Vol.2011-DBS-153 No.2
2011/11/3

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

久保田 朗[†1] 太田 学[†1]

英語を母語としない日本人が英作文を行うと，適切な前置詞の選択に迷うことがある．適切な前置詞であるかどうか判断するには，その前置詞を含む英文を検索フレーズとして検索エンジンで検索し，その検索結果数を調べる方法がある．そこで我々は，有冨らの実装したプロトタイプに基づいて，検索フレーズの生成，検索結果のサマリの収集，サマリにおける前置詞の出現率の自動計算，その結果のユーザへの提示を行うシステムを提案した．このシステムでは生成する検索フレーズの良し悪しが性能の決め手となるが，提案した検索フレーズ生成法では，前置詞誤りの検出，修正ができない例が多く存在した．そこで本稿では，この検索フレーズ生成法の改良法を提案し，有冨らの手法や我々の提案した従来手法と比較するために行った実験について報告する．実験結果からは，本稿の提案により前置詞誤りの自動検出および自動修正の性能を大幅に改善できることが分かった．

Detection and Correction of English Preposition Errors Using a Search Engine

Akira Kubota[†1] and Manabu Ohta[†1]

Japanese people are sometimes at a loss what preposition to use in English composition. There is a mode of confirming how appropriate used prepositions are by using a search engine. That is, we can search for a query phrase including the preposition to be checked and judge its appropriateness based on the number of returned search results. Therefore, we proposed a system that automatically i)generates query phrases including a preposition, ii)collects summaries of the search result, iii)calculates occurrence probabilities of prepositions in the collected summaries, and iv)presents the prepositions and their probabilities to a user. It was based on the prototype system Aritomi et al. implemented. There were, however, several problems with its query phrase generation. Therefore, we propose a method for improving query phrase generation and give some experimental results for evaluating the method performance.

1. はじめに

普段英語を使用していない日本人が英語で文書を作成する際，表現方法に迷うことがある．そこで有冨ら[1)]は，検索エンジンを利用して，Web 上の文書コーパスにおける前置詞の出現頻度を調べ，それによって妥当性を判断し，結果をユーザに提示するシステムを提案した．検索エンジンを利用するためには，適当な検索フレーズを生成する必要があるが，有冨らは前置詞に注目し，検討したい英文を入力すると検索フレーズを自動生成して，検索する前置詞誤り修正支援システムを実装した．さらに我々は，複合名詞や日付の表現ををうまく扱えないといった問題を解決する検索フレーズ生成法を提案し，検出性能，精度の向上を確認した．しかし，検索フレーズ生成において複数の生成規則を適用する場合，順序によって結果が異なる等，複雑な検索フレーズ生成規則の妥当性が証明できなかった[2)]．そこで本稿では，この問題を解決するべく前置詞誤り検出及び修正の改良法を提案し，評価実験により，[2)]で提案した手法及び NativeChecker[9)]と比較する．

2. 関連研究

2.1 検索エンジンを用いた前置詞誤り修正支援

検索エンジンによる英文前置詞誤りの自動検出，修正を行うシステムを，岡山大学の有冨ら[1)]が提案している．検討したい前置詞を含む英文を入力として与えると，システムが検索フレーズを自動生成して検索を行う．次に，返ってきた検索結果から前置詞の出現率を自動計算し，結果をユーザに提示する．しかし彼らの検索フレーズ生成法では，複合名詞等の扱いが考慮されておらず，名詞が複数並ぶ日付の表現や複合名詞等を含む英文に対して，適切な検出，修正ができなかった．我々は，有冨らが課題として挙げていた，複数の名詞に対応した検索フレーズ生成法を提案し．検出性能，修正性能の向上を確認したが，検索結果数が少ない場合は誤り検出，及び適切な修正ができなかった．

また Gamon ら[6)]は，英語学習者の誤りを含む英文とそれを正しく修正した英文から検索フレーズを生成し，検索エンジンに入力して検索結果数を比較した．その結果，正しい前置詞を含む検索フレーズの検索結果数が，誤った前置詞を含む検索フレーズの検索結果数よ

†1 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

りも，概ね多くなることを確認した．彼らはまた前置詞の欠落や挿入，冠詞の誤用についても，誤った英文とそれを修正した英文を利用して，検索エンジンの返す検索結果数が正しい用法を支持することを実験により示している．彼らの実験は，検索エンジンが英語の誤用の発見に有効であることを示したが，検索方法そのものは修正した英文と比較しているので，実用的な誤用検出や修正はできない．それに対して本稿の提案は，修正した前置詞を用いることなく，誤りを含む可能性のある英文のみを入力として，前置詞誤りを検出し，さらにその修正を試みるものである．

