Employing Constituent Dependency Information for Tree Kernel-Based Semantic Relation Extraction between Named Entities

LONGHUA QIAN, GUODONG ZHOU, and QIAOMING ZHU, Soochow University

This article proposes a new approach to dynamically determine the tree span for tree kernel-based semantic relation extraction between named entities. The basic idea is to employ constituent dependency information in keeping the necessary nodes and their head children along the path connecting the two entities in the syntactic parse tree, while removing the noisy information from the tree, eventually leading to a dynamic syntactic parse tree. This article also explores various entity features and their possible combinations via a unified syntactic and semantic tree framework, which integrates both structural syntactic parse information and entity-related semantic information. Evaluation on the ACE RDC 2004 English and 2005 Chinese benchmark corpora shows that our dynamic syntactic parse tree much outperforms all previous tree spans, indicating its effectiveness in well representing the structural nature of relation instances while removing redundant information. Moreover, the unified parse and semantic tree significantly outperforms the single syntactic parse tree, largely due to the remarkable contributions from entity-related semantic features such as its type, subtype, mention-level as well as their bi-gram combinations. Finally, the best performance so far in semantic relation extraction is achieved via a composite kernel, which combines this tree kernel with a linear, state-of-the-art, feature-based kernel.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Information extraction*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Semantic relation extraction, convolution tree kernel, constituent dependency, unified syntactic and semantic tree

ACM Reference Format:
Qian, L., Zhou, G., and Zhu, Q. 2011. Employing constituent dependency information for tree kernel-based semantic relation extraction between named entities. ACM Trans. Asian Lang. Inform. Process. 10, 3, Article 15 (September 2011), 24 pages.
DOI = 10.1145/2002980.2002985 http://doi.acm.org/10.1145/2002980.2002985

1. INTRODUCTION

As one of the key tasks in the field of Natural Language Processing (NLP), information extraction (IE) attempts to identify relevant information from a large amount of text documents in the form of natural language and put them in a structured format. The research on IE was first initiated by the Message Understanding Conferences [MUC

This research is supported by Projects 60873150, 60970056 and 90920004 under the National Natural Science Foundation of China, Project BK2010219 under the Provincial Natural Science Foundation of Jiangsu, China.
Authors' addresses: L. Qian, NLP Lab, School of Computer Science and Technology, Soochow University, Suzhou, China; email: qianlonghua@suda.edu.cn; G. Zhou, NLP Lab, School of Computer Science and Technology, Soochow University, Suzhou, China; email: gdzhou@suda.edu.cn; Q. Zhu (corresponding author), NLP Lab, School of Computer Science and Technology, Soochow University, Suzhou, China; email: qmzhu@suda.edu.cn.

DOI 10.1145/2002980.2002985 http://doi.acm.org/10.1145/2002980.2002985

1987–1998] and then further promoted significantly by the NIST Automatic Context Extraction [ACE 2002–2008] program. Of the three subtasks in information extraction defined by the NIST ACE program, namely Entity Detection and Tracking (EDT), Relation Detection and Characterization (RDC), and Event Detection and Characterization (EDC), this article focuses exclusively on the RDC subtask, which detects and classifies semantic relationships between predefined types of entities previously recognized by the EDT subtask in the ACE corpus. The RDC subtask is often referred to as semantic relation extraction, or more shortly as relation extraction. For example, the sentence "*Microsoft Corp.* is based in *Redmond*, WA" conveys the relation of "GPE-AFF.Based" between "*Microsoft Corp.*" with the entity type of "ORG" and "*Redmond*" with the entity type of "GPE". In addition to information extraction itself, relation extraction is also very useful in many advanced NLP applications, such as question answering and text summarization. However, due to limited accuracy in the current state-of-the-art syntactic and semantic parsing as well as the complexity and variability of the semantic relationships, reliably extracting such semantic relationship between named entities from natural language documents is still a difficult, unresolved problem.

In the literature, feature-based methods have ever dominated the research in relation extraction [Jiang and Zhai 2007; Kambhatla 2004; Zhao and Grishman 2005; Zhou et al. 2005], by first transforming relation examples into the corresponding numerical vectors of various syntactic and semantic features and then applying a machine learning approach (such as SVMs or maximum entropy models) to detect and classify them into predefined types of semantic relationships between named entities. The features used include lexical items, phrase and chunk information, syntactic parse trees, deep semantic information and entity-related information. However, the featured-based methods usually fail to effectively capture the critical structural information inherent in the parse trees. Detailed research [Jiang and Zhai 2007; Zhou and Zhang 2007; Zhou et al. 2005] shows that it is quite difficult for the feature-based methods to extract new effective features to further improve the extraction accuracy. Therefore, researchers turn to kernel-based methods, which attempt to avoid the burden of feature engineering through directly computing the similarity between any two discrete objects, like parse trees with rich structural information. From prior work [Bunescu and Mooney 2005; Culotta and Sorensen 2004; Zelenko et al. 2003] to current research [Nguyen et al. 2009; Zhang et al. 2006; Zhou et al. 2007, 2009], kernel-based methods have been showing more and more potential in relation extraction due to its effectiveness in modeling discrete objects, for example, capturing the structural information in parse trees.

Among the first to employ the kernel-based methods for relation extraction, Zelenko et al. [2003] achieves surprisingly good performance on two simple tasks using the shallow parse tree. For the more challenging ACE task in relation extraction, kernel-based methods using the dependency tree [Culotta and Sorensen 2004] and the shortest dependency path [Bunescu and Mooney 2005] exhibit relatively high precision. However, they suffer from quite low recall in performance. Thanks to the pioneering work in Semantic Role Labeling (SRL) by Moschitti [2004], convolution tree kernels over syntactic parse trees [Collins and Duffy 2001] are successfully adopted by Zhang et al. [2006] and Zhou et al. [2007] to relation extraction and achieve comparable or even better performance than feature-based ones, largely due to their distinctive merit in effectively capturing the structural information in relation instances. This article focuses on relation extraction using the convolution tree kernels.

Given the convolution tree kernels, the key problem for the kernel-based methods on relation extraction is how to properly represent the structural information inherent in relation instances. In this respect, Zhang et al. [2006] investigates five kinds

of tree spans and discover that the simple Shortest Path-enclosed Tree (SPT) achieves the best performance. Zhou et al. [2007] further extend the SPT to the more general Context-Sensitive Shortest Path-enclosed Tree (CS-SPT), which dynamically includes necessary predicate-linked path information beyond the SPT. The problem with both SPT and CS-SPT is that they may still contain unnecessary information. For example, in the sentence "... bought *one* of town's two meat-packing *plants*", the underlined part in the SPT/CS-SPT is unnecessary for determining the relationship between two entities "*one*" and "*plants*". Moreover, a great deal of useful context-sensitive information may be wrongly removed from SPT/CS-SPT, even though the CS-SPT already includes some contextual information related to the predicate-linked path. For example, in the same sentence "... bought *one* of *town*'s two meat-packing plants", the underlined part, which is removed from the SPT/CS-SPT, is critical for determining the relationship between two entities "*one*" and "*town*". Similar phenomena also exist when determining semantic relationships in Chinese narratives.

To address the issues mentioned above, this article proposes a new approach to dynamically determine the tree span for relation extraction by exploiting constituent dependencies to remove the noisy information, as well as to keep the necessary information in the parse tree. The motivation is to properly utilize linguistic dependency knowledge in constructing a concise and effective tree span, specifically targeted for relation extraction. Moreover, various kinds of entity-related semantic information are explored via a unified parse and semantic tree framework.

The layout of the rest article is organized as follows. Section 2 gives a brief overview of related work. Section 3 proposes the dynamic syntactic parse tree while the unified syntactic and semantic tree with integrated entity-related semantic information is presented in Section 4. Section 5 reports the experimental results on various ACE RDC corpora. Finally, we conclude our work and indicate some future work in Section 6.

2. RELATED WORK

The task of relation extraction was first envisioned as part of the Template Element task in MUC-6 [Grishman and Sundheim 1996] and formulated as the Template Relation task in MUC-7 [MUC-7 1998], which was further reformulated as the RDC subtask in the NIST ACE program [ACE 2002–2008]. Thereafter, many machine-learning methods have been proposed for relation extraction, including supervised learning, semi-supervised learning, and unsupervised learning. As a dominant method, supervised learning can be further classified into feature-based [Jiang and Zhai 2007; Kambhatla 2004; Zhao and Grishman 2005; Zhou and Zhang 2007; Zhou et al. 2005], kernel-based [Bunescu and Mooney 2005; Culotta and Sorensen 2004; Zelenko et al. 2003; Zhang et al. 2006; Zhou et al. 2007], and composite kernel-based [Zhang et al. 2006; Zhou et al. 2007].

The problem related to feature-based methods is that it is quite difficult to further improve the performance of relation extraction through feature engineering, since almost all the flat features related to lexical, syntactic, and semantic knowledge have been systematically explored. In addition, most of the structural information, such as dependency information and parse tree, should become flattened (e.g., the chain of constituents along the path connecting the two involved entities in the parse tree) to be included in feature-based methods. This severely weakens the capability of effectively capturing their inherent structural nature.

As an alternative to feature-based methods, kernel-based methods enjoy the advantage of directly operating on discrete objects, leading to the potential of effectively modeling such objects, for example, capturing the structural information in the parse tree. Consequently, it can avoid the burden of feature engineering by directly

computing the similarity between any two trees corresponding to their respective relation instances.

