# 検索エンジンとコーパスを利用した英文の名詞語彙誤り検出の一手法

玉城　悠仁† 　新妻　弘崇†† 　太田　学 †††

† 岡山大学工学部情報系学科　〒 700–8530 岡山県岡山市北区津島中三丁目 1 番 1 号
†† ,††† 岡山大学大学院自然科学研究科　〒 700–8530 岡山県岡山市北区津島中三丁目 1 番 1 号
E-mail: †pj2d3woa@s.okayama-u.ac.jp, ††niitsuma@cs.okayama-u.ac.jp, †††ohta@de.cs.okayama-u.ac.jp

あらまし　英語を母語としない日本人が英文を執筆する際，英文の誤りを発見するのは一般的に困難である．この問題を扱う研究として，検索エンジンを用いて英文の名詞や冠詞などの誤りを検出する研究がある．本稿では，検索エンジンと英語版 Wikipedia コーパスを利用して名詞と名詞以外の語の共起の強さを調べ，文脈と共起しにくい名詞を名詞語彙誤りとして検出する方法を提案する．本稿では，検索エンジンのみを用いる方法，英語版 Wikipedia コーパスのみを用いる方法，検索エンジンとコーパスを併用する方法を用いて，それぞれで名詞語彙誤り検出を行い，検出性能を評価する．KJ コーパスの英文 39 文を対象に名詞語彙誤り検出をしたところ，検索エンジンのみを用いる方法の F 値は 0.4810，英語版 Wikipedia コーパスのみを用いる方法の F 値は 0.4301 となり，これは先行研究である宮城らの F 値 0.3889，牧野らの F 値 0.3913 よりも高かった．

キーワード　検索エンジン，英文誤り検出，名詞，語彙，Wikipedia

## 1. はじめに

現在，日本人の多くが第二言語として英語を学んでいる．しかし，英語を母語としない日本人にとって正しい英語の使い分けは難しく，日本人の英作文には様々な誤りがしばしば見られる．その中でも名詞句に関する誤りは特に多い．日本人英語学習者のコーパスである Konan-JIEM Learner Corpus Third Edition（KJ コーパス）[1] には日本人大学生によって書かれた 233 件の英作文が収録されており，その誤りを種類ごとに分類すると，名詞句に関する誤りが約 4 割を占めている [2]．この結果を踏まえ本稿では，名詞に関する誤りの一つである名詞の語彙誤りを検出する手法を提案する．

英文中の誤りを検出するための一つの方法として検索エンジンを用いる方法がある．宮城ら [3] は検索エンジンによって得られる検索結果数を用いて，英文中の名詞誤りを検出及び修正する手法を提案した．宮城らの手法では，冠詞誤りと名詞の単複誤り，名詞の語彙誤りを扱い，冠詞誤りと名詞の単複誤りでは検出と修正，名詞の語彙誤りについては検出のみを行った．冠詞誤りと名詞の単複誤りの検出と修正では，検出された冠詞の用法と，検索エンジンによる検索結果数を用いた．名詞の語彙誤りの検出では，検索エンジンによる検索結果数を用いて，名詞同士の共起の強さを算出した．また，牧野ら [4] は宮城らが提案した名詞語彙誤り検出を拡張し，名詞と形容詞との共起を考慮した名詞語彙誤り検出法を提案した．

本稿では宮城らと牧野らが行った名詞語彙誤り検出を改良し，名詞と他品詞の共起の強さを考慮した名詞語彙誤り検出手法を提案する．また，本手法では共起の強さの算出に検索エンジンだけでなく英語版 Wikipedia コーパスも利用する．

本稿は次のような構成となっている. 2. 節で，関連研究について述べ，3. 節で，宮城らと牧野らの名詞語彙誤り検出手法について説明する．4. 節では，提案する検出手法について説明する．5. 節では，評価実験の内容と結果を示すと共に，その結果について考察する．6. 節では，本稿のまとめと今後の課題について述べる．

## 2. 関連研究

宮城ら [3] は冠詞誤りと名詞誤り，牧野ら [4] は名詞語彙誤り，尾崎ら [5–7] は冠詞誤り，有冨ら [8]，久保田ら [9] は前置詞誤り，谷本ら [10, 11] は動詞やコロケーション誤りを対象に，検索エンジンを用いて誤りを検出，修正する手法を提案した．以下では，本稿で対象とする名詞に関する誤りの検出と修正，また，検索エンジンを用いた誤りの検出に関連する研究を紹介する．

### 2.1 英文の名詞に関する誤りの修正

Berend ら [12] は CoNLL-2013 [13] の Shared Task である文法誤り修正のタスクに参加し，英語の名詞の単複誤りや冠詞誤りを解決するための教師付き学習システムを開発した．さまざまな修正候補を提示するために，機械学習ライブラリである MALLET API [14] を用いた最大エントロピーに基づく教師付き分類モデルを採用した．さらに，LFG(語彙機能文法) parser の解析によって得られる文法機能の階層構造に基づいて, より深い言語的な特徴の調査を試みた．その結果，LFG parser の出力から得られる名詞の種類 (普通名詞/固有名詞/代名詞) や代名詞の種類 (人称代名詞/再帰代名詞/所有代名詞) などに関する特徴が名詞の単複誤りの特定に有用であることを示した．彼らは，より有用なシステムを構築するために，Web ページ上の約 1 兆語からなる Google N-gram Corpus のような大規模コーパスの統計情報の利用も計画した．

### 2.2 検索エンジンを用いた誤りの検出

#### 2.2.1 フレーズ検索を用いた誤りの検出

英文中の前置詞や冠詞などが適当であるか検索エンジンを用いて検討するシステムが，大鹿ら [15] や網嶋ら [16]，平野

ら [17] によって開発されている．大鹿らのシステムの機能の一部には，語彙選択誤りの検出と修正が実装されている．このシステムでは，例えば英文のフレーズ “the result of the election” について検討する場合に，フレーズ中で調べたい日本語訳として，“結果” を入力すると，EDR の日英対訳辞書を用いて，入力した日本語の訳語候補を複数提示する．提示された訳語候補には概要説明の情報も付随しており，それらの情報を踏まえて，ユーザは適切な訳語を選択できる．さらに選択したそれぞれの訳語候補についてフレーズ検索を行い，どの訳語候補を使った英文が検索結果に多いか表示される．これは検索結果数を示すことで訳語の判断をユーザに委ねるシステムといえるが，ある程度の英語の知識がない人にとっては使いにくいという結論であった．