2.2 検索エンジンを用いた英文冠詞誤りの検出

検索エンジンを使った英文誤り検出の研究は前置詞以外の品詞についても行われており，Lapata ら[3] は，冠詞の誤り検出の研究を行っている．Lapata らは，まず構文解析を用いて名詞句を抽出する．続いて抽出した名詞の冠詞を { a/an, the, φ } に変化させ，3 パターンの検索フレーズを生成し，それぞれ検索を行う．ここで，φは無冠詞を表す．最後に，返ってきたヒット数を比較することで，冠詞誤りを検出する．また，平野ら[4] は，Lapata らの手法を改良し，名詞を単数形と複数形に変化させた検索フレーズを加えることで，検出率，検出精度共に向上させている．

2.3 検索エンジン用いた英作文支援

英作文した文章の前置詞，冠詞，多義語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を，大鹿ら[5] が作成している．その機能の一部として前置詞の誤りの検出が実装されている．検討したいフレーズを入力すると，システムがフレーズ内の前置詞部分をワイルドカードに置き換えて検索し，入力されたものとは異なる前置詞を取得する．次に，入力した前置詞と取得した前置詞それぞれを含む検索フレーズで検索し，検索結果数を表示する．この結果数から，どの前置詞が適切かユーザに判断してもらう．欠点として，何度も検索を行うため応答時間が長い点，英文全体でなく，ユーザ自身が適当なフレーズを考えて入力する必要がある点が挙げられる．

2.4 NativeChecker

NativeChecker[9] は，英語のフレーズを入力すると，そのフレーズの様々な項目を修正することができる Web サービスである．検討したい英語のフレーズを入力すると，入力がそのまま検索フレーズとなり，そのヒット件数が表示される．修正を行う場合，修正したい単語の上にマウスのカーソルをのせると修正方法が提示され，ユーザが適当な修正方法を選ぶと，修正後のフレーズとそのヒット数を表示する．修正できる内容として，スペルミス，類義語，単複数形，to 不定詞，その他の表現，単語の削除，単語の順序の変更がある．

3. 英文前置詞誤り修正支援システム

3.1 概　　要

検索エンジンを使った英文前置詞誤りの検出及び修正の概要を図 1 に示す[1]．まず検討したい英文を入力すると，システムは前置詞を 1 つだけ含むように英文を分割し，その単位で扱う．次に分割された英文を処理して，検索フレーズを生成し，検索を行い，1 回の検索結果から最大 100 件のサマリを取得する．得られたサマリにおける前置詞の出現回数から出現率を計算し，出現率の高いものを正解とする．本稿で提案する手法では，最大 4 回の検索により 400 件までのサマリを取得し，正解判定を行う．英文の分割方法，検索フレーズの生成法ならびにサマリ取得については，4 節で詳しく述べる．

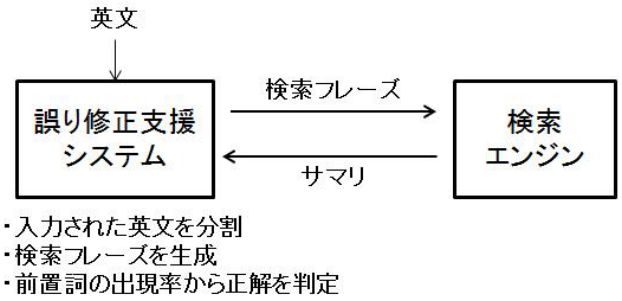

図 1　検索エンジンによる英文前置詞誤り検出の概要

3.2 検　　索

本システムで検索を行う際は，フレーズ検索を用いる．これにより語順を保ったままの検索を行うことができる．検索エンジンには Yahoo!デベロッパーネットワーク[8] で提供されている Yahoo!検索 API を使用する．

3.3 実装システム

図 2 は英文前置詞誤り修正支援システムの実行画面例である．最上部のボックスが英文入力部で，その下のボタン 2 つは，左が前置詞誤り検出を開始するボタン，右側がシステムを初期画面に戻すクリアボタンである．下部の結果表示部分には，前置詞を 1 つだけ含むように入力文を分割処理したものを表示し，その前置詞の下に，出現率の高い前置詞を順番に表示する．その下に最初に生成される検索フレーズと，最大 4 回の検索結果数の合計

