反復文字列階層グラフによる文書からのキーワード自動抽出

白井 智† 鳥井 修† 金井 達徳†

†（株）東芝 研究開発センター

あらまし 現在，ＷＥＢ上や企業内に大量の文書データが存在する．文書にキーワードを付与することは，それら文書の検索効率を大きく向上させる．本稿では反復文字列階層グラフ上での結合度を用いて，与えられた文書から自動的にキーワードを抽出する手法を提案する．反復文字列階層グラフは，文書中に繰り返し現われる文字列を再帰的にくくりだすことで，文書全体を非巡回グラフに再構成したものであり，結合度はグラフ中の文字列の生起確率から算出できるものである．したがって本手法では，形態素解析を用いるキーワード抽出で必要とされる辞書等の事前知識を利用せずに，キーワード抽出が行なえる．

キーワード テキストマイニング，情報検索，知識発見

1. はじめに

インターネット上の Web ページや電子メール，企業内のデータ群，日々発行されるニュース等々，我々が利用する電子化されたドキュメント情報は増加し続けている．

ドキュメントが蓄えられていく一方で，それを取り出す検索技術の重要性も増している．ドキュメントの内容をよく表現するキーワードを付与し，検索時に利用することで,文書の検索効率をあげることができる．

しかしながら，過去から蓄積されてきた文書や，あらかじめキーワードでの検索を想定していない文書にはキーワードが付与されておらず，検索しようとすると全文検索等の手段に頼らざるを得ない．そのようなキーワードの付与されていないドキュメントに人手でキーワードを付与するには膨大なコストを必要とするため，ドキュメントに付与するために，自動的にキーワードを抽出する技術が望まれている．

キーワードの自動抽出を行なう際には，前処理として形態素解析[1]が用いられることが多い．形態素解析では与えられた辞書を元に文章の解析を行なうため，抽出できるキーワードの質は形態素解析に利用した辞書に強く依存する．また，形態素解析を利用しないキーワード自動抽出も研究されている[4]が，この手法では，抽出を行なう際に膨大なドキュメントを処理することによって得られる統計データを利用する．いずれの手法でも，適正なキーワード自動抽出を行なう場合には，良質の辞書や統計データ等の事前知識の準備が重要になっている．

我々は，このような事前知識を必要とせず，解析しようとしている文書そのものだけを利用して，キーワード自動抽出を行なう手法を開発した．本手法では，解析対象の文書から文字の繰り返しに基づいた反復文字列階層グラフへの変換を行い，グラフ中の各ノードをキーワードの候補とする．さらに，反復文字列階層グラフから参照数と結合度を用いて各ノードを評価し，キーワードの自動抽出を行なう．

以下，本稿では 2 節で本手法の概要を説明する．3 節では本手法で用いる反復文字列階層グラフについて説明し，4 節ではキーワード抽出で利用する評価尺度の説明を行なう．その後 5 節で，実際に試作したキーワード抽出システムによる抽出結果を述べ，6 節でまとめを行なう．

2. キーワード抽出方式の概要

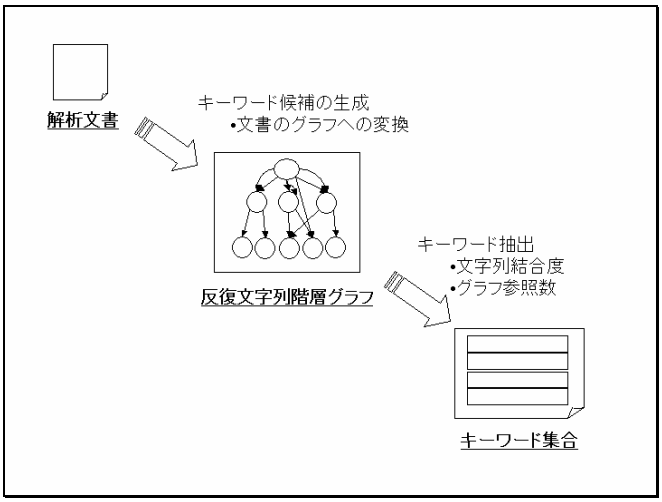

図 1 本手法の概要

文書からキーワードを自動抽出する本手法の概要を**図 1**に示す．

本手法は大きく二つの段階からなる．一段階目で，文書からキーワード候補を生成する．キーワード候補を得るために，文書を反復文字列階層グラフに変換する．反復文字列階層グラフは，文書の文字列の反復を抽出したグラフであり，グラフの各ノードがキーワード候補に対応する．そして，二段階目でキーワード候補から文字列の結合度及びグラフ中の参照数を用いて，キーワード抽出を行なう．

