Anaphora Resolution by Antecedent Identification Followed by Anaphoricity Determination

RYU IIDA, KENTARO INUI, and YUJI MATSUMOTO
Nara Institute of Science and Technology

We propose a machine learning-based approach to noun-phrase anaphora resolution that combines the advantages of previous learning-based models while overcoming their drawbacks. Our anaphora resolution process reverses the order of the steps in the classification-then-search model proposed by Ng and Cardie [2002b], inheriting all the advantages of that model. We conducted experiments on resolving noun-phrase anaphora in Japanese. The results show that with the selection-then-classification-based modifications, our proposed model outperforms earlier learning-based approaches.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Discourse*

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Anaphora resolution, anaphoricity determination, antecedent identification

1. INTRODUCTION

Anaphora resolution is an important process for various NLP applications. In contrast with rule-based approaches, such as Brennan et al. [1987], Lappin and Leass [1994], Baldwin [1995], Nakaiwa and Shirai [1996], Okumura and Tamura [1996], and Mitkov [1997], empirical, or corpus-based approaches to this problem have shown to be a cost-efficient solution achieving performance that is comparable to the best performing rule-based systems [McCarthy and Lehnert 1995; Ge et al. 1998; Soon et al. 2001; Ng and Cardie 2002a; Strube and Müller 2003; Iida et al. 2003; Yang et al. 2003].

Anaphora resolution can be divided into two subtasks: *anaphoricity determination* and *antecedent identification*. Anaphoricity determination is the task

Authors' addresses: R. Iida, Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan; email: ryu-i@is.naist.jp; K. Inui; email: inui@is.naist.jp; Y. Matsumoto; email: matsu@is.naist.jp.

of classifying whether a given noun phrase (NP) is *anaphoric* or *nonanaphoric*. Here we say an NP is anaphoric if it has any antecedent [i.e., NP(s) that are coreferent with it] in the context preceding it in the discourse, and nonanaphoric otherwise. The second task, antecedent identification, is identification of the antecedent(s) of a given anaphoric NP.

Early corpus-based work on anaphora resolution does not address anaphoricity determination; it assumes that the anaphora resolution system knows a priori all the anaphoric noun phrases. However, this problem has recently been the subject of increased attention [Bean and Riloff 1999; Ng and Cardie 2002b; Uryupina 2003; Ng 2004; Poesio et al. 2004], because determining anaphoricity is not a trivial problem, even in languages, such as English and French, where definite articles can be used as clues [Ng and Cardie 2002b], and the overall performance of anaphora resolution crucially depends on the accuracy of anaphoricity determination.

Obviously, the problem of anaphoricity determination is even more critical in the case of languages, such as Japanese, which do not have such clues as definite articles.

Previous efforts to tackle this problem have provided the following findings:

- One of the useful clues for determining the anaphoricity of a given NP can be obtained by searching for an antecedent. If an appropriate candidate for the antecedent is found in the preceding context of the discourse, the NP is likely to be anaphoric [Soon et al. 2001; Ng and Cardie 2002a].
- Anaphoricity determination can be effectively carried out by a binary classifier that learns instances of nonanaphoric NPs, as well as those of anaphoric NPs [Ng and Cardie 2002b; Ng 2004].

As we discuss in the next section, previous approaches to anaphora resolution [Ng and Cardie 2002a, 2002b; Iida et al. 2003] make use of a range of cues, but none of the previous models effectively combines the three previous approaches shown in Section 2. This leaves significant room for improvement in anaphora resolution.

In this paper, we propose a machine learning-based model that effectively combines the sources of evidence used in existing models, while overcoming their drawbacks. We show the effectiveness of our approach through experiments on Japanese anaphora resolution, comparing previous machine learning-based approaches including Ng and Cardie [2002a]'s search-based approach and Ng [2004]'s classification-then-search approach.

The rest of the paper is organized as follows. In Section 2, we review previous machine learning-based approaches to anaphora resolution. Section 3 describes how the proposed model effectively combines advantages of each previous approach. We then report the results of our experiments on Japanese noun phrases anaphora resolution in Section 4. We conclude in Section 5.

2. PREVIOUS APPROACHES

Previous learning-based methods for anaphora resolution can be classified into two approaches: the *search-based approach* and the *classification-based*

Table I. Advantages in Each Approach

	Search	Classification-Then-Search	Tournament
Use contextual clues?	√		√
Use nonanaphoric instances?		√	
Can determine anaphoricity?	√	√	
Balanced training instances?			√

```
Function Search-for-Antecedent ( Ana: candidate anaphor,
                                 C: set of candidate antecedents )

    Max_Ant := φ;   Max_Score := −∞;
    for NP_i ∈ C do
        // judge whether or not Ana is anaphoric with NP_i
        Score := classify-antecedenthood (Ana, NP_i);
        if Score > Max_Score then
            Max_Ant := NP_i;   Max_Score := Score;
        end
    end
    if Max_Score > θ_ant then
        return Max_Ant
    else
        return NULL
    end
end
```

Fig. 1. The search-based model. θ_{ant} is a global variable that indicates a global threshold parameter of antecedenthood.

approach. We discuss their advantages and disadvantages below (see Table I for summary).

