Coreference Resolution Using Competition Learning Approach

Xiaofeng Yang[*+] **Guodong Zhou**[*] **Jian Su**[*] **Chew Lim Tan**[+]

[*]Institute for Infocomm Research,
21 Heng Mui Keng Terrace,
Singapore 119613

[+]Department of Computer Science,
National University of Singapore,
Singapore 117543

[*]{xiaofengy,zhougd,sujian}@i2r.a-star.edu.sg

[+](yangxiao,tancl)@comp.nus.edu.sg

Abstract

In this paper we propose a competition learning approach to coreference resolution. Traditionally, supervised machine learning approaches adopt the single-candidate model. Nevertheless the preference relationship between the antecedent candidates cannot be determined accurately in this model. By contrast, our approach adopts a twin-candidate learning model. Such a model can present the competition criterion for antecedent candidates reliably, and ensure that the most preferred candidate is selected. Furthermore, our approach applies a candidate filter to reduce the computational cost and data noises during training and resolution. The experimental results on MUC-6 and MUC-7 data set show that our approach can outperform those based on the single-candidate model.

1 Introduction

Coreference resolution is the process of linking together multiple expressions of a given entity. The key to solve this problem is to determine the antecedent for each referring expression in a document.

In coreference resolution, it is common that two or more candidates compete to be the antecedent of an anaphor (Mitkov, 1999). Whether a candidate is coreferential to an anaphor is often determined by the competition among all the candidates. So far, various algorithms have been proposed to determine the preference relationship between two candidates. Mitkov's knowledge-poor pronoun resolution method (Mitkov, 1998), for example, uses the scores from a set of antecedent indicators to rank the candidates. And centering algorithms (Brennan et al., 1987; Strube, 1998; Tetreault, 2001), sort the antecedent candidates based on the ranking of the *forward-looking* or *backward-looking* centers.

In recent years, supervised machine learning approaches have been widely used in coreference resolution (Aone and Bennett, 1995; McCarthy, 1996; Soon et al., 2001; Ng and Cardie, 2002a), and have achieved significant success. Normally, these approaches adopt a single-candidate model in which the classifier judges whether an antecedent candidate is coreferential to an anaphor with a confidence value. The confidence values are generally used as the competition criterion for the antecedent candidates. For example, the "Best-First" selection algorithms (Aone and Bennett, 1995; Ng and Cardie, 2002a) link the anaphor to the candidate with the maximal confidence value (above 0.5).

One problem of the single-candidate model, however, is that it only takes into account the relationships between an anaphor and one individual candidate at a time, and overlooks the preference relationship between candidates. Consequently, the confidence values cannot accurately represent the true competition criterion for the candidates.

In this paper, we present a competition learning approach to coreference resolution. Motivated by the research work by Connolly et al. (1997), our approach adopts a twin-candidate model to directly learn the competition criterion for the antecedent candidates. In such a model, a classifier is trained based on the instances formed by an anaphor and a pair of its antecedent candidates. The classifier is then used to determine the preference between any two candidates of an anaphor encountered in a new document. The candidate that wins the most comparisons is selected as the antecedent. In order to reduce the computational cost and data noises, our

approach also employs a candidate filter to eliminate the invalid or irrelevant candidates.

The layout of this paper is as follows. Section 2 briefly describes the single-candidate model and analyzes its limitation. Section 3 proposes in details the twin-candidate model and Section 4 presents our coreference resolution approach based on this model. Section 5 reports and discusses the experimental results. Section 6 describes related research work. Finally, conclusion is given in Section 7.

2 The Single-Candidate Model

The main idea of the single-candidate model for coreference resolution is to recast the resolution as a binary classification problem.

During training, a set of training instances is generated for each anaphor in an annotated text. An instance is formed by the anaphor and one of its antecedent candidates. It is labeled as positive or negative based on whether or not the candidate is tagged in the same coreferential chain of the anaphor.

After training, a classifier is ready to resolve the NPs[1] encountered in a new document. For each NP under consideration, every one of its antecedent candidates is paired with it to form a test instance. The classifier returns a number between 0 and 1 that indicates the likelihood that the candidate is coreferential to the NP.

