Resolving Event Noun Phrases to Their Verbal Mentions

Chen Bin[1] **Su Jian**[2] **Tan Chew Lim**[1]

[1]National University of Singapore
{chenbin,tancl}@comp.nus.edu.sg

[2]Institute for Infocomm Research, Singapore
sujian@i2r.a-star.edu.sg

Abstract

Event Anaphora Resolution is an important task for cascaded event template extraction and other NLP study. Previous study only touched on event pronoun resolution. In this paper, we provide the first systematic study to resolve event noun phrases to their verbal mentions crossing long distances. Our study shows various lexical, syntactic and positional features are needed for event noun phrase resolution and most of them, such as morphology relation, synonym and etc, are different from those features used for conventional noun phrase resolution. Syntactic structural information in the parse tree modeled with tree kernel is combined with the above diverse flat features using a composite kernel, which shows more than 10% F-score improvement over the flat features baseline. In addition, we employed a twin-candidate based model to capture the pair-wise candidate preference knowledge, which further demonstrates a statistically significant improvement. All the above contributes to an encouraging performance of 61.36% F-score on OntoNotes corpus.

1 Introduction

Anaphora resolution, the task of resolving a given text expression to its referred expression in prior texts, is important for intelligent text processing systems. Most previous works on anaphora resolution aim at object anaphora which both the anaphor and its antecedent are mentions of the same real world object.

In contrast, an event anaphora[1] as defined in (Asher, 1993) is an anaphoric reference to an event, fact, and proposition which is representative of eventuality and abstract objects. Consider the following example:

*There has been **[the first break in the case]**.*
…
*The investigation into the attack which crippled the ship took a positive turn when Yemeni investigators **[discovered]** bomb making equipment in a house said to be close to the port where the ship is anchored.*
…
*Now that investigators have had **[their first major break]**, the confidence level here has risen, but U.S. officials still warn a long investigation lies ahead.*

The two anaphors ***[the first break in the case]*** and ***[their first major break]*** in the above example refer to the same event, "*Yemeni investigators **[discovered]** bomb making equipment.*" Here, we take the main verb of the event, ***[discovered]*** as the representation of this event and the antecedent of the two anaphors.

Event anaphora (both pronouns and noun phrases) contributes a significant proportion in an anaphora corpus. For example, OntoNotes has 19.97% of its entity chains contains at least one verb mention. Event anaphora resolution also provides critical links for cascaded event template extraction. It provides useful information for the further inference in other natural language processing (NLP) tasks such as discourse relation and entailment as well. Consider the following sentences from OntoNotes,

*"In northern Iraq, U.S. warplanes **[hit]** targets including a ridge east of Mosul, where Iraqi troops have been entrenched.*
*Two F Tomcats **[struck]** the targets.*
*After **[today's air strikes]**, 13 Iraqi soldiers abandoned their posts and surrendered to Kurdish fighters."*

Resolving the event chain ***[hit]*** - ***[struck]*** - ***[today's air strikes]*** will provide us details about the "*air strike*" event mentioned in different sentences and

[1] The definition according to (Asher, 1993) includes both gerunds (e.g. destruction) and inflectional verbs (e.g. destroying). In our study, we only focus on the inflectional verbs, as the gerunds are well studied in the conventional anaphora resolution systems.

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 872–881,
MIT, Massachusetts, USA, 9-11 October 2010.

also provide us a clue for a temporal/causal relation between these two events, *"Two F Tomcats struck the targets"* and *"13 Iraqi soldiers abandoned their posts and surrendered to Kurdish fighters"*.

In (Asher, 1993) chapter 6, a method to resolve the references to abstract entities using discourse representation theory is discussed. However, no computational system was proposed for event anaphora resolution. (Byron, 2002; Müller, 2007; Chen *et al,* 2010) attempted event pronoun resolution. (Byron, 2002) proposed a knowledge deep approach for a much focused domain like trains spoken dialogue addressed in the paper. Their system resolved limited number of verbs with handcraft knowledge of relevant events. Clearly this approach is not suitable for general event anaphora resolution such as in news articles. Besides, there's also no performance report dedicated on event pronoun resolution, thus it's not clear how effective their approach is. (Müller, 2007) proposed a pronoun resolution system using a set of hand-crafted constraints such as "argumenthood" and "right-frontier condition" together with logistic regression model based on corpus counts. Their system targeted only three pronouns namely, *"it", "this"* and *"that"*. The event pronouns are resolved together with object pronouns. This preliminary explorative work only produced 11.94% F-score for event pronoun resolution which demonstrated the difficulties for event anaphora resolution. In our previous work (Chen *et al,* 2010), we proposed an event pronoun resolution system using various flat and structural knowledge. We managed to achieve a F-score of 57.9% on event pronouns. This paper is a significant improvement and extension from our previous one. Besides, (Pradhan, *et.al,* 2007) applied a conventional co-reference[2] resolution system to OntoNotes corpus using the same set of features for object noun phrase (NP) resolution. There is no specific performance reported on event anaphora resolution. According to our investigation elaborated in section 2.2, the event anaphors may not be correctly resolved in general, as majority of these features are inappropriate for event anaphora resolution.

