

Context-Sensitive Convolution Tree Kernel for Pronoun Resolution

ZHOU GuoDong KONG Fang ZHU Qiaoming

JiangSu Provincial Key Lab for Computer Information Processing Technology

School of Computer Science and Technology

Soochow Univ. Suzhou, China 215006

Email: {gdzhou, kongfang, qmzhu}@suda.edu.cn

Abstract

This paper proposes a context-sensitive convolution tree kernel for pronoun resolution. It resolves two critical problems in previous researches in two ways. First, given a parse tree and a pair of an anaphor and an antecedent candidate, it implements a dynamic-expansion scheme to automatically determine a proper tree span for pronoun resolution by taking predicate- and antecedent competitor-related information into consideration. Second, it applies a context-sensitive convolution tree kernel, which enumerates both context-free and context-sensitive sub-trees by considering their ancestor node paths as their contexts. Evaluation on the ACE 2003 corpus shows that our dynamic-expansion tree span scheme can well cover necessary structured information in the parse tree for pronoun resolution and the context-sensitive tree kernel much outperforms previous tree kernels.

1 Introduction

It is well known that syntactic structured information plays a critical role in many critical NLP applications, such as parsing, semantic role labeling, semantic relation extraction and co-reference resolution. However, it is still an open question on what kinds of syntactic structured information are effective and how to well incorporate such structured information in these applications.

Much research work has been done in this direction. Prior researches apply feature-based methods to select and define a set of flat features, which can be mined from the parse trees, to represent particular structured information in the parse tree, such as the grammatical role (e.g. subject or object), according to the particular application. Indeed, such feature-based methods have been widely applied in

parsing (Collins 1999; Charniak 2001), semantic role labeling (Pradhan et al 2005), semantic relation extraction (Zhou et al 2005) and co-reference resolution (Lapin and Leass 1994; Aone and Bennett 1995; Mitkov 1998; Yang et al 2004; Luo and Zitouni 2005; Bergsma and Lin 2006). The major problem with feature-based methods on exploring structured information is that they may fail to well capture complex structured information, which is critical for further performance improvement.

The current trend is to explore kernel-based methods (Haussler, 1999) which can implicitly explore features in a high dimensional space by employing a kernel to calculate the similarity between two objects directly. In particular, the kernel-based methods could be very effective at reducing the burden of feature engineering for structured objects in NLP, e.g. the parse tree structure in coreference resolution. During recent years, various tree kernels, such as the convolution tree kernel (Collins and Duffy 2001), the shallow parse tree kernel (Zelenko et al 2003) and the dependency tree kernel (Culota and Sorensen 2004), have been proposed in the literature. Among previous tree kernels, the convolution tree kernel represents the state-of-the-art and have been successfully applied by Collins and Duffy (2002) on parsing, Moschitti (2004) on semantic role labeling, Zhang et al (2006) on semantic relation extraction and Yang et al (2006) on pronoun resolution.

However, there exist two problems in Collins and Duffy's kernel. The first is that the sub-trees enumerated in the tree kernel are context-free. That is, each sub-tree enumerated in the tree kernel does not consider the context information outside the sub-tree. The second is how to decide a proper tree span in the tree kernel computation according to the particular application. To resolve above two problems, this paper proposes a new tree span scheme and applies a new tree kernel and to better capture syntactic structured information in pronoun

resolution, whose task is to find the corresponding antecedent for a given pronominal anaphor in text.

The rest of this paper is organized as follows. In Section 2, we review related work on exploring syntactic structured information in pronoun resolution and their comparison with our method. Section 3 first presents a dynamic-expansion tree span scheme by automatically expanding the shortest path to include necessary structured information, such as predicate- and antecedent competitor-related information. Then it presents a context-sensitive convolution tree kernel, which not only enumerates context-free sub-trees but also context-sensitive sub-trees by considering their ancestor node paths as their contexts. Section 4 shows the experimental results. Finally, we conclude our work in Section 5.

2 Related Work

Related work on exploring syntactic structured information in pronoun resolution can be typically classified into three categories: parse tree-based search algorithms (Hobbs 1978), feature-based (Lappin and Leass 1994; Bergsma and Lin 2006) and tree kernel-based methods (Yang et al 2006).