The returned confidence value is commonly used as the competition criterion to rank the candidate. Normally, the candidates with confidences less than a selection threshold (e.g. 0.5) are discarded. Then some algorithms are applied to choose one of the remaining candidates, if any, as the antecedent. For example, "Closest-First" (Soon et al., 2001) selects the candidate closest to the anaphor, while "Best-First" (Aone and Bennett, 1995; Ng and Cardie, 2002a) selects the candidate with the maximal confidence value.

One limitation of this model, however, is that it only considers the relationships between a NP encountered and one of its candidates at a time during its training and testing procedures. The confidence value reflects the probability that the candidate is coreferential to the NP in the overall distribution[2], but not the conditional probability when the candidate is concurrent with other competitors. Consequently, the confidence values are unreliable to represent the true competition criterion for the candidates.

To illustrate this problem, just suppose a data set where an instance could be described with four exclusive features: F1, F2, F3 and F4. The ranking of candidates obeys the following rule:

$CS_{F1} >> CS_{F2} >> CS_{F3} >> CS_{F4}$

Here CS_{Fi} $(1 \le i \le 4)$ is the set of antecedent candidates with the feature Fi on. The mark of ">>" denotes the preference relationship, that is, the candidates in CS_{F1} is preferred to those in CS_{F2}, and to those in CS_{F3} and CS_{F4}.

Let CF_2 and CF_3 denote the class value of a leaf node "F2 = 1" and "F3 = 1", respectively. It is possible that $CF_2 < CF_3$, if the anaphors whose candidates all belong to CS_{F3} or CS_{F4} take the majority in the training data set. In this case, a candidate in CS_{F3} would be assigned a larger confidence value than a candidate in CS_{F2}. This nevertheless contradicts the ranking rules. If during resolution, the candidates of an anaphor all come from CS_{F2} or CS_{F3}, the anaphor may be wrongly linked to a candidate in CS_{F3} rather than in CS_{F2}.

3 The Twin-Candidate Model

Different from the single-candidate model, the twin-candidate model aims to learn the competition criterion for candidates. In this section, we will introduce the structure of the model in details.

3.1 Training Instances Creation

Consider an anaphor *ana* and its candidate set *candidate_set,* $\{C_1, C_2, \ldots, C_k\}$, where C_j is closer to *ana* than C_i if $j > i$. Suppose *positive_set* is the set of candidates that occur in the coreferential chain of *ana*, and *negative_set* is the set of candidates not in the chain, that is, *negative_set = candidate_set - positive_set*. The set of training instances based on *ana*, *inst_set*, is defined as follows:

[1] In this paper a NP corresponds to a Markable in MUC coreference resolution tasks.

[2] Suppose we use C4.5 algorithm and the class value takes the smoothed ration, $\frac{p+1}{t+2}$, where p is the number of positive instances and t is the total number of instances contained in the corresponding leaf node.

$inst_set =$

$\{inst_{(Ci, Cj, ana)} | i > j, C_i \in positve_set, C_j \in negative_set\} \cup$

$\{inst_{(Ci, Cj, ana)} | i > j, C_i \in negative_set, C_j \in positve_set\}$

From the above definition, an instance is formed by an anaphor, one positive candidate and one negative candidate. For each instance, $inst_{(ci, cj, ana)}$, the candidate at the first position, C_i, is closer to the anaphor than the candidate at the second position, C_j.

A training instance $inst_{(ci, cj, ana)}$ is labeled as positive if $C_i \in$ *positive-set* and $C_j \in$ *negative-set*; or negative if $C_i \in$ *negative-set* and $C_j \in$ *positive-set*.

See the following example:

> Any design to link China's accession to the WTO with *the missile tests*$_1$ was doomed to failure.
>
> "If *some countries*$_2$ try to block China TO accession, that will not be popular and will fail to win the support of *other countries*$_3$" she said.
>
> Although *no governments*$_4$ have suggested *formal sanctions*$_5$ on China over *the missile tests*$_6$, the United States has called ***them***$_7$ "provocative and reckless" and other countries said they could threaten Asian stability.

