論　文

KeyGraph：語の共起グラフの分割・統合によるキーワード抽出

大澤　幸生[†]　　ネルス E.ベンソン[†]　　谷内田正彦[†]

KeyGraph: Automatic Indexing by Segmenting and Unifing Co-occurrence Graphs

Yukio OHSAWA[†], Nels E. BENSON[†], and Masahiko YACHIDA[†]

あらまし　文書検索において，検索対象となる各文書からキーワードを抽出しておくことは，検索時間の短縮と検索結果の質の向上の両面において重要である．本論文では，文書の主張の内容を表すキーワードの抽出を目指す．そのようなキーワードは文書の要約をしたり，ユーザの検索語に近い主張をもつ文書を検索したりするのに役立つであろう．このような目的にとっては，ユーザの興味に関連する既存分野を代表する文献を選ぶ場合とは違い，検索対象文書の著者独特の考えを把握する新たな技術が必要となる．そこで本論文では，文書は著者独自の考えを主張するために書かれており，その考えを表すために内容が構成されるという仮定に基づき，共起グラフの分割・統合操作によってキーワードを抽出する新しい手法 KeyGraph を提案し評価する．

キーワード　文書検索，キーワード抽出，著者の主張，共起グラフ

1. まえがき

近年，電子図書館やインターネットの発達につれてさまざまな書類が電子的に蓄えられ，ユーザは自宅からでも大量の文書にアクセスできるようになった．しかし，大量の情報が身近に提供されても，ユーザにそのすべてを読む時間はない．したがって，ユーザにとって重要な文書を計算機が文書データベースから選び出してくる検索技術は重要な使命を担っている．

ここで，検索対象の各文書から内容を要約するキーワードを抽出しておくことは重要である．というのは，ユーザの検索語（ユーザが自分の興味を表すために入力した語すなわち単語あるいは熟語の集合）と文書のキーワードとの照合によって，照合時間が文書の全文と検索語を照合するよりも短縮できるからである．また，キーワードが文書の内容の本質を的確に表していればキーワード抽出は文書にとって意味のない部分を捨てることになるので，全文検索よりかえってユーザの興味を深く満足させる検索結果を得ることができる．

我々は本論文で，文書の主張の内容を表すキーワードの抽出をねらう．そのようなキーワードはユーザの検索語に近い主張をもつ文書を検索するのに役立つので，ユーザは，自分と同じ考えの著者や，内容は見聞しているが出典が不明な文書を探すことができるようになる．例えばユーザがアルゴリズムの研究者で「NP 完全問題を高速に解くために，問題を緩和した容易な問題を解いた近似解を初期点として局所探索を行う」ことを思いついたとする．この考えに近い従来研究を探し出せばこのユーザは自分の位置付けを理解できるし，類似の主張を行った論文がなければ自分の考えの新しさを認めることができる．

注意が必要なのは，このような検索は興味に関連する既存分野の文献を選ぶこととは違うということである．例えば，高速アルゴリズムに関する文献を集めるためならユーザは『アルゴリズム』や『計算時間』を指定すれば良い．しかし，先述のアルゴリズム研究者が『NP 問題，近似解，局所探索』を検索語として入力したとき，選ばれるべき文書のキーワードが高速化アルゴリズム研究の分野で頻出する『アルゴリズム』や『計算時間』という語になっていたらユーザの検索語と一致せず，選ばれなくなってしまう．

ここでの『アルゴリズム』や『計算時間』はその分野で既に普及した言葉であり，著者独特の主張は表していないので，主張がユーザの検索語に近い文書を選び出すという目的には向かない．そして，従来のキーワード抽出技術は **2.** に示すように，このような語をキーワードとしてしまうことが多いのである．

[†] 大阪大学大学院基礎工学研究科，豊中市
Graduate School of Engineering Science, Osaka University, 1–3 Machikaneyama, Toyonaka-shi, 560–8531 Japan

2. 従来のキーワード抽出法

2.1 なぜキーワードを自動抽出するか

日常我々が用いているキーワードには，

（1） 文書に関連する既存分野名
（2） キーワードを示せと指示され著者が付けたもの
（3） 各文書に専門家が付けたキーワード
（4） テキストから自動的に抽出したキーワード

など数種類のものがある．

（1）は，前節で示した本論文の目的から外れている上に，付けられた分野名は信頼性に欠ける．というのは，該当分野を示そうとしても，高い新規性を有する学術論文には既存の分野に収まりにくいものもあるからである．実際，学会に行くと知人の論文が意外なセッションに振り分けられたという話はよく耳にする．（2）のように著者がキーワードを付す場合は，該当する分野を示せばよいのか，それとも自分の主張点を表す語を示すのか迷う場合が多い．すると検索者から見ても，自分の興味を両者のうちどちらのキーワードで表せばよいのかわからないことになる．更に，文書量が膨大になると専門家が（3）のようにキーワードを付すのは不可能となる．したがって,（4）のキーワードのテキストからの自動抽出は，**1.** で述べた本論文の目的を達成する上で不可欠な技術となるのである．

キーワードの自動抽出にも種々のアプローチがあるが，それぞれ本論文の目的にとっては問題点がある．次に，これらの問題点について述べる．

2.2 従来のキーワード自動抽出法

［文書の見出し情報を用いる方法］

タイトルや見出しは，文章のポイントを簡潔に要約したものとなっていることが多い．そこで，タイトルや見出しの中で前置詞や冠詞以外の語をキーワードとする方法が考えられる．しかし，インターネットのホームページや電子メールなどまで考えると，タイトルが付されていないことが多い上に “Welcome to My Homepage !”や “Links”など本文の内容を具体的に表さないタイトルも多い．