As a representative for parse tree-based search algorithms, Hobbs (1978) found the antecedent for a given pronoun by searching the parse trees of current text. It processes one sentence at a time from current sentence to the first sentence in text until an antecedent is found. For each sentence, it searches the corresponding parse tree in a left-to-right breadth-first way. The first antecedent candidate, which satisfies hard constraints (such as gender and number agreement), would be returned as the antecedent. Since the search is completely done on the parse trees, one problem with the parse tree-based search algorithms is that the performance would heavily rely on the accuracy of the parse trees. Another problem is that such algorithms are not good enough to capture necessary structured information for pronoun resolution. There is still a big performance gap even on correct parse trees.

Similar to other NLP applications, feature-based methods have been widely applied in pronoun resolution to explore syntactic structured information from the parse trees. Lappin and Leass (1994) derived a set of salience measures (e.g. subject, object or accusative emphasis) with manually

assigned weights from the syntactic structure output by McCord’s Slot Grammar parser. The candidate with the highest salience score would be selected as the antecedent. Bergsma and Lin (2006) presented an approach to pronoun resolution based on syntactic paths. Through a simple bootstrapping procedure, highly co-reference paths can be learned reliably to handle previously challenging instances and robustly address traditional syntactic co-reference constraints. Although feature-based methods dominate on exploring syntactic structured information in the literature of pronoun resolution, there still exist two problems with them. One problem is that the structured features have to be selected and defined manually, usually by linguistic intuition. Another problem is that they may fail to effectively capture complex structured parse tree information.

As for tree kernel-based methods, Yang et al (2006) captured syntactic structured information for pronoun resolution by using the convolution tree kernel (Collins and Duffy 2001) to measure the common sub-trees enumerated from the parse trees and achieved quite success on the ACE 2003 corpus. They also explored different tree span schemes and found that the simple-expansion scheme performed best. One problem with their method is that the sub-trees enumerated in Collins and Duffy’s kernel computation are context-free, that is, they do not consider the information outside the sub-trees. As a result, their ability of exploring syntactic structured information is much limited. Another problem is that, among the three explored schemes, there exists no obvious overwhelming one, which can well cover syntactic structured information.

The above discussion suggests that structured information in the parse trees may not be well utilized in the previous researches, regardless of feature-based or tree kernel-based methods. This paper follows tree kernel-based methods. Compared with Collins and Duffy’s kernel and its application in pronoun resolution (Yang et al 2006), the context-sensitive convolution tree kernel enumerates not only context-free sub-trees but also context-sensitive sub-trees by taking their ancestor node paths into consideration. Moreover, this paper also implements a dynamic-expansion tree span scheme by taking predicate- and antecedent competitor-related information into consideration.

3 Context Sensitive Convolution Tree Kernel for Pronoun Resolution

In this section, we first propose an algorithm to dynamically determine a proper tree span for pronoun resolution and then present a context-sensitive convolution tree kernel to compute similarity between two tree spans. In this paper, all the texts are parsed using the Charniak parser (Charniak 2001) based on which the tree span is determined.

3.1 Dynamic-Expansion Tree Span Scheme

Normally, 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 “S” node, similar to Yang et al (2006).

Given the parse tree of a text, the problem is how to choose a proper tree span to well cover syntactic structured information in the tree kernel computation. Generally, the more a tree span includes, the more syntactic structured information would be provided, at the expense of more noisy information. Figure 2 shows the three tree span schemes explored in Yang et al (2006): Min-Expansion (only including the shortest path connecting the anaphor and the antecedent candidate), Simple-Expansion (containing not only all the nodes in Min-Expansion but also the first level children of these nodes) and Full-Expansion (covering the sub-tree between the anaphor and the candidate), such as the sub-trees inside the dash circles of Figures 2(a), 2(b) and 2(c) respectively. It is found (Yang et al 2006) that the simple-expansion tree span scheme performed best on the ACE 2003 corpus in pronoun resolution. This suggests that inclusion of more structured information in the tree span may not help in pronoun resolution.