In the above text segment, the antecedent candidate set of the pronoun "*them*$_7$" consists of six candidates highlighted in Italics. Among the candidates, Candidate 1 and 6 are in the coreferential chain of "*them*$_7$", while Candidate 2, 3, 4, 5 are not. Thus, eight instances are formed for "*them*$_7$":

(2,1,7) (3,1,7) (4,1,7) (5,1,7)
(6,5,7) (6,4,7) (6,3,7) (6,2,7)

Here the instances in the first line are negative, while those in the second line are all positive.

3.2 Features Definition

A feature vector is specified for each training or testing instance. Similar to those in the single-candidate model, the features may describe the lexical, syntactic, semantic and positional relationships of an anaphor and any one of its candidates. Besides, the feature set may also contain inter-candidate features characterizing the relationships between the pair of candidates, e.g. the distance between the candidates in the number distances or paragraphs.

3.3 Classifier Generation

Based on the feature vectors generated for each anaphor encountered in the training data set, a classifier can be trained using a certain machine learning algorithm, such as C4.5, RIPPER, etc. Given the feature vector of a test instance $inst_{(ci, cj, ana)}$ (i > j), the classifier returns the positive class indicating that C_i is preferred to C_j as the antecedent of *ana*; or negative indicating that C_j is preferred.

3.4 Antecedent Identification

Let CR($inst_{(ci, cj, ana)}$) denote the classification result for an instance $inst_{(ci, cj, ana)}$. The antecedent of an anaphor is identified using the algorithm shown in Figure 1.

Algorithm ANTE-SEL
Input: *ana*: the anaphor under consideration
candidate_set: the set of antecedent candidates of *ana*, $\{C_1, C_2, \ldots, C_k\}$

for i = 1 **to** K **do**
  Score[i] = 0;
for i = K **downto** 2 **do**
  for j = i – 1 **downto** 1 **do**
    if CR($inst_{(ci, cj, ana)}$) = = *positive* **then**
      Score[i]++;
    else
      Score[j] ++;
    endif
SelectedIdx= $\arg\max_{i} \max_{Ci \in candidate_set} Score[i]$
return $C_{selectedIdx}$;

Figure 1:The antecedent identification algorithm

Algorithm ANTE-SEL takes as input an anaphor and its candidate set *candidate_set*, and returns one candidate as its antecedent. In the algorithm, each candidate is compared against any other candidate. The classifier acts as a judge during each comparison. The score of each candidate increases by one every time when it wins. In this way, the final score of a candidate records the total times it wins. The candidate with the maximal score is singled out as the antecedent.

If two or more candidates have the same maximal score, the one closest to the anaphor would be selected.

3.5 Single-Candidate Model: A Special Case of Twin-Candidate Model?

While the realization and the structure of the twin-candidate model are significantly different from the single-candidate model, the single-candidate model in fact can be regarded as a special case of the twin-candidate model.

To illustrate this, just consider a virtual "blank" candidate C_0 such that we could convert an instance $inst_{(ci, ana)}$ in the single-candidate model to an instance $inst_{(ci, c0, ana)}$ in the twin-candidate model. Let $inst_{(ci, c0, ana)}$ have the same class label as $inst_{(ci, ana)}$, that is, $inst_{(ci, c0, ana)}$ is positive if C_i is the antecedent of *ana*; or negative if not.

Apparently, the classifier trained on the instance set $\{ inst_{(ci, ana)} \}$, T1, is equivalent to that trained on $\{ inst_{(ci, c0, ana)} \}$, T2. T1 and T2 would assign the same class label for the test instances $inst_{(ci, ana)}$ and $inst_{(ci, c0, ana)}$, respectively. That is to say, determining whether C_i is coreferential to *ana* by T1 in the single-candidate model equals to determining whether C_i is better than C_0 w.r.t *ana* by T2 in the twin-candidate model. Here we could take C_0 as a "standard candidate".

While the classification in the single-candidate model can find its interpretation in the twin-candidate model, it is not true vice versa. Consequently, we can safely draw the conclusion that the twin-candidate model is more powerful than the single-candidate model in characterizing the relationships among an anaphor and its candidates.

