Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

A Clause-Level Hybrid Approach to Chinese Empty Element Recovery

Fang Kong[1,2] **and Guodong Zhou**[1*]
[1] School of Computer Science and Technology
Soochow University, China
[2] Department of Computer Science
National University of Singapore, Singapore
{kongfang, gdzhou}@suda.edu.cn

Abstract

Empty elements (EEs) play a critical role in Chinese syntactic, semantic and discourse analysis. Previous studies employ a language-independent sentence-level approach to EE recovery, by casting it as a linear tagging or structured parsing problem. In comparison, this paper proposes a clause-level hybrid approach to address specific problems in Chinese EE recovery, which recovers EEs in Chinese language from the clause perspective and integrates the advantages of both linear tagging and structured parsing. In particular, a comma disambiguation method is employed to improve syntactic parsing and help determine clauses in Chinese. In this way, the noise introduced by sentence-level syntactic parsing and multiple EEs in the same position of a linear sentence can be well addressed. Evaluation on Chinese Treebank 6.0 shows the significant performance improvement of our clause-level hybrid approach over the state-of-the-art sentence-level baselines, and its great impact on a state-of-the-art Chinese syntactic parser.

1 Introduction

Empty elements (EEs) are those nodes in parse trees which do not have corresponding surface words or phrases. Many treebanks include EEs in parse trees to represent non-local dependencies or dropped elements. The use of EEs in the annotation of treebanks begins with the Penn Treebank [Marcus *et al.*, 1993], and this practice continues in the Chinese Treebank (CTB) [Xue *et al.*, 2005]. Nevertheless, most of related studies on these resources ignore EEs. It is only recently that EEs have been drawing some attention. Representative studies include Campbell [2004], Dienes and Dubey [2003], Johnson [2002], and Gabbard et al. [2006]. While the first is rule-based, the others are learning-based. Specifically, Dienes and Dubey [2003] integrate EE recovery into a parser while Johnson [2002] and Gabbard et al. [2006] recover EEs from a parser output in a post-processing step.

Although EEs exist in many languages and serve different purposes, they are particularly important for pro-dropped languages, such as Chinese, where subjects and objects are frequently dropped to keep a discourse concise. [Kim, 2000] compares the use of overt subjects in both English and Chinese, and finds that overt subjects occupy over 96% in English, while this percentage drops to only 64% in Chinese. However, related studies on recovering Chinese EEs are basically language independent, either adapting an existing approach from English to Chinese [Guo *et al.*, 2007; Yang and Xue, 2010; Chung and Gildea, 2010] or proposing a general approach for both English and Chinese [Cai *et al.*, 2011]. Since Chinese is a discourse-driven pro-dropped language and has more varieties of EEs than English, definitions or schemes that work for English may not work well for Chinese. Moreover, previous studies on recovering Chinese EEs employ a sentence-level approach, either adopting a linear view of a sentence as a word sequence (i.e., linear tagging) or a structured view of sentence as a parse trees (i.e., structured parsing). However, such sentence-level approach has its inherent problems. On one hand, while a linear tagging method on EE recovery is easy to implement and fast in running speed, some EEs in the multi-EE positions will be missed inevitably. Our statistics on CTB 6.0 shows that about 30% of EE positions have more than one EE. In comparison, 95% of those multi-EE positions disappear if viewed from the clause perspective. On the other hand, while a structured parsing method recovers EEs from the parse tree and can well address those multi-EE positions, it heavily depends on the performance of syntactic parsing. This largely affects the application of the structured parsing method to EE recovery in Chinese language, since in comparison with English syntactic parsing, the performance of Chinese syntactic parsing is generally much lower (about 8-10 in F-measure).

In this paper, we propose a clause-level hybrid approach to address above problems in Chinese EE recovery, which recovers EEs in Chinese language from the clause perspective and integrates the advantages of both linear tagging and structured parsing. In this way, the noise introduced by sentence-level syntactic parsing and multiple EEs in the same position of a linear sentence can be well addressed. Evaluation on CTB 6.0 justifies the effectiveness of our clause-level hybrid approach over the state-of-the-art sentence-level baselines.