［文書の冒頭からのキーワード抽出］

新聞記事では記事の冒頭部に重要な語が多い [1] が，新聞記者はそのように文章を作るように訓練されているのでこの傾向を一般の文書にあてはめることはできない．実際，文書の種類によって主要部分の位置はさまざまで [2]，重要な意味分類が少数の大段落に集中する [3] としてもそれがどの段落であるかは筆者の章立ての癖などに依存する．

［自然言語解析を用いるキーワード抽出］

自然言語解析によって文書中のどの語が重要であるか理解できるなら，正確なキーワード抽出が実現できるかもしれない．しかし，文法ルールに正しく従うとは限らない文章から自然言語解析によって要点を的確に取り出すのは現時点では困難である．また，重要な語は太字で書かれているとか，重要な語の前に “It is important that . . .”などの前置きがあるという期待も一般にはできない．

［統計量に基づくキーワード抽出］

古くからあるキーワード抽出法に，対象文書中で頻出する語をキーワードとする手法がある [4]．しかし，頻出語が文書の独自な主張を表現する語となることは実際には少ない．例えば，**1.** で例とした高速アルゴリズムに関する論文で「アルゴリズム」や「計算時間」の出現回数が多くても，それらは文書の著者独自の主張を表すのではなく，読者を高速アルゴリズムの分野に誘い込み理解の土台を確保する役割を果たすものである（したがって文書の分類には役立つ [5]）．

一方，対象とする文書 D_i の属する分野のコーパス（例文データベース）を用いて，ある語 T_j の D_i での頻度 tf_{ij} のその分野での平均出現頻度 df_j に対する相対比率（後出の TFIDF など）を語 T_j の重要度とするアプローチ [6], [7] もある．この手法では，単に D_i での出現回数の多い語よりは重要な語を抽出できる．あるいは，文書を分野に正しく分類する効果の高い語を「分野」対「語」の相対情報量に基づいて学習しておき，これらのうち文書 D_i に含まれるものをキーワードとする手法 [8] も文書の分類に役立つ．

しかし，ある特定の分野のコーパスはその分野が不明では作ることができない（逆に，分野を限定しないコーパスは，集めるべき用例が多過ぎて現実には構築できない）．一方，我々の目的はある主張をもつ文書を集めることである．例えば，先の例における高速アルゴリズム研究者がユーザとなる場合，数理アルゴリズムの分野ではなく人工知能の学会で発表された高速推論の文献が自分の考えに先行していると知る場合など，学際領域の文献が有用となる例は少なくない．このような場合，分野を限定したコーパスを用いた手法は通用しなくなる．

3. KeyGraph：共起グラフを用いたキーワード抽出法

2. の背景から本論文では，新しい手法でキーワードを抽出することを提案する．すなわち，文書の見出し情報や自然言語解析を用いず，単純な頻度だけで重要度を比較しないが，著者の主張を表す語を抜き出すことのできるキーワードの自動抽出を目指す．大量の文書のキーワード抽出を行うためには高速なアルゴリズムであることも要請されるが，本論文ではまず著者の主張を表す語を抜き出すことだけに照準を絞ってアルゴリズム KeyGraph の提案を行い，後出の **5.** において計算時間の評価を行うことにする．

3.1 KeyGraph の基本的な考え方

KeyGraph は，文書は著者独自の考えを主張するために書かれるという仮説をもとにしている．文書全体はその主張を目指して一つの流れを形成するというわけで，文書を建物にたとえると我々の仮説は

> 建物が立つには，土台（文書がもとにしている基本概念）が必要である．壁（文章の構成に必要な説明部分），ドアや窓（詳細な記述），さまざまな装飾（比喩や例など，付加的な記述）もある．しかし，建物の本質は日射や雨から住人を守る屋根（主張点）であって，屋根を支えるために柱（内容の主な展開）がある．

ということになる．例えば学術論文には，冒頭に要点が密集している新聞記事とは異なり，文章が論理的な鎖状に構成されているものも多い．その中には数式やその説明，例証などのまとまりもあるが，その中で繰り返される頻出語（**1.** の例の『アルゴリズム』や本論文での『キーワード』）は要点とは別の，いわばその文章が書かれる上で当然のように前提とされる「土台」の概念を表すことが多い．これらの土台の上に立つ「柱」に支えられて文書全体を束ね方向づけるのが主張（「屋根」）である．この土台・屋根・柱を頼りにキーワードとして取り出すのが KeyGraph の基本戦略である．KeyGraph のアルゴリズムは，次の 3 フェーズからなる．

1）土台の形成：　文書形成の準備あるいは前提となる基本概念（具体的には，後述の語の共起グラフにおいて強く連結し合う語の集まり）を土台とする．

2）屋根の形成：　1）で取り出した土台たちに強い力で支えられて文章を統合する語を屋根とする．

3）キーワードの抽出：　土台と屋根を結ぶ強い柱が多く集まった語をキーワードとする．

KeyGraph では，まず土台または屋根の候補となり得る文字列を準備フェーズで取り出しておき，その後上記の三つのフェーズが実行される．以下の小節では，これらの各フェーズについて詳細を述べる．