Zelenko et al. [2003] describe a recursive kernel between shallow parse trees to extract semantic relations, where a relation instance is transformed into the least common sub-tree connecting the two entity nodes. The kernel matches the nodes of two corresponding sub-trees from roots to leaf nodes recursively layer by layer in a top-down manner. Their method shows successful results on two simple extraction tasks. Culotta and Sorensen [2004] propose a slightly generalized version of this kernel between dependency trees, in which a successful match of two relation instances requires the nodes to be at the same layer and in the identical path starting from the roots to the current nodes. These strong constraints make their kernel yield high precision but very low recall on the ACE RDC 2003 corpus. Bunescu and Mooney [2005] develop a shortest path dependency tree kernel, which simply counts the number of common word classes at each node in the shortest paths between two entities in dependency trees. Similar to Culotta and Sorensen [2004], this method also suffers from low recall, although its precision is promising.

Inspired by the successful application of convolution tree kernels to syntactic parsing [Collins and Duffy 2001] and semantic role labeling [Moschitti 2004], Zhang et al. [2006, 2008a] employs a convolution tree kernel to investigate various forms of structural information for relation extraction and find that the Shortest Path-enclosed Tree (SPT) achieves the F-measure of 67.7 on the seven relation types of the ACE RDC 2004 corpus. One problem with SPT is that it loses the contextual information outside SPT, which is usually critical for relation extraction. Zhou et al. [2007] notice the fact that both the SPT and the convolution tree kernel are context-free and thus expand the SPT to the CS-SPT by dynamically including necessary predicate-linked path information and further extend the standard convolution tree kernel to context-sensitive convolution tree kernel, obtaining the F-measure of 73.2 on the seven relation types of the ACE RDC 2004 corpus. However, the CS-SPT only recovers part of contextual information and still contains noisy information as much as the SPT. Another problem related to convolution tree kernels is that entity-related semantic information has not yet been well incorporated into the structural syntactic tree, though they are very helpful to relation extraction.

In order to utilize the advantages of both feature-based methods and kernel-based methods, some researchers resort to composite kernel-based methods. Zhao and Grishman [2005] define several feature-based composite kernels to capture diverse linguistic knowledge and achieve the F-measure of 70.4 on the seven relation types in the ACE RDC 2004 corpus. Zhang et al. [2006] design a composite kernel consisting of an entity linear kernel and a standard convolution tree kernel, obtaining the F-measure of 72.1 on the seven relation types in the ACE RDC 2004 corpus. Zhou et al. [2007] further describe a composite kernel to integrate a context-sensitive convolution tree kernel and a state-of-the-art linear kernel, achieving the so far best F-measure of 75.8 on the seven relation types in the ACE RDC 2004 corpus. Nguyen et al. [2009] explore various combinations of convolution kernels on constituent, dependency and sequential structures, and achieve the best performance of 71.5 in F-measure for the task of relation extraction on the seven major relation types in the ACE RDC 2004 corpus. However, they do not consider refining the syntactic structural representation of relation instances using dependency information, which is the focus of this article.

For feature-based Chinese relation extraction, please refer to Che et al. [2005b], Dong et al. [2007], Li et al. [2008], Zhang et al. [2009]. We will not detail the corresponding related work here due to our focus on kernel-based methods.

For kernel-based Chinese relation extraction, Che et al. [2005a] propose an improved edit distance measurement over Chinese strings to extract person-affiliation

from Chinese texts. Liu et al. [2007] develop a sequence kernel function over Chinese strings for extracting three major relation types and six relation subtypes from an ACE RDC Chinese corpus. However, they didn't mention the exact ACE RDC Chinese corpus they used. Huang et al. [2008] explore the convolution tree kernel and the shortest dependency path kernel for Chinese relation extraction. However, they only achieve the *F*-measure of 34.13 on the ACE RDC 2007 corpus, which is quite disappointing. Our previous work [Yu et al. 2010] exhibits the effectiveness of convolution tree kernel for Chinese relation extraction. With the SPT as the structural representation of relation instances, the *F*-measure of relation extraction on the major relation types reaches as high as 67.0 on the ACE RDC 2005 Chinese corpus.

In this article, we study how to dynamically determine a concise and effective tree span for a relation instance by exploiting constituent dependencies inherent in the parse tree derivation. Moreover, we attempt to capture both the structural syntactic parse information and entity-related semantic information via a unified syntactic and semantic tree framework. Finally, we evaluate the effectiveness of a composite kernel for relation extraction on both English and Chinese corpora in order to further boost the performance.

3. DYNAMIC SYNTACTIC PARSE TREE

This section first discusses constituent dependency and its role in relation extraction, and then describes how to generate the dynamic syntactic parse tree by employing various kinds of constituent dependencies to overcome the problems in the currently widely used tree spans.

3.1 Constituent Dependency

As described in Section 1, one key problem in kernel-based methods is appropriate representation and construction of the structural information in relation instances. Unlike the shallow parse tree [Zelenko et al. 2003] for the simple extraction tasks, the dependency tree [Culotta and Sorensen 2004], and the shortest dependency path tree [Bunescu and Mooney 2005] suffer from low recall in the ACE task due to strict similarity constraints. As for convolution tree kernels, Zhang et al. [2006] explore five kinds of tree spans and find that the Shortest Path-enclosed Tree (SPT, the smallest common sub-tree including the two entities) achieves the best performance. Zhou et al. [2007] further propose Context-Sensitive SPT (CS-SPT), which can dynamically determine the tree span by extending the necessary predicate-linked path information outside SPT. However, the critical problem of how to properly represent the structural syntactic parse tree is only partially resolved. As we indicate in what follows, current tree spans suffer from two disadvantages.

(1) Both SPT and CS-SPT still contain unnecessary information. For example, in the sentence "... bought *one* of town's two meat-packing *plants*" found in the ACE corpus, the condensed form "*one* of *plants*" is sufficient to determine "DISC"[1] relationship between the two entities "*one*" [FAC][2] and "*plants*" [FAC], while SPT/CS-SPT include the redundant underlined part due to their simplistic heuristics in tree construction. Therefore, more unnecessary information can be safely removed from SPT/CS-SPT without affecting the nature of semantic relationship.

[1]In the ACE terminology, a Discourse (DISC) relation is one where a semantic part-whole or membership relation is established only for the purpose of the discourse.
[2]FAC is an acronym introduced by the ACE 2004 program to represent an entity that is a large, functional, primarily man-made structure, such as a building or a plant.

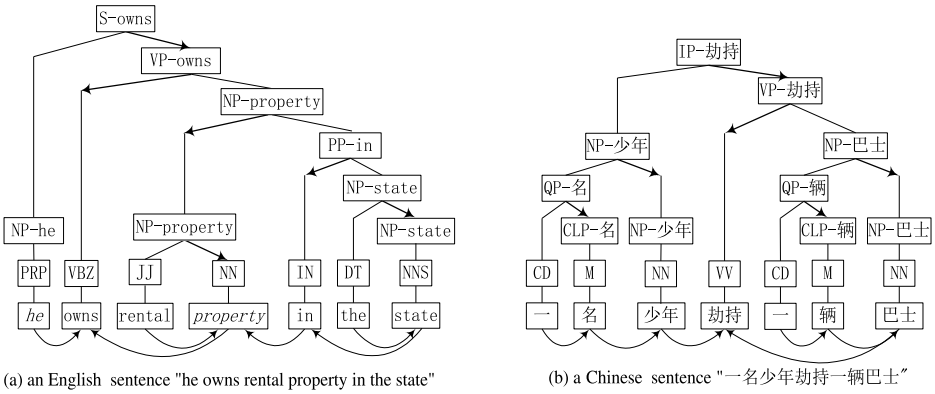

Fig. 1. Two parse tree derivations with upword-extended word dependency.

(2) CS-SPT only captures part of the context-sensitive information related to predicate-linked structure [Zhou et al. 2007] and still loses much useful context-sensitive information. Let's take the same example sentence "... bought *one* of *town*'s two meat-packing plants", where indeed there is no relationship between the entities "*one*" [FAC] and "*town*" [GPE][3] defined in the ACE corpus. Nevertheless, the information represented by SPT/CS-SPT (i.e., "*one* of *town*") may easily lead to their relationship being misclassified as "DISC", which is surely beyond our expectation. Therefore the underlined part outside SPT/CS-SPT should be recovered so as to differentiate it from positive instances.

Since dependency plays a key role in many NLP problems, such as syntactic parsing, semantic role labeling as well as semantic relation extraction, our motivation is to employ various kinds of dependency knowledge to distinguish the necessary evidence from the unnecessary information in the structural syntactic parse tree. Put in another way, this knowledge can be exploited to construct a proper structural representation for relation instances.

On one hand, lexical or word-word dependency indicates the relationship among words occurring in the same sentence. For example, predicate-argument dependency means that arguments are dependent on their target predicates while modifier-head dependency means that modifiers are dependent on their head words. Such dependency relationship offers a condensed representation of the information needed to assess the relationship in the form of the dependency tree [Culotta and Sorensen 2004] or the shortest dependency path tree [Bunescu and Mooney 2005] that includes two given entities.

