# 単語群に基づく文書検索システム

片山聡一郎† 遠山 元道††

† †† 慶應義塾大学理工学部情報工学科 〒 223–8522 神奈川県横浜市港北区日吉 3–14–1
E-mail: †katasou@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

**あらまし** 従来の全文検索手法では、検索を行う際、条件をより絞り込むためには複数の単語を AND または OR で結合する事によって自分の得たい情報を検索していた。しかしながら従来の方法では、ユーザの意思を十分に反映する事ができているとは言いがたく、曖昧な条件を検索するというユーザの要求を満たす事ができなかった。本論文では、自分が検索したい概念を単語群という形で定義し、それを基に「できるだけ多く」、「ひとつは」といった概念による検索・スコアリングを可能とする、単語群に基づく検索を提案する。また、その実現のためにフォルダシステムを提案し、これにより、従来できなかった柔軟な検索の指定をする事ができるようになり、ユーザの曖昧な条件による検索を可能とし、今までよりもユーザの意思を反映した結果を出力する事ができる検索を行う事が可能となる。
**キーワード** 全文検索, 曖昧検索, スコアリング, 単語群

# Full-text Document Retrieval Using Word Group

Souichirou KATAYAMA† and Motomichi TOYAMA††

† ††Department of Information and Computer Science, Faculty of Science and Technology,
Keio University
Hiyoshi3–14–1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223–8522 Japan
E-mail: †katasou@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

**Abstract** If you use existing full-text search technique, you combine two or more words by AND or OR in order to narrow down. However, by the conventional system, it is hard to say that a user's intention can fully be reflected. Demand of searching the ambiguous conditions (e.g. "containing keywords as many as possible") cannot be filled. In this system, you define the concept which you want to search by using the form of word group and use it. Then the system makes power set of these words,searches them, calculates score and unifies these results. By using them,specification of flexible condition (e.g."containing as many as possible","containing at least one") is attained, and it will be able to output the result in which a user's intention is reflected.
**Key words** full-text search,approximate pattern matching,word group,score

## 1. ま え が き

利用者の必要な情報を即座に適切に検索するための手法として全文検索がある。全文検索は、文書内の全ての単語を検索対象とした検索手法であり、現在様々なところで利用されている。代表的なものには google や Namazu といったものがある [?], [?] 。これらのシステムによりユーザは大量の情報の中から自分の必要な情報を取得するという事が可能となった。

しかしながら、ユーザの複雑な要求に対しては、ユーザの意図を十分に反映した検索結果を出す事ができているとは言いがたく、ユーザの複雑な意思を反映できるような検索システムが望まれている。

本研究では、全文検索における、ユーザの曖昧な検索条件に着目し、曖昧な要求を入力及び結果に反映させる事ができるために、検索における新たな概念を提案する。そして、その概念を用いた全文検索システムである、単語群に基づく文書検索システムを提案し、実装する。本システムにより、ユーザは今までよりも自分の意思を検索条件に活かしやすくなり、また、その結果として、ユーザの複雑な要求に対しても、今までよりも適切な検索結果を返す事ができるようになる。

本論文の構成は、??章では全文検索に関する従来の方法及びその問題点をのべ、本システムの必要性を述べる。??章では本研究で提案する概念について説明する。??章ではその概念を利用したシステムの具体的な実装方法について述べる。??章では本システムによる実行結果を示す。??章で本研究の評価・検討を行い最後に??章で結論を述べる。

## 2. 全文検索

### 2.1 全文検索とは

文書内のすべての単語を検索対象とする検索を全文検索という。

全文検索においては、検索条件に該当する文書が複数存在した場合には、通常、システムそれぞれのスコアリングアルゴリズムに基づき検出された文書を順序づけ、スコアの高いものから優先して表示する事により、ユーザの要望により近いものから提示される。

なお、以下の記述では言葉による誤解を避けるため、検索を行うために入力された語句の事を検索語句と書いていく事にし、キーワード、文中の単語・語句とは区別する事にする [?]。

### 2.2 スコアリングについて

スコアリング方法としては、全文検索においては、単語の重要度の評価として TF-IDF 法と呼ばれる手法が一般に広く用いられている。本研究でもこの考え方をスコアの算出において利用する。

tf(term frequency) は、ある文書 d における索引語 t の出現回数 tf(d,t) と定義される。

また idf(inverse document frequency) は検索の対象となっている全ての文書の数 N と、索引語 t が一回以上生起する文書の数 df(t) によって

$$idf(t) = \log \frac{N}{df(t)}$$

と定義される。本論文ではこの log の底を 2 とする。この tf と idf の値を掛け合わせた値が tf-idf の値である。

なお、TF-IDF 法には問題があり、検索対象の文書が繰り返しの少ない短文である場合，tf の結果がどれも同じになってしまい、idf の単語の出現文書数だけで単語重要度が決まってしまう、という問題がある。

