辞書式順序を持つペナルティによるゼロ代名詞解消

磯崎秀樹[†] 賀沢秀人[†] 平尾努[†]

日本語では，主語や目的語などが省略されることが多く，これらの省略はゼロ代名詞と呼ばれる．機械翻訳や質問応答などのシステムでは，ゼロ代名詞の解消，つまり，何が省略されているかの推定が性能向上につながると期待できる．他の自然言語処理タスクと同様，これまでは経験則に基づくアプローチが主であったが，最近，機械学習によるアプローチが注目されている．しかし，高品質な訓練データを大量に準備することは難しい．そこで本論文では，経験則と機械学習の効果的な組合せ方法を提案する．まず，照応解析の機械学習が，通常の分類学習より困難な複数インスタンス学習の一種であることを指摘し，学習を簡単化するために，経験則を導入する．既存の複数の経験則を理解しやすい形で統合するために，ペナルティの辞書式順序を定義し，実験により，選択制限と属性共有を重視した辞書式順序が，SVM に匹敵する性能を出せることを示す．そして同一表記で出現位置の異なる候補が複数ある場合に，その中で辞書式順序で一番条件の良い候補だけを使うと，機械学習の性能が向上することが実証された．さらに，最良の候補を機械学習で選択するさいに，SVM の 3 つのバリエーションの中で,「優先度学習」と呼ばれる手法が安定して高い性能を示すことが判明した．

Japanese Zero Pronoun Resolution Based on Lexicographical Ordering of Penalties

Hideki Isozaki,[†] Hideto Kazawa[†] and Tsutomu Hirao[†]

In Japanese, subjects and objects in a sentence are often omitted and these omissions are called *zero pronouns*. Zero pronoun resolution is expected to be useful for machine translation and question answering systems. Just like other natural language processing tasks, conventional studies used heuristic approaches, but recently, machine learning approach is becoming popular. However, it is difficult to prepare a large amount of training data. In this paper, we propose a method that combines heuristic ranking rules and machine learning. First, we show that anaphora resolution is a kind of Multiple-Instance Learning. In order to alleviate the problem, we introduce comprehensible lexicographical orderings of candidates based on penalties given by conventional heuristic rules. According to our experiments, simple orderings that emphasize *selectional retriction* and *property-sharing constraint* is comparable to SVM. Since some candidates appear repeatedly in a document, we applied the lexicographical ordering method to pick up only the best context for each candidate. Then, the machine learning methods gave better results. Furthermore, among three variations of Support Vector Machines, preference learning showed stable and good performance.

1. はじめに

日本語では，主語や目的語などが省略されることが多く，これらの省略はゼロ代名詞と呼ばれる．機械翻訳の研究分野では，ゼロ代名詞が何を指しているかによって訳文が変わることがあり，以前からゼロ代名詞の解消が課題である．たとえば日英翻訳なら，省略されている主語が女性のとき she を，男性であれば he を補うなどの処理が考えられる．ゼロ代名詞の解消を避けるため，受動態にして主語を明示しない方法が利用されることもある[20]が，受動態の文が連続して不自然な訳になってしまうという問題がある．

質問応答でも，ゼロ代名詞の解消が役立つことが期待できる．NTCIR QAC☆などの，情報検索をベースとする一問一答形式の質問応答システムでは，検索で得られた文書から解答候補を抽出してランキングする．これらのシステムの多くは，たとえば「『伊豆の踊り子』の作者は誰？」のような質問を受け付け，質問文から抽出した単語（検索語）を使って，大量の文書から関連する文書の検索を行う．そして上位の文書から，解答の候補となる文字列（この場合は人名）を抜き出

† 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation

☆ http://www.nlp.is.ritsumei.ac.jp/qac/qac2/index-j.htm

し，パターンや検索語の密度などによって解答候補のスコアリングを行う．大量の文書があれば，その中に「伊豆の踊り子」と「川端康成」がごく近くに現れる文書が含まれていて,「川端康成」という正解が高いスコアで得られる．しかし,「川端康成」の経歴が書かれている文書などでは，主語「川端康成」がしばしば省略されているので,「伊豆の踊り子」からは遠く,「川端康成」は高いスコアを得られないことがある．このような場合に，主語を自動的に補うことができれば，システムが正解を見つけやすくなる．

本論文では，ゼロ代名詞解消の機械学習が，通常の分類学習とは異なる「複数インスタンス学習」「優先度学習」という面を持つことを指摘し，その解決方法を提案する．まず，前者の問題を緩和するために，既存の経験則のそれぞれをペナルティ化し，それらを組み合わせて候補の辞書式順序を定義する．この辞書式順序は，従来の経験則がそのまま利用できるため，理解しやすいという長所を持つ．また，後者の問題に対処するため，通常，分類学習に用いられている SVM と，SVM をベースにした優先度学習の性能を比較する．

我々の実験結果によれば，理解しやすい辞書式順序で，SVM と同程度の性能を達成できることが確認できた．さらに，辞書式順序を機械学習に組み込むことで，精度を向上させることができることが判明した．

1.1 関 連 研 究

これまでにも，日本語ゼロ代名詞の研究は数多くなされているが，小規模な例を用いた，内省に基づく研究[13),25),27)]，や，省略されているもののほとんどが話し手か聞き手である対話を対象とした研究[26)]が多い．一方，新聞記事は長い文章が多く，ゼロ代名詞が指す先行詞の候補が非常に多くなる．そのため，精度良く解消することは難しい．これまでにも新聞記事を対象とした研究はあるが，特定分野のリード文に制限していたり[18)]，文内の照応に限っていたり[4),19)]，組織名に限っていたり[1)]と，候補の数が少なくなるような制限を設けていることが多い．したがって，もっと一般的な状況で精度良く解消できる手法が望まれる．

ゼロ代名詞解消には，経験則に基づくアプローチと，機械学習によるアプローチがあるが，いずれの手法でも候補をフィルタリングするのに「選択制限」がしばしば利用される．これは「書く」という動詞の主語は人間であるというような知識である．経験則によるアプローチでは,「センタリング理論[6)]」が有名である．Walker ら[25)]は，日本語のためのランキングを提案しており，Kameyama[13)]は，ゼロ代名詞と候補の間での「属性共有」の重要性を主張している．Okumura ら[22)]は，複文における接続助詞の役割について実験を行っている．村田ら[17)]は，照応全般について研究を行い，多数の素性を考慮した複雑な手法を提案している．しかし，それでも十分な精度を達成することは難しい．経験則の数が多くなるにつれ，整合性を維持することも難しくなっていく．

そこで，近年はタグつきコーパスからの機械学習によるアプローチが研究されている．ゼロ代名詞解消は難しい課題であり，少ない素性だけで高性能を達成することはできない．そこで，素性の数が多くても過学習しにくい SVM[2)]が利用されてきている[8)]．我々も SVM を利用したゼロ代名詞解消システムを作成しているが，単純に適用しただけではなかなか性能が上がらないことが分かってきた．

1 つの理由は，機械学習によるアプローチで必要となる質の良い大量のタグつきコーパスを用意することが難しいことである．そこで，これまでの主な経験則を統合し，タグつきコーパスの情報を最大限生かせるように機械学習することを目指す．

また，SVM のような既存の分類学習が，ゼロ代名詞解消には必ずしも適していないのではないかとも考えられる．この点について，以下で説明を行う．

1.2 ゼロ代名詞解消に適した機械学習とは

ゼロ代名詞解消の基本的な方法は，その文書の先頭から解消したいゼロ代名詞の位置までに出現する名詞句を集めて候補とし，その中から各ゼロ代名詞にふさわしい候補を選び出すというものである．ゼロ代名詞のような照応解析の機械学習は，以下のような 2 つの観点から，通常の分類学習と異なると考えられる．

- 複数インスタンス学習
- 優先度学習

1.2.1 複数インスタンス学習

通常の分類学習では，1 つ 1 つの素性ベクトルに対して正例・負例のラベルが付与されているが，ゼロ代名詞解消のような照応解析では，やや事情が異なる．

文章中では，1 つの事物を指すのに，複数の表現が使われる．たとえば，ある特定の人物について，最初は「男」と書き，次の文では「犯人」と書いてあった場合，これらは別の候補として別の素性ベクトルで表せる．同じ表現であっても，文書中に何度も出現することがあるので，それぞれの出現位置ごとに，前後の文脈を見て異なる素性ベクトルにすることができる．

それに対し，訓練データにおいて，あるゼロ代名詞の正解の先行詞として与えられるのは,「山田首相」のような文字列だけであり,「山田首相」に対応する素性ベクトルは，すべて正例と見なすことになる．その中

には，そのゼロ代名詞の先行詞の正例として学習に使うには不適切な候補も多いはずである．

たとえば，「山田首相は昨日午後，….山田首相を暗殺した犯人は …」という文脈では，2 番目の「山田首相」は目的格でかつ連体修飾節の中である．このような候補は，過去の研究では先行詞になりにくいとされてきている．この直後に「山田首相」を先行詞とするゼロ代名詞があった場合，機械学習は，この目的格でかつ連体修飾節の中の候補を正例として学習することになる．つまり，正例といいながら，負例の領域に深く入り込んでいるものが多いと考えられる．このような状況で機械学習を行っても，良い性能が得られないことは想像できよう．