日本語の文章のようなわかち書きを行なわない言語からキーワード抽出を行なう場合，連続した文字列から単語の区切り位置を確定させる作業が必要になる．

精度よく区切り位置を決定することは一般に難しい．文章を形態素に分割する処理である，形態素解析では，質のよい辞書等の事前知識や，膨大なドキュメントから得られる統計データを利用して，単語の区切り位置の精度をあげている．

我々は，文書の文字列をグラフ化することで，柔軟に区切り位置を設定したキーワード候補を生成し，その後，それら候補からキーワードを抽出する手法をとった．この手法であれば区切り位置を精度よく確定する必要はないため，膨大な統計データを用いずにキーワードを抽出することができる．

キーワード候補の生成

キーワード自動抽出の一段階目は文字列を反復文字列階層グラフに変換し，キーワード候補を得る処理である.本手法では,「特徴的なキーワードはそのドキュメント中に二回以上反復して出現する」という仮定に基づき，ドキュメント中に二回以上出現する文字列を，キーワード候補とする．キーワード候補の生成には，反復文字列階層グラフを利用する．

反復文字列階層グラフの各ノードには対応する文字列が存在し，ノード間の参照関係はノードが対応する文字列の包含関係を示している．反復文字列階層グラフの概念図を図 2に示す．この概念図は『ABABCDCDDCECE』という文字列を変換した例である．概念図ではノードを円でしめし，そのノードが対応する文字列を円の中に記述した．

『AB』を対応するノードが参照しているノードは，『A』と『B』でありどちらも『AB』の部分文字列となっている．逆も成立する．ただ 1 つだけ，どのノードからも参照されていないノードが存在する.これを，ルートノードと呼ぶ．どこからも参照されていないルートノードには，他のどの文字列の部分文字列にもなりえない文字列が対応する.つまり,ルートノードは，解析しようとしている文字列そのものに対応する．

文書から反復文字列階層グラフを構成する際には，反復して出現した文字列をくくり出すことでルートノード以外のノードが生成される．従って,『AB』というノードの存在は，解析しようとしている文字列の中に『AB』という文字列が少なくとも二回以上反復して出現していることを示している．

このように，反復文字列階層グラフ中の各ノードが，二回以上反復して出現する文字列に対応することになるので，抽出すべきキーワードは反復文字列階層グラフ中のいずれかのノードに含まれることになる．

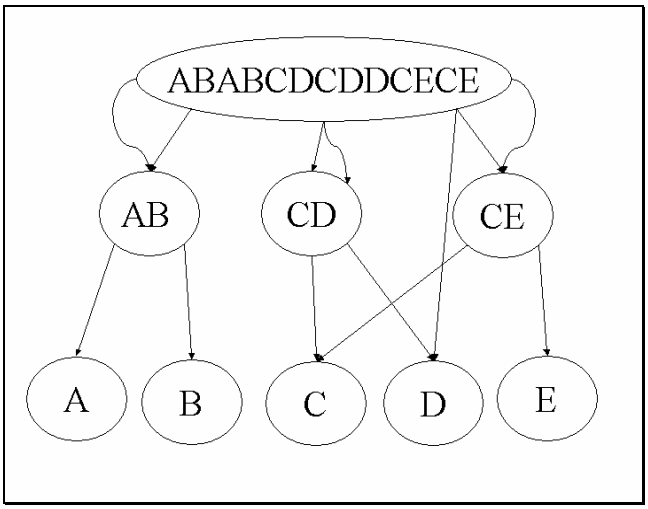

図 2 反復文字列階層グラフ 概念図

キーワード抽出

キーワード自動抽出の二段階目では，一段階目の処理で得られたキーワード候補からキーワード抽出を行なう．

キーワード候補は，前述したように反復文字列階層グラフ中の各ノードである．キーワード抽出の段階では，各ノードに対して，結合度によるキーワードらしさを評価し，全てのノード間の順序を決定する．決定した順序の上位いくつかのノード，又は，ある得点以上を獲得したノードに対応する文字列をキーワードとして抽出する．

本手法では,「繰返し同じ順序で出現する頻度の高い文字列はキーワードを構成している」という仮定に基づき，グラフ上のトポロジから得られる参照数の情報と結合度を用いた候補からのキーワード抽出を行なう．詳細は 4 節で説明するが，グラフ上の参照数は，その文字列がいかに独立してドキュメント中で利用されているかを示す尺度であり，結合度はドキュメント中の文字列の連接頻度から得られる文字列間の結びつきの強さを示す尺度である．

