Global Learning of Noun Phrase Anaphoricity in Coreference Resolution via Label Propagation

ZHOU GuoDong KONG Fang
JiangSu Provincial Key Lab for Computer Information Processing Technology
School of Computer Science and Technology
Soochow University. Suzhou, China 215006
Email:{gdzhou,kongfang}@suda.edu.cn

Abstract

Knowledge of noun phrase anaphoricity might be profitably exploited in coreference resolution to bypass the resolution of non-anaphoric noun phrases. However, it is surprising to notice that recent attempts to incorporate automatically acquired anaphoricity information into coreference resolution have been somewhat disappointing. This paper employs a global learning method in determining the anaphoricity of noun phrases via a label propagation algorithm to improve learning-based coreference resolution. In particular, two kinds of kernels, i.e. the feature-based RBF kernel and the convolution tree kernel, are employed to compute the anaphoricity similarity between two noun phrases. Experiments on the ACE 2003 corpus demonstrate the effectiveness of our method in anaphoricity determination of noun phrases and its application in learning-based coreference resolution.

1 Introduction

Coreference resolution, the task of determining which noun phrases (NPs) in a text refer to the same real-world entity, has long been considered an important and difficult problem in natural language processing. Identifying the linguistic constraints on when two NPs can co-refer remains an active area of research in the community. One significant constraint on coreference, the anaphoricity constraint, specifies that a non-anaphoric NP cannot be coreferent with any of its preceding NPs in a given text. Therefore, it is useful to skip over these non-anaphoric NPs rather than attempt an unnecessary search for an antecedent for them, only to end up with inaccurate outcomes. Although many existing machine learning approaches to coreference resolution have performed reasonably well without explicit anaphoricity determination (e.g., Soon et al 2001; Ng and Cardie 2002b; Strube and Muller 2003; Yang et al 2003, 2008), anaphoricity determination has been studied fairly extensively in the literature, given the potential usefulness of NP anaphoricity in coreference resolution. One common approach involves the design of heuristic rules to identify specific types of non-anaphoric NPs, such as pleonastic pronouns (e.g. Paice and Husk 1987; Lappin and Leass 1994; Kennedy and Boguraev 1996; Denber 1998) and existential definite descriptions (e.g., Vieira and Poesio 2000). More recently, the problem has been tackled using statistics-based (e.g., Bean and Riloff 1999; Bergsma et al 2008) and learning-based (e.g. Evans 2001; Ng and Cardie 2002a; Ng 2004; Yang et al 2005; Denis and Balbridge 2007) methods. Although there is empirical evidence (e.g. Ng and Cardie 2002a, 2004) that coreference resolution might be further improved with proper anaphoricity information, its contribution is still somewhat disappointing and lacks systematic evaluation.

This paper employs a label propagation (LP) algorithm for global learning of NP anaphoricity. Given the labeled data and the unlabeled data, the LP algorithm first represents labeled and unlabeled instances as vertices in a connected graph, then propagates the label information from any vertex to nearby vertices through weighted edges and finally infers the labels of unlabeled instances until a global stable stage is achieved. Here, the labeled data in this paper include all the NPs in the training texts with the anaphoricity labeled and the unlabeled data include all the NPs in a test text with the anaphoricity unlabeled. One major advantage of LP-based anaphoricity determination is that the anaphoricity of all the NPs in a text can be determined together in a global way. Compared with previous methods, the LP algorithm can effectively capture the natural clustering structure in both the labeled and unlabeled data to smooth the labeling function. In particular, two kinds of

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 978–986,
Singapore, 6-7 August 2009.

kernels, i.e. the feature-based RBF kernel and the convolution tree kernel, are employed to compute the anaphoricity similarity between two NPs and weigh the edge between them. Experiments on the ACE 2003 corpus show that our LP-based anaphoricity determination significantly outperforms locally-optimized one, which adopts a classifier (e.g. SVM) to determine the anaphoricity of NPs in a text individually and significantly improves the performance of learning-based coreference resolution. It also shows that, while feature-based anaphoricity determination contributes much to pronoun resolution, its contribution on definite NP resolution can be ignored. In comparison, it shows that tree kernel-based anaphoricity resolution contributes significantly to the resolution of both pronouns and definite NPs due to the inclusion of various kinds of syntactic structured information.

The rest of this paper is organized as follows. In Section 2, we review related work in anaphoricity determination. Then, the LP algorithm is introduced in Section 3 while Section 4 describes different similarity measurements explored in the LP algorithm. Section 5 shows the experimental results. Finally, we conclude our work in Section 6.

