

# 名詞の可算/不可算性を利用した 英文の冠詞誤り検出に関する研究

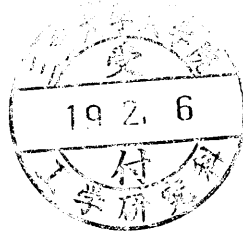
平成 18 年 度

三重大学大学院工学研究科  
博士前期課程 情報工学専攻

若 菜 崇 宏

# 修士論文

## 名詞の可算／不可算性を利用した 英文の冠詞誤り検出に関する研究



平成 18 年度修了

三重大学大学院 工学研究科

博士前期課程 情報工学専攻

若菜 崇宏

## 目次

<b>1. 序論</b>	<b>1</b>
<b>2. 判定規則</b>	<b>2</b>
2. 1 学習データの自動生成	2
2. 2 判定規則の学習	6
2. 3 限定判定規則の優先度算出	7
2. 4 一般判定規則の優先度算出	7
2. 4. 1 一般性評価の方法	8
2. 5 限定判定規則のデフォルト規則	9
<b>3. 可算／不可算の判定</b>	<b>10</b>
<b>4. 実験と評価</b>	<b>11</b>
4. 1 実験条件	11
4. 2 評価結果と考察	11
<b>5. まとめ</b>	<b>16</b>
謝辞	17
参考文献	18
関連文献	19
付録 A 可算／不可算の判定実験用環境	20
付録 B WEB 上のデモシステム	26
付録 C 一般判定規則に関する分析	27
付録 D 修士論文発表資料	29

## 1. 序論

日本人英語学習者が書いた英文に多く見られる，冠詞の誤りや単数／複数の使い分けに関する誤りを検出するためには，名詞の可算／不可算の判定が重要である．なぜなら，可算／不可算の情報が与えられると，表 1 の×で示される部分が誤りで有ることが分かり，上記の誤り（以下では，表記を簡単にするため，これらの誤りを冠詞誤りと呼ぶことにする）が検出できるからである．例えば“I have a furniture.”という英文で，“furniture”が不可算名詞であることが分かれば，表 1 から冠詞の余剰として検出できる．一方，表 1 の○で示される部分は，可算／不可算の情報からは誤りであるかどうかの判断は出来ず検出対象外となるが，学習者の書いた英文においては×で示される部分に比べ少ない誤りである．

表 1 可算／不可算に基づいた誤り検出ルール

	単数			複数		
	不定冠詞	定冠詞	無冠詞	不定冠詞	定冠詞	
可算	○	○	×	×	○	○
不可算	×	○	○	×	×	×

誤りを含まない英文では，冠詞や単数／複数などの表層情報から可算／不可算の判定は比較的容易に行える．例えば，複数形の名詞は可算名詞であるし，無冠詞単数の名詞は不可算名詞である．一方，誤りを含む英文では，冠詞や単数／複数の用法が間違っている可能性があるので，これらの表層情報を用いることが出来ない．従って，冠詞誤りを検出する際には，これらの表層情報を利用しない可算／不可算の判定手法が必要となる．

可算／不可算を判定する先行研究として，英文コーパスから可算／不可算の判定規則（以下，表記を簡単にするため，単に判定規則とする）を生成する手法[4]がある．この手法では可算／不可算を判定の対象となっている名詞（以後ターゲット名詞と呼ぶ）の文脈情報（本論文では，ターゲット名詞周辺の単語のことを文脈情報と呼ぶ．冠詞や代名詞などの機能語は除外する）に基づいて可算／不可算の判定を行う．例えば，ターゲット名詞 **paper** に対する判定規則は，次のような形で生成される：

規則 1      read[-] → 可算  
 規則 2      pencil[+] → 不可算  
 :

規則 10     pulp[+] → 不可算  
 規則 11     author[+] → 可算  
 :  
 規則 n       paper → 可算

ここで[-][+]はターゲット名詞の現れた名詞句からの位置関係であり、それぞれ名詞句の前、及び後ろに現れたことを示す。例えば規則 1 は「read が *paper* の現れた名詞句より前に現れたら可算と判定」という意味である。規則 n はデフォルト規則で、英文コーパス中で *paper* が現れたとき、可算／不可算のどちらが多かったかを示す。デフォルト規則は規則 1～n-1 に適用可能な規則が無い場合にのみ用いられる。

この判定規則を用いて、可算／不可算の判定を行う。例えば、

例文 (1) I read the paper.

例文 (2) The paper is made of hemp pulp.

中の *paper* に対して可算／不可算の判定を行うことを考える。ここでターゲット名詞の文脈情報を見ると、例文 (1) では規則 1 が、例文 (2) では規則 10 が上記判定規則に当てはまることが分かる。よって例文 (1) の *paper* は可算、例文 (2) の *paper* は不可算と正しく判定される。

しかし、上記例で用いた判定規則はターゲット名詞を *paper* に限定し、*paper* と共に用いられた単語を規則としている。このため、学習データ中で、*paper* と共起しなかった単語は、規則に利用されない。例えばターゲット名詞を *paper* とし下記の英文

... She wrapped a thing in paper...

が与えられたとする。もし学習データ中で *paper* が wrap, thing, in のいずれとも共起しなかった場合、デフォルト規則が適用され、可算と判定される。しかしデフォルト規則はターゲット名詞が学習データ中で可算／不可算のどちらで多く使われているかに基づいて判定を行うので、文脈情報に基づいた規則より判定精度は低くなる傾向にある[4]。上記の例でも、デフォルト規則は誤ってターゲット名詞を可算と判定している。一方で、wrap, thing, in が判定規則に含まれていたならば、正しく不可算と判定される可能性が高いといえる。すなわち、判定規則を増加させることで、この問題が解決できると言える。

そこで本論文では、判定規則を効率良く増加させる手法をさらに提案する。提案手法では、ターゲット名詞を特定の単語に限定せず、コーパス中に出現する全ての名詞を1つの名詞として扱う。そうすることで、あらゆる名詞に適用可能な判定規則（以下、一般判定規則）を学習する。一般判定規則と従来のターゲット名詞を限定した判定規則（以下、限定判定規則）と組み合わせることで、規則の増加を図り、判定精度を向上させる。

以下、2. では提案手法である一般判定規則と限定判定規則の学習方法について述べる。3. で限定判定規則と一般判定規則を用いた可算／不可算の判定方法について述べる。4. で可算／不可算の判定実験とその考察を行う。5. でまとめを行う。

## 2. 判定規則

この章では主に判定規則の学習について述べる。以下 2. 1 では学習データの自動生成方法について説明する。2. 2 で判定規則の学習方法及び、一般判定規則と限定判定規則の違いについて説明する。2. 3 で限定判定規則の優先度の算出方法を説明する。2. 4 で一般判定規則の優先度の算出方法を説明する。2. 5 で限定判定規則のデフォルト規則について説明する。

### 2. 1 学習データの自動生成

一般判定規則及び限定判定規則は、ターゲット名詞の可算／不可算の例からなる学習データから学習される。可算／不可算の例とは

She gave a paper／可算 on wild animals.