3.2 KeyGraph の準備フェーズ

まず，キーワードの候補としてふさわしくない単語を対象の文書 D から削除する．この，削除する単語の集合（*Noise* と呼ぶ）だけが KeyGraph で用いる唯一の文書外の知識である．*Noise* は “a”，“and”，“here” などの単語と TeX の書式コマンドや HTML のタグ，削除すべき語尾からなる（“run”，“running”，“runs” はともに語尾が削除されて “run” となる [9]）．

次に，D のうち *Noise* 以外の単語を熟語の要素の候補とする．熟語を取り出す方法としてここでは，連続する 2〜3 単語の組合せのうち D での出現回数と長さが極大のものを取り出す．例えば，$\{abcd\}$ なる 4 単語の並びからはまず $\{abc\}$，$\{ab\}$，$\{bcd\}$，$\{bc\}$，$\{cd\}$ なる熟語の候補を生成する．そしてこれらの候補を出現回数でソートし，各候補 p について p を含む候補 q の出現回数が p 以上ならば p を，p 以下ならば q を候補から捨てる（ここで，$\{bcd\}$ は $\{cd\}$ を含むという）．こうして残った候補を熟語とする．

なお，本論文では単語の切れ目がわかる文書を対象とし，名詞以外もキーワードの候補とする．後者の理由は，文章の著者が主張を展開する場合，動詞や形容詞，副詞に力点を置くこともあるからである（「ダイナミックな」環境に「適応して行動する」ロボットを主張する論文など）．

以下では，この準備フェーズでキーワードの候補になった語の集合を D_{terms} と呼ぶ．D_{terms} は非冗長，すなわち $D_{terms} = \{w_1, w_2, \ldots, w_l\}$ とすると $i \neq j$ なる任意の整数 i, j $(1 \leq i, j \leq l)$ について $w_i \neq w_j$ とする．以後，単に語というときは D_{terms} 中の語をさす．

3.3 土台の形成

このフェーズから，文書 D を表す共起グラフ G を生成する段階に入る．このフェーズでは G を以下のノードと枝から生成する．

- ノード：　D_{terms} 中の語は，D における出現回数によってソートされる．このソートでの上位 M 語からなる集合を *HighFreq* と呼び，はじめに G 中のノード群として与えておく（図 1 中の黒いノード）．実際には経験的に（後述の **4.** での実験で性能が高かっ

た(注1)) $M = min(30, |D_{terms}|)$ とし（詳細は **4.1.2** 参照），30 くらいの語が複数あればそれらも *HighFreq* に含めた．

HighFreq 中の語は土台を形成する要素として用いられる．というのは，土台すなわち基礎概念を表す語は D 中で何度も用いられることが多いからである．

• 枝（リンク）： *HighFreq* 中で共起度の高い語の対をそれぞれ枝で結ぶ．ここで，語の対 (w_i, w_j) の D における共起度 $co(w_i, w_j)$ を次のように定義する．

$$co(w_i, w_j) = \sum_{s \in D} |w_i|_s \, |w_j|_s \, . \qquad (1)$$

ここで $|x|_s$ は文 s における要素 x の出現回数で，x が語の場合に $|x|_s$ は文 s 中の語 x の出現回数である．式 (1) は，ある文 s に出現した語 w_i は s 中のすべての w_j と共起しているとみなした共起度を表す．

2 語間の共起度の定義にはさまざま考えられるが，上記の co で定義したのは以下の理由による．

まず，文単位で共起度を測定した理由を述べる．今，共起度を，文章中で連続する長さ W の範囲中で 2 語両方が出現する頻度とする．W が 1 文より短い場合には，倒置や疑問文において語順が変わると，著者にとって関連の深い語同士が遠く離れ W 内で共起しなくなることがある．逆に W を 1 文より長くすると，2 語のうち片方の内容が指示語の形で複数文にまたがって現れても，その内容ともう片方の語との共起をとらえることができ精度が上がることもある．しかし KeyGraph のグラフ構築では，この原因で精度が向上するよりも多くのケースにおいて，実際には意味のつながりの弱い語の対にまでリンクが張られてしまい，この後で得られるキーワードの精度を劣化させる原因となる．

更に，co 以外にも，語の各対の共起の重要さを考慮した共起度の定義が従来からある．例えば，2 語間の相互情報量 [10] によって，2 語が独立に出現する場合に比べて実際にはどれだけ近くに現れやすいかを表すことができる．しかし，そのためには出現し得る語の総数の 2 乗に比例する記憶容量が必要なので，分野を限定できないほどユニークな主張の中に出現し得るすべての語の対の重要さを蓄えるには膨大な記憶容量が必要となる．更に，我々のいう「土台」は通常一般的な概念であるから，「独立に出現する場合に比べて」（前述）ではなくて，近くに現れる回数自体が多い語の対を選ぶ式 (1) の co の方が適切ともいえる．

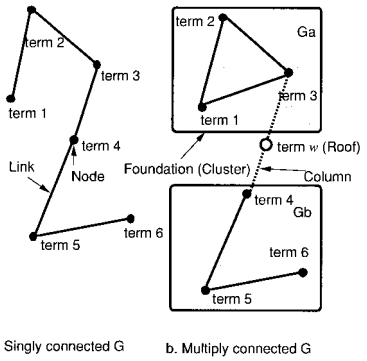

図 1 土台（G_a と G_b）と屋根（キーワード w），柱（点線）
Fig. 1 Foundations (G_a and G_b), roof (keyword w) and columns (dotted lines).