既存技術との比較

日本文の処理，特にキーワード抽出のような処理では，形態素解析が利用されることが多い．

形態素解析には，解析対象の言語の単語辞書，品詞付きの学習データ等を元にして品詞および単語を用いた遷移確率を計算した学習データが必要となる.特に，辞書に登録されていない語句が解析文書中に出現した場合には，正確に解析することが期待できなくなる．

形態素解析の結果を利用するキーワード抽出では，それら形態素解析の解析ミスに影響されるため，キーワード抽出の精度をあげるには形態素解析の解析ミスをおぎなう手法を用いる必要がある．

形態素解析を用いて，専門用語を抽出する手法に“出現頻度と連接頻度に基づく専門用語抽出”[3]がある．この手法では，形態素解析で細分化されてしまった単語を連接頻度を利用して再構成し，形態素解析の解析ミスをおぎなっている．

一方，形態素解析を利用せずに，キーワード自動抽出を行なう研究もなされている．形態素解析を利用しない場合には，大量のドキュメントから統計データを生成し，キーワード抽出の尺度として利用する．

“未踏テキスト情報中のキーワードの抽出システム開発”[4]は，形態素解析を用いずにキーワード抽出を行なう研究であり，この手法では単語の出現頻度と出現集中という統計量を利用してキーワード抽出を行っている．

どちらの手法でも，適正なキーワード抽出を行なう場合には，良質の辞書や統計データ等の事前知識の準備が重要になる．それらと比較して，我々の手法は，キーワード抽出に事前知識を利用しない．

前述したように，本手法では「特徴的なキーワードはそのドキュメント中に二回以上繰返し出現する」という仮定を置いた．これは，特徴的なキーワードがドキュメント中に一度しか出現しないことは稀だと考えられること，また，[4]の抽出結果を見ても，抽出されているキーワードのほとんどがドキュメント中に二回以上出現しているものがほとんどである，という考察の結果である．

また，「繰返し同じ順序で出現する頻度の高い文字列はキーワードを構成している」という仮定に関しては，文書を文字列と見た場合，キーワードを構成する部分の文字列の繋がりは，違う部分の文字列の繋がりより強い頻度で出現していると考えられること，また[3]では，これら結合度が単語を複合させた複合単語を抽出するのによい成果をあげている，という観察の結果である．

これらの情報は解析するドキュメントそのものから取得でき，従って，我々の手法ではドキュメントそのものだけを利用してキーワード抽出が行なえる．

3. 反復文字列階層グラフ

本節では，反復文字列階層グラフについて詳細説明を行なう．以下，まず反復文字列階層グラフの構造を説明し，その後，文書から反復文字列階層グラフへの変換方法を説明する．

3.1. 反復文字列階層グラフの構造

反復文字列階層グラフは，あるノードから同じノードへの参照が複数存在することも許し，また，あるノードから他のノードへの参照の間に順序が規定されるグラフである．データ構造としては，各ノードが他のノードもしくは文字への参照を保持し，これら参照の間の順序を配列等を用いて保存することになる．

図 3に反復文字列階層グラフの構造(一部分抜粋)の例を示す．各ノードを楕円で示し，あるノードからノードへの参照を，楕円の中の文字列から，他のノードへ延びる矢印で示した．矢印の根本の文字列は，参照先のノードに対応する文字列を示している．また，他のノードへの矢印がない文字は，文字への参照を示している．

同じ文字列は同じノードとして表現され得るので，ノードは複数のノードから参照される可能性がある．したがって,反復文字列階層グラフは木構造ではなく，グラフになる．また，参照先のノードに対応する文字列は必ず参照元のノードに対応する文字列より短くなるので，反復文字列階層グラフには巡回する部分は存在せず，非巡回グラフとなる．

例では簡単のために，ノードへの参照をノードに対応する文字列から延びる矢印で表現し，各ノードが対応する文字列を図示したが，各ノードに保存されている順番で参照をトラバースすることによって，対応する文字列を復元できるので，各ノードに対応する文字列そのものの情報を実際に各ノードが含む必要はない．

文字列『解析する』に対応するノード(図 3)を例にとって文字列の復元の様子を説明する．

『解析する』に対応するノードには，二つのノードへの参照が含まれている．参照の順序にしたがって，まず 1 つ目の参照の参照先である『解析』ノードのトラバースを行なう.『解析』ノードでは,『解』という文字と,『析』という文字への参照が格納されている．これを順にとりだすことで,『解析』という文字列が得られる.『解析』ノードにはこれ以上参照がないので,『解析』ノードは『解析』という文字列に対応していることがわかり，したがって,『解析する』ノードが持つ最初の参照先ノードは,『解析』という文字列に対応していることがわかる．