2 Related Work

Given its potential usefulness in coreference resolution, anaphoricity determination has been studied fairly extensively in the literature and can be classified 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), statistics-based (e.g., Bean and Riloff 1999; Cherry and Bergsma 2005; Bergsma et al 2008) and learning-based (e.g. Evans 2001; Ng and Cardie 2002a; Ng 2004; Yang et al 2005; Denis and Balbridge 2007).

For the heuristic rule-based approaches, Paice and Husk (1987), Lappin and Leass (1994), Kennedy and Boguraev (1996), Denber (1998), and Cherry and Bergsma (2005) looked for particular constructions using certain trigger words to identify pleonastic pronouns while Vieira and Poesio (2000) recognized non-anaphoric definite NPs through the use of syntactic cues and case-sensitive rules and found that nearly 50% of definite NPs are non-anaphoric. As a representative, Lappin and Leass (1994), and 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.

For the statistics-based approaches, Bean and Riloff (1999) developed a statistics-based method for automatically identifying 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. They found that existential NPs account for 63% of all definite NPs and 76% of them could be identified by syntactic or lexical means. Using 1600 MUC-4 terrorism news documents as the training data, they achieved 87% in precision and 78% in recall at identifying non-anaphoric definite NPs. Cherry and Bergsma (2005) extended the work of Lappin and Leass (1994) for large-scale anaphoricity determination by additionally detecting non-anaphoric instances of *it* using Minipar's pleonastic category *Subj*. This is done by both employing Minipar's named entity recognition to identify time expressions, such as "it was midnight…", and providing a number of other linguistic patterns to match common non-anaphoric *it* cases, such as in expressions "darn it" and don't overdo it". Bergsma et al (2008) proposed a distributional method in detecting non-anaphoric pronouns by first extracting the surrounding textual context of the pronoun, then gathering the distribution of words that occurred within that context from a large corpus and finally learning to classify these distributions as representing either anaphoric and non-anaphoric pronoun instances. Experiments on the Science News corpus of It-Bank [1] in identifying non-anaphoric pronoun *it* show that their distributional method achieved the performance of 81.4%, 71.0% and 75.8 in precision, recall and F1-measure, respectively, compared with the performance of 93.4%, 21.0% and 34.3 in precision, recall and F1-measure, respectively using the rule-based approach as described in Lappin and Leass (1994), and the performance of 66.4%, 49.7% and 56.9 in precision, recall and F1-measure, respectively using the rule-based approach as described in Cherry and Bergsma (2005).

Among the learning-based methods, Evans (2001) applied a machine learning approach on identifying the non-anaphoricity of pronoun *it*. Ng and Cardie (2002a) employed various domain-independent features in identifying anaphoric NPs and showed how such information

[1] www.cs.ualberta.ca/~bergsma/ItBank/

can be incorporated into a coreference resolution system. Experiments show that their method improves 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, due to much more gain in precision compared with the loss in recall. Ng (2004) examined the representation and optimization issues in computing and using anaphoricity information to improve learning-based coreference resolution systems. He used an anaphoricity classifier as a filter for coreference resolution. Evaluation on the ACE 2003 corpus shows that, compared with a baseline coreference resolution system of no explicit anaphoricity determination, their method improves the performance by 2.8, 2.2 and 4.5 to 54.5, 64.0 and 60.8 in F1-measure (due to the gain in precision) on the NWIRE, NPAPER and BNEWS domains, respectively, via careful determination of an anaphoricity threshold with proper constraint-based representation and global optimization. However, he did not look into the contribution of anaphoricity determination on coreference resolution of different NP types, such as pronoun and definite NPs. 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 thus equipped it with the non-anaphoricity determination capability. Experiments show that the proposed method improves 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, due to much more gain in precision compared with the loss in recall. However, surprisingly, their experiments also 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 models anaphoricity and coreference as a joint task, such that each local model informs the other for final assignments. Experiments on the NWIRE, NPAPER and BNEWS domains of the ACE 2003 corpus shows that this joint anaphoricity-coreference ILP formulation improves the F1-measure by 0.7-1.0 over the coreference-only ILP formulation. 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.

3 Label Propagation

In the LP algorithm (Zhu and Ghahramani 2002), the natural clustering structure in data is represented as a connected graph. Given the labeled data and unlabeled data, the LP algorithm first represents labeled and unlabeled instances as vertices in a connected graph, then propagates the label information from any vertex to nearby vertices through weighted edges and finally infers the labels of unlabeled instances until a global stable stage is achieved. Figure 1 presents the label propagation algorithm.