HighFreq 中の語の各対は co の大きさによってソートされ，上位 $M-1$ 番目までの語対に G 中で枝を張る（M は先述）．なぜなら，これが G 中のノードを冗長な（D の内容の展開を表す上で不要な）リンクなしに結び合わせるために必要最小限の枝数だからである．すなわち，G が連結グラフであれば，$M-1$ 本の枝だけ G 中に存在することは G が単結合であることと同値であり，それは D が冗長さなしに内容が展開する文章としてグラフ化される（例えば図 1-a のように語（term）3 が語 1 からただ一つのパスでたどれるという図になる）ことにあたる．

もし図 1-b のように複数のパスを通って語（term）1 から語 3 がたどられるならば，語 3 と語 4 の関係の強さが土台 G_a の内部の語同士の関係よりも弱いとみなせる．そこで，上記のノードと co の上位 $M-1$ 本の枝からなるグラフ G のうち，自分以外の連結部分グラフ（自分が含むどの 2 ノード間にもパスをもつ G の部分グラフ）に含まれない連結部分グラフ，すなわち極大連結部分グラフを一つの土台とする．

この土台の抽出法は，一つの文は一貫した概念を表現し，その概念をめぐって文中の語が関係し合うという仮説（語彙的連鎖 [11]）に立っている．語彙的連鎖は文書を意味の一貫した部分（我々の土台のように）に分割する目的でも用いられている [12] が，どの部分が文書中で重要であるかは残された課題であった．KeyGraph ではこの語彙的連鎖を用いて文書の土台をつかみ，この後文書全体の流れにとって重要な主張を

(注 1)：以下，「経験的に」というときはこの意味である．

表すキーワードを以下のように取り出すのである．

3.4 屋根の形成

土台，すなわち文書の基礎となる概念は，文書全体から見ると著者が主張しようとすることを導き出すために関連しあっている．我々の取り出したいキーワードは文書の主張を表す語であるから，先述の屋根として土台たちに強い力で支えられて文章を統合する語（図 1-b の term w）でなくてはならない．

KeyGraph では語 w が土台たちに支えられる力を $key(w)$ で表す．$key(w)$ は 0 から 1 までの実数で，「G 中のすべての土台を著者が考慮しているときに w が用いられる」という条件付確率を定式化したものである．まず，$key(w)$ を構成する次の 2 関数を定義する．

$$f(w,g) = \sum_{s \in D} |w|_s \, |g-w|_s \, . \tag{2}$$

$$F(g) = \sum_{s \in D} \sum_{w \in s} |g-w|_s \, . \tag{3}$$

ここで s と w をそれぞれ文と語を指す添字，$|g|_s$ を土台 g に含まれる語の s 中の出現回数として

$$|g-w|_s = \begin{cases} |g|_s - |w|_s & \text{if } w \in g, \\ |g|_s & \text{if } w \notin g \end{cases} \tag{4}$$

とする．すなわち，$f(w,g)$ は語 w と土台 g 中の語の共起度である．式 (4) で w が g に含まれるときに w の出現回数を g の出現回数から差し引くのは，w と，g 中の w 以外の語との共起度を調べるためである．

$key(w)$ は次のように定義する．

$$key(w) = \left[1 - \prod_{g}^{bases} \left(1 - \frac{f(w,g)}{F(g)} \right) \right] . \tag{5}$$

ここで $bases$ は土台の個数である．$f(w,g)/F(g)$ は g 中の語と共起する語が w である確率であり，我々はこれを著者が土台 g の表す概念を考慮しているときに語 w を書く確率として用いている．すなわち，式 (5) の $key(w)$ は

$$probability \left(w | \bigcap_{g \subset G} g \right) , \tag{6}$$

を表そうとしているのであり，式 (6) が論理的に

$$probability \left(\bigcup_{g \subset G} (w|g) \right) , \tag{7}$$

と等価になることから式 (5) を得たのである．なお，式 (7) を式 (5) のように表現する段階で，それぞれの土台を互いに独立とする仮定をおいた．これは，土台すなわち基礎となる概念はそれ以上基礎的な他の概念から導かれるものではなく，互いに独立であるという仮定にあたる．

文書 D 中の key の値の上位（現在は経験的に $min(12, |D_{terms}|)$ 語とし 12 位の語が複数あればそれも含める）の語の集合（「屋根」にあたる）を $HighKey$ と呼ぶ．$HighKey$ 中の語が G にまだ含まれていなければ，新しいノードとして G に加える．

3.5 キーワードの抽出

KeyGraph で取り出すキーワード群は，$HighKey$ そのものではない．というのは，土台中の語でも，屋根に強く結び付いている語は屋根を表現し文書 D を要約する上で重要だからである．これらの語は key の値は小さいかもしれないが，$HighKey$ 中の語と同様に接する「柱」（土台と屋根を結ぶリンク）の強さの和では上位にランクされるものと考えられる．そこで，$HighKey$ 中の語との間の柱の強さ（下記）の和の大きな語を最終的にキーワードとする．