以下同様に二つ目の参照であるノードをトラバースすると,二つ目の参照先のノードに対応する文字列,『する』が得られる．参照は二つしかないのでトラバースは終了し，元のノードは『解析する』という文字列に対応することがわかる．

図 3の例では,ルートノードから『形態素解析』,『解析する』に対応するノードにそれぞれ複数の参照が存在する．これは，ルートノードが対応する文字列，つまり解析する文字列全体の中に，それぞれの文字列(『形態素解析』,『解析する』)が複数回出現していることを示している．

反復文字列階層グラフは，複数回出現した二文字以上の文字列をノードとしてくくり出す処理を再帰的に

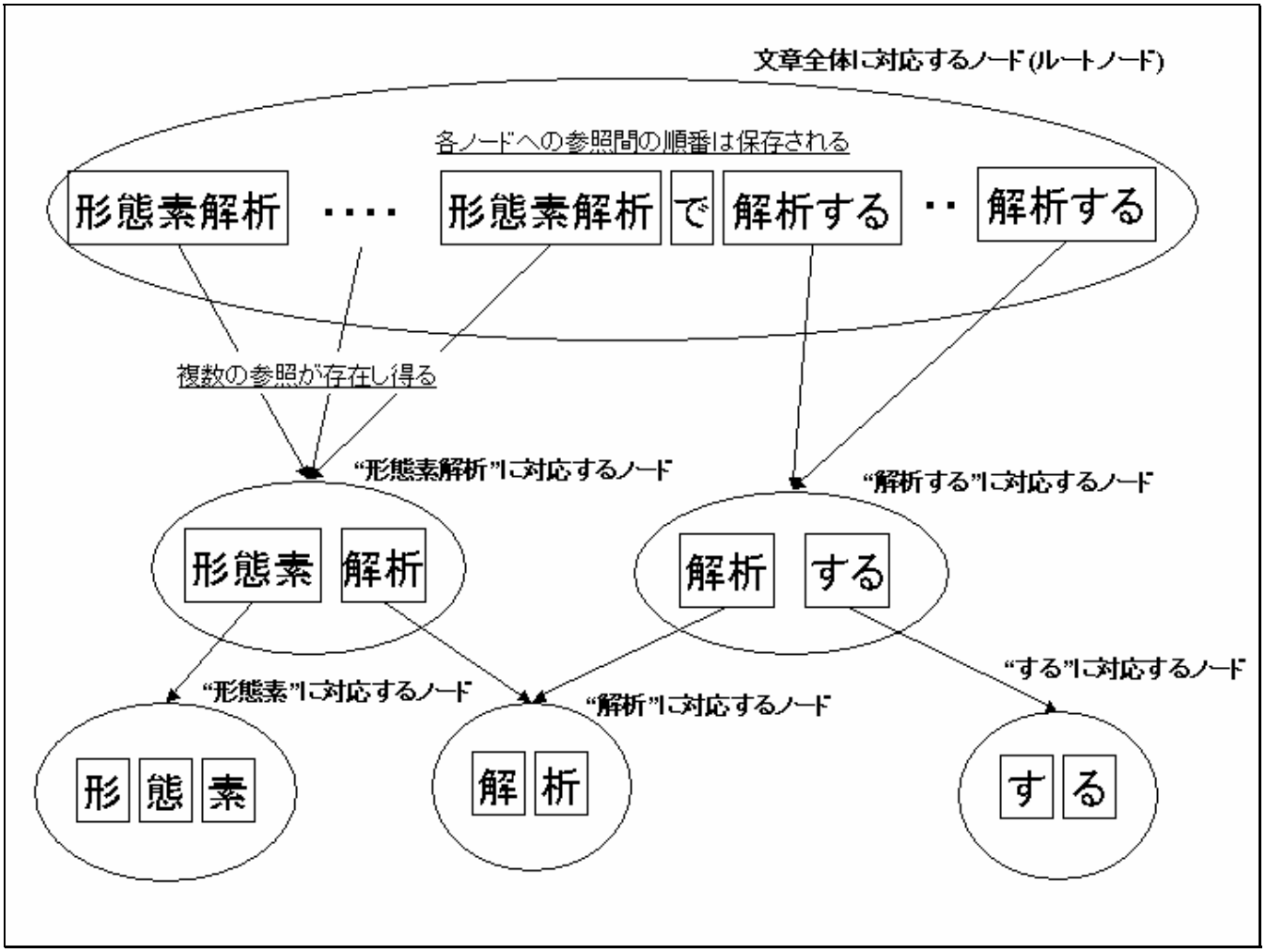

図 3 反復文字列階層グラフの構造

行なうことで生成される．従って，反復文字列階層グラフには，解析する文字列全体の再帰的な反復構造が反映される．本手法では，再帰的な反復構造を利用することで,評価すべきノードの数を削減するとともに，階層的な参照数を用いて，キーワードの評価を行っている．