To better capture structured information in the parse tree, this paper presents a dynamic-expansion scheme by trying to include necessary structured information in a parse tree. The intuition behind our scheme is that predicate- and antecedent competitor- (all the other compatible¹ antecedent candidates between the anaphor and the considered antecedent candidate) related information plays a critical role in pronoun resolution. Given an ana-

phor and an antecedent candidate, e.g. “Mary” and “her” as shown in Figure 1, this is done by:

- 1) Determining the min-expansion tree span via the shortest path, as shown in Figure 1(a).
- 2) Attaching all the antecedent competitors along the corresponding paths to the shortest path. As shown in Figure 1(b), “the woman” is attached while “the room” is not attached since the former is compatible with the anaphor and the latter is not compatible with the anaphor. In this way, the competition between the considered candidate and other compatible candidates can be included in the tree span. In some sense, this is a natural extension of the twin-candidate learning approach proposed in Yang et al (2003), which explicitly models the competition between two antecedent candidates.
- 3) For each node in the tree span, attaching the path from the node to the predicate terminal node if it is a predicate-headed node. As shown in Figure 1(c), “said” and “bit” are attached.
- 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. As shown in Figure 1(d), the left child of the “SBAR” node, the “NP” node, is removed and the sub-tree (NP the/DT woman/NN) is attached to the “SBAR” node directly.

To show the difference among min-, simple-, full- and dynamic-expansion schemes, Figure 2 compares them for three different sentences, given the anaphor “her/herself” and the antecedent candidate “Mary”. It shows that:

- Min-, simple- and full-expansion schemes have the same tree spans (except the word nodes) for the three sentences regardless of the difference among the sentences while the dynamic-expansion scheme can adapt to difference ones.
- Normally, the min-expansion scheme is too simple to cover necessary information (e.g. “the woman” in the 1st sentence is missing).
- The full-expansion scheme can cover all the information at the expense of much noise (e.g. “the man in that room” in the 2nd sentence).
- The simple-expansion scheme can cover some necessary predicate-related information (e.g. “said” and “bit” in the sentences). However, it may introduce some noise (e.g. the left child of

¹ With matched number, person and gender agreements.

the “SBAR” node, the “NP” node, may not be necessary in the 2nd sentence) and ignore necessary antecedent competitor-related information (e.g. “the woman” in the 1st sentence).

- The dynamic-expansion scheme normally works well. It can not only cover predicate-

related information but also structured information related with the competitors of the considered antecedent candidate. In this way, the competition between the considered antecedent candidate and other compatible candidates can be included in the dynamic-expansion scheme.