の様に、ターゲット名詞に可算／不可算のタグが付与されたものである。

学習データはコーパスから以下の手順により自動生成される。

- (1) ターゲット名詞の抽出
- (2) ターゲット名詞のタグ付け
- (3) タグ付けされたターゲット名詞の保存

(1) では主名詞として使用されているターゲット名詞をその周辺の単語とともにコーパスから抽出する。この処理は既存の構文解析などで行うことが出来る。ただし、一般判定規則を学習する場合、ターゲット名詞は限定しない。よって、一般判定規則を学習する場合はコーパス中のあらゆる名詞を 1 つの名詞とみなして抽出する。抽出の際には、単語を小文字かつ原形（例えば、Boxes から box）に変換する。ただし、表 2 中の単語、代名詞や助動詞などの機能語、基数、ターゲット名詞は抽出しない。

(2) では以下に述べるルールを用いて、抽出されたターゲット名詞に可算／不可算のタグを付与する。例えば、

She gave a paper on wild animals.

中の *paper* は単数形で不定冠詞が付いていることから

She gave a paper / 可算 on wild animals.

とタグ付け出来る。

図 1 と表 2 に、言語学の知見[1][2][3]に基づいて作成した可算／不可算のタグ付けのためのルールを示す。なお、詳細については文献[4]を参照されたい。

図 1 中のノードは、ターゲット名詞に適用される質問を表す。例えば、ルートノードは、「ターゲット名詞は複数形であるか。」と解釈される。また、図 1 中の葉は分類結果に対応する。例えば、ルートノードの質問の答えが“yes”であれば可算と分類される。“no”の場合は、次のノードの質問が適用される。図中の“?”は、ルールによって可算／不可算の分類が出来ないことを表す。

(3) で、上記ルールによって可算／不可算のタグ付けされたターゲット名詞とその周辺の単語を保存し、学習データとする。

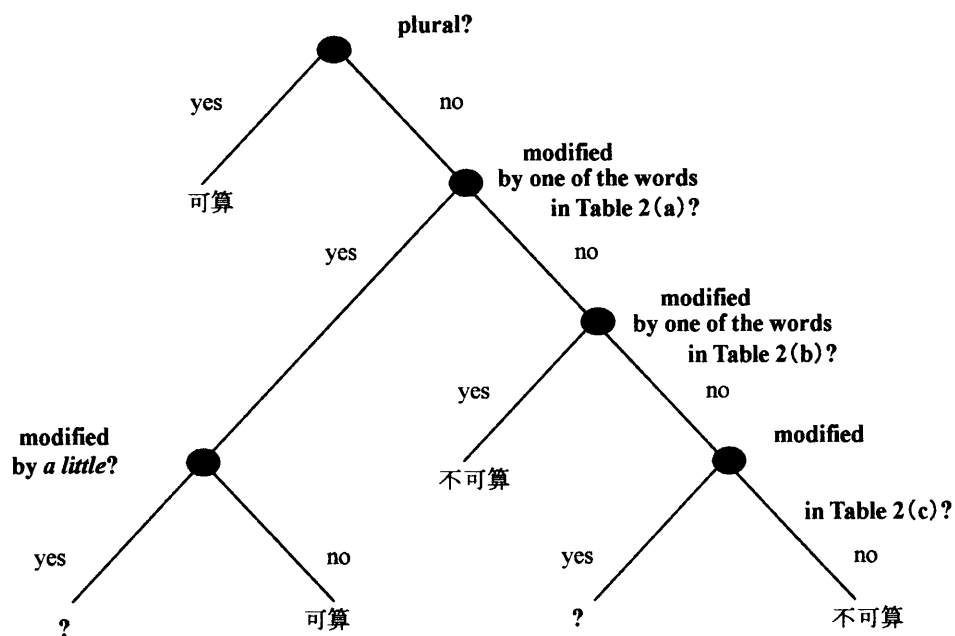


図 1 学習データ生成のためのルール



表 2 図 1 中で使用される単語群

(a)	(b)	(c)
<i>the indefinite article</i>	<i>much</i>	<i>the definite article</i>
Another	less	<i>demonstrative adjectives</i>
One	enough	<i>possessive adjectives</i>
Each	all	<i>interrogative adjectives</i>
-	sufficient	<i>quantifiers</i>
-	-	<i>'s genitive</i>

## 2. 2 判定規則の学習

判定規則のテンプレートを定義するため次の記号を導入する．ターゲット名詞が可算／不可算になることを変数  $MC$  を用いて表す． $MC$  は可算／不可算を値にとると定義する．また，単語を  $w$ ，ターゲット名詞周辺の文脈を  $C$  で表す．文脈  $C$  として，NP（ターゲット名詞が主名詞となっている名詞句内の単語）， $\pm$ （その名詞句から左（-）または右（+）に 3 単語）の 3 種類を定義する．このときテンプレートは

単語  $w$  が 文脈  $C$  に現れたら  $MC$  と判定

と定義する．以下表記を簡単にするためテンプレートを

$$w_C \rightarrow MC \quad (1)$$

で表すことにする．

次に 2. 1 で説明した学習データを用いて判定規則の学習を行う．まず，学習データからターゲット名詞周辺の文脈に現れる単語を抽出しテンプレートに適合する規則を生成する．

以下に規則の生成例を示す．いま，学習データ

He cooked beef／不可算 and fish for dinner.

I ate a piece of chicken／不可算 with salad.

が与えられたとする．ターゲット名詞を *chicken* として限定判定規則を学習する場合，規則

eat[-] → 不可算, piece[-] → 不可算, salad[+] → 不可算

が生成される。一方、一般判定規則を学習する場合は、規則

cook[-] → 不可算, fish[NP] → 不可算, dinner[+] → 不可算  
eat[-] → 不可算, piece[-] → 不可算, salad[+] → 不可算

が生成される。

### 2. 3 限定判定規則の優先度算出

次に、生成されたルールの重要度を決定するために対数尤度比を計算する。対数尤度比は、 $w_C$  が成立するときにターゲット名詞が  $MC$  となる条件付き確率  $p(MC|w_C)$  を用いて

$$\log \frac{p(MC|w_C)}{p(\overline{MC}|w_C)} \quad (2)$$

で計算される。ここで  $\overline{MC}$  は  $MC$  の排反事象である。

条件付き確率  $p(MC|w_C)$  を学習データから推定する。いま、 $f(w_C)$  を、学習データ中で  $w$  が  $C$  に出現した頻度とする。同様に、 $f(w_C, MC)$  を、学習データ中で  $w$  が  $C$  に出現したときにターゲット名詞が  $MC$  となった頻度とする。このとき、条件付き確率を、

$$p(MC|w_C) = \frac{f(w_C, MC) + 0.5}{f(w_C) + 1.0} \quad (3)$$

で推定する。

### 2. 4 一般判定規則の優先度算出

一般判定規則はその学習方法から、多種類の名詞と共起する事が容易に想像できる。ある規則がより多くの名詞と共起することで、その規則は一般的に名詞を可算／不可算にする、一般性の高い規則であると言える。しかし一方で、偏った名詞にしか共起しない単語は一般性が低いとい