4 The Competition Learning Approach

Our competition learning approach adopts the twin-candidate model introduced in the Section 3. The main process of the approach is as follows:

1. The raw input documents are preprocessed to obtain most, if not all, of the possible NPs.
2. During training, for each anaphoric NP, we create a set of candidates, and then generate the training instances as described in Section 3.
3. Based on the training instances, we make use of the C5.0 learning algorithm (Quinlan, 1993) to train a classifier.
4. During resolution, for each NP encountered, we also construct a candidate set. If the set is empty, we left this NP unresolved; otherwise we apply the antecedent identification algorithm to choose the antecedent and then link the NP to it.

4.1 Preprocessing

To determine the boundary of the noun phrases, a pipeline of Nature Language Processing components are applied to an input raw text:

- Tokenization and sentence segmentation
- Named entity recognition
- Part-of-speech tagging
- Noun phrase chunking

Among them, named entity recognition, part-of-speech tagging and text chunking apply the same Hidden Markov Model (HMM) based engine with error-driven learning capability (Zhou and Su, 2000 & 2002). The named entity recognition component recognizes various types of MUC-style named entities, i.e., organization, location, person, date, time, money and percentage.

4.2 Features Selection

For our study, in this paper we only select those features that can be obtained with low annotation cost and high reliability. All features are listed in Table 1 together with their respective possible values.

4.3 Candidates Filtering

For a NP under consideration, all of its preceding NPs could be the antecedent candidates. Nevertheless, since in the twin-candidate model the number of instances for a given anaphor is about the square of the number of its antecedent candidates, the computational cost would be prohibitively large if we include all the NPs in the candidate set. Moreover, many of the preceding NPs are irrelevant or even invalid with regard to the anaphor. These data noises may hamper the training of a good-performanced classifier, and also damage the accuracy of the antecedent selection: too many comparisons are made between incorrect candidates. Therefore, in order to reduce the computational cost and data noises, an effective candidate filtering strategy must be applied in our approach.

During training, we create the candidate set for each anaphor with the following filtering algorithm:

1. If the anaphor is a pronoun,
   (a) Add to the initial candidate set all the preceding NPs in the current and the previous two sentences.

Features describing the candidate:		
1.	ante_DefNp_1(2)	1 if C_i (C_j) is a definite NP; else 0
2.	ante_IndefNP_1(2)	1 if C_i (C_j) is an indefinite NP; else 0
3.	ante_Pron_1(2)	1 if C_i (C_j) is a pronoun; else 0
4.	ante_ProperNP_1(2)	1 if C_i (C_j) is a proper NP; else 0
5.	ante_M_ProperNP_1(2)	1 if C_i (C_j) is a mentioned proper NP; else 0
6.	ante_ProperNP_APPOS_1(2)	1 if C_i (C_j) is a proper NP modified by an appositive; else 0
7.	ante_Appositive_1(2)	1 if C_i (C_j) is in a apposition structure; else 0
8.	ante_NearestNP_1(2)	1 if C_i (C_j) is the nearest candidate to the anaphor; else 0
9.	ante_Embeded_1(2)	1 if C_i (C_j) is in an embedded NP; else 0
10	ante_Title_1(2)	1 if C_i (C_j) is in a title; else 0
Features describing the anaphor:		
11.	ana_DefNP	1 if *ana* is a definite NP; else 0
12.	ana_IndefNP	1 if *ana* is an indefinite NP; else 0
13.	ana_Pron	1 if *ana* is a pronoun; else 0
14.	ana_ProperNP	1 if *ana* is a proper NP; else 0
15.	ana_PronType	1 if *ana* is a third person pronoun; 2 if a single neuter pronoun; 3 if a plural neuter pronoun; 4 if other types
16.	ana_FlexiblePron	1 if *ana* is a flexible pronoun; else 0
Features describing the candidate and the anaphor:		
17.	ante_ana_StringMatch_1(2)	1 if C_i (C_j) and *ana* match in string; else 0
18.	ante_ana_GenderAgree_1(2)	1 if C_i (C_j) and *ana* agree in gender; else 0 if disagree; -1 if unknown
18.	ante_ana_NumAgree_1(2)	1 if C_i (C_j) and *ana* agree in number; 0 if disagree; -1 if unknown
20.	ante_ana_Appositive_1(2)	1 if C_i (C_j) and *ana* are in an appositive structure; else 0
21.	ante_ana_Alias_1(2)	1 if C_i (C_j) and *ana* are in an alias of the other; else 0
Features describing the two candidates		
22.	inter_SDistance	Distance between C_i and C_j in sentences
23.	inter_Pdistance	Distance between C_i and C_j in paragraphs