おおまかなグラフの性質としては，文章全体を表現する文字列に対応するノードから，共通して出現する文字列に対応するノード(図 3では『形態素解析』『解析する』に対応するノード)へ枝が延び，さらにその中で共有して利用されている文字列に対応するノードへ枝が延びることになる．5 節で行った評価では，ルートノードから延びる枝の数は 2737 本,ルートノードを含め 794 個のノードが生成された．

3.2. 反復文字列階層グラフへの変換方法

反復文字列階層グラフは，以下のような制約を保つように構成される．

1) 二回以上繰返した文字列は，必ずいずれかのノードに対応する文字列に含まれる
2) 同じ並び順の文字列の出現はひとつのノードにまとめられる
3) 各ノードの参照関係が，ノードが対応する文字列の包含関係を現わす

反復文字列階層グラフへの変換には，データ圧縮アルゴリズムである，SEQUITUR[2]アルゴリズムを利用した．SEQUITUR は，圧縮するデータの規則性を階層的に抽出し，圧縮するデータ列のみを生成する文脈自由文法を構成する．

抽出する文脈自由文法は，上記の 2)，3)の制約を満たす. SEQUITUR の生成する規則をノードと読みかえ，文脈自由文法を非巡回グラフとして文法を捉えると，SEQUITUR の生成する文脈自由文法はさらに以下の制約を満たす．

- ルートノードをのぞいた各ノードには，そのノードを指す参照が二つ以上存在する

この制約により，SEQUITUR アルゴリズムでは，元

の文章を復元する文法の中で比較的小さな文法が抽出されることになる．我々は，規則性を階層的に抽出する SEQUITUR アルゴリズムを文字列に適用することで，反復文字列階層グラフを生成している．

しかしながら，SEQUITUR アルゴリズムでは反復文字列階層グラフに要求される 1)の制約が満たされないケースがある．

そこで，我々は，まず文字列を SEQUITUR アルゴリズムで解析し上記の制約を満たす文法を得たのち，1)の制約を満たさない部分を探し，我々が欲する性質を満たすように構造を再構成する手段をとった．

具体的には，まず，「文字列中には 2 回以上出現しているが，SEQUITUR アルゴリズムでの構造抽出ではルールに含まれない文字列部分」を探す．次に，発見された部分に関して新規ノードを追加し，グラフを修正する．これによって，1)の制約を満たす反復文字列階層グラフが得られる．

SEQUITUR アルゴリズムは文章の前方から文字列のマッチングを進めていくので，文章中のより先頭の位置で，二回繰返し出現した文字列が優先的に規則にまとめられていく．この結果，我々のキーワード候補も文章の先頭のほうで繰返し出現した文字列が優先的にノードとしてまとめられる．

4. 反復文字列階層グラフからの抽出

本節では，反復文字列階層グラフの各ノードに対応する文字列を評価し，キーワードを抽出する手法を説明する．本手法で利用している評価法は，グラフ上の参照数を用いて評価する方法，文字列の結合度を用いて評価する方法の二種類である．

以下，それぞれの評価方法の説明を行なう．

4.1. グラフ上の参照数を用いた評価

キーワードはそれ自体独立して意味を持つ．従って，文字列の並びという観点で見ると，キーワードの前後に出現する文字列の種類が多くなる.いわば，「隣りあうノードのバリエーションの多いノードがよりキーワードらしい」ということができ，この特徴をキーワードの評価に用いた．

反復文字列階層グラフのノードは，そのノードが対応する文字列が出現する他のノードから参照される．例えば図 3では，『形態素解析』ノードが 3 回，『解析する』ノードが 2 回参照されており，『解析する』および『形態素解析』に対応するノードから『解析』に対応するノードが参照されている．

『解析』という文字列の出現は，全体としては 5 回ある.しかしながら反復文字列階層グラフでは,『解析』の後に『する』が出現するパターンが『解析する』に対応するノードにまとめられ,『解析』の前に『形態素』が出現するパターンが『形態素解析』に対応するノードにまとめられているため，『解析』への参照数が 3 つ減り，2 になっている．つまり，反復文字列階層グラフでは，そのノードに隣接するノードのバリエーションの数がそのまま参照の数となって表現されている．従って，参照数をキーワード評価の尺度に用いることで，キーワードの前後に出現する文字列の種類と同等の尺度が得られる．

