

Combining Dependency and Constituent-based Syntactic Information for Anaphoricity Determination in Coreference Resolution

Fang Kong^{a,b} and Guodong Zhou^{a,b}

^a JiangSu Provincial Key Lab for Computer Information Processing Technology

^b School of Computer Science and Technology Soochow University, Suzhou(215006), China

{kongfang, gdzhou}@suda.edu.cn

Abstract. This paper systematically explores the effectiveness of dependency and constituent-based syntactic information for anaphoricity determination. In particular, this paper proposes two ways to combine dependency and constituent-based syntactic information to explore their complementary advantage. One is a dependency-driven constituent-based structured representation, and the other uses a composite kernel. Evaluation on the Automatic Content Extraction (ACE) 2003 corpus shows that dependency and constituent-based syntactic information are quite complementary and proper combination can much improve the performance of anaphoricity determination, and further improve the performance of coreference resolution.

Keywords: Dependency Relation, Constituent Parser, Anaphoricity Determination, Coreference Resolution, Tree Kernel, Composite Kernel

1 Introduction

Coreference resolution plays a critical role in many NLP applications. It identifies which noun phrases (NPs) refer to the same entity in the real world and can be divided into two subtasks: anaphoricity determination and antecedent identification. The first subtask, anaphoricity determination, determines whether a given noun phrases (NP) is anaphoric or not. Here we say an NP is anaphoric if it has any antecedent in the context preceding it, and non-anaphoric otherwise. The second subtask, antecedent identification, identifies the antecedent of a given anaphoric NP. 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 thus does not need to be resolved.

Recently, anaphoricity determination has been the subject of increased attention in coreference resolution. A variety of techniques have been proposed to address anaphoricity determination as an independent task (e.g. Paice and Husk 1987; Lappin and Leass 1994; Kennedy and Boguraev 1996; Denber 1998; Bean and Riloff 1999; Vieira and Poesio 2000; Evans 2001; Cherry, Cherry and Bergsma 2005; Bergsma et al. 2008). Since then, more studies have been done to incorporate anaphoricity determination into coreference resolution in a pipeline way (e.g. Ng and Cardie 2002a; Yang et al. 2005; Kong et al. 2010) or in a joint way (e.g. Denis and Balbridge 2007; Luo 2007; Finkel and Manning 2008; Zhou and Kong 2009; Ng 2009), and achieved promising results.

While it is well known that NP anaphoricity interacts with various kinds of structured factors, most of previous studies only consider constituent-based syntactic information and there are few studies on exploring dependency-based syntactic information for anaphoricity determination. In this paper, we first study the effectiveness of dependency-based syntactic information for anaphoricity determination. Then, we propose different ways to combine dependency and constituent-based syntactic information for anaphoricity determination in better exploring their complementary nature. Our study shows that dependency and constituent-based syntactic information are quite complementary and proper

combination can much improve the performance of anaphoricity determination and that of coreference resolution as a whole even when simply integrating the anaphoricity determination module into a coreference resolution system as a filter.

The rest of this paper is organized as follows. Section 2 describes some background knowledge about this task. Section 3 briefly overviews related work. Section 4 describes dependency and constituent-based syntactic information and different ways of combining them for anaphoricity determination. Section 5 reports the experimental results. Finally, we conclude our work in Section 6.

2 Background Knowledge

For better understanding, we first describe some background knowledge in this section.

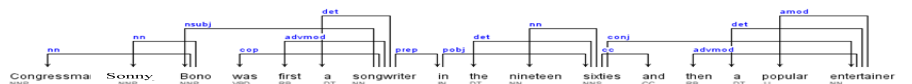
2.1 Anaphoricity determination

As a subtask in coreference resolution (Webber 1979), anaphoricity determination is responsible for determining whether a given NP is anaphoric or not. In principle, anaphoricity determination can work through either filtering out non-anaphoric NPs or keeping anaphoric NPs. From the viewpoint of the former, the simplest way is to include certain patterned syntactic constructions where non-anaphoric NPs often occur. From the viewpoint of the latter, the simplest way is to include the antecedent along with necessary syntactic constructions in the parse tree structure as much as possible.