#### 2.2.2 MI スコアを用いた誤りの検出

谷本ら [10,11] は，検索エンジンを用いて英文中の動詞誤りを検出するシステムを提案した．このシステムは，検討したい英文中のフレーズを検索することで検索結果から妥当なフレーズを調べることができる．動詞の誤りを主語-動詞の一致に関する誤り（一致誤り），時制に関する誤り（時制誤り），語彙選択の誤り（語彙誤り），その他の四つに分類し，そのうち一致誤り，時制誤り，語彙誤りを検出する．検出方法は，それぞれの誤りの種類により異なる．一致誤りに関しては品詞とチャンク情報からルールに基づいて誤りを検出し，ルールに基づく検出が難しい場合は検索エンジンから得られる検索結果数を比較する．時制誤りの検出は，動詞の時制が異なる複数の検索クエリを生成し，それらの検索結果数を比較することによって時制誤りであるかを判定する．語彙誤りの検出は動詞と名詞からなる検索クエリの検索結果より，式 (1) で定義する MI スコアを算出し，その値が閾値に満たない場合，動詞の語彙誤りを検出する．

$$\text{MI スコア} = \frac{\text{共起頻度} \times \text{コーパス総語数}}{\text{共起語頻度} \times \text{中心語頻度}} \tag{1}$$

ここで，共起頻度はコーパス内で動詞と名詞が共起した回数であり，中心語頻度は誤りかどうか調べたい動詞の出現回数，共起語頻度はその動詞の前後の名詞の出現回数である．実験では，KJ コーパスの英文の一部を用いて評価し，検出の F 値は 0.155 であった．

## 3. 名詞の語彙誤り検出

共起の強さに基づく名詞の語彙誤り検出では，比較対象とする適切な名詞候補が前置詞等に比べて予測しにくい．つまり，比較するためには誤った名詞よりも適切な名詞を予め見つけなければならないが，そのような名詞を見つけることは一般に困難である．そこで，本稿では誤り検出対象の名詞とその名詞以外の語との共起の強さに基づいて，名詞の語彙誤りを検出する手法を提案する．3.1 節で検出する名詞の誤りの種類について説明する．3.2 節では語の共起の強さの定義を説明する．また，本稿の提案手法は宮城らと牧野らの手法を改良したものであるため，3.3 節で宮城らの，3.4 節で牧野らの名詞語彙誤り検出手法について説明する．提案手法は 4. 節で説明する．

### 3.1 名詞の誤りの種類

KJ コーパス中の名詞の誤りは名詞の単数形と複数形に関する誤り（単複誤り），適切な語句が使われていない，または名詞の欠落や余剰に関する誤り（語彙誤り），その他の誤り，の 3 種類に分類されている．このうち語彙誤りは，本来必要な名詞が欠落する誤り（欠落誤り），不要な名詞が挿入されている誤り（挿入誤り），名詞の綴りに関する誤り（綴り誤り），正しくは名詞だが他の品詞となっている誤り（品詞誤り），適切な語彙選択に関する誤り（語彙選択誤り）の 5 種類にさらに分類できる．

宮城ら，牧野らの手法および本稿の提案手法では，文章中の名詞のうち他の語との共起の強さが弱い名詞を誤りとして検出する．そのため，文中に誤りとなる名詞が現れない欠落誤りは検出できない．また，品詞誤りは構文解析器によって名詞以外にタグ付けされるとその語を誤りとして検出できない．したがって宮城ら，牧野らの手法および本稿の提案手法は，挿入誤り，綴り誤り，名詞とタグ付けされた語の品詞誤り，語彙選択誤りを検出の対象とする．

### 3.2 語の共起の強さ

宮城ら，牧野らの手法と本稿の提案手法では，誤り検出対象の語が他の語と同一文中に出現する頻度を用いて語の共起の強さを定義する．宮城らは名詞 2 語の共起の強さを式 (2) で，牧野らは名詞，代名詞，形容詞のいずれか 3 語の共起の強さを式 (3) で定義した．

$$\text{共起の強さ} = \frac{\text{A と B を同一文中に含む検索結果の数}}{\text{A と B の AND 検索で取得した検索結果の数}} \tag{2}$$

$$\text{共起の強さ} = \frac{\text{C と D と E を同一文中に含む検索結果の数}}{\text{C と D と E の AND 検索で取得した検索結果の数}} \tag{3}$$

ここで A と B はともに名詞であり，C，D，E は名詞，代名詞，形容詞のいずれかである．検索結果の数は Bing Search API [18] で取得した検索結果を利用して算出する．これが式 (2) および式 (3) の分母である．この検索結果は，最大 1,050 件まで取得可能であり，URL やスニペットなどの情報を持つ．その検索結果から得られるスニペットを解析し，A と B の 2 語，C と D と E の 3 語が一文中にともに出現しているかそれぞれ判定を行い，その頻度，つまり式 (2) と式 (3) の分子を求める．

本研究の手法では 2 語の共起の強さの算出に検索エンジンだけでなく英語版 Wikipedia コーパスも用いる．そのため本研究では共起の強さは式 (4) で定義する．

$$\text{共起の強さ} = \frac{\text{F と G を同一文中に含む文の数}}{\text{F または G の少なくとも一方を含む文の数}} \tag{4}$$

本稿で提案する手法では 2 語の共起の強さを算出するが，その 2 語は名詞，代名詞，形容詞，動詞，前置詞，ing 形のいずれかからなる．ここで ing 形とは現在分詞または動名詞を表す．

したがって式 (4) 中の F，G は名詞，代名詞，形容詞，動詞，前置詞，ing 形のいずれかである．検索エンジンを用いて共起の強さを算出する際は，Bing Search API によって F と G の AND 検索で得られたスニペットを解析する．英語版 Wikipedia コーパスを用いて共起の強さを算出する際は英語版の Wikipedia [19] の全記事から取得した英文を解析する．