Table 1: Feature set for coreference resolution (Feature 22, 23 and features involving C_j are not used in the single-candidate model)

(b) Remove from the candidate set those that disagree in number, gender, and person.
(c) If the candidate set is empty, add the NPs in an earlier sentence and go to 1(b).

2. If the anaphor is a non-pronoun,
(a) Add all the non-pronominal antecedents to the initial candidate set.
(b) For each candidate added in 2(a), add the non-pronouns in the current, the previous and the next sentences into the candidate set.

During resolution, we filter the candidates for each encountered pronoun in the same way as during training. That is, we only consider the NPs in the current and the preceding 2 sentences. Such a context window is reasonable as the distance between a pronominal anaphor and its antecedent is generally short. In the MUC-6 data set, for example, the immediate antecedents of 95% pronominal anaphors can be found within the above distance.

Comparatively, candidate filtering for non-pronouns during resolution is complicated. A potential problem is that for each non-pronoun under consideration, the twin-candidate model always chooses a candidate as the antecedent, even though all of the candidates are "low-qualified", that is, unlikely to be coreferential to the non-pronoun under consideration.

In fact, the twin-candidate model in itself can identify the qualification of a candidate. We can compare every candidate with a virtual "standard candidate", C_0. Only those better than C_0 are deemed qualified and allowed to enter the "round robin", whereas the losers are eliminated. As we have discussed in Section 3.5, the classifier on the pairs of a candidate and C_0 is just a single-candidate classifier. Thus, we can safely adopt the single-candidate classifier as our candidate filter.

The candidate filtering algorithm during resolution is as follows:

1. If the current NP is a pronoun, construct the candidate set in the same way as during training.
2. If the current NP is a non-pronoun,

(a) Add all the preceding non-pronouns to the initial candidate set.
(b) Calculate the confidence value for each candidate using the single-candidate classifier.
(c) Remove the candidates with confidence value less than 0.5.

5 Evaluation and Discussion

Our coreference resolution approach is evaluated on the standard MUC-6 (1995) and MUC-7 (1998) data set. For MUC-6, 30 "dry-run" documents annotated with coreference information could be used as training data. There are also 30 annotated training documents from MUC-7. For testing, we utilize the 30 standard test documents from MUC-6 and the 20 standard test documents from MUC-7.

5.1 Baseline Systems

In the experiment we compared our approach with the following research works:

1. Strube's S-list algorithm for pronoun resolution (Stube, 1998).
2. Ng and Cardie's machine learning approach to coreference resolution (Ng and Cardie, 2002a).
3. Connolly et al.'s machine learning approach to anaphora resolution (Connolly et al., 1997).

Among them, S-List, a version of centering algorithm, uses well-defined heuristic rules to rank the antecedent candidates; Ng and Cardie's approach employs the standard single-candidate model and "Best-First" rule to select the antecedent; Connolly et al.'s approach also adopts the twin-candidate model, but their approach lacks of candidate filtering strategy and uses greedy linear search to select the antecedent (See "Related work" for details).

We constructed three baseline systems based on the above three approaches, respectively. For comparison, in the baseline system 2 and 3, we used the similar feature set as in our system (see table 1).

5.2 Results and Discussion

Table 2 and 3 show the performance of different approaches in the pronoun and non-pronoun resolution, respectively. In these tables we focus on the abilities of different approaches in resolving an anaphor to its antecedent correctly. The recall measures the number of correctly resolved anaphors over the total anaphors in the MUC test data set, and the precision measures the number of correct anaphors over the total resolved anaphors. The F-measure F=2*RP/(R+P) is the harmonic mean of precision and recall.