2.1 Search-Based Model

The search-based approach determines the anaphoricity of a given NP indirectly as a by-product of searching the preceding context for its antecedent. If an appropriate candidate for the antecedent is found, the NP is classified as anaphoric; otherwise, nonanaphoric. Models proposed by Soon et al. [2001] and Ng and Cardie [2002a] fall into this class. In Soon et al.'s method (see Figure 1), for example, given a target NP (*Ana*) for resolution, the model processes each of its preceding NPs (i.e., candidate antecedents) in a right-to-left order, determining whether or not it is coreferent with the NP_i, until a positive answer (i.e., antecedent) comes up. If all the preceding NPs are classified negative, *Ana* is judged to be nonanaphoric. We call this approach the *search-based approach*. It has the advantage of using *broader context information* in the sense that the model determines the anaphoricity of an NP by examining whether the context preceding the NP in the discourse has a plausible candidate antecedent

Table II. Feature Set Used in Soon [2001]'s Model[a]

Feature Type	Feature	Description
Lexical	SOON_STR	C if, after discarding determiners, the string denoting NP_i matches that of NP_j; else I.
Grammatical	PRONOUN_1	Y if NP_i is a pronoun; else N.
	PRONOUN_2	Y if NP_j is a pronoun; else N.
	DEFINITE_2	Y if NP_j starts with the word "the;" else N.
	DEMONSTRATIVE_2	Y if NP_j starts with a demonstrative such as "this," "that," "these," or "those;" else N.
	NUMBER	C if the NP pair agree in number; I if they disagree; NA if number information for one or both NPs cannot be determined.
	GENDER	C if the NP pair agree in gender; I if they disagree; NA if gender information for one or both NPs cannot be determined.
	BOTH_PROPER_NOUNS	C if both NPs are proper names; NA if exactly one NP is a propose name; else I.
	APPOSITIVE	C if the NPs are in an appositive relationship; else I.
Semantic	WNCLASS	C if the NPs have the same WordNet semantic class; I if they don't; NA if the semantic class information for one or both NPs cannot be determined.
	ALIAS	C if one NP is an alias of the other; else I.
Positional	SENTNUM	Distance between the NPs in terms of the number of sentences.

[a]The feature set contains relational and nonrelational features. Nonrelational features test some property P of one of the NPs under consideration and take on a value of YES or NO depending on whether P holds. Relational features test whether some property P holds for the NP pair under consideration and indicate whether the NPs are COMPATIBLE or INCOMPATIBLE w.r.t. P; a value of NOT APPLICABLE is used when property P does not apply.

or not. Soon et al., in fact, defined the feature set including broad contextual information, such as that shown in Table II.

A flaw of this approach, on the other hand, is that models are not designed to learn nonanaphoric cases directly in the training phase. As an example, let us take a closer look at Soon et al.'s model (see Figure 2). For training, their model creates a positive instance from an anaphoric NP paired with its closest antecedent (NP_5-*ANP*) and a negative instance from each of the intervening NPs paired with the anaphor (NP_6-*ANP*, NP_7-*ANP*, and NP_8-*ANP*). Note that no training instance is derived from nonanaphoric NPs. This drawback is shared also by other search-based models, including Ng and Cardie [2002a] and Yang et al. [2003]. As we show in Section 4, this may well significantly degrade performance.

Another drawback of the approach is that it may suffer also from highly imbalanced distributions of positive and negative instances. The aforementioned method of generating training instances tends to generate much more negative instances than positive ones. For example, in the experiments described in Section 4, the ratio of the positive instances to the negative instances is 1 to 22. The model requires proper selection of training instances [Ng and Cardie 2002c]. However, it is not a trivial problem.

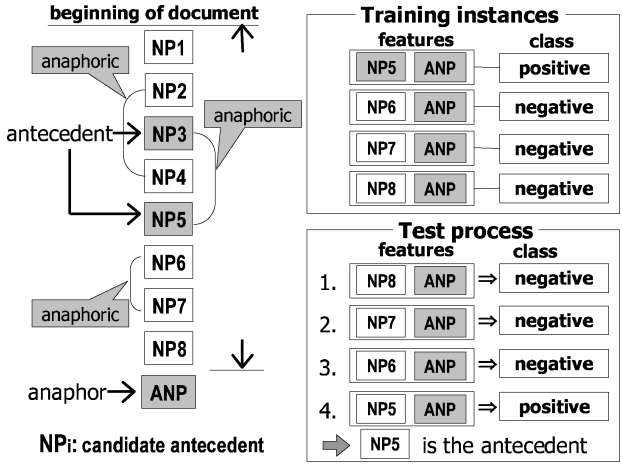

Fig. 2. The search-based model proposed by Soon et al. and Ng and Cardie. The figure illustrates how model training and anaphora resolution are carried out, assuming that there are eight noun phrases, NP_1 through NP_8, which precede a noun phrase *ANP* in question. NP_2 and NP_4, NP_3 and NP_5, and NP_6 and NP_7 are coreferent respectively, and NP_5 (and its coreferent NP_3) is the antecedent of *ANP*. Under this situation, the model detects the antecedent by answering a sequence of candidate-wise boolean classification questions: whether or not NP_i is *ANP*'s antecedent for each $i \in \{1, \ldots, 8\}$.

```
Function Classify-Anaphor-then-Search-for-Antecedent( Ana: candidate anaphor,
                                                      C: set of candidate antecedents )

    // judge whether or not Ana is anaphoric
    Ana_Score := classify-anaphoricity ( Ana );
    if Ana_Score > θ_ana then
        return Search-for-Antecedent ( Ana, C );
    else
        return NULL;
    end
end
```

Fig. 3. The classification-then-search model. θ_{ana} is a global variable that indicates a global threshold parameter of anaphoricity.

