

Dependency-driven Anaphoricity Determination for Coreference Resolution

Fang Kong Guodong Zhou Longhua Qian Qiaoming Zhu *

JiangSu Provincial Key Lab for Computer Information Processing Technology

School of Computer Science and Technology Soochow University

{kongfang, gdzhou, qianlonghua, qmzhu}@suda.edu.cn

Abstract

This paper proposes a dependency-driven scheme to dynamically determine the syntactic parse tree structure for tree kernel-based anaphoricity determination in coreference resolution. Given a full syntactic parse tree, it keeps the nodes and the paths related with current mention based on constituent dependencies from both syntactic and semantic perspectives, while removing the noisy information, eventually leading to a dependency-driven dynamic syntactic parse tree (D-DSPT). Evaluation on the ACE 2003 corpus shows that the D-DSPT outperforms all previous parse tree structures on anaphoricity determination, and that applying our anaphoricity determination module in coreference resolution achieves the so far best performance.

1 Introduction

Coreference resolution aims to identify which noun phrases (NPs, or mentions) refer to the same real-world entity in a text. According to Webber (1979), coreference resolution can be decomposed into two complementary sub-tasks: (1) anaphoricity determination, determining whether a given NP is anaphoric or not; and (2) anaphor resolution, linking together multiple mentions of a given entity in the world. Although machine learning approaches have performed reasonably well in coreference resolution without explicit anaphoricity determination (e.g. Soon et al. 2001; Ng and Cardie 2002b; Yang et al. 2003, 2008; Kong et al. 2009), knowledge of NP anaphoricity is expected to much improve the performance of a coreference resolution system, since a

non-anaphoric NP does not have an antecedent and therefore does not need to be resolved.

Recently, anaphoricity determination has been drawing more and more attention. One common approach involves the design of some heuristic rules to identify specific types of non-anaphoric NPs, such as pleonastic *it* (e.g. Paice and Husk 1987; Lappin and Leass 1994, Kennedy and Boguraev 1996; Denber 1998) and definite descriptions (e.g. Vieira and Poesio 2000). Alternatively, some studies focus on using statistics to tackle this problem (e.g., Bean and Riloff 1999; Bergsma et al. 2008) and others apply machine learning approaches (e.g. Evans 2001; Ng and Cardie 2002a, 2004, 2009; Yang et al. 2005; Denis and Balbridge 2007; Luo 2007; Finkel and Manning 2008; Zhou and Kong 2009).

As a representative, Zhou and Kong (2009) directly employ a tree kernel-based method to automatically mine the non-anaphoric information embedded in the syntactic parse tree. One main advantage of the kernel-based methods is that they are very effective at reducing the burden of feature engineering for structured objects. Indeed, the kernel-based methods have been successfully applied to mine structured information in various NLP applications like syntactic parsing (Collins and Duffy, 2001; Moschitti, 2004), semantic relation extraction (Zelenko et al., 2003; Zhao and Grishman, 2005; Zhou et al. 2007; Qian et al., 2008), semantic role labeling (Moschitti, 2004); coreference resolution (Yang et al., 2006; Zhou et al., 2008). One of the key problems for the kernel-based methods is how to effectively capture the structured information according to the nature of the structured object in the specific task.

This paper advances the state-of-the-art performance in anaphoricity determination by ef-

* Corresponding author

fectively capturing the structured syntactic information via a tree kernel-based method. In particular, a dependency-driven scheme is proposed to dynamically determine the syntactic parse tree structure for tree kernel-based anaphoricity determination by exploiting constituent dependencies from both the syntactic and semantic perspectives to keep the necessary information in the parse tree as well as remove the noisy information. Our motivation is to employ critical dependency information in constructing a concise and effective syntactic parse tree structure, specifically targeted for tree kernel-based anaphoricity determination.

The rest of this paper is organized as follows. Section 2 briefly describes the related work on both anaphoricity determination and exploring syntactic parse tree structures in related tasks. Section 3 presents our dependency-driven scheme to determine the syntactic parse tree structure. Section 4 reports the experimental results. Finally, we conclude our work in Section 5.

2 Related Work

This section briefly overviews the related work on both anaphoricity determination and exploring syntactic parse tree structures.

