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

Improve Tree Kernel-Based Event Pronoun Resolution with Competitive Information

Fang Kong   Guodong Zhou*
JiangSu Provincial Key Lab for Computer Information Processing Technology
School of Computer Science and Technology Soochow University
{kongfang, gdzhou}@suda.edu.cn

Abstract

Event anaphora resolution plays a critical role in discourse analysis. This paper proposes a tree kernel-based framework for event pronoun resolution. In particular, a new tree expansion scheme is introduced to automatically determine a proper parse tree structure for event pronoun resolution by considering various kinds of competitive information related with the anaphor and the antecedent candidate. Evaluation on the OntoNotes English corpus shows the appropriateness of the tree kernel-based framework and the effectiveness of competitive information for event pronoun resolution.

1 Introduction

As one of the most important techniques in discourse analysis, anaphora resolution aims to resolve a given mention to its referred expression in a text. It has been a focus of research in Natural Language Processing (NLP) for decades and achieved much success recently (e.g. Soon et al. 2001; Ng and Cardie 2002; Ng 2007, 2009; Yang et al.2004, 2006, 2008; Kong et al. 2009). In the literature, most previous work aimed at entity anaphora resolution, in which both the anaphor and its antecedent are mentions of the same real world entity.

In comparison, event anaphora resolution is the process of linking a given referring expression to an event/fact/proposition, which is representative of eventuality and an abstract object. For example:

(a) *I plan to **[rest]** a bit and spend time with my family, after **[that]**, to help my party gain force and contribute to the future of the country.*

(b) *Yes, it took a while last night to sort out precisely what the court had **[decided]** by such a narrow margin. **[This]** was a stabilizing decision that restored order to a very chaotic situation.*

Obviously, the example (a) includes an intra-sentence event anaphora (the anaphor "*that*" refers to the event "*rest a bit and spend time with my family*" in the current sentence) while the example (b) includes an inter-sentence event anaphora (the anaphor "*this*" refers to the event "*what the court had decided by such a narrow margin*" in the last sentence). Although event pronouns may actually refer to a paragraph or larger chunks of texts, in this paper we only consider the cases taking clauses as antecedents and take the main predicate[1] (e.g. "*rest*" and "*decided*" in the examples) of the clause as the representation of the corresponding event.

In this paper, we focus on event pronoun resolution, the most difficult type of event anaphora resolution due to the least discriminative information that an event pronoun can provide. Here, event pronouns refer to those pronouns whose antecedents are event predicates. In order to explore various kinds of structured syntactic information, a tree kernel-based framework is proposed for event pronoun resolution. In particular, various kinds of competitive information related with the anaphor and the antecedent candidate are incorporated into the tree kernel-based event pronoun resolution framework. Evaluation shows the appropriateness of the tree kernel-based framework and the effectiveness of such competitive information for event pronoun resolution.

The rest of this paper is organized as follows. Section 2 briefly describes the related work. Section 3 presents our tree kernel-based event pronoun resolution framework. Section 4 reports the experimental results and analysis. Finally, we conclude our work in Section 5.

2 Related Work

Event anaphora resolution has been drawing more and more attention recently. While some hand-crafted constraints have been designed to resolve event anaphora of normally limited kinds of predicates (e.g. Byron, 2002), most previous work adopted a learning-based framework (e.g. Muller, 2007; Pradhan et al., 2007; Chen et al. 2010a, 2010b).

As a representative of using hand-crafted knowledge to resolve specific kinds of predicates, Byron (2002) proposed a semantic filter as a complement to salience calculations in resolving event pronouns. The semantic filter was constructed by using a set of hand-crafted constraints on some specific domains. Obviously, this approach is not suitable for general event pronoun resolution. In addition, the effec-

* Corresponding author

[1] In the coordinated structure, we pick the first predicate as the main predicate.

tiveness of this approach is unclear due to the lack of performance evaluation on an annotated corpus.