In this paper, we provide the first systematic study to resolve NPs to event verbs. First, we explore various lexical, positional and syntactic features useful for the event NP resolution, which turns out quite different from conventional anaphora resolution except the sentence distance information. Syntactic structural information is further incorporated using a composite kernel. Furthermore, the candidate preference information is employed using a twin-candidate model. Our approach shows encouraging performance, 61.39% F-score on OntoNotes.

The rest of this paper is organized as follows. Section 2 introduces the framework and various features useful for event NP resolution. Section 3 presents the structural syntactic features and kernel functions to incorporate such features. Twin-candidate model is further introduced to capture the preference knowledge. Section 4 presents the experiment results with discussions. Section 5 concludes the paper.

2 The Resolution Framework

Our event NP resolution system adopts the common learning-based framework for object anaphora/co-reference resolution, as employed by (Soon *et al.*, 2001) and (Ng and Cardie, 2002a).

2.1 Training and Testing instance

In this learning framework, a training/testing instance has a form of $fv(candi_i, ana)$ where $candi_i$ is the i^{th} antecedent candidate of an anaphor ana. An instance is labeled as positive if $candi_i$ is the antecedent of ana, or negative if $candi_i$ is not. An instance is associated with a feature vector which records different properties and relations between ana and $candi_i$. These features will be discussed later in the paper.

During training, for each event NP, we will consider the preceding and succeeding verbs as antecedent candidates. The succeeding verbs are included to accommodate the cataphora phenomenon in which an antecedent occurs after its anaphor. A positive instance is formed by pairing the anaphor with its antecedent. And a set of negative instances is formed by pairing the anaphor with each of its candidates other than the antecedent, which follows the same negative instance selection strategy discussed in (Ng and Cardie, 2002a). Based on these generated training instances, we can train a

[2] Co-reference means two expressions denoting the same entity (e.g. "*admit guilty*" – "*confess*") while anaphora means the latter expression requires the earlier one's information for a correct interpretation (e.g. "*confess*" – "*the confession*"). Despite the difference in definitions, they share a similar set of features in resolution models.

binary classifier using any discriminative learning algorithm.

Testing instances are generated in the same manner except that all the preceding and succeeding verbs will be considered as candidates.

2.2 Flat Features

Table 1 gives a partial list of features used in conventional NP anaphora/co-reference resolution which focuses on objects in (Soon *et al*, 2001; Ng and Cardie, 2002b; Yang *et al*, 2003; Luo *et al*, 2004).

However, most of these features are not useful for our task except the shallow positional features. In event NP resolution, we focus on events instead of objects. Thus the features describing object characteristics such as number agreement, gender agreement and name alias will no longer function here. Secondly, our anaphor and antecedent pair consists of a verb and an NP. The difference in word syntactic categories will introduce extra difficulties using the conventional lexical features such as string matching and head matching. Furthermore, the difference in word syntactic categories will cripple the NP characteristic features for half of the pair. Grammatical features such as appositive structure are no long useful as well.

Conventional Features	*Applicable to Event Anaphora Resolution*
Positional Features	
Sentence Distance	Yes
Object Characteristics	
Number Agreement	No
Gender Agreement	No
Name alias	No
Lexical Features	
String Matching	No
Head Phrase Matching	No
Grammar Features	
Appositive Structure	No
NP Characteristics	
Definite / Indefinite NP	Partial
Demonstrative / Non-Demo NP	Partial
NP is a Proper Name	Partial

Table 1: Features for Conventional NP Resolution

Thus we have conducted a study on the effectiveness of potential important features for event NP resolution. They are elaborated in detail in the rest of this section.

- **Morphological Feature**

Morphological feature captures the inflectional and derivational relation between an anaphor and its antecedent candidate. The morphological feature helps to bridge the gap between different word syntactic categories. This feature represents how close the anaphor and a candidate are in their meanings. A candidate with an inflectional or derivational relation with the anaphor is more preferred to be the antecedent. For example, *"confess"* will be a better antecedent choice for *"confession"* comparing to other verbs. WordNet is used as our morphology knowledge source.