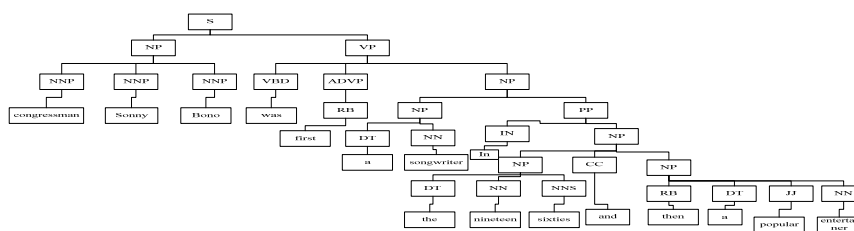
2.2 Dependency parse tree and constituent parse tree

Both the dependency and constituent parse tree structures are widely used to represent the structure of a sentence. Figure 1 shows an example. In the dependency parse tree structure, the dependency relationship between two words is represented by an arrow along with its name. In accordance with the convention adopted in the literature, the word pointed by the arrow is the dependent word, and the word outpointed by the arrow is the head word.

(2-1) Congressman Sonny Bono was first a songwriter in the nineteen sixties and then a popular entertainer.



(a) Dependency parse tree



(a) Constituent parse tree

Figure 1: Example of dependency and constituent parse trees with regard to sentence “Congressman Sonny Bono was first a songwriter in the nineteen sixties and then a popular entertainer.”

However, there exist some important differences between the dependency and constituent parse tree structures. On the one hand, the constituent parse tree structure is normally derived from the phrase structure grammar, which employs linguistic labels (such as part-of-speech and phrase tags) and has dominated the research of syntactic analysis in the literature. On the other hand, the dependency parse tree structure is derived from the dependency grammar, which does not employ linguistic labels and has drawn more and more attention in recent years due to its conciseness and clearness. While the constituent parse tree structure represents the syntactic structure of a sentence in a way of nested multi-word phrase

constituents, the dependency one directly represents the dependency relationship between individual words. That is to say, the dependency parse tree structure mainly concerns the dependency relationship between individual words, instead of the phrase structure contained in the constituent parse tree structure. Therefore, these two kinds of syntactic parse tree structures may behave quite differently in capturing different aspects of syntactic phenomena.

Normally, the constituent parse tree structure is more effective at filtering out non-anaphoric NPs and keeping the anaphoric NPs whose anaphoricity is only related with a local antecedent. For example, as illustrated by the Government and Binding theory, a reflexive (e.g. “himself” in 2-2) often works as a short-distance anaphor and is locally bound by its antecedent (e.g. “John” in 2-2). Such locality can normally be effectively captured by the constituent parse tree structure.

(2-2) John likes pictures of himself.

In comparison, the dependency parse tree structure is more effective at determining the anaphoricity of those NPs which is not bound locally. (2-1) gives an example where it is easy for the dependency parse tree structure to determine the anaphoricity of the NP “a popular entertainer”.

Obviously, dependency and constituent parse tree structures have their own advantages in either filtering out non-anaphoric NPs or keeping anaphoric NPs. This paper focuses on improving the performance of anaphoricity determination and coreference resolution by combining useful structured information in both the dependency and constituent parse tree structures.

3 Related Work

This section briefly overviews related work on anaphoricity determination.

Traditionally, anaphoricity determination has been tackled independently of coreference resolution using a variety of techniques. Representative studies include: Kennedy and Boguraev (1996), Bean and Riloff (1999), Evans (2001), Cherry and Bergsma (2005), and Bergsma et al. (2008).

