論　文

依存構造に基づく用例文検索手法とその評価

加藤　芳秀[†]　　江川　誠二[††]　　松原　茂樹[†††]　　稲垣　康善[††††]

Sentence Retrieval Method Based on Dependency Structure and Its Evaluation

Yoshihide KATO[†], Seiji EGAWA[††], Shigeki MATSUBARA[†††], and Yasuyoshi INAGAKI[††††]

あらまし　本論文では，言語構造を活用した用例文検索手法を提案する．本手法では，クエリとしてキーワード系列を受け取ると，単純にキーワードにマッチする文を検出するのではなく，出現するキーワードが依存関係（単語の修飾・被修飾の関係）にある文を検出し，キーワードが形成する依存関係のパターンに従って文を分類する．依存関係のパターンは自動的に同定されるため，従来手法のようにユーザが構造的なクエリを作成する必要はなく，言語構造を活用した検索を容易に実行できる．被験者実験を通じて，提案手法が，ユーザの必要とする文を高い精度で分類できることを確認した．

キーワード　文検索，テキストコーパス，構文解析，精度，再現率

1. まえがき

外国語学習や言語現象の分析などの場面で，実際に使用されている文例，いわゆる用例文を調べることは有用である．ウェブ検索を活用した用例文検索 [8] など，様々なシステムが開発されている．

従来の用例文検索手法の多くは，いくつかのキーワードをクエリとして受け取り，それらを含む文を検出する．精度の高い検索を実現するために，文字の頻度情報を用いて検出された文を評価する手法 [8] や，文に出現するキーワード間の距離を定義し，ある距離以下でキーワードが出現する文のみを検出する手法 [6] などが提案されている．キーワードマッチングに基づくこれらの検索手法は，直観的でユーザビリティが高い反面，検索において言語構造を活用することができず，用例文を効果的に検出することは難しい．

一方，言語構造をユーザが明示的に指定して検索するシステムが提案されている．これらのシステムでは，句構造や依存構造のパターンをクエリとして文を検索する．

Corley らの提案する Gsearch [2] では，句構造パターンを生成するための文脈自由文法をユーザ自身が設計し，文法，及び文法で定義した統語範疇の系列をシステムに与える．システムは与えられた文法を用いて文を構文解析し，与えられた統語範疇系列を構文解析結果に含む文を検索結果として返す．

兵藤らの手法 [3] や，Resnik らの提案する Linguist's Search Engine（LSE）[7] では，ユーザはまず，検出したい文の一例を入力する．入力された例文は構文解析され，その結果が提示される．ユーザは提示された構文解析結果を編集し，構造的なパターンを作成する．システムは，編集により得られたパターンにマッチする構造をもつ文を検索結果として提示する．

しかし，これらのシステムは，キーワードに基づく検索のような簡単で直観的なインタフェースを実現しているとは言いがたい．文法の設計や句構造パターンの編集ができる程度の文法的知識をもたないユーザが，これらのシステムを利用するのは困難である．

そこで本論文では，キーワードベースの直観的なインタフェースを実現しつつ，言語構造を活用する用

[†] 名古屋大学大学院国際開発研究科，名古屋市
Graduate School of International Development, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
[††] 名古屋大学大学院情報科学研究科，名古屋市
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
[†††] 名古屋大学情報連携基盤センター，名古屋市
Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
[††††] 豊橋技術科学大学，豊橋市
Toyohashi University of Technology, 1–1 Hibarigaoka, Tempaku-cho, Toyohashi-shi, 441–8580 Japan

例文検索手法を提案する(注1)．本手法のクエリはキーワード系列であるが，単にキーワードにマッチする文を検出するのではなく，キーワードが依存関係（単語間の修飾・被修飾の関係）にある文を優先的に検出する．これにより，用例として適した文を検出することができる．更に，検出された各文に対して，キーワード系列が文中で形成する依存構造パターンを同定し，パターンごとに文を分類する．ユーザは，依存構造パターンから，検出された文においてキーワード同士がどのような関係にあるかを知ることができる．また，特定のパターンをもった文を探す場合にも，ユーザはいくつかの依存構造パターンの中から，必要とするパターンを探すだけでよく，文法の設計や構造的なパターンの編集は不要である．

本論文の構成は以下のとおりである．次の **2.** では，提案する用例文検索手法の基本アイデアについて述べる．**3.** では，提案手法の中心をなす依存構造パターン同定アルゴリズムを説明する．**4.** では，依存構造パターンに基づく文の分類について述べる．**5.** では，実験による提案手法の評価について報告する．**6.** は，本論文のまとめである．

2. 依存構造に基づく用例文検索

本論文で提案する用例文検索手法は，次の特徴をもつ．

（1） ユーザからのクエリは，キーワード系列のみとする．それ以外の情報は要求しない．

（2） 出現するキーワードの間に依存関係が成り立つ文を検索結果として返す．

（3） 検索結果の提示においては，出現するキーワードの間に成り立つ依存関係の種類に従って文を分類する．

特徴（1）により，直観的なインタフェースをユーザに提供する．特徴（2）及び（3）を備えることにより，我々は効果的な用例文検索を実現できると考えている．これらの実際の効果については，**5.** において改めて検討するが，本章の以下では，提案手法のイメージをつかむために，具体的な例を用いて特徴（2）と（3）について説明する．

提案手法では，ユーザから次のようなキーワードからなるクエリを受け取る．

parse sentence in 名詞 (1)

キーワードは単語か品詞であり，この例ではキーワード「名詞」は品詞である．このようなクエリに対して，単にこれらのキーワードを含む文を検索結果として返すのではなく，出現するキーワードの間に依存関係が成り立つ文，例えば，次のような文を返す．

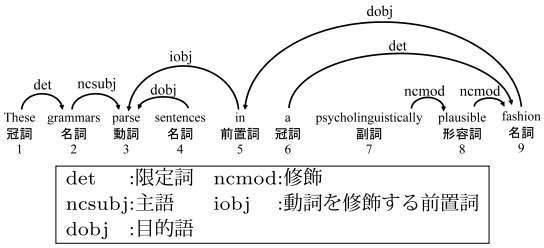

図 1 “These grammars parse sentences in a psycholinguistically plausible fashion.” の依存構造
Fig. 1 Dependency structure for “These grammars parse sentences in a psycholinguistically plausible fashion.”