を表示する．この時表示される表 1 に示す 4 段階評価は，出現率と出現回数に基づいて決定している．最も出現率の高い前置詞を赤く表示し，逆にサマリ中に 1 回しか出現しないものは薄く表示する．出現率が最も高い前置詞と入力文中の前置詞が異なる場合，“[!]”をつけて注意を喚起する．図 2 で，“She is busy for her home work.” という前置詞誤りを含む英文が入力されている．この文では “for”の使用が不適切であり，正解は “with”である．この実行画面でも，“with”の出現率が最も高いことを示している．

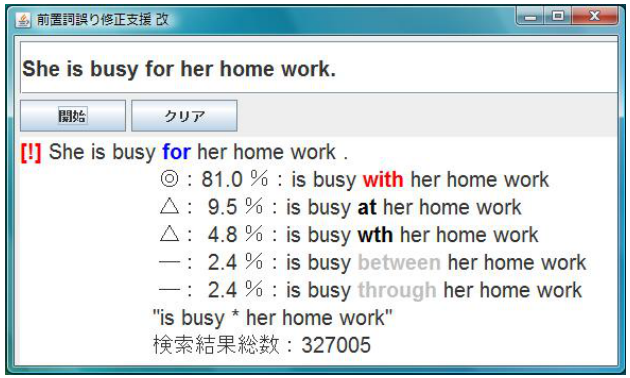

図 2 実行画面の例

表 1 表示する 4 段階評価

	基準
◎	出現率が 50 %以上
○	出現率が 10 %以上 50 %未満
△	出現率が 10 %未満かつ出現回数が 2 回以上
—	出現回数が 1 回

4. 提案手法

本節では，本稿で提案する検索フレーズ生成法，サマリ収集，前置詞の出現率の計算方法について述べる．

4.1 検索フレーズの生成

入力された英文から検索フレーズを生成する処理について説明する．

(1) 複数の英文が入力された場合，文末のピリオドの前後で文を分割する．
(2) 入力された英文がコンマを含む場合，もしくは複数の前置詞を含む場合は英文を分割する．分割処理の方法については 4.1.1 項で述べる．
(3) 入力された英文もしくは分割された英文に対して，品詞のタグ付けを行う．タグ付けには Eric Brill の MontyTagger[7] を用いる．
(4) 特定の品詞だけが残るように，不要な単語を削除する．この単語の削除については 4.1.2 項で詳しく述べる．
(5) 前置詞部分をワイルドカードに置き換え，初期検索フレーズとする．この方法については，例を交えて 4.1.3 項で説明する．

4.1.1 英文の分割処理

入力された英文がコンマを含む場合は，その前後で分割する．入力された英文に前置詞が複数含まれる場合は，その前置詞の数だけ英文を分割する．前置詞の前後の単語列が検索フレーズを構成するため，分割後の英文には重複する部分が存在する．例えば，“I waited for his call until midnight.”という文からは，“I waited for his call”，“his call until midnight”の 2 つの単語列が抽出される．また，前置詞が連続して出現する場合は，そこでは分割しない．

4.1.2 単語の削除

分割処理後の単語列を，前置詞より前の単語列と前置詞より後の単語列に分ける．ただし，連続する 2 つの前置詞は 1 つとみなし，それらの前後に分ける．

前置詞の前の単語列には，最後に現れる動詞を 1 語，最後に現れる冠詞を含む名詞群を 1 つ，および “me”，“yours”といった人称代名詞の目的格及び所有代名詞のうち，最後に現れるもの 1 語を残し，他の単語は全て削除する．名詞群とは，連続して現れる名詞を指す．名詞が存在しない場合は，名詞の代わりに最後に現れる形容詞または副詞 1 語を残す．さらにこのような形容詞および副詞が単語列に存在せず，その他の単語の削除を行うと単語列に何も残らない場合は，その単語を削除しない．これは，単語列が空になると検索結果が膨大になることが多く，前置詞誤りをうまく検出，修正できないからである．

前置詞の後の単語列では，動詞および動名詞が前置詞の直後に存在する場合は，それら以外の単語を全て削除する．動詞，動名詞が存在せず，名詞が存在する場合は最初に現れる名詞群 1 つと，最後に現れる人称代名詞の所有格及び所有代名詞 1 語だけを残し，他の単語