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*. 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. Evans (2001) automatically identified the anaphoricity of pronoun “*it*” using various kinds of lexical and syntactic features. 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, then gathered the distribution of words that occurred within the context from a large corpus, and finally identified the pronoun either anaphoric or non-anaphoric based on the word distribution.

Recently, anaphoricity determination has been incorporated into coreference resolution in a pipeline way. Representative studies include: Ng and Cardie (2002a), Ng (2004), Yang et al. (2005), and Kong et al. (2010). Ng and Cardie (2002a) employed various domain-independent features in identifying anaphoric NPs. Experiments showed that their method improved the performance of coreference resolution by 2.0 and 2.6 in F1-measure to 65.8 and 64.2 on the MUC-6 and MUC-7 corpora, respectively. Ng (2004) further examined the representation and optimization issues in using anaphoricity information to improve the performance of coreference resolution. 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 of coreference resolution by 2.9 and 1.6 to 67.3 and 67.2 in F1-measure on the MUC-6 and MUC-7 corpora, respectively. Kong et al. (2010) proposed a dependency-driven scheme to dynamically determine the constituent parse tree structure for tree kernel-based anaphoricity determination. This was done by exploiting constituent dependencies from both the syntactic and semantic perspectives to distinguish necessary evidence from noisy information in the syntactic parse tree. Experiments showed that this scheme outperformed all previous syntactic parse tree structures on anaphoricity determination, and improved the performance of coreference resolution.

Alternatively, anaphoricity determination and antecedent identification have been modeled as a joint task more recently. Representative studies include: Denis and Balbridge (2007), Luo (2007), Finkel and Manning (2008), Zhou and Kong (2009), and Ng (2009). Denis and Balbridge (2007) employed an integer linear programming (ILP) formulation for joint anaphoricity determination and antecedent identification by informing each other the final assignments. Experiments on the ACE 2003 corpus showed that this joint 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 mentions are already known from the annotated corpus). Therefore, the actual effect of this joint 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 is not coreferential with any candidate antecedent. This twin-model combined the probabilities returned by the creation component (an anaphoricity model) with those returned by 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 the performance of coreference resolution. Experiments on the ACE 2003 corpus demonstrated the effectiveness of such a global learning method with performance improvement of 2.4, 3.1 and 4.1 in F1-measure on the NWIRE, NPAPER and BNEWS domains, respectively. Ng (2009) presented a graphic approach to anaphoricity determination based on graph minimum cuts and demonstrated its effectiveness in overall coreference resolution. Experiments on the ACE 2003 corpus showed that his method improved the overall F1-measure by 2.8, 2.2 and 4.5 to 54.5, 64.0 and 60.8 on the NWIRE, NPAPER and BNEWS domains, respectively.

This paper proposes a new approach to determining NP anaphoricity by combining dependency and constituent-based syntactic information, and studies the application of an explicit anaphoricity determination module in coreference resolution.

4 Combining Dependency and Constituent-based Syntactic Information

In this section, we first introduce the representation of dependency-based syntactic information and the presentation of constituent-based syntactic information individually in tree kernel-based anaphoricity determination. Then we propose two ways to combine these two kinds of structured syntactic information.

4.1 Representation of Dependency-based Syntactic Information

Given a NP in consideration, one key issue is to choose a proper dependency tree span to well cover necessary dependency-based syntactic information for anaphoricity determination in the tree kernel computation. The basic idea is to keep various dependencies related with the focus NP in the same sentence. For this purpose, a dependency relations tree (DRT) is constructed as follows: (Figure 2 illustrates an example dependency relations tree for sentence (2-1).)