（a） These grammars *parse sentences in* a psycholinguistically plausible *fashion*.

この文において，“sentence” と “in” は “parse” に依存し，名詞 “fashion” は “in” に依存している（図 1 参照．矢印は依存関係を表す）．

一方，キーワードが出現していても，それらの間に依存関係が成り立たない文，例えば次の文 (b) は検索結果として返さない(注2)．

（b） We want to map phrase structure trees to *parses* which predict the words of the *sentence in* their left-to-right *order*.

このような文は，用例文として適切ではないと考えられる．このような文を検索結果から排除するのが特徴（2）である．

次に検索結果の分類（特徴（3））について検討するために，別の文

（c） We began by *parsing* the *sentences in* the multilingual *corpus*.

について考える．この文においても，図 2 に示すように，キーワードの間に依存関係が成り立っているが，その依存関係は文 (a) と異なっている．すなわち，文 (a) において，“in” は，“parse” に依存しており，文 (c) においては，“in” は，“sentence” に依存している．いずれの文においても，出現するキーワードの間

(注1)：本論文は，研究速報として既発表の文献 [5] の内容を発展させ，まとめたものである．
(注2)：文 (b) においては，名詞 “order” が “in” に依存するという依存関係があるのみで，その他のキーワードには依存関係がない．

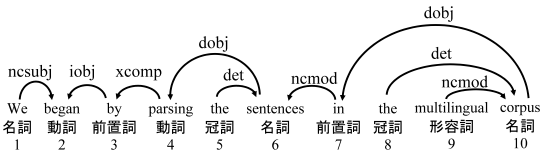

図 2 “We began by parsing the sentences in the multilingual corpus.” の依存構造
Fig. 2 Dependency structure for “We began by parsing the sentences in the multilingual corpus.”

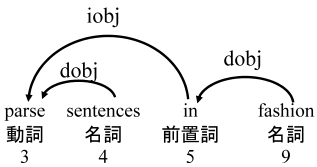

図 3 “parse sentence in 名詞” に対する依存構造パターンの例
Fig. 3 An example of dependency structure pattern for “parse sentence in *noun*.”

に依存関係が成り立つため，これらは検索結果に含まれることになるが，実際にユーザが必要とする文は，状況に応じて (a) のタイプの文である場合もあれば，(c) のタイプの文である場合もある．あるいは別のタイプの文である可能性もある．キーワード情報のみからいずれのタイプが適切であるかを特定することは困難であるものの，文 (a) と文 (c) を混在させて検索結果を提示するのではなく，それらを分けて提示すれば，ユーザはそのつど，必要とするタイプの文を容易に見つけられる．これが本手法の特徴（3）である．

3. 依存構造パターン同定

前章で述べた用例文検索手法を実現するためには，文中に出現するキーワードに依存構造が成り立つかどうかを判定し，キーワードの間に成り立つ依存関係の種類を明らかにしなければならない．これを実現するのが，本章で提案する依存構造パターン同定アルゴリズムである．

提案手法の前提として，検索対象の文には依存構造が付与されているものとする．以下では，検索対象の文とその依存構造の対の集合をコーパスと呼ぶ．ユーザのクエリは，キーワード（単語，または品詞）の系列である．

提案手法では，ユーザのクエリに対して，クエリ中のキーワードをその順序通りに含む文を検出し，検出された各文に対して，出現するキーワードが形成する依存構造パターンを同定する．例として，キーワード系列 (1) と図 1 の文，及び依存構造について考える．依存構造からキーワードにかかわっている依存関係だけを取り出すと図 3 のようになる．提案する依存構造パターン同定アルゴリズムはこのようなパターンを生成する．提案アルゴリズムでは，出現するキーワードの間に依存関係が成り立たない文に対してはパターンを生成しないので，パターン生成の成否により出現するキーワード間の依存関係の有無を判定できる．

3.1 依存構造パターンの同定手法

依存構造パターン同定アルゴリズムは，キーワード系列からボトムアップに依存構造パターンを生成することにより，キーワードが文中で形成する依存構造パターンを同定する．アルゴリズムが出力する依存構造パターンは，文の依存構造にマッチするパターンのみである．

入力として，

クエリ：$q_1 \cdots q_m$ ($q_1, \ldots, q_m$ はキーワード)

文：$s = w_1 \cdots w_n$ ($w_1, \ldots, w_n$ は単語と品詞のペア)

依存構造（＝依存関係の集合）：D

を受け取り，依存構造パターンの集合を出力する．ここで，D は，文 s に出現する単語間の依存関係の集合である．w_j が w_k に依存し，その関係が r であるとき，3 項組 (j, k, r) は D の要素である．以下では，(j, k, r) を $j \xrightarrow{r} k$ と書くことにする．

依存構造パターンは，5 項組 $d = (h, D_L, D_R, T_L, T_R)$ である．h は単語位置であり，これを d の主辞と呼ぶ．D_L，及び D_R は，依存構造パターンのリストである．すなわち，依存構造パターンは再帰的に定義されている．D_L は，その要素である依存構造パターンの主辞が左から h に依存することを意味する．D_R の場合は，右から依存することを意味する．T_L は依存関係の種類のリストである．$|T_L| = |D_L|$ であり，T_L の i 番目の要素 r_i は，D_L の i 番目の要素の主辞と h の間の依存関係が，r_i であることを意味する．T_R についても同様である．