On the other hand, when the syntactic parse tree corresponding to the sentence is derived from the bottom to the top using derivation rules step by step, the word-word dependencies converge upward, making a unique head child containing the head word for every non-terminal constituent. Figure 1 illustrates two example parse trees corresponding to an English sentence "he owns rental property in the state" and an Chinese sentence "一名少年劫持一辆巴士" (A teenager hijacked a bus), where the arrows at the bottoms denote the dependencies between words (the specific type of word dependency is omitted for simplicity since we do not utilize this kind of information), tree nodes except the terminal nodes and POS nodes are additionally tagged with the associated

[3]GPE refers in the ACE 2004 program to a Geo-Political Entity—an entity with a population, a government and a physical location, such as a nation (or province, state, county, city, etc.).

head words. Take the English sentence as an example, "NP-property" means that the node is tagged with "NP" with its head word being "property", and when "JJ NN" is reduced to "NP", the head word of "NN" ("property") is assigned to their parent node "NP" since the "NP" node is dependent on the "NN" node (this dependency relationship is highlighted using a bold line with an arrow pointing to its head child).

Generally, each context-free grammar (CFG) rule has the form of

$$\boldsymbol{P} \rightarrow L_n \ldots L_1 \boldsymbol{H} R_{1 \ldots m}.$$

Where, $\boldsymbol{P}$ is the parent node, $\boldsymbol{H}$ is the head child of the rule, $L_n \ldots L_1$ and $R_1 \ldots R_m$ are left and right modifiers of $\boldsymbol{H}$ respectively, and both n and m may be zero. In other words, the parent node $\boldsymbol{P}$ (dependent) depends on the head child $\boldsymbol{H}$ (governor), this is what we call *constituent dependency*. Vice versa, we can also determine the head child of a constituent in terms of constituent dependency. Our hypothesis stipulates that the contribution of the parse tree to establishing a relationship is almost exclusively concentrated in the path connecting the two entities, as well as the head children of constituent nodes along this path. More clearly speaking, only the path nodes as well as the governing nodes they are dependent on are considered necessary, while others are unnecessary for relation instances.

3.2 Generation of Dynamic Syntactic Parse Tree

Guided by the principle of constituent dependencies, beginning with the Minimum Complete Tree (MCT, the complete sub-tree rooted by the nearest common ancestor of the two entities under consideration) as the structural representation of each relation instance, the head child of every node is found according to the production rule along the path connecting the two entities. Then the path nodes and their head children are kept intact or recovered outside the MCT while any other nodes are gradually removed from the parse tree. Eventually we arrive at a condense tree representation called the Dynamic Syntactic Parse Tree (DSPT), which is dynamically determined by constituent dependencies and only contains necessary information as expected.

Due to the multitude of production rules with regard to parse trees, there exist a considerable number of constituent dependencies in a context-free grammar as described by Collins [2003]. However, since our task is to extract the relationship between two named entities, our focus is on how to condense useful constituents related to relation extraction, most of which are Noun Phrases (NPs). Therefore constituent dependencies can be classified into the following five categories according to constituent types of the CFG rules. Figure 2 and Figure 3 illustrate the application of these constituent dependencies to relation instances in both English and Chinese languages from the ACE RDC 2004/2005 corpora respectively.

(1) *Modification within base-NPs*. Base-NPs mean that they do not directly dominate another NP themselves, unless the dominated NP is a possessive NP. In particular, the noun phrase right above the entity headword, whose mention type is nominal or name, can be categorized into this type. In this case, the entity headword is also the headword of the noun phrase, thus all the constituents before the headword are dependent on the headword, and may be safely removed from the parse tree, while the headword and the constituents right after the headword remain unchanged. For example, in the sentence "... bought *one* of town's two meat-packing *plants*" as illustrated in Figure 2(a), the constituents before the headword "*plants*" can be removed from the parse tree. In this way the parse tree "*one* of *plants*" could capture the "DISC" relationship more concisely and precisely. Another interesting example is shown in Figure 2(b), where the base-NP of the second entity "*town*" is a possessive NP and there is no relationship between the entities "*one*" and "*town*" defined

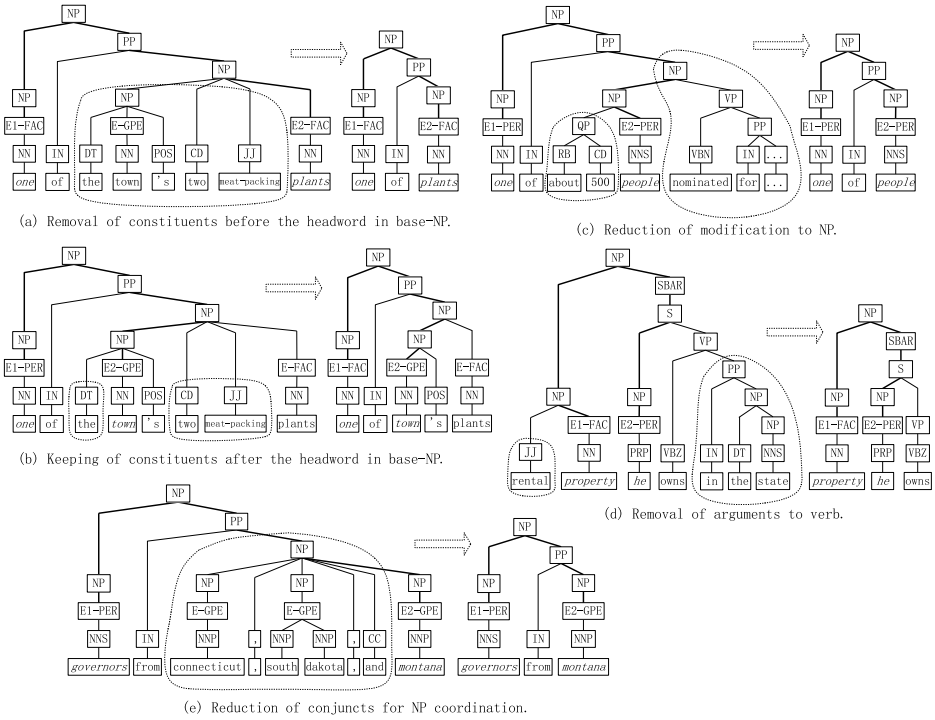

Fig. 2. Removal and reduction of constituents using constituent dependencies in English.

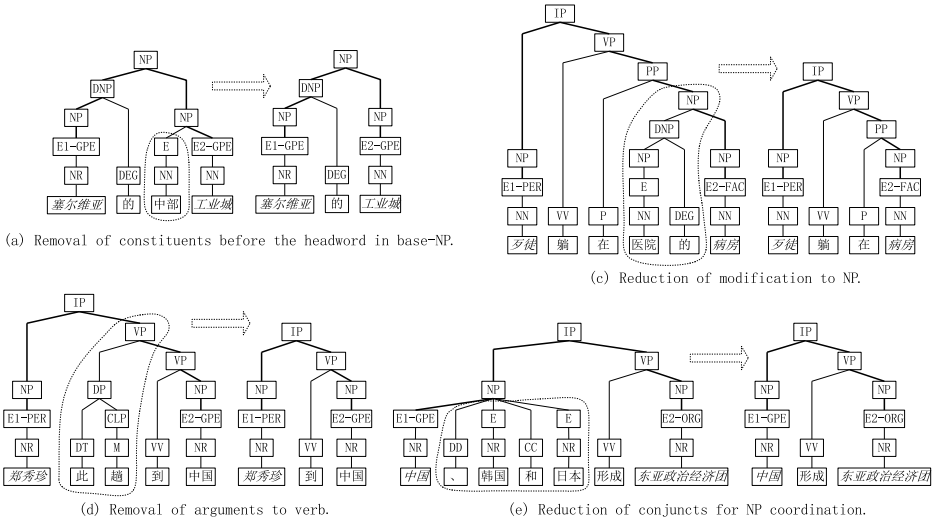

Fig. 3. Removal and reduction of constituents using constituent dependencies in Chinese.

in the ACE corpus. For both SPT and CS-SPT, this example would be condensed to "*one* of *town*" and therefore easily misclassified as the "DISC" relationship between the two entities. In the contrast, our DSPT can avoid this problem by keeping the constituent "'s" and the headword "*plants*".

For Chinese texts, modifications within base NPs are relatively simpler, and most of them are noun compound modifiers [Chang et al. 2009], as illustrated in Figure 3(a). In this phrase "*塞尔维亚*的中部*工业城*" (the central *industrial city* of *Serbia*), the noun modifier "中部" (central) is removed and the phrase can be transformed into "*塞尔维亚*的*工业城*" (the *industrial city* of *Serbia*), which is more concise and accurate for the relationship of PART-WHOLE.Geographical.

(2) *Modification to NPs*. Except base-NPs, other modification to NPs can be classified into this type. Usually these NPs are recursive, meaning that they contain another NP as their child. The CFG rules corresponding to these modifications may have the following forms:

$$\begin{aligned} \textbf{\textit{NP}} &\rightarrow \textbf{\textit{NP}}\ \textit{SBAR} && \text{[relative clause]} \\ \textbf{\textit{NP}} &\rightarrow \textbf{\textit{NP}}\ \textit{VP} && \text{[reduced relative]} \\ \textbf{\textit{NP}} &\rightarrow \textbf{\textit{NP}}\ \textit{PP} && \text{[PP attachment]} \end{aligned}$$

Here, the NPs in bold mean that the path connecting the two entities passes through them. For every right-hand side, the NP in bold is modified by the constituent following them. That is, the NP on the left side is dependent on the first NP on the right, and may be reduced to a single NP. Please note that the operation "reduce" is different from the operation "remove" in that it not only removes the unnecessary part, but also combines two remaining and successive NPs into a single one. For example, in Figure 2(c) we show a sentence "*one* of about 500 *people* nominated for . . . ", where there exists a "DISC" relationship between the entities "*one*" and "*people*". Since the reduced relative "nominated for . . . " modifies and is therefore dependent on the "*people*", they can be removed from the parse tree, that is, the right side ("NP VP") can be reduced to the left hand side, which is exactly a single NP.