Among the learning-based methods to event anaphora resolution, Chen et al. (2010a) explored various kinds of positional, lexical and syntactic features for event pronoun resolution, which turned out quite different from conventional entity pronoun resolution. Besides, they studied the importance of structured syntactic information by incorporating such information into event pronoun resolution via a composite kernel. Chen et al. (2010b) extended their previous work from event pronoun resolution to general event anaphora resolution by considering other types of event anaphors, such as definite noun phrases, and achieved an encouraging performance on the Onto Notes English corpus.

Besides, there are some studies which integrate event anaphora resolution with entity anaphora resolution. For example, Pradhan et al. (2007) proposed a unified event and entity anaphora resolution framework based on a set of widely-used features which have been proven to be effective for entity anaphora resolution. However, they did not report the performance of their framework on event anaphora resolution. Alternatively, Muller (2007) constructed a logistic regression model and resolved event and entity pronouns together. For event pronoun resolution, he achieved 11.94% in F1-measure and found that the types of information effective for event pronoun resolution are quite different from those for entity pronoun resolution. From this respect, it seems better to independently explore event anaphora resolution first and then explore its possible integration (e.g. joint learning) with entity anaphora resolution.

Motivated by the success of the tree kernel-based framework to entity pronoun resolution (Yang et al. 2006; Zhou et al. 2008), this paper focuses on the proposal of a tree kernel-based framework to event pronoun resolution. Although the structured syntactic information effective for event pronoun resolution may be quite different from those for entity pronoun resolution, we believe structured syntactic information should play an important role for event pronoun resolution too. In particular, this paper explores various kinds of competitive information in tree kernel-based event pronoun resolution.

3 Framework of Event Pronoun Resolution

Our tree kernel-based event pronoun resolution framework adopts the common learning-based model for entity anaphora resolution, as described by Soon et al. (2001), and we use the SVM-light toolkit (Joakim, 1998) [2] with the convolution tree kernel function SVMlight–TK (Moschitti, 2004) [3] to compute the similarity between two parse trees.

3.1 Generating Training and Testing Instances

During training, consider a coreference chain A1-A2-A3-A4 found in an annotated training document. For entity anaphora resolution, pairs of noun phrases in the chain that are immediately adjacent (i.e., A1-A2, A2-A3, A3-A4) are used to generate the positive training examples with the former noun phrase in a pair the antecedent and the later the anaphor. In comparison for event pronoun resolution, assume A1 an event predicate, A2 a pronoun, A3 and A4 general noun phrases. Among the above three pairs (i.e. A1-A2, A2-A3 and A3-A4), only A1-A2 is a positive instance for event anaphora resolution while the other two pairs should be better considered in conventional entity anaphora resolution. For each positive pair, e.g. A1-A2, we can simply find any predicates[4] occurring between the event anaphor A2 and the event predicate A1, and pair each of them with A2 as a negative instance. Similarly, testing instances can be generated except that only the preceding predicates[5] in current and previous two sentences will be paired with the event anaphor.

Based on the generated training instances, we can build a binary classifier. To overcome the class imbalance problem between the positive and negative classes, a random down-sampling method (Kubat and Matwin, 1997) is adopted to reduce the majority class samples. As a result, the ratio of the positive and negative instances is reduced from about 1:3 to about 1:1.5 in our experimentation on the Onto Notes English corpus (Release 3.0).

3.2 Generating Syntactic Parse Tree Structure

In the literature, tree kernel-based methods have been explored in conventional entity anaphora resolution to a certain extent and achieved comparable performance with the dominated feature-based methods (Yang et al. 2006; Zhou et al. 2008). One main advantage of kernel-based methods is that they are very effective at reducing the burden of feature engineering for structured objects. For both entity and event anaphora resolution, a proper syntactic parse tree structure that covers an anaphor and its antecedent candidate could provide us much necessary syntactic information.

Normally, syntactic parsing is done on the sentence level. To deal with the cases that an anaphor and an antecedent candidate do not occur in the same sentence, we construct a pseudo parse tree for an entire text by attaching the parse trees of all its sentences to an upper "VNODE" node, similar to Yang et al. (2006).