- **Synonym Feature**

Synonym feature is also to capture the similarity in meanings between the anaphor and its antecedent candidates. For example, *"assault"* is a preferable candidate for anaphor *"attack"* (in noun category). In the actual resolution, synonyms are generated from the derivational forms of the anaphor and candidates. This is to overcome the gap in word syntactic categories between an anaphor and its candidates. Two lists of synonyms (including synonyms of derivational forms) are generated for the anaphor and its candidate respectively. The synonym feature will be evaluated by comparing the two lists. Feature values include cases as "**B**oth are **I**n the others' synonym **L**ist (**BIL**)", "**O**ne **I**n the other's **L**ist (**OIL**)", "**L**ists are **O**verlapping (**LO**)" and "**L**ists are **M**utually **E**xclusive (**LME**)". These four values are considered as ordinal with a descending order of **BIL**>**OIL**>**LO**>**LME**. Higher order indicates a more similar word meaning between a candidate and the anaphor. WordNet is used during the synonym lists generation.

- **Fixed Pairings**

Fixed pairings are a list of commonly used referential pairs. For example, *"say - information"* and *"announce - statement"* are commonly used in an anaphoric relation. From a linguistics point of view, *"information"* is the patient role in a *"say"* action. The relation between *"say"* and *"information"* is different from the synonymy and morphology relation described previously. Fixed pairing list is automatically generated from training data by memorizing all encountered pairs of the head of NP anaphor and its verb antecedent. It is 1 if a candidate anaphor pair makes a hit in the fixed pairing list and 0 if the pair does not exist in the pairing list.

- **Named Entity Feature**

This feature indicates if a given NP is a named entity. A named entity is recognized using named ent-

ity recognizer for object entities representing person, location, organization and etc. An NP marked as person, location and organization is very unlikely to represent an event. This feature provides a strong heuristic to rule out inappropriate candidates.

- **Contextual Information Features**

This group of features measures the similarities and referential relations exist in the contexts of an anaphor and one of its candidates. These features are derived based on the following two intuitions. First, an event is not only represented by its main verb, the related information (e.g. roles of the action) can be extracted from surrounding contexts. Second, when an event is referred in a later occurrence, the related information is likely to reoccur in the contexts as well. Therefore, this group of feature is designed to capture such knowledge. There are two features in this group.

Context Words Similarity

This feature measures similarity between anaphor's contexts and its candidate's context. Stop words (such as *"in", "the"* and etc.) are removed from the contexts before calculating the similarity. The similarity is calculated based on a list of nearby 10 contextual words. The number of common words is used to represent the contextual similarity. Inflectional and derivational forms in the contextual words are considered as match cases.

Co-referential Relation(s) in Contexts

This feature is 1 if at least one object co-referential relation exists between the anaphor's contexts and its candidate's contexts. The idea is still to capture matching roles of action in the two contexts. For example,

*"**[George W. Bush]**$_1$ **{approved}**$_2$ the new military plan .... **{[The president]**$_1$ **'s decision}**$_2$ agitated various anti-war groups ...".*

By knowing that the ***[George W. Bush]***$_1$ and ***[The president]***$_1$ co-refer with each other, ***{approved}***$_2$ is a preferable candidate for ***{The president's decision}***$_2$ as they share a common attribute value "*president Bush*".

- **NP Antecedent(s) Features**

When an NP co-refers a preceding NP, the original phrase will normally be replaced with a more concise expression which is the anaphor. By using the full expression from the antecedent, we can obtain extra knowledge for the later concise expression. For the antecedent knowledge of NPs, we used the OntoNote gold standard annotations for object co-references. There are 3 features in this group.

Morphological Feature with NP's Antecedent(s)

This feature is evaluated by comparing each of an NP's antecedents with its verb candidates for an inflectional or derivational relation. It is considered as a morphological relation if one of the NP's antecedents is inflectional or derivational to the verb.

Synonym Feature with NP's Antecedent(s)

Similar to the above, the synonym list generated from an NP's co-referential expressions is used to compare with its verb candidate's synonym list. The final feature value is taken to be the highest order as described in the previous section on synonym feature.

Named Entity Feature with NP's Antecedent(s)

Similar to the NP's named entity feature described previously, this feature is used to rule out inappropriate NPs for an event anaphoric relation. Consider the object co-referential expressions *"George W. Bush"* and *"the president"*, the first one will be marked as named entity but not the latter. By using the object NP's co-reference knowledge, we can rule out the inappropriate NP *"the president"* as it refers to an object.

- **Grammatical Role**

This set of feature aims to capture the grammatical roles of the anaphor and its antecedent candidates. The details of this set of features are tabulated in Table 2 below.

NP: *M*	
Sbj_Main	1 if *M* is subject in main clause; else 0.
Sbj_Sub	1 if *M* is subject in sub-clause; else 0.
Obj_Main	1 if *M* is object in main clause; else 0.
Obj_Sub	1 if *M* is object in sub-clause; else 0.
Verb: *V*	
Main	1 if *V* in main clause; else 0.
Sub	1 if *V* in sub-clause; else 0.