本アルゴリズムは，クエリ $q_1 \cdots q_m$ に対して，以下の操作をボトムアップに適用することにより，クエリ中のキーワードが文 s において形成する依存構造パターンを生成する．これらの操作により生成される依存構造パターンは，依存構造 D とマッチするものに限られるため，クエリ $q_1 \cdots q_m$ に対して生成される依存構造パターンも，D とマッチすることが保障される．

[初期化]　各 $q_i (1 \leq i \leq m)$, $w_j (1 \leq j \leq n)$ に対し

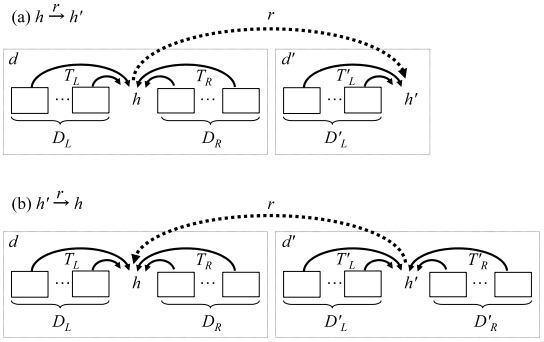

図 4 結合操作
Fig. 4 Combining.

て，q_i が単語 w_j あるいはその品詞にマッチするならば，q_i に対する依存構造パターンとして $(j, \varepsilon, \varepsilon, \varepsilon, \varepsilon)$ を生成する．なお，ε は空リストを表す．

[結合操作] $d = (h, D_L, D_R, T_L, T_R)$，及び $d' = (h', D'_L, D'_R, T'_L, T'_R)$ をそれぞれ，$q_i \cdots q_j$，及び $q_{j+1} \cdots q_k$ に対する依存構造パターンとし，d 中の最も右に出現する単語位置が，d' 中の最も左に位置する単語位置よりも小さいものとする．このとき，$h \xrightarrow{r} h' \in D$ かつ $D'_R = \varepsilon$ ならば，$q_1 \cdots q_j q_{j+1} \cdots q_k$ に対して，依存構造パターン $(h', dD'_L, D'_R, rT'_L, T'_R)$ を生成する（図 4 (a) 参照）(注3)．$h' \xrightarrow{r} h \in D$ ならば，$q_1 \cdots q_j q_{j+1} \cdots q_k$ に対して，依存構造パターン $(h, D_L, D_R d, T_L, T_R r)$ を生成する（図 4 (b) 参照）．

結合操作を繰り返し適用することにより，各キーワードが，別のキーワードに直接依存するようなすべての依存構造パターンを生成できる．逆に，クエリ全体に対してパターンが生成されないことは，文中におけるキーワードに直接の依存関係が成り立たないことを意味する．

3.2 依存構造パターン同定の例

前節で挙げた例，すなわち，クエリ (1) と図 1 の文，及び依存構造に対する，依存構造パターン同定を考える．

まず，クエリ中の 1 番目のキーワード “parse” は，文中の 3 番目の単語とマッチするので，初期化操作により，

$$(3, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \tag{2}$$

を生成する．同様にして，“sentence” に対しては，

$$(4, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \tag{3}$$

を，“in” に対しては

$$(5, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \tag{4}$$

を，「名詞」に対しては，

$$(2, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \tag{5}$$

$$(4, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \tag{6}$$

$$(9, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \tag{7}$$

を生成する．

(2) と (3) は，クエリ中の隣接するキーワード列に対する依存構造パターンであるが，その主辞に注目すると，依存関係 $4 \overset{dobj}{\rightarrow} 3$ が成り立つ．したがって，これらを結合し，依存構造パターン

$$(3, \varepsilon, \langle (4, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \rangle, \varepsilon, \langle dobj \rangle) \tag{8}$$

を生成する．更に，(4) と (7) も隣接するキーワード列に対する依存構造パターンである．その主辞間には，依存関係 $9 \overset{dobj}{\rightarrow} 5$ が成り立つので，“in 名詞” に対して，依存構造パターン

$$(5, \varepsilon, \langle (9, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \rangle, \varepsilon, \langle dobj \rangle) \tag{9}$$

を生成する．生成された (8) と (9) もまた隣接するキーワード列に対する依存構造パターンであり，その主辞の間には，$5 \overset{iobj}{\rightarrow} 3$ が成り立つので，依存構造パターン (10) が生成される．

$$\begin{aligned} &(3, \varepsilon, \\ &\quad \langle (4, \varepsilon, \varepsilon, \varepsilon, \varepsilon),\ (5, \varepsilon, \langle (9, \varepsilon, \varepsilon, \varepsilon, \varepsilon) \rangle, \varepsilon, \langle dobj \rangle) \rangle, \\ &\quad \varepsilon, \\ &\quad \langle dobj,\ iobj \rangle) \end{aligned} \tag{10}$$

(10) を図示したものが，図 3 の依存構造パターンである．

以上のようにして，クエリ (1) 中のキーワードは，図 1 の文において図 3 の依存構造パターンを形成することが分かる．

一方，キーワードの間に依存関係が成り立たないような文（例えば，**2.** で挙げた文 (b)）に対しては，結合操作によって依存構造パターンは生成されない．

(注3)：$D'_R = \varepsilon$ という条件は，同一の依存構造パターンを重複して生成するのを防ぐためのものである．この条件がない場合，ある単語に対して左右両側から別の単語が依存するような依存構造パターンを生成するとき，(1) 左側の単語を考慮してから右側の単語を考慮する場合と，(2) 右側の単語を考慮してから左側の単語を考慮する場合があり，同一の依存構造パターンが複数生成されてしまう．

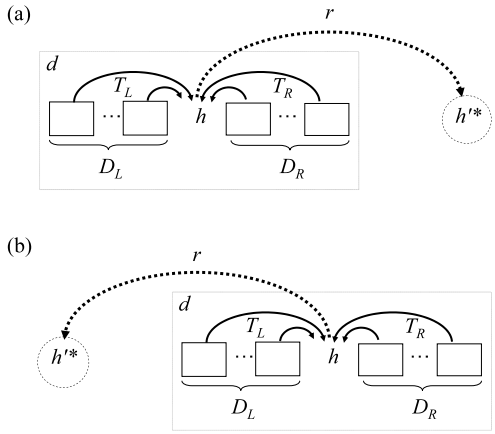

図 5　補 完 操 作
Fig. 5　Interpolation.