一般に，1 つの実体が複数のインスタンス（素性ベクトル）を持ち，各実体が少なくとも 1 つの正例インスタンスを持つか否かは分かるが，どのインスタンスが正例かは分からない，という設定での学習問題は，本当の正例が分からないため，通常の分類学習に比べて難しく，機械学習の分野で複数インスタンス学習[3]と呼ばれている．1 つの実体に対応する情報をまとめて 1 つのベクトルにすることも考えられるが，どうまとめるかが問題となる．

飯田ら[8]は，照応関係にある候補の中で，ゼロ代名詞に一番近い候補だけを利用して学習している．これは，先行詞は近くにあることが多いという経験則により，負例領域にある正例をできるだけ減らそうとしていると考えられる．近さ以外の経験則を用いることも考えられる．河原ら[14]は，文の構造を考えた順序を考えている．本論文では，既存の経験則を組み合わせた辞書式順序を提案する．

1.2.2 優先度学習

決定木学習や SVM などの分類学習は，正例か負例かの判定（分類）基準を学習する手法であって，候補の優劣が比較できるような点数づけを学習するようにデザインされたものではない．

ゼロ代名詞解消で，各候補に対して独立に正例（正解）・負例（不正解）を判定すると，すべての候補が不正解と判断されたり，複数の候補が正解と判断されたりすることがある．このような場合に候補をどう選択するかが問題である．したがって，候補の優劣を数値化できる学習方法の方が，ゼロ代名詞の解消には適していると考えられる．

分類のための機械学習手法でも，内部では何らかの点数づけを行っていることがある．たとえばリニアカーネルを用いた SVM の場合，素性の値の線形結合で点数づけを行い，その値の符号によって 2 値分類を行っている．この符号にする前の実数値を候補のスコアと見なすことができる．この値は分類のための数値であって，大小比較するためのものではないが，点数を簡単に得られる．

Yang ら[28]や飯田ら[8]は，分類器を分類器としてそのまま使う方法として，2 つの候補のどちらが良いかを分類器に判断させる方法を提案している．これらは，2 つの候補 A，B の特徴を比較し，その情報を 1 つのベクトルにして，「左の候補（A）が良い」というクラスと「右の候補（B）が良い」というクラスに分類するという方法である．Yang らの手法は，勝った回数の一番多い候補を選ぶのに対し，飯田らの手法はトーナメント（勝ち抜き戦）である．これらの方法は，分類のための機械学習手法をそのまま利用できる一般的な方法であるが，分類器の個々の出力は，個々の候補に点をつけているわけではなく，推移律の保証がない．つまり，A>B（A の方が B より良い），B>C と判定されても，A<C と判定される可能性がある．

一方，Herbrich ら[7]や Joachims[12]は，SVM をベースにして候補の適切性を数値化する関数を学習する優先度学習（Preference Learning．以下では **PL** と略す）を提案している．こちらの方法は，まさに個々の候補の点数づけをするために考案された方法であり，本論文では，こちらのアプローチも試みる．現在，Joachims の公開しているツール SVM-light[11]☆には，この機能が備わっている．

自然言語処理では，複数の候補を採点したり，ランキングしたりする課題が多いので，優先度学習が SVM などの他の機械学習手法と比べて高性能であれば利用したい．本論文のもう 1 つの目的は，優先度学習の有効性をゼロ代名詞解消という現実のデータで検証することである．

図 1 を用いて SVM と PL の学習データの作り方を説明しよう．SVM では，各ゼロ代名詞の正解候補を正例（+1），それ以外の候補を負例（−1）として訓練データを作る．一方，PL や Yang らや飯田らの手法では，解消したいゼロ代名詞ごとに正解 X と不正解候補 Y のペア $X > Y$ のリストを作り，これを集めて訓練データとする．SVM は，高次元空間において，正例グループと負例グループの間に境界面を引こうとするのに対して，PL は，各ペアの正解候補のスコアと不正解候補のスコアの差がなるべく大きくなるような評価関数を作ろうとする．

リニアカーネルを用いた場合，SVM も PL も

☆ http://svmlight.joachims.org/

	ゼロ代名詞	正解	不正解候補
元のデータ：	Z1	C_1	W_1, W_2, W_3
	Z2	C_2, C_3	W_4, W_5

SVM のデータ：$(+1, C_1), (-1, W_1), (-1, W_2), (-1, W_3), (+1, C_2), (+1, C_3), (-1, W_4), (-1, W_5)$

PL のデータ：$(C_1 > W_1), (C_1 > W_2), (C_1 > W_3), (C_2 > W_4), (C_2 > W_5), (C_3 > W_4), (C_3 > W_5)$

図 1 SVM と優先度学習の学習データの違い
Fig. 1 Difference of training data between SVM and Preference Learning.

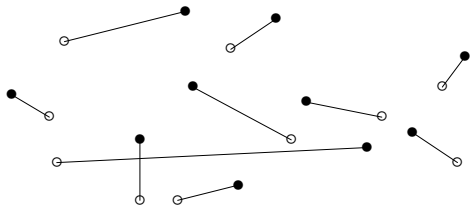


図 2 優先度学習データの分類（● > ○）
Fig. 2 Classification of preference data.

$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}(+b)$ という形の関数を求めることになる．定数項 b はスコアの大小比較には影響を与えないので，結局，SVM で求めた $\boldsymbol{w}$ と PL で求めた $\boldsymbol{w}$ の向きの違いが，両者の成績の違いにつながる．

図 2 のように正解（●）と不正解（○）が複雑に入り混じっているデータの場合，正解と不正解の間に単純な境界面を引くことはできないが，PL であれば，● > ○ という対応関係により，上にある方が下にある方よりスコアが高い，という単純な評価関数を学習できる．著者の知るかぎり，SVM と PL の $\boldsymbol{w}$ が，どのような状況でどの程度の性能の差になるかは，まだあまり明確になっていない．

2. 方 法 論

機械学習を用いると，人手で複雑なプログラムを書かなくてよくなることが理想的である．しかし現状では，ゼロ代名詞解消という問題の難しさに比べて，入手できる訓練データの量があまりに少なく，経験則をまったく使わないで高精度を出すことは難しい．また，複数インスタンス学習の緩和という観点からも，良い経験則は有用と考えられる．そこで本論文では，機械学習と経験則を効果的に組み合わせる手法を提案する．経験則による結果は，機械学習の素性に入れるだけでなく，同じ実体を指す候補を絞るのにも利用できる．

2.1 対象とするゼロ代名詞のデータ

対象とするゼロ代名詞は，他の多くの研究と同様に，すでに出現した名詞句（「こと」や「もの」などの非自立名詞は除く）を指しているゼロ代名詞だけを解消の対象とする．ゼロ代名詞の定義は必ずしも明確ではない．本論文では，関らのデータ[23),24)] に記録されているゼロ代名詞を前提とし，ゼロ代名詞がすでに検出され，その文法的な格が決定されていると仮定して，各ゼロ代名詞が何を指しているかの推定だけを課題とする．ゼロ代名詞の格は，対応する格助詞「が」「を」「に」で表す．この格助詞を「ゼロ代名詞の格助詞」と呼び，以下では **ZP** と略す．また着目したゼロ代名詞を Z で表し，その先行詞の候補の 1 つを C で表す．従来研究と同様に，先行詞候補として，そのゼロ代名詞より前に出現した名詞句だけを考える．C と同じ文節で，C の直後の単語を **CP** で表す．**CP** は多くの場合，助詞か句読点である．たとえば,「東京には」の場合，**CP** は「に」であるが,「東京.」の場合，**CP** は「.」である．

関らのデータは京都大学テキストコーパス 2.0 に基づいており，一般記事 **General** の 30 記事と社説記事 **Editorial** の 30 記事からなる．本論文ではこのテキストを ChaSen 2.2.9☆ と CaboCha 0.34☆☆ で解析し直したものを用いる．これは，京大コーパスで用いられている助詞の分類がやや粗く，いくつかの誤りの原因になっていると思われたためである．

関らの正解に含まれる明らかな誤りや別解の不備は修正した．さらに，引用文中のゼロ代名詞は，話者や聴者が誰かなど，地の文のゼロ代名詞とは別の扱いが必要である．これらを地の文と一緒に処理すると混乱するので，引用文中のゼロ代名詞は対象としない．

この処理の結果，ゼロ代名詞の数は**表 1** のようになった．表から分かるように，85%以上が「ガ格」である．したがって,「ガ格」を中心にして実験を行う．

関らは，正しく検出できたゼロ代名詞だけを対象とした解消実験を行っており，指示している先行詞が正しく同定されたゼロ代名詞の割合（以下，精度）で評価している．General が 355 個のゼロ代名詞に対して 54.0%[24)]，あるいは 404 個に対し 50.7%[23)] であった．一方，Editorial は 498 個に対して 39.8%[24)] であった．本論文では，ゼロ代名詞の検出をしないので対象が増えるが，一方で引用文中のゼロ代名詞を対象としていないので，General のゼロ代名詞の数は減っ

☆ http://chasen.aist-nara.ac.jp/
☆☆ http://cl.aist-nara.ac.jp/~taku-ku/software/cabocha/

表 1 対象とするコーパスに含まれるゼロ代名詞の数
Table 1 Number of zero pronouns in our corpus.