Assume:

Y : the $n*r$ labeling matrix, where y_{ij} represents the probability of vertex $x_i (i = 1 \ldots n)$ with label $r_j (j = 1 \ldots r)$;

Y_L : the top l rows of Y^0. Y_L corresponds to the l labeled instances;

Y_U : the bottom u rows of Y^0. Y_U corresponds to the u unlabeled instances;

$\overline{T}$: a $n*n$ matrix, with $\overline{t}_{ij}$ is the probability jumping from vertex x_i to vertex x_j;

BEGIN (the algorithm)

Initialization:

1) Set the iteration index $t = 0$;
2) Let Y^0 be the initial soft labels attached to each vertex;
3) Let Y_L^0 be consistent with the labeling in the labeled data, where y_{ij}^0 = the weight of the labeled instance if x_i has the label r_j;
4) Initialize Y_U^0;

REPEAT

Propagate the labels of any vertex to nearby vertices by $Y^{t+1} = \overline{T} Y^t$;

Clamp the labeled data, that is, replace Y_L^{t+1} with Y_L^0;

UNTIL Y converges(e.g. Y_L^{t+1} converges to Y_L^0);

Assign each unlabeled instance with a label: for $x_i (l \prec i \le n)$, find its label with $\arg\max_j y_{ij}$;

END (the algorithm)

Figure 1: The LP algorithm

Here, each vertex corresponds to an instance, and the edge between any two instances x_i and x_j is weighted by w_{ij} to measure their similarity. In principle, larger edge weights allow labels to travel through easier. Thus the closer the instances are, the more likely they have similar

labels. The algorithm first calculates the weight w_{ij} using a kernel, then transforms it to $t_{ij} = p(j \to i) = w_{ij} / \sum_{k=1}^{n} w_{kj}$, which measures the probability of propagating a label from instance x_j to instance x_i, and finally normalizes t_{ij} row by row using $\bar{t}_{ij} = t_{ij} / \sum_{k=1}^{n} t_{ik}$ to maintain the class probability interpretation of the labeling matrix Y.

During the label propagation process, the label distribution of the labeled data is clamped in each loop using their initial weights and acts like forces to push out labels through the unlabeled data. With this push originating from the labeled data, the label boundaries will be pushed faster along edges with larger weights and settle in gaps along those with lower weights. Ideally, we can expect that w_{ij} across different classes should be as small as possible and w_{ij} within the same class as big as possible. In this way, label propagation tends to happen within the same class. This algorithm has been shown to converge to a unique solution (Zhu and Ghahramani 2002), which can be obtained without iteration in theory, and the initialization of Y_U^0 (the unlabeled data) is not important since Y_U^0 does not affect its estimation. However, proper initialization of Y_U^0 actually helps the algorithm converge more rapidly in practice. In this paper, each row in Y_U^0, i.e. the label distribution for each test instance, is initialized to the weighted similarity of the test instance with the labeled instances.

4 Kernel-based Similarity

The key issue in label propagation is how to compute the similarity w_{ij} between two instances x_i and x_j. This paper examines two similarity measures: the feature-based RBF kernel and the convolution tree kernel.

Feature Type	Feature	Description
Features related with current NP itself	IsPronoun	1 if current NP is a pronoun, else 0
	IsDefiniteNP	1 if current NP is a define NP, else 0
	IsDemonstrativeNP	1 if current NP is a demonstrative NP, else 0
	IsArg0	1 if the semantic role of current NP is Arg0/agent, else 0
	IsArg0MainVerb	1 if current NP has the semantic role of Arg0/agent for the main predicate of the sentence, else 0
	IsArgs	0 if current NP has no semantic role, else 1
	IsSingularNP	1 if current NP is a singular noun, else 0
	IsMaleFemalePronoun	1 if current NP is a male/female personal pronoun, else 0
Features related with the local context surrounding current NP	StringMatch	1 if there is a full string match between current NP and one of other phrases in the context, else 0
	NameAlias	1 if current NP and one of other phrases in the context is a name alias or abbreviation of the other, else 0
	Appositive	1 if current NP and one of other phrases in the context are in an appositive structure, else 0
	NPNested	1 if current NP is nested in another NP, else 0
	NPNesting	1 if current NP nests another NP, else 0
	WordSenseAgreement	1 if current NP and one of other phrases in the context agree in the WordNet sense, else 0
	IsFirstNPinSentence	1 if current NP is the first NP of this sentence, else 0
	BackwardDistance	The distance between current NP and the nearest backward clause, indicated by coordinating words (e.g. that,which).
	ForwardDistance	The distance between the nearest forward clause, indicated by coordinating words (e.g. that, which), and current NP.