3.3 補 完 操 作

文中に出現するキーワードが，別のキーワードに直接依存しないような文を検出したいケースも考えられる．例として，次のクエリと図 1 の依存構造について考える．

parse sentence fashion　　(11)

このクエリに対して結合操作をいくら適用しても，依存構造パターンは生成できない．“fashion” は，“parse” とも “sentence” とも依存関係にないからである．そこで，このようなクエリに対しても頑健に依存構造パターンを同定するために，次の補完操作を導入する．

[補完操作]　d を $q_i \cdots q_j$ に対する依存構造パターンとし，その主辞を h とする．また，ある h'，及び r が存在し，$h \xrightarrow{r} h' \in D$ とする．このとき，d 中の最も右に出現する単語位置より h' が大きいならば，$q_i \cdots q_j$ に対して，依存構造パターン $(h'^*, d, \varepsilon, r, \varepsilon)$ を生成する（図 5 (a) 参照）．d 中の最も左に出現する単語位置より h' が小さいならば，$q_i \cdots q_j$ に対して，依存構造パターン $(h'^*, \varepsilon, d, \varepsilon, r)$ を生成する（図 5 (b) 参照）．

主辞 h' に付与された記号 $*$ は，h' が補完操作により導入されたものであることを意味する．

この操作により，キーワードに直接の依存関係が成り立たないような文に対しても依存構造パターンを生成できる．しかし，補完操作を無制限に適用すると，キーワードが出現するだけの用例に適さない文に対しても依存構造パターンを生成してしまう．そこで本手法では，補完操作に対してコストを与える．補完操作により生成された依存構造パターンのコストを，その生成に用いられた補完操作の回数と定義する．コストに対するしきい値を設定し，しきい値以下の依存構造パターンのみを生成することにより，補完操作の適用回数の少ない依存構造パターンのみ生成できる．

3.4 補完操作の例

クエリ (11) と図 1 の文，及び依存構造に対する，依存構造パターン同定を考える．コストのしきい値を 0 に設定した場合，これに対する依存構造パターンは生成されない．コストのしきい値を 1 に設定した場合，次のようにして依存構造パターンが生成される．“fashion” に対しては，依存構造パターン (7) が生成される．依存関係 $9 \overset{dobj}{\rightarrow} 5$ が成り立つので，依存構造パターン (7) に対して補完操作を適用することにより，“fashion” に対して次の依存構造パターンが生成される．

$$(5^*, \varepsilon, \langle(9, \varepsilon, \varepsilon, \varepsilon, \varepsilon)\rangle, \varepsilon, \langle dobj\rangle) \qquad (12)$$

この依存構造パターンのコストは 1 である．続いて，この依存構造パターンと (8) は隣接するキーワード列に対する依存構造パターンであるので，結合操作が適用され，図 3 の依存構造パターンが生成される．ただし，ここにおいて，前置詞 “in” は補完操作により補われたものである．

このように，補完操作を導入することにより，出現するキーワードの間に直接の依存関係がない場合でも，依存構造パターンを生成することができる．

3.5 アルゴリズム

図 6 に依存構造パターン同定アルゴリズムを示す．$D[i, j, c]$ は，過程で生成される $q_{i+1} \cdots q_j$ に対するコスト c の依存構造パターンを記録する．$rm(d)$ は，d 中の最も右に位置する単語位置，$lm(d')$ は，d' 中の最も左に出現する単語位置を表す．

アルゴリズムの概要を述べる．最初に，初期化操作により，クエリ中の各キーワードに対して依存構造パターンを生成する．次に，結合操作，及び補完操作をボトムアップに適用し，クエリ中のキーワード列に対する依存構造パターンを生成する．コスト 0 の依存構造パターンから生成し，順次，コストのより大きい依存構造パターンを生成する．$cost$ は，生成する依存構造パターンのコストを制御するための変数であり，0 からしきい値 θ までの値が順に与えられる．最後に，クエリに対して生成された依存構造パターンを返す．

input: query $q_1 \cdots q_m$, sentence $w_1 \cdots w_n$, dependency structure D

initialization:
for $i = 1$ **to** m **do**
  for each j s.t. $w_j = q_i$ **do**
    push $(j, \varepsilon, \varepsilon, \varepsilon, \varepsilon)$ to $D[i-1, i, 0]$;

for $cost = 0$ **to** θ **do**
  combining:
  for $k = 2$ **to** m **do**
    for $j = k-1$ **down to** 1 **do**
      for $i = j-1$ **down to** 0 **do**
        for $c = o$ **to** $cost$ **do**
          for each $d = (h, D_L, D_R, T_L, T_R) \in D[i, j, c]$, $d' = (h', D'_L, D'_R, T'_L, T'_R) \in D[j, k, cost - c]$
                s.t. $rm(d) < lm(d')$ **do**
            if $h \xrightarrow{r} h' \in D \wedge D'_R = \varepsilon$ **then**
              push $(h', dD'_L, D'_R, rT'_L, T'_R)$ to $D[i, k, cost]$;
            if $h' \xrightarrow{r} h \in D$ **then**
              push $(h, D_L, D_R d', T_L, T_R r)$ to $D[i, k, cost]$;

  interpolation:
  for $j = 1$ **to** m **do**
    for $i = j-1$ **down to** 0 **do**
      if $i \neq 0 \vee j \neq m$ **then**
        for each $d = (h, D_L, D_R, T_L, T_R) \in D[i, j, cost]$ **do**
          for h' s.t. $h \xrightarrow{r} h' \in D$ **do**
            if $h' > rm(d)$ **then**
              push $(h'^*, d, \varepsilon, r, \varepsilon)$ to $D[i, j, cost+1]$;
            else if $h' < lm(d)$ **then**
              push $(h'^*, \varepsilon, d, \varepsilon, r)$ to $D[i, j, cost+1]$;

return $D[0, m, 0] \cup \cdots \cup D[0, m, cost]$;

図 6　依存構造パターン同定アルゴリズム
Fig. 6　An algorithm of identifying dependency structure patterns.