うことになる。そこで、本論文では一般判定規則の一般性を評価するため、新たに一般性  $G(r)$  を考える。  $G(r)$  は規則  $r$  の一般性の高さを示す。一般判定規則の最終的な優先度は

$$\log \frac{p(MC | w_c)}{p(\overline{MC} | w_c)} \cdot G(r) \quad (4)$$

で評価することとする。

#### 2. 4. 1 一般性評価の方法

$G(r)$  を算出する方法として下記 4 つを提案する。

##### (a) 共起名詞種類数

単純に規則  $r$  が学習データ中で共起した名詞の種類数を考えたものである。この場合  $G(r)$  は  $N$  を学習データ中の総名詞種類数、  $f(r)$  を規則  $r$  が共起した名詞の種類数として

$$\frac{f(r)}{N} \quad (5)$$

で計算される。

##### (b) 指数共起名詞種類数

(a) において、一般性をどれだけ重視するかを考慮したものである。  $C$  を定数として

$$\left( \frac{f(r)}{N} \right)^C \quad (6)$$

この評価方法では  $C$  の値により、一般性をどれほど重要視するか決めることが出来る。本論文では  $C=0.5$  とした。

(c) 規則の出現割合

$$\frac{1}{\log \left( \frac{N}{f(r)} \right) + 1} \quad (7)$$

(d) エントロピー

$P(t | r)$ を規則  $r$ がターゲット名詞  $t$ と共起する条件付き確率とし、学習データ中の全ての名詞  $t$ に対して

$$- \sum_{t \in T} P(r | t) \log P(r | t) \quad (8)$$

を計算する。  $T$ は学習データ中に含まれるターゲット名詞の集合を示す。

## 2. 5 限定判定規則のデフォルト規則

デフォルト規則とは、判定規則中の他の規則によって可算／不可算の判定が行えないときに使用される規則である。いま、ターゲット名詞を  $t$  で表すことにする。また、ターゲット名詞を限定した場合の学習データ中で、頻度が高い方の  $MC$  の値を  $MC_{major}$  で表す。このとき、デフォルト規則のテンプレートは

$$t \rightarrow MC_{major} \quad (9)$$

と定義される。これは「ターゲット名詞が出現したら頻度の高い方の  $MC$  で判定」と解釈できる。

一般判定規則でもデフォルト規則は学習されるが、本論文では一般判定規則のデフォルト規則は扱わないものとした。

### 3. 可算／不可算の判定

ターゲット名詞の可算／不可算の判定は、2. で学習した一般判定規則と限定判定規則を組み合わせて行う。図2にターゲット名詞を *chicken* としたときの可算／不可算の判定の流れを示す。

まず、ターゲット名詞である *chicken* の限定判定規則の集合から、対数尤度比の高い順に適用可能な規則を検索する。この時点で適用可能な規則があればその規則に従って判定を行う。ただし、適用可能な規則がデフォルト規則のみである場合、この時点ではデフォルト規則は使用しない。限定判定規則に適用可能なものが無い場合は、一般判定規則から適用可能な規則の検索を行う。この時、限定判定規則のデフォルト規則よりも対数尤度比の低い規則は検索対象外とした。この時点でも適用可能な規則が無い場合は限定判定規則のデフォルト規則による判定を行う。図2では、限定判定規則には適用可能な規則が無く、一般判定規則の own[NP] → 可算が適用可能であることが分かるので、この時点で規則の適用を止め、可算と判定する。

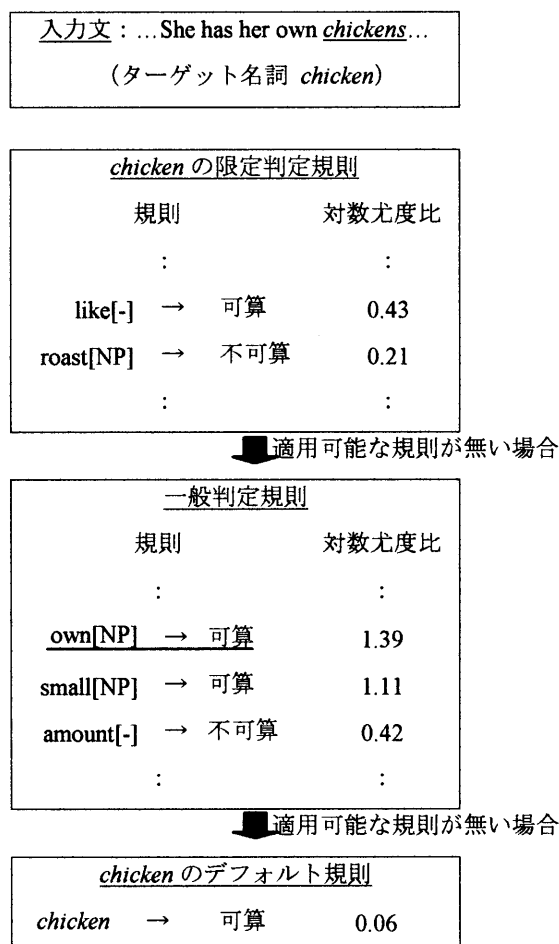


図2 規則適用の流れ

## 4. 実験と評価

この章では一般判定規則と限定判定規則を組み合わせた方法による可算／不可算の判定実験に関して説明する。以下4. 1で実験対象を含めた実験条件について述べる。4. 2で評価結果とそれに対する考察を行う。

### 4. 1 実験条件

本実験では、文献[3]に可算／不可算の両方で使用される名詞として上げられている23種類の名詞をターゲット名詞とした。

コーパスには、BNCを用いた。BNC中のテキストタグで囲まれた部分を一つのテキストとして使用した。ただし、話し言葉のタグが付与されているテキストは除外した。また、長すぎるために、実験に用いたツールで解析できなかった文も除外した。

本実験では、精度を用いて提案手法の判定性能を評価した。精度は

$$\frac{\text{正しく判定されたターゲット名詞の数}}{\text{判定したターゲット名詞の数}} \times 100 \quad (10)$$

で定義した。

本実験では、tenfold cross-validation[5]を用いて、提案手法の性能を評価した。まず、上記コーパス中のテキストから10セットに分割した。ただし、分割はランダムに行い、各セットに含まれるテキストの数がほぼ等しくなるように行なった。この結果、10セットの合計コーパスサイズは約7400万語となった。次に、10セットのうち、1セットを評価データ用のコーパス、残り9セットを学習データの生成、限定判定規則、一般判定規則の学習に使用し、1セット中のターゲット名詞の可算／不可算を判定し性能を評価した。

使用する一般判定規則は、2. 3で述べた(a)～(d)それぞれで一般性を評価したものに加え、一般性を評価しないもの計5つを使用した。

### 4. 2 評価結果と考察

表3に、評価結果を示す。表3中の“頻度平均”とは、評価データ中のターゲット名詞の平均出現数を表す。また、“BL”はターゲットを限定した場合の判定規則におけるデフォルト規則のみで判定したときの精度を示している。“a.”～“d.”までがそれぞれ2. 4で説明した一般性を考慮し