In Chinese documents, however, unlike English sentences, modifications to NPs usually occur before the NP in the form of relative clause (DEC, 的), or less frequently in the associative form (DEG, 的) [Chang et al. 2009]. In both cases, they can be reduced to a more condensed single NP as depicted in Figure 3(c), where the sentence "*歹徒*躺在医院的*病房*" (The *bandit* is lying in the hospital's *ward*) is simplified as "*歹徒*躺在*病房*" (The *bandit* is lying in the *ward*).

(3) *Arguments/adjuncts to verbs*. This type includes the CFG rules in which the left side includes S, SBAR, or VP, etc. An argument represents the subject or object of a verb, while an adjunct indicates the location, date/time or way of the action corresponding to the verb. They depend on the verb and can be removed if they are not included in the path connecting the two entities. However, on the other hand, when the parent tag is S or SBAR, and its head child VP is not included in the path, this VP should be recovered to indicate the predicate verb. Actually, it is sufficient to restore the head constituent of the VP according to constituent dependency again. Figure 2(d) shows a sentence ". . . maintain rental *property he* owns in the state", where the "ART.User-or-Owner" relation holds between the entities "*property*" and "*he*". The VP should be kept in the rule ("S→ NP VP"), while the PP can be removed from the rule ("VP→ VBZ PP"). Similarly, Figure 3(d) shows how a Chinese sentence "*郑秀珍*此趟到*中国*" (This time *Zheng Xiuzheng* came to *China*) can be converted into "*郑秀珍*到*中国*" (*Zheng Xiuzheng* came to *China*) according to this dependency. Consequently, the resulting tree spans look more concise and precise for relation extraction.

(4) *Coordination conjunctions*. In coordination constructions, several peer conjuncts can be reduced into a single constituent. Most conjuncts of coordination constructions in the ACE corpus belong to NP type, while other types occur less frequently.

Although the first conjunct is always considered as the headword [Collins 2003], actually we assume that all the conjuncts in a coordination construction play an equal role in relation extraction. Hence all the conjuncts except the one in the path should be removed, and the remaining constituents can be further reduced. As illustrated in Figure 2(e), the NP coordination in the sentence ("*governors* from connecticut, south dakota, and *montana*") can be reduced to a single NP ("*governors* from *montana*") by keeping the conjunct in the path while removing the other conjuncts, thus leading this relation instance to be easily correctly recognized as the relationship of "DISC" type. Analogously in Chinese as depicted in Figure 3(e), the coordinated NP "*中国*、南韩和日本形成*东亚政治经济团*" (*China*, Korea and Japan form the *Eastern Asian Political and Economical Consortium*) is reduced to a single NP "*中国*形成*东亚政治经济团*" (*China* forms the *Eastern Asian Political and Economical Consortium*). Although its meaning is slightly changed, it's more obvious for recognizing the relationship of ORG-AFF.membership between these two entities.

(5) *Modification to other constituents*. Except for the above four major types, other CFG rules fall into this type, such as modifications to PP, ADVP, and PRN in English and other constituents in Chinese. Since these cases are similar to those of arguments/adjuncts to verbs and usually occur much less frequently in the corpora, we will not detail most of them here. One exception is the large number of occurrences of PP modifications in English, where the noun phrase typically contains the key information related to entities and can't be removed just as the preposition. As a consequence, PP modifications can't be simplified in most cases.

In fact, the SPT [Zhang et al. 2006] can be constructed by carrying out part of the above removal operations using a single heuristic rule (i.e., all the constituents outside the shortest path linking the two entities should be removed) and CS-CSPT [Zhou et al. 2007] further recovers part of necessary context-sensitive information outside the SPT. This intuitively justifies why the SPT performs well, and why the CS-SPT outperforms the SPT.

4. COMBINING SYNTACTIC AND SEMANTIC INFORMATION

This section first explores different representations of entity-related features. Then, two schemes that combine structural syntactic information and entity-related information are discussed. Finally, we present both convolution tree kernels and composite kernels used in our experiments.

4.1 Entity-Related Semantic Tree

Entity-related semantic features, such as entity headword, entity type and subtype, etc., impose a strong constraint on relation types according to the definitions by the ACE RDC task. For example, "PER-SOC" relations describe the relationship between entities of type PER, and no other entity type is allowed as an argument of these relations. Therefore, virtually all the relation extraction systems contain entity features in one form or the other. Particularly, experiments by Zhang et al. [2006] show that the linear kernel using only entity features contributes much when combined with the convolution parse tree kernel. Qian et al. [2007] further points out that among these entity features, entity type, subtype, and mention type, as well as the base form of predicate verb nearest to the second entity mention, contribute most while the contribution of other features, such as entity class, headword, and GPE role, can be ignored.

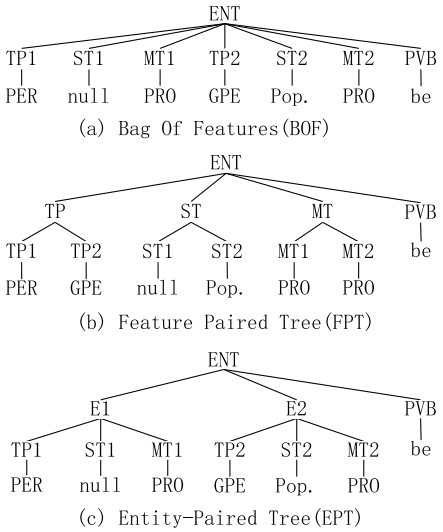

Fig. 4. Different setups for entity-related semantic tree (EST).

In order to effectively capture various kinds of entity-related semantic features and their combinations as well, particularly bi-gram or tri-gram features, we build an Entity-related Semantic Tree (EST) in three ways as illustrated in Figure 4. Take an English sentence "*they* 're *here*" (excerpted from the ACE RDC 2004 corpus) as an example, there exists a relationship "Physical.Located" between the entities "*they*" [PER] and "*here*" [GPE.Population-Center]. The features are encoded in the tags as "TP", "ST", "MT" and "PVB", which denote type, subtype, mention-type of the two entities, and the base form of predicate verb in English if existing (nearest to the second entity along the path connecting the two entities) respectively. For example, the tag "TP1" represents the entity type of the first entity, and the tag "ST2" represents the entity subtype of the second entity. Following are the three entity-related semantic tree setups.

(a) Bag of Features (BOF, e.g., Figure 4(a)). All feature nodes are uniformly attached under the root node "ENT". That is, the tree kernel simply counts the number of common entity features between two relation instances. This tree setup is similar to the linear entity kernel explored by Zhang et al. [2006].
(b) Feature-Paired Tree (FPT, e.g., Figure 4(b)). The features of two entities are first grouped into different types according to their feature names, and then attached under the root node, for example, "TP1" and "TP2" are grouped to "TP", which is further attached under the root node. This tree setup aims to capture the additional similarity between pairs of the same features combined from the first and the second entities.
(c) Entity-Paired Tree (EPT, e.g., Figure 4(c)). All the features relating to an entity are first grouped to nodes "E1" or "E2" respectively, and then are attached under the root node. This enables the tree kernel to further explore the equivalence of combined entity features only relating to one of the entities between two relation instances.

For the preceding three setups, the predicate feature is always attached under the root node, since it is unrelated to any of the two entities. In summary, the BOF only

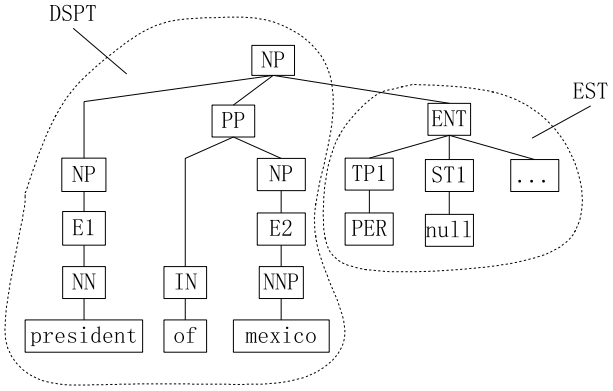

Fig. 5. Unified Syntactic and Semantic Tree (USST).

captures the individual entity features, while the FPT and EPT setups are designed to additionally capture the bi-gram and tri-gram features respectively.

4.2 Unified Syntactic and Semantic Tree

After determining the structural representations for both the syntactic parse tree and entity-related semantic tree, the question is how to combine them into an effective form for relation extraction. One common way is via a composite kernel, as explored by Zhang et al. [2006]. While we will discuss the composite kernel in the next section, here we propose a unified syntactic and semantic tree (USST), which integrates the entity-related semantic tree into the dynamic syntactic parse tree. Although the entity features can be attached under the top node, the entity nodes, or directly embedded into the labels of the entity nodes, we only explore to attach the three kinds of entity-related semantic trees (i.e., BOF, FPT, and EPT) under the top node of the dynamic syntactic parse tree right after its original children since detailed evaluation [Qian et al. 2007] indicates that the unified tree achieves the best performance when the feature nodes are attached under the top node. Figure 5 shows an instance of the USST with the entity-related semantic tree "BOF" attached. This instance is directly chosen from the ACE RDC 2004 corpus. Here, we omit its Chinese counterpart because they look almost similar except that the words and the node tags' names are associated with Chinese.