Table 1: Features in anaphoricity determination of NPs. Note: the semantic role-related features are derived from an in-house state-of-the-art semantic role labeling system.

4.1 Feature-based Kernel

In our feature-based RBF kernel to anaphoricity determination, an instance is represented by 17 lexical, syntactic and semantic features, as shown in Table 1, which are specifically designed for distinguishing anaphoric and non-anaphoric NPs, according to common-sense knowledge and linguistic intuitions. Since the local context surrounding an NP plays a critical role in discriminating whether an NP is anaphoric or not, the features in Table 1 can be classified into two categories: (a) current NP (i.e. the NP in anaphoricity consideration) itself, e.g.

types and semantic roles of current NP; (b) contextual information, e.g. whether current NP is nested in another NP, the distance between current NP and a clause structure, indicated by coordinating words (e.g. that, this, which).

4.2 Tree Kernel

Given a NP in anaphoricity determination, a parse tree represents the local context surrounding current NP in a structural way and thus contains much information in determining whether current NP is anaphoric or not. For example, the commonly used knowledge for anaphoricity determination, such as the grammatical role of current NP or whether current NP is nested in other NPs, can be directly captured by a parse tree structure.

Given a parse tree and a NP in consideration, the problem is how to choose a proper parse tree structure to cover syntactic structured information well in the tree kernel computation. Generally, the more a parse tree structure includes, the more syntactic structured information would be provided, at the expense of more noisy/unnecessary information. In this paper, we limit the window size to 5 chunks (either NPs or non-NPs), including previous two chunks, current chunk (i.e. current NP) and following two chunks, and prune out the substructures outside the window. Figure 2 shows the full parse tree for the sentence "Mary said the woman in the room hit her too", using the Charniak parser (Charniak 2001), and the chunk sequence derived from the parse tree using the Perl script[2] written by Sabine Buchholz from Tilburg University.

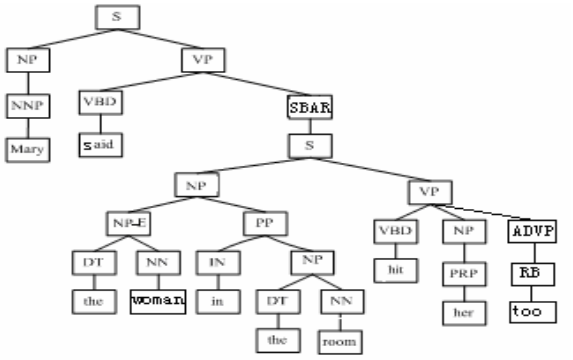
Here, we explore four parse tree structures in NP anaphoricity determination: the common tree (CT), the shortest path-enclosed tree (SPT), the minimum tree (MT) and the dynamically extended tree (DET), motivated by Yang et al (2006) and Zhou et al (2008). Following are the examples of the four parse tree structures, corresponding to the full parse tree and the chunk sequence, as shown in Figure 2, with the NP chunk "(NP (DT the) (NN woman))" in anaphoricity determination.

Common Tree (CT)

As shown in Figure 3(a), CT is the complete sub-tree rooted by the nearest common ancestor of the first chunk "(NP (NNP Mary))" and the last chunk "(NP (DT the) (NN room))" of the five-chunk window.

Shortest Path-enclosed Tree (SPT)

As shown in Figure 3(b), SPT is the smallest common sub-tree enclosed by the shortest path between the first chunk "(NP (NNP Mary))" and the last chunk "(NP (DT the) (NN room))" of the five-chunk window.

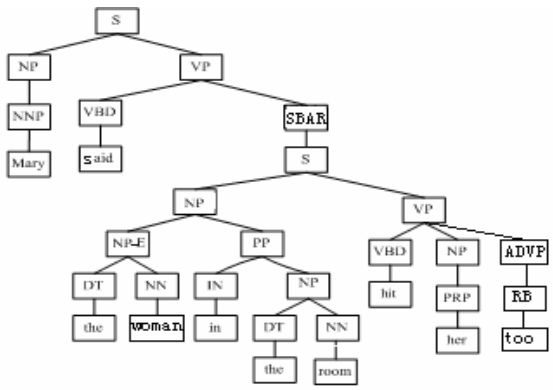


(a) the full parse tree

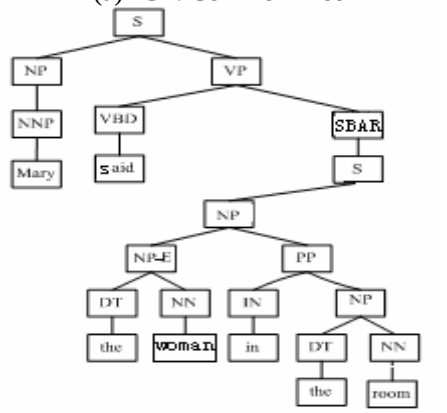
(NP (NNP Mary)) (VP (VBD said)) ***(NP-E (DT the) (NN woman))*** (PP (IN in)) (NP (DT the) (NN room)) (VP (VBD hit)) (NP (PRP her)) (ADVP (RB too))

