Capturing Salience with a Trainable Cache Model for Zero-anaphora Resolution

Ryu Iida
Department of Computer Science
Tokyo Institute of Technology
2-12-1, Ôokayama, Meguro,
Tokyo 152-8552, Japan
ryu-i@cl.cs.titech.ac.jp

Kentaro Inui **Yuji Matsumoto**
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma
Nara 630-0192, Japan
{inui,matsu}@is.naist.jp

Abstract

This paper explores how to apply the notion of caching introduced by Walker (1996) to the task of zero-anaphora resolution. We propose a machine learning-based implementation of a cache model to reduce the computational cost of identifying an antecedent. Our empirical evaluation with Japanese newspaper articles shows that the number of candidate antecedents for each zero-pronoun can be dramatically reduced while preserving the accuracy of resolving it.

1 Introduction

There have been recently increasing concerns with the need for anaphora resolution to make NLP applications such as IE and MT more reliable. In particular, for languages such as Japanese, anaphora resolution is crucial for resolving a phrase in a text to its referent since phrases, especially nominative arguments of predicates, are frequently omitted by anaphoric functions in discourse (Iida et al., 2007b).

Many researchers have recently explored machine learning-based methods using considerable amounts of annotated data provided by, for example, the Message Understanding Conference and Automatic Context Extraction programs (Soon et al., 2001; Ng and Cardie, 2002; Yang et al., 2008; McCallum and Wellner, 2003, etc.). These methods reach a level comparable to or better than the state-of-the-art rule-based systems (e.g. Baldwin (1995)) by recasting the task of anaphora resolution into classification or clustering problems. However, such approaches tend to disregard theoretical findings from discourse theories, such as Centering Theory (Grosz et al., 1995). Therefore, one of the challenging issues in this area is to incorporate such findings from linguistic theories into machine learning-based approaches.

A typical machine learning-based approach to zero-anaphora resolution searches for an antecedent in the set of candidates appearing in all the preceding contexts. However, computational time makes this approach largely infeasible for long texts. An alternative approach is to heuristically limit the search space (e.g. the system deals with candidates only occurring in the N previous sentences). Various research such as Yang et al. (2008) has adopted this approach, but it also leads to problems when an antecedent is located far from its anaphor, causing it to be excluded from target candidate antecedents.

On the other hand, rule-based methods derived from theoretical background such as Centering Theory (Grosz et al., 1995) only deal with the salient discourse entities at each point of the discourse status. By incrementally updating the discourse status, the set of candidates in question is automatically limited. Although these methods have a theoretical advantage, they have a serious drawback in that Centering Theory only retains information about the previous sentence. A few methods have attempted to overcome this fault (Suri and McCoy, 1994; Hahn and Strube, 1997), but they are overly dependent upon the restrictions fundamental to the notion of centering. We hope that by relaxing such restrictions it will be possible for an anaphora resolution system to achieve a good balance between accuracy and computational cost.

From this background, we focus on the issue of reducing candidate antecedents (discourse entities) for a given anaphor. Inspired by Walker's argument (Walker, 1996), we propose a machine learning-based caching mechanism that captures the most salient candidates at each point of the discourse for efficient anaphora resolution. More specifically, we choose salient candidates for each sentence from the set of candidates appearing in that sentence and the candidates which are already

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 647–655,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP

in the cache. Searching only through the set of salient candidates, the computational cost of zero-anaphora resolution is effectively reduced. In the empirical evaluation, we investigate how efficiently this caching mechanism contributes to reducing the search space while preserving accuracy. This paper focuses on Japanese though the proposed cache mechanism may be applicable to any language.

This paper is organized as follows. First, Section 2 presents the task of zero-anaphora resolution and then Section 3 gives an overview of previous work. Next, in Section 4 we propose a machine learning-based cache model. Section 5 presents the antecedent identification and anaphoricity determination models used in the experiments. To evaluate the model, we conduct several empirical evaluations and report their results in Section 6. Finally, we conclude and discuss the future direction of this research in Section 7.

2 Zero-anaphora resolution

In this paper, we consider only zero-pronouns that function as an obligatory argument of a predicate. A zero-pronoun may or may not have its antecedent in the discourse; in the case it does, we say the zero-pronoun is *anaphoric*. On the other hand, a zero-pronoun whose referent does not explicitly appear in the discourse is called a *non-anaphoric* zero-pronoun. A zero-pronoun is typically non-anaphoric when it refers to an extralinguistic entity (e.g. the first or second person) or its referent is unspecified in the context.

