A Twin-Candidate Based Approach for Event Pronoun Resolution using Composite Kernel

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

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

[2]Institute for Inforcomm Research, A-STAR
sujian@i2r.a-star.edu.sg

Abstract

Event Anaphora Resolution is an important task for cascaded event template extraction and other NLP study. In this paper, we provide a first systematic study of resolving pronouns to their event verb antecedents for general purpose. First, we explore various positional, lexical and syntactic features useful for the event pronoun resolution. We further explore tree kernel to model structural information embedded in syntactic parses. A composite kernel is then used to combine the above diverse information. In addition, we employed a twin-candidate based preferences learning model to capture the pair wise candidates' preference knowledge. Besides we also look into the incorporation of the negative training instances with anaphoric pronouns whose antecedents are not verbs. Although these negative training instances are not used in previous study on anaphora resolution, our study shows that they are very useful for the final resolution through random sampling strategy. Our experiments demonstrate that it's meaningful to keep certain training data as development data to help SVM select a more accurate hyper plane which provides significant improvement over the default setting with all training data.

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 mainly aims at object anaphora in which both the anaphor and its antecedent are mentions of the same real world objects

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

This was an all-white, all-Christian community that all the sudden was taken over -- not taken over, that's a very bad choice of words, but ***[invaded]$_1$*** *by, perhaps different groups.*
[It]$_2$ *began when a Hasidic Jewish family bought one of the town's two meat-packing plants 13 years ago.*

The anaphor **[*It*]$_2$** in the above example refers back to an event, "all-white and all-Christian city of Postville is diluted by different ethnic groups." Here, we take the main verb of the event, **[*invaded*]$_1$** as the representation of this event and the antecedent for pronoun **[*It*]$_2$**.

According to (Asher, 1993), antecedents of event pronoun include both gerunds (e.g. destruction) and inflectional verbs (e.g. destroying). In our study, we focus on the inflectional verb representation, as the gerund representation is studied in the conventional anaphora resolution. For the rest of this paper, "event pronouns" are pronouns whose antecedents are event verbs while "non-event anaphoric pronouns" are those with antecedents other than event verbs.

Entity anaphora resolution provides critical links for cascaded event template extraction. It also provides useful information for further inference needed in other natural language processing tasks such as discourse relation and entailment. Event anaphora (both pronouns and noun phrases) contributes a significant proportion in anaphora corpora, such as OntoNotes. 19.97% of its total number of entity chains contains event verb mentions.

In (Asher, 1993) chapter 6, a method to resolve references to abstract entities using discourse representation theory is discussed. However, no computation system was proposed for entity anaphora resolution. (Byron, 2002) proposed semantic filtering as a complement to salience calculations to resolve event pronoun targeted by us. This knowledge deep approach only

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 188–196,
Beijing, August 2010

works for much focused domain like trains spoken dialogue with handcraft knowledge of relevant events for only limited number of verbs involved. Clearly, this approach is not suitable for general event pronoun resolution say in news articles. Besides, there's also no specific performance report on event pronoun resolution, thus it's not clear how effective their approach is. (Müller, 2007) proposed 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. The event pronouns are resolved together with object pronouns. This explorative work produced an 11.94% F-score for event pronoun resolution which demonstrated the difficulty of event anaphora resolution. In (Pradhan, *et.al*, 2007), a general anaphora resolution system is applied to OntoNotes corpus. However, their set of features is designed for object anaphora resolution. There is no specific performance reported on event anaphora. We suspect the event pronouns are not correctly resolved in general as most of these features are irrelevant to event pronoun resolution.

In this paper, we provide the first systematic study on pronominal references to event antecedents. First, we explore various positional, lexical and syntactic features useful for event pronoun resolution, which turns out quite different from conventional pronoun resolution except sentence distance information. These have been used together with syntactic structural information using a composite kernel. Furthermore, we also consider candidates' preferences information using twin-candidate model.

Besides we further look into the incorporation of negative instances from non-event anaphoric pronoun, although these instances are not used in previous study on co-reference or anaphora resolution as they make training instances extremely unbalanced. Our study shows that they can be very useful for the final resolution after random sampling strategy.

We further demonstrate that it's meaningful to keep certain training data as development data to help SVM select a more accurate hyper-plane which provide significant improvement over the default setting with all training data.