The rest of this paper is organized as follows: Section 2 introduces some background knowledge on EEs in Chinese and its possible impact on Chinese syntactic parsing. Sec-

*Corresponding author

ID	Type	occur	Description
1	*T*	4486	Used in topicalization and object preposing constructions (trace of movement)
2	*	132	Used in raising and passive constructions (trace of A-movement)
3	*PRO*	2856	Used in control structures and cannot be substituted by an overt constituent
4	*pro*	2024	Used for dropped subjects or objects
5	*RNR*	217	Used for right node raising
6	*OP*	879	Empty operators in relative constructions

Table 1: EE types and their distribution in CTB 6.0

tion 3 briefly overviews the related work. Section 4 describes two baseline sentence-level approaches of linear tagging and structured parsing. Section 5 presents our clause-level hybrid approach. Finally, we conclude our work in Section 6.

2 Background

To better understand this paper, we introduce some background knowledge on EEs in Chinese and its possible impact on Chinese syntactic parsing.

2.1 Empty Elements in Chinese

Table 1 shows different types of EEs in CTB 6.0, 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 43%, type *PRO* occupies about 27%, type *pro* occupies about 19% while the other three types occupy only about 11% totally.

Following illustrates examples of different EE types from CTB 6.0. For details, please refer to the CTB bracketing manual [Xue and Xia, 2000].

(1) 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.)

(2) 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.)

2.2 Impact of EEs on Syntactic Parser - Preliminary Experimentation

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

		R(%)	P(%)	F
CTB	without EEs	81.14	84.06	82.57
	with Gold EEPs	86.00	88.43	87.20
	with Gold EEPTs	85.48	88.10	86.77
PTB	without EEs	88.88	89.46	89.17
	with Gold EEPs	90.15	90.39	90.27
	with Gold EEPTs	90.24	90.59	90.42

Table 2: Impact of EEs on syntactic parsing of both Chinese and English (EEPs:EE Positions; EEPTs: EE Positions and Types)

We train the Berkeley parser [Petrov *et al.*, 2006] on both PTB (in English) and CTB (in Chinese) using the widely adopted splitting (i.e., On PTB, sections 2-21 selected as the training data, section 23 held out as the test data, and section 24 used as the development data. On CTB, 648 files (chtb 0081 to 0899.fid) for training, 40 files (chtb 0041 to 0080.fid) for development, and 72 files (chtb 0001 to 0040.fid and chtb 0900 to 0931.fid) for testing).

Table 2 shows the impact of EEs on syntactic parsing of both Chinese and English under different experimental settings. In the last two experimental settings (with Gold EEPs and with Gold EPTs), EEs in outputs are stripped before evaluating the syntactic parsing performance. From the results we can find that:

- Given gold EE positions, the syntactic parser performance improves by about 4.63 and 1.1 in F-measure on CTB and PTB, respectively.
- Further consideration of gold EE types has much less impact on syntactic parsing of both Chinese and English languages.
- In both settings of with gold EEPs and with gold EEPTs, the impact of EEs on Chinese parsing is much more significant than English.

Due to the significant importance of EE positions over EE types on syntactic parsing, this paper focuses on recovering EE positions in a parse tree without attempting to determine their specific EE types by treating all the EEs as a unified type. Instead, corresponding EE types are determined in a post-processing stage according to the context around.

3 Related Work

Although EEs have been an integral part of syntactic representation of a sentence ever since the Penn Treebank was first constructed, it is only recently that they begin to receive some deserved attention.

For English EE recovery, representative studies can be classified into rule-based [Campbell, 2004] and learning-based [Johnson, 2002; Dienes and Dubey, 2003; Gabbard *et al.*, 2006].

For Chinese EE recovery, all the studies mainly adapt existing methods for the English language to the Chinese language [Guo *et al.*, 2007; Yang and Xue, 2010; Chung and Gildea, 2010] or proposing a general approach for both English and Chinese [Cai *et al.*, 2011] from the sentence perspective. Such a sentence-level approach can be classified into two types: linear tagging and structured parsing.

For linear tagging, the representative study is [Yang and Xue, 2010]. Similar to Dienes and Dubey [2003], they treat Chinese EE recovery as a linear tagging problem and propose a unified framework to combine lexical and constituent-based syntactic information. They find that given skeletal gold standard parse trees, EEs can be detected at the performance of about 89.0 in F-measure. However, for automatic parse trees, the performance drops dramatically by about 25% in F-measure. This may be due to that they recover EEs on sentence level, scanning every word of a sentence and determining whether there is an EE in a specific position (i.e. before the word). However, our statistics on CTB 6.0 shows that on sentence level, about 30% of EE positions have more than one EE. Therefore, applying the linear tagging method on sentence level will definitely miss some EEs in those multi-EE positions. To avoid this problem, we consider EE recovery on clause level.