The experimental result demonstrates that our competition learning approach achieves a better performance than the baseline approaches in resolving pronominal anaphors. As shown in Table 2, our approach outperforms Ng and Cardie's single-candidate based approach by 3.7 and 5.4 in F-measure for MUC-6 and MUC-7, respectively. Besides, compared with Strube's S-list algorithm, our approach also achieves gains in the F-measure by 3.2 (MUC-6), and 1.6 (MUC-7). In particular, our approach obtains significant improvement (21.1 for MUC-6, and 13.1 for MUC-7) over Connolly et al.'s twin-candidate based approach.

	MUC-6			MUC-7		
	R	P	F	R	P	F
Strube (1998)	76.1	74.3	75.1	62.9	60.3	61.6
Ng and Cardie (2002a)	75.4	73.8	74.6	58.9	56.8	57.8
Connolly et al. (1997)	57.2	57.2	57.2	50.1	50.1	50.1
Our approach	**79.3**	**77.5**	**78.3**	**64.4**	**62.1**	**63.2**

Table 2: Results for the pronoun resolution

	MUC-6			MUC-7		
	R	P	F	R	P	F
Ng and Cardie (2002a)	51.0	89.9	65.0	39.1	86.4	53.8
Connolly et al. (1997)	**52.2**	52.2	52.2	**43.7**	43.7	43.7
Our approach	51.3	**90.4**	**65.4**	39.7	**87.6**	**54.6**

Table 3: Results for the non-pronoun resolution

	MUC-6			MUC-7		
	R	P	F	R	P	F
Ng and Cardie (2002a)	62.2	78.8	69.4	48.4	74.6	58.7
Our approach	**64.0**	**80.5**	**71.3**	**50.1**	**75.4**	**60.2**

Table 4: Results for the coreference resolution

Compared with the gains in pronoun resolution, the improvement in non-pronoun resolution is slight. As shown in Table 3, our approach resolves non-pronominal anaphors with the recall of 51.3 (39.7) and the precision of 90.4 (87.6) for MUC-6 (MUC-7). In contrast to Ng and Cardie's approach, the performance of our approach improves only 0.3 (0.6) in recall and 0.5 (1.2) in precision. The reason may be that in non-pronoun resolution, the coreference of an anaphor and its candidate is usually determined only by some strongly indicative features such as alias, apposition, string-matching, etc (this explains why we obtain a high precision but a low recall in non-pronoun resolution). Therefore, most of the positive candidates are coreferential to the anaphors even though they are not the "best". As a result, we can only see comparatively slight difference between the performances of the two approaches.

Although Connolly et al.'s approach also adopts the twin-candidate model, it achieves a poor performance for both pronoun resolution and non-pronoun resolution. The main reason is the absence of candidate filtering strategy in their approach (this is why the recall equals to the precision in the tables). Without candidate filtering, the recall may rise as the correct antecedents would not be eliminated wrongly. Nevertheless, the precision drops largely due to the numerous invalid NPs in the candidate set. As a result, a significantly low F-measure is obtained in their approach.

Table 4 summarizes the overall performance of different approaches to coreference resolution. Different from Table 2 and 3, here we focus on whether a coreferential chain could be correctly identified. For this purpose, we obtain the recall, the precision and the F-measure using the standard MUC scoring program (Vilain et al. 1995) for the coreference resolution task. Here the recall means the correct resolved chains over the whole coreferential chains in the data set, and precision means the correct resolved chains over the whole resolved chains.

In line with the previous experiments, we see reasonable improvement in the performance of the coreference resolution: compared with the baseline approach based on the single-candidate model, the F-measure of approach increases from 69.4 to 71.3 for MUC-6, and from 58.7 to 60.2 for MUC-7.

6 Related Work

A similar twin-candidate model was adopted in the anaphoric resolution system by Connolly et al. (1997). The differences between our approach and theirs are:

(1) In Connolly et al.'s approach, all the preceding NPs of an anaphor are taken as the antecedent candidates, whereas in our approach we use candidate filters to eliminate invalid or irrelevant candidates.