単純に繰り返し文字列だけを抽出しキーワード候補にし，それぞれの文字列の頻度を調べるだけでは，このような文字列が使われるバリエーションの情報を得ることができない．

例えば,『形態素解析』への参照が非常に多く,『解析』への参照が少ない場合には,『解析』より『形態素解析』のほうがキーワードとしてふさわしい．しかしながら，グラフ構造を持たず，キーワード候補だけで『解析』及び『形態素解析』の出現頻度を比較すると，『解析』のほうの頻度が大きくなる．出現頻度が高いものほど重要，という尺度を用いると,『形態素解析』より『解析』の評価のほうが高い評価になる．

一方，階層グラフでは,『形態素解析』中に『解析』が含まれているという情報に加え,『解析』は『解析する』と『形態素解析』の二つの文字列に出現し，参照数が 2 であること，また『形態素解析』は参照数が 3 であるという情報が得られるため，相対的に『解析』という文字列より『形態素解析』という文字列のほうがキーワードとして評価が高いという判断を下すことができる．

4.2. 文字列の結合度を用いた評価

その文字列の「キーワードらしさ」を，キーワード候補である文字列そのものから計算する尺度を以下のように定めた．

キーワードと見なせる文字列は，文書全体の文字列中に繰り返し出現している．したがって，キーワードを個々の文字の繋がりと捉えると，キーワードを構成する文字の繋がりは，キーワードを構成しない文字間の繋がりより相対的に強いはずである．したがって，文字間の繋がりの強さでキーワードらしさを評価することができる．

文字間の繋がりの強さを測るために，結合度という概念を利用した．結合度は解析対象の文書から得られる連接頻度から算出できる．

連接頻度は解析対象の文書に含まれる全ての文字に対して，その文字の前後にはどの文字がどのような割合で出現するかを示した頻度である．ある文字とその前後に現われる文字に対して定義され，文字 B の出現のうちある文字 A の直後に出現する回数をある文字 A の出現総数で割ったものが，A の直後に B が連接する頻度となる.直前への連接頻度も同様に定義される．

例えば,『あ』という文字の出現総数が 10 であり,『あ』の直後に出現する『い』の総数が 5 であれば,『あ』の直後に『い』が出現する連接頻度は 0.5 となる.

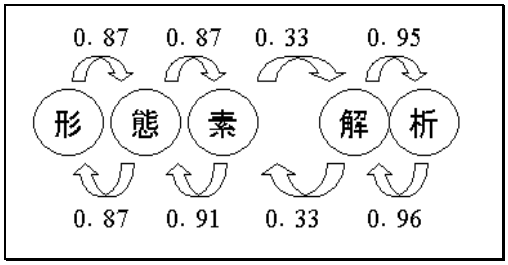

図 4 『形態素解析』の連接頻度

ある文字列の結合度は，その文字列を構成する各文字の連接頻度を平均したものである.図 4のように『形態素解析』という文字列に関する連接頻度が与えられたとする．図中の上段の数字が直後の文字への連接頻度，下段が直前の文字への連接頻度である．結合度を算出するには,『形』の直後に『態』が出現する頻度,『態』の直前に『形』が出現する頻度,『態』の直後に『素』が出現する頻度，というように文字列間の連接頻度を順に求め，足し合わせたものの平均をとる．これが『形態素解析』という文字列の結合度となる．したがって『形態素解析』の結合度は (0.87 + 0.91 + 0.33 + 0.96 + 0.87 + 0.87 + 0.33 + 0.95)/8 となり，0.74 となる.

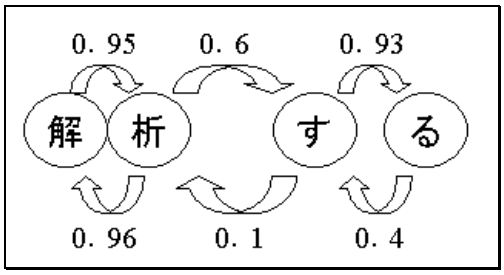

図 5 『解析する』の連接頻度

ある文字列の各々の文字が，その順番で文書に出現する頻度が高ければ高いほど，結合度は高くなる．結合度の高い文字列ほど，直感的には単語としてまとまっている．図 4図 5では，結合度が高く，より単語らしい文字ほどより近くなるように図示を行った.