た一般判定規則を用いたものである。”e.”は一般性を考慮しない一般判定規則を用いて判定したものである。

表 3 評価結果

名詞	頻度平均 (回)	BL (%)	限定判定規則のみ (%)	a. (%)	b. (%)	c. (%)	d. (%)	e. (%)
advantage	628.7	61.3	89.8	88.8	88.8	88.8	88.8	<b>88.8</b>
Aid	581.1	80.4	87.0	88.9	88.9	88.9	88.7	<b>88.9</b>
Auth	1664.0	74.8	80.3	81.0	81.0	81.0	81.0	<b>81.0</b>
building	924.7	78.0	79.7	79.7	79.7	79.7	79.7	<b>79.7</b>
cover	227.8	63.8	72.9	73.9	73.8	74.6	74.4	<b>74.6</b>
detail	1301.0	74.0	88.5	88.6	88.6	88.6	88.6	<b>88.6</b>
discipline	229.9	61.3	76.1	76.2	76.2	76.3	76.6	<b>76.3</b>
Duty	638.0	67.0	79.5	80.6	80.6	80.6	80.6	<b>80.6</b>
football	207.1	92.9	94.0	94.4	94.5	94.5	93.2	<b>94.5</b>
Gold	252.0	92.8	92.8	92.8	92.8	92.9	91.5	<b>92.9</b>
Hair	422.0	86.3	87.9	88.3	88.4	88.4	88.0	<b>88.4</b>
improvement	427.0	72.5	75.6	75.8	75.8	75.8	75.8	<b>75.8</b>
necessity	93.1	53.9	80.1	80.5	81.5	81.4	81.1	<b>81.4</b>
paper	1028.6	59.2	82.2	82.5	82.5	82.6	82.5	<b>82.6</b>
reason	1351.8	83.0	84.9	85.9	85.9	85.9	85.9	<b>85.9</b>
sausage	49.5	78.2	77.4	74.3	74.3	74.3	74.2	<b>74.3</b>
sleep	152.0	84.4	87.8	88.5	88.5	88.4	86.2	<b>88.4</b>
stomach	38.7	66.8	72.4	72.3	72.3	72.1	72.2	<b>72.1</b>
study	1683.4	74.6	79.8	80.7	80.7	80.7	80.7	<b>80.7</b>
Truth	243.6	75.3	80.7	80.8	81.4	81.6	80.3	<b>81.6</b>
Use	1473.6	87.4	88.2	88.4	88.4	88.4	88.2	<b>88.4</b>
Work	3128.8	80.3	83.9	84.5	84.5	84.5	84.3	<b>84.5</b>
worry	122.2	79.6	84.1	84.6	84.6	84.7	84.3	<b>84.7</b>
平均	733.4	75.1	82.9	83.1	83.2	83.2	82.9	<b>83.2</b>

表 3 から、提案手法は、限定判定規則のみを用いたものよりも精度が良いことが確認出来る。また、“限定判定規則のみ”と“a~e.”の平均精度には有意水準 5% で有意差が見られた (paired *t*-test)。限定判定規則のみを用いた方法では、限定判定規則の中に適用可能な規則が無い場合、文脈情報を用いていないデフォルト規則による判定を行う。しかし表 3 の“BL”の示す通りデフォルト規則による判定では、文脈情報を用いた規則に比べ精度が低下する傾向にある。そこで本手法では限定判定規則に適用可能な規則が無かった場合、すぐにデフォルト規則を適用するのではなく、一般判定規則の検索も行う。そうすることで限定判定規則のみを用いた手法に比べ、文脈情報を用いた規則を増加させることができ、判定精度が向上した。例えば、ターゲット名詞を *worry* とした下記英文

...stage could solve small worries before they become...

では、限定判定規則に適用可能な規則が無く、一般判定規則の small[NP] → 可算が使用された。small は名詞句内に現れると、名詞を可算にする可能性が高い単語である。このことは、small[NP] → 可算の対数尤度比が 1.11 と高いことからわかる。この例のように、一般判定規則により、規則の不足が緩和され精度向上につながった。また、本手法で用いた一般判定規則は対象となる名詞を選ばないので、あらゆる名詞に対して適用できる事も限定判定規則のみを用いたものより優れている点であると言える。

次に一般判定規則についての考察を行う。一般性評価を行った”a.”～”d.”の判定精度と一般性評価を行わなかった”e.”の判定制度をみると、一般判定規則において一般性を考慮することで精度は大きく変化しないことが確認できる。一般性を考慮することは、例えば不可算性の高い名詞(”gold”など)と共起する単語が、可算性の高い名詞(”pen”など)の可算／不可算判定に用いられることを防ぐ意味で精度向上に繋がると考えられる。今回考慮した方法は、名詞の種類数に重きをおいた評価尺度であるため、名詞のカテゴリを考慮したものではない。名詞のカテゴリとは例えば beef や chicken を調理に使う名詞郡と捉える事である。

名詞のカテゴリ分けをするための方法として、ある名詞とその他の名詞の限定判定規則中に共通した規則の出現数を考慮することが挙げられる。具体的には共通した規則（以下、共通判定規則）が一定数（または一定割合）以上存在する名詞同士を同カテゴリとみなす方法である。その方法を図3と図4に示す。