(b) the chunk sequence

Figure 2: The full parse tree for the sentence "Mary said the woman in the room hit her too", using the Charniak parser, and the corresponding chunk sequence derived from it. Here, the label "E" indicates the NP in consideration.

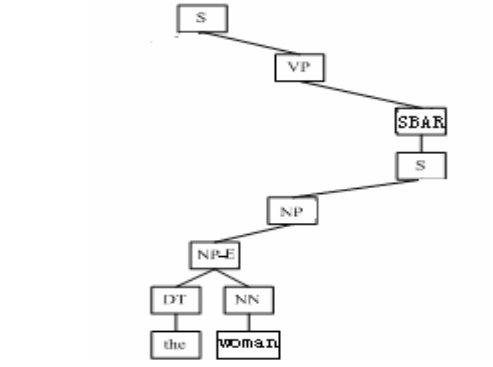


(a) CT: Common Tree

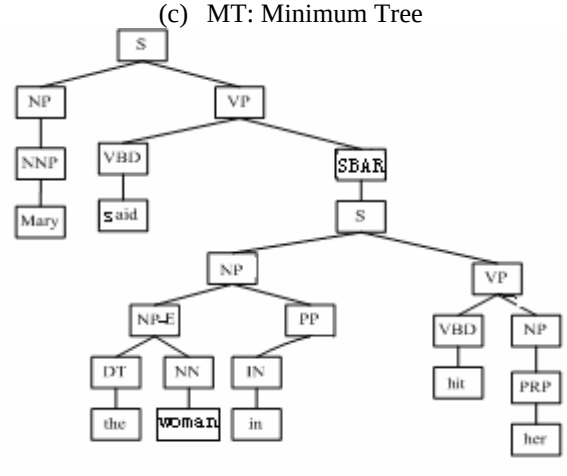


(b) SPT: Shortest Path-enclosed Tree

[2] http://ilk.kub.nl/~sabine/chunklink/



(c) MT: Minimum Tree



(d) DET: Dynamically Extended Tree

Figure 3: Examples of parse tree structures.

Minimum Tree (MT)

As shown in Figure 3(c), MT only keeps the root path from the NP in anaphoricity determination to the root node of SPT.

Dynamically Extended Tree (DET),

The intuitions behind DET are that the information related with antecedent candidates (all the antecedent candidates compatible[3] with current NP in anaphoricity consideration), predicates[4] and right siblings plays a critical role in coreference resolution. Given a MT, this is done by:

1) Attaching all the compatible antecedent candidates and their corresponding paths. As shown in Figure 3(d), "Mary" is attached while "the room" is not since the former is compatible with the NP "the woman" and the latter is not compatible with the NP "the woman". In this way, possible coreference between current NP and the compatible antecedent candidates can be included in the parse tree structure. In some sense, this is a natural extension of the twin-candidate learning method proposed in Yang et al (2003), which explicitly models the competition between two antecedent candidates.
2) For each node in MT, attaching the path from the node to the leaf node of the corresponding predicate, if it is predicate-headed, in the sense that such predicate-related information is useful in identifying certain kinds of expressions with non-anaphoric NPs, e.g. the non-anaphoric *it* in "darn it". As shown in Figure 3(d), "said" and "hit" are attached.
3) Attaching the path to the head word of the first right sibling if the parent of current NP is a NP and current NP has one or more right siblings. Normally, the NP in anaphoricity consideration, NP-E, in the production of "NP->NP-E+PP" introduces a new entity and thus non-anaphoric.
4) Pruning those nodes (except POS nodes) with the single in-arc and the single out-arc and with its syntactic phrase type same as its child node.

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

5 Experimentation

We have systematically evaluated the label propagation algorithm on global learning of NP anaphoricity determination on the ACE 2003 corpus, and its application in coreference resolution.

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.