『解析』という単語を含む文書を例にとると，文書中には『解析する』,『解析した』,『解析後』というように,『解析』+ある文字という形の文字列が多く出現する.『解析』という文字列と『解析する』という文字列の結合度を比較すると,『解析』の『析』の後にくる文字の種類が多くなるため,『析』と『す』の間では,連接頻度が小さくなる(図 5). したがって,『解析』の結合度 0.96((0.95+0.96)/2)より,『解析する』の結合度 0.49((0.95+0.96+0.6+0.1+0.93+0.4)/6)のほうが小さくなり,『解析する』よりも『解析』のほうがより単語らしいといえる.

この評価法は，専門用語を抽出する手法である『出現頻度と連接頻度に基づく専門用語抽出』[3]の一部で用いられている評価法である．[3]は形態素解析をベースにしたキーワード抽出法であり，形態素解析での単語抽出時に細分化されてしまった複合単語を，結合度を利用して再構成することで，形態素解析による細分化された単語抽出の欠点をおぎなっている．一方，本手法では，形態素解析後の文字列ではなく，反復文字列階層グラフのノードに対応する文字列を対象としているため，形態素解析を行なう必要はない.

4.3. キーワード抽出

階層グラフの各ノードに対して，以上二つの評価方法を利用してそれぞれに得点を算出する．結合度による評価で算出される点数は，0 から 1 の間の数値をとり，参照数で算出される点数は，2 以上の自然数となる.

これら二つの得点を最終的な評価点に反映させる計算式は複数考えられるが，今回は参照数の log をとったものと，結合度の積を最終的な評価点として採用した．積を採用したのは，最終的な得点に双方の値を反映させるためであり，log を採用したのは，値域の幅の大きい参照数の影響を抑えるためである.

5. キーワード抽出実験

5.1. 実験の詳細

比較のために，本手法を用いたキーワード抽出システムと，形態素解析をベースとしたキーワード抽出システムをそれぞれ実装し，本稿の予備原稿をサンプルとして抽出実験を行った.

形態素解析ベースでのキーワード抽出は，まず MeCab (和布蕪)[1]によって形態素解析を行い，名詞または未知語の品詞を持つと判定された単語をキーワードとして抽出する．また，複合単語を補足するために，連続した未知語及び名詞はそれぞれの連続した組合せをキーワードとして加えることにした．例えば,『反復階層グラフ』という文字列が『反復』,『階層』,『グラフ』の三つの名詞に分割された場合,『反復階層グラフ』,『反復階層』,『階層グラフ』,『反復』,『階層』,『グラフ』の 6 つの単語がキーワードとして抽出される． このように抽出されたキーワード群に対して，それぞれのキーワードの出現頻度を調べ，大きいものから順位をわりふった.表 1表 2にそれぞれ上位 30 位以内までをリストアップしたものを示す.

[1] 辞書にはipadic version 2.6.1を用いた

順位	結合度	参照数	キーワード
1	0.912162	24	文字列
2	0.965116	19	反復文字列階層グラフ
3	1	17	評価
4	0.951807	18	抽出
5	1	13	参照
6	0.962701	14	解析
7	0.899061	16	文字
8	0.715802	32	する
9	0.914286	13	結合度
10	0.916667	11	を行なう
11	0.761905	16	という
12	0.954185	9	グラフ
13	0.641901	23	から
14	0.655002	19	キーワード
15	0.70811	13	形態素解析
16	1	6	候補
17	1	6	検索
18	1	6	情報
19	0.958333	6	処理
20	0.880952	7	利用
21	0.9375	6	データ
22	0.653846	13	解析
23	0.761167	9	ドキュメント
24	0.857143	7	を利用
25	0.899235	6	階層グラフ
26	0.981481	5	登場
27	0.765766	7	手法
28	0.578947	12	として
29	0.731492	7	0
30	0.480198	19	ノード

表 1 本手法での抽出結果

順位	出現頻度	キーワード
1	168	文字
2	144	列
3	143	文字列
4	129	″
5	107	解析
6	105	キーワード
7	97	ノード
8	75	抽出
9	63	.
10	61	グラフ
11	49	階層
12	47	階層グラフ
13	46	形態素
14	45	解析″
15	44	形態素解析
16	42	反復文字列
16	42	反復文字
16	42	反復
19	41	,″
19	41	文字列階層
19	41	列階層グラフ
19	41	反復文字列階層グラフ
19	41	反復文字列階層
19	41	列階層
19	41	文字列階層グラフ
26	39	こと
27	36	度
27	36	評価
29	35	参照
29	35	結合度
29	35	対応
29	35	結合