$HighKey$ 中の語 w_i と，いずれかの土台に含まれる語 w_j を結ぶ柱の強さ $c(w_i, w_j)$ は，次の式で表す．

$$c(w_i, w_j) = \sum_{s \in D} |w_i|_s \, |w_j|_s \, . \tag{8}$$

そして，G 中のノードで，接する柱すべての c の和が上位の語をキーワードとする．キーワードの個数は経験的に上位 $min(12, |D_{terms}|)$ 語とし，12 位の語が複数あればそれらも含めている．なお，図 1〜図 3 では c の値が上位の柱を点線として加えたグラフ G を示した．

3.6 「誤リンク」削除による精度向上

上述のように，土台の取り出しでは，テキスト中で意味的に一貫したまとまりを共起グラフ中の極大な連結グラフとして取り出す．しかし，柱として後で取り出されるべきリンクが土台中の枝として先に取り出されてしまう（これを誤リンクと呼ぶ）と，キーワード抽出精度が劣化する原因となる．

例えば，図 2 はある文書 D（文献 [13] の LaTeX ソース）に対する土台（実線と黒いノード），柱（点線），キーワード（2 重丸のノード）の抽出結果の一例である．文書 D は，述語論理仮説推論を整数計画問題に帰着し高速な近似解法によって高速化する手法を提

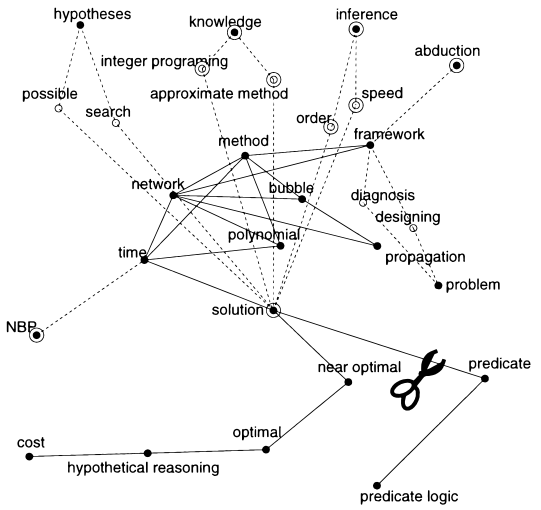

図 2　誤リンク（はさみの印）のせいで KeyGraph がキーワード predicate logic の抽出に失敗する例
Fig. 2　A bad link in the scissors, leading to missing "predicate logic."

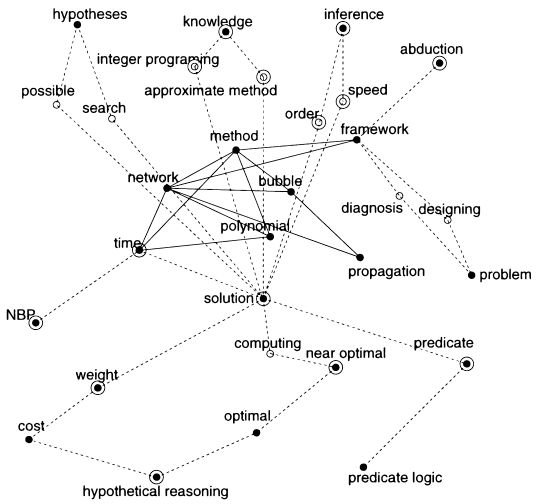

図 3　グラフ分割の仕上げによって改善された結果
Fig. 3　Improved result by further segmentation.

案するものであった．したがって，"speed"，"integer programming" と "approximate method" が低い出現頻度（6899 単語中 2 回以下で，出現頻度の順位は 100 番以下）にかかわらずキーワードに選ばれている点は，KeyGraph の優れた性能を示している．

しかし，"predicate"（『述語』）も "predicate logic"（『述語論理』）も選ばれていない．これは，D が筆者のそれまでの成果を命題論理から述語論理に拡張した論文であることを考えるとまだ精度が不十分であることを示している．この誤りは，図 2 ではさみ印の付された誤リンクが土台中の枝として取り出され，"predicate" が "solution" を含む土台から **3.3** において分離できなかったことによる．すなわち，後で柱を得る際に（**3.4** 参照）はさみ印のリンクを柱とすることができず，"predicate" をキーワードに選べなかったのである．

その対策として，土台から誤リンクを取り去る操作を KeyGraph に追加した．すなわち，**3.3** の操作の直後に次の操作を挿入する．

［グラフ分割の仕上げ操作］

G 中の枝 e をとり，$G-e$ 中に e の両端ノードを結ぶ e 以外のパスが存在しなければ，e を G から取り除く．これを G 中のすべての枝を e として行う．

簡単にいうと，グラフ G から枝 e を取り去ると G が二つの連結部分グラフに分離されてしまうというほど弱く連結された連結部分グラフ間の枝 e を，すべて実際に G から取り除いてから土台を抽出するのである．

図 2 と同じ文書 D についてこの仕上げ操作を加えたところ，図 3 の結果を得た．先の "predicate" に加え，重要な土台として "time"（『計算時間』）も抽出されているのは精度の向上を示している．

この操作に要する時間は，G 中の枝数を L とおくと $O(L^2)$ である．というのは，上の操作では L 本の枝が交互に「仮に」除去され，その各枝 $e(w_i, w_j)$ について $G-e$ 中で (w_i, w_j) 間のパスの有無を調べるからである．ここで同時に「仮に」除く枝の本数 d を多くすると精度が更に改善される可能性もあるが，グラフ分割の仕上げ時間が $O(L^{d+1})$ となってしまうので，ここでは操作を高速に行うために $d=1$ とした．

4. KeyGraph の性能の実験評価

KeyGraph を C 言語で記述し，64 M の RAM を備えた Pentium Pro 200 MHz のマシン上に実装して以下の実験を行った．

4.1 文書の著者による評価

4.1.1 出力キーワードの評価

KeyGraph によって得られるキーワードが文章の主張と一致するかどうかを評価するため，さまざまな学術論文について KeyGraph の出力結果についてのインタビューを各論文の著者に対して行った．被験者（著

表 1　抽出されたキーワードの，文書の著者による評価
Table 1　Performance evaluated by document authors.