For structured parsing, representative studies include [Guo *et al.*, 2007; Chung and Gildea, 2010; Cai *et al.*, 2011]. Among them,

[Guo *et al.*, 2007] extend the work of Cahill et al. [2004] based on various kinds of LFG (Lexical-Functional Grammar) f-structures and achieves the performance of 64.7 in F-measure on trace insertion.

[Chung and Gildea, 2010] employ several methods for both Korean and Chinese EE recovery, and apply them to machine translation. In particular, they extend the work of Johnson [2002] by including various kinds of contexts into the minimally-connected tree fragments as patterns and applying extracted patterns to recover two types of EEs (*PRO* and *pro*). Although EE recovery in both Korean and Chinese is still not satisfactory (e.g. with the performances of 63.0 and 44.0 in F-measure on Chinese EE recovery for *PRO* and *pro* respectively), it nevertheless improves the end translation performance by about 0.96 in BLEU.

[Cai *et al.*, 2011] present a simple language-independent method for integrating EE recovery into syntactic parsing. They take a state-of-the-art parsing model, the Berkeley parser [Petrov *et al.*, 2006], train it on data with explicit EEs, and test it on word lattices that can non-deterministically insert EEs anywhere. Evaluation on CTB 6.0 shows that they achieve the performance of 67.0 in F-measure on Chinese EE recovery, which represents the state-of-the-art.

However, all these studies do not report the influence of Chinese EE recovery on Chinese syntactic parsing.

In this paper, we propose a clause-level hybrid approach to EE recovery in Chinese language. In particular, 1) a hybrid clause-level approach is proposed to integrate the advantages of both linear tagging and structured parsing; 2) considering the specificity of Chinese commas, we incorporate comma disambiguation to improve syntactic parsing and help determine clauses in Chinese; 3) we apply a binary classifier to filter out the abundance of clause instances without EEs to address the class imbalance problem.

4 Baselines: A Sentence-level Approach

In this section, we present two baselines using a sentence-level approach: one adopts a linear tagging method and the other adopts a structured parsing method.

ID	Type	Description
1	*obj	Object (e.g. dobj,iobj,pobj and so on)
2	*subj	Subject (e.g. nsubj,csubj and so on)
3	*mod	Modifider (e.g. amod, advmod and so on)
4	nn	Noun compound modifier
5	conj	Coordinating conjunction
6	dep	Unknown

Table 3: Dependency relationships closely related with EEs

4.1 Baseline 1: A Linear Tagging Method

Similar to Yang and Xue [2010], our sentence-level linear tagging baseline adopts a linear view of a sentence as a word sequence, treating EEs the same as overt word tokens, and casts EE recovery as a classification task. During training, if an EE candidate has a counterpart in the same position in the standard corpus, a positive instance is generated. Otherwise, a negative instance is generated. During testing, each EE candidate is presented to the learned EE detector to determine whether it is a true EE or not.

At first glance, it seems that an EE can occur before any word tokens. Fortunately, an exploration of the CTB 6.0 corpus shows that EEs have close relationship with predicates and subordinate compounds. Therefore, some simple collocations and patterns can be applied to filter out those unlikely locations in generating EE candidates.

Having cast EE recovery as a classification task, choosing an appropriate set of features becomes crucial. We first extract the same set of 11 lexical features and 8 syntactic features as basic features, as described in Yang and Xue [2010]. Considering the importance of dependency relations between words, we further explore a corpus-based scheme to extract useful dependency features.

Firstly, we get various kinds of dependency relations using gold parse trees with EEs.

Secondly, we conduct statistic analysis and get the list of dependency relations which closely relate with EEs.

Finally, all the dependency relations related with current word are used to construct a feature vector as its representation. Table 3 shows the dependency relationships closely related with EEs.

4.2 Baseline 2: A Structured Parsing Method

Our sentence-level structured parsing baseline extracts all the constituents from the given parse tree of a sentence, scans each constituent and determines whether there is an EE in the specific position of the parse tree (i.e. before the considered constituent).

Similar to the linear tagging method, our structured parsing method also casts EE recovery as a classification task. During training, if there is an EE before a specific constituent in the standard corpus, a positive instance is generated. Otherwise, a negative instance is generated. During testing, each constituent is presented to the learned EE detector to determine whether there is an EE before or not.