2.2 Classification-Then-Search Model

The second approach is to introduce the process of anaphoricity determination separate from antecedent identification [Ng and Cardie 2002b; Ng 2004]. We call this approach the *classification-based approach*. Unlike the search-based approach, it has an advantage in that it uses labeled instances derived from nonanaphoric NPs as well as those from anaphoric NPs to induce an anaphoricity classifier. For example, Ng [2004] proposed the following model (see Figure 3):

Table III. Partial Feature List Relevant to the Larger Context Information Used in Ng and Cardie [2002b]'s Model[a]

Feature Type	Feature	Description
Lexical	STR_MATCH	Y if there exists an NP NP_i preceding NP_j such that, after discarding determiners, NP_i and NP_j are the same string; else N.
	HEAD_MATCH	Y if there exists an NP NP_i preceding NP_j such that NP_i and NP_j have the same head; else N.
Semantic	ALIAS	Y i there exists an NP NP_i preceding NP_j such that NP_i and NP_j are aliases; else N.
	SUBCLASS	Y if there exists an NP NP_i preceding NP_j such that NP_i and NP_j have an ancestor-descendant relationship in WordNet; else N.

[a] NP_i and NP_j indicate a candidate anaphor and a candidate antecedent, respectively.

1. First it carries out anaphoricity determination using a classification-based model to filter out a target NP (*Ana*) whose anaphoricity score *Ana_Score* is below threshold θ_{ana};
2. It then searches for the antecedent for the remaining *Ana*;
3. Finally, it outputs the best-scored candidate antecedent *Max_Ant* if its score *Ant_Score* is above threshold θ_{ant}, or classifies the *Ana* as nonanaphoric otherwise.

Here we term this model the *classification-then-search model* because the model first determines the anaphoricity of a given candidate anaphor and then searches for the antecedent for the candidate anaphor.

The classification-then-search model cautiously filters out nonanaphoric NPs according to the threshold parameter θ_{ana} at the first step. Second, the model also determines the anaphoricity of the remaining candidate anaphor according to the threshold parameter θ_{ant} as well as identifies an antecedent. This two-step anaphoricity determination model is designed because the anaphoricity determination component is not powerful enough to entirely free the antecedent identification component from the charge of anaphoricity determination. As Ng [2004] reports, optimizing the two threshold parameters could improve the performance for the overall task of anaphora.

As reported in Ng and Cardie [2002b] and also in Section 4 of this paper, this model significantly outperforms the search-based model. However, it still has several drawbacks, and there is room for improvement.

First, Ng and Cardie [2002b] reports that the performance of the anaphoricity determination component is so low that applying it would not improve the performance of the overall task unless it incorporated features that effectively capture contextual information (see Table III). This indicates that it is crucially important in anaphoricity determination to know whether or not the preceding context of the discourse contains NPs that are likely to be the antecedent of a current target NP. While such features as in Table III appear to be useful clues for this reason, they appear to be rather *ad hoc* and only provide an extremely rough summary of the context.

Second, in the classification-then-search model, not only the anaphoricity classifier but also the antecedent identification component takes charge of

```
Function Select-Antecedent-by-Tournament ( Ana: candidate anaphor,
                                           C: set of candidate antecedents )

    SC := sort_by_reverce_order_of_ appearance C;
    Max_Ant := SC_1; // the right-most candidate in SC
    SC := SC \ SC_1;
    for i = 2, ..., n do
        // select which candidate is anaphoric with Ana
        Score := compare_antecedenthood ( Ana, SC_i, Max_Ant );
        if Score > 0 then
            Max_Ant := SC_i;
        end
    end
    return Max_Ant;
end
```

Fig. 4. The tournament model.

anaphoricity determination. This rather unclear way of division of labor constrains the range of algorithms that can be used for antecedent identification. The model cannot employ as, for example, the tournament model, which we review below.

Third, as long as it employs such an algorithm as Ng and Cardie [2002a] for the antecedent identification subtask, the model inherits the drawbacks of the algorithm; in particular, it is important to note the problem of imbalanced distribution of positive and negative training instances.

2.3 Tournament Model

For the task of antecedent identification alone, it is worth referring to a model called the tournament model proposed by Iida et al. [2003] (Figure 4). The model conducts a tournament consisting of a series of matches in which candidate antecedents compete with each other for a given anaphor. In the tournament, it processes the candidate antecedents in the right-to-left order. In the first round, the model consults a trained classifier to judge which of the right-most two candidates is more likely to be the antecedent for the anaphor. The winner then plays a match with the third right-most candidate. Likewise, each of the following matches is arranged, in turn, between the current winner and a right-most new challenger until the left-most candidate antecedent. The model selects the winner of tournament.

