可算／不可算判定を用いた英文の冠詞誤り検出

若菜　崇宏†　　永田　　亮†　　桝井　文人†　　河合　敦夫†
† 三重大学工学部情報工学科
E-mail: †{wakana,nagata,masui,kawai}@ai.info.mie-u.ac.jp

1. はじめに

日本人英語学習者の英文に多く見られる誤りとして，冠詞の脱落（e.g.，*I have pen(注 1)），冠詞の余剰（e.g.，*an information），単数／複数の誤り（e.g.，*informations）がある [6].

上記の誤り（以下，表記を簡単にするため，これらの誤りを冠詞誤りと呼ぶ）を検出するためには，英語名詞の可算／不可算の情報が重要となる [7]．なぜなら，名詞の可算／不可算の情報が与えられることによって，表 1 の「×」で示される部分が誤りであることが分かり，冠詞誤りが検出できるからである．たとえば “I have *pen.” という英文で，“pen” が可算名詞であると言うことがわかれば，表 1 から冠詞の脱落として検出可能である．

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比較的容易に可算／不可算の判定は行える．例えば，複数形の名詞は可算名詞であり，無冠詞単数の名詞は不可算名詞となる．

一方，誤りを含む英文では，冠詞及び単数／複数の用法が誤っている可能性があるため，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ためには，これらの表層情報を用いない可算／不可算の判定が必要となる．

これまでに，可算／不可算を判定する手法は多く提案されている（例えば，文献 [1], [3] など）．Bond ら [3] は，意味情報を用いて可算／不可算の判定を行う手法を提案している．しかしながら，これらの手法は，正しい英文を対象にしている手法，または，文中の情報のみを用いた名詞の可算／不可算の判定は行えない手法なので，冠詞誤りの検出に応用するには不十分である．

そこで本論文では，誤り検出対象の名詞（以後，ターゲット名詞と呼ぶ）の周辺単語(注 2)に基づいて可算／不可算の判定を行う手法を提案する．また，その判定結果を利用した冠詞誤り検出手法を提案する．可算／不可算の判定は，決定リストを用いて行う．提案手法では，コーパスから学習データを自動生成し，決定リストの学習を行う．

表 1：　名詞の可算／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	無冠詞
可算	○	○	×	×	○	○
不可算	×	○	○	×	×	×

(注 1)：本論文では，非文法な文に*を付けて表す．
(注 2)：ただし，ターゲット名詞周辺の単語に冠詞は含まない．

以下，**2.** で可算／不可算判定のための決定リストを学習する方法について説明する．**3.** では可算／不可算の判定を用いた冠詞誤りの検出手法について説明する．**4.** で実験について述べる．**5.** で実験結果を考察する．

2. 可算／不可算判定のための決定リスト

本章では，提案手法で使用する決定リストの学習方法について述べる．**2.1** で学習データの自動生成について説明する．**2.2** で決定リストの学習について説明する．なお，可算／不可算判定の詳細については文献 [9] を参照されたい．

2.1 学習データの自動生成

決定リストは，ターゲット名詞の可算／不可算の例から学習される．可算／不可算の例とは

She went to the field to pick $flowers$ ／可算.

の様に，ターゲット名詞に可算／不可算の情報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．

（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 went to the field to pick $flowers$.

中の $flowers$ は複数形である事から

She went to the field to pick $flowers$ ／可算.

とタグ付け出来る．

以下，言語学の知見 [1] [4] [5] 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の一部を示す．詳細は [9] を参照されたい．