The rest of this paper is organized as follows. Section 2 introduces the framework for event pronoun resolution, the considerations on training instance, the various features useful for event pronoun resolution and SVM classifier with adjustment of hyper-plane. Twin-candidate model is further introduced to capture the preferences among candidates. Section 3 presents in details the structural syntactic feature and the kernel functions to incorporate such a feature in the resolution. Section 4 presents the experiment results and some discussion. Section 5 concludes the paper.

2 The Resolution Framework

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

2.1 Training and Testing instance

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

During training, for each event pronoun, we consider the preceding verbs in its current and previous two sentences as its antecedent candidates. A positive instance is formed by pairing an anaphor with its correct antecedent. And a set of negative instances is formed by pairing an anaphor with its candidates other than the correct antecedent. In addition, more negative instances are generated from non-event anaphoric pronouns. Such an instance is created by pairing up a non-event anaphoric pronoun with each of the verbs within the pronoun's sentence and previous two sentences. This set of instances from non-event anaphoric pronouns is employed to provide extra power on ruling out non-event anaphoric pronouns during resolution. This is inspired by the fact that event pronouns are only 14.7% of all the pronouns in the OntoNotes corpus. Based on these generated training instances, we can train a binary classifier using any discriminative learning algorithm.

The natural distribution of textual data is often imbalanced. Classes with fewer examples are under-represented and classifiers often perform far below satisfactory. In our study, this becomes a significant issue as positive class (event anaphoric) is the minority class in pronoun resolution task. Thus we utilize a random down sampling method to reduce majority class samples to an equivalent level with the minority class samples which is described in (Kubat and Matwin, 1997) and (Estabrooks *et al*, 2004). In (Ng and Cardie, 2002b), they proposed a negative sample selection scheme which included only negative instances found in between an anaphor and its antecedent. However, in our event pronoun resolution, we are distinguishing the event-anaphoric from non-event anaphoric which is different from (Ng and Cardie, 2002b).

2.2 Feature Space

In a conventional pronoun resolution, a set of syntactic and semantic knowledge has been reported as in (Strube and Müller, 2003; Yang *et al*, 2004;2005a;2006). These features include number agreement, gender agreement and many others. However, most of these features are not useful for our task, as our antecedents are inflectional verbs instead of noun phrases. Thus we have conducted a study on effectiveness of potential positional, lexical and syntactic features. The lexical knowledge is mainly collected from corpus statistics. The syntactic features are mainly from intuitions. These features are purposely engineered to be highly correlated with positive instances. Therefore such kind of features will contribute to a high precision classifier.

- **Sentence Distance**

This feature measures the sentence distance between an anaphor and its antecedent candidate under the assumptions that a candidate in the closer sentence to the anaphor is preferred to be the antecedent.

- **Word Distance**

This feature measures the word distance between an anaphor and its antecedent candidate. It is mainly to distinguish verbs from the same sentence.

- **Surrounding Words and POS Tags**

The intuition behind this set of features is to find potential surface words that occur most frequently with the positive instances. Since most of verbs occurred in front of pronoun, we have built a frequency table from the preceding 5 words of the verb to succeeding 5 surface words of the pronoun. After the frequency table is built, we select those words with confidence[1] > 70% as features. Similar to Surrounding Words, we have built a frequency table to select indicative surrounding POS tags which occurs most frequently with positive instances.

- **Co-occurrences of Surrounding Words**

The intuition behind this set of features is to capture potential surface patterns such as "*It caused…*" and *"It leads to"*. These patterns are associated with strong indication that pronoun "*it*" is an event pronoun. The range for the co-occurrences is from preceding 5 words to succeeding 5 words. All possible combinations of word positions are used for a co-occurrence words pattern. For example "*it leads to*" will generate a pattern as "***S1_S2_lead_to***" where ***S1*** and ***S2*** mean succeeding position 1 and 2. Similar to previous surrounding words, we will conduct corpus statistics analysis and select co-occurrence patterns with a confidence greater than 70%. Following the same process, we have examined co-occurrence patterns for surrounding POS tags.

- **Subject/Object Features**

This set of features aims to capture the relative position of the pronoun in a sentence. It denotes the preference of pronoun's position at the clause level. There are 4 features in this category as listed below.