As a baseline coreference resolution system, a raw test text is first preprocessed automatically by a pipeline of NLP components, including sentence boundary detection, POS tagging, named entity recognition and phrase chunking, and then a training or test instance is formed by a anaphor and one of its antecedent candidates, similar to Soon et al (2001). Among them, named entity recognition, part-of-speech tagging and noun phrase chunking apply the same Hidden Markov Model (HMM)-based engine with

[3] With matched number, person and gender agreements.

[4] For simplicity, only verbal predicates are considered in this paper. However, this can be extended to nominal predicates with automatic identification of nominal predicates.

error-driven learning capability (Zhou and Su, 2000 & 2002). During training, for each anaphor encountered, a positive instance is created by pairing the anaphor and its closest antecedent while a set of negative instances is formed by pairing the anaphor with each of the non-coreferential candidates. Based on the training instances, a binary classifier is generated using a particular learning algorithm. In this paper, we use SVMLight developed by Joachims (1998). During resolution, an anaphor is first paired in turn with each preceding antecedent candidate to form a test instance, which is presented to a classifier. The classifier then returns a confidence value indicating the likelihood that the candidate is the antecedent. Finally, the candidate with the highest confidence value is selected as the antecedent. As a baseline, the NPs with mismatched number, person and gender agreements are filtered out. On average, an anaphor has ~7 antecedent candidates. In particular, the test corpus is resolved in document-level, i.e. one document by one document.

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). For separate scoring of different NP types, a recognized reference is considered correct if the reconized antecedent is in the coreferential chain of the anaphor. To see whether an improvement is significant, we 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.

5.2 Experimental Results

Table 2 shows the performance of LP-based anaphoricity determination using the feature-based RBF kernel. It shows that our method achieves the accuracies of 74.8/84.4, 76.2/81.3 and 71.8/81.7 on identifying anaphoric/non-anaphoric NPs in the NWIRE, NPAPER and BNEWS domains, respectively. This suggests that our approach can effectively filter out about 82% of non-anaphoric NPs. However, it can only keep about 74% of anaphoric NPs. Table 2 also shows the performance on different NP types. Considering the effectiveness of anaphoricity determination on indefinite NPs (due to that most of anaphoric indefinite NPs are in an appositive structure and thus can be easily captured by the IsAppositive feature) and that most of errors in anaphoricity determination on proper nouns are caused by the named entity recognition module in the preprocessing), it indicates the difficulty of anaphoricity determination in filtering out non-anaphoric pronouns and identifying anaphoric definite NPs. As a comparison, Table 2 also shows the performance of locally-optimized anaphoricity determination using a classifier (SVM with the feature-based RBF kernel, as adopted in this paper) to determine the NPs in a text individually. It shows that the LP-based method systematically outperforms (>>>) the SVM-based method. This suggests the effectiveness of the LP algorithm in global modeling of the natural clustering structure in anaphoricity determination.

Table 3 shows the performance of LP-based anaphoricity determination using the convolution tree kernel on different parse tree structures. It shows that while MT performed worst due to its simple structure, DET outperforms MT(>>>), SPT(>>>) and CT(>>>) on all the three domains due to fine inclusion of necessary structural information, although inclusion of more information in both CT and SPT also improves the performance. It again verifies that LP-based anaphoricity determination outperforms (>>>) SVM-based one, using the tree kernel. Table 4 further indicates that all the three kinds of structural information related with antecedent candidates, predicates and right siblings in DET contribute significantly (>>>). In addition, Table 5 shows the detailed performance of LP-based anaphoricity determination on different anaphor types using DET. Compared with the feature-based RBF kernel as shown in Table 2, it shows that the convolution tree kernel significantly outperforms (>>>) the feature-based RBF kernel in all the three domains, with much contribution due to performance improvement on both pronouns and definite NPs, although the tree kernel performs moderately worse than the feature-based RBF kernel due to the effectiveness of anaphoricity determination on proper nouns and indefinite NPs using the IsNameAlias and IsAppositive features respectively.

Anaphor Type	NWIRE		NPAPER		BNEWS	
	Acc^+ (%)	Acc^- (%)	Acc^+ (%)	Acc^- (%)	Acc^+ (%)	Acc^- (%)
Pronoun	88.7	56.2	90.2	58.6	87.4	57.8
ProperNoun	72.5	85.2	74.6	80.5	70.6	78.8
DefiniteNP	66.6	83.1	72.1	77.5	65.3	81.5
InDefiniteNP	95.4	93.7	90.5	95.8	87.2	97.3
Overall	74.8	84.4	76.2	81.3	71.8	81.7
Overall(SVM)	*71.3*	*80.2*	*73.5*	*79.1*	*68.4*	*78.6*