2.1 Anaphoricity Determination

Previous work on anaphoricity determination can be broadly divided into three categories: heuristic rule-based (e.g. Paice and Husk 1987; Lappin and Leass 1994; Kennedy and Boguraev 1996; Denber 1998; Vieira and Poesio 2000; Cherry and Bergsma 2005), statistics-based (e.g. Bean and Riloff 1999; Cherry and Bergsma 2005; Bergsma et al. 2008) and learning-based methods (e.g. Evans 2001; Ng and Cardie 2002a; Ng 2004; Yang et al. 2005; Denis and Balbridge 2007; Luo 2007; Finkel and Manning 2008; Zhou and Kong 2009; Ng 2009).

The heuristic rule-based methods focus on designing some heuristic rules to identify specific types of non-anaphoric NPs. Representative work includes: Paice and Husk (1987), Lappin and Leass (1994) and Kennedy and Boguraev (1996). For example, Kennedy and Boguraev (1996) looked for modal adjectives (e.g. “necessary”) or cognitive verbs (e.g. “It is

thought that...”) in a set of patterned constructions) in identifying pleonastic *it*.

Among the statistics-based methods, Bean and Riloff (1999) automatically identified existential definite NPs which are non-anaphoric. The intuition behind is that many definite NPs are not anaphoric since their meanings can be understood from general world knowledge, e.g. “the FBI”. They found that existential NPs account for 63% of all definite NPs and 76% of them could be identified by syntactic or lexical means. Cherry and Bergsma (2005) extended the work of Lappin and Leass (1994) for large-scale anaphoricity determination by additionally detecting pleonastic *it*. Bergsma et al. (2008) proposed a distributional method in detecting non-anaphoric pronouns. They first extracted the surrounding context of the pronoun and gathered the distribution of words that occurred within the context from a large corpus, and then identified the pronoun either anaphoric or non-anaphoric based on the word distribution.

Among the learning-based methods, Evans (2001) automatically identified the non-anaphoricity of pronoun *it* using various kinds of lexical and syntactic features. Ng and Cardie (2002a) employed various domain-independent features in identifying anaphoric NPs. They trained an anaphoricity classifier to determine whether a NP was anaphoric or not, and employed an independently-trained coreference resolution system to only resolve those mentions which were classified as anaphoric. Experiments showed that their method improved the performance of coreference resolution by 2.0 and 2.6 to 65.8 and 64.2 in F1-measure on the MUC-6 and MUC-7 corpora, respectively. Ng (2004) examined the representation and optimization issues in computing and using anaphoricity information to improve learning-based coreference resolution. On the basis, he presented a corpus-based approach (Ng, 2009) for achieving global optimization by representing anaphoricity as a feature in coreference resolution. Experiments on the ACE 2003 corpus showed that their method improved the overall performance by 2.8, 2.2 and 4.5 to 54.5, 64.0 and 60.8 in F1-measure on the NWIRE, NPAPER and BNEWS domains, respectively. However, he did not look into the contribution of anaphoricity determi-

nation on coreference resolution of different NP types. Yang et al. (2005) made use of non-anaphors to create a special class of training instances in the twin-candidate model (Yang et al. 2003) and improved the performance by 2.9 and 1.6 to 67.3 and 67.2 in F1-measure on the MUC-6 and MUC-7 corpora, respectively. However, their experiments show that eliminating non-anaphors using an anaphoricity determination module in advance harms the performance. Denis and Balbridge (2007) employed an integer linear programming (ILP) formulation for coreference resolution which modeled anaphoricity and coreference as a joint task, such that each local model informed the other for the final assignments. Experiments on the ACE 2003 corpus showed that this joint anaphoricity-coreference ILP formulation improved the F1-measure by 3.7-5.3 on various domains. However, their experiments assume true ACE mentions (i.e. all the ACE mentions are already known from the annotated corpus). Therefore, the actual effect of this joint anaphoricity-coreference ILP formulation on fully automatic coreference resolution is still unclear. Luo (2007) proposed a twin-model for coreference resolution: a link component, which models the coreferential relationship between an anaphor and a candidate antecedent, and a creation component, which models the possibility that a NP was not coreferential with any candidate antecedent. This method combined the probabilities returned by the creation component (an anaphoricity model) with the link component (a coreference model) to score a coreference partition, such that a partition was penalized whenever an anaphoric mention was resolved. Finkel and Manning (2008) showed that transitivity constraints could be incorporated into an ILP-based coreference resolution system and much improved the performance. Zhou and Kong (2009) employed a global learning method in determining the anaphoricity of NPs via a label propagation algorithm to improve learning-based coreference resolution. Experiments on the ACE 2003 corpus demonstrated that this method was very effective. It could improve the F1-measure by 2.4, 3.1 and 4.1 on the NWIRE, NPAPER and BNEWS domains, respectively. Ng (2009) presented a novel approach to the task of anaphoricity determina-

tion based on graph minimum cuts and demonstrated the effectiveness in improving a learning-based coreference resolution system.