タグ付けはターゲット名詞に以下の質問を適用することで行われる．

（A） ターゲット名詞が複数形であるか？
（yes）可算タグを付与し終了
（no）（B）へ

（B） 表 2 の（a）の語のいずれかによってターゲット名詞が修飾されているか？
（yes）可算タグを付与し終了
（no）（C）へ

（C） 表 2 の（b）の語のいずれかによってターゲット名詞が修飾されているか？
（yes）不可算タグを付与し終了
（no）（D）へ

（D） 表 2 の（c）の語のいずれかによってターゲット名詞が修飾されているか？
（yes）“?” タグを付与し終了
（no）不可算タグを付与し終了

“?” が付与されたターゲット名詞は，可算／不可算が不明なので，学習データには含めない．

（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．

2.2 決定リストの学習

決定リストは，規則(注 3)の集合からなる．各規則のテンプレートは，決定リストが対象としている問題によって異なる．提案手法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 mass（不可算），count（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），$\pm k$(その名詞句から左（−）または右（+）に k 単語) の 3 種類を定義する．このときテンプレートを

表 2： 学習データ生成ルールに使用される単語

(a)	(b)	(c)
the indefinite article	much	*the definite article*
another	less	*demonstrative adjectives*
one	enough	*possessive adjectives*
each	all	*interrogative adjectives*
—	sufficient	*quantifiers*
—	—	*'s genitive*

(注 3)：学習データの生成に用いたルールと区別するために，ここでは規則という言葉を用いる．

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

$$w_C \rightarrow MC \tag{1}$$

で表すことにする．

上記テンプレートに加え，デフォルト規則のテンプレートも定義する．デフォルト規則とは，決定リスト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，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を

$$t \rightarrow MC_{major} \tag{2}$$

と定義する．式 (2) 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．

次に **2.1** で説明した学習データを用いて決定リストの学習を行う．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．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．

以下に規則の生成例を示す．例えば，ターゲット名詞を *flower* としたとき

He has small *flowers* ／可算 in his hand.

からは

$$have_{-3} \rightarrow count, \quad small_{np} \rightarrow count, \quad hand_{+3} \rightarrow count$$

が生成される．ただし，k の値を 3 とした場合である．

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

$$log\frac{p(MC|w_C)}{p(\overline{MC}|w_C)} \tag{3}$$

で計算される．ここで $\overline{MC}$ は MC の排反事象である．式（3）は，単語 w が文脈 C に現れたとき，ターゲット名詞が MC になりやすいほど，また $\overline{MC}$ になりにくいほど高い値を示す．すなわち，w_C によって，ターゲット名詞がどの程度 MC になりやすいかを表す．

条件付き確率 $p(MC|w)$ 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，

$$p(MC|w_C) = \frac{f(w_C, MC) + 0.5}{f(w_C) + 1.0} \tag{4}$$

で推定する．

3. 冠詞誤りの検出手法

ターゲット名詞の可算／不可算の判定は，決定リスト中の規則を対数尤度比の大きい順に適用し，適用可能な規則が見つかった時点で，その規則に従って行う．例えば，以下のような文に対し（ターゲット名詞 *chicken*）と決定リストが与えられた場合，

I ate a piece of *chickens* with salad.

決定リスト

規則	対数尤度比
$piece_{-3} \rightarrow mass$	1.25
$peck_{+3} \rightarrow count$	1.12
$pig_{np} \rightarrow count$	1.03
:	:

$piece_{-3} \rightarrow mass$ が，適用可能である事が分かる．従って，この時点で規則の適用をやめ不可算と判定する．

誤り検出は以下の 2 つ手順に分けて行う．2 つの手順のうち，少なくとも 1 つが誤りと判定した場合に冠詞誤りとして検出する．1 つ目の手順は可算／不可算の情報とターゲット名詞の表層情報を用いて表 1 から冠詞誤りを検出する．上記例の *chickens* は不可算と判定され，表層情報から無冠詞，複数であることがわかる．ここで表 1 を用いると「不可算名詞が無冠詞，複数で用いられることは誤りである」という誤りとして検出できる．2 つ目の手順は，名詞の可算／不可算の情報，単数／複数の情報，更にターゲット名詞を修飾している語を用いた誤り検出ルール（表 3，表 4）を用いて誤りを検出する．例えばターゲット名詞が可算と判定され，単数形だったとする．このときターゲット名詞が much に修飾されている場合，表 3 から,「可算名詞単数であるターゲット名詞が，much に修飾されることは誤りである」と判定される．