Table 2: Features representing Grammatical Roles

- **Positional and NP Characteristics Features**

Positional Features:	***NP: M; Verb: V***
SentDist	# of Sentences between *M* and *V*;
PhraseDist	# of NPs between *M* and *V;*
WordDist	# of words between *M* and *V;*
NP Characteristic Features:	
NP_Def	1 if *M* is definite; else 0;
NP_Demo	1 if *M* is demonstrative; else 0;
NP_First	1 if *M* is the first NP in its sentence;

Table 3: Positional & NP Characteristic Features

A set of positional and NP characteristic features is employed in our resolution system. Positional fea-

tures extend their counterparts in conventional NP anaphora resolution by incorporating phrase distance and word distance. NP characteristic features are applied to (possible) NP Anaphora. These features are tabulated in Table 3.

3 Incorporating Structural Syntactic Information

A parse tree that covers an event NP and its antecedent could provide us much syntactic information related to the pair. The commonly used syntactic knowledge for anaphora resolution, such as the governing relations, can be directly described by the tree structure. Other syntactic knowledge that may be helpful for anaphora resolution could also be implicitly represented in the parse tree. Such syntactic knowledge can be captured using a convolution tree kernel by comparing the number of common sub-structures in two trees. The implicit syntactic structural knowledge can be further combined with other knowledge through a composite kernel.

Normally, parsing is done on the sentence level. However, in many cases an event NP and its antecedent do not occur in the same sentence. To present their syntactic properties and relations in a single tree structure, we construct a syntax tree for the entire text, by attaching the parse trees of all its sentences to a pseudo upper node. Having obtained the parse tree of a text, we shall consider how to select the appropriate portion of the tree as the structural feature for a given instance. As each instance is related to an event NP and one of its candidates, the structured feature at least should be able to cover both of these two expressions.

3.1 Structural Syntactic Feature

Generally, the more portion of the parse tree is included, the more syntactic information would be provided. But at the same time, the more noisy information that comes from parsing errors and other sources would likely be introduced as well. In our study, we examine three possible structural features that contain different portions of the parse tree:

- **Minimum Expansion Tree**

This feature records the minimal structure covering both the NP and its verb candidate in the parse tree. It only includes the nodes occurring in the shortest path connecting the NP and its candidate, via the nearest commonly commanding node. When the anaphor and its antecedent are from different sentences, we will find a path through a pseudo "TOP" node which links all the parse trees of sentences of a text.

- **Simple Expansion Tree**

Minimum-Expansion could, to some degree, describe the syntactic relationships between an anaphor and its candidate. However, the tree structure surrounding the expression is not taken into consideration. To incorporate such information, feature Simple-Expansion not only contains all the nodes in Minimum-Expansion, but also includes the first-level children of those nodes between the antecedent and anaphor pair except the punctuations. Thus, Simple-Expansion contains a concise representation of surrounding syntax structures.

- **Full Expansion Tree**

This feature focuses on the whole tree structure between the candidate and anaphor. It not only includes all the nodes in Simple-Expansion, but also the nodes (beneath the nearest commanding parent) that cover the words between one candidate and the anaphor. In multi-sentence cases, only sentences containing the anaphors/antecedents are expanded. Such a feature keeps the most information related to the anaphor and candidate pair.

Figure 1 illustrates the differences of three syntactic tree structures. In each expansion tree, the involved sub-tree is shaded in grey with bold fonts and thicker lines.

3.2 Convolution Parse Tree Kernel and Composite Kernel

To capture the structural features in parse tree, we use the convolution tree kernel which is defined in (Collins and Duffy, 2002) and (Moschitti, 2004). Given two trees, the kernel will enumerate all their sub-trees and compute the number of common sub-trees. As been proved, the convolution kernel can be efficiently computed in polynomial time on average. The above tree kernel only aims for the structured features. We also need a composite kernel to combine together the structured features and the flat features described in section 2.2. In our study we define the composite kernel as follows:

$$K_{comp}(x_1, x_2) = \frac{K_{tree}(x_1, x_2)}{|K_{tree}(x_1, x_2)|} + \frac{K_{flat}(x_1, x_2)}{|K_{flat}(x_1, x_2)|}$$

where K_{tree} is the convolution tree kernel defined for the structured features, and K_{flat} is the kernel applied to the flat features. Both kernels are divided by their respective length ($|K(x_1, x_2)| =$

$\sqrt{K(x_1, x_1) \cdot K(x_2, x_2)}$) for normalization. The new composite kernel K_{comp}, defined as the summation of normalized K_{tree} and K_{flat}.

3.3 Twin-Candidate Framework using Ranking SVM Model

In a ranking SVM kernel as in (Moschitti *et al,* 2006) for Semantic Role Labeling, two argument annotations (as argument trees) are presented to a ranking SVM model to decide which one is better. In our case, we have a slightly different setting. Instead of two argument trees, we present two syntactic trees from two candidates to the ranking SVM model. The idea is inspired by (Yang, *et.al,* 2005;2008). The intuition behind the twin-candidate model is to capture the information of how much one candidate is more preferred than another. The candidate wins most of the pair-wise comparisons will be selected as the antecedent.