In summary, although anaphoricity determination plays an important role in coreference resolution and achieves certain success in improving the overall performance of coreference resolution, its contribution is still far from expectation.

2.2 Syntactic Parse Tree Structures

For a tree kernel-based method, one key problem is how to represent and capture the structured syntactic information. During recent years, various tree kernels, such as the convolution tree kernel (Collins and Duffy, 2001), the shallow parse tree kernel (Zelenko et al 2003) and the dependency tree kernel (Culota and Sorensen, 2004), have been proposed in the literature. Among these tree kernels, the convolution tree kernel represents the state-of-the-art and has been successfully applied by Collins and Duffy (2002) on syntactic parsing, Zhang et al. (2006) on semantic relation extraction and Yang et al. (2006) on pronoun resolution.

Given a tree kernel, the key issue is how to generate a syntactic parse tree structure for effectively capturing the structured syntactic information. In the literature, various parse tree structures have been proposed and successfully applied in some NLP applications. As a representative, Zhang et al. (2006) investigated five parse tree structures for semantic relation extraction and found that the Shortest Path-enclosed Tree (SPT) achieves the best performance on the 7 relation types of the ACE RDC 2004 corpus. Yang et al. (2006) constructed a document-level syntactic parse tree for an entire text by attaching the parse trees of all its sentences to a new-added upper node and examined three possible parse tree structures (Min-Expansion, Simple-Expansion and Full-Expansion) that contain different substructures of the parse tree for pronoun resolution. Experiments showed that their method achieved certain success on the ACE 2003 corpus and the simple-expansion scheme performs best. However, among the three explored schemes, there exists no obvious overwhelming one, which can well cover structured syntactic information. One problem of Zhang et al. (2006)

and Yang et al. (2006) is that their parse tree structures are context-free and do not consider the information outside the sub-trees. Hence, their ability of exploring structured syntactic information is much limited. Motivated by Zhang et al. (2006) and Yang et al. (2006), Zhou et al. (2007) extended the SPT to become context-sensitive (CS-SPT) by dynamically including necessary predicate-linked path information. Zhou et al. (2008) further proposed a dynamic-expansion scheme to automatically determine a proper parse tree structure for pronoun resolution by taking predicate- and antecedent competitor-related information in consideration. Evaluation on the ACE 2003 corpus showed that the dynamic-expansion scheme can well cover necessary structured information in the parse tree for pronoun resolution. One problem with the above parse tree structures is that they may still contain unnecessary information and also miss some useful context-sensitive information. Qian et al. (2008) dynamically determined the parse tree structure for semantic relation extraction by exploiting constituent dependencies to keep the necessary information in the parse tree as well as remove the noisy information. Evaluation on the ACE RDC 2004 corpus showed that their dynamic syntactic parse tree structure outperforms all previous parse tree structures. However, their solution has the limitation in that the dependencies were found according to some manually-written ad-hoc rules and thus may not be easily applicable to new domains and applications.

This paper proposes a new scheme to dynamically determine the syntactic parse tree structure for anaphoricity determination and systematically studies the application of an explicit anaphoricity determination module in improving coreference resolution.

3 Dependency-driven Dynamic Syntactic Parse Tree

Given a full syntactic parse tree and a NP in consideration, one key issue is how to choose a proper syntactic parse tree structure to well cover structured syntactic information in the tree kernel computation. Generally, the more a syntactic parse tree structure includes, the more structured syntactic information would be

available, at the expense of more noisy (or unnecessary) information.

It is well known that dependency information plays a key role in many NLP problems, such as syntactic parsing, semantic role labeling as well as semantic relation extraction. Motivated by Qian et al. (2008) and Zhou et al. (2008), we propose a new scheme to dynamically determine the syntactic parse tree structure for anaphoricity determination by exploiting constituent dependencies from both the syntactic and semantic perspectives to distinguish the necessary evidence from the unnecessary information in the syntactic parse tree. That is, constituent dependencies are explored from two aspects: syntactic dependencies and semantic dependencies.