をすべて削除する．名詞が存在しない場合は，人称代名詞の所有格及び所有代名詞１語と，最後に現れる形容詞または副詞だけを残す．人称代名詞は存在する場合のみ残す．

4.1.3 初期検索フレーズ

4.1.2 項で述べた単語の削除を行った後，前置詞の前後の単語列の間にワイルドカードを挿入して，初期検索フレーズとする．例えば，“This country is very excellent in the information technology.”という英文から初期検索フレーズを生成してみる．まず，前置詞“in”の前後で分割し，“This country is very excellent”と “the information technology”という２つの単語列を得る．次に，前置詞の前の単語列から動詞の前に存在する単語を削除し，“is very excellent”とする．この単語列には名詞が存在せず，副詞 “very”と形容詞“excellent”が含まれるので，最後に現れる形容詞の “excellent”だけを残す．よって，前置詞の前の単語列からは “is excellent”という単語列が生成される．前置詞の後の単語列には名詞が存在するので，最初に登場する名詞群を残し，“the information technology”とする．こうして得られた２つの単語列の間にワイルドカードが挿入され，“is excellent * the information technology” という初期検索フレーズが生成される．

ここで，連続する２つの前置詞を含む場合は，片方の前置詞だけをワイルドカードに置き換えた 2 パターンの初期検索フレーズを生成する．例えば，“My hamster appeared from under the desk.”という英文からは，“appeared * under the desk”と “appeared from * the desk”の２つの初期検索フレーズを生成し，それぞれで検索を行う．

4.2 再検索フレーズ

初期検索フレーズの検索結果だけでは，十分な数のサマリが得られないことがしばしばある．そこで，サマリ不足を補うため，再検索フレーズ生成して再検索を行い，新たにサマリを取得する．具体的には以下 3 つの処理により再検索フレーズを生成し，最大 3 パターンの再検索フレーズで検索してサマリを取得する．

(1) 前置詞の前の名詞群が 2 つ以上の名詞から構成されるならば，最初に現れる名詞１語を削除する．
(2) 前置詞の後の名詞群が 2 つ以上の名詞から構成されるならば，最初に現れる名詞１語を削除する．
(3) 前置詞の前の単語列の動詞を削除する．

各再検索フレーズ毎に最大 100 件のサマリを取得後，サマリに現れる各前置詞の出現回数，及び全ての前置詞の出現回数をカウントし，初期検索フレーズのそれらと合計する．合計した値から，各前置詞の出現率を計算し，正解判定を行う．前置詞出現率の計算方法は 4.4 節で詳しく述べる．

先の例のフレーズ “is excellent * the information technology”であれば，(2) の処理より名詞 “information”が削除され，“excellent * the technology”という再検索フレーズが生成される．また，(3) の処理により，“is”が削除され，“excellent * the information technology”という再検索フレーズが生成される．

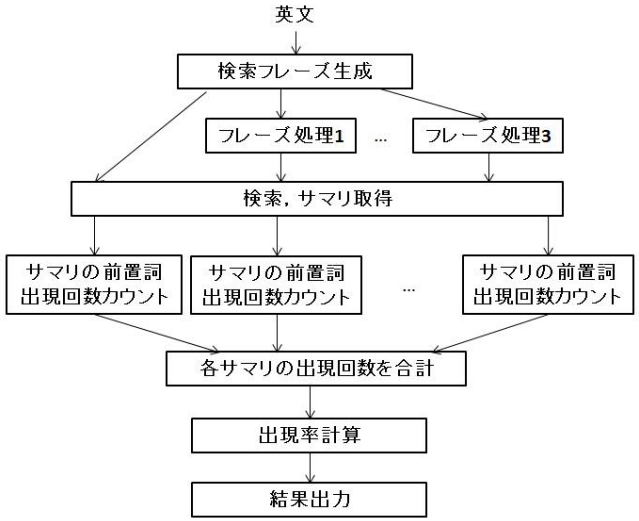

図 3 提案手法

4.3 前置詞が文頭または文末に存在する場合の検索フレーズ

前置詞が文頭及び文末にある場合は，前置詞より前，または後の単語列が必ず空になるので，検索の手がかりとなる単語が少ない．このような場合には，適切な誤り検出，修正ができないことが多い．そこで，前置詞が文末または文頭にある場合は，4.1.3 項及び 4.2 節とは異なる方法で検索フレーズを生成し，それによって得られた最大 100 件の検索結果のみを用いて前置詞の正解判定を行う．

4.3.1 前置詞が文頭にある場合

入力された英文の文頭に前置詞がある場合は，文頭から前置詞の目的語，後に続く文の主語までを検索フレーズとする．途中の単語を削除せず文章をそのまま抜き出すことで，よ

り検索結果を限定できる．具体的には，文頭から，初めて名詞が現れて以降，形容詞，冠詞および名詞以外の品詞が現れるまでを検索フレーズとしてそのまま利用する．例えば，“ In some countries many people die from poverty. ”という英文の前置詞 In に関する部分，“In some countries many people die”からは，“In some countries many people”を抜き出し，前置詞部分をワイルドカードで置き換え，“ * some countries many people ”という検索フレーズが生成される．