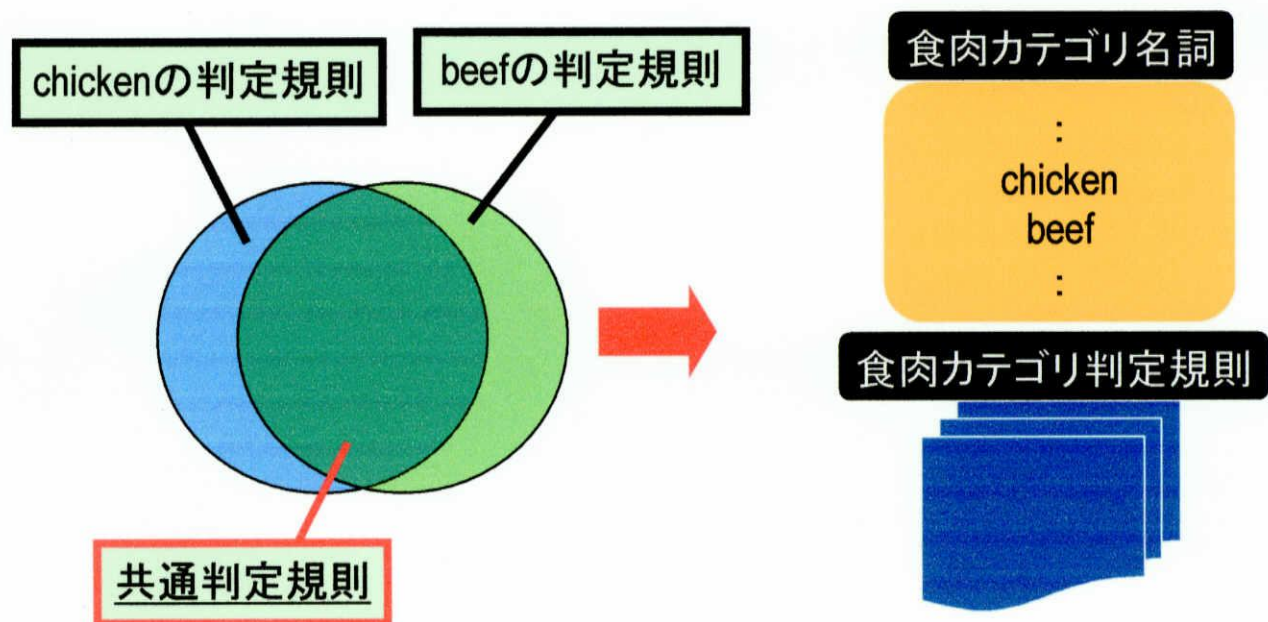


図3 名詞のカテゴリ判定方法

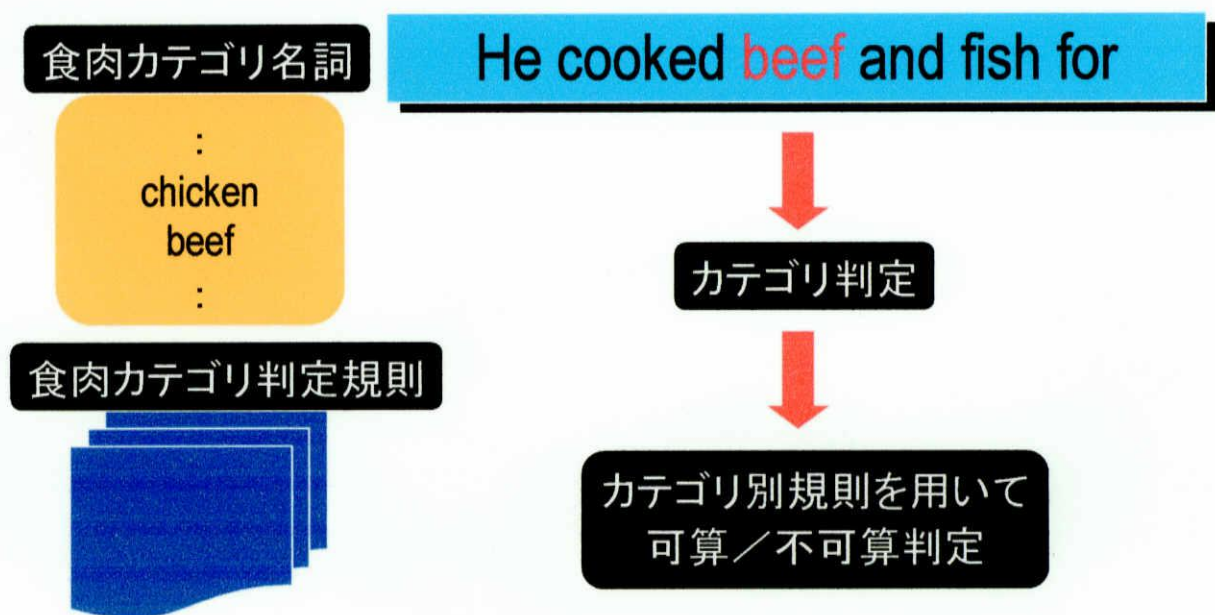


図4 カテゴリ別判定規則適用の流れ

名詞をカテゴリ分けした後，図3に示される共通判定規則をそのカテゴリで用いる判定規則とする．可算／不可算の判定を行う際は対象となる名詞のカテゴリを判別し，同カテゴリの判定規則を用いて判定を行う．このような方法をとることで，精度の向上が期待できる．この場合，未知名詞に関してはカテゴリ判定を行うことが難しい．よって未知名詞の場合は一般判定規則を用いるなどの対策が必要になる．

また、本研究では学習データとして BNC を用いたが、判定対象にする英文に応じて学習データを切り替えることで、判定精度向上が期待できる。例えば、医科学分野の論文に掲載された名詞の可算／不可算判定を行いたい場合は、同じく医科学分野の他の論文から獲得した学習データにより限定判定規則及び一般判定規則を学習することで BNC を用いた物より正確な判定が可能になると考えられる。

## 5. まとめ

本論文では、名詞の可算／不可算の判定を用いた冠詞誤り検出に関する従来研究の名詞の可算／不可算判定手法における規則の不足を補う手法を提案した。具体的には、ターゲット名詞を特定の単語に限定せず、コーパス中に出現する全ての名詞を1つの名詞として扱う事で、判定規則の数を増加させる。この規則をターゲット名詞を限定した規則と組み合わせて判定を行うことで、可算／不可算判定の平均精度を0.3%向上することが出来た。今後の課題としては、更なる判定精度の向上のために、一般判定規則の名詞カテゴリ毎の学習すること、学習データを判定対象別に切り替える事による判定精度の変化を調査すること挙げられる。また、本手法を用いた可算／不可算の判定を冠詞誤り検出[6]に応用することが挙げられる。

## 謝辞

日頃研究を進めるに辺り, ご指導頂いた兵庫教育大学の永田亮助手, 三重大大学の井須尚紀教授, 河合敦夫助教授, 榊井文人助手に大変感謝いたします。ありがとうございました。また, OAK Systemの開発者であるニューヨーク大学の関根聡氏に感謝致します。

## 参考文献

- [1] K. Allan, "Nouns and Countability," *Language: Journal of the Linguistic Society of America*, 56 (3) , pp. 541-567, Sep. 1980.
- [2] B. Gillon, "The Lexical Semantics of English Count and Mass Nouns," *Proc.of the Special Interest Group on the Lexicon of the Association for Computational Linguistics*, pp. 51-61, June 1996.
- [3] R.Huddleston and G.Pullum, *The Cambridge Grammar of the English Language*, Cambridge University Press, 2002.
- [4] R. Nagata, F. Masui, A. Kawai, and N. Isu, "An unsupervised method for distinguishing mass and count nouns in context," *Proc. of 6th International Workshop on Computational Semantics*, pp.213-224, Jan. 2005.
- [5] I. Witten and E. Frank,. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers. 2000.
- [6] 若菜崇宏, 永田亮, 河合敦夫, 榊井文人: "可算／不可算判定を用いた英文の冠詞誤り検出", 言語処理学会第 11 回年次大会論文集, 2005.3

## 関連文献

- [1] 若菜崇宏, 永田亮, 河合敦夫, 榊井文人: “可算／不可算判定を用いた英文の冠詞誤り検出”, 言語処理学会第 11 回年次大会論文集, 2005.3
- [2] Nagata,R., Wakana,T., Masui,F., Kawai,A. and Isu,N.: "Detecting Article Errors Based on the Mass Count Distinction" In Proceedings of the 2nd Int. Joint Conf. on Natural Language Processing, pp.815-826, 2005.10
- [3] 若菜崇宏, 永田亮, 河合敦夫, 榊井文人, 井須尚紀: “対象となる名詞を選ばない規則を用いた可算／不可算判定手法”, 第 5 回情報科学技術フォーラム(FIT2006)講演論文集, E-024, 2006.9

## 付録A. 可算／不可算の判定実験用環境

- ・主要プログラムと実験の方法について掲載する

### ○可算／不可算 判定実験ディレクトリ

/home/wakana/Natural/kenkyu/jikken/DISL/KNOWLEDGE/MC\_DET/

> (注意：マシン再起同時は、hda3 を/home/にマウントすること)

#### 1. 主要プログラム

##### ■bnc\_converter\_matrix\_v2.pl

###### 使用用途

>BNC を、1 行 1 行がある名詞とその文脈情報を示すものに変換する

(以下マトリックスデータ)

>マトリックスデータは主に下記情報を持つ

- ・表層情報により判定された MC
- ・周辺単語の情報
- ・及び上記周辺単語がその行の名詞と何回共起したか

eg. ...[NP a/DT small/JJ flower/NN]...

↓

flower:可算#-]... NP]small:1 +]...

###### 利点

>BNC の情報を圧縮して扱うことができる

###### 欠点

>元の英文に復元することが不可能

>評価用英文の作成には適さない

>評価用英文を作成する際は関連プログラムを参照されたい

###### 使用方法

>perl bnc\_converter\_matrix\_v2.pl -f [対象ディレクトリ名] [出力先ディレクトリ名]

###### 備考

- ・使用するBNCは

/home/wakana/Natural/kenkyu/jikken/DISL/KNOWLEDGE/TRAINING/

以下に存在する

##### ●関連プログラム

- ・merge\_matrix.pl

###### 使用用途

>作成されたマトリックスデータ同士の同一レコードをマージした

新たなマトリックスデータを作成する

>10fold cross validation で用いるデータセット毎のマトリックスデータを作成するときに使用する

### 利点

>学習処理の高速化

### 使用方法

perl merge\_matrix.pl [ファイル 1] [ファイル 2]… > 出力ファイル  
(ファイル1 ファイル 2… は directry/\* でも可)

### ・bnc\_converter\_dirFULL\_v2.pl

#### 使用用途

>BNC を、1 行 1 行がある名詞とその文脈情報を示すものに変換する

>マトリックスデータとの違い

- ・元の英文の情報を復元しやすい
- ・共起した単語の数量情報が無い
- ・評価用英文を作成可能

### 利点

>マトリックスデータほどでは無いが、学習処理の高速化、データの利用率が上げられる

### 欠点

>元の BNC よりデータが大きくなる可能性がある

>1 つの英文に複数のターゲット名詞が存在する場合  
違うレコードとして登録されるため

eg. ...cooked chicken and beef ...