4. 実験と評価

4.1 実験対象

本実験では，実験対象として日本人英語学習者の書いた英文エッセー [2] を用いた．この英文エッセーに対して英語ネイティブスピーカによる冠詞誤りチェックを行った．次にチェック後の英文のうち，誤りを含む 205 文を抽出した．205 文中に誤りを含む単語は 250 個存在した．その誤りを人手で修正した 205 文と修正前の 205 文とを合わせ 410 文を実験で使用する．以後，この 410 文をエラーセットと呼ぶことにする．エラーセット中には誤りを含まない名詞も存在するが，本実験では誤りを含む 250 個とそれを修正した 250 個合わせて 500 個をターゲット名詞として実験を行う事にする．

表 3： 可算／不可算と表層情報に基づいた誤り検出ルール

	可算		不可算	
	単数	複数	単数	複数
語群 A	×	○	×	×
語群 B	○	○	○	×
語群 C	×	○	○	×
語群 D	×	×	○	×
語群 E	○	×	×	×
語群 F	○	×	○	×

学習データ作成の際使用するコーパスには BNC [8] を用い，BNC 中のテキストタグで囲まれた部分を 1 つのテキストとして用いた．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，英文が長すぎるために，実験で用いたツールで解析できなかった分も除外した．最終的に使用したテキスト総数は 2879 個（約 8000 万語）となった．

4.2 実験方法

4.2.1 実験手順

2. で説明した手法を用いて，実験対象のターゲット名詞の決定リストを学習した（前処理としてスペルチェッカーによるスペルチェックを行った）．決定リストは文献 [9] にならい，文脈 C の k の値を 3 として規則を生成した．名詞句の抽出は OAK システム(注 4)を用いて行った．**3.** に示す手法により冠詞誤りを検出した．更に本手法の冠詞誤り検出性能の有効性を調査するために，市販の英文法の誤り検出ツール GrammarianProX(注 5)を用いて比較実験を行った．

4.2.2 評価方法

本実験では，2 種類の尺度を用いて冠詞誤り検出の性能を評価する．

第一の尺度では，冠詞誤りの検出率（Recall）を評価する．Recall とは，実験対象中の 250 個の冠詞誤りのうち，何割の誤りを検出できたかを表す．したがって，Recall を R で表すと，

$$R = \frac{\text{No. of errors detected correctly}}{\text{No. of errors in the test sentences}} \qquad (5)$$

と定義される．

第二の尺度では，冠詞誤りの検出精度（Precision）を評価する．Precision では，誤りとして検出された名詞のうち，何

表 4： 表 3 で用いる語群

語群 A：	one を除く数詞, few, many, several, both numerous, countless, various
語群 B：	some, any, no
語群 C：	a lot of, lots of, enough, sufficient
語群 D：	much
語群 E：	either, neither, one, each
語群 F：	every, a little, little

(注 4)：OAK System Homepage: http://nlp.cs.nyu.edu/oak/
(注 5)：GrammarianProXVer1.5
(http://www.mercury-soft.com/jp/index.html)

割が実際に誤りであったかを評価する．よって，Precision を P で表すと，

$$P = \frac{\text{No. of errors detected correctly}}{\text{No. of errors detected}} \quad (6)$$

と定義される．

4.3 評価結果

表 5 に，本手法と GrammarianProX との性能の比較結果を示す．GrammarianProX は冠詞誤り以外の誤りも対象にしているが，表 5 には冠詞誤りだけを対象とした場合の性能を示す．また，本手法の可算／不可算の判定精度は 87%であった．

表 5： 評価結果

Method	Recall	Precision
本手法	0.59	0.84
GrammarianProX	0.13	1.00

5. 考　　察

表 5 を見ると，本手法では，GrammarianProX に比べ Recall が 4 倍以上良いことが分かる．本手法では可算／不可算の判定精度は 87%と高く，不定冠詞の余剰や複数形の s の脱落などをうまく検出することが出来た．一方，GrammarianProX では不定冠詞の余剰，複数形の s の脱落はほとんど検出されなかった．さらに本手法は Precision に関しても高く，総合的な性能は本手法の方が優れていることが分かる．