1) Syntactic Dependencies: The Stanford dependency parser¹ is employed as our syntactic dependency parser to automatically extract various syntactic (i.e. grammatical) dependencies between individual words. In this paper, only immediate syntactic dependencies with current mention are considered. The intuition behind is that the immediate syntactic dependencies carry the major contextual information of current mention.

2) Semantic Dependencies: A state-of-the-art semantic role labeling (SRL) toolkit (Li et al. 2009) is employed for extracting various semantic dependencies related with current mention. In this paper, semantic dependencies include all the predicates heading any node in the root path from current mention to the root node and their compatible arguments (except those overlapping with current mention).

We name our parse tree structure as a dependency-driven dynamic syntactic parse tree (D-DSPT). The intuition behind is that the dependency information related with current mention in the same sentence plays a critical role in anaphoricity determination. Given the sentence enclosing the mention under consideration, we can get the D-DSPT as follows: (Figure 1 illustrates an example of the D-DSPT generation given the sentence “Mary said the woman in the room bit her” with “woman” as current mention.)

¹ <http://nlp.stanford.edu/software/lex-parser.shtml>

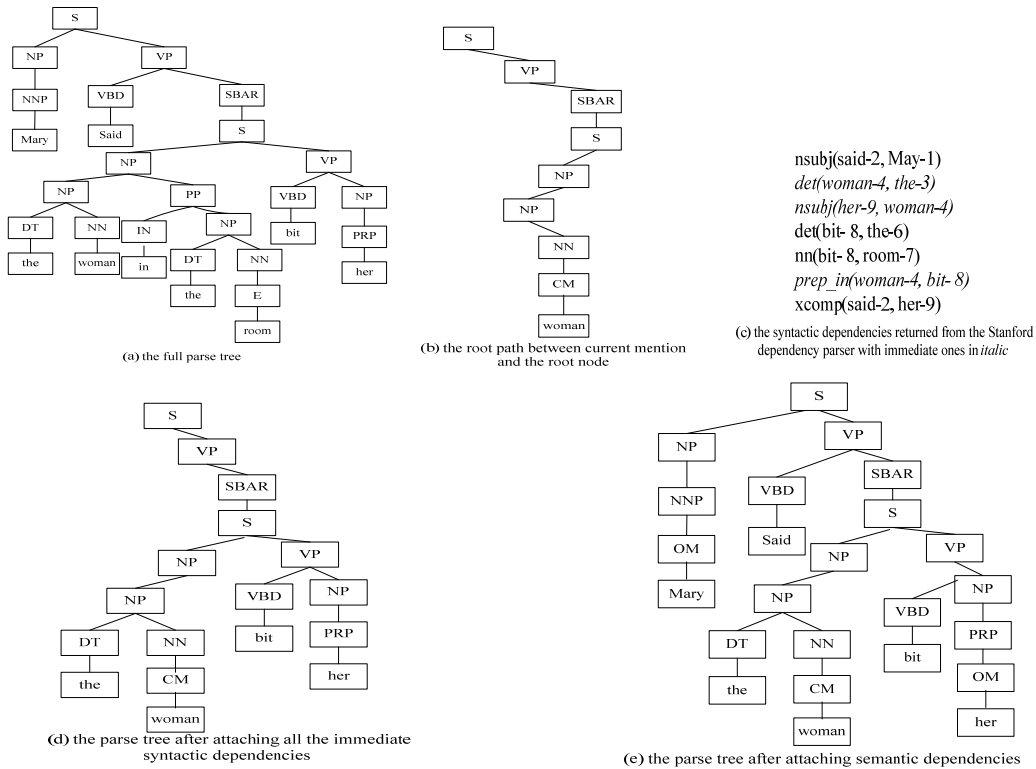


Figure 1: An example of generating the dependency-driven dynamic syntactic parse tree