4.3 Tree Kernel and Composite Kernel

After having rendered relation instances as structured representations—DSPT and USST, we now turn to discuss how to measure the similarity between two relation instances with a convolution kernel. A convolution kernel [Haussler 1999] aims to capture common structural information between two discrete objects in terms of their sub-structures, such as the convolution tree kernel [Collins and Duffy 2001], the string kernel [Lodhi et al. 2002], and the graph kernel [Suzuki et al. 2003]. Following previous studies in relation extraction [Zhang et al. 2006; Zhou et al. 2007], this article also adopts the convolution tree kernel [Collins and Duffy 2001] to compute the similarity between two relation instances due to its effectiveness in capturing the structural information inherent in the parse tree.

For the convolution tree kernel to be incorporated into statistical classifiers such as SVMs, a training or test instance should be ultimately casted as a feature vector, in which features are all of its subtree types and the value of each feature is the

number of occurrences of that subtree type in the whole tree. Formally, a tree T can be represented as a vector of integer counts of each subtree type, that is:

$$\phi(T) = (\#subtree - type_1(T), ..., \#subtree - type_n(T)). \quad (1)$$

Where $\#subtree\text{-}type_i(T)$ denotes the number of occurrences of the ith subtree type in the tree T. Intuitively, the convolution tree kernel counts the number of common subtree types as the structure similarity between two trees T_1 and T_2. Since the number of subtree types in a tree is exponential with its size, it is infeasible to enumerate all the subtree types and directly use the feature vector $\phi(T)$. In order to address this computational problem, Collins and Duffy [2002] propose the following method to implicitly compute the dot product between two high-dimensional vectors $\phi(T_1)$ and $\phi(T_2)$:

$$\begin{aligned} K_C(T_1, T_2) &= < \phi(T_1), \phi(T_2) > \\ &= \sum_i \#subtree - type_i(T_1) \cdot \#subtree - type_i(T_2) \\ &= \sum_i \left(\sum_{n_1 \in N_1} I_{subtree_i}(n_1) \right) \cdot \left(\sum_{n_2 \in N_2} I_{subtree_i}(n_2) \right) \\ &= \sum_{n_1 \in N_1, n_2 \in N_2} \Delta(n_1, n_2), \end{aligned} \quad (2)$$

where N_1 and N_2 are the sets of nodes in tree T_1 and T_2, respectively, and $I_{subtree_i}(n)$ is a binary function whose value is 1 if the *subtree-type*$_i$ occurs with its root at node n or zero otherwise, and $\Delta(n_1, n_2)$ evaluates the number of common subtree types rooted at n_1 and n_2,[4] that is:

$$\Delta(n_1, n_2) = \sum_i I_{subtree_i}(n_1) \cdot I_{subtree_i}(n_2). \quad (3)$$

$\Delta(n_1, n_2)$ can be computed recursively as follows.

1) If the context-free productions (CFG rules) at n_1 and n_2 are different, then $\Delta(n_1, n_2) = 0$; otherwise go to Step 2.
2) If both n_1 and n_2 are POS tags, $\Delta(n_1, n_2) = 1 \times \lambda$; otherwise go to Step 3.
3) Calculate $\Delta(n_1, n_2)$ recursively as:

$$\Delta(n_1, n_2) = \lambda \prod_{k=1}^{\#ch(n_1)} (1 + \Delta(ch(n_1, k), ch(n_2, k))), \quad (4)$$

where $\#ch(n)$ is the number of children of node n, $ch(n, k)$ is the kth child of node n and $\lambda (0 < \lambda < 1)$ is the decay factor in order to make the kernel less variable with respect to different sub-tree sizes.

This kernel has been successfully applied to many tasks in natural language processing, such as syntactic parsing [Collins and Duffy 2002], semantic role labeling [Moschitti 2004, 2006; Moschitti et al. 2006, 2008; Zhang et al. 2008b] and relation extraction [Zhang et al. 2006, 2008a; Zhou et al. 2007] as well.

In order to integrate the power of different kernels, composite kernels are usually an effective way to combine several kernels over different structures. With kernels over the dynamic syntactic parse tree and the entity-related semantic tree (or even a

[4]That is, each node n encodes the identity of a sub-tree rooted at n and, if there are two nodes in the tree with the same label, the summation will go over both of them.

feature-based linear kernel) at hand, we can further consider two kinds of composite kernels as follows:

1) Linear combination

$$K_{LC}(R_1, R_2) = \alpha \cdot K_1(R_1, R_2) + (1 - \alpha) \cdot K_2(R_1, R_2). \quad (5)$$

where K_1 and K_2 denotes the two kernels under consideration and α is the coefficient, which can be determined using two-fold cross-validation on the training dataset.

2) Polynomial combination

$$K_{PC}(R_1, R_2) = \alpha \cdot K_1^P(R_1, R_2) + (1 - \alpha) \cdot K_2(R_1, R_2), \quad (6)$$

where K_1^P denotes the polynomial expansion of K_1 with the degree $d = 2$, that is, $K_1^P = (K_1 + 1)^2$, the others are the same as those in Equation (5). Since the set of kernels is closed under normalization, linear and polynomial combination [Schölkopf and Smola 2001], these two composite kernels are also proper kernels.

5. EXPERIMENTATION

This section will evaluate the effectiveness of the dynamic syntactic parse tree and the contribution of various kinds of entity-related semantic information via the unified syntactic and semantic tree framework.

5.1 Experimental Setting

For evaluation, we use the ACE RDC 2004 English corpus and the ACE RDC 2005 Chinese corpus as the benchmark data (hereafter we will refer to them as the English corpus and the Chinese corpus respectively). The English corpus was gathered from various sources of newspapers, newswire, and broadcasts. This data set contains 451 documents and 5,702 relation instances. It defines seven entity types, seven major relation types and 23 relation subtypes. For fair comparison with previous work [Zhang et al. 2006; Zhou et al. 2007], evaluation is done on the same subset of 348 documents (nwire/bnews) and 4,400 relation instances using 5-fold cross-validation. The Chinese corpus contains 633 documents, which were collected from newswire, broadcasts, and weblogs. It defines seven entity types, six major relations types, and 18 relations subtypes. Due to many grammatically ill-formed sentences in the weblogs and for comparability with the English corpus, evaluation is only done on 533 documents (nwire/bnews) and 7,630 relation instances of the Chinese corpus using the same 5-fold cross-validation scheme.

The English corpus is first parsed using the state-of-the-art Charniak's parser [Charniak 2001] with the boundaries of all the entity mentions kept. Then, relation instances are generated by iterating over all pairs of entity mentions occurring in the same sentence with the given "true" mentions and co-referential information. For fair comparison, various kinds of structured representations of relation instances are generated respectively, such as DSPT described in Section 3, EST in Subsection 3.1, USST in Subsection 4.2 as well as the feature vector used in composite kernels in Table VI and Table VII. This same preprocessing strategy is also applied to the Chinese corpus, except that the Chinese sentences are first segmented by the Stanford Chinese Word Segmenter and then parsed by the Stanford Parser.[5]

[5]See http://nlp.stanford.edu/software/.

Table I. Contributions of Constituent Dependencies in the Individual Mode (Inside Parentheses) and the Accumulative Mode (Outside Parentheses) Respectively

Dependency types	ACE 2004 (English)			ACE 2005 (Chinese)		
	P (%)	R (%)	*F*	P (%)	R (%)	*F*
MCT (baseline)	75.1	53.8	62.7	77.5	46.1	57.8
Modification within base-NPs	76.5 (59.8)	59.8 (59.8)	67.1(***) (67.1)(***)	75.3 (75.3)	46.5 (46.5)	57.5 (57.5)
Modification to NPs	77.0 (76.2)	63.2 (56.9)	69.4(***) (65.1)(***)	76.0 (76.0)	54.1 (54.1)	63.2(***) (63.2)(***)
Arguments/adjuncts to verbs	77.1 (76.1)	63.9 (57.5)	69.9(***) (65.5)(***)	78.3 (75.9)	55.2 (50.1)	64.7(***) (60.4)(**)
Coordination conjunctions	77.3 (77.3)	65.2 (55.1)	70.8(**) (63.8)(***)	79.1 (75.6)	56.4 (49.3)	65.8(***) (59.7)(*)
Other modifications	**77.4** (75.0)	**65.4** (53.7)	**70.9**(**) (62.6)	**79.1** (76.1)	**56.9** (46.6)	**66.2**(***) (57.8)

In our experimentations, SVM^{light} [Joachims 1998] with the tree kernel function SVM-TK toolkits [Moschitti 2004][6] is selected as our classifier. For efficiency consideration, we apply the *one vs. others* strategy, which builds K classifiers so as to separate one class from all others. For the purposes of meaningful comparison, the training parameters C (SVM) and λ (tree kernel) are also set to 2.4 and 0.4 respectively, which had been empirically verified to exhibit the best performance in previous relation extraction research. As to the coefficients α in the composite kernels, 2-fold cross-validations on the training dataset are used to determine their optimal values. That is, the training dataset is divided into two disjoint and independent folds, one of which is used to induce the classifier models, while the other is tested to find the optimal parameter values to attain the best performance score. For reasonable comparison between English and Chinese corpora, the same set of parameter values is applied to both the classifiers.

5.2 Experimental Results

Our experiments are presented in the following order:

(1) evaluating the contributions of various kinds of constituent dependencies in DSPT,
(2) evaluating the contributions of various kinds of entity-related semantic features in EST,
(3) comparing the performance between different entity-related semantic tree setups in USST,
(4) evaluating the effectiveness of composite kernels by DSPT and EST,
(5) comparing with other state-of-the-art relation extraction systems.