4. 依存構造パターンに基づく文の分類

提案する用例文検索手法では，まず，**3.** で述べた依存構造パターン同定アルゴリズムにより，文中に出現するキーワードが形成する依存構造パターンを同定する．同定された依存構造パターンが同一の文をまとめることにより文を分類する．ただし，依存構造パターン中の単語位置については考慮しない．具体的には，以下に定義する同値関係 $\equiv$ が成り立つとき，同一の依存構造パターンとみなす．なお，$aster(h)$ は，h に $*$ が付与されているとき 1 を返し，それ以外のとき 0 を返すものとする．また，$D_{L,i}$ は D_L の i 番目の要素（依存構造パターン）を表す．$D'_{L,i}$, $D_{R,i}$ 及び $D'_{R,i}$ についても同様である．

[依存構造パターンの同一性]　$d = (h, D_L, D_R, T_L, T_R)$ 及び $d' = (h', D'_L, D'_R, T'_L, T'_R)$ を依存構造パターンとする．条件（1）〜（7）をすべて満たすとき，$d \equiv d'$ と定義する．

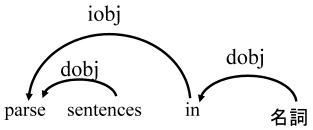

図 7　“parse sentence in 名詞” に対する依存構造パターンの例
Fig. 7　An example of dependency structure pattern for “parse sentence in *noun*.”

（1）　$aster(h) = aster(h')$.
（2）　$|D_L| = |D'_L|$.
（3）　$|D_R| = |D'_R|$.
（4）　$\forall i (1 \leq i \leq |D_L|), D_{L,i} \equiv D'_{L,i}$.
（5）　$\forall i (1 \leq i \leq |D_R|), D_{R,i} \equiv D'_{R,i}$.
（6）　$T_L = T'_L$.
（7）　$T_R = T'_R$.

例としてクエリ (1) について考える．このクエリと次の文に対しては，図 7 の依存構造パターンが同定される．

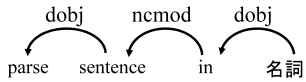

図 8 “parse sentence in 名詞” に対する依存構造パターンの別の例
Fig. 8 Another example of dependency structure pattern for “parse sentence in *noun*.”

- These grammars *parse sentences in* a psycholinguistically plausible *fashion*. (=文 (a))
- Our method can still *parse* large *sentences in* a reasonable *amount* of time.

同定される依存構造パターンは同一であるため，これらは同じグループにまとめられる．これは，“in” で始まる前置詞句が動詞 “parse” に依存する文のグループである．

一方，次のような文に対しては図 8 のような依存構造パターンが同定される．

- We began by *parsing* the *sentences in* the multilingual *corpus*. (=文 (c))
- We *parse* all the *sentences in* the domain document *collection*.

これらの文は上述とは別のグループにまとめられる．“in” で始まる前置詞句が名詞 “sentence” に依存する文のグループである．

このように，依存構造パターン同定アルゴリズムにより，キーワード間の関係に従って文を分類することができる．

5. 実験による評価

2. では，提案手法の特徴を例を用いて説明したが，その効果の程度については明らかではない．本章では，提案手法の有効性を定量的に評価する実験について報告する．

提案手法では，依存構造パターンに従って文を分類する．**2.** で述べたように，必要とする文をユーザが容易に見つけられるようにするためである．提案手法がそのような目的に適っているとすれば，ユーザが必要とする文とそうでない文とを分類できるはずである．そこで本実験では，そのような観点から提案手法の精度と再現率を評価する．

実験の概略は以下のとおりである．まず，検索対象となる依存構造付き英文コーパスを作成した．次に，これに対するクエリ，並びにそれに適合する英文を被験者実験を通じて収集した．収集したデータに対して，提案手法が，クエリに適合する英文とそうでない英文に分類できるかにより精度と再現率を評価した．

以下では，各項目について順に説明する．

5.1 英文コーパスの作成

検索対象となる英文コーパスは，情報工学分野の英語論文（PDF ファイル）をもとに作成した．作成手順は以下のとおりである．

（1） PDF 形式の論文ファイルを pdf2html [9] により XML 形式のファイルに変換する．XML 形式のファイルには，テキストとその位置情報が保存される．

（2） 文献 [4] の手法（テキストの位置情報から段落を同定する手法）に基づき，XML ファイルから段落（本文）のみを抽出する．

（3） 依存構造解析器 RASP [1] を用いて，得られたテキストを解析し，依存構造を付与する．

以上の手順により，英文 185,488 文からなるコーパスが得られた．

5.2 被験者実験による正解データの作成

提案手法の定量的評価のために，正解データを作成した．正解データは，

- 検索クエリ
- 検索クエリに適合する英文セット
- 検索クエリに適合しない英文セット

からなる．一般に，クエリに対する英文の適合性は，クエリを入力したユーザの判断による．このため，本研究では，被験者実験により正解データを作成した．

被験者は，英文用例を検索するためのクエリを作成し，それに対する英文の適合性を判定する．クエリの作成にあたっては，被験者に英文検索を動機づけることを目的に，英作文に関する課題を作成し，提示した．一方，適合性の判定は，コーパス中のすべての英文を対象とするのが理想的であるが，英文の数が膨大であり，被験者の負担を考えると現実的ではない．このため，本実験では，コーパスからクエリごとに 20 文程度の英文を取り出し，適合性判定の対象として提示した．英文を取り出す手順は以下のとおりである．

（1） クエリ中のキーワードをその順に含むコーパス上のすべての英文を，依存構造パターンに従って分類する．

（2） 各依存構造パターンに対応する文集合から文を取り出す．各々の文集合から少なくとも 1 文，最大で 10 文取り出す．取り出す文の合計は原則 20 文とするが，パターンの総数が 20 を超える場合，取り出す文は 20 文を超えて，また，文の総数が 20 文に満たな

表 1　クエリの分布
Table 1　Distribution of queries.