4.3.2 前置詞が文末にある場合

文末の前置詞の場合は，入力された英文の主語を新たな手がかりとして検索フレーズに加える．まず 4.1.3 項の初期検索フレーズ生成法に従って検索フレーズを生成する．その後に，前置詞より前の単語列において動詞より前に現れる単語のうち，最後に現れる名詞 1 語をそのフレーズの先頭に加える．例を示すと．“ Long skirts have come in. ”という英文からは，まず “come * ”というフレーズが生成される．その後，削除した動詞より前の単語のうち，最後に現れる名詞 “skirts”を先頭に加え，“skirts come *”を検索フレーズとする．

4.4 前置詞の出現率の計算

4.1.3 項の方法で生成した初期検索フレーズから得た検索結果，および 4.2 節の再検索フレーズによる検索から得た最大 3 つの検索結果から，最大 100 件ずつのサマリを取得し，合計最大 400 件のサマリを用いて前置詞の出現率を計算する．ただし，前置詞が文頭または文末にある場合は，4.3 節で説明した検索フレーズで検索した最大 100 件のサマリのみを用いる．

4.4.1 前置詞の出現率計算

それぞれ取得したサマリから，検索フレーズのワイルドカード部分に相当する前置詞を抽出し，各前置詞の出現回数及び全ての前置詞の出現回数をカウントする．ただし，カウントの際には，各サマリの重みを考慮する．重みの設定方法については，次項で詳しく説明する．その後，抽出した前置詞それぞれに対して，式 (1) で前置詞の出現率 R_i を求める．ここで，4.1.3 項の初期検索フレーズの検索結果のサマリにおける前置詞 i の出現回数を $Prep_i$，全ての前置詞の出現回数を N とする．同様に，4.2 節の再検索のパターン (j) の検索結果のサマリにおける前置詞 i の出現回数を $Prep_i^{(j)}$，全前置詞の出現回数を $N^{(j)}$，重みを $w^{(j)}$ とする．また，初期検索フレーズの重みは，重みの合計が 1 になるように設定する．

$$R_i = \frac{wPrep_i + \sum_{j=1}^{3} w^{(j)} Prep_i^{(j)}}{wN + \sum_{j=1}^{3} w^{(j)} N^{(j)}} \times 100(\%) \quad (1)$$

$$w + \sum_{j=1}^{3} w^{(j)} = 1 \quad (2)$$

4.4.2 サマリの重みの設定

初期検索フレーズと再検索フレーズの検索結果のサマリに重みを設定する．4.1.3 項の初期検索フレーズのみの場合と，4.2 節の再検索フレーズのパターン (1)〜(3) を個別に加えた場合で，システムが正しい正誤判定ができるか実験を行い，再検索フレーズパターン (1)〜(3) を加えた場合のそれぞれの正解数の比を基に，(1)〜(3) のサマリの重みを決定する．実験ではシステムに誤りのない英文を入力し，システムがサマリ中で最も出現率が高いと示した前置詞と，入力した英文中の前置詞が一致すれば，正しい正誤判定ができたとみなす．テストデータとして，NewYorkTimes[11] の記事から，(1)〜(3) それぞれのパターンに対して英文を 30 ずつ選び使用した．また本実験では検索するドメインを.com に制限している．

結果を表 2 に示す．再検索フレーズの結果を加えると，(1) は 1 件，(2) および (3) では 3 件，正解数が増えた．予備実験として，再検索パターン (1)〜(3) の重みを表 2 のそれぞれの正解数の比の値とし，初期検索フレーズの重みを 0.1，0.5，1.0，2.0，4.0，8.0，16.0，100.0 と変化させ，正誤判定の正解数の変化を調べた結果，初期検索フレーズの重みを大きくすると正解数が増え，4.0 以上の値では大きな変化がなかった．そこで，正解数の上昇率を 10 で割った値を再検索パターン (1)〜(3) の結果の重み，すなわち $w^{(1)} = 0.104$, $w^{(2)} = 0.113$, $w^{(3)} = 0.116$ として設定した．また，式 (2) から，初期検索フレーズの重み w を 0.667 に設定した．

表 2 再検索による正誤判定の上昇率

	初期検索による正解数	初期＋再検索による正解数	正解数の比
再検索パターン (1)	23	24	1.04
再検索パターン (2)	24	27	1.13
再検索パターン (3)	19	22	1.16

5. 評 価 実 験

本稿の提案する手法を用いた英文前置詞誤り修正支援システムの性能を示すため，以下の 2 つの評価項目について実験を行い，我々が DBS 研究会報告で提案した手法や NativeChecker と結果を比較した．