記事種	記事数	文数	ガ格	ヲ格	ニ格	全格
General	30	417	295	22	29	346
Editorial	30	861	443	35	37	514
計	60	1,278	738	57	66	860

て 346 個，Editorial は増えて 514 個となっている．Editorial の方が難しい理由として，1 記事あたりの文数が考えられる．General は平均 13.9 文，Editorial は 28.7 文であり，Editorial の方が 2.1 倍多い．それにともない，先行詞候補の数が平均 1.6 倍になっていて，これが難しい原因と考えられる．

システムの出力を採点する場合，以下の点に注意しなければならない．1 つの文章の中では，同じ事物について話をしていることが多いので，ゼロ代名詞解消においても，同じ候補が正解として続くことが多い．そのため，直前のゼロ代名詞の正解が分かっていると正解しやすい．今回用いたデータでは，ガ格の 42%のゼロ代名詞が，直前のガ格のゼロ代名詞と同じ正解である．したがって，最初のちょっとした判断の誤りが，そのあとの結果に大きな影響を及ぼすことになる．

関らは，ガ格，ヲ格，ニ格のゼロ代名詞を対象とし，システムはそれまでのゼロ代名詞の正解を利用できないという現実的な設定で実験を行っている．一方，飯田ら[8] は，ゼロ代名詞をガ格に絞り，それまでのゼロ代名詞の正解を含む照応関係をすべて利用できるとして各ゼロ代名詞の解消の精度の評価を行っている．

ここでは，それまでのゼロ代名詞の正解をシステムが利用できるという設定での精度を「**独立評価**」，利用できないという設定での精度を「**一括評価**」と呼ぶ．

過去の研究[4),22)] では，長い複雑な文をあらかじめ**接続助詞など**の場所で自動的に分割しておくことが行われている．そこで，本論文でも，あらかじめ自動的に分割を行っておく．これで得られる単純な文を「**分割後の文**」と呼ぶ．過去の文献を参考に，以下の場所で文を切る．品詞，活用形は ChaSen の体系に従う．

つつ，ながら，まま，たり，ので，のに，けれど，ば，から，ため，後，と（接続助詞），て，ても，たら，なら，が（接続助詞），ながら，つつ，から，けれど，のに，ので，と（格助詞），て，ことは，のは，とき，時，連用形，形容詞連用テ接続，ず，ずに

ただし，連体修飾節に接続助詞などが現れた場合，そこでは文を切らない．たとえば，以下の例では，最初の例は「買って」で切るが，2 番目の例は切らない．

- 彼女はその本を買って彼に売った．
- 彼女は彼が買って大事にしていた本を売った．

2.2 解消処理の概要

以上の実験設定のもとで，各ゼロ代名詞の指す候補の様々な素性に着目して出現位置ごとに候補をベクトル化し，正解かどうかのラベルを与える．

本論文のゼロ代名詞解消システムは，機械学習を用いるか否かにかかわらず，すべて以下の流れにそって文書を先頭から読みながら処理をすすめる．

- 新しい名詞句があると，その文脈での素性を調べ，「候補リスト」に加える．(付録 A.1) ただし，「同氏」などの人を指す表現は，直前の人名で置き換える．「同社」は直前の組織名で置き換える．人名や組織名の抽出には，SVM に基づく固有表現抽出[10] を用いる．それ以外の「同 +X」のパターン（「同省」「同町」など）は，X を含む一番近い候補（「文部省○○課」や「山田町大字△△」）の，先頭から X までの文字列で置き換える．
- ゼロ代名詞の位置に達すると，その時点の「候補リスト」を，指定された「辞書式順序」により良い順にソートする．機械学習を用いない場合は，その先頭要素を出力する．機械学習を用いる場合は，まず，表記が同じ候補群は，その中で最上位の候補だけにしぼり，その順位で何位かの情報を素性ベクトルに加える．
- ゼロ代名詞の位置に先行詞（独立評価では正解，一括評価では最良の候補）が書かれていると見なして，候補リストに加え，同じ表記の候補は削除する．

しかし，予備実験を行ったところ，そのままでは学習に時間がかかりすぎたり，精度があまり良くなかったりすることが分かったので，可能性の低い遠くの候補（距離が分割後の文の数で D 文を超える）をはずすことにした．ただし，ゼロ代名詞として出現したものを候補に含めることにより，実際には数十文離れている候補も，重要なものはカバーできると予想される．

さらに，ヲ格やニ格のゼロ代名詞を対象とする場合，以下のような処理を加える．1 つの用言の複数の格に同じものが入ることはまれである，という経験則[17)] がある．たとえば，「A 氏に（Z が）電話した.」という文の Z として「A 氏」は普通あてはまらない．そこで，ゼロ代名詞 Z の候補 C がすでに同じ用言を修飾している場合は，C を Z の候補としては採用しない．同様に，ある用言が複数のゼロ代名詞を持つとき，まず，ガ格のゼロ代名詞を最初に解消し，それで選ばれた候補は，ヲ格，ニ格の候補としては採用しない．

2.3 経験則の統合

ゼロ代名詞解消について，これまでに以下のような経験則が提案されている．ここでは，これらを統合したランキング方式について考える．

(1) **選択制限**[27),30)]：ゼロ代名詞が「食べる」の主語であれば，それは人か動物であろう，と推定できる．本論文では，日本語語彙大系[9)] の構文大系を用いてこのチェックを行う（付録 A.2）．

(2) **属性共有**[13)]：ある用言の主語がゼロ代名詞であれば，その先行詞は，それまでのどこかの文の主語である可能性が高い．同様に，ある動詞の目的語がゼロ代名詞であれば，やはり先行詞も目的語である可能性が高い．

(3) **距離**[24)]：多くの先行詞はゼロ代名詞の近くにある．ここでは，先行詞候補からゼロ代名詞までの「分割後の文」の数とする．

(4) **連体修飾節**中の候補の回避：ゼロ代名詞は通常，連体修飾節中の名詞句を指さない[4)]．

(5) **センタリング理論**によるランキング[25)]：(トピック > empathy > 主語 > 間接目的語 > 直接目的語 > その他)．

さて，上記の経験則の統合方法が問題となる．各経験則から得られるペナルティを線形結合してスコアをつけるというのが標準的な手法であろう．しかし，本論文で利用する SVM は，まさにその線形結合を学習する優れた方法であり，それを超える係数決定法を考えるのは難しい．そこで，別の方法を考える．

上記の各経験則は，それぞれ，候補集合を好ましさによって分割する．そこで，ある経験則で候補集合を分割して，空でない最良の部分集合を選び，それを別の経験則でさらに分割して，空でない最良の部分集合を選ぶ，ということを再帰的に繰り返すことで最良の候補を選ぶ方法を試す．それでも最後まで最良の候補が複数残ってしまった場合は，たとえば，候補の出現位置など，確実に一意に決まる基準で選ぶ．この再帰的な選択法は，これまでの研究で蓄積された経験則を生かす非常に簡単な方法であり，また，ある経験則が別の経験則より優先されることがはっきりしているので，理解しやすいという長所がある．

この方法は，各候補を以下のようなペナルティの数値ベクトルで表し，その数値ベクトルの中で，辞書式順序で最小の候補を選ぶという方法と等価である．

- 選択制限ペナルティ$\mathbf{v}$（violation）：用言の選択制限を満たしている候補は 0，破っている候補は 1．
- 格一致のペナルティ$\mathbf{a}$（agreement penalty）：候補とゼロ代名詞の格が一致（$\mathbf{CP} = \mathbf{ZP}$）していれば 0，不一致なら 1．ただし，「は」と「も」の多くは主語を表すので，これらも「が」と見なす．
- 距離のペナルティ$\mathbf{d}$（distance）：候補 C とゼロ代名詞 Z の間にある（分割後の）文の数である．なお，$\mathbf{d}$ が上限値 D を超えた候補は候補リストからはずす．
- 連体修飾節ペナルティ$\mathbf{r}$（relative clause）：連体修飾節の外の候補は 0，中の候補は 1．ただし，連体修飾節はネストしたり，ゼロ代名詞を含んだりすることがある．先行詞の可能性が低いのは，ゼロ代名詞の位置ですでに終わっている連体修飾節の中の候補だけと思われるので，その場合のみ $\mathbf{r}$=1 とする．
- センタリング理論による格のペナルティ$\mathbf{s}$（salience）：$\mathbf{CP}$ が「は」のとき 0，「が」は 1，「に」は 2，「を」は 3，それ以外は 4．なお，*empathy* は，新聞にはほとんど出現しないため，考慮しない．

ただし，これらのペナルティだけでは，同点となる候補が存在する．そこで，タイブレークのため，以下のペナルティを加える．

- 候補番号 $\mathbf{i}$：文書の先頭から，候補名詞句が出現した順にふった番号で，$\mathbf{i} = 0, 1, 2, \ldots$．これをペナルティとするということは，最初に出てきた候補ほど重要である，という経験則を表している．なお，ゼロ代名詞の先行詞を解消した結果も新しい候補となるが，この新しい候補の番号は，ゼロ代名詞の正解先行詞の候補番号を引き継ぐ．

すると，各候補のペナルティが $(\mathbf{v}, \mathbf{a}, \mathbf{d}, \mathbf{r}, \mathbf{s}, \mathbf{i}) = (0, 0, 0, 2, 1, 2)$ のように得られる．これをそのまま用いて辞書式順序を作ると，$(0, 0, 0, 2, 1, 2) < (1, 0, 0, 0, 0, 0)$ などの順序関係が成り立つ．この順序で一番小さい候補を最良の候補と見なす．

辞書式順序において，各ペナルティをどのような順序に並べるかが問題である．$\mathbf{i}$ はタイブレーク用なので最後としても，残り 5 つの経験則を並べる順列は 5!=120 通りもあるので，ここでは過去の文献を参考に，調べる順列を絞り込む．まず，選択制限はしばしば候補が必ず守るべき条件として使われており，非常に重要である．したがって，$\mathbf{v}$ は最優先か，そうでないとしても，かなり上位に置けばよいと予想される．また，Okumura らの実験[22)] によれば，亀山の手法（$\mathbf{a}$）の方が Walker の手法（$\mathbf{s}$）より精度が高いので，$\mathbf{a}$ を $\mathbf{s}$ より優先する．つまり，$(\mathbf{v}, \ldots, \mathbf{a}, \ldots, \mathbf{s}, \ldots)$ というパターンの辞書式順序の成績が良いと期待される．そこで，この順序に近い辞書式順序を中心に調べる．

以下の実験では，予備実験の結果により距離の上限

表 2 辞書式順序による精度（ガ格のみ，%）
Table 2 Performance of lexicographical ordering methods (percentage of correctly resolved subjects).