↓

chicken:不可算 ...-]cook

beef:不可算 ...-]cook

### 使用方法

>bnc\_converter\_matrix\_v2.pl と同様である

### 備考

>評価用英文は現在これを用いて作成されている

### ■mc\_DLlist\_boost\_template\_fast\_wakana\_matrix\_v3.pl

#### 使用用途

マトリックスデータに対応した決定リストの作成を行う

### 利点

通常決定リスト作成プログラムよりも高速で実行可能

### 使用方法

perl mc\_DLlist\_boost\_template\_fast\_wakana\_matrix\_v3.pl [学習用データ] > 出力ファイル

ル

### オプション

-q [ターゲット名詞] 必須オプション

(一般モデルの場合は”を与えておく)

単数/複数形の正規表現を与える

例: flower の場合 -q 'flowers?'



**-d** [作成する決定リストのタイプ] 必須オプション (一般モデルの場合のみ)  
(特に指定しない場合はターゲット名詞モデル)

“general” : 一般モデル

> 出力に出現頻度の低い MC 規則の頻度も出力する場合 “\_add” とする  
例: general\_add

**-s** [決定リストの優先度算出方法] 必須オプション (一般モデルの場合のみ)  
(主に一般モデルを学習する際に用いる)

“default” : 対数尤度比を用いる

“var” : 単純ターゲット名詞種類数

“freq” : 指数ターゲット名詞種類数

“idf” : IDF に似た尺度

“ent” : エントロピー

## 2. 主要実験概要

・各学習モデルについて下記構造で実験環境が存在する (例: ターゲット名詞モデル)

```
eval_2_large_tar_TRA1_10-N/  
eval_2_large_tar_TRA1_10-1/  
:  
eval_2_large_tar_TRA1_10-10/
```

> 基本的に 10fold cross validation を行う目的でこの構成になっている

> ディレクトリ名に関しては 1 から 10 セットの内

[N 番目を評価用セット] とし [それ以外を学習データセット] としたものと定義

> 各ディレクトリの構成

- ・ 実験用プログラム
- ・ 各名詞毎の評価英文 (例: advantage.chk)
- ・ 各名詞毎の正解データ (例: advantage.dat)

各ディレクトリに含まれる主要プログラム

■ mc\_classifier\_wakana.pl

使用用途

> 作成された決定リストにより評価用データの MC 判定を行う

使用方法

> perl mc\_classifier\_wakana.pl [評価用データ] > 判定結果

オプション

**-q** [ターゲット名詞] 必須オプション

(判定対象の意味なので、一般モデルの実験であっても必ず指定すること)

**-f** [用いる決定リストファイル名] 必須オプション

(基本的に DBMS ファイルを用いる)

**-o** [出力タイプ] 必須オプション

(複数指定する場合は det\_com のように繋げること)

“det” : 判定の詳細を書き出す (ファイルに保存する場合は 2> ファイル名 とすること)

(実験の際は基本的に毎回つけること)

“com” : 判定数を書き出す

表示

default judge ... デフォルト規則を用いて判定した回数

rule judge ... 通常規則を用いて判定した回数

general\_rule judge ... 一般モデル規則を用いて判定した回数  
(混合モデル実験の際に用いる)

“step2” : 混合モデルによる判定を行う

>他に指定しない場合使用される一般モデルは  
ソース中の\$GENE\_LISTに記載されているものを用いる

>指定できる一般モデル規則は下記4つである

“\_var” : 単純ターゲット名詞種類数

“\_freq” : 指数ターゲット名詞種類数

“\_idf” : IDFに似た尺度

“\_ent” : エントロピー

>”\_noLLR”で一般性評価値を優先度とした一般モデル規則を使用する

#### 備考

・対象名詞 26 個分の判定を行うシェルスクリプトとして下記がある

classify\_1\_mil.sh ... ターゲット名詞モデル

classify\_1\_mil\_step2\_各種 ... 混合モデル

#### ■mc\_scoring.pl

##### 使用用途

>MC 判定されたデータと正解データと比べ、判定精度を評価する

##### 使用方法

>perl mc\_scoring.pl [判定データ] > 評価結果

##### オプション

-f [正解データ] 必須オプション

>判定対象に応じた.dat 拡張子のデータを使用する

このデータは make\_dat\_wakana.pl で作成可能

#### 備考

・対象名詞 26 個分の評価を行うシェルスクリプトとして下記がある

scoring1.sh ... ターゲット名詞モデル

scoring1\_step2\_各種 ... 混合モデル

・convert\_eval2csv.pl で評価結果を.csv 形式に変換可能

### 3. その他プログラム

#### ■mk\_statistics.pl

##### 使用用途

>評価結果の整理に用いる

**10fold cross validation** の評価結果をまとめたファイルを作成する

#### 利点

>実験結果分析の時間短縮

#### 使用方法

>perl mk\_statistics.pl [評価結果ファイル(.csv)] > 出力ファイル

## 付録 B. WEB 上のデモシステム

### ・可算／不可算の判定を用いた冠詞誤り検出システム 構成

・ mc\_detector.0.1\_wakana.cgi

> 誤り検出を行う perl プログラム

判定結果をクライアントに返す

> 用いる学習データはデフォルトで JACET8000\_kurdyla\_dl.dbm

> 学習データは WEB ページのフォームで選択する

> 用いる学習データはソースの dl\_file で指定する

> 今後学習データを追加する場合は適宜変更すること

### ・ WEB ページ

> index.html

学内公開用

> system.html

一般公開用

1. 検出手法

本システムは、検出対象の名詞の周辺の単語からその名詞が可算名詞であるか不可算名詞を判定します。その後、判定結果に基づき冠詞の誤りや単数・複数に関する誤りを検出します。可算名詞／不可算名詞の判定は大規模なテキストから自動抽出されたルールに基づいて行います。  
[Other System](#)

2. 英文入力

下のフォーム(左側)に英文を入力してください。 ※[英文入力時の注意](#)

I ate a chicken.]

I ate a **chicken**.