Given a full syntactic parse tree and an anaphor in consideration, one key issue is how to choose a proper syntactic parse tree structure to well cover necessary 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. As baselines, we examine three tree expansion schemes, as

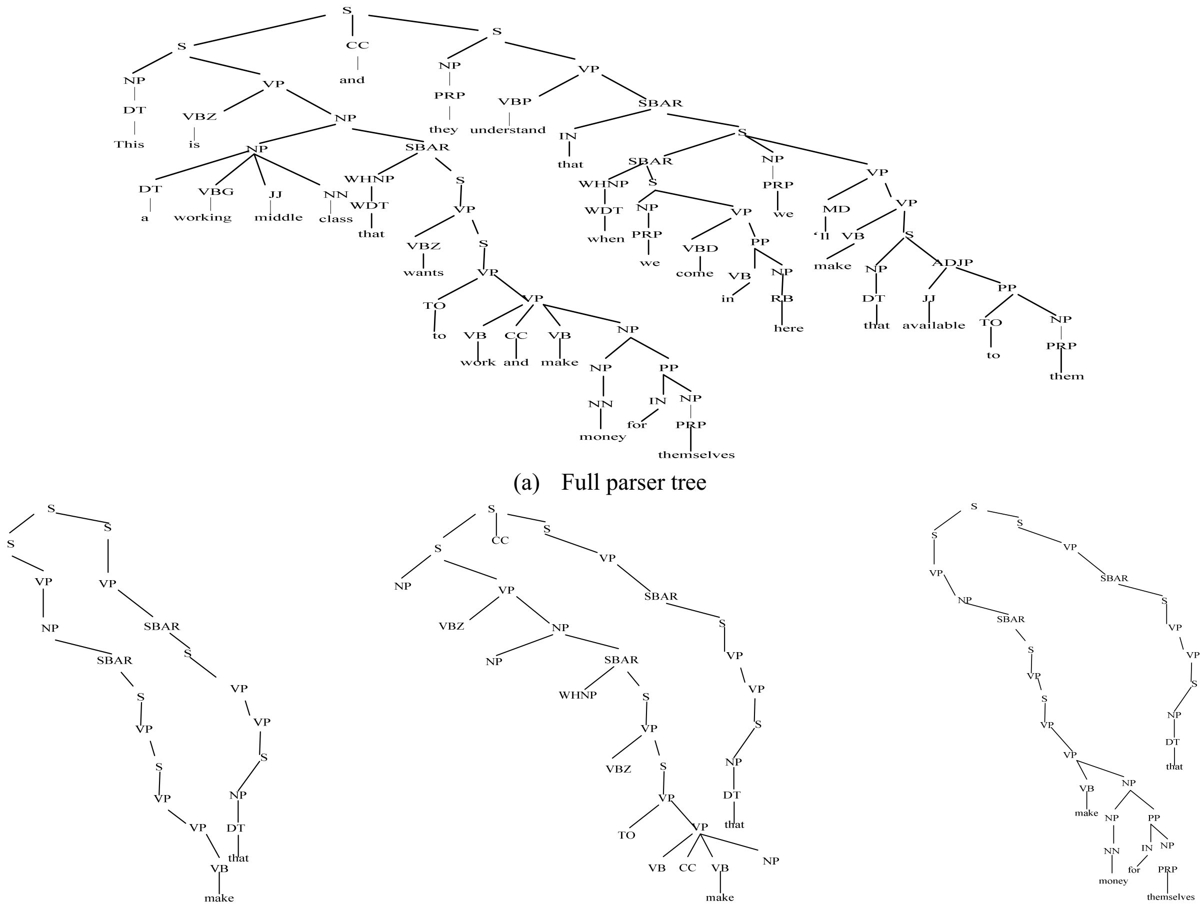
[2] http://svmlight.joachims.org/

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

[4] Not all the predicates between the antecedent and the anaphor are necessarily paired to generate negative instances. A set of constraints (e.g. those described by Byron (2002)) can be normally applied to filter out unlikely-referred ones. For example, a predicate appearing in a parenthesis can be safely filtered out.