	TFIDF	KeyGraph
suff	65/88	76/88
necc	159/239	274/310

者）からの回答は，以下の指標で評価した．

1）抽出されたキーワードの十分さ

$$suff = |A \cap K|/|A| \tag{9}$$

2）抽出されたキーワードの必要度

$$necc = |A \cap K|/|K| \tag{10}$$

ここで，A，K はそれぞれ以下の集合である．

［A：著者の主張を表すキーワード集合］

著者が KeyGraph の結果を見る前に著者自身の主張を表すキーワードを挙げてもらい，これらをキーワード群 A とした．これは，**2.1** 冒頭の（2）で著者にただキーワードを示すように指示するのとは，こちらの要求する「キーワード」とは何かを明示している点で異なる．

［K：KeyGraph によって得られたキーワード集合］

キーワード群 A を挙げる著者のバイアスとならないように，上のキーワード群 A が挙げられた後で K を得るようにした．

$|A \cap K|$ は，式 (9) と式 (10) で求め方を変えた．式 (9) の $|A \cap K|$ は，KeyGraph で K を得た時点で A のうちいくつ K に含まれたかを数えれば求められる．一方，真のキーワード群 A のうち著者の挙げなかったキーワードが K の中には含まれることがあるので，式 (10) ではキーワード群 K のうち著者が「私の主張を表す」と認めた語集合を $|A \cap K|$ とした．

実際には多数の著者の回答から以上の情報を得るのは難しく，23 人の著者からしかデータを得ることができなかった（対象文書はコンピュータサイエンス及び医学に関する論文又はその概要の計 23 件で，長さは 200 単語台から 10000 単語台までほぼ一様に選んだ）．しかし，得られた結果はコーパス中での語の頻度 df_j を用いた従来手法（**2.2** 参照）より良好であった．結果を表 1 に示す．表中の整数は語数である．ここで KeyGraph と比較した手法は，従来の TFIDF [6] で文書中の語をソートし **3.5** 同様に個数を決めた上位の語をキーワードとする方法である．この TFIDF では，語の一般的な頻度 df_j を求めるのに用いるコーパスとして，医学・コンピュータサイエンス・その他の分野のそれぞれについてインターネット上で Yahoo!（http://www.yahoo.com）から収集した英文書 5680 件をとった．また，熟語・語尾処理は KeyGraph 同様（**3.2** 参照）とした．

表 1 に見られる KeyGraph の性能の高さが土台と屋根がうまく取り出せていることで裏づけられれば，KeyGraph の効果はねらいどおりであったことになる．しかし，文書の著者がこの実験で「主張を表す」と認める単語が屋根として直接主張を表すか，あるいは屋根の主張のために用いた土台なのか，どちらか片方に判別するのは著者自身でさえ実際には難しく，土台と屋根の構成精度を別に測定するのは困難であった．そこで，グラフ G が文章の意味構造を反映しているかどうかを調べる別の方法として，土台と屋根の間を結ぶ「柱」が著者の主張にとって重要な単語を絞り込むのに役立っているかどうかを土台中の語について調べた．

4.1.2　土台キーワードの評価

著者が重要な土台とみなした（「主張の背景となる重要な基礎概念を表す」と著者が認めた）語は，KeyGraph の抽出キーワードのうち *HighFreq* に含まれた 230 語のうち 170 語（73%），TFIDF によるキーワード 239 語のうち 61 語（26%）であった．したがって，KeyGraph が土台を得る精度は TFIDF より高いといえる．更に，すべての *HighFreq* 中の語についての著者のコメントでは，788 語のうち 296 語（38%）だけが重要な土台とみなされた．このことは，文書の内容の流れ（柱）と各語との結び付きを考慮することによって，屋根だけでなく土台の中でも重要な語が絞り込まれることを示している．

なお，一般に *HighFreq* の大きさ M を大きくすると *HighFreq* の中には単一で一つの土台となる語が多くできてしまい，それらと文章中の語の間の柱も増える結果，文章にとって重要でないキーワードが得られてしまうことが多くなる．M を変化させて本節と同様の実験を行うと，$M = 12$ 以下では主張に深くかかわるキーワードが得られにくく（$suff < 50\%$），$M = 15$〜40 では著者にとって妥当なキーワードが本節の実験に用いた KeyGraph と同程度の精度で得られた．しかし，$M = 50$ 以上になると著者が不要とする語がキーワードに目立って含まれ，$necc < 50\%$ と精度が低下した．**3.3** で与えた $M = 30$ は，キーワードの精度が安定で，精度の高い範囲のほぼ中央の値である．

4.2　サーチエンジンによる評価

次に，サーチエンジンで用いるキーワード（**1.** 参

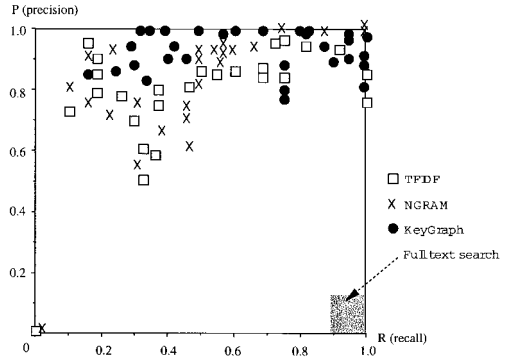

図 4　他手法との適合率（P）と（R）による比較
Fig. 4　Performance evaluated by a P–R curve.