Table 2: The performance of LP-based anaphoricity determination using the feature-based RBF kernel

Parse Tree structure Scheme		NWIRE (%)	NPAPER (%)	BNEWS (%)
CT	Acc^+	72.6	74.3	74.2
	Acc^-	82.1	80.2	72.3
SPT	Acc^+	72.4	74.1	73.8
	Acc^-	80.8	79.5	72.5
MT	Acc^+	71.4	70.5	66.9
	Acc^-	77.2	75.3	78.2
DET	Acc^+	79.2	81.2	76.5
	Acc^-	87.8	84.5	85.3
DET(SVM)	*Acc^+*	*76.5*	*78.9*	*74.3*
	Acc^-	*82.3*	*81.6*	*83.2*

Table 3: The performance of LP-based anaphoricity determination using the convolution tree kernel on different parse tree structures

Performance Change		NWIRE (%)	NPAPER (%)	BNEWS (%)
- antecedent candidates	Acc^+	-4.0	-3.8	-4.3
	Acc^-	-5.2	-5.3	-4.5
-predicate	Acc^+	-5.2	-4.8	-5.6
	Acc^-	-4.3	-3.5	-4.9
-first right sibling	Acc^+	-3.6	-4.1	-3.1
	Acc^-	-4.8	-5.2	-4.4

Table 4: The contribution of structural information in DET

Anaphor Type	NWIRE		NPAPER		BNEWS	
	Acc^+ (%)	Acc^- (%)	Acc^+ (%)	Acc^- (%)	Acc^+ (%)	Acc^- (%)
Pronoun	90.1	75.6	90.7	79.2	89.2	77.5
ProperNoun	71.4	83.5	72.8	78.1	68.3	77.2
DefiniteNP	74.6	89.1	77.3	85.5	75.3	88.7
InDefiniteNP	93.2	92.1	90.2	94.2	89.4	95.5
Overall	79.2	87.8	81.2	84.5	76.5	85.3

Table 5: The performance of LP-based anaphoricity determination using the tree kernel on DET

Finally, we evaluate the effect of LP-based anaphoricity determination on coreference resolution by including it as a preprocessing step to a baseline coreference resolution system without explicit anaphoricity determination, which employs the same set of features, as adopted in the single-candidate model of Yang et al (2003), using a SVM-based classifier and the feature-based RBF kernel. It shows that anaphoricity determination with the feature-based RBF Kernel much improves (>>>) the performance of coreference resolution with most of the contribution due to pronoun resolution while its contribution on definite NPs can be ignored. It indicates the usefulness of anaphoricity determination in filtering out non-anaphoric pronouns and the difficulty in identifying anaphoric definite NPs, using the feature-based RBF kernel. It also shows that tree kernel-based anaphoricity determination can not only improve (>>>) the performance on pronoun resolution but also improve (>>>) the performance on definite NP resolution due to the much better performance of tree kernel-based anaphoricity determination on definite NPs. This suggests the necessity of exploring structural information in identifying anaphoric definite NPs.

System		NWIRE			NPAPER			BNEWS		
		R%	P%	F	R%	P%	F	R%	P%	F
BaseLine (No Anaphoricity)	Pronoun	66.5	61.6	64.0	70.1	64.2	67.0	61.7	63.2	62.4
	DefiniteNP	26.9	80.3	40.2	34.5	62.4	44.4	30.5	71.4	42.9
	Overall	53.1	67.4	59.4	57.7	67.0	62.1	48.0	65.9	55.5
+Anaphoricity determination with the feature-based RBF kernel	Pronoun	64.1	67.9	66.0	67.3	72.4	69.8	59.5	75.7	66.6
	DefiniteNP	26.7	80.6	40.3	34.2	62.5	44.3	30.4	71.9	43.1
	Overall	50.6	75.4	60.7	54.4	77.1	63.8	45.9	76.9	57.4
+Anaphoricity determination with the convolution tree kernel	Pronoun	63.5	70.9	67.0	68	74.9	71.3	61.1	77.6	68.3
	DefiniteNP	28.5	82.4	42.1	36.2	65.3	46.1	32.3	73.1	44.2
	Overall	51.6	77.2	61.8	55.2	78.6	65.2	47.5	80.3	59.6

Table 6: Employment of anaphoricity determination in coreference resolution

6 Conclusion

This paper systematically studies a global learning method in identifying the anaphoricity of noun phrases via a label propagation algorithm and the application of an explicit anaphoricity determination module in improving learning-based coreference resolution. In particular, two kinds of kernels, i.e. the feature-based RBF kernel and the convolution tree kernel, are employed to compute the anaphoricity similarity

between two NPs. Evaluation on the ACE 2003 corpus indicates that LP-based anaphoricity determination using both the kernels much improves the performance of coreference resolution. It also shows the usefulness of various structural information, related with antecedent candidates, predicates and right siblings, in tree kernel-based anaphoricity determination and in coreference resolution of both pronouns and definite NPs.