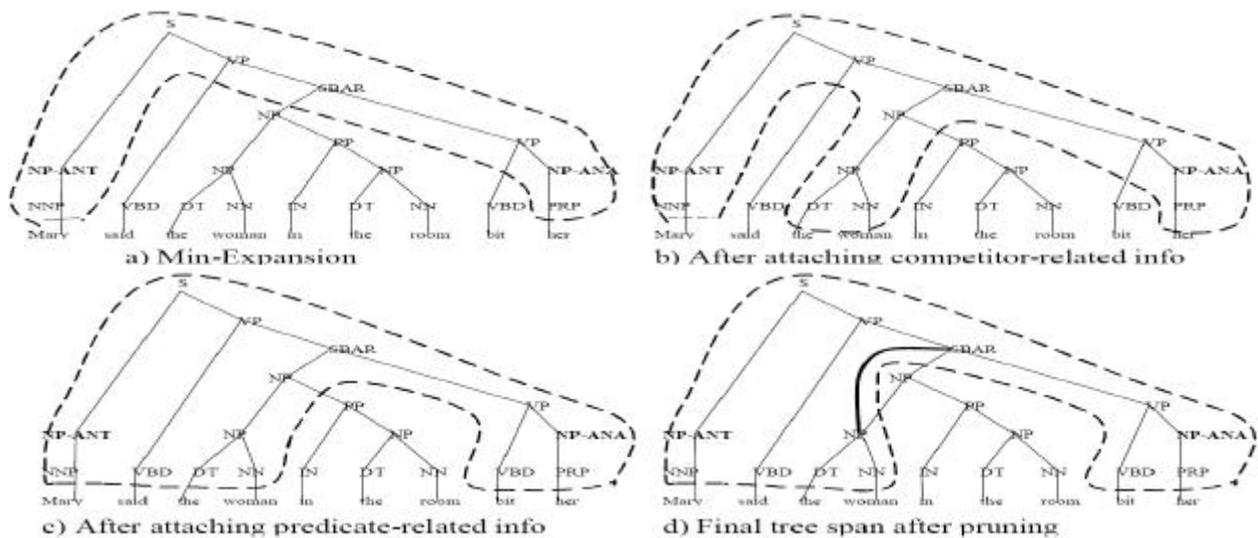


Figure 1: Dynamic-Expansion Tree Span Scheme

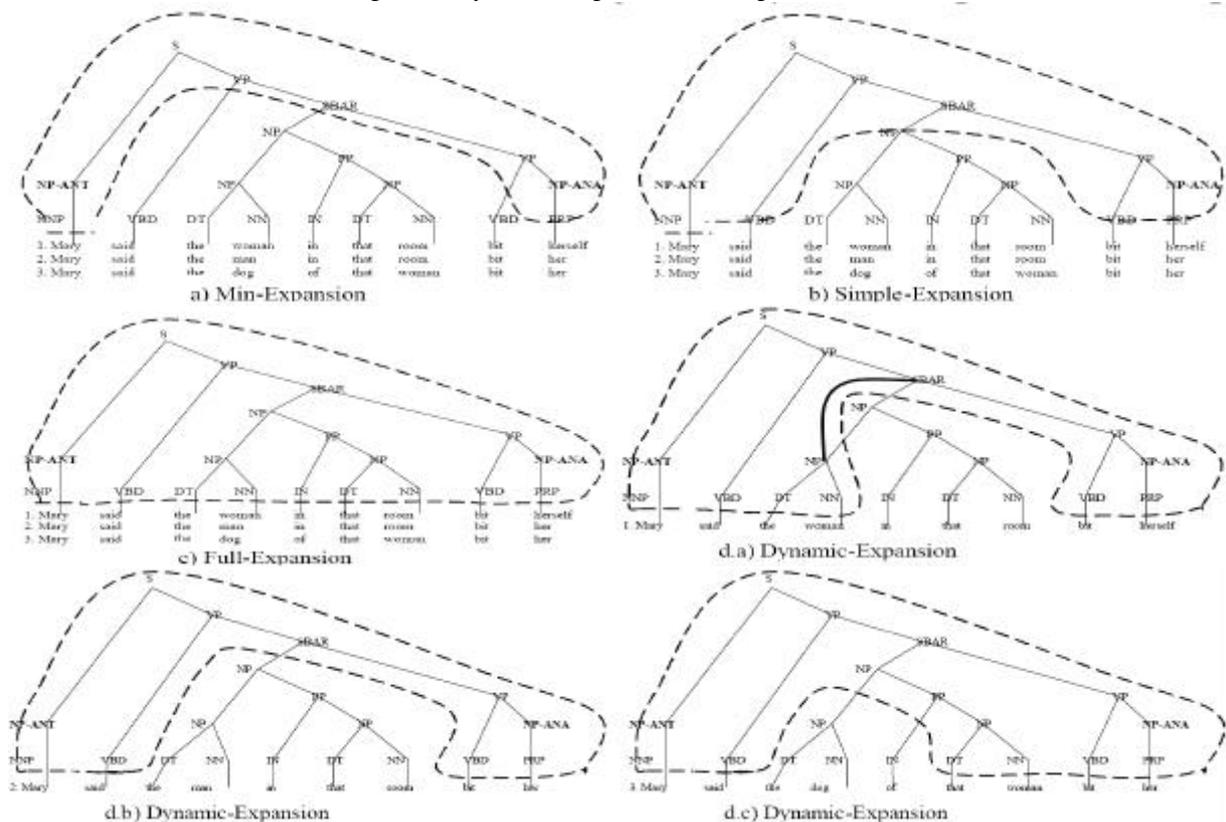


Figure 2: Comparison of Min-, Simple-, Full- and Dynamic-Expansions: More Examples

3.2 Context-Sensitive Convolution Tree Kernel

Given any tree span scheme, e.g. the dynamic-expansion scheme in the last subsection, we now study how to measure the similarity between two tree spans using a convolution tree kernel.

A convolution kernel (Haussler D., 1999) aims to capture structured information in terms of sub-structures. As a specialized convolution kernel, the convolution tree kernel, proposed in Collins and Duffy (2001), counts the number of common sub-trees (sub-structures) as the syntactic structure similarity between two parse trees. This convolution tree kernel has been successfully applied by Yang et al (2006) in pronoun resolution. However, there is one problem with this tree kernel: the sub-trees involved in the tree kernel computation are context-free (That is, they do not consider the information outside the sub-trees.). This is contrast to the tree kernel proposed in Culota and Sorensen (2004) which is context-sensitive, that is, it considers the path from the tree root node to the sub-tree root node. In order to integrate the advantages of both tree kernels and resolve the problem in Collins and Duffy’s kernel, this paper applies the same context-sensitive convolution tree kernel, proposed by Zhou et al (2007) on relation extraction. It works by taking ancestral information (i.e. the root node path) of sub-trees into consideration:

$$K_C(T[1], T[2]) = \sum_{i=1}^m \sum_{\substack{n_1^i[1] \in N_1^i[1] \\ n_1^i[2] \in N_1^i[2]}} \Delta(n_1^i[1], n_1^i[2]) \quad (1)$$

where $N_1^i[j]$ is the set of root node paths with length i in tree $T[j]$ while the maximal length of a root node path is defined by m ; and $\Delta(n_1^i[1], n_1^i[2])$ counts the common context-sensitive sub-trees rooted at root node paths $n_1^i[1]$ and $n_1^i[2]$. In the tree kernel, a sub-tree becomes context-sensitive via the “root node path” moving along the sub-tree root. For more details, please refer to Zhou et al (2007).

4 Experimentation

This paper focuses on the third-person pronoun resolution and, in all our experiments, uses the ACE 2003 corpus for evaluation. This ACE corpus

contains ~3.9k pronouns in the training data and ~1.0k pronouns in the test data.

Similar to Soon et al (2001), an input raw 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 pronoun and one of its antecedent candidates. 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. In this paper, the NPs occurring within the current and previous two sentences are taken as the initial antecedent candidates, and those with mismatched number, person and gender agreements are filtered out. On average, an anaphor has ~7 antecedent candidates. The performance is evaluated using F-measure instead of accuracy since evaluation is done on all the pronouns occurring in the data.

Scheme/ m	1	2	3	4
Min	78.5	79.8	80.8	80.8
Simple	79.8	81.0	81.7	81.6
Full	78.3	80.1	81.0	81.1
Dynamic	80.8	82.3	83.0	82.9

Table 1: Comparison of different context-sensitive convolution tree kernels and tree span schemes (with entity type info attached at both the anaphor and the antecedent candidate nodes by default)

In this paper, the m parameter in our context-sensitive convolution tree kernel as shown in Equation (1) indicates the maximal length of root node paths and is optimized to 3 using 5-fold cross validation on the training data. Table 1 systematically evaluates the impact of different m in our context-sensitive convolution tree kernel and compares our dynamic-expansion tree span scheme with the existing three tree span schemes, min-,

simple- and full-expansions as described in Yang et al (2006). It also shows that that our tree kernel achieves best performance with $m = 3$ on the test data, which outperforms the one with $m = 1$ by ~ 2.2 in F-measure. This suggests that the parent and grandparent nodes of a sub-tree contain much information for pronoun resolution while considering more ancestral nodes doesnot further improve the performance. This may be due to that, although our experimentation on the training data indicates that more than 90% (on average) of subtrees has a root node path longer than 3 (since most of the subtrees are deep from the root node and more than 90% of the parsed trees are deeper than 6 levels in the ACE 2003 corpus), including a root node path longer than 3 may be vulnerable to the full parsing errors and have negative impact. It also shows that our dynamic-expansion tree span scheme outperforms min-expansion, simple-expansion and full-expansion schemes by ~ 2.4 , ~ 1.2 and ~ 2.1 in F-measure respectively. This suggests the usefulness of dynamically expanding tree spans to cover necessary structured information in pronoun resolution. In all the following experiments, we will apply our tree kernel with $m=3$ and the dynamic-expansion tree span scheme by default, unless specified.

We also evaluate the contributions of antecedent competitor-related information, predicate-related information and pruning in our dynamic-expansion tree span scheme by excluding one of them from the dynamic-expansion scheme. Table 2 shows that 1) antecedent competitor-related information contributes much to our scheme; 2) predicate-related information contributes moderately; 3) pruning only has slight contribution. This suggests the importance of including the competition in the tree span and the effect of predicate-argument structures in pronoun resolution. This also suggests that our scheme can well make use of such predicate- and antecedent competitor-related information.

Dynamic Expansion	Effect
- Competitors-related Info	81.1(-1.9)
- Predicates-related Info	82.2 (-0.8)
- Pruning	82.8(-0.2)
All	83.0

Table 2: Contributions of different factors in our dynamic-expansion tree span scheme

Table 3 compares the performance of different tree span schemes for pronouns with antecedents in

different sentences apart. It shows that our dynamic-expansion scheme is much more robust than other schemes with the increase of sentences apart.

