Vol. 48 No. SIG 11(TOD 34) 情報処理学会論文誌：データベース June 2007

Web事典からのシソーラス辞書構築手法

中山 浩太郎† 原 隆浩† 西尾 章治郎†

近年，Wikipediaに代表されるような，記事どうしがハイパーリンク（以降リンク）で結び付けられたWebベースの事典が数多く公開されてきた．筆者らはこれまでの研究でWeb事典に対してWebマイニング手法を適用することで精度の良いシソーラス辞書を構築できることを示してきた．しかし，膨大な記事数を持つWeb事典を解析するためには，効率的かつ精度の高いシソーラス辞書の構築手法が必要とされている．そこで，本研究ではnホップ先までのリンク構造を効率的に解析し，語どうしの関連度を算出する手法*lfibf*および3つの応用手法「単純法」「対数近似法」「Forward/Backwardリンク重みづけ手法」を提案し，実験によりその有効性を示す．

A Thesaurus Construction Method from Web Dictionaries

Kotaro Nakayama,† Takahiro Hara† and Shojiro Nishio†

Web based encyclopedias, such as Wikipedia, have become dramatically popular among internet users. We have already proved how effective they are to construct a Web thesaurus. However, we still need efficient methods to analyze the huge amount of Web pages and Web links among articles in encyclopedias. In this paper, we propose "lfibf," an efficient method to construct a Web thesaurus from Web encyclopedias like Wikipedia by using three sub approaches: the "Simple method," the "Log method" and the "Forward/Backward link weight optimization method."

1. はじめに

近年，WWWの爆発的な普及にともない，Wikipediaに代表される多数のWeb事典が公開されてきた．Wikipediaは，Wikiを利用して構築された百科事典であり，文化，歴史，数学，科学，社会，テクノロジなどの幅広い分野の語（記事）をカバーしている（図1）．Wikipediaでは，Webブラウザを通じて，他のユーザと議論しながら自由に記事を投稿できることが大きな特徴である．Wikipediaには，2006年9月の段階で約137万もの膨大な数の記事（英語のみ）が公開されており，市販の百科事典の記事数が数万～10万であることと比較してもその規模が膨大であることが分かる．Nature誌の調査によると，Wikipediaの記事数および精度は，多くの専門家が集まって作成した百科事典「Britannica」と同等であると報告されている[7]．

Wikipediaなどの Web事典と通常の電子事典の最大の違いは，記事（概念）どうしがリンクで互いに参照されていることである．筆者らは，予備調査としてWikipedia内におけるリンクの数をカウントしたところ，約65万ページ（2005年10月の段階）から約1,000万の内部リンク（Wikipedia内へのリンク）を抽出した．Wikipediaは閉じられた語彙空間の中で密なリンク構造を持っており，多いものでは数百のサイト内部へのリンクを持つ記事も存在した．この中で，リンク切れやリンク間違いなどの無効リンクを取り除いても，約715万の内部リンクが存在した．

一方，情報検索におけるクエリ拡張などの有用な応用から，シソーラス辞書の重要性が認識されている．クエリ拡張とは，（Web）文書の検索システムにおいて，ユーザがクエリとして入力したキーワードを拡張し，意味的に関連する語を抽出することで，キーワードを直接含まない文書であっても関連度を計算することができる．シソーラス辞書は，語彙どうしの関係を定義した辞書であり，関係性（is-aやpart-ofなど）を明確に定義した「関連シソーラス」（Relation Thesaurus）と，与えられたクエリから連想される語を抽出するための「連想シソーラス」（Association Thesaurus）に大別される．本研究では後者の連想シソーラスの構築に関する研究を進めてきた．連想シソーラスは，各概念をノード，関連度をエッジとする一種の重み付き有

† 大阪大学大学院情報科学研究科マルチメディア工学専攻
Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University

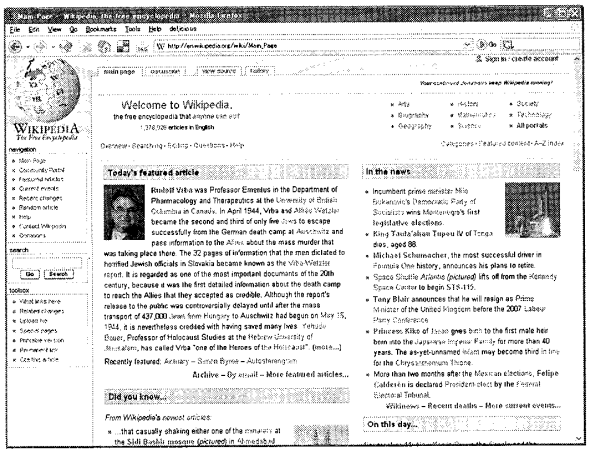


図 1 Wikipedia
Fig. 1 Wikipedia.

向グラフとして表現される．関連シソーラスのような階層構造（Hierarchy）ではなく，語と語の関係がネットワーク状に配置されており，与えられた語から関連する概念のリストを高速に抽出することが可能である．

筆者らは，これまでの研究において，Wikipedia のリンク構造を解析することで，語彙どうしの関係を定義した連想シソーラス辞書を高精度で構築できることを示してきた[10]．クエリ拡張の際に，本研究によって構築された膨大な量のシソーラス辞書を利用することで，広い範囲の語彙をカバーすることが可能となる．

しかし，実験を進めていく中で，これまでの提案手法における 2 つの問題点が明らかになった．1 つ目の問題点は，特定の条件下でシソーラス辞書構築の精度が低下することである．これは，詳細な調査の結果，一般的な語の解析の際に精度が低下していたことが分かった．この問題を回避するためには，Web 事典のリンク特性を考慮してより最適化された手法が必要となる．

2 つ目の問題点は，スケーラビリティである．文献 10) の手法では，再帰的な探索により，n ホップ先までの語彙どうしの関連性を抽出するアルゴリズムと自然言語とのマッピングを実現したが，Wikipedia のように日に日にその数が増加するような膨大な記事数を持つ Web 事典を解析するためには，より高い精度とスケーラビリティを兼ね備えたシソーラス辞書の構築手法が必要となる．そこで，本研究では n ホップ先までのリンク構造を効率的に解析し，語どうしの関連度を算出する手法 *lfibf* と，3 つの応用手法「単純法」「対数近似法」「Forward/Backward リンク重み付け手法（以下 FB 法）」を提案し，実験によりその有効性を示す．文献 10) の手法は，計算方法は異なるがその結果は単純法の結果に相当する．さらに，隣接行列を効率良く圧縮するデータ構造である二重二分木を提案し，計算の効率化を図る．