対象名詞: chicken  
対象名詞の前に **eat** が現れると不可算になるので**不可算名詞**と判定されました

3. 分野選択

学習データの分野を選択してください。  
デフォルトではBNCが選択されます。

☒ BNC ☐ セラミック ☐ 医科学 ☐ パターン認識

4. 検出開始

英文の最後にピリオド[.]がある場合、自動的に検出を開始します。検出結果は入力フォームの右に表示されます。

図 1 冠詞誤り検出システム WEB 画面

## 図1 解説

- ①：英文入力フォーム（文の終わりに必ず[.]（ピリオド）を入れること）
  - ②：学習データ分野選択（デフォルトでは本文で用いた BNC が選択されている）  
＞2007 年 2 月時点では、BNC 以外のデータは未公開
  - ③：処理結果出力フォーム  
＞誤りが含まれている部分は赤文字で示される
- ・ `ajax_lib.js`  
＞WEB ページで用いている `javascript` のライブラリ  
主にサーバとの非同期通信を行うために用いる（Ajax）

## 付録 C 一般判定規則に関する分析

本文で述べた一般判定規則を学習する際は BNC を 10 分割し、それぞれ 10 通りの一般判定規則を学習した。このとき、有力な一般判定規則は 10 通り中、何度も現れると考えられる。そこで 10 通りの一般判定規則の内、ある回数以上（ここでは 6 回以上）重複した規則を抽出した。その結果、6 回以上重複したのもので、文脈[NP]は 3524 件、文脈[-]は 5745 件、文脈[+]は 4881 件存在した。また、これらの規則の中でさらに順位を付けるために対数尤度とその規則の出現頻度を用いて

対数尤度 × 出現頻度

で順位付けを行った。各文脈（[-][NP][+]）の上位 10 件の規則を表 1 に示す。

表1 一般判定規則の重複規則

規則		重複数	対数尤度平均値	対数尤度*出現頻度平均値
other[NP]	→ 可算	10	1.13	42801.80
mr[NP]	→ 不可算	10	1.78	41462.04
different[NP]	→ 可算	10	1.54	22212.51
YEAR[NP]	→ 不可算	10	0.77	21298.01
large[NP]	→ 可算	10	1.26	17726.92
small[NP]	→ 可算	10	1.13	16933.19
such[NP]	→ 可算	10	0.64	14199.89
new[NP]	→ 可算	10	0.51	13731.14
most[NP]	→ 可算	10	0.99	12433.42
john[NP]	→ 不可算	10	1.48	12289.06
have[-]	→ 可算	10	0.32	47149.04
with[-]	→ 可算	10	0.17	23168.76
in[-]	→ 不可算	10	0.07	22241.54
as[-]	→ 可算	10	0.20	20555.88
NUM[-]	→ 不可算	10	0.22	20336.84
per[-]	→ 不可算	10	0.67	17097.34
number[-]	→ 可算	10	0.87	15161.13
for[-]	→ 可算	10	0.09	14013.22
of[-]	→ 不可算	10	0.02	13287.46
at[-]	→ 不可算	10	0.16	12422.81
off[+]	→ 可算	10	0.39	187203.20
in[+]	→ 可算	10	0.13	33527.57
YEAR[+]	→ 不可算	10	0.42	17445.13
for[+]	→ 可算	10	0.14	16375.02
NUM[+]	→ 不可算	10	0.18	15811.99
say[+]	→ 不可算	10	0.34	14407.57
from[+]	→ 可算	10	0.17	10763.72
with[+]	→ 可算	10	0.11	9567.40
ago[+]	→ 可算	10	1.13	8491.80
such[+]	→ 可算	10	0.43	8441.74

## 付録 D 修士論文発表資料



## 名詞の可算／不可算性を利用した 英文の冠詞誤り検出に関する研究

工学研究科博士前期過程情報工学専攻  
人工知能研究室 M2  
若菜 崇宏

## 研究の背景・目的

- 日本人英語学習者の書く英文に  
多く見られる誤りに以下の様な物がある

・冠詞の脱落

例: I have pen...

・冠詞の余剰

例: an information

・単数/複数の誤り

例: information s

これらの誤りを検出するには...

**名詞の可算/不可算の情報が重要**

## 可算/不可算情報の重要性

表1: 可算／不可算に基づいた誤り検出ルール

	単数			複数		
	a	the	φ	a	the	φ
可算	○	○	×	×	○	○
不可算	×	○	○	×	×	×

I have a furniture.

このfurniture が不可算と分かれば..

I have ~~a~~ furniture.

**冠詞の余剰**として検出可能

## 可算/不可算判定の難しさ

- 大部分の名詞は  
可算/不可算の**両方**で用いられる

I read a paper this morning.

「私は今朝新聞を読んだ」

→ **可算**

The paper is made of hemp pulp

「その紙はヘンプパルプで出来ている」

→ **不可算**

## 可算/不可算判定の難しさ

- 大部分の名詞は可算/不可算の  
両方で用いられる

可算／不可算辞書

I read a paper this morning.

「私は今朝新聞を読んだ」

paper = 不可算

→ **不可算**

単純に辞書を用いることは難しい

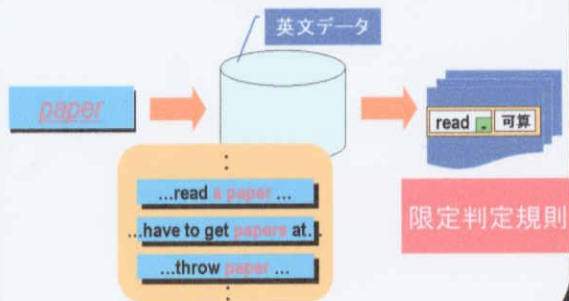
## 可算／不可算の判定方法

- 対象名詞の周辺単語を用いた判定 (永田ら 2005)

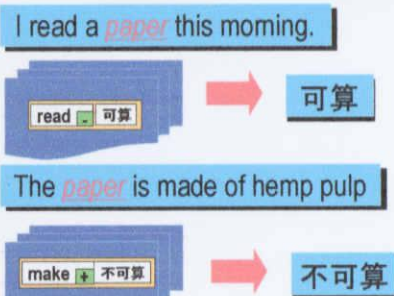
I read a paper this morning.

- ... 対象名詞の現れた **名詞句** の単語群
- np ... 対象名詞の現れた **名詞句** の単語群
- + ... 対象名詞の現れた **名詞句** の単語群

## 可算／不可算の判定方法



## 可算／不可算の判定方法



## 限定判定規則の問題点

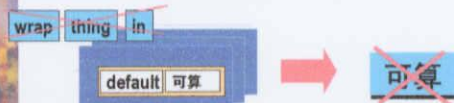
She wrapped a thing in paper.



規則が存在しない場合、default規則が適用される

## 限定判定規則の問題点

She wrapped a thing in paper.



default規則  
・文脈情報を用いない規則  
判定精度は低い

## 問題点の改善

She wrapped a thing in paper.



対象名詞によらない規則を学習  
することで規則を補う

## 問題点の改善

She wrapped a thing in paper.



対象名詞によらない規則を学習  
することで規則を補う



## 判定規則の学習

### 学習の流れ

1. 可算/不可算例の生成
2. 判定規則の生成
3. 判定規則の優先度の算出

## 可算/不可算例の生成



## 限定判定規則の生成

He cooked beef/不可算 and fish for dinner.  
 I ate a piece of chicken/不可算 with salad.

対象を chicken とした場合

## 限定判定規則の生成

He cooked beef/不可算 and fish for dinner.  
 I ate a piece of chicken/不可算 with salad.

eat - 不可算 --- piece np 不可算 --- salad + 不可算 ---

eat - 不可算 ---  
 eat が chicken の現れた名詞句から  
 前の単語群に現れたら不可算と判定

## 一般判定規則の生成

He cooked beef/不可算 and fish for dinner.  
 I ate a piece of chicken/不可算 with salad.