### 3.3 宮城らの誤り検出手法

宮城らの誤り検出手法 [3] では，英文の先頭から順番に三つの名詞を選択して，その中の全ての組み合わせである 3 通りの名詞のペアについて式 (2) で共起の強さを求め，誤りを検出する．

まず，誤り検出対象の英文を MontyTagger [20] でタグ付けし，文中のすべての名詞を抽出する．例えば “In A, B gives me many C to meet D.” という文において，MontyTagger により A，B，C，D が名詞とタグ付けされたとする．宮城らの手法ではまず A，B，C について，A と B，B と C，A と C の 3 通りのペアについて Bing Search API の返すスニペットを解析し共起の強さを式 (2) で求める．そして， A=B，B-C，A-C のように一つの名詞のみが他の二つの名詞との共起の強さが小さい場合，その名詞，この例では C，を誤りとして検出する．ここで，「=」は共起の強さが閾値以上であること，「-」は共起の強さが閾値未満であることを示す．また閾値には KJ コーパスの英文 118 文を用いて算出された 0.098085 が用いられた．次に B，C，D についても同様にして 3 通りのペアの共起の強さを求め，同様にして誤りを検出する．このように，三つの名詞間での共起の強さを考慮して誤りを検出する．また，3 語の名詞の共起関係を扱うため名詞が 2 語以下しかない文については誤り検出ができない．

### 3.4 牧野らの誤り検出手法

牧野らの誤り検出手法 [4] では，英文の先頭から順番に名詞，代名詞，形容詞のいずれか三つを選択して式 (3) で共起の強さを求めるとともに，名詞については順番に関係なく 1 文中の全ての 2 語の組み合わせについて式 (2) で共起の強さを求め，誤りを検出する．

まず，誤り検出対象の英文を MontyTagger でタグ付けし，文中のすべての名詞，代名詞，形容詞を抽出する．例えば “In A, B gives me many C to meet D.” という文において，MontyTagger により A，B，C，D が名詞とタグ付けされ，me が代名詞とタグ付けされ，many が形容詞とタグ付けされたとする．牧野らの手法ではまず A，B，me について Bing Search API の返すスニペットを解析し共起の強さを式 (3) で求める．次に B，me，many について．その次は me，many，C というように先頭から順に 3 語ずつの組み合わせで同様にそれぞれ共起の強さを求める．また，1 文中の 3 語の共起の強さの平均も算出し，共起の強さが閾値と平均のいずれよりも小さい組み合わせを全て抽出する．3 語の閾値には KJ コーパスの英文 57 文を用いて算出された 0.016339 を用いた．続いて A と B，A と C，A と D，B と C などの全ての名詞 2 語のペアについて，それぞれ式 (2) で共起の強さを求める．また，1 文中の名詞ペアの共起の強さの平均も算出し共起の強さが閾値と平均のいずれよりも小さい組み合わせを全て抽出する．2 語の閾値には宮城らが使用した 0.098085 を用いた．3 語および 2 語の組み合わせの中で合わせて 2 回以上抽出された名詞を語彙誤りとして検出する．

## 4. 提 案 手 法

宮城らは 3 語の名詞の組み合わせ内の各ペアの共起の強さを調べ，その大小関係に基づいて誤りを検出した．また牧野らは名詞，代名詞，形容詞のいずれか 3 語での共起の強さを調べ，さらに名詞同士では同一文中の全てのペアについて共起の強さを調べることで誤りを検出した．本稿では誤り判定は宮城らの方法を採用し，牧野らが行ったように名詞と他品詞との共起の強さを考慮する．ここで他品詞とは代名詞，形容詞，動詞，前置詞，ing 形のいずれかである．以下，これらの品詞を “共起品詞” と呼ぶ．英文の先頭から順番に三つの名詞または共起品詞を選択して，その中の全ての組み合わせである 3 通りのペアについて式 (4) で共起の強さを求めて比較し，誤りを検出する．牧野らは名詞以外に代名詞や形容詞を考慮するため，名詞，代名詞，形容詞のいずれかからなる 3 語の共起の強さを調べた．しかし，この方法では 3 語内での名詞，代名詞，形容詞の数は考慮していない．つまり名詞 3 語からなる場合も名詞と代名詞と形容詞 1 語ずつからなる場合も同じ閾値で誤り判定を行っている．そこで提案手法では宮城らのように 2 語のペアで共起の強さを調べ，その品詞の組み合わせごとに閾値を設定して誤りを判定する．この方法では先頭から 3 語ずつの単語の組み合わせからペアを抽出するので，名詞との共起の強さを考慮する共起品詞の語が多いと離れた語のペアが抽出できない．そのため，先頭から 3 語ずつの組み合わせを考える際に一度に全ての品詞を用いるのではなく，名詞または代名詞のみでペアを抽出する場合や名詞と代名詞と形容詞のいずれかのみでペアを抽出する場合など，共起品詞の選び方も実験により比較する．

まず，誤り検出対象の英文を MontyTagger でタグ付けし，文中のすべての名詞，共起品詞を抽出する．例えば “In A, B gives me many C to meet D.” という文において，MontyTagger により A，B，C，D が名詞とタグ付けされ，me が代名詞とタグ付けされ，many が形容詞とタグ付けされたとする．名詞と代名詞の共起の強さを考慮する場合は，提案手法ではまず A，B，me について，A と B，B と me，A と me の 3 通りのペアについて共起の強さを式 (4) で求める．次に B，me，C についても同様にして 3 通りのペアの共起の強さを求める．名詞と形容詞，名詞と動詞，名詞と前置詞の共起の強さを考慮する場合も，同様である．そして共起の強さに基づいて宮城らと同様に，A=B，B-C，A-C のように一つの名詞のみが他の二つの単語との共起の強さが小さい場合，その名詞，この例では C，を誤りとして検出する．

本稿では共起の強さの算出やそれに対する閾値の設定の際に，検索エンジンだけでなく英語版 Wikipedia コーパスも用いる．4.1 節から 4.3 節でそれぞれについて説明する．また，提案手法で用いる閾値については 4.4 節で説明する．