The task of zero-anaphora resolution can be decomposed into two subtasks: *anaphoricity determination* and *antecedent identification*. In anaphoricity determination, the model judges whether a zero-pronoun is anaphoric (i.e. a zero-pronoun has an antecedent in a text) or not. If a zero-pronoun is anaphoric, the model must detect its antecedent. For example, in example (1) the model has to judge whether or not the zero-pronoun in the second sentence (i.e. the nominative argument of the predicate 'to hate') is anaphoric, and then identify its correct antecedent as 'Mary.'

(1)

Mary$_i$-wa	*John$_j$-ni*	(ϕ_j-*ga*)	*tabako-o*
Mary$_i$-TOP	John$_j$-DAT	(ϕ_j-NOM)	smoking-OBJ

yameru-youni	*it-ta*	.
quit-COMP	say-PAST	PUNC

Mary told John to quit smoking.

(ϕ_i-*ga*)	*tabako-o*	*kirai-dakarada*	.
(ϕ_i-NOM)	smoking-OBJ	hate-BECAUSE	PUNC

Because (she) hates people smoking.

3 Previous work

Early methods for zero-anaphora resolution were developed with rule-based approaches in mind. Theory-oriented rule-based methods (Kameyama, 1986; Walker et al., 1994), for example, focus on the Centering Theory (Grosz et al., 1995) and are designed to collect the salient candidate antecedents in the *forward-looking center* (*Cf*) list, and then choose the most salient candidate, *Cp*, as an antecedent of a zero-pronoun according to heuristic rules (e.g. *topic* $>$ *subject* $>$ *indirect object* $>$ *direct object* $>$ *others*[1]). Although these methods have a theoretical advantage, they have a serious drawback in that the original Centering Theory is restricted to keeping information about the previous sentence only. In order to loosen this restriction, the Centering-based methods have been extended for reaching an antecedent appearing further from its anaphor. For example, Suri and McCoy (1994) proposed a method for capturing two kinds of *Cp*, that correspond to the most salient discourse entities within the local transition and within the global focus of a text. Hahn and Strube (1997) estimate hierarchical discourse segments of a text by taking into account a series of *Cp* and then the resolution model searches for an antecedent in the estimated segment. Although these methods remedy the drawback of Centering, they still overly depend on the notion of Centering such as *Cp*.

On the other hand, the existing machine learning-based methods (Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Soon et al., 2001; Ng and Cardie, 2002; Seki et al., 2002; Isozaki and Hirao, 2003; Iida et al., 2005; Iida et al., 2007a, etc.) have been developed with less attention given to such a problem. These methods exhaustively search for an antecedent within the list of all candidate antecedents until the beginning of the text. Otherwise, the process to search for antecedents is heuristically carried out in a limited search space (e.g. the previous N sentences of an anaphor) (Yang et al., 2008).

4 Machine learning-based cache model

As mentioned in Section 2, the procedure for zero-anaphora resolution can be decomposed into two subtasks, namely anaphoricity determination and antecedent identification. In this paper, these two subtasks are carried out according to the *selection-then-classification model* (Iida et al.,

[1] '$A > B$' means *A* is more salient than *B*.

2005), chosen because it it has the advantage of using broader context information for determining the anaphoricity of a zero-pronoun. It does this by examining whether the context preceding the zero-pronoun in the discourse has a plausible candidate antecedent or not. With this model, antecedent identification is performed first, and anaphoricity determination second, that is, the model identifies the most likely candidate antecedent for a given zero-pronoun and then it judges whether or not the zero-pronoun is anaphoric.

As discussed by Iida et al. (2007a), intra-sentential and inter-sentential zero-anaphora resolution should be dealt with by taking into account different kinds of information. Syntactic patterns are useful clues for intra-sentential zero-anaphora resolution, whereas rhetorical clues such as connectives may be more useful for inter-sentential cases. Therefore, the intra-sentential and inter-sentential zero-anaphora resolution models are separately trained by exploiting different feature sets as shown in Table 2.

In addition, as mentioned in Section 3, inter-sentential cases have a serious problem where the search space of zero-pronouns grows linearly with the length of the text. In order to avoid this problem, we incorporate a caching mechanism originally addressed by Walker (1996) into the following procedure of zero-anaphora resolution by limiting the search space at step 3 and by updating the cache at step 5.

Zero-anaphora resolution process:

1. *Intra-sentential antecedent identification*: For a given zero-pronoun *ZP* in a given sentence S, select the most-likely candidate antecedent A_1 from the candidates appearing in S by the intra-sentential antecedent identification model.
2. *Intra-sentential anaphoricity determination*: Estimate plausibility p_1 that A_1 is the true antecedent, and return A_1 if $p_1 \geq \theta_{intra}$[2] or go to 3 otherwise.
3. *Inter-sentential antecedent identification*: Select the most-likely candidate antecedent A_2 from the candidates appearing in the cache as explained in Section 4.1 by the inter-sentential antecedent identification model.
4. *Inter-sentential anaphoricity determination*: Estimate plausibility p_2 that A_2 is the true antecedent, and return A_2 if $p_2 \geq \theta_{inter}$[3] or return *non-anaphoric* otherwise.
5. After processing all zero-pronouns in S, the cache is updated. The resolution process is continued until the end of the discourse.