- 1) Generating the full syntactic parse tree of the given sentence using a full syntactic parser. In this paper, the Charniak parser (Charniak 2001) is employed and Figure 1 (a) shows the resulting full parse tree.
- 2) Keeping only the root path from current mention to the root node of the full parse tree. Figure 1(b) shows the root path corresponding to the current mention “woman”. In the following steps, we attach the above two types of dependency information to the root path.
- 3) Extracting all the syntactic dependencies in the sentence using a syntactic dependency parser, and attaching all the nodes, which have immediate dependency relationship with current mention, and their corresponding paths to the root path. Figure 1(c) illustrates the syntactic dependences extracted from the sentence, where the ones in italic mean immediate dependencies with current mention. Figure 1(d) shows the parse tree structure after considering syntactic dependencies.
- 4) Attaching all the predicates heading any node in the root path from current mention to the root node and their corresponding paths to the root path. For the example sentence, there are two predicates “said” and “bit”, which head the “VP” and “S” nodes in the root path re-

spectively. Therefore, these two predicates and their corresponding paths should be attached to the root path as shown in Figure 1(e). Note that the predicate “bit” and its corresponding path has already been attached in Stop (3). As a result, the predicate-related information can be attached. According to Zhou and Kong (2009), such information is important to definite NP resolution.

- 5) Extracting the semantic dependencies related with those attached predicates using a (shallow) semantic parser, and attaching all the compatible arguments (except those overlapping with current mention) and their corresponding paths to the root path. For example, as shown in Figure 1(e), since the arguments “Mary” and “her” are compatible with current mention “woman”, these two nodes and their corresponding paths are attached while the argument “room” is not since its gender does not agree with current mention.

In this paper, the similarity between two parse trees is measured using a convolution tree kernel, which counts the number of common sub-tree as the syntactic structure similarity between two parse trees. For details, please refer to Collins and Duffy (2001).

4 Experimentation and Discussion

This section evaluates the performance of dependency-driven anaphoricity determination and its application in coreference resolution on the ACE 2003 corpus.

4.1 Experimental Setting

The ACE 2003 corpus contains three domains: newswire (NWIRE), newspaper (NPAPER), and broadcast news (BNEWS). For each domain, there exist two data sets, training and devtest, which are used for training and testing.

For preparation, all the documents in the corpus are preprocessed automatically using a pipeline of NLP components, including tokenization and sentence segmentation, named entity recognition, part-of-speech tagging and noun phrase chunking. Among them, named entity recognition, part-of-speech tagging and noun phrase chunking apply the same state-of-the-art HMM-based engine with error-driven learning capability (Zhou and Su, 2000 & 2002). Our statistics finds that 62.0%, 58.5% and 61.4% of entity mentions are preserved after preprocessing on the NWIRE, NPAPER and BNEWS domains of the ACE 2003 training data respectively while only 89.5%, 89.2% and 94% of entity mentions are preserved after preprocessing on the NWIRE, NPAPER and BNEWS domains of the ACE 2003 devtest data. This indicates the difficulty of coreference resolution. In addition, the corpus is parsed using the Charniak parser for syntactic parsing and the Stanford dependency parser for syntactic dependencies while corresponding semantic dependencies are extracted using a state-of-the-art semantic role labeling toolkit (Li et al. 2009). Finally, we use the SVM-light² toolkit with the tree kernel function as the classifier. For comparison purpose, the training parameters C (SVM) and λ (tree kernel) are set to 2.4 and 0.4 respectively, as done in Zhou and Kong (2009).

For anaphoricity determination, we report the performance in Acc^+ and Acc^- , which measure the accuracies of identifying anaphoric NPs and non-anaphoric NPs, respectively. Obviously, higher Acc^+ means that more anaphoric NPs would be identified correctly, while

higher Acc^- means that more non-anaphoric NPs would be filtered out. For coreference resolution, we report the performance in terms of recall, precision, and F1-measure using the commonly-used model theoretic MUC scoring program (Vilain et al. 1995). To see whether an improvement is significant, we also conduct significance testing using paired t-test. In this paper, ‘****’, ‘***’ and ‘**’ denote p-values of an improvement smaller than 0.01, in-between (0.01, 0.05] and bigger than 0.05, which mean significantly better, moderately better and slightly better, respectively.