本論文の以下では，2 章で関連研究について述べ，3 章で Wikipedia マイニングの手法を解説する．4 章では，提案手法の詳細について記述し，5 章で 2 つの実験により，筆者らの提案手法により生成されたシソーラス辞書を評価し，その有用性を示す．最後に，6 章でまとめと今後の展開を記述する．

2. 関連研究

2.1 自然言語処理によるシソーラス辞書構築

シソーラス辞書は，語の意味的な関係性を表現する辞書として，自然言語処理だけでなく幅広い研究領域で利用されてきた[12]．特に，情報検索（IR）の分野では，語彙のミスマッチを防ぐことや同義語・類義語などを提案することなどで検索精度を向上させることに利用されてきた．

シソーラス辞書を構築する最も単純な方法は，人間の手によるものである．今までに，WordNet[9] や EDR 電子化辞書に代表される機械可読なシソーラス辞書を構築する取り組みが行われてきた．しかし，このようなシソーラス辞書の構築においては，概念を追加・更新するためには人間の手作業による膨大な手間がかかるため，最新の概念や一般的でない語彙などへの対応が難しいのが現状である．そのため，精度の高いシソーラス辞書を低コストで（半）自動的に構築する手法が必要とされている[13]．

自然言語処理によるシソーラス辞書構築の研究の歴史は古く，コーパス解析により（半）自動的に構築する手法は数多く提案されてきた．たとえば，語の共起関係に基づいて構築するもの[12] や，語のフィルタリングやクラスタリング手法を用いる研究[2),5)] などがある．しかし，自然言語処理において，語義やかかり受けなどの曖昧性および多義性の解消，同義語の同定などの諸問題はいまだ残っており，シソーラス辞書構築の精度低下の主要因となっている．

また形態素解析の問題もある．自然言語処理によりシソーラス辞書を構築する場合，前処理として，入力文を意味を持つ最小の言語単位である形態素にわけ，品詞タグを付与する必要がある．形態素解析および品詞タグを付与するツールとしては，Brill の Tagger[1] などが有名であるが，未知語への対応や曖昧性の取扱いなどが問題となっている．

2.2 Web サイトからのシソーラス辞書構築

Web コーパスと通常の文書コーパスの性質の最も大きな違いは，ハイパーリンクである．リンクは，単に他ドキュメントへ移動するための機能を提供するだけ

でなく，トピックの局所性やリンクテキストなど重要な情報を豊富に有している[4]．トピックの局所性とは，リンクでつながっているページどうしは，つながっていないページどうしに比べて同じトピックに関する記述である場合が多いという性質である．Davison の研究[6] は，このトピックの局所性が多くの場合に正しいことを示している．また，リンクテキストも Web マイニングによるシソーラス辞書構築において重要な役割を果たす．リンクテキストとは，リンク（A タグ）における内部テキスト部分を示す．たとえば，以下のようなハイパーテキストを考えた場合，テキスト部分「Apple」がリンクテキストに相当する．リンクテキストは一般的に被リンクページの内容（要約）を表現していることが多い．

```
<a href="http://en.wikipedia.com/wiki/Apple_Computer">
Apple
</a>
```

上記のような Web コーパスの特徴を活かし，リンク構造を解析することで，シソーラス辞書を自動的に生成する研究が最近注目を集めている．Web マイニングによるシソーラス辞書構築では，Web コンテンツの増加・更新に従い，新しい語や他の語との関係などの情報を更新することができることが大きな特徴である．たとえば，Chen ら[3] は，Web ページどうしのリンク構造を解析することで Web シソーラス辞書を自動的に構築する新しい手法を提案している．Chen らの手法では，Web サイトの階層構造からサブツリーと呼ばれる多数の木を形成し，木の中での語の共起性を利用することで語の関連度を計測している．しかし，Chen らの手法は，スケーラビリティに関する考察がなく，Wikipedia のような大規模な Web サイトに適用した場合，多数のサブツリーを構成することになり，膨大な計算時間を必要とするという問題がある．そのため，Web 事典の特性を考慮したスケーラビリティの高いシソーラス辞書構築手法が必要とされている．

3. Wikipedia マイニングによるシソーラス辞書構築：従来アプローチ

Wikipedia マイニングとは，筆者らの造語で，Wikipedia に対して Web マイニングを行い，有益な情報を抽出する手法の総称である．Wikipedia を解析してシソーラス辞書を構築する場合，適用可能な従来手法がいくつか存在する．本章では比較対象の従来手法として TF-IDF と Chen らの手法を Wikipedia の解析に適用する方法を解説する．

3.1 TF-IDF

TF-IDF[11] は，文書中の重要なキーワードを抽出するための手法であり，情報検索の研究領域を中心に数多くの研究がなされてきた．TF-IDF は，TF（Term Frequency）と IDF（Inverse Document Frequency）の 2 つの指標を利用し，文書中の各語の重要度を計算する．まず，TF は単純に文書中における単語の出現頻度であり，文書中に多く含まれる単語が特徴語として検出される．次に，IDF は単語が他の文書に出現する割合であり，他の文書に含まれる数が多い単語は IDF の値が少なくなる．つまり，IDF は一般語のフィルタとして働く．

TF-IDF は，一般的には文書中の重要な単語を抽出するアルゴリズムとして利用されるが，Web 事典においては特定の概念に対して関連度の高い概念を抽出するために適用可能であると考えられる．これは，Web 事典においては 1 ページが 1 概念に対応し，リンクは他の概念に対する意味的かつ明示的な関係を示すためである．そこで，文書中のリンクの重要度を以下の式によって求める．

$$tfidf(l,d) = tf(l,d) \cdot idf(l), \tag{1}$$

$$idf(l) = \log \frac{N}{df(l)}. \tag{2}$$

$tf()$ は，文書 d におけるリンク l の出現回数である．N は総ドキュメント数を示し，$df(l)$ はリンク l を含む文書の数を返す関数である．つまり，リンク l の重要度は文書内での出現頻度と IDF に応じて増加する．