[5] During testing, we employ the same constraints as used in training phase to filter out unlikely-referred predicates.

described in Yang et al, (2006), to generate a proper parse tree structure:



(a) Full parser tree

(b) Min-Expansion Tree (c) Simple-Expansion Tree (d) Full-Expansion Tree

Figure 1: Different tree expansion schemes to generate a parse tree structure

1) **Min-Expansion Tree (MT)**: the sub-tree which only includes the shortest path connecting the anaphor and the antecedent candidate.
2) **Simple-Expansion Tree (ST)**: the Min-Expansion Tree extended with the 1st sibling of the antecedent candidate. While the Min-Expansion Tree could describe the syntactic relationship between the candidate and the anaphor, it is incapable of capturing the syntactic properties of the antecedent candidate itself. Therefore, some context surrounding the antecedent candidate is included in the Simple-Expansion Tree
3) **Full-Expansion Tree (FT)**: another extension of the Min-Expamsion Tree. While the right sub-tree of the Full-Expansion Tree is a Min-Expansion Tree which only includes the shortest path connecting the anaphor and the lowest common node of the anaphor and the antecedent candidate, its left sub-tree is enclosed by the short path linking the lowest common node and the antecedent candidate.

Figure 1 illustrates an example of the three parse tree structures given the sentence "This is a working middle class that wants to work and ***make*** money for themselves and they understand that when we come in here, we'll make ***that*** available to them." with "***make***" as the antecedent candidate and "***that***" as the anaphor.

3.3 Incorporating Competitive Information

To better capture structured information in the parse tree structure, this paper presents a new tree expansion scheme by including various kinds of necessary competitive information in a parse tree. The intuition behind our scheme is that the related competitive information, such as the event pronoun predicate (i.e. the predicate of the event pronoun), event antecedent competitors and event pronoun competitors between the anaphor and the considered antecedent candidate, plays a critical role in event pronoun resolution.

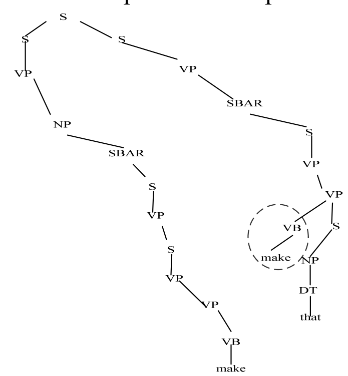
Given an anaphor and an antecedent candidate, e.g. "make" and "that" as shown in Figure 1, the new tree expansion scheme works as follows:

1) Determining the Min-Expansion Tree (the shortest path), as shown in Figure 1(b).
2) Attaching the event anaphor predicate. As shown in Figure 2(a). The event anaphor predicate is useful for determining the grammatical role of the anaphor. In Figure 2(a), the sub-tree inside the dash circle is the event anaphor predicate. Our statistics on the Onto Notes English corpus (Release 3.0) shows that most (72.62%)

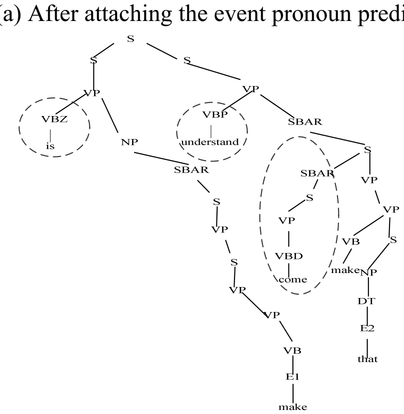
of event pronouns occur in the subject position, and others act no grammatical role at all.

3) Attaching all the event antecedent competitors along the corresponding paths to the shortest path, as shown in Figure 2(b). In Figure 2(b), the sub-trees inside the dash circles are the event antecedent competitors. In some sense, this is a natural extension of the twin-candidate learning approach as proposed in Yang et al. (2003), which explicitly models the competition between two antecedent candidates.