### 4.1 検索エンジンを用いる方法

宮城らの手法と同様に Bing Search API を用いて 2 語で AND 検索を行い，スニペットを解析して 2 語の共起の強さを算出す

る．式 (4) を用いて，名詞または共起品詞のペアの共起の強さを算出する．そして品詞ペアの種類ごとに定めた閾値により，2 語の共起の強弱を判定する．また，共起の強さを算出する際，Bing Search API の返すスニペットを文ごとに分割する必要がある．スニペットでは複数の文が含まれていたり，一文が長すぎたりする場合に途中や末尾が省略されている．そのため一部が省略された不完全な文が含まれる．提案手法では共起の強さの算出に文の数を用いるのでこのような不完全な文は除く．一方，宮城ら，牧野らはこのような文を取り除いていない．

### 4.2 英語版 **Wikipedia** コーパスを用いる方法

Bing Search API によって最大 1,050 件の検索結果のスニペットを得られるが，スニペット 1 件に含まれる文の数は数文程度である．式 (4) で算出される共起の強さは 2 語が同一文中に現れる割合であるので，その分母が小さいと算出された値が適切でなくなる恐れがある．そこで，英語版 Wikipedia コーパスを用いることでその問題の改善を試みる．具体的には 2016 年 11 月 1 日時点での英語版 Wikipedia の記事データを整形し 32,918,629 文の英文を抽出した．これらの文には 14,523,541 種類の英単語が含まれている．この 32,918,629 文を検索する 2 語が含まれているかどうか解析し，式 (4) を用いて，名詞または共起品詞のペアの共起の強さを算出する．そして品詞ペアの種類ごとに定めた閾値により，2 語の品詞ペアの共起の強弱を判定する．ここで用いる閾値は英語版 Wikipedia コーパスを用いて算出するため，4.1 節で説明した検索エンジンの閾値とは異なる．

### 4.3 検索エンジンとコーパスを併用する方法

検索エンジンを用いる方法および英語版 Wikipedia コーパスを用いる方法のいずれにおいても解析する文章中に一度も現れない語の共起の強さは全て 0 と算出される．一般的でない人名や専門用語などの固有名詞は特にその傾向がある．そこで，検索エンジンと Wikipedia コーパスの一方で出現回数が少ない語を含むペアについては，もう一方で共起の強さを算出する．その際，検索エンジンと英語版 Wikipedia コーパスのどちらを先に用いるかによって 2 通りの方法がある．すなわち，“検索エンジンを優先する方法” と，“コーパスを優先する方法” である．検索エンジンを優先する方法では解析するスニペット中に 50 回上現れる語同士のペアについては検索エンジンを用いて共起の強さを算出し，そうでないペアについては英語版 Wikipedia コーパスを用いて共起の強さを算出する．また，英語版 Wikipedia コーパスを優先する方法では，英語版 Wikipedia コーパスの英文中に 100 回以上現れる語同士のペアについては英語版 Wikipedia コーパスを用いて共起の強さを算出し，そうでない単語については検索エンジンを用いて共起の強さを算出する．検索エンジンを用いる手法では最大 1,050 件の検索結果のスニペットを用いるのに対して，英語版 Wikipedia コーパスを用いる方法では 32,918,629 文の英文を用いるので単語の出現回数に大きな差がでる場合がある．本稿では，共起の強さの算出手段を変える単語の出現回数を，検索エンジンを優先する方法では 50 回，コーパスを優先する方法では 100 回とした．そのため検索エンジンを優先する方法では 50 回以上出現する単語を用いて閾値を設定し，コーパスを優先する方法では 100 回以上出現する単語を用いて閾値を設定した．

### 4.4 閾値の設定

#### 4.4.1 学習データ

本稿では閾値の決定のための学習データとして，KJ コーパス [1] の英文から 5. 節の実験でテストデータとして使用しない 99 文を用いる．この 99 文は名詞語彙誤りを含む英文であり，それ以外の誤りは含まない．この 99 文の中から，共起の強さを調べる名詞または共起品詞のペアを抽出し，式 (4) を用いてそれぞれの共起の強さを算出する．学習データ 99 文は名詞語彙誤りをそれぞれ一つ以上含む文であるので，抽出されたペアは正しい語同士のペア，正しい語と誤った語のペア，誤った語同士のペアの 3 種類となる．以下，正しい語と正しい語のペアを “正正ペア”，正しい語と誤った語のペアを “正誤ペア”，誤った語と誤った語のペアを “誤誤ペア” と呼ぶ．抽出したペアのうち，名詞を含むペアがこの 3 種類のどれにあたるか，ペアの品詞の組み合わせごとに表 1 にまとめる．

学習データ 99 文は名詞語彙誤り以外の誤りを含まないため，表 1 に示したように誤誤ペアは名詞と名詞のペアにしかなく，正正ペアや正誤ペアに比べ数が少ない．したがって誤誤ペアは誤り判定の閾値の設定には用いず，正正ペアと正誤ペアのみを用いて閾値を設定する．

#### 4.4.2 閾値の設定方法

4.4.1 項で述べた学習データを用いて名詞を含むペアの品詞の組み合わせごとに共起の強弱を決める閾値を決定する．閾値の候補は品詞のペアごとに九つ定めた．全体の平均値と中央値，正正ペアの平均値と中央値，正誤ペアの平均値と中央値の六つと，共起の強さの分布を表すグラフを用いて求めた三つの値である．この九つの候補のうち学習データでの誤り検出性能が最も良いものをテストデータでの閾値とする．以下で，共起の強さの分布グラフを用いて求めた三つの値について説明する．

正正ペアと正誤ペアの共起の強さの分布が正規分布であると仮定し，それぞれのグラフを描く．正正ペアと正誤ペアの共起の強さの分布の例を図 1 と図 2 に示す．図 1 は検索エンジンを用いて算出した名詞と名詞のペアの共起の強さの分布であり，2 は検索エンジンを用いて算出した名詞と動詞のペアの共起の強さの分布である．グラフ中の Correct pair は正正ペアを表し，Erroneous pair は正誤ペアを表す．