The feature vector for each training instance has a form of $fv = (candi_i, candi_j)$. An instance is positive if $candi_i$ is a better choice than $candi_j$. Otherwise, it is a negative instance. For each feature vector, both tree structural features and flat features are used. Thus each feature vector has a detailed form of $fv = (t_i, t_j, v_i, v_j)$ where t_i and t_j are the parse trees of candidate i and j respectively; v_i and v_j are the flat feature vectors of candidate i and j respectively. Therefore the final SVM kernel is a joint kernel of tree kernel and flat kernel. In the training instances generation, we only generate those instances with one candidate being the antecedent. This follows the same strategy used in (Yang *et al*, 2008) for object anaphora resolution.

In the resolution process, a list of m candidates is extracted as described in section 2.1. A total of $\binom{m}{2}$ instances are generated by pairing-up the m candidates pair-wisely. We used a Round-Robin scoring scheme for antecedent selection. Suppose a SVM output for a testing instance $fv = (candi_i, candi_j)$ is positive, we will give a score 1 for $candi_i$ and -1 for $candi_j$. Similarly, if a SVM output for a testing instance $fv = (candi_i, candi_j)$ is negative, we will give a score -1 for $candi_i$ and 1 for $candi_j$. At last, the candidate with the highest final score is selected as the antecedent[3]. In order to handle a non-event NP encountered during the resolution, we empirically set a threshold to distinguish event anaphoric from non-event anaphoric.

4 Experiments and Discussions

4.1 Experimental Setup

OntoNotes Release 2.0 English corpus is used in our study. It 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 4 shows nearly 20% of the entity chains annotated in OntoNotes are event chains with at least one verb mention. This significant proportion demonstrates importance of event anaphora resolution in a text processing system.

Chain Type	*Count*	*Percentage*
Event Chain	1235	19.97%
Object Chain	4952	80.03%
Total	6187	100%

Table 4: Percentage of Event Chains in OntoNotes

Our resolution system focuses on the verb-NP links existing in the event chains. In total, we have extracted 977 verb-NP pairs from the OntoNotes corpus. To illustrate the task difficulties, the sentence distance between the anaphor and its antecedent is tabulated below in Table 5. The average sentence distance is 2.97.

SenDist	**0**	**1**	**2**	**3**	**4**	**5**
Count	102	436	143	66	38	69
SenDist	**6**	**7**	**8**	**9**	**10**	**>10**
Count	14	15	11	13	7	63

Table 5: Distribution of Sentence Distance

Due to the long separation distance, the number of candidates during resolution is very large. There are on average 30.49 candidates in training and 583.74 candidates in testing. There is no simple baseline for the task. The most recent verb for each NP will result an F-score almost zero as majority (78.7%) of the NPs are not anaphoric in OntoNotes**.** On the other hand, only 2.31% of verbs in OntoNotes are referred by an NP.

To conduct event NP resolution, an input raw text was preprocessed automatically by a pipeline of NLP components. The NP identification and the predicate-argument extraction were done based on Stanford Parser (Klein and Manning, 2003a;b) with F-score of 86.32% on Penn Treebank corpus. The named entity recognition process uses a SVM based NER trained and tested on ACE 2005 with 92.5(P) 84.3(R) 88.2(F).

[3] In a tie-breaking scenario, the candidate closer to anaphor is selected as antecedent.

4.2 Experiment Results and Discussion

In this section, we will present our experimental results with discussions. The performance measures we used are precision, recall and F-score. All the experiments are done with a 10-folds cross validation to evaluate the performances. In each experiment, the whole corpus is divided into 10 equal sized portions. In each fold, one of the portions is selected to be testing corpus while the remaining 9 are used for training. In case of statistical significance test for differences is needed, a one-tailed, paired-sample Student's t-Test is performed at 0.05 level of significance.

In the first set of experiment results, we are investigating effectiveness of each flat feature. The effectiveness of an individual feature is measured in a leave-one-out manner. That is the performance loss by removing a particular feature from the feature list. The greater performance drop after removing a feature indicates the more effective that feature is for event NP resolution. Table 6 presents the results of this set of experiments.