クエリの長さ（語）	クエリ数
1	42　(11)
2	128　(34)
3	95　(27)
4	26　(9)
5	11　(1)

括弧内は適合する文が検出されたクエリの数

い場合は，すべての文を取り出すこととする．
なお，依存構造パターン同定にコストを設けることの効果についても評価するために，英作文の課題ごとにコストの有無を定めた．すなわち，全体の 1/3 の課題はコストのしきい値を 1 とし，残りの 2/3 の課題をしきい値 0 とした．

実験にあたっては，依存関係の情報が適合性の判定に影響を与えないようにするために，被験者には，これらの英文をランダムに並べ換えたリストを提示した．依存関係の情報は取り除かれており，英文以外の情報は一切提示されない．提示された英文に対して，被験者は各々の判断で，文が用例文として適合するか否かを判定する．

以上の要領で被験者実験を実施した．被験者は，情報工学の研究に従事する学生 7 名である．検索の実行回数は，合計 302 回であった．302 回のクエリの長さの平均は，2.5 語であった．表 1 にその分布を示す．302 回の検索のうち，68 回の検索については，該当する文が存在しなかった．コーパス中に存在しないキーワード（タイプミスも含む）の使用や，依存構造パターンが生成される文が存在しなかったことが原因である(注4)．58 回の検索については，適合性の判定がなされなかった．94 回の検索については，文は提示されたが，適合する文は存在しないと判断された．残りの 82 回の検索については，適合する文があると判断された．以下では，この 82 回の検索のうち，1 語からなるクエリによる検索を除いた 71 回の検索についての精度と再現率を評価する．1 語からなるクエリを除外するのは，提案手法とキーワードベースの手法が，これらのクエリに対して同様の動作をするからである．

5.3　評価手法

収集したクエリ，並びに適合性判定の結果を用いて，提案手法がクエリに適合する文を分類できるか否かを定量的に評価した．情報検索の分野でよく用いられる評価尺度である精度と再現率の概念をベースにした評価である．以下では，評価方法について説明する．

表 2　精度と再現率
Table 2　Precision and recall.

	精度（%）	再現率（%）	F 値
ベースライン 1	44.2	100.0	0.613
ベースライン 2	100.0	45.9	0.629
提案手法	69.0	77.9	0.732

提案手法では，生成された依存構造パターンに従って文が分類される．生成された依存構造パターンを $d_1, \ldots, d_l$，依存構造パターンが $d_i (1 \leq i \leq l)$ である英文の集合を S_{d_i} とし，クエリに適合する文の集合を C とすると，C と完全に一致する文集合 S_d^* が検索結果 $\{S_{d_1}, \ldots, S_{d_l}\}$ の中に存在するのが理想的な分類である．そこで，本実験では，C と一致する文集合が検索結果 $\{S_{d_1}, \ldots, S_{d_l}\}$ に存在するか否かにより手法を評価する．ただし，C と完全に一致する S_{d_i} が存在しない場合についても，その一致の程度を評価するために，精度と再現率を用いる．具体的な評価方法は，以下のとおりである．

（1）　各 S_{d_i} について，精度 P_{d_i} と再現率 R_{d_i} により C との一致の程度を評価する．

（2）　精度と再現率の F 値 $= \frac{2P_{d_i}R_{d_i}}{P_{d_i}+R_{d_i}}$ が最大である（すなわち，C と最も一致している）S_{d_i} を S_d^* とみなし，その精度と再現率を分類の精度と再現率とする．

この評価方法では，C と完全に一致する S_{d_i} が存在するならば，分類の精度と再現率はともに 100%である．

なお，S_{d_i} の精度と再現率は次のように定義する．

$$\text{精度 } P_{d_i} = \frac{|S_{d_i} \cap C|}{|S_{d_i}|} \tag{13}$$

$$\text{再現率 } R_{d_i} = \frac{|S_{d_i} \cap C|}{|C|} \tag{14}$$

複数のクエリに関する精度と再現率はそれぞれ，分類の精度の平均，再現率の平均として定義する．

5.4　実験結果

71 回の検索に関する精度，再現率を表 2 に示す．ベースライン 1 は，検索結果全体が一つに分類されたとみなした場合である．したがって，再現率は常に 100%である．ベースライン 2 は，すべての文をそれぞれ別のクラスに分類した場合である．精度は常に 100%である．精度と再現率の総合的な指標である F 値は提案手法の方が高く，総合的に見て，依存構造パ

(注4)：依存構造パターンが生成されなかった文の中に，クエリに適合する文が存在する可能性もある．しかし，**5.6** で述べるように，そのようなケースは少ない．

表 3　精度と再現率（誤り修正後）
Table 3　Precision and recall (after error correction).