To determine whether an improvement is significant or not, we conduct significance tests using the paired t-test method. In this article, "***", "**", and "*" denote p-values of less than 0.01, 0.01–0.05, and greater than 0.05, which mean significantly better, moderately better, and slightly better, respectively.

Evaluating the contributions of various kinds of constituent dependencies. Table I evaluates the contributions of different kinds of constituent dependencies on the major relation types of the ACE corpora using the convolution parse tree kernel (with only the entity-

[6]See http://ai-nlp.info.uniroma2.it/moschitti/.

type information attached as a child under the tree root node in the BOF manner as illustrated in Figure 4(a)). The MCT with only the entity-type information is first used as the baseline, and various constituent dependencies are then applied sequentially to dynamically reshaping the tree in two different modes.

—[M1] Individual Mode. Every constituent dependency is individually applied on the MCT, hence this mode is aimed to assess the separate contribution of each constituent dependency. In this mode, significance tests are conducted between each specific dependency type and the MCT baseline.
—[M2] Accumulative Mode. Every constituent dependency is incrementally applied on the previously derived tree span, which begins with the MCT and eventually gives rise to a Dynamic Syntactic Parse Tree (DSPT). In this mode, significance tests are conducted between each dependency type and its previously-considered ones.

The table reports that the final DSPT achieves the best performance of 77.4%/65.4%/70.9 for English and 79.1%/56.9%/66.2 for Chinese in precision/recall/*F*-measure respectively after applying all the dependencies, with the increase of the *F*-measure by 8.2 units (English) and 8.4 units (Chinese) over the baseline MCTs. This indicates that reshaping the tree by exploiting constituent dependencies can significantly improve the extraction accuracy largely due to the increase in the recalls. It further suggests that the knowledge on constituent dependencies is very effective and can be fully utilized in tree kernel-based relation extraction. Furthermore, this table also shows the following.

(1) Modification within base-NPs significantly (***) improves the accuracy for English, with the increase of the *F*-measure by 4.4 units in both modes (since it is the first constituent dependency applied to the MCT, there is no distinction in the performance between these two modes). Since there are many entity mentions, for example, "the town's two meat-packing plants", covering base-NPs, it is beneficial to remove redundant information from these NPs while keeping their headwords intact according to the dependency relationship. For Chinese base NPs, however, it's surprising that removing the modifiers slightly hurts the performance. The reason may be twofold. One is that there are much fewer adjectival modifiers in Chinese than in English. The other one is that there are a large amount of compound noun modifiers in Chinese [Chang et al. 2009], which are usually critical for relation extraction due to the branching of the compound NP and therefore should not be removed. For example, in the Chinese phrase "人民军总政治局局长" (the director of the General Political Bureau of the People's Army), there is a relationship of PART-WHOLE.Subsidiary between "人民军" (the People's Army) and "总政治局" (the General Political Bureau) and a relationship of ORG-AFF. Employment between "总政治局" and "局长" (the director), but no relationship between "人民军" and "局长", which would be easily recognized as a positive relation if the noun phrase "总政治局" is removed. Henceforth, in the subsequent experiments such dependency information will not be applied to the Chinese corpus.
(2) Modification to NPs contributes much to the performance improvement (***), increasing the *F*-measure by 2.4/2.3 units in individual and accumulative modes respectively for English and 5.4/5.4 units for Chinese. The reason may be that a noun phrase is usually located at the parent node of the entity node, thus simplifying its structure is very helpful to the performance improvement. This phenomenon is more prevalent in Chinese texts where there are a large number of relative clausal modifiers or associative modifiers marked by the particular Chinese word "的" (DEC/DEG) before an NP. The improvement from this modification together with the modification within base-NPs, suggests the local characteristic

Table II. Contributions of Different Kinds of Entity-Related Semantic Features on the ACE RDC Corpora

#	Entity Info	ACE 2004 (English)			ACE 2005 (Chinese)		
		P (%)	R (%)	*F*	P (%)	R (%)	*F*
1	DSPT (entity type)	76.4	65.2	70.9	79.1	56.9	66.2
2	Entity subtype (+)	78.2	66.3	72.2	**80.6**	58.3	67.7
3	Mention level (+)	80.0	68.1	74.0	79.7	**60.9**	**69.1**
4	Entity class	80.2	67.8	73.9	80.0	59.8	68.4
5	GPE role	79.8	67.7	73.7	80.0	60.1	68.6
6	Head word	80.0	67.5	73.6	79.4	60.8	68.8
7	LDC type	80.0	67.7	73.7	79.6	60.9	69.0
8	Predicate base (+)	**80.2**	**69.2**	**74.7**	-	-	-

of semantic relations, which can be effectively captured by NPs near the two involved entity mentions in the DSPT.

(3) Modification of arguments/adjuncts to verbs improves the *F*-measure by 2.8/2.6 units (English/Chinese) in the individual mode, but only 0.5/1.5 units (English/Chinese) in the accumulative mode. Since the MCT may contain much noisy information related to the modifications to verbs, removal of such noise leads to significant simplification of the tree structure in the individual mode, thereby a substantial increase in recall. While in the accumulative mode, this kind of modification fails to much improve the performance since most of such noisy information has already been pruned away in the preceding two steps and the verbs are relatively far away from the entity mentions.

(4) Reduction of coordination conjunctions, in particular noun phrase conjunctions in English and verb phrase conjunctions in Chinese exhibits positive results for relation extraction, increasing the *F*-measure by 0.9/1.1 units in the individual and accumulative modes respectively for English and 1.9/1.1 units for Chinese. As Figure 2(e) and Figure 3(e) shows, the complexity of the tree structure can be greatly reduced after we compress the coordination structure into a single conjunct with the nature of the semantic relationship kept intact. Therefore, it is not surprising that both modes achieve comparable improvements. However, one thing that should be pointed out is that in Chinese, the performance increase comes largely from the reduction of VP coordination conjuncts which are prevalent in Chinese documents and can also be parsed correctly, while the reduction of NP coordination conjuncts lower the performance largely due to their parsing errors.

(5) Other modifications show trivial effects on the relation extraction performance, with the decrease of the *F*-measure by 0.1/0.0 units (English/Chinese) in the individual mode and the increase of the *F*-measure only by 0.1/0.4 units (English/Chinese) in the accumulative mode. This is due to the fact that such modifications occur much less frequently and little simplification could be made further.

In summary, the DSPT produced by applying constituent dependencies to a parse tree involving two given entities can dramatically boost the performance of relation extraction for both English and Chinese. Particularly, the modification within base-NPs in English and the modification to NPs in both languages contribute most to the improvement, while other modifications help relatively less. Since most of entity mentions are encapsulated in noun phrases, this result indicates the locality of semantic relationship occurring in the ACE corpora.

Table III. Performance of the Unified Syntactic and Semantic Trees (USSTs) on the Seven Relation Types of the ACE RDC Corpora

Tree setups	ACE 2004 (English)			ACE 2005 (Chinese)		
	P (%)	R (%)	*F*	P (%)	R (%)	*F*
SPT(baseline)	76.3	59.8	67.1	80.2	53.3	64.0
DSPT	77.4	65.4	70.9(***)	79.1	56.9	66.2(***)
USST (BOF)	80.4	69.7	74.7(***)	80.5	58.6	67.8(***)
USST (FPT)	**80.1**	**70.7**	**75.1**(***)	**79.8**	**61.0**	**69.2**(***)
USST (EPT)	79.9	70.2	74.8(***)	80.5	57.8	67.3(***)

Evaluating the contributions of various kinds of entity-related semantic features in EST. Table II further evaluates the contributions of various kinds of entity-related semantic features on the extraction of the major relation types on the ACE RDC corpora using the BOF setup by adding them incrementally in the decreasing order of their potential importance. Note that the plus sign after a specific feature means that this feature is useful and should be included in the next round. This table reports that our system achieves the best performance of 80.2%/69.2%/74.7 in terms of P/R/F (English) and 79.7%/60.9%/69.1 (Chinese) respectively. It's interesting to notice that both English and Chinese exhibit consistent behavior on these features. Specifically, it also shows the following.

(1) Both entity subtype and mention level information moderately improve the *F*-measure by 1.3/1.8 units (English) and 1.5/1.4 units (Chinese) respectively due to consistent increases in both precision and recall. This indicates that gracefully defined entity type/subtype and mention type information can contribute a great deal in the ACE RDC corpora. This is not surprising since entity type/subtype impose strong constraints on the relation types between entities according to their definitions. For the mention level, when combined with entity type and subtype, it also exhibits some kind of discrimination.
(2) It is a bit surprising, however, to observe that the other four kinds of information, including entity class, GPE role, headword and LDC mention type etc., decrease the *F*-measure by 0.4/0.3/1.0/1.0 units (English) and 0.7/0.5/0.3/0.1 units (Chinese) respectively. This is contrary to earlier findings [Zhang et al. 2006] that headword and LDC mention type are helpful to relation extraction in the form of composite kernels. The reason may be that the feature is either too specific (like headword) or too general to be discriminative (like entity class).
(3) Finally for English, the predicate verb (in its basic form) closest to the second entity along the path connecting the two entities slightly improves the *F*-measure by 0.6 units. This suggests that the predicate verb may help relation extraction when its basic form is attached as a child under the tree root.