4.1 Dynamic cache model

Because the original work of the cache model by Walker (1996) is not fully specified for implementation, we specify how to retain the salient candidates based on machine learning in order to capture both local and global foci of discourse.

In Walker (1996)'s discussion of the cache model in discourse processing, it was presumed to operate under a limited attention constraint. According to this constraint, only a limited number of candidates can be considered in processing. Applying the concept of cache to computer hardware, the cache represents working memory and the *main memory* represents long-term memory. The cache only holds the most salient entities, while the rest are moved to the main memory for possible later consideration as a cache candidate. If a new candidate antecedent is retrieved from main memory and inserted into the cache, or enters the cache directly during processing, other candidates in the cache have to be displaced due to the limited capacity of the cache. Which candidate to displace is determined by a cache replacement policy. However, the best policy for this is still unknown.

In this paper, we recast the cache replacement policy as a ranking problem in machine learning. More precisely, we choose the N best candidates for each sentence from the set of candidates appearing in that sentence and the candidates that are already in the cache. Following this cache model, named the *dynamic cache model*, anaphora resolution is performed by repeating the following two processes.

1. *Cache update*: cache C_i for sentence S_i is created from the candidates in the previous sentence S_{i-1} and the ones in the previous cache C_{i-1}.
2. *Inter-sentential zero-anaphora resolution*: cache C_i is used as the search space for inter-sentential zero-anaphora resolution in sentence S_i (see Step 3 of the aforementioned zero-anaphora resolution process).

For each cache update (see Figure 1), a current cache C_i is created by choosing the N most salient candidates from the M candidates in S_{i-1} and the N candidates in the previous cache C_{i-1}. In order to implement this mechanism, we train the model

[2] θ_{intra} is a preselected threshold.

[3] θ_{inter} is a preselected threshold.

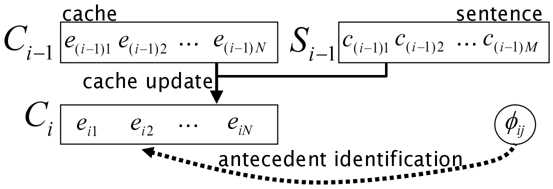

cache
sentence
C_{i-1}
$e_{(i-1)1}$ $e_{(i-1)2}$ ··· $e_{(i-1)N}$
S_{i-1}
$c_{(i-1)1}$ $c_{(i-1)2}$ ··· $c_{(i-1)M}$
cache update
C_i
e_{i1} e_{i2} ··· e_{iN}
ϕ_{ij}
antecedent identification

Figure 1: Anaphora resolution using the dynamic cache model

so that it captures the salience of each candidate.

To reflect this, each training instance is labeled as either *retained* or *discarded*. If an instance is referred to by an zero-pronoun appearing in any of the following sentences, it is labeled as *retained*; otherwise, it is labeled as *discarded*. Training instances are created in the algorithm detailed in Figure 2. The algorithm is designed with the following two points in mind.

First, the cache model must capture the salience of each discourse entity according to the recency of its entity at each discourse status because typically the more recently an entity appears, the more salient it is. To reflect this, training instances are created from candidates as they appear in the text, and are labeled as *retained* from the point of their appearance until their referring zero-pronoun is reached, at which time they are labeled as *discarded* if they are never referred to by any zero-pronouns in the succeeding context.

Suppose, the situation shown in Figure 3, where c_{ij} is the j-th candidate in sentence S_i. In this situation, for example, candidate c_{12} is labeled as *retained* when creating training instances for sentence S_1, but labeled as *discarded* from S_2 onwards, because of the appearance of its zero-pronoun. Another candidate c_{13} which is never referred to in the text is labeled as *discarded* for all training instances.