Feature	Precision	Recall	F-score
ALL	43.87%	42.86%	43.35%
-Morph	8.74%	5.84%	6.99%
-Synonym	7.24%	4.63%	5.64%
-Fixed_Pair	9.94%	5.43%	7.01%
-NE	12.40%	7.04%	8.95%
-Cont_Sim	10.35%	4.63%	6.37%
-Cont_Coref	8.17%	4.43%	5.72%
-Ante_Morph	11.00%	6.64%	8.26%
-Ante_Syn	11.95%	7.04%	8.84%
-Ante_NE	10.36%	7.24%	8.51%
-Gram_Role	11.76%	6.64%	8.45%
-Position	47.47%	32.11%	38.31%
-NP_Def	11.85%	6.04%	7.99%
-NP_Demo	7.85%	4.23%	5.48%
-NP_First	12.98%	7.04%	9.12%

Table 6: Effectiveness of Individual Flat Feature

In Table 6, the first line is performance using all the flat features. Each line below is the performance after removing the feature in that line from the resolution system. The observations in Table 6 suggest that all the features we have discussed in section 2.2 contribute a significant part in the resolution system. For most of the features (except position), the overall system is almost not functioning for the identification of the antecedent. The performance drops for most of features are over 30% in F-score. The conclusion we can draw from these observations is that the flat features are co-dependent to perform the event NP resolution task. Each feature's individual contribution is hard to be separated from the overall performance. All of them are essential parts in the resolution system. Positional feature will incur a 5.04% drop in F-score. Although it is comparatively smaller than the performance drop of the other features, it is still a significant part in the overall performance. Especially, after removing positional features, the recall decreases by 10.75%. Therefore, in the later on experiments, all the flat features are used for event NP resolution.

In the next set of experiments, we are aiming to investigate the individual effectiveness of each knowledge source. Table 7 reports the performance of these experiments.

	Precision	Recall	F-score
Flat	43.87%	42.86%	43.35%
Min-Exp	33.35%	19.95%	24.82%
Simple-Exp	22.22%	8.45%	12.24%
Full-Exp	33.33%	5.63%	9.63%

Table 7: Contribution from Single Knowledge Source

From Table 7, the flat feature set yields a baseline system with 43.35% F-score. By using each tree structure alone, we can only achieve a performance of 24.82% F-score using the minimum-expansion tree. These results indicate that the syntactic structural information alone cannot resolve event anaphoric noun phrases.

As we explained in section 3.2, a composite kernel can be used to combine flat features with syntactic structure feature. The third set of experiments is conducted to verify the performances of various tree structures combined with the flat features. The performances are reported in Table 8.

	Precision	Recall	F-score
Flat	43.87%	42.86%	43.35%
Flat+Min-Exp	65.78%	53.60%	59.01%
Flat+Sim-Exp	62.85%	49.64%	55.43%
Flat+Full-Exp	64.56%	50.77%	56.77%

Table 8: Comparison of Different Combinations

As Table 8 presents, all the three types of structural information improve the overall performance by over 10% in F-score. Obviously, syntactic structural information is very useful for event NP resolution when combined with flat features. Minimum expansion tree performs better than the other two structures. The performance difference in simple expansion and full expansion are statistically insignificant. This result shows that contextual structural information is considered as harmful rather than helpful in an event NP resolution. The

minimum structural information covering the anaphor and antecedent is the most helpful as it introduces least noises. This finding is different from the conclusion in conventional pronoun resolution as reported in (Yang *et al,* 2006;) where simple expansion tree performs best. We consider this difference is caused by the distance of separation from the anaphor to its antecedent.

In the last set of experiment, we will present the performance from the twin-candidates based approach in Table 9. The first line is the best performance from single candidate model. The second line is performance using the twin-candidate approach.

Flat+Min-Exp	**Precision**	**Recall**	**F-score**
Single Candidate	65.78%	53.60%	59.01%
Twin-Candidates	66.41%	57.93%	61.36%

Table 9: Single Candidate V.S. Twin Candidates

Comparing to the single candidate model, the recall is significantly improved with similar level of precision. The difference in results is statistically significant. It reinforced our intuition that preference knowledge between two candidates is a contributive information source in event NP resolution.

Last but not least, we have conducted a study on the performance impact of various training data size. 10% from the whole corpus is kept as testing data. The remaining training data is further split into 10 equal sized portions. The experiments are conducted by gradually increase the number of training portion by one at each run. Due to time constraints, the learning curve experiments are conducted using single candidate model as the twin-candidate model is very time consuming. But it should be representative for twin-candidate scenario as well. The resulted learning curve is plotted in Figure 2.

As presented in Figure 2, the training data size increases, the performance curve quickly converged around 0.55 for F-score when the percentage of training data reaches near 50% of training data. After that, F-score still increases as more training data are used. But the rate of increment slows down. It reaches a relatively flat shape near 80% of training data are used. The last 3 experiment results (corresponding to last 3 points plotted in learning curve) are tested for statistical differences. The results show that they are statistically indifferent which suggests a saturated performance is achieved when the training data proportion reaches 80%.