3.2 Chen らの手法

Chen ら[3] は，Web ページどうしのリンク構造を解析することで Web シソーラス辞書を自動的に構築する新しい手法を提案している．Chen らの研究ではドメインを限定して Web サイトを選定した後にリンク構造の解析を行い，リンクテキスト上に出現する語の共起性を利用して語どうしの関連度を算出している．Chen らの手法の概要を以下に示す．

(1) ドメイン特有の Web サイト集合を Google Directory などから抽出する．

(2) FOM（Function-based Object Model）により，各ページ内のリンクを上位ディレクトリへのリンクや下位ディレクトリへのリンク，同ディレクトリへのリンクであるかなどの情報を利用して Semantic リンクと Navigation リンクへと分類する．

(3) Semantic リンクを用いてコンテンツの階層構造を分析する．

(4) 各ページの Backward Link を解析し，TF-IDF

によりそのページを最もよく示したリンクテキストをページの要約として採用する．

(5) Webサイト集合を融合し，1つの階層構造にまとめる．この際，手順(3)と(4)で構築した階層構造を構成する各ノードに対して，NLPWin（自然言語処理ツール）を用いて単語に分割して利用する．

(6) 各ノードに対して子孫ノード，祖先ノード，兄弟ノードを再帰的に深さ d で探索し，語の共起性を利用して関連度を求める．

Chenらの手法は，ディレクトリ階層を利用してサイトにおける概念階層を構築する．しかし，Wikipediaでは記事は1つのディレクトリにまとめて格納されており，階層構造が存在しないため，概念階層を構築することができない．つまり，子孫ノードサブツリーと祖先ノードサブツリーを構築できないため，兄弟ノードのサブツリーが主な解析対象となる．

Chenらの方法には大きく2つの問題がある．1つ目は同義語や多義語に関する考察がなく，語の関連性を解析する際に精度低下が発生する点である．そして2つ目は，大規模なWebサイトに対して適用した場合，計算に膨大な時間が必要となるため，現実的な時間で解析が収束しないという点である．

4. lfibf

筆者らは，Wikipediaが膨大なコンテンツ量を持っていながら，サイト内部で密なリンク構造ができていることに着目し，リンク構造を解析することで概念どうしの関係を抽出できることを示してきた[10]．しかし，文献10)の手法では，ドメイン特有の語彙や固有名詞に関しては精度が高くシソーラス辞書の構築ができているものの，一般名詞や国名など特定の状況下ではシソーラス精度が低下することに気づいた．また，シソーラス辞書構築に必要となる時間も課題であった．急速に増加する記事数に応じて，リアルタイムにシソーラス辞書を構築するためには，より高速な構築手法が必要となる．

そのため，本研究では大規模Web事典に対して効率的かつ高精度にシソーラス辞書を構築する手法の実現を目指し，*lfibf* と3つの応用手法を提案する．

本章では，*lfibf* の基本アルゴリズムと，3つの応用手法「単純法」「対数近似法」「Forward/Backwardリンク重み付け手法」について詳述する．

4.1 lfibfの基本方針

Web事典は，記事とその関連記事が互いにリンクで参照されているネットワークであるため，記事をノード集合 V，参照（リンク）をエッジ集合 E とする有向グラフ $G = \{V, E\}$ で表現できる（**図2**）．このとき，2記事間（v_i，v_j）の関連の強さを計測する問題を考えた場合，関連の強さは以下の2つの要素に依存すると考えられる．

- 記事 v_i から記事 v_j へのパスの多さ
- 記事 v_i から記事 v_j への各パスの短さ

つまり，記事 v_i から記事 v_j へのパスが多ければ多いほど，記事間の関連性は強く，またそのパスの長さが短ければ短いほど強く関連すると考えられる．ここで，パスとはリンクを伝って記事 v_i から記事 v_j へと移動可能な経路を示す．

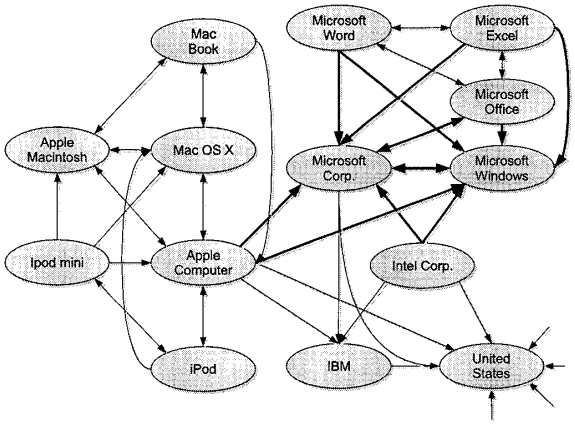


図2 Wikipediaにおけるリンクのグラフの例
Fig. 2 Link graph on Wikipedia.

ただし，パスを抽出する際には，リンクの順方向だけでなく，逆方向のパスも抽出するものとする．これは，ある記事 v_i から記事 v_j の関連の強さを計測するとき，記事 v_i のForward Link先の記事が重要であることと同様に，Backward Link先の記事関係も関連度を算出するための指標として重要であると考えられるためである．

たとえば，図2（太線部分）において，記事「Microsoft」は記事「Microsoft Windows」に対して多くの短いパス（直接のリンクと長さ2のパスを5本）を持っており，強い関連性を持っていることが容易に予想できる．