順序	独立評価		一括評価	
	全記事	(General)	全記事	(General)
vadrs	**64.6**	73.2	57.3	66.1
avdrs	64.4	73.2	56.9	66.1
vadsr	64.2	72.9	56.9	65.8
avdsr	64.0	72.9	56.5	65.8
vards	62.9	**74.2**	**58.3**	**69.2**
avrds	62.7	**74.2**	58.0	**69.2**
vrads	61.5	73.9	57.6	68.8
d	41.2	44.7	28.7	30.5
(−i)	9.6	9.2	9.6	9.2

として D=5 を採用した．ガ格に限定した場合，精度の良い辞書式順序は，**表 2** のようになり，**v** や **a** が優先されている順序が上位を占めた．なお，表中では，$(\mathbf{v}, \mathbf{r}, \mathbf{a}, \mathbf{d}, \mathbf{s}, \mathbf{i})$ を vrads のように略記し，最後の **i** は省略している．

この表から分かるように，独立評価と一括評価で成績の良い順序は若干異なっている．独立評価では，**d** が **r** より優先されている辞書式順序の成績が良く，一括評価では，**r** が **d** より優先されている辞書式順序の成績が良い．**d** が優先されていると，直前のゼロ代名詞の正解を優先しやすいためではないかと推定される．逆に一括評価で大きく間違えないためには，一度間違えても正しい候補に戻せなくてはならない．

これらの経験則による処理は，学習が不要で，きわめて高速に実験を行えるため，これ以外にも様々な順序を試したが，**s** や **d** を優先させた辞書式順序は概して成績が悪かった．たとえば，全記事の独立評価の成績で，dvars は 48.5%，svadr は 46.1%であった．最後の **i** を **−i** にして近い候補を優先した場合，表 2 の上位の順序で，いずれも精度が 1.5%程度低下した．

a を **d** よりも優先した方が成績が良いので，格が一致しない近い候補より，遠くても格が一致する候補を優先する方が良い，といえる．

a と **s** はどちらも助詞の扱いであり，その最大の違いは，**s** が「は」を「が」より優先するのに対して，**a** は「は」を「が」と見なすという点である．そこで，**a** において「は」を「が」と見なさなかった場合の成績を調べると，vadrs は 64.6%から 52.3%へ，avdrs は 64.4%から 50.5%へ大幅に低下した．

逆に **s** の「は」と「が」のペナルティを同じにすると，svadr の成績は 65.7%に向上し，vadrs を超えた．逆に「は」と「が」の順位を入れ替えると，51.2%に下がったが，これは「は」を優先した場合より良い．このことから，**s** は「は」を「が」に比べて優先しすぎるのが問題であると思われる．その理由として考えられるのは，直前のガ格のゼロ代名詞である．実際，直前のゼロ代名詞の先行詞の正解が分からない一括評価にすると，独立評価で 65.7%に向上した svadr は 54.7%となり，表の vadrs や vards よりもスコアが低い．センタリング理論との整合性を保つには，ガ格のゼロ代名詞の先行詞の **CP** が，もともと「は」であったことを候補の素性として残すなどの改良が必要となるであろう．

表 3 辞書式順序（ガヲニ格・一括評価）と関らの成績との比較
Table 3 Comparison of lexicographical ordering methods and Seki et al.'s method.

	General		Editorial	
	分母	精度	分母	精度
関ら[24]	355	54.0%	498	39.8%
関ら[23]	404	50.7%		
vadrs	346	60.7%	514	47.5%
vards	346	63.3%	514	47.3%

ベースラインとして，文の区切りを無視して，ゼロ代名詞に一番近い候補を選ぶ辞書式順序（**−i**）と，d つまり（$\mathbf{d}, \mathbf{i}$）も試した．これによれば，（**−i**）は 9.6%と非常に悪いが，（$\mathbf{d}, \mathbf{i}$）は 41.2%もあり，**d** と **i** の組合せが重要なことが分かる☆．

なお，我々は関らのデータに手を加えている（2.1 節）ので，完全に公平な比較はできないが，これらの経験則の性能を判断する 1 つの参考データとして，関らと同様にガ格・ヲ格・ニ格の一括評価の結果を**表 3** に載せておく．オープンなテストではないとはいえ，利用した経験則がきわめて単純で，微妙なパラメータの調整ができないことを考えると，高い性能であるといえよう．

2.4 機械学習の手法

すでに述べたように，機械学習を適用する際には，ゼロ代名詞の位置において，候補リストに含まれる各先行詞候補を素性ベクトル $\boldsymbol{x}_i$ に変換し，その先行詞候補が正解のとき $y_i = +1$，不正解のとき $y_i = -1$ とする．PL では，$\boldsymbol{x}_{i.1}$ として正解の先行詞を，$\boldsymbol{x}_{i.2}$ には不正解の先行詞を用いて，$\boldsymbol{x}_{i.1} > \boldsymbol{x}_{i.2}$ というデータを生成する．いずれの方法も，実行時には，SVM あるいは PL の決定関数の出力をスコアとし，スコアの最大値を与える候補をゼロ代名詞の先行詞の推定結

☆（**−i**）の性能の悪さの 1 つの原因は，ゼロ代名詞が先行詞の **i** を引き継いでいることである．候補番号が古い先行詞は他の候補より不利になる．ゼロ代名詞に新しい候補番号を付与すると，精度は 23.2%に上昇する．この変更は，他の辞書式順序の成績を 0.5%ほど低下させる．

果として出力する．場合によっては，すべての候補が負になることもあり，それはSVMではすべての候補が先行詞ではないと判断されたことを意味するが，今回の実験では，先行詞がすでに現れているゼロ代名詞だけを対象としているので，その場合でも，最大スコアの候補をゼロ代名詞の先行詞とする．

SVMは，訓練データの誤分類に対するペナルティを表すパラメータCを持っており，Cが大きいほどトレーニングデータにおける誤分類に厳しくなる．ゼロ代名詞のデータでは，負例の数に比べて正例の数が圧倒的に少ない．このようなデータで学習をすると，正例を負例と判定する可能性が高くなる．Morikら[16]は，このようなアンバランスなデータに対して，正例と負例のCを以下のように変える方法を提案している．

正例のC = 負例のC × 負例の数/正例の数

ここでは，負例のCをユーザが指定し，正例のCは上記の式から自動的に計算する．この変更を加えると，SVMは負例の間違いより正例の間違いを問題視するようになる．そのため，正例と判断されやすくなる．これを本論文では**SVMJ**と略記し，SVMと同じ方法で先行詞を推定する．

なお，初期の実験では，決定木学習やガウシアンプライアありの最大エントロピー法も比較していたが，これらSVMベースの手法に比べると成績が悪かったため，本論文の実験には含めなかった．

なお，比較のため，飯田らの手法「センタリング素性を用いたトーナメントモデル」[8]を再実装した．飯田らの提案しているトーナメントモデルは，2つの候補の素性を1つのベクトルにエンコードし，どちらの候補が良いかを判定するSVMを学習する，という手法である．ただし，飯田らは，記事中に出現している名詞句間の照応関係がすべて分かっているデータを前提としている．たとえば記事中に現れる「男」と「犯人」が同一人物を指している，という情報が付与されている．そして，トーナメントを行う際に，照応関係にある候補のうち一番近い候補どうしを比較する．一方，我々のデータには，このような照応関係が付与されていないので，より難しいデータになっている．我々によるトーナメントモデルの再実装では，照応関係のかわりに，表層の一致を用いる．つまり,「犯人」という表記の候補が複数ある場合，その中で一番近い候補を用いる．

また，飯田らは，名詞と動詞の共起の良さの統計モデルを学習するのに，日経新聞11年分，毎日新聞9年分を利用している．我々の再実装では，そのかわりに毎日新聞14年分を用いた．

なお，トーナメントモデルは素性によらず適用できる枠組みである．そこで，トーナメントモデルによる学習を一般的に**TM**と書き，とくに飯田らの論文に書かれている素性を用いたものを**ITM**と略記する．