Subject of Main Clause

This feature indicates whether a pronoun is at the subject position of a main clause.

Subject of Sub-clause

This feature indicates whether a pronoun is at the subject position of a sub-clause.

Object of Main Clause

This feature indicates whether a pronoun is at the object position of a main clause.

Object of Sub-clause

This feature indicates whether a pronoun is at the object position of a sub-clause.

- **Verb of Main/Sub Clause**

Similar to the Subject/Object features of pronoun, the following two features capture the rela-

[1] $Confidence = \frac{\#\ of\ word_i occurred\ with\ positive\ instance}{\#\ of\ word_i occurrences}$

tive position of a verb in a sentence. It encodes the preference of verb position between main verbs in main/sub clauses.

Main Verb in Main Clause

This feature indicates whether a verb is a main verb in a main clause.

Main Verb in Sub-clause

This feature indicates whether a verb is a main verb in a sub-clause.

2.3 Support Vector Machine

In theory, any discriminative learning algorithm is applicable to learn a classifier for pronoun resolution. In our study, we use Support Vector Machine (Vapnik, 1995) to allow the use of kernels to incorporate the structure feature. One advantage of SVM is that we can use tree kernel approach to capture syntactic parse tree information in a particular high-dimension space.

Suppose a training set S consists of labeled vectors $\{(x_i, y_i)\}$, where x_i is the feature vector of a training instance and y_i is its class label. The classifier learned by SVM is:

$$f(x) = sign\left(\sum_{i=1} y_i a_i x \cdot x_i + b\right)$$

where a_i is the learned parameter for a support vector x_i. An instance x is classified as positive if $f(x) \geq 0$. Otherwise, x is negative.

- **Adjust Hyper-plane with Development Data**

Previous works on pronoun resolution such as (Yang *et al*, 2006) used the default setting for hyper-plane which sets $\mathrm{f(x)} = 0$. And an instance is positive if $f(x) \geq 0$ and negative otherwise. In our study, we look into a method of adjusting the hyper-plane's position using development data to improve the classifier's performance.

Considering a default model setting for SVM as shown in Figure 2(for illustration purpose, we use a 2-D example).

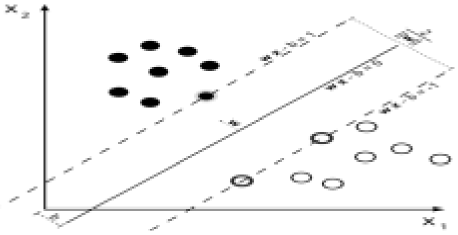

Figure 2: 2-D SVM Illustration

The objective of SVM learning process is to find a set of weight vector w which maximizes the margin (defined as $\frac{2}{\|w\|}$) with constraints defined by support vectors. The separating hyper-plane is given by $w \cdot x + b = 0$ as bold line in the center. The margin is the region between the two dotted lines (bounded by $w \cdot x + b = 1$ and $w \cdot x + b = -1$). The margin is a space without any information from training instances. The actual hyper-plane may fall in any place within the margin. It does not necessarily occur in the. However, the hyper-plane is used to separate positive and negative instances during classification process without consideration of the margin. Thus if an instance falls in the margin, SVM can only decide class label from hyper-plane which may cause misclassification in the margin.

Based on the previous discussion, we propose an adjustment of the hyper-plane using development data. For simplicity, we adjust the hyper-plane function value instead of modeling the function itself. The hyper-plane function value will be further referred as a threshold θ. The following is a modified version of a learned SVM classifier.

$$f(x, \theta) = \begin{cases} 1 & \text{if} \left(\sum_{i=1} y_i a_i x \cdot x_i + b\right) \geq \theta \\ -1 & \text{if} \left(\sum_{i=1} y_i a_i x \cdot x_i + b\right) < \theta \end{cases}$$

where θ is the threshold, a_i is the learned parameter for a feature x_i and y_i is its class label. A set of development data is used to adjust the hyper-plane function threshold θ in order to maximize the accuracy of the learned SVM classifier on development data. The adjustment of hyper-plane is defined as:

$$\theta_{best} = argmax_{\theta \in \Theta}\left(\sum_{x \in X} I(y, f(x, \theta))\right)$$

where $I(y, f)$ is an indicator function which output 1 if $f(x, \theta)$is same sign as y and 0 otherwise. Thereafter, the learned threshold θ is applied to the testing set.