Second, we need to capture the 'relative' salience of candidates appearing in the current discourse for each cache update, as also exploited in the tournament-based or ranking-based approaches to anaphora resolution (Iida et al., 2003; Yang et al., 2003; Denis and Baldridge, 2008). To solve it, we use a ranker trained on the instances created as described above. In order to train the ranker, we adopt the Ranking SVM algorithm (Joachims, 2002), which learns a weight vector to rank candidates for a given partial ranking of each discourse entity. Each training instance is created from the set of retained candidates, R_i, paired with the set of discarded candidates, D_i, in each sentence. To define the partial ranking of candidates, we simply rank candidates in R_i as first place and candidates in D_i as second place.

```
Function makeTrainingInstances ($T$: input text)
    $C$ := NULL // set of preceding candidates
    $S$ := NULL // set of training instances
    $i$ := 1; // init
    while (exists $s_i$) // $s_i$: i-th sentence in $T$
        $E_i$ := extractCandidates($s_i$)
        $R_i$ := extractRetainedInstances($E_i, T$)
        $D_i$ := $E_i \backslash R_i$
        $r_i$:= extractRetainedInstances($C, T$)
        $R_i$ := $R_i \cup r_i$
        $D_i$ := $D_i \cup (C \backslash r_i)$
        $S$ := $S \cup \{\langle R_i, D_i \rangle\}$
        $C$ := updateSalienceInfo($C$)
        $C$ := $C \cup E_i$
        $i$ := $i$ + 1
    endwhile
    return $S$
end
Function extractRetainedInstances ($S$, $T$)
    $R$ := NULL // init
    while ($elm \in S$)
        if ($elm$ is anaphoric with a zero-pronoun located
             in the following sentences of $T$)
        $R$ := $R \cup elm$
        endif
    endwhile
    return $R$
end
Function updateSalienceInfo ($C$, $s_i$)
    while ($c \in C$)
        if ($c$ is anaphoric with a zero pronoun in $s_i$)
            $c$.position := $i$; // update the position information
        endif
    endwhile
    return $C$
end
```

Figure 2: Pseudo-code for creating training instances

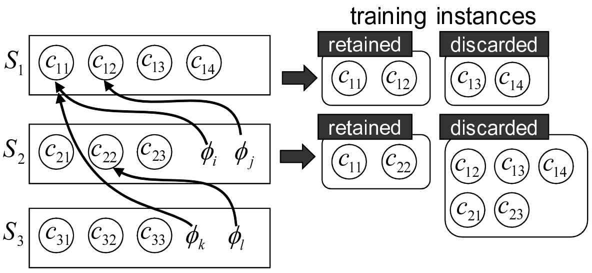

training instances
S_1
c_{11} c_{12} c_{13} c_{14}
retained
c_{11} c_{12}
discarded
c_{13} c_{14}
S_2
c_{21} c_{22} c_{23} ϕ_i ϕ_j
retained
c_{11} c_{22}
discarded
c_{12} c_{13} c_{14} c_{21} c_{23}
S_3
c_{31} c_{32} c_{33} ϕ_k ϕ_l

Figure 3: Creating training instnaces

4.2 Static cache model

Other research on discourse such as Grosz and Sidner (1986) has studied *global focus*, which generally refers to the entity or set of entities that are salient throughout the entire discourse. Since global focus may not be captured by Centering-based models, we also propose another cache model which directly captures the global salience of a text.

To train the model, all the candidates in a text which have an inter-sentential anaphoric relation with zero-pronouns are used as positive instances and the others used as negative ones. Unlike the

Table 1: Feature set used in the cache models

Feature	Description
POS	Part-of-speech of *C* followed by IPADIC[4].
IN_QUOTE	1 if *C* is located in a quoted sentence; otherwise 0.
BEGINNING	1 if *C* is located in the beginnig of a text; otherwise 0.
CASE_MARKER	Case marker, such as *wa* (TOPIC) and *ga* (SUBJECT), of *C*.
DEP_END	1 if *C* has a dependency relation with the last bunsetsu unit (i.e. a basic unit in Japanese) in a sentence ; otherwise 0.
CONN*	The set of connectives intervening between *C* and *Z*. Each conjunction is encoded into a binary feature.
IN_CACHE*	1 if *C* is currently stored in the cache; otherwise 0.
SENT_DIST*	Distance between *C* and *Z* in terms of a sentence.
CHAIN_NUM	The number of anaphoric chain, i.e. the number of antecedents of *Z* in the situation that zero-pronouns in the preceding contexts are completely resolved by the zero-anaphora resolution model.

C is a candidate antecedent, and *Z* stands for a target zero-pronoun. Features marked with an asterisk are only used in the dynamic cache model.

dynamic cache model, this model does not update the cache dynamically, but simply selects for each given zero-pronoun the N most salient candidates from the preceding sentences according to the rank provided by the trained ranker. We call this model the *static cache model*.

4.3 Features used in the cache models

The feature set used in the cache model is shown in Table 1. The 'CASE_MARKER' feature roughly captures the salience of the local transition dealt with in Centering Theory, and is also intended to capture the global foci of a text coupled with the BEGINNING feature. The CONN feature is expected to capture the transitions of a discourse relation because each connective functions as a marker of a discourse relation between two adjacent discourse segments.