For every instance, we view the constituent as a special ‘ word token ’ and extract the features similar to the way in the linear tagging method.

Systems		R(%)	P(%)	F
Our Baseline 1: (Linear Tagging)	Gold	78.5	98.4	87.3
	Auto	54.7	82.7	65.8
Our Baseline 2: (Structured Parsing)	Gold	87.2	90.3	88.7
	Auto	62.7	75.1	68.3
Yang and Xue [2010]: (Linear Tagging)	Gold	83.0	95.9	89.0
	Auto	52.1	80.3	63.2
Cai et al. [2011]: (Structured Parsing)	Gold	-	-	-
	Auto	61.3	74.0	67.0

Table 4: Performance comparison on EE recovery

4.3 Results and Analysis

For comparison with previous studies [Yang and Xue, 2010; Cai *et al.*, 2011], we use the same splitting of CTB 6.0 in all our experiments. This splitting is also adopted in section 2.2 for syntactic parsing. Similar to Yang and Xue [2010], we evaluate our baselines under two experimental settings: 1) with gold standard parse trees from CTB 6.0 (with EEs stripped off); and 2) with automatic parse trees produced by the Berkeley parser (with gold word segmentation and the performance of 82.57 in F-measure). In addition, we use the SVM-light toolkit with the radial basis kernel and default learning parameters. For evaluation, we use precision, recall and F-measure. To see whether an improvement is significant, we conduct significance testing using paired t-test.

Table 4 shows the performance of our two baselines using the sentence-level approach and compares the performance of our baselines with two state-of-the-art ones on both gold standard parse trees and automatic parse trees. It shows that:

- Compared with gold parse trees, the performance of our linear tagging baseline drops significantly by 15.7%, 23.8% and 21.5 in precision, recall and F-measure respectively when using automatic parse trees, while our structured parsing baseline drops significantly by 15.2%, 24.5% and 20.4 in precision, recall and F-measure respectively when using automatic parse trees. This suggests the heavy dependency of EE recovery on the performance of a syntactic parser.
- On both gold and automatic parse trees, our linear tagging baseline achieves better precision than our structured parsing baseline, and our structured parsing baseline achieves better recall than our linear tagging baseline. This is due to that while the structured parsing baseline can resolve the multi-EEs problem and achieve better recall, it encounters more serious class imbalance problem and results in lower precision. In our linear tagging baseline, the ratios of negative and positive instances are 13.5:1, when using automatic parse trees, and 11.3:1, when using gold parse trees. In comparison, the ratios increase to 18.2:1 and 15.7:1, respectively, in our structured parsing baseline.
- Using the similar liner tagging method, our baseline 1 performs a bit lower than Yang and Xue [2010] by 1.7 in F-measure on gold parse trees, while on automatic parse trees, our baseline 1 achieves better performance than Yang and Xue [2010] by 2.7 in F-measure, partially due to the employment of dependency relations.
- Using the similar structured parsing method, our baseline 2 achieves better performance than Cai et al. [2011] by 1.3 in F-measure, partially due to the employment of more rich syntactic features.
- In general, structured parsing methods perform better than linear tagging methods. This suggests the importance of addressing the multi-EEs problem in Chinese EE recovery.

5 A Clause-level Hybrid Approach

In this section, we present a clause-level hybrid approach to Chinese EE recovery, which recovers EEs in Chinese language from the clause perspective and integrates the advantages of both linear tagging and structured parsing.

First, a simplified semantic role labeling (SRL) framework is adopted to determine the clauses from a parse tree. In particular, a comma disambiguation method is proposed to improve syntactic parsing. In this paper, a clause is defined as the minimal sub-tree governed by a predicate and all its arguments.

Then, a hybrid approach is employed to recover EEs on clause level from bottom up. Here, we view the clauses in a parse tree hierarchically in a bottom-up way. For the terminal clauses (i.e., not containing sub-clauses), we adopt a linear view and conduct EE recovery using a linear tagging method. For non-terminal clauses, all the sub-clauses, which have been resolved, are viewed as an inseparable ‘ constituent ’ so that each of these non-terminal clauses can be viewed linearly and a linear tagging method can be applied recursively. In this way, the advantages of both linear tagging and structured parsing can be well integrated.