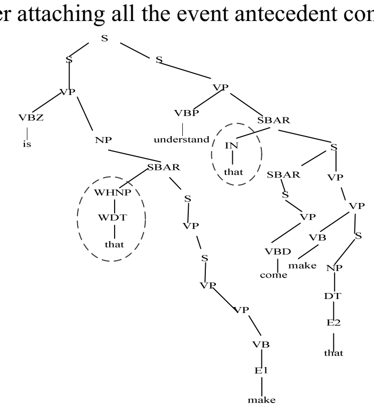
3) Attaching all the event pronoun competitors along the corresponding paths to the shortest path, as shown in Figure 2(c). In Figure 2(c), the sub-trees inside the dash circles are the event pronoun competitors.



(a) After attaching the event pronoun predicate



(b) After attaching all the event antecedent competitors



(c) After attaching all the event pronoun competitors

Figure 2. New tree expansion scheme in generating proper parse tree structure

In order to distinguish between the event pronoun competitors, the event antecedent competitors and the event pronoun predicate, we explicitly mark up the event antecedent candidate and the event pronoun by appending a string tag "E1" and "E2" in their respective POS nodes.

3.4 Convolution Tree Kernel

Given any tree expansion scheme, e.g. the new tree expansion scheme as described in the last subsection, the similarity between two parse trees is calculated using a convolution tree kernel. For more details about the kernel, please refer to Collins and Duffy (2002) and Moschitti (2004).

4 Experiments and Discussions

This section systematically evaluates the performance of our tree kernel-based event pronoun resolution framework on the Onto Notes English corpus (Release 3.0).

4.1 Experimental Setting

The Onto Notes English corpus (Release 3.0) contains 300K words of English newswire data (from the Wall Street Journal) and 200K words of English broadcast news data (from ABC, CNN, NBC, Public Radio International and Voice of America). Table 1 shows the statistics of the corpus. From Table 1 we can find that, only about 9% of coreference chains are event related. Among 3550 event pronoun candidates (i.e. all the occurrences of "this", "that" and "it", which function as pronoun), only 504 are event pronouns, accounting for about 14%. This indicates the difficulty of indentifying event pronouns.

Category	Num
Coreference chains	8154
Coreference chains related with events	737
Event pronoun candidates	3550
Event pronouns	504

Table 1: Statistics on Onto Notes English corpus (Release 3.0)

	Sentences	Main predicates	Event pronoun competitors	Event antecedent competitors
0	174	347	405	127
1	274	135	75	141
2	37	16	20	92
3	8	0	3	68
4	5	1	0	33
5	2	2	1	18
6	1	1	0	6
7	1	2	0	5
8	1	0	0	3
9	0	0	0	2
>=10	1	0	0	9
Total	504	504	504	504

Table 2: Various distributions of event pronouns (Note that the statistics are based on the output of the Charniak parser. Therefore, the statistics is prone to syntactic parsing errors.)

Table 2 shows various distributions of event pronouns over different distances between an event pronoun and its antecedent. From Table 2, we can find that:

1) About 96% of event pronouns refer to an event in current and previous two sentences (Column 2);
2) About 99% of event pronouns have at most 2 main predicates between its antecedent and the anaphor itself (Column 3);
3) About 80% of event pronouns have no event pronoun competitors between its antecedent and the anaphor itself (Column 4);
4) Although event pronouns tend to refer to local events, simply using the most recent predicate to resolve an event pronoun is bound to fail (Column 5).

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. In addition, the corpus is parsed using the Charniak parser. Finally, we use the SVM-light toolkit with the tree kernel function as the classifier.

For performance evaluation, we report the performance of event pronoun resolution with 10-fold cross validation 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

Baseline Parse Tree Structures

Table 3 shows the effectiveness of the three baseline parse tree structures on event pronoun resolution. It shows that MT(Min-Expansion Tree) achieves the best performance of 46.91 in F1-measure and FT(Full-Expansion Tree) achieves the worst performance on both precision and recall. Compared with MT(Min-Expansion Tree), ST(Simple-Expansion Tree) performs slightly worse in F1-measure, much due to much lower precision It is interesting that MT performs best since it contains the least information. This may be also due to possible inclusion of unnecessary information for ST and FT, in particular the later. It raises the problem of how to generate a proper parse tree structure for event pronoun resolution by only including necessary structured syntactic information.