図 1 では正正ペアと正誤ペアの交点が一つしかないのでこの交点を閾値として，交点よりも右側を共起が強い，左側を共起が弱いと判定する．一方，図 2 では交点が二つある．そこで，

表 1 学習データから抽出されたペアの種類

| ペアの品詞 | 正正ペア | 正誤ペア | 誤誤ペア |
|---|---|---|---|
| 名詞と名詞 | 94 | 114 | 12 |
| 名詞と代名詞 | 71 | 61 | 0 |
| 名詞と形容詞 | 45 | 29 | 0 |
| 名詞と動詞 | 145 | 88 | 0 |
| 名詞と前置詞 | 93 | 53 | 0 |
| 名詞と ing 形 | 15 | 15 | 0 |

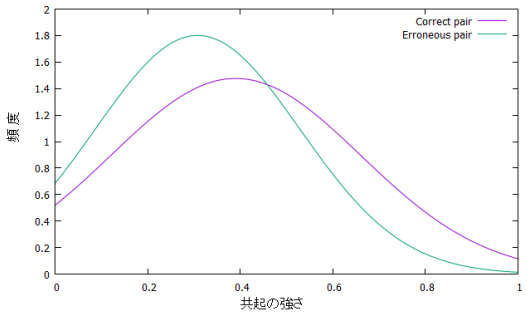


図 1　検索エンジンを用いた名詞と名詞のペアの分布

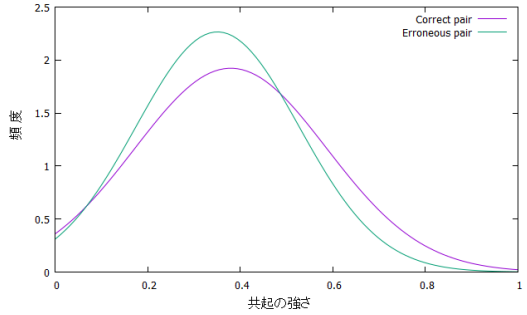


図 2　検索エンジンを用いた名詞と動詞のペアの分布

このような場合に左側の交点を閾値とするものと右側の交点を閾値とするもの，グラフの高い方に合わせるものの三つを考えた．これらを順に “正規分布グラフの左交点”, “正規分布グラフの右交点”, “正規分布グラフの高低” と呼ぶ．ここでグラフの高い方に合わせるパターン，すなわち正規分布グラフの高低とは，例えば図 2 の場合，共起の強さを $x$ とすると $x = 0.0673$ と $x = 0.4863$ がグラフの交点の $x$ 座標であり，その間では正誤ペアのグラフが上にあり，それ以外の範囲では正正ペアのグラフが上にある．そこで共起の強さが $0.0673 < x < 0.4863$ の範囲であればそのペアの共起が弱いと判定し，$x \leqq 0.0613, 0.4863 \leqq x$ の範囲であればそのペアの共起が強いと判定する．したがって，閾値の候補を改めてまとめると，全体の平均値と中央値，正正ペアの平均値と中央値，正誤ペアの平均値と中央値，正規分布グラフの左交点と右交点と高低，の九つとなる．

#### 4.4.3　閾　　値

4.4.2 項で述べた九つの閾値の候補について，検索エンジンを用いる方法，英語版 Wikipedia コーパスを用いる方法，検索エンジンとコーパスを併用する方法のそれぞれで学習データを用いて実験し，最も性能が良いものをテストデータに対する実験で使用する．テストデータに使用する閾値を表 2 にまとめる．最も性能が良いものの中に正規分布の高低を利用した閾値は無かったため表 2 の中の閾値以上である場合に共起が強いと判定し，その値未満であった場合に共起が弱いと判定する．

## 5.　評 価 実 験

名詞語彙誤りを含む英文を入力として与え，語彙誤りを検出できるかを実験により評価する．テストデータには KJ コーパス [1] から 39 文を選んだ．この 39 文は名詞を 3 語以上含んでおり，かつ名詞語彙誤りを含む英文である．また，この 39 文には合計で 48 語の名詞語彙誤りがあり，それ以外の誤りは含まない．また，3.1 節で述べたように欠落誤りや誤り箇所が名詞以外にタグ付けされる品詞誤りも含まない．

表 2　テストデータに用いる閾値

| 手法 | ペアの品詞 | 閾値 |
|---|---|---|
| 検索エンジンを用いる方法 | 名詞と名詞 | 0.345154 |
| | 名詞と代名詞 | 0.367584 |
| | 名詞と形容詞 | 0.397933 |
| | 名詞と前置詞 | 0.472251 |
| | 名詞と動詞 | 0.379121 |
| | 名詞と ing 形 | 0.378284 |
| 英語版 Wikipedia コーパスを用いる方法 | 名詞と名詞 | 0.000720 |
| | 名詞と代名詞 | 0.013382 |
| | 名詞と形容詞 | 0.025604 |
| | 名詞と前置詞 | 0.001103 |
| | 名詞と動詞 | 0.002004 |
| | 名詞と ing 形 | 0.012282 |
| 検索エンジンを優先する方法 | 名詞と名詞 | 0.073424 |
| | 名詞と代名詞 | 0.105848 |
| | 名詞と形容詞 | 0.162409 |
| | 名詞と前置詞 | 0.287286 |
| | 名詞と動詞 | 0.207169 |
| | 名詞と ing 形 | 0.249836 |
| コーパスを優先する方法 | 名詞と名詞 | 0.001721 |
| | 名詞と代名詞 | 0.004070 |
| | 名詞と形容詞 | 0.007764 |
| | 名詞と前置詞 | 0.012282 |
| | 名詞と動詞 | 0.002472 |
| | 名詞と ing 形 | 0.001164 |