そのため，v_i から v_j への全経路 $T = \{t_1, t_2, \ldots, t_n\}$ が与えられたとき，記事 v_i から記事 v_j の関連性 lf（Link Frequency）を以下の式により表現する．

$$lf(v_i, v_j) = \sum_{k=1}^{n} \frac{1}{d(|t_k|)}. \quad (3)$$

d は経路 t_k の経路長に応じて増加する関数であり，単調増加関数や指数関数を利用することができる．一方，個々の記事が持つリンクの数も記事間の関連性に

影響すると考えられる．たとえば無数のリンクを持つ記事は，他のどの記事に対しても多数の短いパスを持つことが考えられる．たとえば，図 2 において，記事「Microsoft」から「United States」に対して短いパスが複数存在するが，「United States」は他の記事からも非常に多く参照されている一種の一般語である．このような記事は，上記の lf だけでは他のどの記事に対しても強い関連度を持つことになる．そのため，上記の 2 つの要素に加え，被参照数も考慮して関連度を計算する必要がある．そのため，上述の lf に加え，ibf（Inversed Backward link Frequency）を導入し，$lfibf$ を以下のとおりに定義する．

$$lfibf(v_i, v_j) = lf(v_i, v_j) \cdot ibf(v_j). \quad (4)$$

$$ibf(v_j) = \log \frac{N}{df(v_j)}. \quad (5)$$

N は全記事数，$df(v_j)$ は記事 v_j が持つ他の記事へのリンク数とする．つまり，$lfibf$ は多くのリンク先を共有するが，他の記事とはリンク先を共有しない記事により高い値を示す．また，同じ距離（たとえば距離 1，直接リンク関係にある）の記事であっても，より多くリンク先を共有する記事に対して高い値を示す．

このとき，ibf では記事 v_j の持つ Backward リンク数だけ考慮し，中継ノードの Forward リンクの数を考慮していないのは，Forward リンクを多く含む記事（通常は多くのユーザによって精査されている記事）内のリンクが軽視されることを防ぐためである．

4.2 単純法

前述のとおり，$lfibf$ はリンク構造を解析し，経路の多さと距離に応じたスコアリング（lf）と被リンク頻度（ibf）を利用した，語彙どうしの関連性の数値化アルゴリズムである．しかし，グラフ内の全経路を算出することは，ノード数とリンク数に応じて莫大な計算量が必要となる NP 困難問題であることが知られている．そのため，$lfibf$ では探索距離を限定し，近似解を求めるアルゴリズムを実現している．以下にアルゴリズムの詳細を解説する．

有向グラフ G は，隣接行列（A）で表現することが可能であり，隣接行列のべき乗 A^n を計算することで距離 n のすべてのノード（記事）への経路数を算出できることは周知である．ところで，記事 v_i は，Forward リンクと Backward リンクの 2 種類のリンクを持つ．v_i の Forward リンクとは，記事 v_i から別の記事へジャンプするリンクの集合であり，$F_{v_i} = \{f_{i1}, f_{i2}, f_{i3}, \ldots, f_{im}\}$ と定義する．また，Backward リンクは別の記事から記事 v_i へジャンプするリンクの集合であり，$B_{v_i} = \{b_{i1}, b_{i2}, b_{i3}, \ldots, b_{il}\}$ と定義する．このとき，$lfibf$ では Forward リンクだけでなく Backward リンクの解析も行うために，隣接行列 A と転置行列 A^T を加算した行列 A' を解析に利用する．

$$A' = A + A^T. \quad (6)$$

次に，行列 A' 内の各要素を各列における要素の和（被リンク数）で除算した推移確率行列 P およびその積 P^n を利用することで，n ホップ先のノード（記事）への推移確率を算出し，記事どうしの関連性を算出する．ここで，各行における要素の和ではなく，各列の要素の和で除算するのは，隣接行列 A' の各列の和は，記事 v_j の被リンク数（被リンク数）であり，前述の ibf を近似できるためである．

ここで，v_i から v_j への n ホップ以内の全推移確率行列 $P^1, P^2, \ldots, P^n$ が与えられたとき，$lfibf$ を以下のとおり定義する（$p^l_{i,j}$ は P^l の i 行 j 列要素）．

$$lfibf(i,j) = \sum_{l=1}^{n} \frac{1}{d(n)} \cdot p^l_{i,j}. \quad (7)$$

$d(n)$ は，ホップ数 n に応じて増加する単調増加関数もしくは指数関数である．また，P^{n+1} の算出には P^n の解析結果を利用するため，解析対象の記事を起点として再帰的に解析を繰り返す方式に比べ，効率良く解析することが可能である．

4.3 対数近似法

上述の単純法のように，行列 A' の各要素を各列の和で単純に除算した場合，被リンク数に反比例して値が線形で減少するため，ibf の近似としては不十分であることが考えられる．そのため，行列内の各要素を被リンク数で除算する代わりに，対数関数を利用して除算することで，ibf を近似する手法（対数近似法）を提案する．対数近似法では，A' から推移行列 P を計算する前に，A' の各要素を以下の数式に従って更新する．

$$a'_{i,j} = a'_{i,j} \cdot log \frac{N}{|B_{v_j}|}. \quad (8)$$

4.4 Forward/Backward リンク重み付け手法

ここで，リンク解析における Forward リンクと Backward リンクの重要性について考察する．Forward リンクは，通常記事の著者の主観に依存し，重要な単語へのリンクがあるか，その数が妥当かなど，情報の信頼性にはばらつきがみられるのが一般的である．これは，Wikipedia などのように不特定多数のユーザが協調してコンテンツを構築するようなシステムでは特に顕著であり，新しい記事の場合は内容のミスやリンクの過不足などが多々見受けられる．しかし，その一方で多数のユーザによって長期間洗練された記事は，

関連の深い記事へのリンクが豊富であり，内容に間違いも少ない．

このような記事（ページ）の情報の信頼性は，Backward リンクの数によって判断できる場合が多い．これは，Backward リンクがページに対する「投票」と見なすことができるためである．HITS アルゴリズムや PageRank アルゴリズム[8] など，最近の Web 構造解析のアルゴリズムでも，Backward リンク解析が客観的な情報を得るために有用であることが示されている．