In addition, the recency of a candidate antecedent can be even important when an entity occurs as a zero-pronoun in discourse. For example, when a discourse entity e appearing in sentence s_i is referred to by a zero-pronoun later in sentence $s_{j(i<j)}$, entity e is considered salient again at the point of s_j. To reflect this way of updating salience, we overwrite the information about the appearance position of candidate e in s_j, which is performed by the function *updateSalienceInfo* in Figure 2. This allows the cache model to handle updated salience features such as CHAIN_NUM in proceeding cache updates.

5 Antecedent identification and anaphoricity determination models

As an antecedent identification model, we adopt the tournament model (Iida et al., 2003) because in a preliminary experiment it achieved better performance than other state-of-the-art ranking-based models (Denis and Baldridge, 2008) in this task setting. To train the tournament model, the training instances are created by extracting an antecedent paired with each of the other candidates for learning a preference of which candidate is more likely to be an antecedent. At the test phase, the model conducts a tournament consisting of a series of matches in which candidate antecedents compete with one another. Note that in the case of intersentential zero-anaphora resolution the tournament is arranged between candidates in the cache. For learning the difference of two candidates in the cache, training instances are also created by only extracting candidates from the cache.

For anaphoricity determination, the model has to judge whether a zero-pronoun is anaphoric or not. To create the training instances for the binary classifier, the most likely candidate of each given zero-pronoun is chosen by the tournament model and then it is labeled as anaphoric (positive) if the chosen candidate is indeed the antecedent of the zero-pronoun[5], or otherwise labeled as non-anaphoric (negative).

To create models for antecedent identification and anaphoricity determination, we use a Support Vector Machine (Vapnik, 1998)[6] with a linear kernel and its default parameters. To use the feature set shown in Table 2, morpho-syntactic analysis of a text is performed by the Japanese morpheme analyzer *Chasen* and the dependency parser *CaboCha*. In the tournament model, the features of two competing candidates are distinguished from each other by adding the prefix of either 'left' or 'right.'

6 Experiments

We investigate how the cache model contributes to candidate reduction. More specifically, we ex-

[4] http://chasen.naist.jp/stable/ipadic/

[5] In the original selection-then-classification model (Iida et al., 2005), positive instances are created by all the correct pairs of a zero-pronoun and its antecedent, however in this paper we use only antecedents selected by the tournament model as the most likely candidates in the set of candidates because this method leads to better performance.

[6] http://svmlight.joachims.org/

Table 2: Feature set used in zero-anaphora resolution

Feature Type	Feature	Description
Lexical	HEAD_BF	Characters of right-most morpheme in *NP* (*PRED*).
	PRED_FUNC	Characters of functional words followed by *PRED*.
Grammatical	PRED_VOICE	1 if *PRED* contains auxiliaries such as '*(ra)reru*'; otherwise 0.
	POS	Part-of-speech of *NP* (*PRED*) followed by IPADIC (Asahara and Matsumoto, 2003).
	PARTICLE	Particle followed by *NP*, such as '*wa* (topic)', '*ga* (subject)', '*o* (object)'.
Semantic	NE	Named entity of *NP*: PERSON, ORGANIZATION, LOCATION, ARTIFACT, DATE, TIME, MONEY, PERCENT or N/A.
	SELECT_PREF	The score of selectional preference, which is the mutual information estimated from a large number of triplets $\langle$*Noun*, *Case*, *Predicate*$\rangle$.
Positional	SENTNUM	Distance between *NP* and *PRED*.
	BEGINNING	1 if *NP* is located in the beggining of sentence; otherwise 0.
	END	1 if *NP* is located in the end of sentence; otherwise 0.
	PRED_NP	1 if *PRED* precedes *NP*; otherwise 0.
	NP_PRED	1 if *NP* precedes *PRED*; otherwise 0.
Discourse	CL_RANK	A rank of *NP* in forward looking-center list.
	CL_ORDER	A order of *NP* in forward looking-center list.
	CONN**	The connectives intervesing between *NP* and *PRED*.
Path	PATH_FUNC*	Characters of functional words in the shortest path in the dependency tree between *PRED* and *NP*.
	PATH_POS*	Part-of-speech of functional words in shortest patn in the dependency tree between *PRED* and *NP*.