本実験の検索エンジンには Bing Search API [18] を用いる．名詞語彙誤り検出の評価指標として，再現率 ($R$)，適合率 ($P$)，これらの調和平均の F 値を用いる．これらの指標は，それぞれ以下の式で表される．

$$\text{再現率}\,(R) = \frac{\text{語彙誤りとして正しく検出された名詞の数}}{\text{語彙誤りである名詞の数}} \quad (5)$$

$$\text{適合率}\,(P) = \frac{\text{語彙誤りとして正しく検出された名詞の数}}{\text{語彙誤りとして検出した名詞の数}} \quad (6)$$

$$\text{F 値} = \frac{2 \times R \times P}{R + P} \quad (7)$$

### 5.1　共起の強さを考慮する品詞

提案手法では先頭から 3 語ずつの単語の組み合わせからペアを抽出するため，2 語以上離れた語の共起の強さは考慮しない．名詞との共起の強さを考慮する共起品詞の語が多いと，離れた語の共起が考慮できなくなる．そこで，名詞とどの品詞の組み合わせが名詞語彙誤り検出に有効であるか検討する．まず名詞と高々 1 種類の共起品詞からなるペアを利用して，誤りを検出した結果を表 3 にまとめる．

続いて，それぞれの方法において表 3 で F 値が良かった 3 種類の共起品詞のうちの各 2 種類と名詞を組み合わせて誤りを検出する．また，それら 3 種類全てと名詞を組み合わせて誤りを

検出する．それらの結果を表 4 にまとめる．

表 3 と表 4 より，検索エンジン用いる方法では名詞と形容詞と ing 形の共起を利用するものの F 値が最も高く，英語版 Wikipedia コーパスを用いる方法では名詞と代名詞と前置詞の共起を利用するものの F 値が最も高かった．また，検索エンジンを優先する方法では名詞と形容詞の共起を利用するもの，コーパスを優先する方法では名詞と形容詞と動詞と前置詞の共起を利用するものの F 値が最も高いが，いずれも検索エンジンのみを用いる方法や英語版 Wikipedia コーパスのみを用いる方法に比べて低かった．

### 5.2 名詞語彙誤り検出の性能評価

ここでは提案手法による名詞語彙誤り検出の性能を評価する．提案手法について 5.1 節で説明した最も F 値が高かったそれぞれの実験結果を，既存システムである Grammarly [21] と比較する．Grammarly は，250 種類以上の文法ルールにより英文をチェックする機能を持ち，英文中の誤りに対して修正候補となる類義語等を提案してくれる．また，Grammarly は名詞だけでなく様々な品詞に対応している．さらに宮城ら，牧野らの手法とも検出性能を比較する．

Grammarly の検出結果を表 5 に，宮城らの検出結果を表 6 に，牧野らの手法による検出結果を表 7 にまとめる．また，提案手法について，検索エンジンを用いる方法，英語版 Wikipedia コーパスを用いる方法，検索エンジンを優先する方法，コーパスを優先する方法の検出結果をそれぞれ表 8，表 9，表 10，表 11 にまとめる．Grammarly および宮城ら，牧野らの手法に比べ，提案した検索エンジンを用いる方法の F 値は 10 ポイント程度高かった．一方，提案した検索エンジンを優先する方法とコーパスを優先する方法の F 値は低かった．

表 3　名詞と高々 1 種類の共起品詞のペアによる検出性能

| 手法 | 品詞の組み合わせ | 再現率 | 適合率 | F 値 |
|---|---|---|---|---|
| 検索エンジンを用いる方法 | 名詞のみ | 0.3750 | 0.5294 | 0.4390 |
| | 名詞と代名詞 | 0.3542 | 0.5312 | 0.4250 |
| | 名詞と形容詞 | **0.3958** | 0.5588 | **0.4634** |
| | 名詞と前置詞 | **0.3958** | 0.5278 | 0.4524 |
| | 名詞と動詞 | 0.3541 | 0.4857 | 0.4096 |
| | 名詞と ing 形 | 0.3750 | **0.5625** | 0.4500 |
| 英語版 Wikipedia コーパスを用いる方法 | 名詞のみ | 0.2291 | 0.5238 | 0.3188 |
| | 名詞と代名詞 | **0.3125** | 0.5172 | **0.3896** |
| | 名詞と形容詞 | 0.2083 | **0.5882** | 0.3076 |
| | 名詞と前置詞 | 0.2916 | 0.4666 | 0.3589 |
| | 名詞と動詞 | **0.3125** | 0.4687 | 0.3750 |
| | 名詞と ing 形 | 0.2500 | 0.5714 | 0.3478 |
| 検索エンジンを優先する方法 | 名詞のみ | 0.1875 | **0.4500** | 0.2647 |
| | 名詞と代名詞 | 0.1875 | 0.2812 | 0.2250 |
| | 名詞と形容詞 | **0.2708** | 0.4482 | **0.3376** |
| | 名詞と前置詞 | 0.1875 | 0.3333 | 0.2400 |
| | 名詞と動詞 | 0.2500 | 0.3750 | 0.3000 |
| | 名詞と ing 形 | 0.1875 | 0.4090 | 0.2571 |
| コーパスを優先する方法 | 名詞のみ | 0.1666 | 0.2666 | 0.2051 |
| | 名詞と代名詞 | 0.1875 | 0.2368 | 0.2093 |
| | 名詞と形容詞 | 0.1875 | 0.3000 | 0.2307 |
| | 名詞と前置詞 | 0.2500 | **0.3428** | 0.2891 |
| | 名詞と動詞 | **0.3125** | 0.3191 | **0.3157** |
| | 名詞と ing 形 | 0.1666 | 0.2500 | 0.2000 |