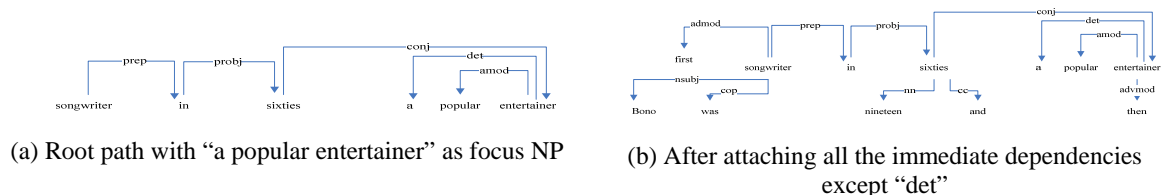


Figure 2: DRT: Representation of dependency-based syntactic information

From the dependency parse tree, as shown in Figure 1(a), we first extract the path between the focus NP and the topmost node along the reverse direction of the arrows from the focus NP. We call this path as root path. Figure 2(a) shows an example root path with "a popular entertainer" as focus NP.

Attaching all the nodes which have the immediate dependency relationship with a node (except the "det" relation) in the root path. Figure 2(b) shows the result after such dependency attachment.

Attaching the dependency relation type to the dependent word. In every dependent word node, we also record its corresponding POS.

4.2 Representation of Constituent-based Syntactic Information

It is well proven that constituent-based syntactic information plays an important role for anaphoricity determination. In this paper, we adopt the state-of-the-art Dynamic Expansion Tree (DET) extraction scheme, as proposed in Zhou and Kong (2009). DET takes predicate- and antecedent candidates-related information and right siblings into consideration (For details, please refer to Zhou and Kong, 2009). Generally, the more a parse tree structure includes, the more structured syntactic information would be provided, at the expense of more noisy information. In this paper, we limit the window size to 5 chunks (either NPs or non-NPs), including previous two chunks, current chunk and following two chunks.

Figure 3 illustrates a dynamic expansion tree example for the example sentence (2-1). Obviously, the antecedent “a songwriter” of the focus NP “a popular entertainer” is not bound locally and fails to be covered in DET.

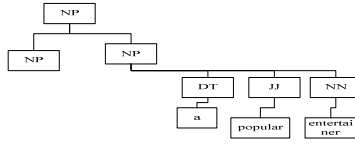


Figure 3: DET: Representation of constituent-based syntactic information

4.3 Combining Dependency and Constituent-based Syntactic Information

Just as described above, both dependency and constituent-based syntactic information have their own advantages and are helpful for anaphoricity determination. In this subsection, we propose two ways to combine dependency and constituent-based syntactic information. One is to generate a better representation of constituent-based syntactic information with the guidance of dependency relations and the other is to combine them via a composite kernel.

Dependency-driven constituent parse tree

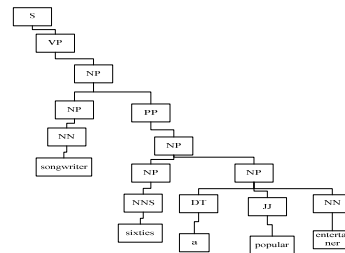
Given a full constituent parse tree, the root path from the focus NP to the root node is first kept. Then, all the nodes with the immediate dependency relationship with the focus NP, and their corresponding paths are attached to the root path. The intuition behind is that immediate dependencies normally carry major contextual information of the focus NP. Figure 4 illustrates the dependency-driven constituent parse tree (DCPT) of the example sentence (2-1) with the focus NP “a popular entertainer”.

```

nn(Bono-3, Congressman-1)
nn(Bono-3, Sormy-2)
nsubj(songwriter-7, Bono-3)
cop(songwriter-7, was-4)
advmod(songwriter-7, first-5)
det(songwriter-7, a-6)
det(sixties-11, the-9)
nn(sixties-11, nineteen-10)
prep_in(songwriter-7, sixties-11)
advmod(entertainer-16, then-13)
det(entertainer-16, a-14)
amod(entertainer-16, popular-15)
prep_in(songwriter-7, entertainer-16)
conj_and(sixties-11, entertainer-16)