	精度（%）	再現率（%）	F 値
誤り修正前	64.6	81.5	0.720
誤り修正後	62.7	89.0	0.736

ターンに基づく文の分類が，用例文の分類として効果的であると考えられる．

次に，依存構造解析器の解析誤りの影響について検討する．コーパスに誤った依存関係が付与されると，その結果，同定される依存構造パターンも誤ったものとなり，分類も誤ったものとなる．その影響を調べるため，71 件の検索において提示された文について，それに付与された依存関係の誤りを人手により修正し，依存関係が正しく付与された場合の精度と再現率を評価した．ただし，誤り修正において，いくつかの依存関係についてはその種類の特定が困難であったため，それらについては，その関係をまとめ上げ，いくつかの関係のうちのいずれかであることを表す関係を新たに導入し，これを付与した(注5)．解析誤りの影響のみを考慮するために，誤りを修正していないコーパスについても，まとめ上げた関係を導入して精度と再現率を計算した．表 3 が評価結果である．コーパスに誤りがある場合は，誤りがない場合に比べて，F 値は若干低い．また，個々の検索の精度と再現率を調べると，F 値が 1 であった検索，すなわち，クエリに適合する文がうまく分類できた検索の数は，コーパスに誤りがない場合は，20 件，コーパスに誤りがある場合は，12 件であった．これらの結果から，依存構造の誤りが分類に悪影響を及ぼしていることが分かる．一方で，現状の依存構造解析器の精度でも，提案手法が有用であることを確認した．

次に，同定された依存構造パターンの数を調べた．誤り修正前は，1 クエリ当り平均 6.3 パターン，修正後は，平均 5.4 パターンと少なかった．依存構造解析器の解析誤りにより誤った依存構造パターンが生成され，文の分類が不必要に細かくなってしまうという悪影響が見られる．

5.5　分類誤りの分析

本節では，分類誤りの主な原因について分析する．解析誤りのないコーパスを用いての分析である．

5.5.1　再現率 100%で精度が 100%でないケース

まず，再現率が 100%で精度が 100%でなかった検索，すなわち，クエリに適合する文を一つのクラスにまとめることができたが，より詳細な分類が必要な検索の件数を調べた．このような検索は 28 件あった．これらの検索については，分類により，ユーザが調べる範囲を絞り込んでおり，ある一定の分類効果は認められるケースである．このような分類誤りの原因として，次のようなものが考えられる．

- 文の適合性がキーワード以外の単語に依存するケース
- キーワードに品詞を使用するケース

以下では実例を挙げて説明する．

ある被験者は，クエリ “the same way” に対して，“The power is chosen in *the same way* as dPLRM.” や “We approached this problem in *the same way* as in the case of personal names.” を適合する文と判定し，“Reversed words and pseudowords are not reconstructed in *the same way*.” や “Much *the same way* that tying mixtures at the state level across phonemes sharing linguistic properties is used...” を適合しない文と判定した．これらの文はすべて，同一のクラスに分類された．適合する文に共通に観察され，適合しない文には見られない点は，“as” が “way” に依存していることである．すなわち，“as” と “way” の依存関係が適合性に影響を与えていると考えられるが，提案手法では，キーワード以外の依存関係については考慮しないため，これらの文を別のクラスに分類することはできない．28 件の検索のうち 14 件の検索でこのような現象が観察された．

次にキーワードに品詞を使用したケースについて考える．ある被験者は，クエリ “動詞 processing speed” に対して，“We can *improve* the photonic label *processing speed*.” を適合する文と判定し，“This approach *accelerates* the *processing speed* markedly.” や “We *compare* the *processing speed* of the proposed method.” を適合しない文と判定した．これら 3 文は，同一のクラスに分類される．被験者は，ある特定の単語（このケースでは “improve”）を探すことを意図していたと考えられるが，このような場合，あるクラスの一部の文のみがクエリに適合するため，精度は低くなってしまう．28 件の検索のうち 7 件の検索

(注5)：具体的には，(1) 前置詞句が，項 ($iobj$) か修飾句 ($ncmod$) のどちらか判断が難しいケース，(2) 不定詞句や節などが，補部 ($xcomp$, $ccomp$, $pcomp$) か修飾句 ($xmod$, $cmod$, $pmod$) のどちらか判断が難しいケースについて，両者のいずれかであることを表す関係を新たに導入した．

が，このようなケースに該当していた．

5. 5. 2 再現率が 100%でないケース

再現率が 100%でない 23 件の検索については，依存構造パターンによる分類が，何らかの理由でうまく機能しなかったと考えられる．それらの検索について調べたところ，文法体系が文の分類に適していないケースがあることが分かった．以下では実例を用いてこれについて説明する．

23 件の検索のうち，7 件については，依存関係の種類が要因であると考えられる．ある被験者は，クエリ “be useful for” に対して，“Document ranking *is useful for* the readability of retrieved results.” や This *is useful for* representing unclear speech. を適合であると判定した．これらの文において，“for” は “useful” に依存するが，その依存関係の種類は異なる．本実験で用いた RASP では，前置詞 “for” の項が名詞か動名詞かに応じて，“for” と “useful” の間に異なる種類の依存関係を与えるからである．その結果，生成される依存構造パターンは異なるものとなり，同一のクラスに分類できない．

文法体系に起因する分類誤りの別のケースとして，等位構造に対する依存関係の与え方が分類誤りの原因であると考えられる検索が，3 件存在した．クエリ “allow 名詞” に対して，“We do not *allow* cyclic *graphs*.” や “It *allows* modifier *insertion* and component order alterlation.” が適合すると判定された．これらの文において，“allow” と「名詞」（“graphs” と “insertion”）は動詞と目的語の関係にある．ところが，前者は，“graphs” が “allow” に直接依存するという構造をとる一方で，後者は，“insertion” が “and” に依存し，“and” が “allows” に依存するという構造をとる．このため，これらの文を同一のクラスに分類することができなかった．

5. 5. 3 正解データの問題

いくつかの検索については，キーワードにマッチした部分以外の場所に，ユーザが必要とした表現が偶然存在した．このような現象は，8 件の検索において観察された．以下では，実例を挙げてこれを説明する．

クエリ “be easy to” に対して，“It *is easy to* determine a rule that transforms B backs to A.” や “Such sequences *are easy* to map *to* lexical solutions.” が適合する文と判定された．クエリ中の “to” は不定詞を導く to であると考えられる．後者の文において，キーワード “to” は前置詞句を導く to にマッチしているが，不定詞を導く to も文中に存在するため，被験者は適合すると判定したと考えられる．一方，これらの文に対して生成される依存構造パターンは異なるため，同一のクラスに分類されない．しかしながら，このようなケースは，分類誤りというよりも正解データの問題と位置づけられる．

表 4 コストとクエリに適合した文の数の関係
Table 4 Relation between cost and number of sentences relevant to queries.