3 Incorporating Structural Syntactic Information

A parse tree that covers a pronoun and its antecedent candidate could provide us much syntactic information related to the pair which is explicitly or implicitly represented in the tree. Therefore, by comparing the common sub-structures between two trees we can find out to what degree two trees contain similar syntactic information, which can be done using a convolution tree kernel. The value returned from tree kernel reflects similarity between two instances in syntax. Such syntactic similarity can be further combined with other knowledge to compute overall similarity between two instances, through a composite kernel. Normally, parsing is done at sentence level. However, in many cases a pronoun and its antecedent candidate 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 an entire text, by attaching the parse trees of all its sentences to an upper node. Having obtained the parse tree of a text, we shall consider how to select the appropriate portion of the tree as the structured feature for a given instance. As each instance is related to a pronoun and a candidate, the structured feature at least should be able to cover both of these two expressions.

3.1 Structural Syntactic Feature

Generally, the more substructure of the tree is included, the more syntactic information would be provided, but at the same time the more noisy information that comes from parsing errors would likely be introduced. In our study, we examine three possible structured features that contain different substructures of the parse tree:

- **Minimum Expansion Tree**

This feature records the minimal structure covering both pronoun and its candidate in parse tree. It only includes the nodes occurring in the shortest path connecting the pronoun and its candidate, via the nearest commonly commanding node. When the pronoun and candidate are from different sentences, we will find a path through pseudo "TOP" node which links all the parse trees. Considering the example given in section 1,

This was an all-white, all-Christian community that all the sudden was taken over -- not taken over, that's a very bad choice of words, but **[*invaded*]$_1$** *by, perhaps different groups.*

[*It*]$_2$ *began when a Hasidic Jewish family bought one of the town's two meat-packing plants 13 years ago.*

The minimum expansion structural feature of the instance {***invaded, it***} is annotated with bold lines and shaded nodes in figure 1.

- **Simple Expansion Tree**

Minimum-Expansion could, to some degree, describe the syntactic relationships between the candidate and pronoun. However, it is incapable of capturing the syntactic properties of the candidate or the pronoun, because 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 these nodes[2] except the punctuations. The simple-expansion structural feature of instance **{*invaded, it*}** is annotated in figure 2. In the left sentence's tree, the node "NP" for "*perhaps different groups*" is terminated to provide a clue that we have a noun phrase at the object position of the candidate verb.

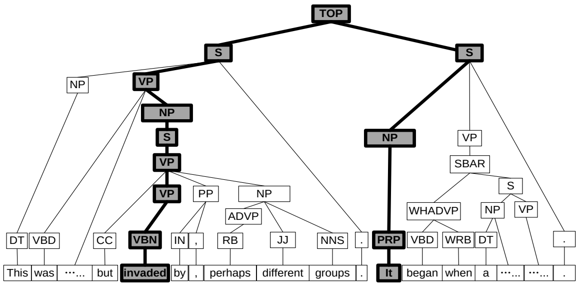

Figure 1: Minimum-Expansion Tree

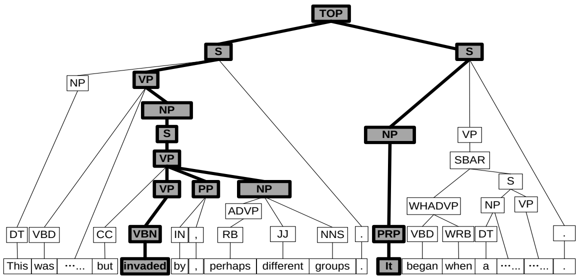

Figure 2: Simple Expansion Tree

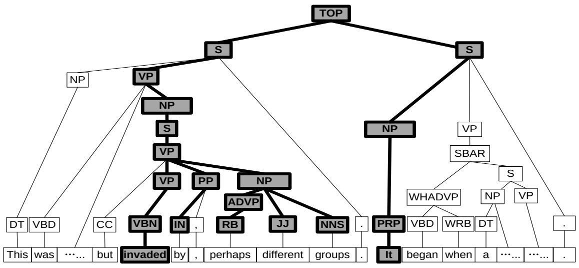

Figure 3: Full-Expansion Tree

- **Full Expansion Tree**

This feature focuses on the whole tree structure between the candidate and pronoun. It not only includes all the nodes in Simple-Expansion, but also the nodes (beneath the nearest commanding parent) that cover the words between the candi-