NP and *PRED* stand for a bunsetsu-chunk of a candidate antecedent and a bunsetsu-chunk of a predicate which has a target zero-pronoun respectively. The features marked with an asterisk are used during intra-sentential zero-anaphora resolution. The feature marked with two asterisks is used during inter-sentential zero-anaphora resolution.

plore the candidate reduction ratio of each cache model as well as its coverage, i.e. how often each cache model retains correct antecedents (Section 6.2). We also evaluate the performance of both antecedent identification on inter-sentential zero-anaphora resolution (Section 6.3) and the overall zero-anaphora resolution (Section 6.4).

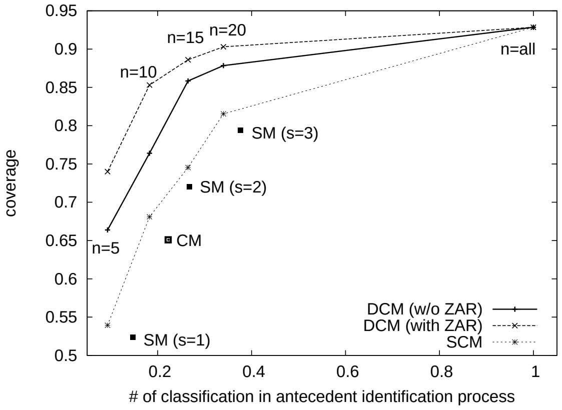
6.1 Data set

In this experiment, we take the ellipsis of nominative arguments of predicates as target zero-pronouns because they are most frequently omitted in Japanese, for example, 45.5% of the nominative arguments of predicates are omitted in the NAIST Text Corpus (Iida et al., 2007b).

As the data set, we use part of the NAIST Text Corpus, which is publicly available, consisting of 287 newspaper articles in Japanese. The data set contains 1,007 intra-sentential zero-pronouns, 699 inter-sentential zero-pronouns and 593 exophoric zero-pronouns, totalling 2299 zero-pronouns. We conduct 5-fold cross-validation using this data set. A development data set consists of 60 articles for setting parameters of inter-sentential anaphoricity determination, θ_{inter}, on overall zero-anaphora resolution. It contains 417 intra-sentential, 298 inter-sentential and 174 exophoric zero-pronouns.

6.2 Evaluation of the caching mechanism

In this experiment, we directly compare the proposed static and dynamic cache models with the heuristic methods presented in Section 2. Note that the salience information (i.e. the function *update-SalienceInfo*) in the dynamic cache model is disregarded in this experiment because its performance crucially depends on the performance of the zero-anaphora resolution model. The performance of the cache model is evaluated by *coverage*, which is a percentage of retained antecedents when appearing zero-pronouns refer to an antecedent in a preceding sentence, i.e. we evaluate the cases of inter-sentential anaphora resolution.



CM: centering-based cache model, SM: sentence-based cache model, SCM: static cache model, DCM (w/o ZAR): dynamic cache model disregarding *updateSalienceInfo*, DCM (with ZAR): dynamic cache model using the information of correct zero-anaphoric relations, n: cache size and s: # of sentences.

Figure 4: Coverage of each cache model

As a baseline, we adopt the following two cache models. One is the Centering-derived model which only stores the preceding '*wa*' (topic)-marked or ‘*ga*’ (subject)-marked candidate antecedents in the cache. It is an approximation of the model proposed by Nariyama (2002) for extending the local focus transition defined by Centering Theory. We henceforth call this model the *centering-based cache model*. The other baseline model stores candidates appearing in the N previous sentences of a zero-pronoun to simulate a heuristic approach used in works like Soon et al. (2001). We call this model the *sentence-based cache model*. By comparing these baselines with our cache models, we can see whether our models contribute to more efficiently storing salient candidates or not.

The above dynamic cache model retains the salient candidates independently of the results of antecedent identification conducted in the preceding contexts. However, if the zero-anaphora resolution in the current utterance is performed correctly, it will be available for use as information about the recency of candidates and the anaphoric chain of each candidate. Therefore, we also investigate whether correct zero-anaphora resolution contributes to the dynamic cache model or not. To integrate zero-anaphora resolution information, we create training instances of the dynamic cache model by updating the recency using the function ‘*updateSalienceInfo*’ shown in Figure 2 and also using an additional feature, CHAIN_NUM, defined in Table 1.

The results are shown in Figure 4[7]. We can see the effect of the machine learning-based cache models in comparison to the other two heuristic models. The results demonstrate that the former achieves good coverage at each point compared to the latter. In addition, the difference between the static and dynamic cache models demonstrates that the dynamic one is always better then the static. It may be this way because the dynamic cache model simultaneously retains global focus of a given text and the locally salient entities in the current discourse.