(1) 提示する修正候補の適切性の評価
(2) 前置詞誤りの自動検出性能と自動修正精度

5.1 DBS 手法

提案手法と，DBS-151 で提案した手法[2] との主な違いを説明する．本稿の提案手法では，初期検索フレーズの検索結果数に関係なく 4.2 節の再検索を行い，全ての検索結果のサマリを取得するのに対し，DBS-151 の手法では，検索結果が一定数に満たない場合のみ検索フレーズを修正し，再検索を行う．また，4.2 節の再検索フレーズパターンに優先順位をつけて順次検索を行い，5 件以上の検索結果が得られた時点で検索を打ち切りサマリを取得する．

5.2 前置詞誤り修正候補の適切性の評価

英文修正システムに前置詞を含む誤りのない英文を入力として与え，そこに現れる前置詞を修正候補として提示できるかを実験により評価する．テストデータには英文法書の総合英語 Forest[10] の例文 100 文を用い，この中に含まれる前置詞 126 個を評価の対象とする．評価基準は，“システムが最も出現率が高いと提示した前置詞と入力した英文中の前置詞が一致するか”と，“システムが 4 段階評価の ‘○’ 以上として提示した前置詞のいずれかと入力した英文中の前置詞が一致するか”の 2 つとする．DBS 手法については，前節で述べた最大 100 件のサマリを取得する手法に加え，提案手法と同じくサマリを最大 400 件取得する手法でも実験を行って比較した．

結果を表 3 に示す．この表から，‘○’ 以上の前置詞の中に正解が含まれれば良いという基準では，DBS 手法の正解率は，サマリを 100 件取得した場合は 0.770，400 件取得した場合は 0.810，提案手法は 0.849 だった．

表 3 提示する修正候補の適切性

手法	条件	正解数	誤検出	正解率
DBS 手法 (100)	出現率が最も高いものと一致	80	46	0.635
	○以上の前置詞と一致	97	29	0.770
DBS 手法 (400)	出現率が最も高いものと一致	86	40	0.683
	○以上の前置詞と一致	102	24	0.810
提案手法	出現率が最も高いものと一致	94	32	0.746
	○以上の前置詞と一致	107	19	0.849

まず，提案手法が DBS 手法 (100) を上回ったのは，再検索方法の改良により利用するサマリの件数を増やしたことの寄与が大きいと考えられる．DBS 手法 (100) では最大 100 件のサマリを用いて正解判定を行うのに対し，提案手法では，初期検索 1 回と再検索最大 3 回を合わせ，400 件までのサマリを正解判定に用いる．また，DBS 手法 (100)，及び DBS 手法 (400) では，初めて 5 件以上の検索結果が得られたときにサマリを取得し，そこで検索を打ち切り正解判定を行う．特に，再検索を行う場合は，初期検索結果を破棄して，採用したパターンの再検索結果のみを利用しているため，サマリを取得する検索フレーズに結果が大きく依存する．一方，提案手法では初期検索フレーズから得た検索結果数とは関わりなく再検索を行い，その結果も正解判定に用いる．さらに，入力された英文に近い初期検索の結果を最も重視して正解判定に利用できる．これらの違いにより，最大 400 件取得する条件を揃えた場合でも提案手法が DBS 手法 (400) を上回った．

提案手法を用いても正誤判定が困難な事例には，システムが示した前置詞は例文とは異なるが不自然な英文でない，等が挙げられる．これは有冨らの手法と共通する問題である[1]．不自然な英文でないとは，例えば，“The book under the desk is not mine.” という文の “under” の部分は “on” でも正しいといった場合である．これは，英文の日本語訳など意味を与えなければ修正は難しい．

また，名詞，形容詞，副詞，動詞以外にタグ付けされる単語を含むイディオムが文に現れる場合，適切な正誤判定ができない例が存在した．例として，“a service that allows users to chat with one another” の with に関する部分では，4.1.2 項の単語削除を行うと，“another” が削除されるので，検索フレーズにイディオム “one another” を残すことができない．

その他にも，前置詞+1 単語のように，4.1.1 項の分割後の文が短すぎる場合や，“take” のように非常に多くの単語と結びつく語を含む場合は，検索ノイズが大きく，適切な正誤判定が難しかった．

5.3 前置詞誤りの自動検出と自動修正の精度

一定割合の前置詞誤りを含む英文を作成し，システムに入力として与え，それぞれの手法を用いてどの程度誤りを自動検出，自動修正できるかを実験により評価する．テストデータには，New York Times[11] の記事中の 50 文を用いる．この中に含まれる 200 個の前置詞のうち，100 個の前置詞を無作為に選んだ他の前置詞に置換することで誤りとし，この英文をシステムへの入力とする．