照）を KeyGraph で抽出した場合の再現率（R：得るべき文書のうち得られた文書の比率）と適合率（P：得られた文書のうち得るべき文書の比率）の評価を行った．**4.1** で TFIDF のコーパスとした 5680 件の文書から，自分の読みたい主張の内容をキーワードで指定してユーザに文書検索を行ってもらい，得られた文書が自分の希望に合うかどうか評価してもらった．

実験結果を図 4 に示す．比較した手法は **4.1** と同様の TFIDF(注2)，NGRAM [7] による抽出文字列を TFIDF で順位づけし **3.5** と同様に決めた個数の上位をキーワードとしたもの，そして KeyGraph である．

実験手法は，まず検索対象の各文書をこれら各手法によりキーワード集合に変換し，キーワード集合がユーザの検索語をすべて含む文書を出力するものとした．そして，3 手法によって得られた文書すべてをシャフルし（混ぜ合わせて順序をランダムに並べ替え）たリストを表示し，表示された文書をユーザに評価してもらうと R，P が自動的に計算されるようにした．

図 4 は各ドットで 1 回の検索についての (R, P) を表し，48 回の検索結果を表示したものである（ドットの重なりもある）．KeyGraph のデータはグラフの右上に集中した上に R の低い範囲での P の値も高く，KeyGraph の高い性能がわかる(注3)．

文書からキーワード集合への変換を用いる上記の検索手法に対し，近年普及している全文検索では，文書中にユーザの検索語が含まれればその文章を出力するので多くの文書が得られ再現率が高い．しかし，ユーザの希望に近い主張の文書を探すという厳しい条件での検索では，全文検索の適合率は極めて低くなる．実際，P–R グラフでは，全文検索（検索語をすべて含む文書をすべて表示させた場合）の性能は図 4 の右下端の灰色の範囲に集中した(注4)．

4.3 考　　察

本章の実験結果を総合すると次のようになる．まず **4.1.1** で，著者の主張を表すキーワードが KeyGraph によって高い精度で得られた．その理由として，KeyGraph では土台と屋根の間の柱を用いて文書全体の流れにとっての重要な語を絞り込んでいることが **4.1.2** で見出された．語同士の共起に着目したキーワード抽出法は従来もあった [14] が，ローカルな共起関係だけではなく文書全体の流れに着目したのが KeyGraph の特徴といえる．

そして，このように文書の主張をキーワードで表現することによって，検索ユーザにとっては入力した検索語に近い主張をもつ文書が得られることになるというのが **4.2** で KeyGraph の性能が高かった理由であると我々は考えている．

以上から，KeyGraph は著者の主張を表すキーワードを抽出する能力が高く，特定の主張をもつ文書を探す検索に適しているといえよう．実際，上述の結果以外に，箇条書き形式で書かれたリンク集などよりも，ひとまとまりの文章となっている文書の方が主張をもつので KeyGraph の性能が高いという傾向も得られている（「ひとまとまりの文章らしさ」などの数値化が困難なため，この性質の評価方法は今後の検討課題である）．

5. KeyGraph の処理時間

KeyGraph の処理時間の上限を評価する．

T_{dp}（準備フェーズ）：　D 中の語数を W とすると，熟語を頻度でソートする時間が $T_{dp} = O(W \cdot \log(W))$ となる（熟語の個数が W に比例するから）．

T_{ef}（土台の形成）：　T_{FF} と T_{links} をそれぞれ *HighFreq* と G 中のリンクを得る時間とする

(注2)：この TFIDF ではコーパスも **4.1** と同じもの，すなわちここでの検索対象文書の集合全体とした．これは，与えた文書集合のうちで特にユーザの希望に合うものを選べたかどうかを調査するためである．
(注3)：サーチ結果の文書を重要さの順で並べ，上位から順に出力文書集合を増やしながら P と R を評価するという従来の評価方法ではサーチの性能曲線（P–R 曲線）が右下がりとなるが，ここではそうなっていない．これは，各検索語に対する出力文書集合全体について P と R を計算したからである．従来の評価手法をとらなかった理由は，ユーザの希望に近い主張の文書を探すという検索では全文書中でも適合するものが少ない（42%の検索語に対して 1 件以下であった）ため，P–R 曲線のさまざまな R での P の平均値が同じ検索式集合については計算できないからである．
(注4)：ドットの重なりが多いため，全ドットを含む範囲のみ図示した．