For example, in Chinese sentence S:

尽管 浦东 新区 制定 的 法规性 文件 有些 比较 “粗”， 有些 还 只是 暂行 规定， 有待 在 实践 中 逐步 完善， 但 这种 法制 紧跟 经济 和 社会 活动 的 做法， 受到 了 国内外 投资者 的 好 评， 他们 认为， 到 浦东 新区 投资 办事 有 章法， 讲 规矩， 利益 能 得到 保障 。 *(In spite of the fact that of the regulatory documents that the Pudong new region has formulated, some are relatively ” crude ” and some are still only provisional regulations awaiting step-by-step completion as they are put into practice, nevertheless, this kind of approach, with the legal system tightly coupled with economic and social activities, has received positive comments from domestic and foreign investors. They believe that in coming to the Pudong new region to invest there is methodicalness and attention to rules in the handling of business, and interests can receive safeguards.)*

We can have following terminal clauses, to which a linear tagging method is adopted.

- 0-A [浦东 新区 制定 的 法规性 文件]
- 0-B [在 实践 中 逐步 完善]
- 0-C [法制 紧跟 经济 和 社会 活动]
- 0-D [他们 认为， 到 浦东 新区 投资 办事 有 章法， 讲 规矩， 利益 能 得到 保障]

From bottom up, we can have following non-terminal clauses, to which a linear tagging method can be adopted again.

- 1-A [0-A 有些 比较 “ 粗 ”]
- 1-B [但 这 种 0-C 的 做法 ， 受到 了 国内外 投资者 的 好 评]
- 2-A [1-A ， 有些 还 只 是 暂行 规定]
- 3-A [2-A ， 有待 0-B]

Obviously, the key to our clause-level hybrid approach is how to determine clauses effectively.

5.1 Clause Determination in Chinese

Given the definition of clause that every clause should correspond to an independent predicate and all its arguments, we can determine clauses from a parse tree by extracting all the predicates and related arguments using a shallow semantic parser, e.g. a semantic role labeling (SRL) toolkit. The problem is that the performance of Chinese SRL heavily depends on the performance of syntactic parsing. Fortunately, clause determination only wants to know the sub-trees corresponding to the predicate argument structure, instead of recognizing the extract predicate argument structure. Furthermore, previous research on SRL shows that given a predicate, its semantic arguments are usually siblings of the predicate or siblings of its ancestor. That is, the generation of the predicate argument structure is only associated with partial parse tree instead of full parse tree.

Motivated by these observations, we propose a simplified SRL framework to clause determination, including predicate recognition, argument pruning and argument identification, to extract the clause sub-trees. For more details, please refer to Li et al. [2009]. With a recognized predicate and corresponding identified arguments, we can easily get the minimal governing sub-tree and retrieve its content as a clause.

5.2 Comma Disambiguation in Chinese

Error statistics of above clause determination framework on the development data shows that about 11.4% of errors are due to frequently-occurring ambiguous commas in signaling sentence (or clause) boundaries, a special characteristic in Chinese language. This is largely due to that Chinese people tend to use commas instead of periods in many kinds of contexts in signaling the boundary of a sentence.

Similar to Xue and Yang [2011], we classify commas in Chinese into two categories: EOS (end of a sentence), non-EOS (not the end of a sentence) and train a binary classifier for comma disambiguation. Besides the 11 basic features, as described in Xue and Yang [2011], we further add following features, considering some specific patterns (e.g., 把/BA, 被/BEI and so on) and dependency relations across the comma:

- whether 把/BA or 被/BEI appears
- whether 的/DIE,得/DE,地/DI appears
- whether a localizer appears
- whether the last word is a localizer
- whether the last word is a noun following 的/DIE
- the number of dependency lines cross over the comma

Systems	R(%)	P(%)	F
Our CLH	63.2	84.6	72.4
CLH with filtering	65.7	86.4	74.6

Table 5: Performance using clause-level hybrid approach on automatic parse trees (CLH: Clause-level hybrid approach)

Our comma disambiguation classifier is trained and evaluated on the same data corpus as Xue and Yang [2011], with automatic parse trees produced by the Berkeley parser using the model achieved in subsection 2.2. Evaluation shows that our classifier achieves the overall accuracy of 92.8% with 85.4%, 71.9% and 78.1% in precision, recall and F1 score respectively for EOS commas, and 93.6%, 97.1% and 95.3% in precision, recall and F1 score respectively for non-EOS commas. In comparison, Xue and Yang [2011] achieves the overall accuracy of 89.2%. Obviously, our comma disambiguation model achieves much better performance.