2.5 機械学習のための素性

ここでは，機械学習に用いる素性を説明する．まず，経験則でも利用した**v**，**r**，**a**はもともと0/1のブール変数で表せているので，そのまま機械学習の素性として利用する．

sは「**CP**=’が’」などの命題で表せるので，**CP**の値ごとにブール変数とする．一方，**d**は0からDまでの整数である．SVMはこれをそのまま素性の値とすることも可能であるが，値が大きいのでカーネルの値がこの素性にひきずられてしまうという問題がある．いくつかの回避策が考えられるが，これも,「$\mathbf{d} = 0$」などの命題に分けることでブール変数にする．

機械学習では多数の素性が利用できるので，ここではさらに，以下のような候補の素性を含める．

- 候補Cの最後の単語（主辞）の**品詞**とその日本語語彙大系[9]の**意味カテゴリ**：たとえば,「太郎」は固有名詞,「大統領」は普通名詞で人間である，など．なお，固有表現抽出の結果も用い，Cに含まれる固有表現のラベルと，その固有表現の種類に対応する意味カテゴリも素性として用いる．
- **CP**の品詞：**CP**は助詞，句読点，引用符などに分けられる．
- 候補Cと**兄弟の文節の格助詞**，つまり，候補Cが修飾する用言を修飾している他の文節の格助詞：属性共有では，先行詞候補の格が重要であるが，**CP**が「も」や「は」の場合，格が分からない．たとえば「太郎も連れてきた.」の場合,「太郎」が本来ガ格なのかヲ格なのか曖昧である.「太郎」をゼロ代名詞の候補として考える場合，その格を推定しておきたい.「太郎も次郎が連れてきた.」であれば，すでにガ格があるので,「太郎」はガ格でないと判断できる．同じ用言を修飾する他の格の格助詞は，このような判断をするためのヒントになる．
- Zと同一の文で，Zの直前に出てきた**接続助詞など**（文分割で用いたものと同じもの）：接続助詞などは，その前後に同じ主語を要求するものと，違う主語を要求するものがあることが知られている[22),29)]．これまでは人間が経験的に分類していたが，研究者により分類が異なる．本論文では，人手による分類はせず，接続助詞などの表現をそ

のまま素性に入れることにした．

- C が Z の文中に**明示的**に現れているかどうか：従来の研究[14),22)]でも，ゼロ代名詞と同じ文の中に現れる候補は重視されてきた．我々の手法では，ゼロ代名詞の位置に先行詞があると見なすので，同一文中に現れていない候補も $\mathbf{d} = 0$ になることがある．また，$\mathbf{d}$ は分割後の文の数なので，分割前の同じ文にあったかどうかは分からない．そこで，分割前の同一文において，明示的に現れる候補かどうかを素性とする．
  本論文では，ゼロ代名詞の指す先行詞が，ゼロ代名詞より前に，一度は現れている場合だけを対象としている．その意味で前方照応だけを対象としている．しかし，以下の例のように，ゼロ代名詞 Z の指すものが前にも後ろにも現れることがある．
  > 太郎さんと次郎さんは花子さんの家に呼ばれました．(Z が）遅刻しそうになったので，次郎さんは花子さんに電話をしました．

  この例は，前方照応と考えると難しいが，同一文の後ろを指す後方照応と考えると易しい☆．このような場合にも，この素性は有効であると考えられる．
- C の現れる節と Z の現れる節が**並列**関係にあるかどうか：これは以下のような場合に有効と考えて導入した素性である．
  > 太郎がテレビ（=C）をつけ，花子が（Z を）消した．

  我々の実装では，以下のような簡単な基準で判定した．CaboCha で得られる文節間の係り受け関係において，先行詞候補の文節（テレビを）の修飾する文節（つけ）が，ゼロ代名詞の文節（消した）を修飾している場合に並列とする．
- 候補を含む文節がゼロ代名詞の文節の**直前**かどうか：たとえば，以下の例では，Z の先行詞として「彼」が自然である．
  > 彼は （Z が）遅刻したので 入れなかった．

  予備実験において，ゼロ代名詞の直前の文節に正しい先行詞があるのに，他の候補を選ぶパターンが目立ったので，これを素性として追加した．

以上において，意味カテゴリや品詞などは，それぞれの品詞や意味カテゴリごとに，対応する素性番号をつけ，該当すれば 1，該当しなければ 0 とする．

☆ 前方照応か後方照応か曖昧なケースがあることは，Mitkov の本[15)] の pp.19–20 などでも指摘されている．

また，連体修飾節の中かどうかの判定について，連体修飾節はネストでき，ゼロ代名詞が連体修飾節の中に現れることもあるので，経験則では，ゼロ代名詞の位置ですでに終わっている節だけを **r**=1 とした．しかし，これが適切かどうか不明なので，機械学習では，「**連体修飾節中**」（候補が連体修飾節の中にある）と「**連体修飾節未終了**」（候補を含む最小の連体修飾節がゼロ代名詞の位置でまだ終了していない）という 2 つの素性として扱う．

さらに，ゼロ代名詞は話題の中心であることが多いので，以下の素性を唯一の実数値の素性として加える．これを「**正規化頻度**」と呼ぶ．

- 同一表記の候補がこれまでに生成された回数：これは，候補リスト中の各候補の生成回数の中で最大の値で割って 0〜1 に正規化した数値を利用する．

以上のようにして，各候補をベクトルで表現することができる．候補 C_i の素性ベクトルを $\boldsymbol{x}_i$ で表す．なお，素性はプログラムにより自動で付与し，訓練データ中に 3 度以上現れない素性は削除した．

3. 実験結果

以下の実験は，全記事のガ格だけを対象とした独立評価で行う．最初に，辞書式順序と機械学習の組合せ方に関する実験を行う．次に，飯田らのトーナメントモデルを再実装したシステムとの比較を行う．最後に，最近公開された京大コーパス 4.0 を用いて，手法の一般性の検証を行う．

3.1 辞書式順序と機械学習の組合せ

まず，前述の素性だけを用い，すでに述べた以下の機械学習方式を比較する．

- 通常の SVM
- 正例の誤分類に厳しい SVMJ
- SVM に基づく優先度学習 PL
- SVM によるトーナメント方式 TM

カーネルとしては，自然言語処理でよく用いられるリニアカーネルと 2 次の多項式カーネルを用いた実験を行った．以下では，リニアカーネル $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1 \cdot \boldsymbol{x}_2$ を用いた場合の SVM，SVMJ，PL，TM を SVM1，SVMJ1，PL1，TM1 のように表し，二次の多項式カーネル $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1 \cdot \boldsymbol{x}_2 + 1)^2$ を用いた場合を SVM2，SVMJ2，PL2，TM2 のように表す．リニアカーネルの実験には SVM-light 6.01 を用いたが，二次のカーネルの学習は時間がかかるため，より高速な我々の独自の実装を用いた．

また，各記事に対して，その記事をテストとして，

表 4 経験則を利用しなかった場合の独立評価（全記事，ガ格，D=5）
Table 4 Performance of the Machine Learning methods without using heuristic rules.
括弧内は C の値．3×10^{-2} を 3e-2 のように表す

SVM1	SVM2	SVMJ1	SVMJ2	PL1	PL2
64.1%	**64.4%**	62.9%	63.1%	63.8%	63.1%
(3e-2)	(3e-4)	(1e-1)	(1e-3)	(1e-2)	(1e-4)

残り 59 記事をトレーニングに使う実験を 60 回繰り返す．そしてシステムの成績は，この 60 回の実験で対象となるガ格の 738 のゼロ代名詞の正解率で表す．そして，各手法でソフトマージンのパラメータ C を 10^{-5}, 3×10^{-5}, 10^{-4}, ..., 100 と変えたとき（対数でほぼ等間隔）の最良の成績を**表 4** に示す．これによれば，機械学習を用いても，成績の良い辞書式順序（表 2）と同程度の性能しか得られていない．

また，PL と SVM の差はほとんどない．PL の性能が発揮できない原因として，初めに指摘した，複数インスタンス学習の問題設定になっていることが考えられる．

SVMJ は，PL や SVM に比べて若干悪い．SVMJ は，少数派である正例を負例と判断する誤分類がなるべくないように学習を行うが，逆に負例を正例と判断する誤分類を犯しやすい．もともと負例の方が数が多いこともあり，SVM に比べると，スコアの高い負例が生じやすい．そのため，スコア最大のものを選ぶという使い方をすると，SVM に比べ，成績が下がると考えられる．

本論文では，まず，辞書式順序によって同一表記の候補を 1 つに絞り込んでから機械学習を適用することにより，複数インスタンス問題を緩和する．次に辞書式順序の順位を素性として取り込む．つまり，以下の 2 段階を試す．

- **重複候補選択**：同じ表記で出現位置の異なる重複する候補から 1 つを選ぶときの優先順序として，独立評価でベストだった vadrs と，一括評価でベストだった vards を試す．これらの順序でベストの候補は，正例インスタンスを推定する性能が，単純な近さによるものよりも，優れていると推測される．
- **順位素性追加**：重複候補選択を行ったあと，辞書式順序による順位を学習時の素性として入れる．なお，他の素性と同様,「辞書式順序 1 位」などの命題に分解してブール変数の素性とする．

C を先ほどと同じように変えて得られた一番良い成績を**表 5** にまとめておく．これによれば，vadrs や vards を用いて「重複候補選択」を行うと，いずれの学習手法でも，全体的に成績が上がっている．複数インスタンス学習の問題として指摘したように，条件の良くない正例が少なくなり，正例と負例との差がつけやすくなったことが推定される．ただし，SVMJ の精度は，SVM や PL に比べやや低い．

表 5 機械学習 3 手法の独立評価（全記事，ガ格，D=5）
Table 5 Comparison of the Machine Learning methods (independent evaluation of zero subjects).