表 4　名詞と 2 種類以上の共起品詞のペアによる検出性能

| 手法 | 名詞と組み合わせる品詞 | 再現率 | 適合率 | F 値 |
|---|---|---|---|---|
| 検索エンジンを用いる方法 | 形容詞と前置詞 | 0.3750 | 0.5294 | 0.4390 |
| | 形容詞と ing 形 | **0.3958** | **0.6129** | **0.4810** |
| | 前置詞と ing 形 | **0.3958** | 0.5489 | 0.4578 |
| | 形容詞と前置詞と ing 形 | 0.3750 | 0.5455 | 0.4444 |
| 英語版 Wikipedia コーパスを用いる方法 | 代名詞と動詞 | 0.3125 | 0.3846 | 0.3448 |
| | 代名詞と前置詞 | **0.4166** | **0.4444** | **0.4301** |
| | 動詞と前置詞 | 0.3541 | 0.3777 | 0.3655 |
| | 代名詞と動詞と前置詞 | 0.3958 | 0.3725 | 0.3838 |
| 検索エンジンを優先する方法 | 形容詞と動詞 | 0.2500 | 0.3750 | 0.3000 |
| | 形容詞と ing 形 | 0.2500 | **0.5000** | **0.3333** |
| | 動詞と ing 形 | 0.2500 | 0.3636 | 0.2963 |
| | 形容詞と動詞と ing 形 | **0.2708** | 0.3823 | 0.3170 |
| コーパスを優先する方法 | 形容詞と前置詞 | 0.3125 | 0.3000 | 0.3061 |
| | 形容詞と動詞 | 0.3125 | 0.2678 | 0.2884 |
| | 動詞と前置詞 | 0.3958 | 0.3064 | 0.3454 |
| | 形容詞と動詞と前置詞 | **0.4791** | **0.3150** | **0.3801** |

表 5　Grammaly の検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 14 | 10 |
| | 正 | 34 | 103 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| 0.2917 | 0.5833 | 0.3889 |

表 6　宮城らの検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 14 | 11 |
| | 正 | 34 | 102 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| 0.2917 | 0.5600 | 0.3836 |

表 7　牧野らの検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 27 | 63 |
| | 正 | 21 | 50 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| **0.5625** | 0.3000 | 0.3913 |

表 8　検索エンジンを用いる方法の検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 19 | 12 |
| | 正 | 29 | 101 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| 0.3958 | **0.6129** | **0.4810** |

表 9　英語版 Wikipedia コーパスを用いる方法の検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 20 | 25 |
| | 正 | 28 | 88 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| 0.4166 | 0.4444 | 0.4301 |

表 10　検索エンジンを優先する方法の検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 13 | 16 |
| | 正 | 35 | 97 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| 0.2708 | 0.4482 | 0.3376 |

表 11　コーパスを優先する提案手法の検出結果

| | | 真の結果 | |
|---|---|---|---|
| | | 誤 | 正 |
| 推定結果 | 誤 | 23 | 50 |
| | 正 | 25 | 63 |

| 再現率 | 適合率 | F 値 |
|---|---|---|
| 0.4791 | 0.3150 | 0.3801 |

### 5.3 考　　察

#### 5.3.1 提案手法と他の手法の比較

検索エンジンを用いる方法で正しく誤りを判定できた文に “I am an English teacher in a cram school as a part time job.” という文がある．この文では job が誤りであり，正しくは worker である．宮城らの手法や提案手法のように，先頭から 3 語ずつの組み合わせで job を誤りかどうか判定すると，part と time と job が最後に選ばれ，この中で共起の強さを考えるペアは part と time，time と job，part と job の 3 ペアとなる．宮城らの手法ではこの 3 ペア全ての共起が弱いと判定されたため誤りを検出しなかった．一方，提案手法では part と time の共起は強く，time と job，part と job の共起はどちらも弱いと判定されたため，job だけが他の 2 語との共起が弱いことになり，誤りとして検出された．よって共起の強弱を決める閾値の選択が重要であると考えられる．

また，英語版 Wikipedia コーパスを用いる方法で正しく誤りを検出できた文に “He gave me his sign and a painting he painted in my cartoon book.” という文がある．この文では sign が誤りであり，正しくは autograph である．この文では sign と painting，cartoon，book の 4 語が名詞とタグ付けされる．宮城らの手法では，先頭から 3 語ずつの組み合わせとして sign と painting と cartoon が最初に選ばれ，この中で共起の強さを考えるペアは sign と painting，painting と cartoon，sign と cartoon となる．その結果，sign と painting の共起は強く，painting と cartoon，sign と cartoon の共起はどちらも弱いと判定されたため，cartoon だけが他の 2 語との共起が弱いことになり，cartoon が誤検出され，本来の誤りである sign は検出されなかった．また，牧野らの手法では名詞，代名詞，形容詞について，先頭から順に 3 語の共起の強さと全ての名詞のペアの共起の強さ考えるが，共起が弱いと判断されたものは sign と cartoon のペアのみであり，2 回以上共起が弱いと判定された名詞が無いために誤りは検出されなかった．一方，本稿で提案した英語版 Wikipedia コーパスを用いる方法では，名詞と代名詞の共起も考えるため，先頭から 3 語の組み合わせとして He と me と sign が選ばれる．このうち He と sign，me と sign のペアの共起の強さは他と同様に計算され判定されるが，He と me のペアは名詞を含まないため共起は強いと判定される．実験では，He と sign，me と sign の共起が弱いと判定されたため，sign だけが他の 2 語との共起が弱いことになり，誤りとして正しく検出された．名詞と代名詞の共起の強さを用い，ペアの品詞の組み合わせごとに共起の強さの判定をしたために検出が可能になった例と言える．

どの手法においても誤りを検出できなかった文に “My faculty is intelligence information.” などがあった．この文では faculty が誤りで正しくは major である．提案手法では名詞と共起品詞との共起の強さを調べるため，共起品詞が無い場合には効果が無い．また，1 文中の単語数が少ない場合には共起を考えるペアも少なくなる．このような文に対応するためには，1 文中だけでなく前後の文に含まれる単語との共起も利用することなどを検討すべきと考える．

#### 5.3.2 検索エンジンとコーパスを併用する方法の課題

表 8 から表 11 に示したように，検索エンジンを優先する方法やコーパスを優先する方法は，検索エンジンのみや英語版 Wikipedia コーパスのみを用いる方法に比べて性能が低かった．検索エンジンを用いる方法では誤りを正しく検出できたが，検索エンジンを優先する方法では検出できなかった文に “Because tomatoes in the shop have good taste and body.” がある．この文では body が誤りであり，正しくは appearance である．body を誤りかどうか判定する際に用いる 3 語の組み合わせは形容詞を共起品詞に含めると，検索エンジンのみを用いる方法と検索エンジンを優先する方法において共通で，good と taste と body である．共起の強さを考えるペアは good と taste，taste と body，good と body であり，この 3 語は全て，解析するスニペットに 50 回以上現れる単語であるため，共起の強さはどちらの方法でも検索エンジンを用いて算出した値となる．しかし，この二つの方法では定めた閾値が異なるため，検索エンジンを用いる方法では body のみが他の 2 語との共起が弱いと判定されたが，検索エンジンを優先する方法では good と body の共起は弱く，他のペアは共起が強いと判定された．そのため後者では誤りを検出しなかった．