This model has several advantages over such previous antecedent identification models as reviewed in Section 2.1. First, it can incorporate the learning of some of centering factors, such as the expected center order, proposed in Centering Theory Grosz et al. [1995]. Second, unlike the previous models, the task of the classifier is to determine which of a pair of candidates is more likely to be the antecedent. This way of task decomposition inherently avoids the problem of

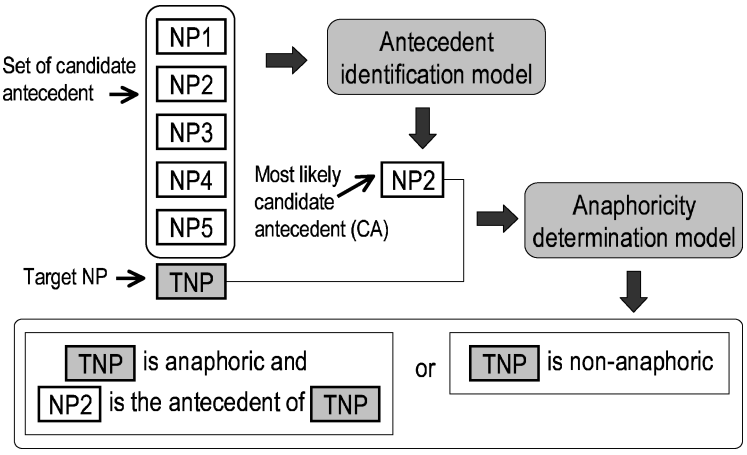

Fig. 5. Process of NP-anaphora resolution.

imbalanced distributions of positive and negative instances which such a model as Soon et al. (2001) and Ng and Cardie (2002a, 2002b) would suffer from. Due to these advantages, Iida et al. [2003] report that the tournament model outperforms the Ng and Cardie [2002a]'s model in Japanese zero-anaphora resolution.

Despite these advantages, however, the tournament model has a strict limitation; namely, it is not capable of anaphoricity determination because it always selects a candidate antecedent for a given NP whether the NP is anaphoric or not.

3. SELECTION-THEN-CLASSIFICATION APPROACH

This section discusses how to design an anaphora resolution model that inherits all the advantages of the previous models reviewed in the last section.

We explore an alternative way of incorporating contextual clues into anaphoricity determination. One way that has not yet been examined before is to implement an anaphora resolution process that reverses the steps of the classification-then-search model. Assuming that we have an antecedent identification model and an anaphoricity classification model, the new model processes each target–noun phrase (*TNP*) in a given text in two steps (see Figure 5):

1. Select the *most likely candidate antecedent CA* (NP_2 in Figure 5) for *TNP* using an antecedent identification model.
2. Classify *TNP* paired with *CA* as either *anaphoric* or *non-anaphoric* using an anaphoricity classification model. If pair *CA-TNP* is classified as *anaphoric*, *CA* is identified as the antecedent of *TNP*; otherwise, *TNP* is judged *nonanaphoric*.

To contrast the classification-then-search model, we call this model the *selection-then-classification model*.

To implement this new model, we extend a anaphoricity determination component designed in the classification-based approach so that the model determines whether a given NP paired with its most likely candidate antecedent is

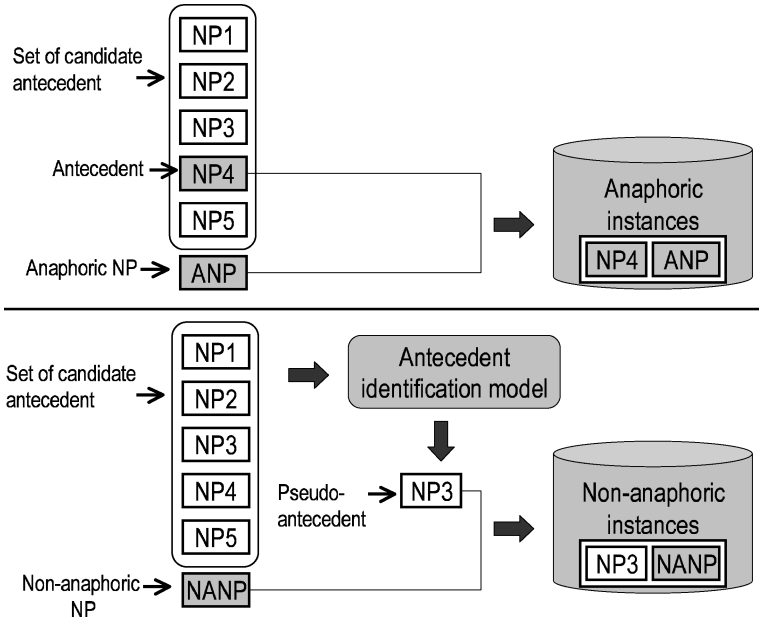

Fig. 6. Training data collection for the anaphoricity determination model.

anaphoric or not. For training the classifier, we create positive (anaphoric) and negative (nonanaphoric) training sets in the following way:

1. For each NP appearing in the training corpus, we add the pair of the NP and its corresponding antecedent to the positive (anaphoric) training set if the NP is anaphoric. This is illustrated in the upper part of Figure 6, where an anaphoric noun-phrase *ANP* paired with its antecedent NP_4 is added to the set of anaphoric instances.
2. If the NP is nonanaphoric, we first use the antecedent identification model that we employ in the antecedent identification step to select the most likely candidate antecedent for the NP, which we call the *pseudo-antecedent* of the NP. We then add the pair of the NP and its pseudo-antecedent to the negative training set. In the case of Figure 6 (the lower part of the figure), where we have a nonanaphoric noun phrase (*NANP*), we first select its most likely candidate antecedent NP_3 from candidate antecedents NP_1 through NP_5 and then add the pair NP_3-*NANP* to the nonanaphoric training set.

By providing anaphoric and nonanaphoric training sets, we can use a wide range of classifier induction algorithms.

The new model might not look considerably different from such previous models as the classification-then-search model. However, the model, in fact, effectively combines the advantages of all the previous models we reviewed in Section 2.