4.2 Experimental Results

Performance of anaphoricity determination

Table 1 presents the performance of anaphoricity determination using the convolution tree kernel on D-DSPT. It shows that our method achieves the accuracies of 83.27/77.13, 86.77/80.25 and 90.02/64.24 on identifying anaphoric/non-anaphoric NPs in the NWIRE, NPAPER and BNEWS domains, respectively. This suggests that our approach can effectively filter out about 75% of non-anaphoric NPs and keep about 85% of anaphoric NPs. In comparison, in the three domains Zhou and Kong (2009) achieve the accuracies of 76.5/82.3, 78.9/81.6 and 74.3/83.2, respectively, using the tree kernel on a dynamically-extended tree (DET). This suggests that their method can filter out about 82% of non-anaphoric NPs and only keep about 76% of anaphoric NPs. In comparison, their method outperforms our method on filtering out more non-anaphoric NPs while our method outperforms their method on keeping more anaphoric NPs in coreference resolution. While a coreference resolution system can detect some non-anaphoric NPs (when failing to find the antecedent candidate), filtering out anaphoric NPs in anaphoricity determination would definitely cause errors and it is almost impossible to recover. Therefore, it is normally more important to keeping more anaphoric NPs than filtering out more non-anaphoric NPs. Table 1 further presents the performance of anaphoricity determination on different NP types. It shows that our method performs best at keeping pronominal NPs and filtering out proper NPs.

² <http://svmlight.joachims.org/>

NP Type	NWIRE		NPAPER		BNEWS	
	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻
Pronoun	95.07	50.36	96.40	56.44	98.26	54.03
Proper NP	84.61	83.17	83.78	79.62	87.61	71.77
Definite NP	87.17	46.74	82.24	49.18	86.87	53.65
Indefinite NP	86.01	47.52	80.63	48.45	89.71	47.32
Over all	83.27	77.13	86.77	80.25	90.02	64.24

Table 1: Performance of anaphoricity determination using the D-DSPT

Performance Change	NWIRE		NPAPER		BNEWS	
	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻
D-DSPT	83.27	77.13	86.77	80.25	90.02	64.24
-Syntactic Dependencies	78.67	72.56	80.14	73.74	87.05	60.20
-Semantic Dependencies	81.67	76.74	83.47	77.93	89.58	60.67

Table 2: Contribution of including syntactic and semantic dependencies in D-DSPT on anaphoricity determination

System		NWIRE			NPAPER			BNEWS		
		R%	P%	F	R%	P%	F	R%	P%	F
Without anaphoricity determination (Baseline)	Pronoun	70.8	57.9	63.7	76.5	63.5	69.4	70.0	60.3	64.8
	Proper NP	80.3	80.1	80.2	81.8	83.6	82.7	76.3	76.8	76.6
	Definite NP	35.9	43.4	39.2	43.1	48.5	45.6	47.9	51.9	49.8
	Indefinite NP	40.3	26.3	31.8	39.7	22.9	29.0	23.6	10.7	14.7
	Over all	55.0	63.8	59.1	62.1	65.0	63.5	53.2	60.5	56.6
With D-DSPT-based anaphoricity determination	Pronoun	65.9	70.2	68.0	72.6	78.7	75.5	67.7	75.8	71.5
	Proper NP	80.3	81.0	80.6	81.2	85.1	83.1	76.3	84.4	80.1
	Definite NP	32.3	63.1	42.7	38.4	61.7	47.3	42.5	66.4	51.8
	Indefinite NP	36.4	55.3	43.9	34.7	50.7	41.2	20.3	45.4	28.1
	Over all	52.4	79.6	63.2	58.1	80.3	67.4	50.1	79.8	61.6
With golden anaphoricity determination	Pronoun	68.6	71.5	70.1	75.2	80.4	77.7	69.1	77.8	73.5
	Proper NP	81.7	89.3	85.3	82.6	90.1	86.2	78.6	88.7	83.3
	Definite NP	41.8	85.9	56.2	44.9	85.2	58.8	45.2	87.9	59.7
	Indefinite NP	40.3	67.6	50.5	41.2	65.1	50.5	40.9	50.1	45.1
	Over all	54.6	81.7	65.5	60.4	82.1	69.6	51.9	82.1	63.6

Table 3: Performance of anaphoricity determination on coreference resolution

System		NWIRE			NPAPER			BNEWS		
		R%	P%	F	R%	P%	F	R%	P%	F
Zhou and Kong (2009)	Without anaphoricity determination (Baseline)	53.1	67.4	59.4	57.7	67.0	62.1	48.0	65.9	55.5
	With Dynamically Extended Tree-based anaphoricity determination	51.6	77.2	61.8	55.2	78.6	65.2	47.5	80.3	59.6
Ng (2009)	Without anaphoricity determination (Baseline)	59.1	58.	58.6	60.8	62.6	61.7	57.7	52.6	55.0
	With Graph Minimum Cut-based anaphoricity determination	54.1	69.0	60.6	57.9	71.2	63.9	53.1	67.5	59.4