と，$T_{ef} = T_{FF} + T_{links}$ となる．T_{FF} は D_{terms} 中の語のソートで $O(|D_{terms}| \cdot \log|D_{terms}|)$ のオーダ，T_{links} は D_{terms} 中の語のペアのソートに $O(|HighFreq|^2 \cdot \log|HighFreq|)$ と **3.6** のグラフ分割の仕上げに $O(G$ 中の枝数$^2)$ の時間がかかる．G の大きさは $|HighFreq|$ すなわち定数オーダであるから，結局 T_{ef} は $O(|D_{terms}| \cdot \log|D_{terms}|)$ のオーダである．

T_{ec}（屋根・キーワードの抽出）：T_{ec} は $O(|D_{terms}| \cdot bases)$ 個の柱をソートする時間であるから，$bases$ を定数とすると（多くとも $|HighFreq|$ 以下だから）T_{ec} は T_{FF} 以下の時間となる．

以上から，$T_{dp} + T_{ef} + T_{ec}$ は $O(W \cdot \log(W))$ 以内のオーダとなり，TFIDF と同程度となる．実際，計算時間は W に対して線形よりわずかに速く増大した（$W = 7000$〜8000 では平均 1 s 以内であり，$W = 625$ では 0.1 s であった）．

6. むすび

KeyGraph のように対象文書の関連分野ではなく，著者が新しく主張しようとする内容を表現する語を得ることは，文書の全文を読む前に著者の考えを反映した検索を行うための技術として重要である．インターネットの普及によって今後さまざまな主張が自由に公開されることになろうが，その中で文書の内容を重視した文献検索の役割はこれからも大きくなっていくと我々は考えている．

謝辞　本研究は稲盛財団研究助成を受けて行われました．研究へのコメントをいただきました大阪大学産業科学研究所の北橋忠宏教授，学術情報センターの安達淳教授，東京工業大学の山田誠二助教授に記して感謝します．また，関連研究の資料を提供してくださいました富士通研究所の仲尾由雄氏に感謝します．最後になりますが，査読者のコメントが論文の内容を改善する上で役立ちましたので，記して感謝します．

文　献

[1] 野本忠司，松本裕治，“テキスト構造を利用した主題の推定について，” 情処学研報，NL114, pp.47–54, 1996.

[2] 木本晴夫，“日本語新聞記事からのキーワード自動抽出と重要度評価，” 信学論（D-I），vol.J74-D-I, no.8, pp.556–566, Aug. 1991.

[3] 鈴木　斎，増山　繁，内藤昭三，“語の意味分類の出現傾向を考慮したキーワード抽出の試み，” 情処学研報，NL98-10, pp.73–80, 1993.

[4] H.P. Luhn, “A statistical approach to the mechanized encoding and searching of literary information,” IBM J. Research and Development, vol.1, no.4, pp.309–317, 1957.

[5] 西野文人，“日本語テキスト分類における特徴素抽出，” 情処学研報，NL112, pp.95–102, 1996.

[6] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing and Management, vol.14, no.5, pp.513–523, 1988.

[7] J. Cohen, “Highlights: Language- and domain-independent automatic indexing terms for abstracting,” J. American Society for Information Science, 46, pp.162–174, 1995.

[8] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell, “WebWatcher: A Learning Apprentice for the World Wide Web,” AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments, March 1995.

[9] M.F. Porter, “An algorith for suffix stripping,” Automated Library and Information Systems, vol.14, no.3, pp.130–137, 1980.

[10] K.W. Church and P. Hanks, “Word association norms, mutual information, and lexicography,” Computational Lingustics, vol.6, no.1, 1997.

[11] J. Morris and G. Hirst, “Lexical cohesion computed by thesaural relations as an indicator of the structure of text,” Computational Linguistics, vol.17, no.1, pp.21–48, 1991.

[12] M.A. Hearst, “Multi-paragraph segmentation of expository text,” Proc. Annual Meeting of the Association for Computational Linguistics, pp.9–16, 1994.

[13] Y. Ohsawa and M. Ishizuka, “A polynomial-time predicate-logic hypothetical reasoning by networked bubble propagation method,” Advances in Artificial Intelligence LNAI-1081, G. McCalla, ed., pp.375–387, Springer Verlag, 1996.

[14] 原　正己，中島浩之，木谷　強，“単語共起と語の部分一致を利用したキーワード抽出法の検討，” 情処学研報告，NL106, pp.1–6, 1995.

（平成 10 年 3 月 16 日受付，7 月 30 日再受付）

大澤　幸生

1990 東大・工・電子卒．1995 同大大学院博士課程了．博士（工学）．現在，大阪大学基礎工学部助手．1994 人工知能学会全国大会優秀論文賞．人工知能学会，情報処理学会各会員．

ネルス E.ベンソン

1996 ワシントン大学コンピュータサイエンス学科卒．Oren Etzioni に師事した後日本に留学．1997 より大阪大学大学院基礎工学研究科修士課程に在籍中．

谷内田正彦　（正員）

1971 阪大大学院工学研究科修士課程了．同年同大基礎工学部制御工学科助手．同助教授を経て同学部情報工学科教授，1994 同学部システム工学科教授．1967〜68 デンマーク原子力研究所留学．1972〜73 米イリノイ大学にて Research Associate．1980〜81 西独ハンブルグ大学 Research Fellow．1982 米ミネソタ大学 CDC Professor．情報処理学会，ロボット学会，人工知能学会等会員．著書ロボットビジョン（昭晃堂），コンピュータビジョン（丸善，編著）等．工博．