```

(a) Dependencies achieved by Stanford Parser



(b) DCPT: Dependency-driven constituent parse tree

Figure 4: Example of DCPT

Composite kernel

For simplicity, this paper only adopts a simple linear kernel to equally combine a dependency-based tree kernel K_{dep} and constituent-based tree kernel K_{con} with normalization

$$K_{comp}(x_1, x_2) = \frac{K_{dep}(x_1, x_2)}{|K_{dep}(x_1, x_2)|} + \frac{K_{con}(x_1, x_2)}{|K_{con}(x_1, x_2)|}$$

In particular, we use the composite kernel to combine the dependency-based representation DRT and the constituent-based representation DET. We also consider the combination of DRT with the dependency-driven constituent-based representation DCPT to further capture the complementary nature of dependency and constituent-based syntactic information.

5 Experimentation

This section systematically evaluates our proposal on anaphoricity determination and its application in coreference resolution using the ACE 2003 corpus.

5.1 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 respectively.

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 Hidden Markov Model (HMM) based engine with error-driven learning capability (Zhou and Su, 2000 & 2002). In addition, all the texts are parsed using the Charniak parser (Charniak 2001) to get the constituent parse tree structure while the dependency parse tree structure is achieved by the Stanford Parser (Marie et al. 2006). Finally, we use the SVM-light toolkit (Joakim, 1998)¹ with the convolution tree kernel function SVM^{light}-TK (Moschitti, 2004)² as the classifier. For comparison, the training parameters C (SVM) and λ (tree kernel) are simply set to 2.4 and 0.4 respectively, as adopted 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 overall 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).

5.2 Experimental Results

Performance of anaphoricity determination

Table 1 compares the performance of anaphoricity determination using different settings. It shows that:

Table 1: Comparison of different settings on anaphoricity determination

Corpus	DRT		DET		DRT+DET		DCPT		DRT+DCPT	
	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻	Acc ⁺	Acc ⁻
NWIRE	69.3	71.6	72.1	79.8	78.7	78.9	83.3	77.1	85.4	79.7
BNEWS	73.7	62.4	81.9	66.4	82.7	72.0	86.8	70.3	84.7	79.5
NPAPER	75.8	60.8	80.8	61.1	84.7	66.8	90.0	64.2	91.1	68.8

1) The constituent-based representation DET noticeably outperforms the dependency-based representation DRT on both keeping anaphoric NPs and filtering out non-anaphoric NPs. This indicates the advantage of the constituent-based representation over the dependency-based one on anaphoricity determination.

¹ <http://svmlight.joachims.org/>

² <http://ai-nlp.info.uniroma2.it/moschitti/>

2) The combination of DRT and DET via a simple composite kernel significantly improves the performance in both keeping anaphoric NPs and filtering out non-anaphoric NPs by about 4-8% in accuracy. This justifies the complementary nature of dependency and constituent-based syntactic information.

3) Although dependency-driven constituent-based representation DCPT performs worse than the composite kernel combination of DRT and DET by about 2% in filtering out non-anaphoric NPs, DCPT significantly outperforms the composite kernel combination by about 4-6% in keeping anaphoric NPs. Actually, this characteristic is preferred since a coreference resolution system can often detect some non-anaphoric NPs when failing to find their antecedent candidates while it is almost impossible to recover those filtered anaphoric NPs caused by anaphoricity determination. Therefore, keeping more anaphoric NPs is always preferred over filtering out more non-anaphoric NPs in anaphoricity determination from the viewpoint of overall coreference resolution.

4) The composite kernel combination of DCPT with DRT outperforms the composite kernel combination of DET with DRT on both keeping anaphoric NPs and filtering out non-anaphoric NPs. This further indicates the effectiveness of dependency-based syntactic information.