辞書式順序 vadrs (64.6%) との組合せ

	なし	重複候補選択	順位素性追加	差
SVM1	64.1	**66.8** (3e-2)	64.8 (3e-1)	+0.7
SVM2	**64.4**	66.3 (3e-4)	65.4 (1e-3)	+1.0
SVMJ1	62.9	64.0 (3e-2)	66.5 (1e-3)	+3.6
SVMJ2	63.1	63.7 (1e-3)	66.4 (3e-4)	+3.3
PL1	63.8	66.7 (1e-2)	**67.9** (3e-3)	+4.1
PL2	63.1	66.5 (3e-4)	67.6 (1e-4)	+4.5

辞書式順序 vards (62.9%) との組合せ

	なし	重複候補選択	順位素性追加	差
SVM1	64.1	66.9 (3e-2)	65.4 (1e-1)	+1.0
SVM2	**64.4**	66.3 (1e-3)	65.9 (1e-3)	+1.5
SVMJ1	62.9	64.6 (1e-2)	67.9 (1e-2)	+5.0
SVMJ2	63.1	64.4 (3e-4)	67.6 (3e-4)	+4.5
PL1	63.8	**67.1** (1e-2)	67.9 (3e-2)	+4.1
PL2	63.1	66.3 (3e-4)	**68.4** (3e-4)	+5.3

辞書式順序 d (41.2%) との組合せ

	なし	重複候補選択	順位素性追加	差
SVM1	64.1	65.3 (3e-2)	66.1 (3e-1)	+2.0
SVM2	**64.4**	**66.4** (1e-3)	65.3 (3e-3)	+0.9
SVMJ1	62.9	62.5 (3e-2)	64.6 (1e-1)	+1.7
SVMJ2	63.1	63.3 (1e-3)	64.6 (3e-4)	+1.5
PL1	63.8	66.0 (1e-2)	**66.4** (1e-2)	+2.8
PL2	63.1	65.4 (1e-4)	65.9 (1e-3)	+2.8

「順位素性追加」を行うと，SVMJ や PL ではさらに成績の向上につながった．とくに vards と PL2 の組合せでは，合わせて 5.3%向上し，68.4%の精度が達成された．しかし，SVM では逆効果で，順位素性を加えない方が良かった．

PL は，順位素性の有無にかかわらず，3 つの学習手法の中で，ベストかそれに近い成績が得られている．これは，単なる 2 クラスへの分類ではなく，相対的な良し悪しを考慮できるようにデザインされた PL の特長が発揮されたためと考えられる．

なお，辞書式順序として d を用いた場合は，vadrs や vards に比べて上がり方が少なく，vadrs や vards の方が d より良い順序であると考えられる．

vards の場合に McNemar 検定を行ったところ，SVM1 の「なし」と「重複候補選択」は $p < 0.05$ で有意差があったが,「順位素性追加」は成績が下がっているため,「なし」との有意差がなかった．SVMJ2 や PL2 の「なし」と「順位素性追加」は $p < 0.01$ で有

意差があった．

また，vards の「順位素性追加」どうしでは，SVM1 と PL2 の間に $p < 0.01$，SVM2 と PL2 の間に $p < 0.05$ の有意差があった．SVMJ2 と PL2 の間に有意差はなかった．

3.2 各素性の有効性の確認

機械学習でどの素性がどの程度有効だったか調べるため，リニアカーネルで一番成績の良い PL1（vards の順位素性追加）で，各素性の重みを調べた結果を**表 6** に示す．これによれば，「辞書式順序 1 位」，**d**=0，**a**=0，などの素性が大きな正の重みを得ている．また，**d**=5，**v**=1，「連体修飾節中」などの素性が負の大きな重みを得ている．これらの重みは，我々の用いた経験則を裏づける結果となっている．

「連体修飾節未終了」は「連体修飾節中」をほぼ打ち消す重みを持っており，我々の経験則における **r**=1 の定義は適切であったと考えられる．

ただし，大きな重みを得た素性が不可欠とは限らず，冗長な素性の可能性もある．とくに，辞書式順序に現れていない素性は，これまでの研究であまり注目されてこなかった素性である．そこで，これらの素性がどの程度貢献しているか調べる．

表 7 は，PL1 と PL2 において，各素性を除いた場合の成績の変化である．これによると，PL1 は PL2 に比べ，素性の削除に対して成績が低下しやすい傾向があることがうかがえる．PL2 は他の素性の組合せを考慮するので，削除された素性をある程度カバーできるため，PL1 に比べて精度の低下が少ないと考えられる．

表 6 各素性の重み
Table 6 Weights of features.
vards，PL1 (3e-2)，順位素性追加，全記事，ガ格，独立評価

重み	素性
0.515	辞書式順序 1 位
0.446	**d**=0
0.369	正規化頻度
0.283	**a**=0
0.253	**CP**=「が」
0.246	**d**=1
0.237	意味カテゴリ「主体」
0.231	並列
0.196	連体修飾節未終了
0.160	主辞の品詞が「接尾」
0.159	明示的
:	:
−0.168	**d**=4
−0.212	**d**=3
−0.216	**v**=1
−0.232	連体修飾節中
−0.267	**d**=5

「接続助詞など」は，同じゼロ代名詞のすべての候補に共通して出現する性質であるため，PL1 ではまったく影響がなかったが，PL2 では他の素性との組合せが考慮されるため，若干成績が変化している．

また，「直後」も効果がなかった．表中のこれ以外の素性はいずれも PL1 において 1%前後の性能向上に寄与している．

vards の場合の「辞書式順序 1 位」の素性に対する重み w_0 を調べてみた．学習によって得られた決定関数を $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}(+b)$ と表したときに，w_0 の重みの占める比率を $|w_0|/\|\boldsymbol{w}\|$ で表す．すると，PL は 60 回の実験でいずれも 0.36 前後，SVMJ は 0.38 前後であったのに対し，SVM は 0.58 前後であり，SVM は「辞書式順序 1 位」に依存しすぎていることが分かった．

3.3 飯田らの手法との比較

飯田らの手法の再実装の実験結果を**表 8** に示す．飯田らは，SRL と呼ばれる経験則[21] を用いているので，SRL だけを用いた結果も示す．SRL だけの性能は全記事で 44.3%，General に限れば 56.6%であり，この

表 7 各素性を削除した場合の成績の変化（vards 順位素性追加）
Table 7 How accuracy changed when a feature was removed.

削除した素性	PL2 (3e-4)	PL1 (3e-2)
並列	0.0%	−1.0%
直後	0.0%	−0.1%
正規化頻度	+0.6%	−1.1%
兄弟の格助詞	−0.4%	−1.5%
明示的	−0.2%	−1.0%
意味カテゴリ	−2.5%	−1.4%
接続助詞など	−0.5%	0.0%

表 8 飯田らのトーナメントモデルとの比較（独立評価，ガ格）
Table 8 Comparison with Iida et al.'s Tournament Model.

手法		全記事	(General)
飯田らの素性を用いた場合			
SRL のみ		44.3	(56.6)
TM1	(1e-0)	*64.2*	*(71.9)*
TM2	(1e-3)	*64.6*	*(71.2)*
SVM1	(3e-0)	49.5	(63.7)
SVMJ1	(3e-2)	58.1	(70.2)
PL1	(3e-2)	59.6	(73.2)
本論文の素性を用いた場合			
vadrs		64.6	(73.2)
vards		62.9	(74.2)
TM1	vadrs, (1e-2)	66.1	(74.2)
TM2	vadrs, (1e-3)	67.8	(76.6)
PL1	vadrs, (3e-3)	67.9	(73.6)
TM1	vards, (1e-2)	67.2	(76.3)
TM2	vards, (1e-3)	67.9	(77.6)
PL2	vards, (3e-4)	**68.4**	(76.6)

結果は，飯田らの論文における「Nariyama の解析モデル」の数字 45.6%に近い．

また，我々の再実装による TM1 のスコアは 64.2%，TM2 は 64.6%であった．飯田らが用いた素性の中で，単一の候補の素性だけを用いて学習した場合の SVM1 の最大スコアによる成績は 49.5%であったので，TM の方が SVM に比べ 15%もスコアが良い．ただし，単一候補の素性だけを用いた PL1 は 59.6%に上昇し，差は 5%まで縮まった．また，General だけの成績では，PL1 の方が TM より良かった．

つまり，飯田らの素性を用いた場合，優先度学習は SVM よりも良いが，トーナメントモデルの方がさらに成績が良い．PL2 は PL1 よりも良くなる可能性があるが，飯田らの方法は，候補の数が多く，PL2 での学習は時間がかかりすぎて結果が得られなかった．

飯田らの論文の学習曲線のグラフによれば，ゼロ代名詞の数が同じくらいのところで 72〜73%の性能であり，General の 71.9%と近い．このことから，飯田らのデータは，General に近い性質を持つと推定される．

一方，本論文の素性を用いると，vadrs の辞書式順序だけで，飯田らの素性を用いた TM2 の成績と同じになった．また，本論文の素性を用いた場合の TM2 の結果は 67.9%で，PL2 の 68.4%に近い結果が得られた．これは，辞書式順序という強力なヒントが得られたためであろう．