全ての名詞を同一と見なす

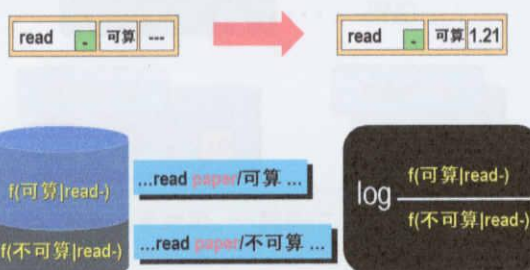
## 一般判定規則の生成

He cooked beef/不可算 and fish for dinner.  
 I ate a piece of chicken/不可算 with salad.

cook - 不可算 --- fish np 不可算 --- dinner + 不可算 ---  
 eat - 不可算 --- piece np 不可算 --- salad + 不可算 ---

eat - 不可算 ---  
 eat が名詞の現れた名詞句から  
 前の単語群に現れたら不可算と判定

## 規則の優先度を算出



## 規則の優先度を算出

一般判定規則の優先度

最終的な優先度 = 規則の元々の優先度  $\times G(r)$

一般性

## 規則の優先度を算出

一般判定規則の一般性を考慮



## 規則の優先度を算出

一般判定規則の一般性を考慮



## 規則の優先度を算出

一般性を評価する4つの指標

- 単純ターゲット名詞種類数
- 指数ターゲット名詞種類数
- 規則の出現割合
- エントロピー

## 実験条件

使用英文データ ... BNC

学習データ

1 2 ... 10

評価データ

1

paperを始め、一般的に可算/不可算の両方で用いられる名詞23個を対象名詞とする



## 実験条件

使用英文データ ... BNC

学習データ

1 2 ... 10

評価データ

2

paperを始め、一般的に  
可算/不可算の両方で用いられる  
名詞23個を対象名詞とする

## 実験条件

・評価方法

可算/不可算の判定精度

default規則の適用数

判定精度

$\frac{\text{正しく判定されたターゲット名詞の数}}{\text{判定したターゲット名詞の数}} \times 100$

## 実験手順

英文データから以下の規則を学習する

限定判定規則

- 単純ターゲット名詞種類数
- 指数ターゲット名詞種類数
- 規則の出現割合
- エントロピー  
一般性評価なし

一般判定規則

## 実験手順

一般判定規則の適用方法

She has her own chickens.

chickenの限定判定規則

一般判定規則

default

不可算

own 可算 1.39

可算

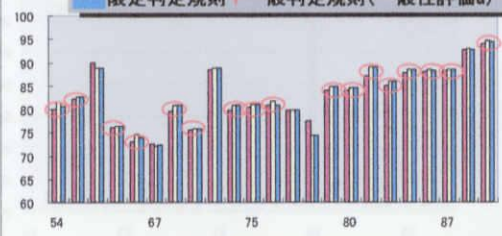
## 実験結果1: 判定精度

精度

限定判定規則

限定判定規則 + 一般判定規則(一般性評価なし)

限定判定規則 + 一般判定規則(一般性評価a)



対象名詞の可算不可算割合

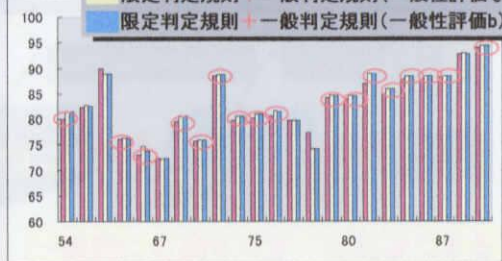
## 実験結果1: 判定精度

精度

限定判定規則

限定判定規則 + 一般判定規則(一般性評価なし)

限定判定規則 + 一般判定規則(一般性評価b)

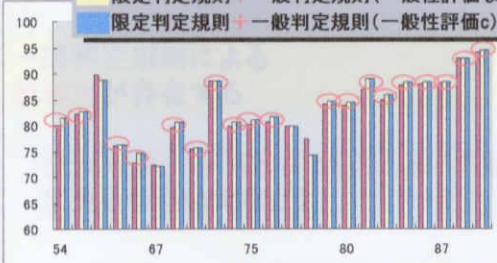


対象名詞の可算不可算割合

## 実験結果1: 判定精度

精度

- 限定判定規則
- 限定判定規則 + 一般判定規則 (一般性評価なし)
- 限定判定規則 + 一般判定規則 (一般性評価c)

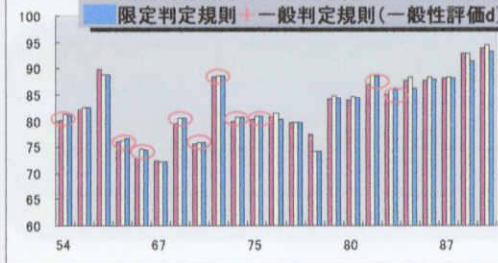


対象名詞の可算/不可算割合

## 実験結果1: 判定精度

精度

- 限定判定規則
- 限定判定規則 + 一般判定規則 (一般性評価なし)
- 限定判定規則 + 一般判定規則 (一般性評価d)



対象名詞の可算/不可算割合

## 実験結果1: 判定精度

表2a: 平均精度比較

限定判定規則	82.9
限定判定規則 + 一般判定規則 (一般性評価なし)	83.2
限定判定規則 + 一般判定規則 (一般性評価a)	83.1
限定判定規則 + 一般判定規則 (一般性評価b)	83.2
限定判定規則 + 一般判定規則 (一般性評価c)	83.2
限定判定規則 + 一般判定規則 (一般性評価d)	82.9

一般性を考慮することによる  
判定精度の変化は無い

## 実験結果1: 判定精度

表2b: 平均精度比較

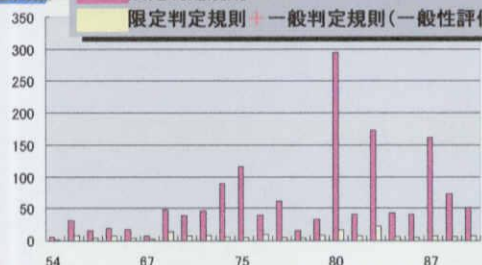
限定判定規則	82.9
限定判定規則 + 一般判定規則 (一般性評価なし)	83.2

対応のある2群間のt検定の結果  
有意水準5%で有意差が見られた

## 実験結果2: default規則適用数

適用数

- 限定判定規則
- 限定判定規則 + 一般判定規則 (一般性評価なし)



対象名詞の可算/不可算割合

## 考察

...could solve small **worries** before they become



一般的に名詞を可算/不可算にする  
規則を用いることが出来る



## 問題点

一般判定規則による  
誤判定が存在する

元々可算／不可算の偏りが強い名詞  
に見られる (gold など)

## 今後の課題

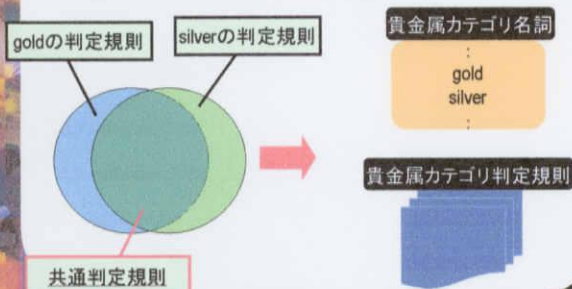
一般判定規則による誤判定の改善

本手法を冠詞誤り検出に応用

学習データ分野別の一般判定規則の調査

## 誤判定の改善方法

・名詞の**カテゴリ別規則**を学習する



## 誤判定の改善方法

・名詞の**カテゴリ別規則**を学習する



## まとめ

一般判定規則による  
英語名詞の可算／不可算判定の改善

一般判定規則により精度の向上を確認

冠詞誤り検出に応用した場合の調査

ご清聴ありがとうございました