Comparing the performance between different entity semantic tree setups in USST. We compare in Table III the performance of the Unified Syntactic and Semantic Trees with different kinds of Entity Semantic Tree setups described in Section 4, while the SPT and DSPT with only entity-type information are listed for reference. Significant tests are conducted between USSTs and SPT. Note that for the English corpus since only four features, such as entity type, entity subtype, mention level and predicate base, show positive effects from the previous experiments, these features are incorporated into the USSTs as illustrated in Figure 4. The same scheme is also applied to the

Table IV. Improvements of Different Proposed Tree Setups over the Widely-Used SPT on the ACE RDC Corpora

Tree setups	ACE 2004 (English)			ACE 2005 (Chinese)		
	P (%)	R (%)	*F*	P (%)	R (%)	*F*
CS-SPT over SPT[7]	1.5	1.1	1.3	-	-	-
DSPT over SPT	1.1	5.6	3.8(***)	-1.1	3.6	2.2(***)
USST (FPT) over SPT	3.8	10.9	8.0(***)	-0.4	7.7	5.2(***)

Chinese corpus, except that the predicate base is unavailable. The table shows the following.

(1) All the three setups of USST significantly (***) outperform the DSPT setup, obtaining an average increase of ∼4 units for English and ∼2.7 units for Chinese in terms of the *F*-measure. This means that they can effectively capture both the structured syntactic information and the flat entity-related semantic information through the USST framework.
(2) The Unified Syntactic and Semantic Tree with FPT (the Feature-Paired Tree structure) achieves the best performance of 80.1/70.7/75.1 (English) and 79.9/61.0/69.2 (Chinese) in terms of P/R/F respectively, with the increase of the *F*-measure by 0.4/0.3 units over the BOF and EPT setups respectively for English and with the corresponding increase by 0.1/0.7 units for Chinese. This suggests that additional bi-gram entity features captured by the FPT are more useful than tri-gram entity features captured by the EPT.

In Table IV we further summarize the improvements of different tree setups over the SPT for both English and Chinese corpora, namely, CS-SPT, DSPT and USST (FPT). Since these results are produced in a similar setting, that is, extraction of major relation types on the same corpus with the same convolution parse tree kernel using the same 5-fold validation scheme, they can be compared fairly. It shows that for English our DSPT significantly (***) outperforms SPT by 3.8 units in *F*-measure, while the CS-SPT outperforms the SPT by 1.3 units in *F*-measure, and for Chinese our DSPT significantly (***) outperforms SPT by 2.2 in *F*-measure. This suggests that the DSPT performs best among these tree spans. It also shows that the Unified Syntactic and Semantic Tree with FPT (Feature-Paired Tree) performs significantly better than the other two tree setups (i.e., CS-SPT and DSPT) by 6.7/4.2 units in *F*-measure respectively for English, though the increase is not so distinctive for Chinese. This implies that the entity-related semantic information is very useful and contributes much when they are incorporated into the parse tree for relation extraction.

Evaluating the effectiveness of composite kernels by DSPT and EST. These experiments are aimed to evaluate the effectiveness of composite kernels by using either linear combination or polynomial combination, and to compare the difference between the composite kernels and the unified syntactic and semantic tree kernels. Table V reports the results in terms of P/R/F corresponding to the composite kernels by the DSPT and three different EST setups, using linear combination (outside parentheses) and polynomial combination (inside parentheses) respectively. These results are obtained on the extraction of the major relation types in the ACE RDC corpora. For

[7]We arrive at these values by subtracting P/R/F (79.6/5.6/71.9) of Shortest-enclosed Path Tree from P/R/F (81.1/6.7/73.2) of Dynamic Context-Sensitive Shortest-enclosed Path Tree according to Table II [Zhou et al. 2007].

Table V. Comparison of Composite Kernels on the ACE RDC Corpora Using Linear Interpolation (Outside Parentheses When Coefficient α = 0.4) and Polynomial Interpolation (Inside Parentheses When d = 2, α = 0.2)

Composite kernels	ACE 2004 (English)			ACE 2005 (Chinese)		
	P (%)	R (%)	*F*	P (%)	R (%)	*F*
DSPT (baseline)	77.4	65.4	70.9	79.1	56.9	66.2
DSPT+EST-BOF	78.2	69.3	73.5(***)	81.3	60.4	69.3(*)
	(78.1)	(69.6)	(73.6)(***)	(80.9)	(62.5)	(70.5)(***)
DSPT+EST-FPT	**79.8**	**71.6**	**75.4**(***)	81.3	61.4	69.9(**)
	(78.7)	(71.9)	(75.1)(***)	(80.8)	(63.0)	(70.8)(***)
DSPT+EST-EPT	78.3	71.2	74.6(***)	81.2	61.0	69.6(*)
	(78.1)	(71.3)	(74.6)(***)	(80.9)	(61.8)	(70.1)(***)

reference purposes, the results for the DSPT with only entity-type information are also listed. Significant tests are conducted between composite kernels and the single DPST kernel.

When determining the coefficient for linear interpolation in Equation (5), K_1 represents the EST tree kernel while K_2 represents the DSPT tree kernel. Using 2-fold cross-validation on the training data, the coefficient α is fine-tuned to 0.4. For polynomial combination in Equation (6), K_1 and K_2 denote the same kernels as for linear combination. Same as the most common setting in the literature [Zhang et al. 2006; Zhou et al. 2007], the polynomial order d is set to 2, while the coefficient α is fine-tuned to 0.2 in the same way as for the linear combination. From Table V we can see the following.

(1) All the three setups of composite kernels, regardless of their combination methods, substantially outperform the single DSPT kernel with only entity type information, obtaining an average increase of ∼3.6/∼3.8 units (English/Chinese) in terms of *F*-measure. This suggests that as an alternative method to combine syntactic information and semantic information, composite kernels can also work effectively to boost the performance.
(2) Among these composite kernels, for English the linear one with FPT achieves the best performance of 79.8%/71.6%/75.4 (***) in terms of P/R/F, while for Chinese the polynomial one with FPT achieves the best performance of 80.8%/63.0%/70.8 (***) in terms of P/R/F, both moderately higher than the single USST one with FPT. The reasons are twofold. One is that the contribution of bi-gram semantic features is more useful than tri-gram features, as is illustrated in Table III. The other is that, since we can adjust the weight of the composite kernel, the influences of syntactic and semantic information can be balanced to maximize the ultimate performance. Put in another way, this point can also be used to justify that without any parameter estimation, the single USPT kernel can achieve comparable results compared to the composite kernels.
(3) In contrast to the findings by Zhang et al. [2006] and Zhou et al. [2007] that the polynomial combination of composite kernels by a linear kernel and a tree kernel achieves better performance than the linear one. In our setting, however, these two combination methods perform comparably for English and the polynomial one performs slightly better than the linear one for Chinese. The reason may be that the FPT and EPT setups have already captured bi-gram/tri-gram semantic features, thus the polynomial combination will not further significantly promote the performance.

Table VI. Comparison of Different Methods on the ACE RDC 2004 English Corpus

Systems	P (%)	R (%)	*F*
Ours: composite kernel	**83.0**	**72.0**	**77.1**
Zhou et al. [2007]: composite kernel	82.2	70.2	75.8
Zhou et al. [2007]: composite kernel	76.1	68.4	72.1
Nguyen et al. [2009]: Composite kernel	76.6	70.2	71.5
Jiang and Zhai [2007]: Composite kernel	72.4	70.2	71.3
Zhao and Grishman [2005]: composite kernel	69.2	70.5	70.4
Ours: CTK with USST	80.1	70.7	75.1
Zhou et al. [2007]: context-sensitive CTK with CS-SPT	81.1	66.7	73.2
Zhang et al. [2006]: CTK with SPT	74.1	62.4	67.7
Zhou et al. [2005]: Feature-based	81.9	58.3	68.1

Table VII. Comparison of Different Methods on the ACE RDC 2005 Chinese Corpus

Systems	P (%)	R (%)	*F*
Ours (linear+tree): Composite kernel	**80.9**	**61.8**	**71.1**
Ours (DSPT+FPT): Composite kernel	80.8	63.0	70.8
Zhang et al. [2009]: Composite kernel	81.83	49.79	61.91
Ours: CTK with USST	79.8	61.0	69.2
Yu et al. [2010]: CTK with USST	75.3	60.4	67.0
Li et al. [2008]: Feature-based	81.67	61.70	70.29

Comparing with other state-of-the-art relation extraction systems. Finally, Table VI compares our system with other state-of-the-art kernel-based systems on the major relation types of the ACE RDC 2004 corpus. It shows that on the ACE RDC 2004 corpus our USST (FPT) outperforms all previous tree setups using one single kernel, even better than two previous composite kernels [Zhang et al. 2006; Zhao and Grishman 2005]. Furthermore, when the USST (FPT) kernel is combined with a linear state-of-the-state feature-based kernel [Zhou et al. 2005] into a composite one polynomial combination in a setting similar to Zhou et al. [2007] (i.e., polynomial degree d = 2 and coefficient α = 0.3), we get the so far best performance of 77.1 in *F*-measure for seven major relation types on the ACE RDC 2004 English corpus.