Given the comma disambiguation classifier, clause determination is refined as follows.

First, we do comma disambiguation for every comma in both the training and test data sets.

Then, we retrain the Berkeley parser on new sentence boundaries (ignoring higher level boundaries).

Finally, using the retrained parsing model, we parse the test set on new sentence boundaries and adopt the simplified SRL framework to determine clauses for the parse tree.

5.3 Results and Analysis

For fair comparison, all our experiments in this subsection have been done using the same experimental settings as our baseline system.

EE recovery in Chinese

Table 5 shows the experimental results employing our clause-level hybrid approach. We can find that compared with our sentence-level linear tagging baseline, our clause-level hybrid approach improves the performance of EE recovery by 6.5 in F-measure, largely due to a 8.5% increase in recall, while compared with our sentence-level structured parsing baseline, our clause-level hybrid approach achieves better performance by 4.1 in F-measure, largely due to a 9.5% increase in precision. This justifies the appropriateness of addressing EE recovery from clause level and the effectiveness of our clause-level hybrid approach in recovering multiple EEs in a specific position of a linear sentence and reducing the dependency on syntactic parsing.

As stated in subsection 4.3, one problem in EE recovery lies in the class imbalance, the high ratio of clauses without EEs and EEs. To address this problem, we adopt a simple but effective binary classifier to filter out those clauses without EEs as many as possible in the preprocessing stage, using various kinds of dependency relationships (shown in Table 3) as features. Specifically, we employ the Stanford dependency parser to extract various dependencies between individual words in a given clause and keep a dependency if a governor and its dependent word are both covered by the given clause. In this way, we can get a list of dependencies related with the given clause.

Table 5 shows that the filter improves the performance by 2.2 in F-measure due to the increase in both precision and

Automatic EE Recovery System	R(%)	P(%)	F
Baseline 1 (Linear Tagging)	82.31	85.28	83.77
Baseline 2 (Structured Parsing)	84.38	84.57	84.47
CLH System with filtering	84.79	86.12	85.45

Table 7: The impact of automatic EE recovery on syntactic parsing of Chinese (CLH: Clause-level hybrid approach)

recall. With the filter, the ratio of clauses without EEs and EEs is reduced half to about 6.8:1 with only 3.0% of positive instances wrongly filtered.

For reference, Table 6 shows the performance of our final clause-level hybrid approach (i.e. with filtering) for different types of EEs using automatic parse trees. Similar to [Yang and Xue, 2010] and [Cai *et al.*, 2011], we avoid predicting specific types of EEs and only give the percentage of EEs our model recovers for each type. Table 6 indicates the advantage of our clause-level approach over the sentence-level approach in all the three major EE types (*PRO*, *pro* and *T*). Since the remaining types only occupy about 10%, the performance on recovering them can be simply ignored.

Impact of EE recovery on syntactic parsing of Chinese

Impact of EE recovery on syntactic parsing of Chinese For the impact of automatic EE recovery on syntactic parsing of Chinese. we integrate Chinese EE recovery into a state-of-the-art syntactic parsing model, the Berkeley parser [Petrov *et al.*, 2006], with gold word segmentation and the same training/development/test data splitting on CTB6.0 as adopted in our previous experiments, by training it on data with explicit gold EE positions, testing on automatically recovered EEs and evaluating with automatically recovered EEs stripped off. Table 7 shows that EE recovery using our sentence-level linear tagging and structured parsing baselines much improves the performance of the Berkeley parser from 82.57 to 83.77 and 84.47 in F-measure, respectively, indicating the superiority of structured parsing over linear tagging. It also shows that EE recovery using our final clause-level hybrid approach improves the performance of the Berkeley parser from 82.57 to 85.45 in F-measure, indicating the superiority of a clause-level approach over a sentence-level approach. Although there is still a big gap to the performance of 87.2 in F-measure when the positions of the gold EEs are known to the Berkeley parser, our study steps a big stride towards the right direction.

6 Conclusion and Further Work

In this paper, we present a clause-level hybrid approach to Chinese EE recovery. Our contributions include:

- the proposal of a clause-level approach to EE recovery. Evaluation shows that it is better to recover EEs from clause level instead of traditional sentence level, due to clause ' s noisy robustness and the higher possibility of recovering multiple EEs in a specific position of a sentence from clause level.
- the proposal of a hybrid method to integrate the advantages of linear tagging and structured parsing.
- the proposal of a simplified SRL framework to clause determination with the help of comma disambiguation.