First, the new model inherits the advantage of the search-based model. It determines the anaphoricity of a given NP, taking into account the information of its most likely candidate antecedent. The candidate, antecedent selected

```
Function Select-Antecedent-then-Classify-Anaphor ( Ana: candidate anaphor,
                                                   C: set of candidate antecedents )

    Max_Ant := Select-Antecedent-by-Tournament ( Ana, C );
    // judge whether or not Ana is anaphoric with Max_Ant
    Score := classifier-anaphoricity ( Ana, Max_Ant );
    if ( Score > θana ) then
        return Max_Ant;
    else
        return NULL;
    end
end
```

Fig. 7. The selection-then-classification model. θ_{ana} is a global variable that indicates a global threshold parameter of anaphoricity.

in the first step can be expected to provide contextual information useful for anaphoricity determination; if the best candidate does not appear to be the real antecedent of the target NP, it is unlikely that the target NP has any antecedent in the discourse. In this respect, the proposed model makes better use of contextual clues than the classification-then-search model, which accesses to contextual information only through ad hoc string-based features.

Second, the proposed model uses nonanaphoric instances together with anaphoric instances to induce an anaphoricity classifier, which is an important advantage inherited from the classification-then-search model.

Third, in the proposed model, the division of labor between the two components is clearer than that in the selection-then-classification model. The antecedent identification component always selects a candidate antecedent for a given NP (i.e. candidate anaphor) whether the NP is anaphoric or not. This way of task decomposition allows us to employ the tournament model in antecedent identification (see Figure 7). Recall that in the classification-then-search model, the anaphoricity determination component is not reliable enough to entirely free the antecedent identification component from the charge of anaphoricity determination. This deficiency prohibits the model from incorporating the tournament model. As we report in Section 4.4, this gives a significant advantage to the new model.

4. EXPERIMENTS ON NP-ANAPHORA RESOLUTION

We conducted an empirical evaluation of our method by applying it to Japanese newspaper articles. In the experiments, we compared three models: the search-based model, the classification-then-search model, and the selection-then-classification model.

4.1 Models

For the search-based model, we created a model designed to simulate the model described in Ng and Cardie [2002a]. Pseudocode describing the model is given

in Figure 1. We employed Support Vector Machines [Vapnik 1998] for learning and used the distance between an input feature vector and the hyperplane as the score for classification.

For the classification-then-search model, we created a model based on the pseudocode given in Figure 3. In these experiments, instead of preparing the development data for the estimation of two thresholds, we evaluated the performance by fine-tuning these thresholds by hand. In addition to the original classification-then-search model, we also implemented the model using the tournament model for the antecedent identification model instead of the search-based model. Thus, we can investigate whether or not the tournament model improves the classification-then-search model.

Regarding the selection-then-classification model, we implemented the model based on the process in Figure 7.

In addition to the original selection-then-classification model, we also implemented a model using the search-based model for the antecedent identification model instead of the tournament model. Thus, we can evaluate the effectiveness of the tournament model itself by comparing the two selection-then-classification models.

Like the search-based model, the classification-then-search model and the selection-then-classification model also used SVMs for both antecedent identification and anaphoricity classification.

4.2 Training and Test Instances

We created a coreference-tagged corpus consisting of 90 newspaper articles (1104 sentences). The corpus contained 884 anaphoric NPs and 6591 non-anaphoric NPs (7475 NPs in total), each anaphoric NP being annotated with information indicating its antecedent. For each experiment, we conducted ten-fold cross-validation over 7475 noun phrases so that the set of the noun phrases from a single text was not divided into the training and test sets.

4.3 Feature Sets

We used the following five types of features:

- *ANA*: Features designed to capture the lexical, syntactic, semantic, and positional information of a target noun phrase (i.e. a candidate anaphor);
- *ANT*: Features designed to capture the lexical, syntactic, semantic, and positional information of a candidate antecedent;
- *ANA-ANT*: Features designed to capture the relation between the candidate antecedent and the target NP (e.g. the distance, semantic compatibility between the two);
- *ANT-ANT*: Features designed to capture the relation between two candidate antecedents (e.g., the distance between the two);
- *ANT_SET*: Features designed to capture the relation between the set of the candidate antecedents in the preceding context and the target NP (e.g., the binary feature that a target NP and an candidate antecedent in the preceding context contain the same string).

Table IV. Features Used in Each Model[a]

			SCM	
	SM	CSM	Antecedent Identification	Anaphoricity Determination
ANA	√	√	√	√
ANT	√		√	√
ANA-ANT	√		√	√
ANT_SET		√		
ANT-ANT			√	

[a] SM, the search-based model, CSM, the classification-then-search model, and SCM, the selection-then-classification model.

The features of the types *ANA*, *ANT*, and *ANA-ANT* cover the feature set that Ng and Cardie [2002a] used in their search-based model. On the other hand, the *ANT-ANT* type of features were those that cannot be used in the search-based model, but only in the tournament model, because the search-based model refers only to a single candidate antecedent at the time of classification. The *ANT_SET* type of features is based on the feature set in Ng and Cardie's work [2002b]. Table IV summarizes which types of features were used for each model. Table V and Table VI present the details of the feature set.

In the experiment, all the features were automatically computed with the help of publicly available NLP tools, the Japanese morphological analyzer *ChaSen* [Matsumoto et al. 2000] and the Japanese dependency structure analyzer *CaboCha* [Kudo and Matsumoto 2002], which also performed named-entity chunking.