[2] If the pronoun and the candidate are not in the same sentence, we will not include the nodes denoting the sentences before the candidate or after the pronoun.

date and the pronoun[3]. Such a feature keeps the most information related to the pronoun and candidate pair. Figure 3 shows the structure for feature full-expansion for instance {***invaded, it***}. As illustrated, the "NP" node for **"***perhaps different groups***"** is further expanded to the POS level. All its child nodes are included in the full-expansion tree except the surface words.

3.2 Convolution Parse Tree Kernel and Composite Kernel

To calculate the similarity between two structured features, we use the convolution tree kernel that is defined by Collins and Duffy (2002) and Moschitti (2004). Given two trees, the kernel will enumerate all their sub-trees and use the number of common sub-trees as the measure of similarity between two trees. The above tree kernel only aims for the structured feature. We also need a composite kernel to combine the structured feature and the flat features from 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 feature, and K_{flat} is the kernel applied on the flat features. Both kernels are divided by their respective length[4] for normalization. The new composite kernel K_{comp}, defined as the sum of normalized K_{tree} and K_{flat}, will return a value close to 1 only if both the structured features and the flat features have high similarity under their respective kernels.

3.3 Twin-Candidate Framework using Ranking SVM Model

In a ranking SVM kernel as described in (Moschitti *et al, 2006*) for Semantic Role Labeling, two argument annotations (as argument trees) are presented to the ranking SVM model to decide which one is better. In our case, we present two syntactic trees from two candidates to the ranking SVM model. The idea is inspired by (Yang, *et.al,* 2005b;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 is selected as antecedent.

The feature vector for each training instance has a form of $fv = (candi_i, candi_j)$. An instance is positive if $cand_i$ is a better antecedent 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 form of $fv = (t_i, t_j, v_i, v_j)$ where t_i and t_j are trees of candidate i and j respectively, v_i and v_j are flat feature vectors of candidate i and j respectively.

In the training instances generation, we only generate those instances with one candidate is the correct 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 from a three sentences window. 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 an instance $fv = (candi_i, candi_j)$ is 1, we will give a score 1 for $candi_i$ and -1 for $candi_j$ and vice versa. At last, the candidate with the highest score is selected as antecedent. In order to handle a non-event anaphoric pronoun, we have set a threshold to distinguish event anaphoric from non-event anaphoric. A pronoun is considered as event anaphoric if its score is above the threshold. In our experiments, we kept a set of development data to find out the threshold in an empirical way.

4 Experiments and Discussions

4.1 Experimental Setup

OntoNotes Release 2.0 English corpus as in (Hovy *et al*, 2006) is used in our study, which 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 distribution of various entities. We focused on the resolution of 502 event pronouns encountered in the corpus. The resolution system has to handle both the event pronoun identification and antecedent selection tasks. To illustrate the difficulty of event pronoun resolution, 14.7% of all pronoun mentions are event anaphoric and only 31.5% of

[3] We will not expand the nodes denoting the sentences other than where the pronoun and the candidate occur.

[4] The length of a kernel K is defined as $|K(x_1, x_2)| = \sqrt{K(x_1, x_1) \cdot K(x_2, x_2)}$

event pronoun can be resolved using “most recent verb” heuristics. Therefore a most-recent-verb baseline will yield an f-score 4.63%.

To conduct event pronoun resolution, an input raw text was preprocessed automatically by a pipeline of NLP components. The noun phrase 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.

<table>
<tr><td colspan="3"><b>Non-Event Anaphora:</b> <i>4952 80.03%</i></td></tr>
<tr><td rowspan="4"><b>Event Anaphora:</b> <i>1235 19.97%</i></td><td colspan="2"><b>Event NP:</b> <i>733 59.35%</i></td></tr>
<tr><td rowspan="3"><b>Event Pronoun:</b> <i>502 40.65%</i></td><td><b>It:</b> <i>29.0%</i></td></tr>
<tr><td><b>This:</b> <i>16.9%</i></td></tr>
<tr><td><b>That:</b> <i>54.1%</i></td></tr>
</table>

Table 1: The distribution of various types of 6187 anaphora in OntoNotes 2.0

For each pronoun encountered during resolution, all the inflectional verbs within the current and previous two sentences are taken as candidates. For the current sentence, we take only those verbs in front of the pronoun. On average, each event pronoun has 6.93 candidates. Non-event anaphoric pronouns will generate 7.3 negative instances on average.