By comparing the dynamic cache model using correct zero-anaphora resolution (denoted by DCM (with ZAR) in Figure 4) and the one without it (DCM (w/o ZAR)), we can see that correct zero-anaphora resolution contributes to improving the caching for every cache size. However, in the practical setting the current zero-anaphora resolution system sometimes chooses the wrong candidate as an antecedent or does not choose any candidate due to wrong anaphoricity determination, negatively impacting the performance of the cache model. For this reason, in the following two experiments we decided not to use zero-anaphora resolution in the dynamic cache model.

6.3 Evaluation of inter-sentential zero-anaphora resolution

We next investigate the impact of the dynamic cache model shown in Section 4.1 on the antecedent identification task of inter-sentential zero-anaphora resolution altering the cache size from 5 to the number of all candidates. We compare the following three cache model within the task of inter-sentential antecedent identification: the centering-based cache model, the sentence-based cache model and the dynamic cache model disregarding *updateSalienceInfo* (i.e. DCM (w/o ZAR) in Figure 4). We also investigate the computational time of the process of inter-sentential antecedent identification with each cache model altering its parameter [8].

The results are shown in Table 3. From these results, we can see the antecedent identification model using the dynamic cache model obtains almost the same accuracy for every cache size. It indicates that if the model can acquire a small number of the most salient discourse entities in the current discourse, the model achieves accuracy comparable to the model which searches all the preceding discourse entities, while drastically reducing the computational time.

The results also show that the current antecedent identification model with the dynamic cache model does not necessarily outperform the model with the baseline cache models.

For example, the sentence-based cache model using the preceding two sentences (SM (s=2)) achieved an accuracy comparable to the dynamic cache model with the cache size 15 (DCM (n=15)), both spending almost the same computational time. This is supposed to be due to the limited accuracy of the current antecedent identification model. Since the dynamic cache models provide much better search spaces than the baseline models as shown in Figure 4, there is presumably more room for improvement with the dynamic cache models. More investigations are to be concluded in our future

[7] Expressions such as verbs were rarely annotated as antecedents, so these are not extracted as candidate antecedents in our current setting. This is the reason why the coverage of using all the candidates is less than 1.0.

[8] All experiments were conducted on a 2.80 GHz Intel Xeon with 16 Gb of RAM.

Table 3: Results on antecedent identification

model	accuracy	runtime	coverage (Figure 4)
CM	0.441 (308/699)	11m03s	0.651
SM(s=1)	0.381 (266/699)	6m54s	0.524
SM(s=2)	0.448 (313/699)	13m14s	0.720
SM(s=3)	0.466 (326/699)	19m01s	0.794
DCM(n=5)	0.446 (312/699)	4m39s	0.664
DCM(n=10)	0.441 (308/699)	8m56s	0.764
DCM(n=15)	0.442 (309/699)	12m53s	0.858
DCM(n=20)	0.443 (310/699)	16m35s	0.878
DCM(n=1000)	0.452 (316/699)	53m44s	0.928

CM: centering-based cache model, SM: sentence-based cache model, DCM: dynamic cache model, n: cache size, s: number of the preceding sentences.

work.

6.4 Overall zero-anaphora resolution

We finally investigate the effects of introducing the proposed model on overall zero-anaphora resolution including intra-sentential cases. The resolution is carried out according to the procedure described in Section 4. By comparing the zero-anaphora resolution model with different cache sizes, we can see whether or not the model using a small number of discourse entities in the cache achieves performance comparable to the original one in a practical setting.

For intra-sentential zero-anaphora resolution, we adopt the model proposed by Iida et al. (2007a), which exploits syntactic patterns as features that appear in the dependency path of a zero-pronoun and its candidate antecedent. Note that for simplicity we use bag-of-functional words and their part-of-speech intervening between a zero-pronoun and its candidate antecedent as features instead of learning syntactic patterns with the Bact algorithm (Kudo and Matsumoto, 2004).

We illustrated the recall-precision curve of each model by altering the threshold parameter of intra-sentential anaphoricity determination, which is shown in Figure 5. The results show that all models achieved almost the same performance when decreasing the cache size. It indicates that it is enough to cache a small number of the most salient candidates in the current zero-anaphora resolution model, while coverage decreases when the cache size is smaller as shown in Figure 4.