5.3.1 前置詞誤りの自動検出および自動修正の正誤判定

有冨らの手法，DBS 手法 (100/400)，提案手法を用いた英文前置詞誤り修正支援システムでは，英文前置詞誤りの自動検出と自動修正の正誤判定を以下のように行った．

- 自動検出の正誤判定
  入力した英文中の前置詞と，システムが提示する出現率の最も高い前置詞が一致しない

とき，誤りとして検出する．誤りでない前置詞を検出した場合は，誤検出となる．

- 自動修正の正誤判定
  システムが提示した前置詞の出現率上位 3 件以内，かつ 4 段階評価が ‘○’ 以上のものの中に正解の前置詞が含まれていれば，修正できたとみなす．後述する NativeChecker[9] と公平に比較を行うため，出現率上位 3 件以内という条件を加えた．

5.3.2 NativeChecker

NativeChecker による前置詞誤りの自動検出，修正を行い，提案手法を用いたシステムと結果を比較する．NativeChecker は，英文をそのまま入力とすると適切に前置詞の修正ができないので，提案手法により生成された検索フレーズのワイルドカードを，元の前置詞に置き換えたフレーズを入力とする．前置詞誤りの修正，検出を行うには，NativeChecker の修正方法のうち，“その他の表現”を利用し，入力されたフレーズの前置詞を，他の前置詞に置き換えた結果を表示する．ただし，NativeChecker は，前置詞の “その他の表現”を選ぶと，予め用意された前置詞セットに含まれる前置詞に置き換え，そのヒット数と共に表示する．そのため，“up”や “out”等，NativeChecker の前置詞セットに含まれない前置詞が誤っている場合は “その他の表現”で修正できず，また, これらの前置詞が正解である場合でも，修正候補として現れない．そこで，前置詞セットに含まれない前置詞を含み，かつ誤っている文を扱う場合は，前置詞部分を in,at 等の前置詞セットに含まれる単語に置き換え，“その他の表現”により修正候補を提示させる．また，前置詞セット外の前置詞が正解前置詞である場合は，その正解である単語に置き換え，そのヒット件数を調べた．NativeChecker による自動検出の方法と，自動修正の正誤判定は以下のように行う．

- 自動検出の方法
  入力フレーズ中の前置詞が，修正候補の最上位の前置詞と一致しない場合，誤りとして検出する．誤りでない前置詞を検出した場合は誤検出となる．入力された前置詞が前置詞セットに含まれない場合は，前置詞部分を前置詞セットに含まれる単語に置き換え，“その他の表現”で最上位に現れる修正候補のヒット件数が入力文のヒット件数を上回る場合，誤りとして検出したとみなす．
- 自動修正の正誤判定
  NativeChecker が提示した修正候補の上位 3 件以内に，正解前置詞が含まれる場合，修正できたとみなす．正解前置詞が前置詞セットに含まれない場合は，正解の前置詞に置き換えた場合のヒット件数を，“その他の表現”で上位に現れる修正候補と比較し，上位 3 件に入るならば，修正できたとみなす．

5.3.3 前置詞誤りの検出精度

前置詞誤りの自動検出の正誤判定の結果および自動検出性能を表 4，表 5 にまとめる．提案手法を用いた場合は，自動検出の検出率 0.990，検出精度は 0.818，F 値は 0.896 であり，全てにおいて提案手法が他の全ての手法を上回った．提案手法が上回った理由は，誤り検出の場合と同じ様に，検索結果数とは無関係に再検索を行っていること，及び初期検索フレーズを重視した正解判定を行っていることが考えられる．NativeChecker は提案手法に近い性能を示しているが，NativeChecker を使う際の様々な制約を考慮すると，実質的には提案手法が優れていると考えられる．

表 4 前置詞誤りの自動検出結果

	誤っている前置詞		誤りのない前置詞	
	検出	非検出	非検出	誤検出
有冨らの手法	97	3	54	46
DBS 手法 (100)	98	2	64	36
DBS 手法 (400)	99	1	67	33
提案手法	99	1	78	22
NativeChecker	97	3	77	23

表 5 前置詞誤りの自動検出性能

	検出率	検出精度	F 値
有冨らの手法	0.970	0.678	0.804
DBS 手法 (100)	0.980	0.731	0.837
DBS 手法 (400)	0.990	0.750	0.853
提案手法	0.990	0.818	0.896
NativeChecker	0.970	0.808	0.882