このように提案手法では誤りの検出が閾値によって左右される．検索エンジンとコーパスを併用する方法では，検索エンジンの閾値と英語版 Wikipedia コーパスの閾値の両方を用いるため，さらにその影響が大きい．また，検索エンジンとコーパスの併用の仕方も，現在は単語の出現回数によって切り替えるという単純なものである．両者の併用方法をさらに検討することは今後の課題の一つである．

## 6. ま　と　め

本稿では検索エンジンと英語版 Wikipedia コーパスを用いて，名詞と名詞以外の品詞との共起の強さに基づいて名詞語彙誤りを検出する手法を提案した．宮城らは英文中の名詞 2 語の共

起に基づいて，牧野らは名詞，代名詞，形容詞のいずれかからなる 3 語の共起も利用して語彙誤りを検出したが，提案手法では牧野らの手法で扱った代名詞，形容詞に加え，動詞，前置詞，ing 形も考慮し，その 2 語の共起に基づいて誤りを検出した．また，共起の強さの算出に検索エンジンだけでなく英語版 Wikipedia コーパスも用いる．評価実験として，宮城らの手法と牧野らの手法，Grammarly，および提案手法の誤り検出性能を，KJ コーパスの英文 39 文を使って比較した．その結果，提案した検索エンジンを用いる方法や英語版 Wikipedia コーパスを用いる方法が，他の手法よりも検出性能が高いことを確認した．しかし，検索エンジンとコーパスを併用する方法は検出性能が低かった．

今後の課題としては，検索エンジンとコーパスを併用する方法の検討が挙げられる．また，提案手法は連続する 3 語の組み合わせで共起の強さを算出するため 2 語以上離れた語の共起は考慮されない．考慮する文脈をもう少し拡張することも今後の課題であるといえる．

## 文　献


[1] Konan-JIEM Learner Corpus Third Edition (KJ コーパス), http://www.gsk.or.jp/catalog/gsk2012-a/

[2] 水本智也, 林部祐太, 小町守, 永田昌明, 松本裕治, “大規模英語学習者コーパスを用いた英作文の文法誤り訂正の課題分析”, 情報処理学会研究報告第 208 回自然言語処理研究会, Vol.2012-NL-208, No. 5, pp. 1-8, 2012.

[3] 宮城雄太, 新妻弘崇, 太田学, “検索エンジンを用いた英文名詞句誤りの修正支援” , DEIM2015, D7-2, 2015.

[4] 牧野剛典, 新妻弘崇, 太田学, “検索エンジンによる英文の名詞語彙選択誤り検出の一手法” , DEIM2016, C1-5, 2016.

[5] 尾崎弘明, 太田学, “Web 資源を用いた冠詞の用法に基づく冠詞誤り自動修正”, 第 4 回データ工学と情報マネジメントに関するフォーラム (DEIM2012), E9-2, 2012.

[6] 尾崎弘明, 谷本太郁由, 太田学, “冠詞の用法規則と検索エンジンを利用した英文冠詞誤りの自動修正”, 第 5 回 Web とデータベースに関するフォーラム (WebDB Forum2012), B4, 2012.

[7] 尾崎弘明, 新妻弘崇, 太田学, “機械学習による冠詞の用法と検索結果数に基づく英文冠詞誤りの自動修正”, 情報処理学会研究報告, Vol. 2013-DBS-158, No. 14, pp. 1-7, 2013.

[8] 有冨隼, 太田学, “検索エンジンによる英文前置詞誤り修正支援”, 日本データベース学会論文誌, Vol. 9, No. 1, pp. 70-75, 2010.

[9] 久保田朗, 太田学, “検索エンジンを用いた英文前置詞誤りの自動検出と修正”, 情報処理学会研究報告, データベース・システム研究会報告, Vol. 2011-DBS-153, No. 2, pp. 1-8, 2011.

[10] 谷本太郁由, 太田学, “検索エンジンを用いた動詞名詞コロケーションに基づく英文動詞誤りの検出と修正”, 情報処理学会研究報告, データベース・システム研究会報告, Vol. 2010-DBS-151, No.36, pp. 1-7, 2010.

[11] 谷本太郁由, 太田学, “検索エンジンを用いた英文動詞誤り検出システム”, 情報処理学会研究報告, データベース・システム研究会報告, Vol. 2012-DBS-156, No. 2, pp. 1-8, 2012.

[12] Gabor Berend, Veronika Vincze, Sina Zarriess, Richard Farkas“LFG-based Features for Noun Number and Article Grammatical Errors”, Proceedings of the Seventeenth Conference on Computational Natural Language Learning：Shared Task, pp. 62-67, 2013.

[13] CoNLL-2013, http://www.clips.uantwerpen.be/conll2013/

[14] Andrew Kachites, McCallum, “MALLET：A Machine Learning for Language Toolkit”, http://mallet.cs.umass.edu.

[15] 大鹿広憲, 佐藤学, 安藤進, 山名早人, “Google を活用した英作文支援システムの構築”, DEWS2005, 4B-i8, 2005.

[16] 網嶋祐一, 川崎優太, 安藤一秋, “検索エンジンを用いた英作文支援ツール” , 電子情報通信学会技術研究報告, 教育工学, Vol. 106, No. 583, pp. 87-92, 2007.

[17] 平野孝佳, 平手勇宇, 山名早人, “検索エンジンを用いた英文冠詞誤りの検出”, 日本データベース学会 Letters, Vol. 6, No. 3, pp. 1-4, 2007.

[18] Bing Search API ― Windows Azure Marketplace, http://datamarket.azure.com/dataset/bing/search

[19] Wikipedia, https://en.wikipedia.org/wiki/Main_Page

[20] MontyTagger, http://web.media.mit.edu/~hugo/montytagger/

[21] Grammarly, https://app.grammarly.com