実際に，ドメイン特有の単語（専門用語）の場合には特に Backward リンクが重要な役割を果たすことが予備実験によって判明している．これは，ドメイン特有の単語の場合，ドメイン内で密なリンク構造が形成されており，Forward リンク解析では発見できなかった語彙どうしの関連情報を Backward リンク解析から抽出できたためだと考えられる．しかし，その逆に一般的な語の場合は，様々な分野の記事から参照されるため，Backward リンク解析の結果が分散してしまい，関連語の精度が下がってしまうという現象がみられた．これは，Backward リンク数の多い語（一般的な語）は，記事の内容が信頼できるため，Forward リンクを重視して解析することが望ましい一方で，Backward リンク解析の結果は分散してしまうため，比重を下げる必要があることを示唆している．そのため，記事の Backward リンク数に応じた Forward/Backward リンクの重み付け手法を適用した *lfibf*（FB 法）を提案する．FB 法では，A' の各要素を以下のとおりに更新する．

$$a'_{i,j} = W(|B_{v_j}|)\cdot a^T_{i,j} + (1 - W(|B_{v_j}|))\cdot a_{i,j}. \quad (9)$$

$$W(|B_{v_j}|) = 0.5/(|B_{v_j}|^{\alpha}). \quad (10)$$

$W()$ は，記事 v_j の持つ Backward リンク数に応じて Backward リンクの重みを変更する関数である．α はパラメータであり，これまでの予備実験から，平均して数十から最大数百のリンクを持つ Wikipedia においては，0.05 程度が妥当な値であるという知見が得られている．

4.5 二重二分木による高速化

有向グラフのデータ構造としては，通常隣接リストによるデータ表現か，隣接行列を二次元配列によって表現する方法が一般的に利用される．

まず，隣接リスト方式は，必要とするデータ容量が少ないが，各要素にアクセス（読み出し・書き込み）するために必要となる計算量が多く，大規模な Web 事典に対する解析に不向きであるという問題がある．

一方，隣接行列方式は，非常に少ない計算量（$O(1)$）で各要素にアクセスできるため，任意のノード間にリンクがあるかを判断することやリンクを追加するために要する時間は最少である．その一方で，解析対象が Wikipedia のように膨大な記事数を保持する Web 事典の場合，大量のデータ容量と解析時間を必要とする．Wikipedia には 100 万件以上の記事（英語のみ）が公開されているが，このデータに対して隣接行列を作成する場合，最低 100 万行 × 100 万列の膨大な行列が必要となり，行列データを保持するだけで数百 GB〜数 TB の記憶容量が必要となるうえに，行列積に要する計算量は $O(N^3)$ と膨大なものとなる．

しかし，記事どうしの参照関係を定義した隣接行列は，0 要素の数（リンクのない記事どうし）のほうが圧倒的に多い疎行列であるため，実時間で解析を進めるためには，データを効率良く格納するデータの圧縮方法と解析アルゴリズムが重要となる．*lfibf* では，任意の記事が保持するリンク情報にアクセスするための二分木と，記事を指定して他記事への推移確率を保持するための 2 種類の二分木を利用し，非 0 要素だけを格納することで，行列を圧縮する（**図 3**）．

二重二分木は，i 木（i-Tree）と j 木（j-Tree）の 2 種類の木から構成される．i 木の各要素は，行列の行に対応し，記事の ID から j 木へアクセスするためのアドレスを保持する．j 木は特定の記事から他の記事へのリンク情報を保持する木であり，行列の列に相当する．つまり，記事 v_i が記事 v_j へのリンクを持つかどうかを調べるためには，まず i 木から記事 v_i の j 木へのアドレスを抽出し，j 木を探索することで i 行 j 列の要素を抽出する．この結果，リンクの挿入，参照ともに $O(logN)$ の計算量で要素にアクセス可能となる．

また，二重二分木の大きな特徴は，大規模な行列に対して効率良く行列積を計算できる点である．二重二分木での行列積の計算アルゴリズムを以下に示す．

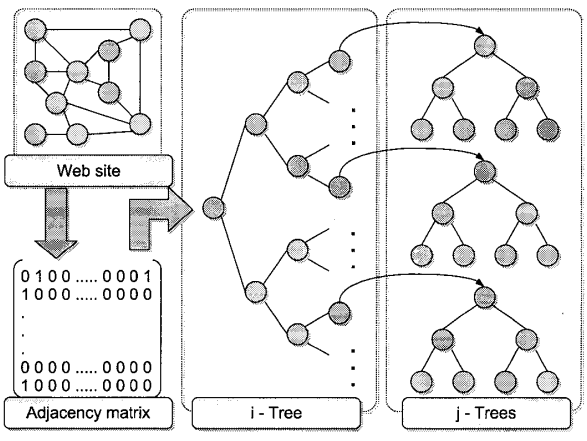


図 3 二重二分木によるグラフ表現の概念
Fig. 3 Double binary tree for graph expression.

Algorithm *MultiplyMatrix*(A)
1: *For* i *in* i-*Tree*
2: *For* j *in* j-*Tree*(i)
3: *For* k *in* j-*Tree*(j)
4: $R_{i,k} = R_{i,k} + A_{j,k} * A_{i,j}$;

ここで，変数 i，j，k はそれぞれ各記事の ID とし，関数 j-*Tree*() は記事 ID をキーとして各記事に対応する j 木へのアドレスを検索する関数である．本アルゴリズムでは，非 0 要素を順次たどり，i 木から各記事の ID を抽出し，j-*Tree*() によって行列 A の i 行目の要素を抽出する．次に，j 行目の要素をすべて抽出し，記事 v_i の行列積として積の対象となる要素に順次値を乗算し，計算結果を保持する二重二分木 R の i 行 k 列要素に加算することで，行列積の計算を行う．このとき，1 行目の繰返し処理の計算量は記事数 N に依存するものの，2 行目，3 行目の繰返し処理は，記事の平均リンク数に依存するため，N に比べて非常に小さい値となる．二重二分木に格納された要素を取り出すために必要となる計算量は $O(logN)$ であるため，行列積に要する計算量は $O(NlogN)$ となり，効率的な計算が可能である．ただし，2 行目，3 行目の繰返し回数が十分に小さいのは，Web 事典においては，各記事の平均リンク数が記事数に比べて非常に少なく，行列が非常に疎な状態に保たれているためである．そのため，行列積を 3 乗，4 乗と繰り返し，n ホップ先までの解析を行うためには，行列積の計算が終了するたびに，行列 R の各行において，スコアの高い順にトップ k 件の要素だけを残し，行列を疎な状態に保つ必要がある．

ここで，注目するべき点は，i 木において i 行目の要素について解析をしている際には，解析結果 R への操作も i 行目の要素に限定される点である．これは，一番外側の繰返し処理は，並列分散処理可能であるため，複数の計算機を利用して並列で解析を行うことで，計算時間をさらに短縮することが可能であることを示している．