Table 4: Performance comparison with other systems

Table 2 further presents the contribution of including syntactic and semantic dependencies in the D-DSPT on anaphoricity determination by excluding one or both of them. It shows that both syntactic dependencies and semantic dependencies contribute significantly (***)

Performance of coreference resolution

We have evaluated the effect of our D-DSPT-based anaphoricity determination module on coreference resolution by including

it as a preprocessing step to a baseline coreference resolution system without explicit anaphoricity determination, by filtering out those non-anaphoric NPs according to the anaphoricity determination module. Here, the baseline system employs the same set of features, as adopted in the single-candidate model of Yang et al. (2003) and uses a SVM-based classifier with the feature-based RBF kernel. Table 3 presents the detailed performance of the coreference resolution system without ana-

phoricity determination, with D-DSPT-based anaphoricity determination and. with golden anaphoricity determination. Table 3 shows that:

1) There is a performance gap of 6.4, 6.1 and 7.0 in F1-measure on the NWIRE, NPAPER and BNEWS domain, respectively, between the coreference resolution system with golden anaphoricity determination and the baseline system without anaphoricity determination. This suggests the usefulness of proper anaphoricity determination in coreference resolution. This also agrees with Stoyanov et al. (2009) which measured the impact of golden anaphoricity determination on coreference resolution using only the annotated anaphors in both training and testing.

2) Compared to the baseline system without anaphoricity determination, the D-DSPT-based anaphoricity determination module improves the performance by 4.1(***), 3.9(***) and 5.0(***) to 63.2, 67.4 and 61.6 in F1-measure on the NWIRE, NPAPER and BNEWS domains, respectively, due to a large gain in precision and a much smaller drop in recall. In addition, D-DSPT-based anaphoricity determination can not only much improve the performance of coreference resolution on pronominal NPs (***) but also on definite NPs(***) and indefinite NPs(***) while the improvement on proper NPs can be ignored due to the fact that proper NPs can be well addressed by the simple abbreviation feature in the baseline system.

3) D-DSPT-based anaphoricity determination still lags (2.3, 2.2 and 2.0 on the NWIRE, NPAPER and BNEWS domains, respectively) behind golden anaphoricity determination in improving the overall performance of coreference resolution. This suggests that there exists some room in the performance improvement for anaphoricity determination.

Performance comparison with other systems

Table 4 compares the performance of our system with other systems. Here, Zhou and Kong (2009) use the same set of features with ours in the baseline system and a dynamically-extended tree structure in anaphoricity determination. Ng (2009) uses 33 features as described in Ng (2007) and a graph minimum cut algorithm in anaphoricity determination. It shows that the overall performance of our

baseline system is almost as good as that of Zhou and Kong (2009) and a bit better than Ng's (2009).

For overall performance, our coreference resolution system with D-DSPT-based anaphoricity determination much outperforms Zhou and Kong (2009) in F1-measure by 1.4, 2.2 and 2.0 on the NWIRE, NPAPER and BNEWS domains, respectively, due to the better inclusion of dependency information. Detailed evaluation shows that such improvement comes from coreference resolution on both pronominal and definite NPs (Please refer to Table 6 in Zhou and Kong, 2009). Compared with Zhou and Kong (2009) and Ng (2009), our approach achieves the best F1-measure so far for each dataset.

5 Conclusion and Further Work

This paper systematically studies a dependency-driven dynamic syntactic parse tree (DDST) for anaphoricity determination and the application of an explicit anaphoricity determination module in improving learning-based coreference resolution. Evaluation on the ACE 2003 corpus indicates that D-DSPT-based anaphoricity determination much improves the performance of coreference resolution.

To our best knowledge, this paper is the first research which directly explores constituent dependencies in tree kernel-based anaphoricity determination from both syntactic and semantic perspectives.

For further work, we will explore more structured syntactic information in coreference resolution. In addition, we will study the interaction between anaphoricity determination and coreference resolution and better integrate anaphoricity determination with coreference resolution.

Acknowledgments

This research was supported by Projects 60873150, 60970056, and 90920004 under the National Natural Science Foundation of China, Project 200802850006 and 20093201110006 under the Specialized Research Fund for the Doctoral Program of Higher Education of China.