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. In each fold of experiments, the whole corpus is divided into 10 equal sized portions. One of them is selected as testing corpus while the remaining 9 are used for training. In experiments with development data, 1 of the 9 training portions is kept for development purpose. In case of statistical significance test for differences is needed, a two-tailed, paired-sample Student’s t-Test is performed at 0.05 level of significance.

In the first set of experiments, we are aiming to investigate the effectiveness of each single knowledge source. Table 2 reports the performance of each individual experiment. The flat feature set yields a baseline system with 40.6% f-score. By using each tree structure along, we can only achieve a performance of 44.4% f-score using the minimum-expansion tree. Therefore, we will further investigate the different ways of combining flat and syntactic structure knowledge to improve resolution performances.

	Precision	**Recall**	**F-score**
Flat	0.406	0.406	0.406
Min-Exp	0.355	0.596	0.444
Simple-Exp	0.347	0.512	0.414
Full-Exp	0.323	0.476	0.385

Table 2: Contribution from Single Knowledge Source

The second set of experiments is conducted to verify the performances of various tree structures combined with flat features. The performances are reported in table 3. Each experiment is reported with two performances. The upper one is done with default hyper-plane setting. The lower one is done using the hyper-plane adjustment as we discussed in section 2.3.

	Precision	**Recall**	**F-score**
Min-Exp + Flat	0.433 *(0.727)*	0.512 *(0.446)*	0.469 *(0.553)*
Simple-Exp +Flat	0.423 *(0.652)*	0.534 *(0.492)*	0.472 *(0.561)*
Full-Exp + Flat	0.416 *(0.638)*	0.526 *(0.496)*	0.465 *(0.558)*

Table 3: Comparison of Different Tree Structure +Flat

As table 3 shows, minimum-expansion gives highest precision in both experiment settings. Minimum-expansion emphasizes syntactic structures linking the anaphor and antecedent. Although using only the syntactic path may lose the contextual information, but it also prune out the potential noise within the contextual structures. In contrast, the full-expansion gives the highest recall. This is probably due to the widest knowledge coverage provides by the full-expansion syntactic tree. As a trade-off, the precision of full-expansion is the lowest in the experiments. One reason for this may be due to OntoNotes corpus is from broadcasting news domain. Its texts are less-formally structured. Another type of noise is that a narrator of news may read an abnormally long sentence. It should appear as several separate sentences in a news article. However, in broadcasting news, these sentences maybe simply joined by conjunction word “and”. Thus a very nasty and noisy structure is created from it. Comparing the three knowledge source, simple-expansion achieves moderate precision and recall which results in the highest f-score. From this, we can draw a conclusion that simple-expansion achieves a balance between the indicative structural information and introduced noises.

In the next set of experiments, we will compare different setting for training instances generation. A typical setting contains no negative instances generated from non-event anaphoric pronoun. This is not an issue for object pronoun resolution as majority of pronouns in an article is anaphoric. However in our case, the event pronoun consists of only 14.7% of the total pronouns in OntoNotes. Thus we incorporate the instances from non-event pronouns to improve the precision of the classifier. However, if we include all the negative instances from non-event anaphoric pronouns, the positive instances will be overwhelmed by the negative instances. A down sampling is applied to the training instances to create a more balanced class distribution. Table 4 reports various training settings using simple-expansion tree structure.