5. 実験と考察

本章では，*lfibf* の有効性を示すために行った実験について述べる．

5.1 実験の概要

本研究では，提案手法の有効性を示すために，*lfibf* の 3 応用手法，ページ内の重要語を抽出するアルゴリズムとして実績のある TF-IDF，Chen らの手法を 2 つの実験により比較した．

第 1 の実験は，シソーラス辞書の精度に関する実験である．本実験では，クエリとして入力されたキーワードから関連語を 30 件抽出する簡易の検索エンジンを構築し，評価に利用した．この検索エンジンでは，まず与えられたクエリに対して各手法で構築された関連語のリストを関連度の高い順に 30 件抽出する．次に，それぞれのシソーラス辞書で構築した関連語を順不同で被験者に提示し，被験者が各関連語とクエリとの関連度を 5 段階（1：関係しない← 3：どちらともいえない→ 5：関係する）で評価した．

ただし，関係があるか否かの判断が被験者の偏った主観に依存することを防ぐために，is-a 関係や is-a-part-of 関係など，語から連想できる語のことを「関係ある語」と定義していることを被験者に明確に示したうえで実験を行った．さらに，実験結果をより公正なものとするために，被験者には「関係のある語も関係のない語も含まれている可能性がある」と伝えた．評価値としては，シソーラス辞書の精度評価でよく利用される CP 値（Concept precision）[2] を以下の式により算出した．ここで，「関係が深いと評価された関連語の数」は，被験者の評価で 5 もしくは 4 と評価された関連語の数であり，「システムが抽出した関連語の数」は，クエリから抽出された関連語の数を示す．

$$CP = \frac{\text{関係が深いと評価された関連語の数}}{\text{システムが抽出した関連語の数}}. \quad (11)$$

第 2 の実験は，単語の認知度に関する実験である．関連語 30 個に対し，その単語をどれだけよく知っているかを 3 段階（よく知らない ← どちらでもない → よく知っている）で評価をし，認知度を以下の数式によって算出した．

$$\text{認知度} = \frac{\text{回答「よく知っている」の数}}{\text{全回答数}}. \quad (12)$$

5.2 実験の結果と考察

本実験では，第 1，第 2 の実験ともに，合計 15 名の被験者に対して検索エンジンを利用させ，それぞれ別々の任意の 1 単語をクエリとして，各シソーラス構築手法で抽出された関連語 30 件に対して CP 値による評価を行った．

5.2.1 シソーラス辞書の精度

シソーラス辞書の精度に関する実験の結果を**図 4**，**図 5**，**図 6** に示す．横軸の Rank は，関連語 30 件を関連度の高い順にソートしたときの順位（ランク）を示し，縦軸の CP は該当ランクでの CP 値の平均値である．Simple，Log，FB はそれぞれ *lfibf* の 3 手法を示す．まず，単純法と対数近似法を比較した場合，対

数近似法のほうが精度良くシソーラス辞書を構築できていることが分かる．これは，推移確率行列 P を作成する際には，単に被リンク数で除算するより，対数関数で ibf を近似するほうが精度が良いシソーラス辞書構築ができることを示している．次に対数近似法とFB法を比較した場合，「Microsoft」や「Playstation」といったドメインに特化した専門的な用語などの場合は，ランキング下位ではFB法のほうが約10%ほど高い精度を実現していたが，ランキング上位において精度に大差は生じなかった（図4）．しかし，「Book」や「Music」，「Sport」といった，一般的な語（Backwardリンクの数が非常に多い語）の関連語に関しては，FB法の方が精度良くシソーラス辞書を構築できており（図5），全体の精度に約4.74%の差が生じた（図6）．TF-IDFは，一般的な語を解析する場合は，対数近似法を上回る精度でシソーラス辞書構築ができていたが，専門的な語の解析では大きく精度が低下し，全体の精度も対数近似法を下回る結果となった．また，Chenらの手法は，形態素解析による精度低下が発生した．これは，語の共起性解析において，リンクテキストを自然言語処理ツールにより空白，ハイフン，カンマ，ピリオドなどの区切り文字で単語・フレーズに分割する際に，適切ではない箇所で形態素に分割されたことに起因する．実験に利用した語のリストと，FB法によって抽出されたシソーラス辞書の関連語を**表1**に示す．

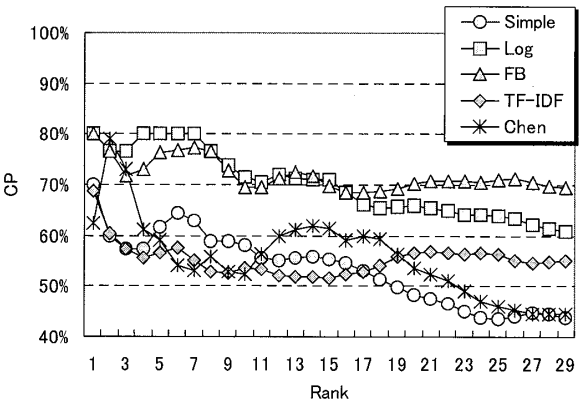


図4 専門的な語のランクとCP値の関係
Fig. 4 CP vs. Rank on domain-specific words.

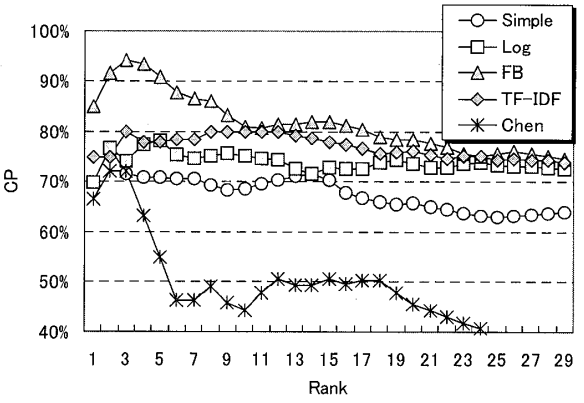


図5 一般的な語のランクとCP値の関係
Fig. 5 CP vs. Rank on general words.