3.4 京大コーパス 4.0 による検証

これまでの実験結果は，小規模な開発用のデータによる評価であり，手法の一般性は不明である．そこで，開発に用いなかった他のデータを用いて検証をする．最近公開された京大コーパス 4.0☆ には，照応・省略関係のタグが付与されており，ゼロ代名詞のデータとしても利用できる．そこで，このデータを我々が扱っていたのと同じ前提・同じ形式のデータになるよう自動変換して実験を行った．

ただし，開発用データと，京大コーパス 4.0 の照応・省略関係のデータには，共通の記事が 10 件あり，この 10 件は除いた．先行詞が「不特定」など名詞句の選択で扱えないケースや非自立名詞（「こと」など），引用文中のものは，我々のプログラムの処理対象外であるので，除いた．その結果，合計 2,897 個のガ格のゼロ代名詞が得られた．

なお，正解先行詞が「同 +X」や「彼」「彼女」などの場合，曖昧なので，照応関係のタグを遡って，具体的な名詞句に置き換えられる場合は置き換えた．また，同じ格に複数の正解先行詞が記述されている場合は，そのいずれか 1 つが答えられればよいものとした．

まず，辞書式順序だけの精度を調べると，**表 9** のようになった．開発用データで成績の良かった vadrs や vards も良い成績であったが，ベストは avrds であった．svadr や dvars などの d や s から始まる順位は，開発用データと同様に精度が低い．

このデータを，記事に含まれているゼロ代名詞の数の分布がなるべく同じになるように 5 等分し，機械学習を用いて 5 分割交差検定を行った．つまり，5 分割して得られる各ファイルは，平均 579.4 個のゼロ代名詞を含む．辞書式順序としては表 5 で良好だった vards を用い，表 5 で最適だった C を利用したときの機械学習で得られた精度を**表 10** に示す．機械学習を用いた方法の性能は，開発データに比べ，2%ほど下がっている．この結果によれば，やはり PL の方が SVM より良い．そして，PL2 と TM1 は，ほぼ同程度の性能であった．

飯田法の再実装は，やはりこのデータでも良い成績が得られなかった．開発用データと違い，京大コーパ

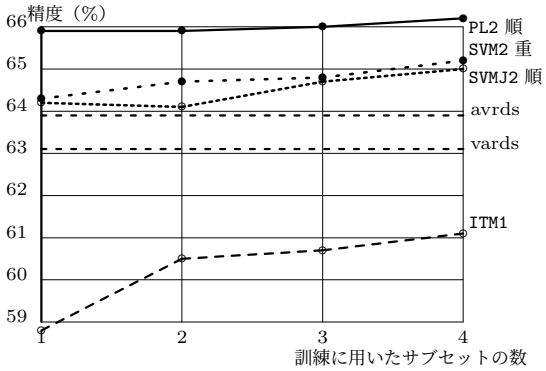
表 9 京大コーパス 4.0 による辞書式順序の独立評価（全記事，ガ格）
Table 9 Performance of lexicographical ordering rules for Kyoto Corpus 4.0.

辞書式順序	精度（%）
avrds	**63.9**
vards	63.1
vrads	63.0
avdrs	62.7
avdsr	62.4
vadrs	61.8
vadsr	61.6
svadr	49.7
dvars	47.6
d	37.0

表 10 京大コーパス 4.0 による機械学習の独立評価（全記事，ガ格）
Table 10 Performance of machine learning methods for Kyoto Corpus 4.0.
「重」は重複候補選択，「順」は順位素性追加

機械学習		精度
SVM1	vards 重 (3e-2)	63.5
SVM2	vards 重 (1e-3)	65.2
SVMJ1	vards 順 (1e-2)	64.7
SVMJ2	vards 順 (3e-4)	65.0
PL1	vards 順 (3e-3)	65.5
PL2	vards 順 (3e-4)	**66.2**
TM1	vards 順 (1e-2)	65.8
TM2	vards 順 (1e-3)	65.2
ITM1	(1e-0)	61.1

☆ http://pine.kuee.kyoto-u.ac.jp/nl-resource/corpus.html



「重」は vards による重複候補選択,「順」は順位素性追加

図 3 京大コーパス 4.0 による学習曲線（5 分割交差検定の平均精度）
Fig. 3 Learning curve for Kyoto Corpus 4.0.

ス 4.0 には照応関係の情報が付与されているので，これらの情報を生かした実装にすれば，もっと成績は上がるであろう．

訓練データの量による性能の変化を**図 3** に示す．5 分割で得られた 5 つのサブセットのうち，テスト用を除く 4 つのサブセットが訓練データとして利用できる．この 4 つのサブセットのうち，いくつを訓練に利用したかが横軸である．C の値は，それぞれ開発データでベストの値を利用している．

この図によれば，PL2 がこれら 4 つの学習手法で一番良い．しかし，その精度は訓練ファイルの数を増やしても，ほとんど向上していない．これは，vards の辞書式順序によるベースの成績が高いためと考えられる．したがって，今後は学習の効果がもっと上がるような素性を探す必要がある．

4. 考　　察

「順位素性追加」で SVMJ や PL が SVM より好成績だった理由について考えてみよう．リニアカーネルの SVM の場合,「辞書式順序 1 位」に対応する次元の追加により，この次元の値の 0 と 1 の間で，正解と不正解の間のマージンを広くとることができる．そのため,「辞書式順序 1 位」という素性への依存が高くなる.「辞書式順序 1 位」でない正例は誤分類されやすくなるが，トレーニングデータ全体から見れば，その数は少ないので，学習結果をあまり大きく左右しない．

一方，PL は正例と負例の差のベクトルに着目するので，あるゼロ代名詞の正例 1 つに対して負例が 10 個あれば，その差が 10 倍に強調される．その結果,「辞書式順序 1 位」でない正例の持つ特徴が強調して学習されると考えられる．SVMJ も正例の誤りが強調されるので，同様に「辞書式順序 1 位」でない正例の持つ特徴が学習されると考えられる．

本論文では，機械学習と経験則の効果的な組合せ方法を探った．これまでに様々な経験則が提案されているが，それを統合する方法として，選択制限と属性共有を重視した vards などの単純な辞書式順序の有効性が分かった．

また，今回は，機械学習手法単独では，経験則と同程度の精度しか達成できなかったが，大量のコーパスがあれば，もっと高精度になることが期待できる．その場合に，今回の手法がどの程度有効かの検証は，今後の課題である．

本論文の実験では，経験則を用いて同一表記の複数の候補を 1 つに絞ることで，学習にかかる時間を削減しつつ，性能を向上させることができた．さらに，経験則による順位を学習の素性として導入することで，SVM 以外の学習手法の性能が向上することを確認した．

また，SVM や SVMJ に比べて優先度学習が安定して好成績であった．著者の知る限り，このような比較実験はこれまでほとんど行われていない．自然言語処理では，複数の候補をランキングする，という処理がしばしば利用されるので，人工データではない現実のデータでこれからも優先度学習の有効性を検証していきたい．優先度学習の手法としては，SVM に基づくもののほかに，RankBoost [5] などの手法が提案されており，これらとの比較も今後の課題である．

さらに精度を向上させるための参考にするため，京大コーパス 4.0 で PL2 が間違えた場所を調べた．

まず，正解が候補リストに入っていないゼロ代名詞が 206 個，7.1%あった．そのほとんどは，候補が遠すぎるせいである．これらは，距離の上限 D を増やすことでかなりカバーできるようになるが，今のままのシステムでは，単純に D を増やしても精度は逆に成績は下がっていく．たとえば D=6 の場合，開発データでの PL2 の精度は 67.3%に下がる．したがって，何か別の解決方法を考える必要がある．

形態素解析や係り受け解析が失敗し，候補を生成できなかった場合もある．たとえば，人名の次に記号があったり，平仮名だけで人名が書かれていたりする場合は，単語や文節の区切りを誤るので，正しく候補が生成できない．

また，PL2 の出力で正解が 2 位のゼロ代名詞は 10.4%，3 位は 5.1%あった．これらを 1 位にできればよい．そこで，正解が 2 位になった場合を調べてみると，夫婦，親子，団体と構成員などで，一方が話題の

中心，もう一方が話者や脇役である場合に間違うケースが多く見受けられた．たとえば,「〜と，田中次郎さんの妻は語る.」のようなパターンでは,「田中次郎さん」が話題の中心だが，助詞からすると「妻」の方が有利になる．

また，体言止めも問題を引き起こしている．たとえば,「〜した山田太郎.」というパターンでは「山田太郎」が話題の中心だが，助詞がないので格が一致せず，ランクが下がる．今後は，これらの問題点を解決し，さらに精度を高めたい．

5. 結　　論

本論文では，ゼロ代名詞解消が「複数インスタンス学習」の一種であるという観点から，正例にまぎれこんだ負例をなるべく排除するために，既存の複数の経験則をベースとした候補間の辞書式順序を考案し，実験を行った．その結果，選択制限と属性共有を重視した辞書式順序は，単純な距離による順序よりも精度が良く，SVM などの機械学習と同程度の性能を達成できた．そして，この辞書式順序により機械学習に用いる候補を絞ったり，その順位を機械学習に用いたりすることで，機械学習の精度が向上することが明らかになった．また，SVM をベースとする 3 種類の機械学習を比較し,「優先度学習」が比較的安定して成績が良いことが判明した．飯田らのトーナメントモデルとの比較実験を行ったところ，飯田らの素性では優先度学習よりもトーナメントモデルの方が良かったが，辞書式順序を含む本論文の素性では，優先度学習の方が良い成績が得られた．