4.4 Results

To compare the performance of the three models on the task of anaphora resolution, we plot a recall-precision curve for each model, as shown in Figure 8, by altering threshold parameter θ_{ana} [and θ_{ant} in the case of the classification-then-search model using the search-based model (CSM_SM)], where recall R and precision P are calculated by:

$$R = \frac{\text{\# of detected anaphoric relations correctly}}{\text{\# of anaphoric NPs}}$$

$$P = \frac{\text{\# of detected anaphoric relations correctly}}{\text{\# of NPs classified as anaphoric}}$$

Note that the curves of the classification-then-search model using the search-based model (CSM_SM) are plotted by altering two threshold parameters θ_{ana} and θ_{ant}. The curves indicate the upperbound of the performance of CSM_SM because in practical settings, these two parameters would have to be trained beforehand.

For the SCM algorithm, we implemented two models. One model employed SM for antecedent identification (SCM_SM) and the other employed the tournament model (SCM_TM).

The comparison between the search-based model and the classification-then-search model supports Ng and Cardie [2002b]'s claim that incorporating the anaphoricity classification process into the search-based model

Table V. Feature Set Used in Our Experiments (1/2)[a]

Feature Type	Feature	Description
Lexical	BF_COMB AT	Combination of two characters of right-most morpheme in *ANP* and NP_i.
	DOU_MATCH AT	1 if *ANP* contains the word "*dou* (i.e. same)" and the string of NP_i matches the *ANP* except for the word "*dou*"; otherwise 0.
	DOU_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that *ANP* contains the word "*dou* (i.e. same)" and the string of NP_i matches the *ANP* except for the word "*dou*"; otherwise 0.
	FIRST_PERSON_MATCH AT	1 if *ANP* and NP_i are classified as "Person" named entity class and *ANP* and NP_i share the same string; otherwise 0.
	FIRST_PERSON_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that *ANP* and NP_i are classified as "Person" named entity class and *ANP* and NP_i share the same string; otherwise 0.
	FULL_MATCH AT	1 if *ANP* and NP_i share the same string; otherwise 0.
	FULL_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that *ANP* and NP_i share the same string; otherwise 0.
	FINAL_MATCH AT	1 if *ANP* and NP_i share the same string-final morpheme; otherwise 0.
	FINAL_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that *ANP* and NP_i share the same string-final morpheme; otherwise 0.
	FIRST_MATCH AT	1 if *ANP* and NP_i share the same first morpheme; otherwise 0.
	FIRST_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that *ANP* and NP_i share the same first morpheme; otherwise 0.
	PART_MATCH AT	1 if *ANP* and NP_i share the same morpheme; otherwise 0.
	PART_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that *ANP* and NP_i share the same morpheme; otherwise 0.
	FINAL_INCUDED_MATCH AT	1 if NP_i and *ANP* share the same string-final morpheme and characters of *ANP* are included in NP_i; otherwise 0.
	FINAL_INCUDED_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that NP_i and *ANP* share the same string-final morpheme and characters of *ANP* are included in NP_i; otherwise 0.
	FIRST_INCUDED_MATCH AT	1 if NP_i and *ANP* share the same first morpheme and characters of *ANP* are included in NP_i; otherwise 0.
	FIRST_INCUDED_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that NP_i and *ANP* share the same first morpheme and characters of *ANP* are included in NP_i; otherwise 0.
	STRING_MATCH AT	1 if morphemes in ANP_i are included in NP_i in the same order; otherwise 0.
	STRING_MATCH_SET TS	1 if an NP_i preceding *ANP* exists such that morphemes in ANP_i are included in NP_i in the same order; otherwise 0.

[a] *ANP* indicates an anaphor, and $NP_{i\in\{1,2\}}$ indicates a candidate antecedent. '*'-ed features are used only in the experiments of antecedent identification. *A*, *T*, *AT*, *TS*, and *TT* indicate *ANA*, *ANT*, *ANA-ANT*, *ANT_SET*, and *ANT-ANT* features, respectively.

Table VI. Feature Set Used in Our Experiments (2/2)

Feature Type	Feature	Description
Grammatical	POS A, T	Part-of-sppech of NP_i (*ANP*) followed by IPADIC [1].
	DEFINITE A, T	1 if NP_i (*ANP*) contains the article corresponding to DEFINITE "the", such as "*sore*" or "*sono*"; otherwise 0.
	DEMONSTRATIVE A, T	1 if NP_i (*ANP*) contains the article corresponding to DEMONSTRATIVE "that" or "this", such as "*kono*", "*ano*"; otherwise 0.
	PARTICLE A, T	Particle followed by NP_i (*ANP*), such as "*wa* (topic)", "*ga* (subject)", "*o* (object)".
	DOU A, T	1 if NP_i (*ANP*) contains the word "*dou* (same)"; otherwise 0.
	DEP_PAST* A, T	1 if some predicate (past form) depends on NP_i (*ANP*); otherwise 0.
	DEP_PRED* A, T	1 if some predicate (not past form) depends on NP_i (*ANP*); otherwise 0.
Semantic	NE A, T	Named entity of NP_i (*ANP*): PERSON, ORGANIZATION, LOCATION, ARTIFACT, DATE, TIME, MONEY, PERCENT or N/A.
	EDR_HUMAN A, T	1 if NP_i (*ANP*) is included among the concept "a human being" or "attribute of a human being" in EDR dictionary; otherwise 0.
	EDR_AGENT A, T	NP_i (*ANP*) is included among the concept "agent" in EDR dictionary; otherwise 0.
	PRONOUN_TYPE A, T	Pronoun type of NP_i (*ANP*). (e.g. "*kare* (he)" → PERSON, "*koko* (here)" → LOCATION, "*sore* (this)" → OTHERS)
	SEM_PATH AT	Depth of the lowest (most specific) common node between *ANP* and *NP* in Japanese thesaurus *Bunrui Goi Hyo* [Natural Language Research Institute 1964].
Positional	SENTNUM_ANP AT	Distance between NP_i and *ANP*.
	SENTNUM_NPS* TT	Distance between NP_1 and NP_2.
	BEGINNING T, A	1 if NP_i (*ANP*) is located in the beginning of sentence; otherwise 0.
	END A, T	1 if NP_i (*ANP*) is located in the end of sentence; otherwise 0.
	DEP_NE* A, T	1 if NP_i (*ANP*) has the modifier "NAMED ENTITY+no (of)"; otherwise 0.
	DEP_NO* A, T	1 if NP_i (*ANP*) has the modifier "*no* (of)"; otherwise 0.
	DEP_ANA AT	1 if NP_i depends on *ANP*; otherwise 0.