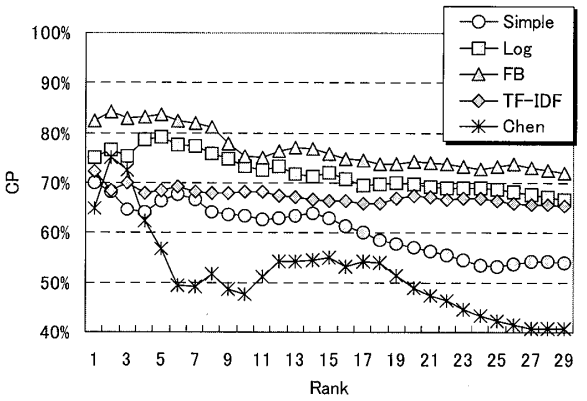


図6 最終的なランクとCP値の関係
Fig. 6 CP vs. Rank.

表1 FB法によって抽出された関連語の例
Table 1 Examples of associated terms by FB method.

クエリ	関連語		
Sports	Basketball	Baseball	Volleyball
Microsoft	Microsoft Windows	Operating system	Microsoft Office
Video game	Nintendo	Japan	Video game developer
Apple Computer	Apple Macintosh	Mac OS X	iPod
Book	Library	Diamond Sutra	Printing
PlayStation	PlayStation 2	Sony	Nintendo 64
Nintendo	Nintendo Entertainment System	Game Boy	Shigeru Miyamoto
Google	Search engine	PageRank	Google search
Tennis	Pete Sampras	French Open	Professional Tennis Championships
Sausage	Blood sausage	Braunschweiger	Leberwurst
Music	Music theory	Musician	Chord
Macintosh	Apple Computer	Microsoft Windows	Mac OS X
iPod	Apple Computer	IPod mini	ITunes
RPG	Video game developer	Piranha Bytes	Final Fantasy
Apple	Apple pie	Cider	Toffee

表 4 単語の認知度
Table 4 Recognizability of words.

手法	よく知らない	どちらでもない	よく知っている	認知度
単純法	42	110	148	49.33%
対数近似法	44	76	180	60.00%
FB 法	66	75	159	53.00%
TF-IDF	129	73	68	25.19%
Chen らの手法	52	82	136	50.37%

表 2 性能評価のための環境
Table 2 Environment for performance evaluation.

マシン	項目	仕様
ワークステーション	CPU	Pentium4 2.4 GHz × 2
（解析用）	メモリ	8,192 MB
	OS	Solaris 10
	開発言語	C++
検索エンジン用サーバ	CPU	Pentium4 3.0 GHz × 1
	メモリ	1,024 MB
	OS	Windows 2003 Server
	開発言語	C#, ASP .NET
	DB	MySQL 5.0

表 3 シソーラス辞書構築に要する時間
Table 3 Total time to analyze.

手法	計算時間
単純法	161,310 秒
対数近似法	196,491 秒
FB 法	222,813 秒
TF-IDF	637 秒
Chen らの手法	平均 121 秒/単語

5.2.2 シソーラス辞書構築に要した時間

各シソーラス辞書を構築するために利用した計算機環境と，要した時間を**表 2**，**表 3** に示す．解析対象には，2005 年 10 月の Wikipedia データを利用し，記事数約 65 万，総リンク数約 1,000 万の解析を行った．

Chen らの手法に関しては，膨大な計算量を必要とするため，実際にすべての語について解析を行うことは不可能であった．ランダムに抽出した数十の記事に対しては，平均 121 秒の解析時間が必要であった．そのため，百万を超える記事数を持つ Wikipedia では現実的な解析時間では終了しなかった．一方，*lfibf* は，多量の要素を含む行列積の計算を行っているが，合計約 62 時間（FB 法）という現実的な時間で解析処理が終了している．実際には，複数の解析用ワークステーションで並列分散処理を行ったため，十数時間で解析処理を終了している．また，FB 法によってすべての記事を解析するのに要する時間は，従来手法で一部のデータを選択的に解析する時間と同程度であり，十分に高速に処理ができていることが分かった．TF-IDF は，再帰的な解析ではないというアルゴリズムの特性上，最も早く解析が終了した．

5.2.3 単語の認知度

単語の認知度に関する評価結果を**表 4** に示す．単語の認知度に関しては，*lfibf* と Chen らの手法が高い数値を示しているのに対し，TF-IDF は約半分程度の認知度となった．これは，TF-IDF の基本方針として，「他の記事に出現しない度合」である IDF（Inversed Document Frequency）が高い語が重要語として抽出されるためである．しかし，これではあまり一般的ではない語が抽出されるため，一般的なシソーラス辞書を構築する際には有用とはいえない．たとえば，検索語として「Microsoft」という語が与えられたときに，FB 法では「Microsoft Windows」「Microsoft Office」「Operating system」などの関連語が上位に表示される一方で，TF-IDF では「Microsoft Dynamics NAV」「Microsoft Zone」「La morsure du dragon」（Microsoft 社と中国に関するフランスの小説）などといった他の記事には登場しない語が関連語として抽出された．これは，TF-IDF が文書中の特徴語を抽出する用途に向いている一方で，シソーラス辞書を構築するためには不適切であることを示唆している．

6. まとめと今後の展開

本論文では，Wikipedia などの大規模 Web 事典をマイニングし，シソーラス辞書を構築する手法として *lfibf* を提案した．諸実験の結果から，生成されたシソーラス辞書は，関連度の高い語を抽出していることが分かった．特に，被リンク頻度を考慮した Forward/Backward リンク数の重み付け手法は高精度のシソーラス辞書を構築するうえで有効であることが分かった．

本プロジェクトの成果は，Wikipedia を解析することで有用な情報を抽出するプロジェクトである「Wikipedia Mining」プロジェクトとして下記のサイトで公開している．

http://wikipedia-lab.org/

本サイトでは，*lfibf*（FB 法）によって構築されたシ

ソーラス辞書を利用することが可能である．また，開発者用のAPIをXML Webサービスとして公開しており，WSDL（Web Services Description Language）によって仕様を定めている．