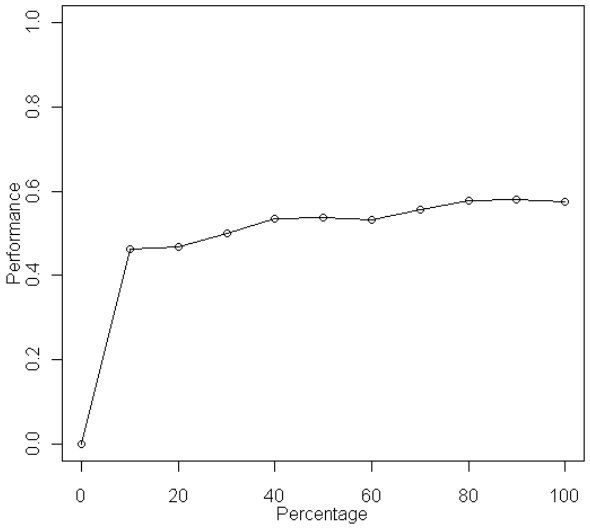


Figure 2: Learning Curve on Various Training Size

This learning curve shows that a satisfactory performance can be achieved with reasonable sized training corpus such as OntoNotes.

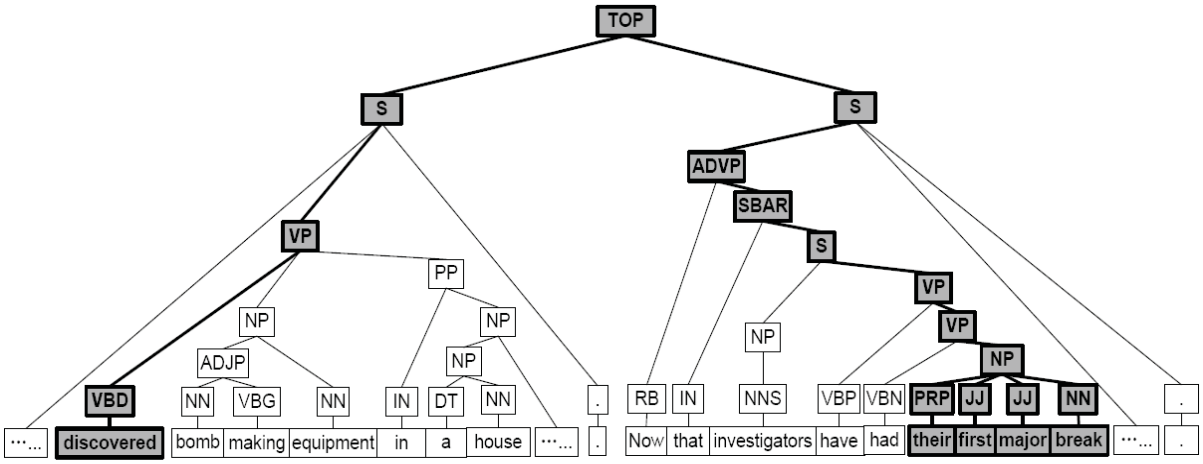
5 Conclusion and Future Work

The purpose of this paper is to conduct a systematic study of the event NP resolution, which is not available in the literature. We propose a resolution system utilizing a set of flat positional, lexical, syntactic features and structural syntactic features. The state-of-arts convolution tree kernel is used to extract indicative structural syntactic knowledge. A twin-candidates preference based approach is incorporated to reinforce the resolution system with candidate preference knowledge. Last but not least, we also proposed a study to examine how various training corpus sizes will affect the resolution performance.

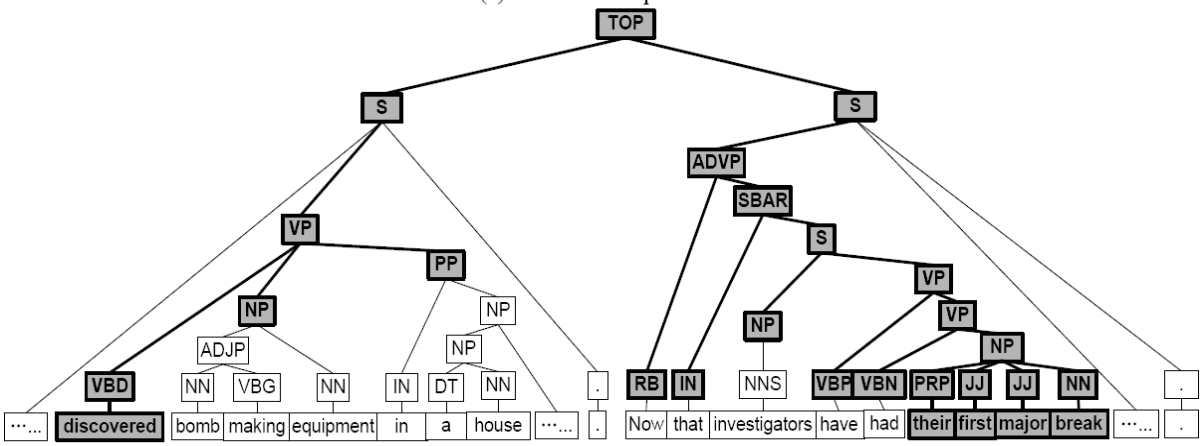
In the future, we would like to further explore more semantic information can be employed into the system such as semantic role labels and verb frames.