(2) The antecedent identification in Connolly et al.'s approach is to apply the classifier to successive pairs of candidates, each time retaining the better candidate. However, due to the lack of strong assumption of transitivity, the selection procedure is in fact a greedy search. By contrast, our approach evaluates a candidate according to the times it wins over the other competitors. Comparatively this algorithm could lead to a better solution.

(3) Our approach makes use of more indicative features, such as Appositive, Name Alias, String-matching, etc. These features are effective especially for non-pronoun resolution.

7 Conclusion

In this paper we have proposed a competition learning approach to coreference resolution. We started with the introduction of the single-candidate model adopted by most supervised machine learning approaches. We argued that the confidence values returned by the single-candidate classifier are not reliable to be used as ranking criterion for antecedent candidates. Alternatively, we presented a twin-candidate model that learns the competition criterion for antecedent candidates directly. We introduced how to adopt the twin-candidate model in our competition learning ap-

proach to resolve the coreference problem. Particularly, we proposed a candidate filtering algorithm that can effectively reduce the computational cost and data noises.

The experimental results have proved the effectiveness of our approach. Compared with the baseline approach using the single-candidate model, the F-measure increases by 1.9 and 1.5 for MUC-6 and MUC-7 data set, respectively. The gains in the pronoun resolution contribute most to the overall improvement of coreference resolution.

Currently, we employ the single-candidate classifier to filter the candidate set during resolution. While the filter guarantees the qualification of the candidates, it removes too many positive candidates, and thus the recall suffers. In our future work, we intend to adopt a looser filter together with an anaphoricity determination module (Bean and Riloff, 1999; Ng and Cardie, 2002b). Only if an encountered NP is determined as an anaphor, we will select an antecedent from the candidate set generated by the looser filter. Furthermore, we would like to incorporate more syntactic features into our feature set, such as grammatical role or syntactic parallelism. These features may be helpful to improve the performance of pronoun resolution.

References

Chinatsu Aone and Scott W.Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Pages 122-129.

D.Bean and E.Riloff. 1999. Corpus-Based identification of non-anaphoric noun phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Pages 373-380.

Brennan, S, E., M. W. Friedman and C. J. Pollard. 1987. A Centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of The Association for Computational Linguistics*, Page 155-162.

Dennis Connolly, John D. Burger and David S. Day. 1997. A machine learning approach to anaphoric reference. *New Methods in Language Processing*, Page 133-144.

Joseph F. McCarthy. 1996. A trainable approach to coreference resolution for Information Extraction. Ph.D. thesis. University of Massachusetts.

Ruslan Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *Proceedings of the 17th Int. Conference on Computational Linguistics (COLING-ACL'98)*, Page 869-875.

Ruslan Mitkov. 1999. Anaphora resolution: The state of the art. Technical report. University of Wolverhampton, Wolverhampton.

MUC-6. 1995. Proceedings of the Sixth Message Understanding Conference (MUC-6). Morgan Kaufmann, San Francisco, CA.

MUC-7. 1998. Proceedings of the Seventh Message Understanding Conference (MUC-7). Morgan Kaufmann, San Francisco, CA.

Vincent Ng and Claire Cardie. 2002a. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40rd Annual Meeting of the Association for Computational Linguistics,* Pages 104-111.

Vincent Ng and Claire Cardie. 2002b. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of 19th International Conference on Computational Linguistics (COLING-2002)*.

J R. Quinlan. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

Wee Meng Soon, Hwee Tou Ng and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4), Page 521-544.

Michael Strube. Never look back: An alternative to Centering. 1998. In *Proceedings of the 17th Int. Conference on Computational Linguistics and 36th Annual Meeting of ACL*, Page 1251-1257

Joel R. Tetreault. 2001. A Corpus-Based evaluation of Centering and pronoun resolution. *Computational Linguistics*, 27(4), Page 507-520.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L.Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message understanding Conference (MUC-6)*, Pages 42-52.

GD Zhou and J. Su, 2000. Error-driven HMM-based chunk tagger with context-dependent lexicon. In *Proceedings of the Joint Conference on Empirical Methods on Natural Language Processing and Very Large Corpus (EMNLP/ VLC'2000)*.

GD Zhou and J. Su. 2002. Named Entity recognition using a HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* P473-478.