can improve the performance if the threshold parameters are appropriately selected.

By comparing the selection-then-classification model using the search-based model (SCM_SM) with the classification-then-search model using the search-based model (CSM_SM), one can measure the effects of using the most likely antecedent while preserving the advantage of referring to the nonanaphoric information. The performance of the SCM_SM approached the upper bound of the performance of the CSM_SM. Recall that the CSM_SM algorithm requires the two interdependent threshold parameters to be trained beforehand while the proposed model need to tune only one parameter. We consider it as an

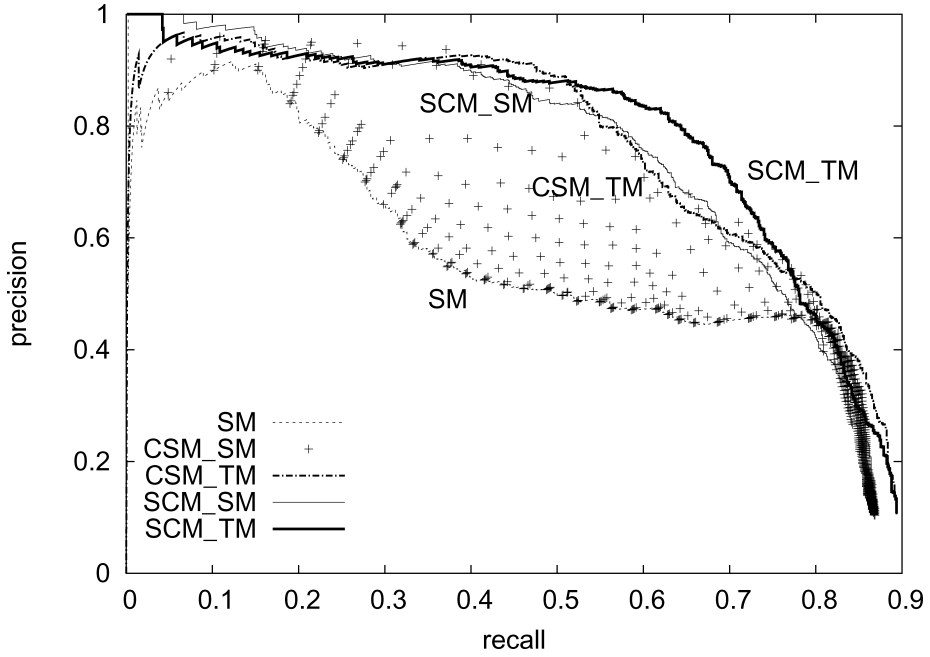

Fig. 8. Recall-precision curves in NP-anaphora resolution. SM, the search-based modelp; CSM_SM, the classification-then-search model using the search-based model; CSM_TM, the classification-then-search model using the tournament model; SCM_SM, the selection-then-classification model using the search-based model; and SCM_TM, the selection-then-classification model using the tournament model.

Table VII. Result in the Experiments of Antecedent Identification

	Search-Based Model	Tournament Model
Accuracy	86.9% (768/884)	89.4% (790/884)

important advantage of the proposed model. This advantage comes from the design of the proposed model, where the model makes use of anaphoric/nonanaphoric training instances, as well as contextual clues, given by most likely candidate antecedents simultaneously in the anaphoricity determination phase.

The results also indicate that even if the parameters for CSM_SM are optimally tuned, the proposed model significantly outperforms it when it employs the tournament model for antecedent identification (i.e., SCM_TM). The performance of the search-based model (SM) and the tournament model (TM) for antecedent identification alone is compared in Table VII. The table shows that TM outperforms SM by 2.5 points in accuracy. This difference is clearly reflected in the difference between SCM_TM and the SCM_SM. This is also an important advantage of the proposed model because previous model such as Ng [2004] cannot employ the tournament model, as we noted in Section 3.

By comparing the selection-then-classification model using the tournament model (SCM_TM) with the classification-then-search model using the

tournament model (CSM_TM), we can see whether or not the tournament model improves the CSM_TM. The results show that even if the tournament model is incorporated into the classification-then-search model, the SCM_TM still outperforms it.