実験結果から，本手法の冠詞誤り検出の有効性が確認されたが，可算／不可算情報を用いた本手法でも，エラーセット中の約 4 割の冠詞誤りは検出できなかった．以下，検出できなかった原因を考察する．

文脈に応じた冠詞の選択；単複の選択誤りで検出に失敗したものは原因の約 5 割を占めた．この種の誤りを検出するためには，文脈の考慮が必要な場合が多い．しかしながら，本手法では文脈情報は考慮していないため，この種類の誤りを検出できない．

実験に用いた解析ツールのミスによって検出に失敗したものが 2 割弱存在した．例えば，名詞句がうまく解析されないと誤りを検出できない．

名詞の可算／不可算判定がうまくいかずに検出が失敗してしまったものは約 2 割存在した．可算／不可算判定が失敗してしまう主な要因は 2 つある．1 つ目は決定リスト中に適用可能な規則が存在しても，規則となる単語がターゲット名詞から遠すぎる場合である．ターゲット名詞から規則となる単語までは名詞句から左右 3 単語までと設定したが，4 単語以上の位置に規則となる単語が存在した場合，規則として利用できない．2 つ目は決定リスト中に適用可能な規則が存在しない場合である（この他 3 単語だけだが決定リストそのものが学習できなかったものも存在した）．例えば “homestay” はコーパス中に現れず，決定リストが学習されなかった．1 つ目の要因に関しては，考慮する文脈を広げて（k の値を大きくして）学習すれば改善できる可能性がある．しかしながら，文脈を広くしすぎるとターゲット名詞に関連の薄いものまで規則として学習してしまい，精度が低くなってしまう可能性がある．2 つ目の要因に関してはコーパスサイズを更に大きくすることで改善できる可能性がある．

6. おわりに

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手法を提案した．実験の結果，学習者の書いた英文に対する決定リストによる可算／不可算の判定精度は 87%と非常に高いことを確認した．また，本手法の可算／不可算判定を用いた冠詞誤り検出は検出率 0.59，検出精度 0.84 であり，冠詞誤り検出に有効であることも確認された．

今後の課題としては，他の学習者コーパスを用いての冠詞誤り検出実験を行い，性能の幅を調査することが挙げられる．

謝辞 “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] 朝尾幸次郎，第二言語習得者のための英語学習者コーパスの構築とその利用，科学研究費補助金（基盤研究 B）研究成果報告書，2000.

[3] F. Bond and C. Vatikiotis-Bateson, “Using an Ontology to Determine English Countability, ” Proc. of 19th International Conference on Computational Linguistic : COLING–2002, pp. 99–105, Taipei, Taiwan, Aug. 2002

[4] 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.

[5] R. Huddleston and G. Pullum, *The Cambridge Grammar of the English Language*, Cambridge University Press, 2002.

[6] 和泉絵美，齋賀豊美，T. Supnithi，内元清貴，井佐原均，“エラータグ付き日本人英語学習者発話コーパスを用いた学習者の冠詞習得傾向の分析,” 言語処理学会第 9 回年次大会発表論文集，pp.19–22，Mar. 2003.

[7] 河合敦夫，杉原厚吉，杉江　昇，“英文の誤りを検出するシステム ASPEC-I,” 情処学論，vol.25，no.6，pp.1072–1079，Nov. 1984.

[8] L. Burnard ed., Users reference guide for the British National Corpus. version 1.0, Oxford University Computing Services, Oxford, 1995.

[9] R. Nagata, F. Masui, A. Kawai, and N. Isu, “An unsupervised method for distingyishing mass and count nouns in context,” Proc. of the Sixth International Workshop on Computatinal Semantics IWCS-6, Tilburg, The Netherlands, pp. 213–224, Jan. 2005.