Simple-Exp Tree	Precision	Recall	F-score
Without Non-event Negative	0.423	**0.534**	0.472
Incl. All Negative	**0.733**	0.410	0.526
Balanced Negative	0.599	0.506	0.549
Development Data	0.652	0.492	**0.561**

Table 4: Comparison of Training Setup, Simple-Exp

In table 4, the first line is experiment without any negative instances from non-event pronouns. The second line is the performance with all negative instances from non-event pronouns. Third line is performance using a balanced training set using down sampling. The last line is experiment using hyper-plane adjustment. The first line gives the highest recall measure because it has no discriminative knowledge on non-event anaphoric pronoun. The second line yields the highest precision which complies with our claim that including negative instances from non-event pronouns will improve precision of the classifier because more discriminative power is given by non-event pronoun instances. The balanced training set achieves a better f-score comparing to models with no/all negative instances. This is because balanced training set provides a better weighted positive/negative instances which implies a balanced positive/negative knowledge representation. As a result of that, we achieve a better balanced f-score. In (Ng and Cardie, 2002b), they concluded that only the negative instances in between the anaphor and antecedent are useful in the resolution. It is same as our strategy without negative instances from non-event anaphoric pronouns. However, our study showed an improvement by adding in negative instances from non-event anaphoric pronouns as showed in table 4. This is probably due to our random sampling strategy over the negative instances near to the event anaphoric instances. It empowers the system with more discriminative power. The best performance is given by the hyper-plane adaptation model. Although the number of training instances is further reduced for development data, we can have an adjustment of the hyper-plane which is more fit to dataset.

In the last set of experiments, we will present the performance from the twin-candidates based approach in table 5. The first line is the best performance from single candidate system with hyper-plane adaptation. The second line is performance using the twin-candidates approach.

Simple-Exp Tree	Precision	Recall	F-score
Single Candidate	0.652	0.492	0.561
Twin-Candidates	0.626	**0.540**	**0.579**

Table 5: Single vs. Twin Candidates, Simple-Exp

Comparing to the single candidate model, the recall is significantly improved with a small trade-off in precision. The difference in results is statistically significant using t-test at 5% level of significance. It reinforced our intuition that preferences between two candidates are contributive information sources in co-reference resolution.

5 Conclusion and Future Work

The purpose of this paper is to conduct a systematic study of the event pronoun resolution. We propose a resolution system utilizing a set of flat positional, lexical and syntactic feature and structural syntactic feature. The state-of-arts convolution tree kernel is used to extract indicative structural syntactic knowledge. A twin-candidates preference learning based approach is incorporated to reinforce the resolution system with candidates' preferences knowledge. Last but not least, we also proposed a study of the various incorporations of negative training instances, specially using random sampling to handle the imbalanced data. Development data is also used to select more accurate hyper-plane in SVM for better determination.

To further our research work, we plan to employ more semantic information into the system such as semantic role labels and verb frames.

Acknowledgment

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

References

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

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

M. Kubat and S. Matwin, 1997. Addressing the curse of imbalanced data set: One sided sampling. In *Proceedings of the Fourteenth International Conference on Machine Learning*,1997. pg179–186.

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

W. Soon, H. Ng, and D. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. In *Computational Linguistics*, Vol:27(4), pg521– 544.

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

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

V. Ng and C. Cardie. 2002a. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*. July 2002. , USA. pg104–111.

V. Ng, and C. Cardie. 2002b. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING02). (2002)*

M. Strube and C. Müller. 2003. A Machine Learning Approach to Pronoun Resolution in Spoken Dialogue. . In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, 2003

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)*, 2003. pg423-430.

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)*, 2003. pg176–183.

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)*, pg335–342.

A. Estabrooks, T. Jo, and N. Japkowicz. 2004. A multiple resampling method for learning from imbalanced data sets. In *Computational Intelligence V*ol:20(1). pg18–36.

X. Yang, J. Su, G. Zhou, and C. Tan. 2004. Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of 42th Annual Meeting of the Association for Computational Linguistics*, 2004. pg127–134.

X. Yang, J. Su and C.Tan. 2005a. Improving Pronoun Resolution Using Statistics-Based Semantic Compatibility Information. *In Proceedings of Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05).* June 2005.

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

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

X. Yang, J. Su and C.Tan. 2006. Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL'06)*. July 2006. Australia.

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

C. Müller. 2007. Resolving it, this, and that in unrestricted multi-party dialog. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*. 2007. Czech Republic. pg816–823.

X. Yang, J. Su and C.Tan. 2008. A Twin-Candidates Model for Learning-Based Coreference Resolution. In *Computational Linguistics*, Vol:34(3). pg327-356.

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)*, Sep. 2007.