Performance of coreference resolution

We have evaluated the effectiveness of our anaphoricity determination module with different settings on coreference resolution by including it as a preprocessing step to a state-of-the-art baseline coreference resolution system without explicit anaphoricity determination. This is done by filtering out those non-anaphoric NPs according to the anaphoricity determination module. 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. 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 2 presents the detailed performance of the coreference resolution system with different anaphoricity determination settings. It shows that:

Table 2: Performance of anaphoricity determination on coreference resolution

System	NWIRE			NPAPER			BNEWS		
	R%	P%	F	R%	P%	F	R%	P%	F
Without anaphoricity determination (Baseline)	55.0	63.8	59.1	62.1	65.0	63.5	53.2	60.5	56.6
With DRT -based anaphoricity determination	53.5	68.6	60.1	60.0	69.5	64.4	50.8	66.7	57.7
With DET-based anaphoricity determination	53	73.8	61.7	58.1	74.5	65.3	50	70.8	58.6
With anaphoricity determination using the combination of DRT and DET	52.8	75.6	62.2	57.8	77.5	66.2	49.8	78.4	60.9
With DCPT -based anaphoricity determination	52.4	79.6	63.2	58.1	80.3	67.4	50.1	79.8	61.6
With anaphoricity determination using the combination of DRT and DCPT	52.7	82.1	64.2	57.6	83.3	68.1	49.5	83.9	62.3
With golden anaphoricity determination	54.6	81.7	65.5	60.4	82.1	69.6	51.9	82.1	63.6

There is a large performance gap of 6.4, 6.1 and 7.0 in F1-measure on the NWIRE, NPAPER and BNEWS domains, respectively, between the coreference resolution system with golden anaphoricity determination and the baseline system without anaphoricity determination. This suggests the necessity of proper anaphoricity determination.

Compared to the baseline system without anaphoricity determination, the DRT-based anaphoricity determination module improves the F1-measure by 1.0(***) , 0.9(***) and 1.1(***) to 60.1, 64.4 and 57.7 on the NWIRE, NPAPER and BNEWS domains, respectively, while the DET-based anaphoricity determination module improves the F1-measure by 2.6(***) , 1.8(***) and 2.0(***) to 61.7, 65.3 and 58.6 on the NWIRE, NPAPER and BNEWS domains, respectively. This indicates the effectiveness of automatic anaphoricity determination using both dependency and constituent-based syntactic information, respectively, and the superiority of the constituent-based one over the dependency-based one.

Compared to the baseline system, the anaphoricity determination module with the combination of DRT and DET via a composite kernel improves the F1-measure by 3.1(***) , 2.7(***) and 4.3(***) to 62.2, 66.2 and 60.9 on the NWIRE, NPAPER and BNEWS domains, respectively. This proves the complementary nature of a dependency-based anaphoricity determination module and a constituent-based anaphoricity determination module on overall coreference resolution.

The coreference resolution system with the DCPT-based anaphoricity determination module consistently performs better than the one with anaphoricity determination using the composite kernel combination of DRT and DET by 1.0(***) , 1.2(***) and 0.7(***) in F1-measure on the NWIRE, NPAPER and BNEWS domains, respectively. This justifies our claim on the preference of keeping more anaphoric NPs over filtering out more non-anaphoric NPs.

Compared to the coreference resolution system with DCPT-based anaphoricity determination, the further combination of DCPT with DRT via the simple composite kernel consistently improve the F1-measure by 1.0(**) , 0.7(**) and 0.7(**) , on the NWIRE, NPAPER and BNEWS domains, respectively. Although there are still some performance gaps with 1.3, 1.5 and 1.3 in F1-measure compared to the coreference resolution system with golden anaphoricity determination, our gains over the baseline system without anaphoricity determination are much more obvious with 5.1, 4.6 and 5.7 in F1-measure, on the NWIRE, NPAPER and BNEWS domains, respectively.