Acknowledgement

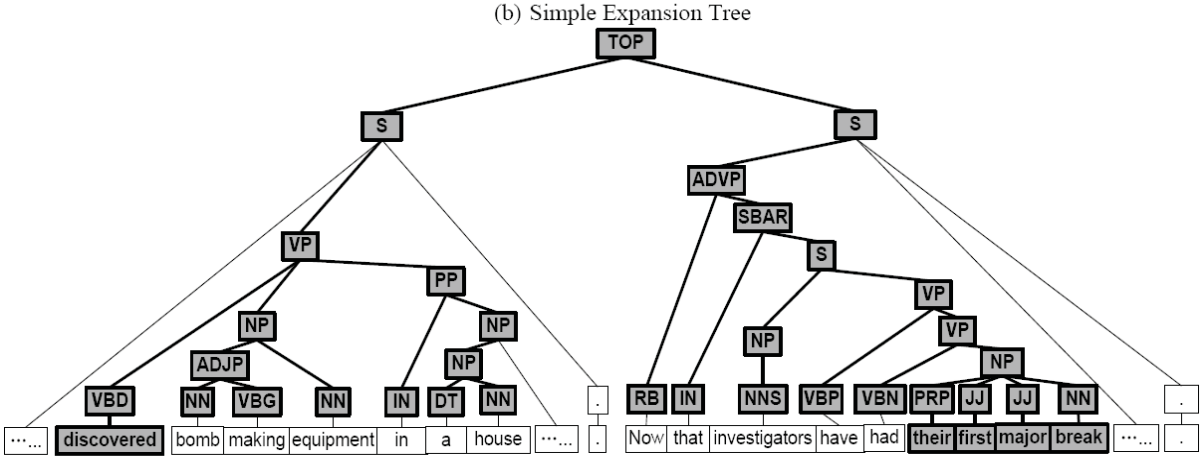
We would like to thank Professor Massimo Poesio from University of Trento for the initial discussion of this work.



(a) Minimum-Expansion Tree



(b) Simple Expansion Tree



(c) Full-Expansion Tree

Figure 1: Comparison Different Syntactic Structures using part of example from section 1.
"… ***[discovered]*** $_2$ bomb equipment in a house… .
Now that investigators have had ***[their first major break]*** $_3$… ."

References

N. Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publisher.

V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

T. Joachims. 1999. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

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

D. Byron. 2002. Resolving Pronominal Reference to Abstract Entities, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, 2002, USA

M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*. July 2002. , USA

V. Ng and C. Cardie. 2002a. Combining sample selection and error-driven pruning for machine learning of coreference rules. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 55–62, Philadelphia.

V. Ng and C. Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 104–111, Philadelphia.

D. Klein and C. Manning. 2003a. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002),* Cambridge, MA: MIT Press, pp. 3-10.

D. Klein and C. Manning. 2003b. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, pp. 423-430.

X. Yang, G. Zhou, J. Su, and C. Tan. 2003. Coreference Resolution Using Competition Learning Approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, pp 176–183.

X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics(ACL'04)*,2004

A. Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pg 335–342.

X. Yang, J. Su and C.Tan. 2005. A Twin-Candidates Model for Coreference Resolution with Non-Anaphoric Identification Capability. In *Proceedings of IJCNLP-2005*. Pp. 719-730, 2005

A. Moschitti, Making tree kernels practical for natural language learning. In *Proceedings EACL 2006*, Trento, Italy, 2006.

X. Yang, J. Su and C.Tan. 2006. Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. *In Proceedings of ACL 2006*. July 2006. Sydney, Australia.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2006. OntoNotes: The 90\% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL*, 2006

S. Pradhan, L. Ramshaw, R. Weischedel, J. MacBride, and L. Micciulla. 2007. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC),* Irvine, CA, Sep. 17-19, 2007.

C. Müller. 2007. Resolving it, this, and that in unrestricted multi-party dialog. *In Proceedings of ACL-07 conference*, pages 816–823.

X. Yang, J. Su and C.Tan. 2008. A Twin-Candidates Model for Learning-Based Coreference Resolution. In *Computational Linguistics*, 34(3):327-356.

G. Miller. 2009. "WordNet–About Us." *WordNet.* Princeton University, 2009 http://wordnet.princeton.edu

B. Chen, J. Su and C. Tan, 2010. A Twin-Candidate Based Approach for Event Pronoun Resolution using Composite Kernel. In *Proceedings of the 23rd International Conference on Computational Linguistics (CoLing'10),* 2010.pg188-196, Beijing, China,