5.3.4 前置詞誤り修正精度

5.3.3 項の実験で，前置詞誤りとして検出できた前置詞を対象に，自動修正を行った．有冨らの手法と NativeChecker は 97 個，DBS 手法 (100) は 98 個，DBS 手法 (400) および提案手法では 99 個の前置詞がそれぞれ対象となる．

結果を表 6 にまとめる．提案手法は，他の手法と比較して，1 割以上の精度向上を示した．ただし修正においても，システムが提示した前置詞が正解とは異なるが，不自然な英文ではない場合があった．この場合修正はできていない．5.2 節の修正候補の適切性評価の実験

と同じ理由で，英文の和訳を与えるなどしない限り，このような英文の自動修正は難しい．また，入力された英文が短い場合も，適切な修正が困難な場合があった．

表 6 前置詞誤りの自動修正精度

	修正できた	修正できない	修正精度
有冨らの手法	68	29	0.701
DBS 手法 (100)	72	26	0.735
DBS 手法 (400)	73	26	0.737
提案手法	83	16	0.838
NativeChecker	70	27	0.722

5.4 実行時間

提案手法は必ず再検索を行うため，DBS 手法 (100) より実行時間がかかる．表 7 は，5.2 節の実験における，DBS 手法 (100) と提案手法を用いた場合の実行時間とアクセス回数を比較したものである．なお実行時間の測定は以下の計算機で行った．

- ＣＰＵ： Intel Core(TM)2 2.40GHz
- メモリ： 2.00GB
- ＯＳ： Windows Vista

提案手法のほうが実行時間がかかったのは，サマリ取得回数を増やしたため，Yahoo!検索 API[8] へのアクセス回数が増えたことが原因である．Yahoo!検索 API は，一度のリクエストに対し，結果取得は 20 件が上限であり，検索結果数は一度のアクセスで取得できるが，サマリを 100 件取得するには 5 回のアクセスが必要となる．DBS 手法 (100) は最大 100 件のサマリを取得するので，検索結果数取得とサマリ取得を合わせると，少なくとも 6 回のアクセス，また多くとも 10 回程度のアクセスで実行できる[2]．一方，提案手法では 4.3 節の再検索フレーズパターン (1)〜(3) それぞれについて検索するので，1 パターンの検索につき，検索結果数とサマリ取得で計 6 回のアクセスを必要し．初期検索と合わせると最大 24 回のアクセスを要する．

6. まとめ

本稿では，検索エンジンを用いた前置詞誤りの自動検出，修正のための検索フレーズ生成法を提案した．本研究は，有冨らの英文前置詞誤り修正支援システムの要である検索フレーズ生成法を改良し，実験により検出精度，修正精度共に，昨年著者らが本研究会で提案した検索フレーズ生成法に比べて改善することを確認した．特に修正精度は 1 割以上向上させることができた．実験結果から，検索フレーズを工夫して網羅的に再検索を行い，なるべく多くのサマリを利用する方が英文前置詞修正支援システムでは有効であることが分かった．今後の課題として，実行時間の短縮や，前置詞の欠落や挿入誤りへの対処，また，前置詞以外の品詞の誤りの修正が挙げられる．

表 7 前置詞誤り修正候補の適切性評価の実験の実行時間（100 文，126 前置詞）

	実行時間合計 (s)	1 前置詞当たりの実行時間 (s)	検索エンジンアクセス回数
DBS 手法 (100)	326	2.59	775
提案手法	538	4.27	1230

参考文献

1) 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”，DBSJ Journal vol.9 No.1，pp. 70-75, 2010.
2) 久保田朗，太田学: “検索エンジンによる英文前置詞誤り修正のための検索フレーズ生成法”，情報研報 (DBS), Vol.2010-DBS-151 No.37 , 2010.
3) M.Lapata, and F.Keller, “Web-based models for natural language processing”, ACM Trans. Speech and Language Processing, Vol.2, No.1, pp.1-31, Feb. 2005.
4) 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”, 情報研報 (DBS), Vol.2007, No.65, pp.139-144, 2007.
5) 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.
6) Michael Gamon and Claudia Leacock: “Search right ant thou shalt find ... Using Web Queries for Learner Error Detection”，Proseedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Application, pp.37-44，2010.
7) Monty Tagger
http://web.media.mit.edu/hugo/montytagger/
8) Yahoo!デベロッパーネットワーク
http://developer.yahoo.co.jp/
9) NativeChecker
http://native-checker.com/native-checker/
10) 塙タカユキ: “総合英語 Forest”，桐原書店，2006.
11) The New York Times
http://www.nytimes.com/