表 2 形態素解析を用いた抽出結果

5.2. 抽出結果

本手法でのキーワード抽出結果を見ると,『評価』,『解析』,『抽出』といった単純な単語は，文字列の結合度も高く，またいろいろな繋がりの中で出現するため，参照数も高くなっており，上位に抽出される．また，頻出する単語に加えて，本稿での特徴的な複合キーワードである『反復文字列階層グラフ』が抽出できていることがわかる．また，40 位以内ではないが，91 位には，名詞の複合ではないキーワード『対応する文字列』が抽出されている．

双方において,『反復文字列階層グラフ』,『文字列』,『文字』 や 『形態素解析』,『解析』等，キーワード間に包含関係のある組が同時に抽出されているが，抽出のされかたには大きな差がある．

形態素解析ベースの抽出では，連続する名詞，未知語の組合せをそれぞれ単語と認識させているため，包含関係にある単語もリストアップされているものの，連続した名詞，未知語のほぼ全ての組合せが同程度の

順位で登場している.例えば,『反復文字列階層グラフ』という単語を構成する名詞の組合せが，キーワードらしくないものを含め，16 位，19 位に集中していることがわかる．形態素解析をベースとして利用した場合でも，連続した名詞からなる複合単語の組合せの中からキーワードとして適切な文字列を抽出するには出現頻度以外の評価軸が必要になるといえる.

本手法では，階層グラフ中の参照数というグラフのトポロジを用いた尺度を利用しており，連続した名詞からなる複合単語の組合せの中から比較的独立して使われる部分がキーワードとして抽出されている．例えば，形態素解析ベースのものでは『列階層』や『反復文字』といった名詞を複合したキーワードではあるものの，本稿では独立しては現われない文字列が抽出されているが，本手法ではこれらの文字列をキーワードとして抽出していない.

形態素解析をベースにしたキーワード抽出では，キーワードの区切りがおかしいものはあまり抽出されてこない．これは，品詞情報を利用して，助詞等をあらかじめ候補から外しているためである．逆にこれらの品詞をあらかじめ候補から外しておかないと，いわゆる「てにをは」が高い順位で抽出されることになる．ただ，このフィルタリングも万能ではなく，たとえば『,』や『.』のような記号等も未知語として判定される場合があるため,これらも抽出されてしまっている.

一方，本手法では,『する』,『から』,『という』という結合度も高く，いろいろな使われ方をする助詞や接続詞や『を利用』のようにたまたま文中で多く使われた名詞以外の語句など，キーワードの区切りとしてはおかしいものも抽出されている．本手法の場合，品詞情報を一切利用していないので，明らかに助詞や接続詞とわかるノイズも多く抽出されている．しかしその反面，形態素解析の品詞判定の誤りに影響されることもない．品詞判定の誤りの影響を抑えながら，ノイズを抽出しないような評価軸を加え，キーワード抽出の精度を改善するのは今後の課題である.

6. まとめ

本稿では,与えられた文書から自動的にキーワードを抽出する手法を説明した.

本手法では,文書そのものから求められる文字間の結合度及び繰り返しによる階層構造を利用してキーワードの抽出を行なう．形態素解析等の前処理は必要ではなく，形態素解析時に必要とされる辞書および統計データを必要としない．したがって，専門分野に応じた辞書の修正，言葉の移りかわりによる辞書のアップデート等のコストをかけずに単一の文書からキーワード抽出を行なうことができる.

ただ,本手法ではキーワードらしくないものも抽出されており，完全に適正なキーワードのみに高い評価を与えることができるとはいいがたい．改善の方針としては，辞書や統計データを利用しないという前提で研究を進めている現状では，キーワードを抽出する評価軸に工夫を加えることによって，キーワード抽出の精度を上げることが考えられる．将来的には，形態素解析の結果を利用したキーワード評価を行なうこと，例えば形態素解析の結果を利用して，不要な語を候補から外す処理を加えるといったことも考えられる.

文　献

[1] 永田昌明,“形態素解析”,言語と心理の統計,pp.62-73,岩波書店,2003.

[2] Nevill-Manning,C.G,Witten.I.H,“Identifying Hierarchical Structure in Sequences:A linear-time algorithm”, Journal of Artificial Intelligence Research,pp.67-82,1997.

[3] 中川裕志,森辰則,湯本紘彰,“出現頻度と連接頻度に基づく専門用語抽出”,自然言語処理,Vol.10 No.1,pp.27-45,2003.

[4] 梅村恭司,“未踏テキスト情報中のキーワードの抽出システム開発”,未踏ソフトウェア創造事業,2000.