Parse Tree Structure	P(%)	R(%)	F
FT (Full-Expansion Tree)	33.92	48.89	40.05
ST (Simple-Expansion Tree)	36.33	60.0	45.26
MT (Min-Expansion Tree)	43.24	51.25	46.91

Table 3: Performance of baseline parse tree structures

New Parse Tree Structure

Table 4 shows the contribution of different kinds of competitive information in our new tree expansion scheme, which builds on the best (also simplest) baseline structure, MT (Min-Expansion Tree). It shows that:

1) The inclusion of the event anaphor predicate significantly improves the performance by 4.38(***) in F1-measure. This suggests the impact of the event anaphor predicate in determining the grammatical role of the anaphor.
2) Only including event antecedent competitors improves the F1-measure by 1.00(**) much due to the more increase in precision compared with the decrease in recall.
3) The inclusion of event pronoun competitors harms the performance by 4.83(***) in F1-measure. This suggests that using such event pronoun-related competitive information alone would have a negative effect on the performance, much due to the lack of the discriminative power themselves and the need of the help from other kinds of competitive information (as shown in Item 4).
4) Including both competitive predicates (event antecedent competitors) and competitive event pronouns (event pronoun competitors) significantly improves the performance by 2.93(***) in F1-measure. This suggests that these two types of competitive information work as collaborators to improve the performance of event pronoun resolution.
5) Further inclusion of the event anaphor predicate further improves the performance by 5.90(***) in F1-measure, due to both much higher precision and recall. This suggests that the event anaphor predicate and the competitors of both event antecedents and event pronouns are much complementary on event pronoun resolution.

	P(%)	R(%)	F
MT	43.24	51.25	46.91
+Event Anaphor Predicate (EAP)	41.67	66.67	51.29
+Event Antecedent Competitors (EAC)	47.67	48.15	47.91
+Event Pronoun Competitors (EPC)	38.89	45.83	42.08
+EAC+EPC	48.83	49.06	48.94
+EAP+EAC+EPC	47.06	65.71	54.84

Table 4: Contribution of different kinds of competitive information

Distance	P(%)	R(%)	F
<=0	60.0	88.89	71.64
<=1	55.56	72.22	62.80
<=2	47.06	65.71	54.84

Table 5: Performance over different sentence distances

Table 5 shows the performance over different sentence distances. From the results, it is not surprising to observe that, intra-sentence event pronoun resolution is much easier than inter-sentence event pronoun resolution.

Comparison with the State-of-the-Art

Table 6 illustrates the performance of Chen et al. (2010a) using different parse tree structures. Chen et al (2010a) achieved the performance of 40.6 in F1measure using a set of flat features via a feature-based method, and the best performance of 44.4 in F1-measure using MT(the Min-Expansion Tree) via a tree kernel-based method. In our study, our tree kernel-based method with the baseline

MT(Min-Expansion Tree) achieves the performance of 46.91 in F1-measure, outperforming Chen et al (2010a) by about 2.5 in F1-measure.

Chen et al (2010a) further studied different ways of combining a set of flat features and a parse tree structure to improve the performance and achieved the best performance of 47.2 in F1-measure when combining the set of flat features with ST(Simple-Expansion Tree). They further looked into the incorporation of negative instances from non-event anaphoric pronoun and achieved the best performance of 54.9 in F1-measure. In comparison, our tree kernel-based framework adopts a simple tree expansion scheme to generate a single parse tree structure by better capturing various kinds of competitive information and achieves the performance of 54.84 in F1-measure.