参 考 文 献

1) Aone, C. and Bennett, S.W.: Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies, *Proc. ACL-1995*, pp.122–129 (1995).
2) Cristianini, N. and Shawe-Taylor, J.: *An Introduction to Support Vector Machines*, Cambridge University Press (2000).
3) Dietterich, T.G., Lathrop, R.H. and Lozano-Perez, T.: Solving the Multiple-Instance Problem with Axis-Parallel Rectangles, *Artificial Intelligence*, Vol.89, No.1–2, pp.31–71 (1997).
4) 江原暉将，金　淵培：確率モデルによるゼロ主語の補完，自然言語処理，Vol.3, No.4, pp.67–86 (1996).
5) Freund, Y., Iyer, R. and Singer, Y.: An Efficient Boosting Algorithm for Combining Preferences, *Journal of Machine Learning Research*, Vol.4, pp.933–969 (2003).
6) Grosz, B.J., Joshi, A.K. and Weinstein, S.: Centering: A Framework for Modelling the Local Coherence of Discourse, *Computational Linguistics*, Vol.21, No.2, pp.203–226 (1995).
7) Herbrich, R., Graepel, T., Bollmann-Sdorra, P. and Obermayer, K.: Learning Preference Relations for Information Retrieval, *Proc. ICML-98 Workshop on Text Categorization and Machine Learning*, pp.80–84 (1998).
8) 飯田　龍，乾健太郎，松本裕治：文脈的手がかりを考慮した機械学習による日本語ゼロ代名詞の先行詞同定，情報処理学会論文誌，Vol.45, No.3, pp.906–918 (2004).
9) 池原　悟，宮崎正弘，白井　諭，横尾昭男，中岩浩巳，大山芳史，林　良彦：日本語語彙大系，岩波書店 (1997).
10) 磯崎秀樹，賀沢秀人：固有表現抽出のための SVM の高速化，情報処理学会論文誌，Vol.44, No.3, pp.970–979 (2003).
11) Joachims, T.: Making Large-Scale Support Vector Machine Learning Practical, *Advances in Kernel Methods*, Schölkopf, B., Burges, C.J.C. and Smola, A.J. (Eds.), chapter 16, pp.170–184, MIT Press (1999).
12) Joachims, T.: Optimizing Search Engines using Clickthrough Data, *Proc. ACM Conference on Knowledge Discovery and Data Mining* (2002).
13) Kameyama, M.: A Property-Sharing Constraint in Centering, *Proc. ACL-1986*, pp.200–206 (1986).
14) 河原大輔，黒橋禎夫：自動構築した格フレーム辞書と先行詞の位置選好順序を用いた省略解析，自然言語処理，Vol.11, No.3, pp.3–19 (2004).
15) Mitkov, R.: *Anaphora Resolution*, Longman (2002).
16) Morik, K., Brockhausen, P. and Joachims, T.: Combining statistical learning with a knowledge-based approach — A case study in intensive care monitoring, *Proc. ICML-1999*, pp.268–277 (1999).
17) 村田真樹，長尾　真：用例や表層表現を用いた日本語文章中の指示詞・代名詞・ゼロ代名詞の指示対象の推定，自然言語処理，Vol.4, No.1, pp.41–56 (1997).
18) 中岩浩巳，池原　悟：日英翻訳システムにおける用言意味属性を用いたゼロ代名詞照応解析，情報処理学会論文誌，Vol.34, No.8, pp.1705–1715 (1993).
19) 中岩浩巳，池原　悟：語用論的・意味論的制約を用いた日本語ゼロ代名詞の文内照応解析，自然言語処理，Vol.3, No.4, pp.49–65 (1996).

20) 中岩浩巳，白井 諭，池原 悟：日英機械翻訳における語用論的・意味論的制約を用いたゼロ代名詞の文章外照応解析，情報処理学会論文誌，Vol.38, No.11, pp.2167–2178 (1997).
21) Nariyama, S.: Grammar for ellipsis resolution in Japanese, *Proc. 9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp.135–145 (2002).
22) Okumura, M. and Tamura, K.: Zero Pronoun Resolution Based on Centering Theory, *Proc. COLING-1996*, pp.871–876 (1996).
23) Seki, K., Fujii, A. and Ishikawa, T.: A Probabilistic Method for Analyzing Japanese Anaphora Integrating Zero Pronoun Detection and Resolution, *Proc. COLING-2002*, pp.911–917 (2002).
24) 関 和弘，藤井 敦，石川徹也：確率モデルを用いた日本語ゼロ代名詞の照応解析，自然言語処理，Vol.9, No.3, pp.63–85 (2002).
25) Walker, M., Iida, M. and Cote, S.: Japanese Discourse and the Process of Centering, *Computational Linguistics*, Vol.20, No.2, pp.193–233 (1994).
26) Yamamoto, K., Sumita, E., Furuse, O. and Iida, H.: Ellipsis Resolution in Dialogues via Decision-Tree Learning, *Proc. NLPRS-1997*, pp.423–428 (1997).
27) Yamura-Takei, M., Fujiwara, M., Yoshie, M. and Aizawa, T.: Automatic Linguistic Analysis for Language Teachers: The Case of Zeros, *Proc. COLING-2002*, pp.1114–1120 (2002).
28) Yang, X., Zhou, G., Su, J. and Tan, C.L.: Coreference Resolution Using Competition Learning Approach, *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, pp.176–183 (2003).
29) 吉本 啓：談話処理における日本語ゼロ代名詞の扱いについて，情報処理学会自然言語処理研究会報告，NL-56-4, pp.1–8 (1986).
30) 吉野圭一：機械学習を用いた日本語ゼロ代名詞照応関係の同定，修士論文，奈良先端科学技術大学院大学 (2001).

付 録

A.1 候補の列挙

(1) 助詞や句読点が末尾に来る文節は，助詞や句読点を除いたものを候補 C として取り出す．また，文節の最後にこれらがない名詞句も候補 C として取り出す．

(2) C の末尾の単語が動詞，形容詞，助動詞，副詞，非自立名詞（「こと」「もの」「とき」など）の場合は，C を候補からはずす（末尾が閉じ括弧の場合は，その直前の単語で判断する）．

(3) C の末尾が副詞可能名詞（「今年」「昨日」「すべて」など）ならば，C を候補からはずす．

さらに，ゼロ代名詞 Z を解消したら，選ばれた候補 C_a を候補リストに追加する．つまり，ゼロ代名詞 Z の位置に，はじめから C_a が書かれているかのように処理を行う．なお，このとき，C_a の素性を以下のように更新する．C_a の後続助詞はゼロ代名詞の格 **ZP** であり，C_a の位置は Z の位置である．機械学習の訓練段階では，正解そのものを C_a と見なす．

A.2 選択制限の実現

我々が用いた構文大系は 6,103 の日本語の動詞に対して 14,730 の日英翻訳用のパターンが定義されているもので，候補が受理できるものかどうかを確認するのに利用できる．また語彙大系は 3,000 弱の意味カテゴリに分類された約 30 万語の辞書を持つ．

1 つの単語が複数の意味カテゴリを持つことがある．また，各意味カテゴリは上位のカテゴリを持つ．たとえば「父」というカテゴリは「親」という上位カテゴリに属する．ここでは，これらすべてのカテゴリを用いた．また，ChaSen では 1 語として扱われるが，語彙大系に対応する単語がない場合は，その語を語彙体系ベースの形態素解析器により分割して得られる最後の単語の意味カテゴリを用いた．

さらに，固有表現抽出を用いて，組織名，人名，地名に対応するものは対応する意味カテゴリを追加し，人工物名，日付，時刻，金額，割合は，それを新しい意味カテゴリとして追加した．

動詞が複数の翻訳パターンを持つ場合は，それらの選択制限のいずれかを満たせばよいとした．たとえば，「読む」という単語は 3 つの翻訳パターンを持つ．第 1 と第 2 のパターンの主語は「主体」であり，第 3 のパターンの主語は「人」である．したがって，「読む」の主語は「主体」か「人」の場合に選択制限を満たす．

(平成 17 年 6 月 27 日受付)
(平成 18 年 4 月 4 日採録)

磯崎　秀樹（正会員）

1983 年東京大学工学部計数工学科卒業．1986 年同工学系大学院修士課程修了．同年日本電信電話（株）入社．1990〜1991 年スタンフォード大学ロボティクス研究所客員研究員．現在，NTT コミュニケーション科学基礎研究所特別研究員．博士（工学）．人工知能・自然言語処理の研究に従事．電子情報通信学会，人工知能学会，言語処理学会，AAAI，ACL 各会員．

賀沢　秀人（正会員）

1995 年東京大学理学部物理学科卒業．1997 年同大学院理学系研究科修士課程修了．同年日本電信電話（株）入社．現在，NTT コミュニケーション科学基礎研究所所員．自然言語処理・機械学習の研究に従事．ACM，IEEE 各会員．

平尾　　努（正会員）

1995 年関西大学工学部電気工学科卒業．1997 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同年 NTT データ通信株式会社（現，株式会社 NTT データ）入社．2000 年より日本電信電話株式会社 NTT コミュニケーション科学基礎研究所に所属．2002 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．自然言語処理の研究に従事．言語処理学会，ACL 各会員