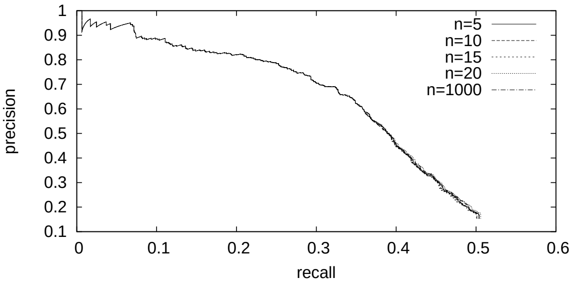


Figure 5: Recall-precision curves on overall zero-anaphora resolution

7 Conclusion

We propose a machine learning-based cache model in order to reduce the computational cost of zero-anaphora resolution. We recast discourse status updates as ranking problems of discourse entities by adopting the notion of caching originally introduced by Walker (1996). More specifically, we choose the N most salient candidates for each sentence from the set of candidates appearing in that sentence and the candidates which are already in the cache. Using this mechanism, the computational cost of the zero-anaphora resolution process is reduced by searching only the set of salient candidates. Our empirical evaluation on Japanese zero-anaphora resolution shows that our learning-based cache model drastically reduces the search space while preserving accuracy.

The procedure for zero-anaphora resolution adopted in our model assumes that resolution is carried out linearly, i.e. an antecedent is independently selected without taking into account any other zero-pronouns. However, trends in anaphora resolution have shifted from such linear approaches to more sophisticated ones which globally optimize the interpretation of all the referring expressions in a text. For example, Poon and Domingos (2008) has empirically reported that such global approaches achieve performance better than the ones based on incrementally processing a text. Because their work basically builds on inductive logic programing, we can naturally extend this to incorporate our caching mechanism into the global optimization by expressing cache constraints as predicate logic, which is one of our next challenges in this research area.

References

C. Aone and S. W. Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of 33th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 122–129.

M. Asahara and Y. Matsumoto, 2003. *IPADIC User Manual*. Nara Institute of Science and Technology, Japan.

B. Baldwin. 1995. *CogNIAC: A Discourse Processing Engine*. Ph.D. thesis, Department of Computer and Information Sciences, University of Pennsylvania.

P. Denis and J. Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669.

B. J. Grosz and C. L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204.

B. J. Grosz, A. K. Joshi, and S. Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

U. Hahn and M. Strube. 1997. Centering in-the-large: computing referential discourse segments. In *Proceedings of the 8th conference on European chapter of the Association for Computational Linguistics*, pages 104–111.

R. Iida, K. Inui, H. Takamura, and Y. Matsumoto. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the 10th EACL Workshop on The Computational Treatment of Anaphora*, pages 23–30.

R. Iida, K. Inui, and Y. Matsumoto. 2005. Anaphora resolution by antecedent identification followed by anaphoricity determination. *ACM Transactions on Asian Language Information Processing* (*TALIP*), 4(4):417–434.

R. Iida, K. Inui, and Y. Matsumoto. 2007a. Zero-anaphora resolution by learning rich syntactic pattern features. *ACM Transactions on Asian Language Information Processing* (*TALIP*), 6(4).

R. Iida, M. Komachi, K. Inui, and Y. Matsumoto. 2007b. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceeding of the ACL Workshop 'Linguistic Annotation Workshop'*, pages 132–139.

H. Isozaki and T. Hirao. 2003. Japanese zero pronoun resolution based on ranking rules and machine learning. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 184–191.

T. Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142.

M. Kameyama. 1986. A property-sharing constraint in centering. In *Proceedings of the 24th ACL*, pages 200–206.

T. Kudo and Y. Matsumoto. 2004. A boosting algorithm for classification of semi-structured text. In *Proceedings of the 2004 EMNLP*, pages 301–308.

A. McCallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*, pages 79–84.

J. F. McCarthy and W. G. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1050–1055.

S. Nariyama. 2002. Grammar for ellipsis resolution in japanese. In *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 135–145.

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th ACL*, pages 104–111.

H. Poon and P. Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659.

K. Seki, A. Fujii, and T. Ishikawa. 2002. A probabilistic method for analyzing japanese anaphora integrating zero pronoun detection and resolution. In *Proceedings of the 19th COLING*, pages 911–917.

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

L. Z. Suri and K. F. McCoy. 1994. Raft/rapr and centering: a comparison and discussion of problems related to processing complex sentences. *Computational Linguistics*, 20(2):301–317.

V. N. Vapnik. 1998. *Statistical Learning Theory*. Adaptive and Learning Systems for Signal Processing Communications, and control. John Wiley & Sons.

M. Walker, M. Iida, and S. Cote. 1994. Japanese discourse and the process of centering. *Computational Linguistics*, 20(2):193–233.

M. A. Walker. 1996. Limited attention and discourse structure. *Computational Linguistics*, 22(2):255–264.

X. Yang, G. Zhou, J. Su, and C. L. Tan. 2003. Coreference resolution using competition learning approach. In *Proceedings of the 41st ACL*, pages 176–183.

X. Yang, J. Su, J. Lang, C. L. Tan, T. Liu, and S. Li. 2008. An entity-mention model for coreference resolution with inductive logic programming. In *Proceedings of ACL-08: HLT*, pages 843–851.