	P(%)	R(%)	F
Flat	40.6	40.6	40.6
Min-Exp	35.5	59.6	44.4
Simple-Exp+Flat	42.3	53.4	47.2
Balanced Negative	59.9	50.6	54.9

Table 6: Performance of Chen et al. (2010) on event pronoun resolution

5 Conclusion and Further Work

This paper studies the impact of competitive information related with the anaphor and the antecedent candidate on event pronoun resolution. In particular, a tree kernel-based framework is proposed and a simple but effective tree expansion scheme is introduced to capture various kinds of competitive information, such as the predicate of the event pronoun, event antecedent competitors and event pronoun competitors. Evaluation on the Onto Notes English corpus (Release 3.0) shows the effectiveness of such competitive information for event pronoun resolution.

For further work, we will explore more structured syntactic information in event anaphora resolution. In addition, we will study joint learning of entity anaphora resolution and event anaphora resolution.

Acknowledgments

This research was supported by Projects 60873150, 90920004 and 61003153 under the National Natural Science Foundation of China.

References

[Byron, 2002] Byron D. *Resolving Pronominal Reference to Abstract Entities*. ACL'2002

[Chen *et al*., 2010(a)]Chen B., Su J. and Tan C.L. *A Twin-Candidate Based Approach for Event Pronoun Resolution using Composite Kernel*. COLING'2010

[Chen *et al*., 2010(b)] Chen B., Su J. and Tan C.L. *Resolving Event Noun Phrases to Their Verbal Mentions*. EMNLP'2010

[Collins and Duffy, 2002] Collins M. and Duffy N. 2001. *Covolution kernels for natural language* [C]. NIPS' 2001:625-632.

[Kubat and Matwin,1997] Kubat M. and Matwin S. *Addressing the curse of imbalanced data set: One sided sampling*. In Proceedings of the Fourteenth International Conference on Machine Learning, 1997:179-186

[Kong *et al*., 2010] Kong F., Zhou G.D., Qian L.H. and Zhu Q.M. *Dependency-driven Anaphoricity Determination for Coreference Resolution*. COLING'2010

[Moschitti, 2004] Moschitti A. 2004. *A Study on Convolution Kernels for Shallow Semantic Parsing*, ACL'2004:335-342

[Muller, 2007] Muller C. *Resoving it, this, and that in unrestricted multi-party dialog.* ACL'2007:816-823

[Ng and Cardie, 2002] Ng V. and Cardie C. 2002. *Improving machine learning approaches to coreference resolution*. ACL'2002: 104-111

[Ng, 2007] Ng. V. 2007. *Semantic Class Induction and Coreference Resolution*. ACL'2007 536-543.

[Ng, 2009] V. Ng 2009. *Graph-cut based anaphoricity determination for coreference resolution*. NAACL'2009:575-583

[Pradhan *et al*., 2007] Pradhan S., Ramshaw L., Weischedel R., Macbride J. and Micciulla L. *Unrestricted Coreference: Identifying Entities and Events in OntoNotes*. ICSC'2007

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

[Yang *et al*., 2004] Yang X.F., Su J., Zhou G.D. and Tan C.L. 2004. *Improving pronoun resolution by incorporating coreferential information of candidates* [A]. ACL'2004: 127- 134

[Yang *et al*., 2006]Yang X.F., Su J. and Chew C.L., 2006. *Kernel-based pronoun resolution with structured syntactic knowledge*. COLING-ACL'2006

[Yang *et al*., 2008] Yang X.F., Su J. and Tan C.L. 2008. *A Twin-Candidate Model for Learning-Based Anaphora Resolution*. Computational Linguistics 34(3):327-356

[Vilain et al. 1995] Vilain M., Burger J., Aberdeen J., Connolly D. and Hirschman L. *A Model-Theoretic Coreference Scoring Scheme*. Proceedings of Sixth Message Understanding Conference (MUC-6), pp.45-52, 1995.

[Zhou et al., 2008] Zhou G.D., Kong F. and Zhu Q.M. 2008. Context-sensitive convolution tree kernel for pronoun resolution. IJCNLP'2008:25-31