To our knowledge, this is the first systematic exploration of both feature-based and tree kernel methods in anaphoricity determination and the application of an explicit anaphoricity determination module in learning coreference resolution.

Acknowledgement

This research is supported by Project 60873150 under the National Natural Science Foundation of China, project 2006AA01Z147 under the "863" National High-Tech Research and Development of China, project 200802850006 under the National Research Foundation for the Doctoral Program of Higher Education of China.

References

Bean D. and Riloff E. (1999). Corpus-based Identification of Non-Anaphoric Noun Phrases. *ACL'1999*:373-380.

Bergsma S., Lin D.K. and Goebel R. (2008). Distributional Identification of Non-Referential Pronouns. *ACL'2008*: 10-18.

Charniak E. (2001). Immediate-head Parsing for Language Models. *ACL'2001*: 129-137.

Cherry C. and Bergsma S. (2005). An expectation maximization approach to pronoun resolution. *CoNLL'2005*:88-95.

Collins M. and Duffy N. (2001). Convolution kernels for natural language. *NIPS'2001*: 625-632.

Denber M. (1998). Automatic Resolution of Anaphora in English. Technical Report, Eastman Kodak Co.

Denis P. and Baldridge J. (2007). Joint determination of anaphoricity and coreference using integer programming. *NAACL-HLT'2007*:236-243.

Evans R. (2001). Applying machine learning toward an automatic classification of *it*. *Literary and Linguistic Computing*, 16(1):45.57.

Joachims T. (1998). Text Categorization with Support Vector Machine: learning with many relevant features. *ECML-1998*: 137-142.

Kennedy C. and Boguraev B. (1996). Anaphora for everyone: pronominal anaphora resolution without a parser. *COLING'1996*: 113-118.

Lappin S. and Leass H.J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535.561.

Ng V. and Cardie C. (2002a). Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. *COLING'2002*:730-736.

Ng V. and Cardie C. (2002b). Improving machine learning approaches to coreference resolution. *ACL'2002*: 104-111

Ng V. (2004). Learning Noun Phrase Anaphoricity to Improve Conference Resolution: Issues in Representation and Optimization. *ACL'2004*: 151-158

Paice C.D. and Husk G.D. (1987). Towards the automatic recognition of anaphoric features in English text: the impersonal pronoun *it*. *Computer Speech and Language*, 2:109-132.

Soon W.M., Ng H.T. and Lim D. (2001). A machine learning approach to coreference resolution of noun phrase. *Computational Linguistics*, 2001, 27(4):521-544.

Strube M. and Muller C. (2003). A machine learning approach to pronoun resolution in spoken dialogue. *ACL'2003*: 168-175

Vieira R. and Poesio M. (2000). An empirically based system for processing definite descriptions. *Computational Linguistics*, 27(4): 539–592.

Vilain M., Burger J., Aberdeen J., Connolly D. and Hirschman L. (1995). A model theoretic coreference scoring scheme. *MUC-6*: 45–52.

Yang X.F., Zhou G.D., Su J. and Chew C.L. (2003). Coreference Resolution Using Competition Learning Approach. *ACL'2003*:177-184

Yang X.F., Su J. and Tan C.L. (2005). A Twin-Candidate Model of Coreference Resolution with Non-Anaphor Identification Capability. *IJCNLP'2005*:719-730.

Yang X.F., Su J. and Tan C.L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. *COLING-ACL'2006*: 41-48.

Yang X.F., Su J., Lang J., Tan C.L., Liu T. and Li S. (2008). An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming. *ACL'2008*: 843-851.

Zhou G.D. and Su. J. (2000). Error-driven HMM-based chunk tagger with context-dependent lexicon. *EMNLP-VLC'2000:* 71–79

Zhou G.D. and Su J. (2002). Named Entity recognition using a HMM-based chunk tagger. In *ACL'2002*:473–480.

Zhou G.D., Kong F. and Zhu Q.M. (2008). Context-sensitive convolution tree kernel for pronoun resolution . *IJCNLP'2008*:25-31.

Zhu X. and Ghahramani Z. (2002). Learning from Labeled and Unlabeled Data with Label Propagation. *CMU CALD Technical Report*.CMU-CALD-02-107.