References

- D. Bean and E. Riloff 1999. Corpus-based Identification of Non-Anaphoric Noun Phrases. ACL'1999
- S. Bergsma, D. Lin and R. Goebel 2008. Distributional Identification of Non-referential Pronouns. ACL'2008
- C. Cherry and S. Bergsma. 2005. An expectation maximization approach to pronoun resolution. CoNLL'2005
- M. Collins and N. Duffy. 2001. Covolution kernels for natural language. NIPS' 2001
- M. Denber 1998. Automatic Resolution of Anaphora in English. Technical Report, Eastman KodakCo.
- P. Denis and J. Baldridge. 2007. Global, joint determination of anaphoricity and coreference resolution using integer programming. NAACL/HLT'2007
- R. Evans 2001. Applying machine learning toward an automatic classification of it. Literary and Linguistic Computing, 16(1):45-57
- F. Kong, G.D. Zhou and Q.M. Zhu. 2009 Employing the Centering Theory in Pronoun Resolution from the Semantic Perspective. EMNLP'2009
- F. Kong, Y.C. Li, G.D. Zhou and Q.M. Zhu. 2009. Exploring Syntactic Features for Pronoun Resolution Using Context-Sensitive Convolution Tree Kernel. IALP'2009
- S. Lappin and J. L. Herbert. 1994. An algorithm for pronominal anaphora resolution. Computational Linguistics, 20(4)
- J.H. Li, G.D. Zhou, H. Zhao, Q.M. Zhu and P.D. Qian. Improving nominal SRL in Chinese language with verbal SRL information and automatic predicate recognition. EMNLP '2009
- X. Luo. 2007. Coreference or not: A twin model for coreference resolution. NAACL-HLT'2007
- V. Ng and C. Cardie 2002. Identify Anaphoric and Non-Anaphoric Noun Phrases to Improve Coreference Resolution. COLING'2002
- V. Ng and C. Cardie 2002. Improving machine learning approaches to coreference resolution. ACL'2002
- V. Ng 2004. Learning Noun Phrase Anaphoricity to Improve Coreference Resolution: Issues in Representation and Optimization. ACL' 2004
- V. Ng 2009. Graph-cut based anaphoricity determination for coreference resolution. NAACL'2009
- L.H. Qian, G.D. Zhou, F. Kong, Q.M. Zhu and P.D. Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. COLING'2008
- W. M. Soon, H. T. Ng and D. Lim 2001. A machine learning approach to coreference resolution of noun phrase. Computational Linguistics, 27(4):521-544.
- V. Stoyanov, N. Gilbert, C. Cardie and E. Riloff. 2009. Conundrums in Noun Phrase Coreference Resolution: Making Sense of the State-of-the Art. ACL'2009
- B. L. Webber. 1979. A Formal Approach to Discourse Anaphora. Garland Publishing, Inc.
- X.F. Yang, G.D. Zhou, J. Su and C.L. Chew. 2003. Coreference Resolution Using Competition Learning Approach. ACL'2003
- X.F. Yang, J. Su and C.L. Chew. 2005. A Twin Candidate Model of Coreference Resolution with Non-Anaphor Identification Capability. IJCNLP'2005
- X.F. Yang, J. Su and C.L. Chew. 2006. Kernel-based pronoun resolution with structured syntactic knowledge. COLING-ACL'2006
- X.F. Yang, J. Su and C.L. Tan 2008. A Twin-Candidate Model for Learning-Based Anaphora Resolution. Computational Linguistics 34(3):327-356
- M. Zhang, J. Zhang, J. Su and G.D. Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. COLING/ACL'2006
- S. Zhao and R. Grishman. 2005. Extracting relations with integrated information using kernel methods. ACL'2005
- D. Zelenko, A. Chinatsu and R. Anthony. 2003. Kernel methods for relation extraction. Machine Learning Researching 3(2003):1083-1106
- G.D. Zhou, F. Kong and Q.M. Zhu. 2008. Context-sensitive convolution tree kernel for pronoun resolution. IJCNLP'2008
- G.D. Zhou and F. Kong. 2009. Global Learning of Noun Phrase Anaphoricity in Coreference Resolution via Label Propagation. EMNLP'2009
- G.D. Zhou and J. Su. 2002. Named Entity recognition using a HMM-based chunk tagger. ACL'2002
- G.D. Zhou, M. Zhang, D.H. Ji and Q.M. Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. EMNLP/CoNLL'2007