For future work, we will explore better ways of recovering EEs from clause level and integrating EE recovery in syntactic, semantic and discourse analysis.

Acknowledgments

This research is supported by Projects 61003153, 61273320 and 61272257 under the National Natural Science Foundation of China, Project 2012AA011102 under the National 863 Program of China, Project 11KJA520003 under the Natural Science Major Fundamental Research Program of the Jiangsu Higher Education Institutions. This research is also partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

References

[Cahill *et al.*, 2004] Aoife Cahill, Michael Burke, Ruth O'Donovan, Josef Van Genabith, and Andy Way. Long-distance dependency resolution in automatically acquired wide-coverage pcfg-based lfg approximations. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 319–326, Barcelona, Spain, July 2004.

[Cai *et al.*, 2011] Shu Cai, David Chiang, and Yoav Goldberg. Language-independent parsing with empty elements. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 212–216, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Campbell, 2004] Richard Campbell. Using linguistic principles to recover empty categories. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 645–652, Barcelona, Spain, July 2004.

[Chung and Gildea, 2010] Tagyoung Chung and Daniel Gildea. Effects of empty categories on machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 636–645, Cambridge, MA, October 2010. Association for Computational Linguistics.

[Dienes and Dubey, 2003] Péter Dienes and Amit Dubey. Antecedent recovery: Experiments with a trace tagger. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 33–40, 2003.

[Gabbard *et al.*, 2006] Ryan Gabbard, Seth Kulick, and Mitchell Marcus. Fully parsing the penn treebank. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 184–191, New York City, USA, June 2006. Association for Computational Linguistics.

[Guo *et al.*, 2007] Yuqing Guo, Haifeng Wang, and Josef van Genabith. Recovering non-local dependencies for Chinese. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing*

ID	EE Type	Total	Correctly Recovered (Recall %)				
			[Yang and Xue, 2010] (Linear Tagging)	[Cai *et al.*, 2011] (Structured Parsing)	Our Baseline 1 (Linear Tagging)	Our Baseline 2 (Structured Parsing)	Clause-level Hybrid Approach with filtering
1	*T*	578	338(58.5)	388(67.1)	301(52.1)	378(65.4)	380(65.7)
2	*pro*	290	125(43.1)	159(54.5)	181(62.4)	178(61.4)	224(77.2)
3	*PRO*	299	196(65.6)	199(66.6)	216(72.2)	220(73.6)	229(76.6)
4	*RNR*	32	20(62.5)	15(46.9)	4(12.5)	5(15.6)	4(12.5)
5	*OP*	134	20(14.9)	65(48.5)	35(26.1)	62(46.3)	48(35.8)
6	*	19	5(26.3)	3(15.8)	3(15.8)	5(26.3)	3(15.8)
Overall		1352	704(52.1)	829(61.3)	740(54.7)	848(62.7)	888(65.7)

Table 6: Performance comparison of different types of EEs on EE recovery using automatic parse trees

and Computational Natural Language Learning (EMNLP-CoNLL), pages 257–266, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[Johnson, 2002] Mark Johnson. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 136–143, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[Kim, 2000] Young-Joo Kim. Subject/object drop in the acquisition of korean: A cross-linguistic comparison. *Journal of East Asian Linguistics*, 9:325–351, 2000.

[Li *et al.*, 2009] Junhui Li, Guodong Zhou, Hai Zhao, Qiaoming Zhu, and Peide Qian. Improving nominal SRL in Chinese language with verbal SRL information and automatic predicate recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1280–1288, Singapore, August 2009. Association for Computational Linguistics.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

[Petrov *et al.*, 2006] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July 2006. Association for Computational Linguistics.

[Xue and Xia, 2000] Nianwen Xue and Fei Xia. The bracketing guidelines for penn chinese treebank project. Technical report, Pennsylvania, 2000.

[Xue and Yang, 2011] Nianwen Xue and Yaqin Yang. Chinese sentence segmentation as comma classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 631–635, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Xue *et al.*, 2005] Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11:207–238, 2005.

[Yang and Xue, 2010] Yaqin Yang and Nianwen Xue. Chasing the ghost: recovering empty categories in the chinese treebank. In *Coling 2010: Posters*, pages 1382–1390, Beijing, China, August 2010. Coling 2010 Organizing Committee.