For Chinese, we implemented a feature-based relation extraction system with the similar features as Zhou et al. [2005], including lexical words, entity features, overlapping features and chunking information etc., and achieve the performance of 77.98%/52.34%/62.64 in terms of P/R/F on the ACE RDC 2005 Chinese corpus. Then this linear kernel is further combined with the USST (FPT) kernel for Chinese relation extraction in the same way and using the same parameters as the composite kernel for English. The experiment results are compared in Table VII with other state-of-the-art Chinese relation extraction systems on the ACE RDC 2005 Chinese corpus. However, this comparison is by no means fair due to their distinct evaluation schemes and therefore only for reference. For example, while we adopt the 5-fold cross validation strategy on the ACE RDC 2005 Chinese corpus, both Li et al. [2008] and Zhang et al. [2009]) use 75% of the total relation instances as the training set and the remaining instances as the test data, and furthermore the total number of relation instances are also different due to different preprocessing procedures. Specifically, Li et al. [2008] explore character-based unigram and bi-gram features as well as nine positional structures,

plus some correction and inference mechanisms based on the relation hierarchy and co-reference information, while we use convolution tree kernels over syntactic parse trees and entity-related semantic trees. Actually, syntactic parse trees include some kind of positional structures. In Zhang et al. [2009], they combine an entity semantic kernel and a string semantic similarity kernel for Chinese relation extraction, while ignoring the structural information inherent in parse trees. Although comparison of our work with these two works is difficult, nevertheless, one important thing to be sure is that, the USST employing constituent dependencies significantly outperforms that in our previous work [Yu et al. 2010] under the same setting, thus indicating the effectiveness of the convolution tree kernels over syntactic parse trees for extracting Chinese semantic relations as well as for extracting English semantic relations.

6. CONCLUSION

This article systematically explores the potential of structured syntactic information for tree kernel-based relation extraction on both English and Chinese. In particular, a new approach is proposed to dynamically determine the tree span (DSPT) for relation instances by exploiting constituent dependencies. In addition, we investigate different ways of integrating various kinds of entity-related semantic information via a Unified Syntactic and Semantic Tree (USST). Evaluation on the ACE RDC corpora shows that our DSPT is appropriate for structural syntactic representation of relation instances for both English and Chinese languages. It also shows that, in addition to individual entity features, combined entity features (especially bi-gram) contribute much when they are combined with the DSPT into the USST framework. Finally, this article reports the so-far best performance for both English and Chinese corpora when combining the USST-based tree kernel and a state-of-the-art feature-based linear kernel via a composite kernel.

For the future work, on one hand, we will focus on improving the performance of complex structural parse trees, where the path connecting the two entities involved in a relationship is too long for current kernel methods to take effect. Our preliminary experiments of employing a discourse theory exhibit certain positive results. On the other hand, while most recent research on relation extraction focuses on semantic relations occurring within one sentence, extracting inter-sentential relations that occur across multiple sentences pose a more challenging problem. It is estimated that 28.5% of MUC6 relations and 9.4% of ACE 2003 relations are inter-sentential [Swampillai and Stevenson 2010], though the ACE project since 2004 limits to annotating relations which are expressed within a single sentence. While some of these inter-sentential relations may be automatically extracted using syntax and co-reference information, others are more difficult and can only be resolved using real-world background knowledge. Therefore, further research is necessary to address the issue of inter-sentential relation extraction.

REFERENCES

ACE. 2002–2008. Automatic content extraction. http://www.ldc.upenn.edu/Projects/ACE/.

BUNESCU, R. C. AND MOONEY, R. J. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (EMNLP'05)*. 724–731.

CHANG, P. C., TSENG, H., JURAFSKY, D., AND CHRISTOPHER, D. M. 2009. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the 3rd Workshop on Syntax and Structure in Statistical Translation (SSST'09)*. 51–59.

CHARNIAK, E. 2001. Intermediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL'01)*. 116–123.

CHE, W. X., JIANG, J. M., SU, Z., PAN, Y., AND LIU, T. 2005a. Improved-edit-distance kernel for Chinese relation extraction. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'05)*.

CHE, W. X., LIU, T., AND LI, S. 2005b. Automatic entity relation extraction. *J. Chi. Inf. Proc. 19*, 2, 1–6.

COLLINS, M. 2003. Head-driven statistics models for natural language parsing. *Comput. Linguist. 29*, 4, 589–617.

COLLINS, M. AND DUFFY, N. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS'01)*. 625–632.

COLLINS, M. AND DUFFY, N. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics (ACL'02)*.

CULOTTA, A. AND SORENSEN, J. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics (ACL'04)*. 423–439.

DONG, J., SUN, L., FENG, Y. H., AND HUANG, R. H. 2007. Chinese automatic entity relation extraction. *J. Chi. Inf. Proc. 21*, 4, 80–85, 91.

GRISHMAN, R. AND SUNDHEIM, B. 1996. Message understanding Conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics (COLING'96)*. 466–471.

HAUSSLER, D. 1999. Convolution Kernels on Discrete Structures. Tech. rep. UCS-CRL-99-10, University of California, Santa Cruz.

HUANG, R. H., SUN, L., AND FENG, Y. Y. 2008. Study of kernel-based methods for Chinese relation extraction. *Lecture Notes in Computer Science 4993*: 598–604. Springer: Berlin/Heidelberg.

JIANG, J. AND ZHAI, C. X. 2007. A systematic exploration of the feature space for relation extraction. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT-NAACL'07)*. 113–120.

JOACHIMS, T. 1998. Text categorization with support vector machine: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*. 137–142.

KAMBHATLA, N. 2004. Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'04)*. 178–181.

LI, W. J., ZHANG, P., WEI, F. R., HOU, Y. X., AND LU, Q. 2008. A novel feature-based approach to Chinese entity relation extraction. In *Proceedings of the Human Language Technology Conference (HLT'08)*. 89–92.

LIU, K. B., LI, F., LIU, L., AND HANG, Y. 2007. Implementation of a kernel-based Chinese relation extraction system. *J. Comput. Res. Dev. 44*, 8, 1406–1411.

LODHI, H., SAUNDERS, C., SHAW-TAYLOR, J., CRISTIANINI, N., AND WATKINS, C. 2002. Text classification using string kernel. *J. Mach. Learn. Res. 2*, 419–444.

MOSCHITTI, A. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics (ACL'04)*.

MOSCHITTI, A. 2006. Making tree kernels practical for natural language learning. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*. 113–120.

MOSCHITTI, A., PIGHIN, D., AND BASILI, R. 2006. Tree kernel engineering in semantic role labeling systems. In *Proceedings of the Workshop on Learning Structured Information for Natural Language Applications, the Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*. 49–56.

MOSCHITTI, A., PIGHIN, D., AND BASILI, R. 2008. Tree kernels for semantic role labeling, special issue on semantic role labeling. *Comput. Linguist. 34*, 2, 194–224.

MUC. 1987–1998. Available online at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

MUC-7. 1998. *Proceedings of the 7th Message Understanding Conference (MUC'98)*. Morgan Kaufmann, San Mateo, CA.

NGUYEN, T. T., MOSCHITTI, A., AND RICCARDI, G. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'09)*. 1378–1387.

QIAN, L. H., ZHOU, G. D. ZHU, Q. M., AND QIAN, P. D. 2007. Relation extraction using convolution tree kernel expanded with entity features. In *Proceedings of the 21st Pacific Asian Conference on Language, Information and Computation (PACLIC'07)*. 415–421.

SCHÖLKOPF, B. AND SMOLA, A. J. 2001. *Learning with Kernels: SVM, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 407–423.

SUZUKI, J., HIRAO, T., SASAKI, Y., AND MAEDA, E. 2003. Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'03)*. 32–39.

SWAMPILLAI, K. AND STEVENSON, M. 2010. Inter-sentential relations in information extraction corpora. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*. 2637–2641.

YU, H. H., QIAN, L. H., ZHOU, G. D., AND ZHU, Q. M. 2010. Chinese semantic relation extraction based on unified syntactic and entity semantic tree. *J. Chi. Inf. Proc. 24*, 5, 17–23.

ZELENKO, D., AONE, C., AND RICHARDELLA, A. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res. 3*, 1083–1106.

ZHANG, M., ZHANG, M., SU, J., AND ZHOU, G. D. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association of Computational Linguistics (COLING/ACL'06)*. 825–832.

ZHANG, M., ZHOU, G. D., AND AW, A. T. 2008a. Exploring syntactic structured features over parse trees for relation extraction using kernel methods. *Inf. Proc. Man. 44*: 687–701.

ZHANG, M., CHE, W. X., ZHOU, G. D., AW, A. T., TAN, C. L., LIU, T., AND LI, S. 2008b. Semantic role labeling using a grammar-driven convolution tree kernel. *IEEE Trans. Audio, Speech Lang. Proc. 16*, 7, 1315–1329.

ZHANG, J., OUYANG, Y., LI, Y., AND HOU, Y. X. 2009. A novel composite approach to Chinese relation extraction. In *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages (ICCPOL'09)*.

ZHAO, S. B. AND GRISHMAN, R. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association of Computational Linguistics (ACL'05)*. 419–426.

ZHOU, G. D. AND ZHANG, M. 2007. Extracting relation information from text documents by exploring various types of knowledge. *Inf. Proc. Man. 43*, 969–982.

ZHOU, G. D., SU, J., ZHANG, J., AND ZHANG, M. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association of Computational Linguistics (ACL'05)*. 427–434.

ZHOU, G. D., ZHANG, M., JI, D. H., AND ZHU, Q. M. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL'07)*. 728–736.

ZHOU, G. D., QIAN, L. H., AND ZHU, Q. M. 2009. Label propagation via bootstrapped support vectors for semantic relation extraction between named entities. *Comput. Speech Lang. 23*, 4, 464–478.

Received November 2010; revised February 2011; accepted April 2011