生成された依存構造パターンのコスト	クエリに適合した文の数	文の総数
0	98	243
1	24	218
2 以上	5	85

5. 6 コストの観点からの評価

本手法では，出現するキーワードの間に依存関係が成り立つ文は，用例文に適していると仮定している．この仮定が正しいとすれば，上述の実験において，クエリに適合する文の多くに対しては，コスト 0 の依存構造パターンが生成されるはずである．そこで，71 クエリのうち，コスト 0 と 1 の依存構造パターンが生成されたクエリ，すなわち，コストのしきい値を 1 として依存構造パターンを生成した 23 クエリについて，それに適合する文の依存構造パターンのコストを調べた．結果を表 4 に示す．なお，解析誤り修正の影響で，コスト 2 以上の依存構造パターンが生成された文が結果には含まれている．この結果が示すように，適合する文の多くに対して，コスト 0 の依存構造パターンが生成されている（χ^2 検定を行った結果，有意差（$p < 0.01$）が見られた）．したがって，出現するキーワードの間に依存関係が成り立つ文の方が，用例文に適していると考えられる．一方で，コスト 1 の依存構造パターンをもつ文で適合する文も若干存在している．これは，補完操作によりクエリに適合する文が検出されることを意味し，補完操作によってより頑健な検索が可能になるといえる．

6. むすび

本論文では，キーワード系列が文中において形成する依存構造パターンを同定し，同定された依存構造パターンに基づき文を分類する用例文検索手法を提案した．被験者実験により，出現するキーワード間に依存関係が成り立つ文が，検索結果として提示される用例として適していることを確認した．また，形成される

依存構造パターンが同一である文をまとめることにより，ユーザが必要とする文とそうでない文とに分類できることを確認した．

依存構造解析の解析誤りにより，本来形成し得ない依存構造パターンが同定されることがある．この問題を回避するために，依存構造解析の解析結果から信頼性の高い部分のみを利用することが考えられるが，その具体的な方法については今後検討したい．また，より詳細な分類を実現するためには，キーワード間に成り立つ文法的な関係だけでなく，キーワードとそれ以外の単語との関係も考慮する必要があるが，それについても今後検討したい．

文　献

[1] T. Briscoe, J. Carroll, and R. Watson, “The second release of the RASP system,” Proc. COLING/ACL 2006 Interactive Presentation Sessions, pp.77–80, Sydney, Australia, July 2006.

[2] S. Corley, M. Corley, F. Keller, M. Crocker, and S. Trewin, “Finding syntactic structure in unparsed corpora: The gsearch corpus query system,” Computers and the Humanities, vol.35, no.2, pp.81–94, 2001.

[3] 兵藤安昭，河田実成，応江　黔，池田尚志，“構文付きコーパスの作成と類似用例検索システムへの応用，” 自然言語処理，vol.3, no.2, pp.73–88, 1996.

[4] Y. Ishitani, “Logical structure analysis of document images based on emergent computation,” IEICE Trans. Inf. & Syst., vol.E88-D, no.8, pp.1831–1842, Aug. 2005.

[5] 加藤芳秀，松原茂樹，稲垣康善，“依存構造に基づくコーパス検索，” 信学論（D），vol.J89-D, no.12, pp.2766–2770, Dec. 2006.

[6] 難波英嗣，森下智史，相沢輝昭，“論文データベースからのイディオム用例検索，” 情処学 NL 研報，NL-170, pp.53–59, 2005.

[7] P. Resnik and A. Elkiss, “The linguist’s search engine: An overview,” Proc. ACL Interactive Poseter and Demonstration Sessions, pp.33–36, June 2005.

[8] K. Tanaka-Ishii and H. Nakagawa, “A multilingual usage consultation tool based on Internet searching — More than a search engine, less than QA,” Proc. 14th International Conf. on World Wide Web, pp.363–371, Chiba, Japan, May 2005.

[9] http://pdftohtml.sourceforge.net/

（平成 20 年 5 月 30 日受付，9 月 2 日再受付）

加藤　芳秀　（正員）

1997 名大・工・情報卒．2003 同大大学院博士課程了．博士（工学）．同年，同大大学院国際開発研究科助手．現在，助教．自然言語処理，情報検索に関する研究に従事．情報処理学会，言語処理学会，ACL 各会員．

江川　誠二

2007 名大・工・情報卒．現在，同大大学院博士前期課程在学中．自然言語処理に関する研究に従事．情報処理学会会員．

松原　茂樹　（正員）

1993 名工大・工・電気情報卒．1998 名大大学院博士課程了．博士（工学）．同年，同大助手を経て，2002 名古屋大学情報連携基盤センター助教授．現在，准教授．この間，日本学術振興会特別研究員，ATR 音声言語コミュニケーション研究所客員研究員，情報通信研究機構研究員を兼任．自然言語処理，情報検索，ディジタル図書館の研究に従事．情報処理学会，人工知能学会，言語処理学会，IEEE，ACM 各会員．

稲垣　康善　（名誉員）

1962 名大・工・電子卒．1967 同大大学院博士課程了．同大助教授，三重大学教授を経て，1981 名古屋大学工学部教授．1997 同工学部長・工学研究科長．2003 名古屋大学名誉教授，愛知県立大学情報科学部教授．2007 愛知県立大学名誉教授，愛知工業大学経営情報学部教授．2008 豊橋技術科学大学理事・副学長．工博．コンピュテーションとコミュニケーションの理論，オートマトン言語理論，ソフトウェア基礎論，自然言語処理に関する研究に従事．本会情報・システムソサイエティ会長，副会長を歴任，功績賞受賞．情報処理学会名誉会員・フェロー，日本ソフトウェア科学会，人工知能学会，言語処理学会，IEEE，ACM，EATCS 各会員．