Table 3: Comparison of different anaphoricity determination on overall coreference resolution

System		NWIRE			NPAPER			BNEWS		
		R%	P%	F	R%	P%	F	R%	P%	F
Zhou and Kong (2009)	without anaphoricity determination	53.1	67.4	59.4	57.7	67.0	62.1	48.0	65.9	55.5
	with DET -based anaphoricity determination in a pipeline way	51.6	77.2	61.8	55.2	78.6	65.2	47.5	80.3	59.6
	performance gain			2.4			3.1			4.1
Ng (2009)	without anaphoricity determination	59.1	58.	58.6	60.8	62.6	61.7	57.7	52.6	55.0
	with graph minimum cut-based anaphoricity determination in a joint way	54.1	69.0	60.6	57.9	71.2	63.9	53.1	67.5	59.4
	performance gain			2.0			2.2			4.4
Ours	without anaphoricity determination	55.0	63.8	59.1	62.1	65.0	63.5	53.2	60.5	56.6
	with DRT+DCPT-based anaphoricity determination in a pipeline way	52.7	82.1	64.2	57.6	83.3	68.1	49.5	83.9	62.3
	performance gain			5.1			4.6			5.7

Comparison with other anaphoricity determination approaches

Table 3 compares the contribution of different anaphoricity determination approaches on overall conference resolution. Here, Zhou and Kong (2009) use the same set of features with ours in the baseline

system and apply a DET-based anaphoricity determination module in a pipeline way. Ng (2009) builds his baseline system using 33 features as described in Ng (2007) and integrates anaphoricity determination into coreference resolution in a joint way using a graph minimum cut algorithm. From Table 3, we can find 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).

Using DET-based anaphoricity determination in a pipeline way, Zhou and Kong (2009) improve the F1-measure of overall coreference resolution by 2.4, 3.1 and 4.1 on the three domains of NWIRE, NPAPER and BNEWS, respectively. Integrating anaphoricity determination into coreference resolution in a joint way using a graph minimum cut algorithm, Ng (2009) improves the F1-measure of overall coreference resolution by 2.0, 2.2 and 4.4 on the three domains, respectively. In comparison, our approach is much more effective with the performance gain of 5.1, 4.6 and 5.7 on the three domains, respectively.

6 Conclusion and Further Work

This paper systematically explores the effectiveness of dependency-based syntactic information for anaphoricity determination. Furthermore, we study two ways of integrating dependency-based syntactic information with constituent-based syntactic information for anaphoricity determination. Evaluation on the ACE 2003 corpus indicates the usefulness of both dependency-based and constituent-based syntactic information in anaphoricity determination. It also shows that dependency and constituent-based syntactic information complement each other in capturing different aspects of syntactic phenomena for anaphoricity determination and coreference resolution.

To our best knowledge, this is the first systematic research on exploring the effectiveness of dependency-based syntactic information and its combination with constituent-based syntactic information for anaphoricity determination.

For future work, we will explore more kinds of structured syntactic information in anaphoricity determination and their application in overall coreference resolution. In addition, we will study better ways of integrating anaphoricity determination and antecedent identification.

Acknowledgments

This research was supported by Projects 60873150, 90920004 and 61003153 under the National Natural Science Foundation of China, Project KJS1025 under the Open Projects Fund of Jiangsu Provincial Key Laboratory for Computer Information Processing Technology.

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*
- E. Charniak. 2001. Immediate-head Parsing for Language Models. *ACL' 2001*
- 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 Anaphoria 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, G.D. Zhou, L.H. Qian and Q.M. Zhu. 2010. Dependency-driven Anaphoricity Determination for Coreference Resolution. *COLING'2010*
- S. Lappin and J. L. Herbert. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4)
- X. Luo. 2007. Coreference or not: A twin model for coreference resolution. *NAACL-HLT'2007*
- M.C. Marneffe, B. MacCartney and C.D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *LREC 2006*.
- 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*
- V. Ng 2010. Supervised Noun Phrase Coreference Research: The First Fifteen Years. *ACL'2010*
- 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, 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. Tan 2008. A Twin-Candidate Model for Learning-Based Anaphora Resolution. *Computational Linguistics* 34(3):327-356
- 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*