4.5 Discussion

According to our error analysis, a majority of errors are caused by the difficulty of judging the semantic compatibility between a candidate anaphor and candidate antecedent. For example, the lexical resources we employed in the experiments did not contain gender information; the model did not know that "*ani* (elder brother)" was semantically incompatible with "*kanojo* (she)" and thus could not be an antecedent of it. This raises an interesting issue, namely, how to develop a lexical resource that includes a broad range of semantically compatible relations between nouns; for example, the model needs to know that *Russia* can be an antecedent of *Russian government*, but *president* is not compatible with *yesterday*. One of our future directions should aim at this issue.

There is also still room for improvement in the architecture of the proposed model. The model could make better use of the semantic information of candidate antecedents if it also referred to ancestors of coreference chains. For example, if a named-entity expression is referred to by such a word as "*dousha* (the/this company)" in the preceding context, we can enrich the coreference-chain information about by combining the relevant information from each noun phrase. This line of refinement will also lead us to explore methods to search for a globally optimal solution to a set of anaphora resolution problems for a given text, as discussed by McCallum and Wellner [2003].

5. CONCLUSION

In this paper, we reported that our selection-then-classification approach to anaphora resolution improves the performance of the previous learning-based models by combining their advantages, while overcoming their drawbacks. It does so in the following two respects: (1) our model uses nonanaphoric instances together with anaphoric instances to induce an anaphoricity classifier, retaining the advantage inherited from the classification-based approach and (2) our model determines the anaphoricity of a given NP taking the information of its most likely candidate antecedent into account. Our argument has been supported by empirical evidence obtained from our experiment on Japanese NP-anaphora resolution.

Analogous to NP-anaphora resolution, zero-anaphora resolution also deals with the issue of anaphoricity determination. Motivated by this parallelism between NPs and zero-anaphora, in future work, we want to attempt anaphoricity determination for zero pronouns using the selection-then-classification approach proposed here.

REFERENCES

BALDWIN, B. 1995. Cogniac: A discourse processing engine. Ph.D. thesis, Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia, PA.

BEAN, D. L. AND RILOFF, E. 1999. Corpus-based identification of non-anaphoric noun phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*). 373–380.

BRENNAN, S. E., FRIEDMAN, M. W., AND POLLARD, C. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics* (*ACL*). 155–162.

GE, N., HALE, J., AND CHARNIAK, E. 1998. A statistical approach to anaphora resolution. In *Proceedings of the 6th Workshop on Very Large Corpora*. 161–170.

GROSZ, B. J., JOSHI, A. K., AND WEINSTEIN, S. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics 21*, 2, 203–226.

IIDA, R., INUI, K., TAKAMURA, H., AND MATSUMOTO, Y. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics* (*EACL*) *Workshop on The Computational Treatment of Anaphora*. 23–30.

KUDO, T. AND MATSUMOTO, Y. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning 2002* (*CoNLL*). 63–69.

LAPPIN, S. AND LEASS, H. J. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, *20*, 4, 535–561.

MATSUMOTO, Y., KITAUCHI, A., YAMASHITA, T., HIRANO, Y., MATSUDA, H., TAKAOKA, K., AND ASAHARA, M. 2000. *Morphological Analysis System ChaSen version 2.2.1 Manual*.

MCCALLUM, A. AND WELLNER, B. 2003. Object consolidation by graph partitioning with a conditionally trained distance metric. In *Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*. 19–24.

MCCARTHY, J. F. AND LEHNERT, W. G. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (*IJCAI*). 1050–1055.

MITKOV, R. 1997. Factors in anaphora resolution: they are not the only things that matter. a case study based on two different approaches. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics* (*ACL*) *and the 8th Conference of the European Chapter of the Association for Computational Linguistics* (*EACL*) *Workshop on Operational Factors in Practical, Robust Anaphora Resolution*.

NAKAIWA, H. AND SHIRAI, S. 1996. Anaphora resolution of japanese zero pronouns with deictic reference. In *Proceedings of the 16th International Conference on Computational Linguistics* (*COLING*). 812–817.

NATURAL LANGUAGE RESEARCH INSTITUTE. 1964. *Burui Goi Hyo* (*in Japanese*). Shuuei Publishing.

NG, V. 2004. Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (*ACL*). 152–159.

NG, V. AND CARDIE, C. 2002a. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL*). 104–111.

NG, V. AND CARDIE, C. 2002b. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*). 730–736.

NG, V. AND CARDIE, C. 2002c. Combining sample selection and error-driven pruning for machine learning of coreference rules. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). 55–62.

OKUMURA, M. AND TAMURA, K. 1996. Zero pronoun resolution in japanese discourse based on centering theory. In *Proceedings of the 16th International Conference on Computational Linguistics* (*COLING*). 871–876.

POESIO, M., URYUPINA, O., VIEIRA, R., ALEXANDROV-KABADJOV, M., AND GOULART, R. 2004. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (*ACL*) *Workshop on Reference Resolution and its Applications*. 47–54.

SOON, W. M., NG, H. T., AND LIM, D. C. Y. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, *27*, 4, 521–544.

STRUBE, M. AND MÜLLER, C. 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*). 168–175.

URYUPINA, O. 2003. High-precision identification of discourse new and unique noun phrases. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (*ACL*) *Student Research Workshop*. 80–86.

VAPNIK, V. N. 1998. *Statistical Learning Theory*. Adaptive and Learning Systems for Signal Processing Communications, and control. Wiley, New York.

YANG, X., ZHOU, G., SU, J., AND TAN, C. L. 2003. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (*ACL*). 176–183.

Received April 2005; revised July 2005; accepted August 2005