 7 in our paper with Table 2 in [17] can easily see the superiority of our methodology.

7 Conclusions and Future Work

In this paper, we addressed EE recovery in Chinese, which plays a critical role and should deserve high attention in syntactic parsing of Chinese. In particular, two approaches were proposed to recover EEs and employ them to improve the performance of syntactic parsing of Chinese. Evaluation on CTB 5.1 shows the impact of our proposed approaches in both EE recovery and its application in syntactic parsing of Chinese, in particular the chunk-based dependency parsing approach. This is very promising and encouraging, considering the difficulty of syntactic parsing of Chinese.

Further examination on automatic syntactic parsing results shows that subject attachment errors and VP/IP coordination errors mostly affect the performance of EE recovery in Chinese. Inspired by previous work on dependency parsing of Chinese[27] and previous work on joint syntactic and semantic parsing of Chinese[25], we will extend our work by integrating the knowledge from a large amount of unlabeled data together with semantic information to further improve the performance of EE recovery and its application in syntactic parsing of Chinese. Besides, we will explore recovering EEs in a global optimization way[28] and representing an EE instance using a better structure[29].

References

[1] Marcus M P, Marcinkiewicz M A, Santorini B. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 1993, 19(2): 313-330.

[2] Collins M. Head-driven statistical models for natural language parsing [Ph.D. Thesis]. University of Pennsylvania, 1999.

[3] Charniak E. A maximum-entropy-inspired parser. In *Proc. the 1st North American Chapter of the Association for Computational Linguistics Conference*, April 2000, pp.132-139.

[4] Petrov S, Klein D. Improved inference for unlexicalized parsing. In *Proc. Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, April 2007, pp.404-411.

[5] Zhao S H, Ng H T. Identification and resolution of Chinese zero pronouns: A machine learning approach. In *Proc. the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, June 2007, pp.541-550.

[6] Kong F, Zhou G D. A tree kernel-based unified framework for Chinese zero anaphora resolution. In *Proc. the 2010 Conference on Empirical Methods in Natural Language Processing*, October 2010, pp.882-891.

[7] Kim Y J. Subject/object drop in the acquisition of Korean: A cross-linguistic comparison. *Journal of East Asian Linguistics*, 2000, 9(4): 325-351.

[8] Chung T, Gildea D. Effects of empty categories on machine translation. In *Proc. the 2010 Conference on Empirical Methods in Natural Language Processing*, October 2010, pp.636-645.

[9] Campbell R. Using linguistic principles to recover empty categories. In *Proc. the 42nd Annual Meeting of the Association for Computational Linguistics*, July 2004, pp.645-652.

[10] Guo Y Q, Wang H F, van Genabith J. Recovering non-local dependencies for Chinese. In *Proc. the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, June 2007, pp.257-266.

[11] Bikel D M. On the parameter space of generative lexicalized statistical parsing models [Ph.D. Thesis]. University of Pennsylvania, 2004.

[12] Johnson M. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proc. the 40th Annual Meeting of the Association for Computational Linguistics*, July 2002, pp.136-143.

[13] Dienes P, Dubey A. Antecedent recovery: Experiments with a trace tagger. In *Proc. the 2003 Conference on Empirical Methods in Natural Language Processing*, July 2003, pp.33-40.

[14] Dienes P, Dubey A. Deep syntactic processing by combining shallow methods. In *Proc. the 41st Annual Meeting of the Association for Computational Linguistic*, July 2003, pp.431-438.

[15] Yang Y Q, Xue N W. Chasing the ghost: Recovering empty categories in the Chinese TreeBank. In *Proc. the 23rd International Conference on Computational Linguistics,* August 2010, pp.1382-1390.

[16] Xue N W, Yang Y Q. Dependency-based empty category detection via phrase structure trees. In *Proc. the 2013 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, June 2013, pp.1051-1060.

[17] Cai S, Chiang D, Goldbery Y. Language- independent parsing with empty elements. In *Proc. the 49th Annual Meeting of the Association for Computational Linguistics*, June 2011, pp.212-216.

[18] Cahill A, Burke M, O'Donovan R, van Genabith J, Way A. Long-distance dependency resolution in automatically acquired wide-coverage pcfg-based LFG approximations. In *Proc. the 42nd Annual Meeting of the Association for Computational Linguistics*, July 2004, pp.319-326.

[19] Schmid H. Trace prediction and recovery with unlexicalized PCFGs and slash features. In *Proc. the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, July 2006, pp.177-184.

[20] Xue N W, Xia F. The bracketing guidelines for Penn Chinese Treebank project. Technical Report, IRCS 00-08, University of Pennsylvania.

[21] Finkel R J, Manning D C. Joint parsing and named entity recognition. In *Proc. the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, May 2009, pp.326-334.

[22] Nivre J. An efficient algorithm for projective dependency parsing. *In Proc. the 8th International Workshop on Parsing Technology*, April 2003, pp.149-160.
[23] Xue N W. Labeling Chinese predicates with semantic roles. *Computational Linguistics*, 2008, 34(2): 225-255.
[24] Li J H, Zhou G D, Zhao H, Zhu Q M, Qian P D. Improving nominal SRL in Chinese language with verbal SRL information and automatic predicate recognition. In *Proc. the 2009 Conference on Empirical Methods in Natural Language Processing*, August 2009, pp.1280-1288.
[25] Li J H, Zhou G D, Ng H T. Joint syntactic and semantic parsing of Chinese. In *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, July 2010, pp.1108-1117.
[26] Cohen P R. Empirical Methods for Artificial Intelligence. Cambridge, USA: MIT Press, 1995.
[27] Chen W L, Kazama J, Uchimoto K, Torisawa K. Improving dependency parsing with subtrees from auto-parsed data. In *Proc. the 2009 Conference on Empirical Methods in Natural Language Processing*, August 2009, pp.570-579.
[28] Zhou G D, Kong F. Learning noun phrase anaphoricity in coreference resolution via label propagation. *Journal of Computer Science and Technology*, 2011, 26(1): 34-44.
[29] Zhou G D, Zhu Q M. Kernel-based semantic relation detection and classification via enriched parse tree structure. *Journal of Computer Science and Technology*, 2011, 26(1): 45-56.

Guo-Dong Zhou received the Ph.D. degree in computer science from the National University of Singapore in 1999. He joined the Institute for Infocomm Research, Singapore, in 1999, and had been an associate scientist, scientist and associate lead scientist at the institute until August 2006. Currently, he is a distinguished professor at the School of Computer Science and Technology, Soochow University, Suzhou. His research interests include natural language processing, information extraction and machine learning. He has been a senior member of CCF since 2008 and a member of ACM and IEEE since 1999.

Pei-Feng Li received the Ph.D. degree in computer science from Soochow University, China in 2006. Currently, he is an associate professor at the university. His research interests include natural language processing and information extraction. He has been a member of CCF since 2008.