Scheme / #Sentences Apart	0	1	2
Min	86.3	76.7	39.6
Simple	86.8	77.9	43.8
Full	86.6	77.4	35.4
Dynamic	87.6	78.8	54.2

Table 3: Comparison of tree span schemes with antecedents in different sentences apart

5 Conclusion

Syntactic structured information holds great potential in many NLP applications. The purpose of this paper is to well capture syntactic structured information in pronoun resolution. In this paper, we proposes a context-sensitive convolution tree kernel to resolve two critical problems in previous researches in pronoun resolution by first automatically determining a dynamic-expansion tree span, which effectively covers structured information in the parse trees by taking predicate- and antecedent competitor-related information into consideration, and then applying a context-sensitive convolution tree kernel, which enumerates both context-free sub-trees and context-sensitive sub-trees. Evaluation on the ACE 2003 corpus shows that our dynamic-expansion tree span scheme can better capture necessary structured information than the existing tree span schemes and our tree kernel can better model structured information than the state-of-the-art Collins and Duffy’s kernel.

For the future work, we will focus on improving the context-sensitive convolution tree kernel by better modeling context-sensitive information and exploring new tree span schemes by better incorporating useful structured information. In the meanwhile, a more detailed quantitative evaluation and thorough qualitative error analysis will be performed to gain more insights.

Acknowledgement

This research is supported by Project 60673041 under the National Natural Science Foundation of China and Project 2006AA01Z147 under the “863” National High-Tech Research and Development of China.

References

- Aone C and Bennett W.W. (1995). Evaluating automated and manual acquisition of anaphora resolution strategies. *ACL'1995*:122-129.
- Bergsma S. and Lin D.K.(2006). Bootstrapping path-based pronoun resolution. *COLING-ACL'2006*: 33-40.
- Charniak E. (2001). Immediate-head Parsing for Language Models. *ACL'2001*: 129-137. Toulouse, France
- Collins M. (1999) Head-driven statistical models for natural language parsing. *Ph.D. Thesis*. University of Pennsylvania.
- Collins M. and Duffy N. (2001). Convolution Kernels for Natural Language. *NIPS'2001*: 625-632. Cambridge, MA
- Culotta A. and Sorensen J. (2004). Dependency tree kernels for relation extraction. *ACL'2004*. 423-429. 21-26 July 2004. Barcelona, Spain.
- Hausler D. (1999). Convolution Kernels on Discrete Structures. *Technical Report UCS-CRL-99-10*, University of California, Santa Cruz.
- Hobbs J. (1978). Resolving pronoun references. *Lingua*. 44:339-352.
- Joachims T. (1998). Text Categorization with Support Vector Machine: learning with many relevant features. *ECML-1998*: 137-142. Chemnitz, Germany
- Lappin S. and Leass H. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*. 20(4):526-561.
- Mitkov R. (1998). Robust pronoun resolution with limited knowledge. *COLING-ACL'1998*:869-875. Montreal, Canada.
- Moschitti A. (2004). A study on convolution kernels for shallow semantic parsing. *ACL'2004*:335-342.
- Pradhan S., Hacioglu K., Krugler V., Ward W., Martin J.H. and Jurafsky D. (2005). Support Vector Learning for Semantic Argument Classification. *Machine Learning*. 60(1):11-39.
- Soon W. Ng H.T.and Lim D. (2001). A machine learning approach to creference resolution of noun phrases. *Computational Linguistics*. 27(4): 521-544.
- Yang X.F., Zhou G.D., Su J. and Tan C.L., Coreference Resolution Using Competition Learning Approach, *ACL'2003*:176-183. Sapporo, Japan, 7-12 July 2003.
- Yang X.F., Su J. and Tan C.L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. *COLING-ACL'2006*: 41-48.
- Zelenko D., Aone C. and Richardella. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*. 3(Feb):1083-1106.
- Zhang M., Zhang J., Su J. and Zhou G.D. (2006). A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features. *COLING-ACL-2006*: 825-832. Sydney, Australia
- Zhou G.D., Su J. Zhang J. and Zhang M. (2005). Exploring various knowledge in relation extraction. *ACL'2005*. 427-434. 25-30 June, Ann Arbor, Michigan, USA.
- Zhou G.D., Zhang M., Ji D.H. and Zhu Q.M. (2007). Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. *EMNLP-CoNLL'2007*