本シソーラス辞書の検索システムをベータ公開し，ユーザからのフィードバックを受け付けたところ，興味深い知見がいくつか得られた．たとえば，各種ドメイン（分野）がシソーラス辞書の精度に影響を与える点などである．コンピュータサイエンスやテクノロジなどの分野の解析では，抽出された関連語は高い確率でユーザの連想する語とマッチすることが判明しているが，その一方で，「Osaka」や「Kyoto」といった地理情報（Wikipediaのカテゴリでは Geography）をクエリとし，連想語を抽出すると，「Kadoma」や「Uji」などの地名が検索結果のトップの語として抽出される．これらの地名は，検索クエリに深く関係する語であるが，ユーザが直観的に連想する語としては改良の余地があると思われる．この知見（ドメインによって抽出結果に偏りが生じる現象）は，さらに本提案手法をドメインに応じて最適化する必要があることを示唆している．

今後の展開としては，Wikinewsなど他プロジェクトも含めたWeb構造マイニングを行うことで，さらに即時性の高い語彙の抽出や精度向上が図れるものと考えられる．また，日本語を含めた多言語Wikipediaでの実験も非常に興味深い．たとえば，言語間リンクの解析による翻訳用シソーラス辞書の構築などが応用例として考えられる．ただし，これら別プロジェクトとの連携するためには十分な量のコーパスが必要となるが，現在の段階では十分なデータが他プロジェクトに揃っていないのが現状である．

また，自然言語処理技術の適用も課題の1つである．リンクの前後の文章を構文解析することで，関連度だけでなく，関連の種類（is-aやpart-of）の抽出も可能であると考えられる．

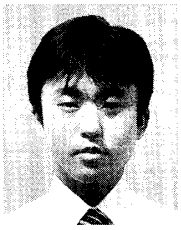
謝辞 本研究の一部は，文部科学省特定領域研究（18049050）およびマイクロソフト産学連携研究機構CORE連携研究プロジェクトの助成によるものである．ここに記して謝意を表す．

参考文献

1) Brill, E.: A Simple Rule-based Part of Speech Tagger, *Proc. Conference on Applied Computational Linguistics (ACL)*, pp.112–116 (1992).
2) Chen, H., Yim, T. and Fye, D.: Automatic Thesaurus Generation for an Electronic Community System, *Journal of the American Society for Information Science*, Vol.46, No.3 (1995).
3) Chen, Z., Liu, S., Wenyin, L., Pu, G. and Ma, W.Y.: Building a Web Thesaurus from Web Link Structure, *Proc. ACM SIGIR*, pp.48–55 (2003).
4) Craswell, N., Hawking, D. and Robertson, S.: Effective Site Finding using Link Anchor Information, *Proc. ACM SIGIR*, pp.250–257 (2001).
5) Crouch, C.J.: A Cluster Based Approach to Thesaurus Construction, *Proc. ACM SIGIR*, pp.309–320 (1988).
6) Davison, B.D.: Topical Locality in the Web, *Proc. ACM SIGIR*, pp.272–279 (2000).
7) Giles, J.: Internet Encyclopaedias Go Head to Head, *Nature*, Vol.438, pp.900–901 (2005).
8) Lawrence, P., Sergey, B., Rajeev, M. and Terry, W.: The PageRank Citation Ranking: Bringing Order to the Web, Technical Report, Stanford Digital Library Technologies Project, pp.39–41 (1999).
9) Miller, G.A.: WordNet: A Lexical Database for English, *Comm.ACM*, Vol.38, No.11, pp.39–41 (1995).
10) 中山浩太郎，原 隆浩，西尾章治郎：Wikipediaマイニングによるシソーラス辞書の構築手法，情報処理学会論文誌，Vol.47, No.10, pp.2917–2928 (2006).
11) Salton, G. and McGill, M.: *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company (1984).
12) Schutze, H. and Pedersen, J.O.: A Cooccurrence-based Thesaurus and Two Applications to Information Retrieval, *International Journal of Information Processing and Management*, Vol.33, No.3, pp.307–318 (1997).
13) Tseng, Y.H.: Automatic Thesaurus Generation for Chinese Documents, *Journal of the American Society for Information Science and Technology*, Vol.53, No.13, pp.1130–1138 (2002).

(平成18年9月15日受付)
(平成19年2月27日採録)

(担当編集委員 石川 博，有次 正義，片山 薫，木俵 豊，中島 伸介)



中山浩太郎（正会員）

2001 年関西大学総合情報学部卒業．2003 年同大学院総合情報学研究科修士課程修了．この間（株）関西総合情報研究所代表取締役，同志社女子大学非常勤講師に就任．2004 年関西大学大学院を中退後，大阪大学大学院情報科学研究科にて博士号を取得し，2007 年 4 月から大阪大学大学院情報科学研究科特任研究員となり，現在に至る．人工知能および WWW からの知識獲得に関する研究に興味を持つ．IEEE，ACM，電子情報通信学会，人工知能学会各会員．

原　隆浩（正会員）

1995 年大阪大学工学部情報システム工学科卒業．1997 年同大学院工学研究科博士前期課程修了．同年同大学院工学研究科博士後期課程中退後，同大学院工学研究科情報システム工学専攻助手，2002 年同大学院情報科学研究科マルチメディア工学専攻助手，2004 年より同大学院情報科学研究科マルチメディア工学専攻助教授となり，現在に至る．工学博士．1996 年本学会山下記念研究賞受賞．2000 年電気通信普及財団テレコムシステム技術賞受賞．2006 年日本データベース学会論文賞受賞．データベースシステム，分散処理に興味を持つ．IEEE，ACM，電子情報通信学会，日本データベース学会各会員．

西尾章治郎（正会員）

1975 年京都大学工学部数理工学科卒業．1980 年同大学院工学研究科博士後期課程修了．工学博士．京都大学工学部助手，大阪大学基礎工学部および情報処理教育センター助教授，大阪大学大学院工学研究科情報システム工学専攻教授を経て，2002 年より同大学院情報科学研究科マルチメディア工学専攻教授となり，現在に至る．2000 年より大阪大学サイバーメディアセンター長，2003 年より大阪大学大学院情報科学研究科長を併任．この間，カナダ・ウォータールー大学，ビクトリア大学客員．データベース，マルチメディアシステムの研究に従事．現在，Data & Knowledge Engineering 等の論文誌編集委員．本会理事を歴任．電子情報通信学会フェローを含め，ACM，IEEE 等 8 学会の会員．