### 2.3 従来の検索方法の問題点とその原因・解決方法

この様に、従来の検索手法を用いる事によって、様々な条件を検索する事ができる。しかしながら、実際に全文検索を利用してみると、複数の語句を用いて検索した時に検索の条件がゆるすぎるために該当する文書が膨大に出てきてしまって、ユーザが希望する文書をその中から見つけ出す事ができなかったり、逆に、検索の条件がきつすぎるために該当する文書が 1 件も存在しなくなってしまったりと、ユーザが要求する文書になかなか辿り着けないといった事態に陥いってしまう事がしばしばあった。

その原因として、ユーザが検索したい要望を反映させるための方法が、根本的には複数の単語を and,or,not といった、Boolean（ブーリアン）記号で結合する事しかなかったために、「できるだけ」といった曖昧な条件を検索するという要求を満たす事ができなかった事が挙げられる。and,or,not によって結合された検索式では各単語それぞれについて、文書に含まれているか含まれていないかという、事しか反映させる事ができない。そのため、ユーザの曖昧な要求に対して、十分な結果を出力する事ができなかったと考えられる。

従って、ユーザの曖昧な要求を反映できる全文検索システムを作り出すためには、含む・含まないだけではなく、「できるだけ多く」含むといった曖昧さのある概念を導入する必要があると言える。その様な演算子の導入により、ユーザの複雑な要求を検索式及び結果に反映させる事が可能となる。

また別の原因として、従来の手法ではユーザの「色々な」といった概念に対応した入力フォームが無いため、ユーザがどの様なものを指して色々なといっているか、というのを入力に反映させる事ができず、検索システム側もユーザの要望を適切に反映した結果を出力ができないという事が挙げられる。

従って、ユーザの「色々な」という概念に対応するために、ユーザの色々がどの様なものを指しているか、わかる様な入力フォームが必要であると言える。

これらの問題を解決する方法として、本論文では単語群に基づく検索を提案し、次の章以降でその説明を行っていく。

### 2.4 関連研究

全文検索に関して、ユーザの希望する情報を今までよりも労力をかけず、かつ適切なものが検出されるシステムの研究の一つにシソーラスを利用して意味的に近いものを類推して探すもの、が挙げられる。しかしながら、この検索はあらかじめ類似する言葉の専用辞書を用意する必要がある。また、検索の演算処理自体は従来のものを使用している。本研究では、その辞書にあたる部分をユーザが手軽に定義する事ができるため、あらかじめ辞書を用意する必要が無いだけでなく、ユーザが頭に思い描いている単語群をダイレクトに検索に反映させる事ができる。また、入力された検索単語群に対してより適切な結果が出るように、独自のスコアリングを用いている点も異なる。

ユーザの要求により近づける事を目的とした検索システムとして、対話的に検索を繰り返し行い結果をユーザの要求に近づけていく方法があるが、これはユーザが希望する検索結果が得られなかった場合に、何度も検索する部分の支援であり、本方式は 1 回の検索でいかにユーザの要求する検索結果に近づけられるかという事に着目しており、これらの検索手法とは異なる。

## 3. 単語群検索の考え方

前章で挙げた問題点を解決するため、本研究では単語群による検索という手法を提案する。この検索により、従来の検索では不可能であった、ユーザの曖昧な条件を反映した検索が可能となる。まず始めにいくつか定義をする。

### 3.1 スコアリング関数

関数 $\phi$（）は変数を２つとる、ある文書におけるスコアリング関数と定義する。今、$\alpha$ をある 1 つの文書とし、X を検索式とした時、

$\phi$（$\alpha$,X）

は、文書 $\alpha$ に関する、検索式 X に対するスコアを返すものとする。この際、スコアは 1 元的で連続的な値とする。従って、そこには順序が存在し、スコアの値が高いものほど、ユーザの要求を満たしている文書であるとする。また、文書 $\alpha$ 中に検索語

句 x を 1 つも含まない時、

$\phi$（$\alpha$,x）=0

と定義する。

そして、1 つの検索語句に対するスコア、すなわち

$\phi$（$\alpha$,x)

の値は tf-idf の値とする。

スコアリング関数において、検索式の and,or,not は以下の様に定義する。なお、$\Psi$（$\alpha$,X）は文書$\alpha$が、検索式 X を満たす時には 1 を返し、満たさない時には 0 を返す関数と定義する。また、小文字のアルファベットは検索語句 1 語に対応し、大文字のアルファベットは検索式を表すとする。

$\phi$（$\alpha$,a and b)=($\phi$（$\alpha$,a）+$\phi$（$\alpha$,b）)×$\Psi$（$\alpha$,a)×$\Psi$（$\alpha$,b)

$\phi$（$\alpha$,a or b)=($\phi$（$\alpha$,a）+$\phi$（$\alpha$,b）)

$\phi$（$\alpha$,a not b)=$\phi$（$\alpha$,a）×(1-$\Psi$（$\alpha$,b))

### 3.2 曖昧な言葉の定義

次にユーザの曖昧な要求に対してどの様な検索をすれば良いのか、その目的をはっきりさせるために、いくつかの曖昧な言葉（概念）に対して、定義を行う。

「できるだけ多く」という概念を以下のように言葉で定義する。

『検索のために入力された語句を多く含んでいる文書に対して、大きい値が出力される。文書中での検索語句の出現回数も考慮するが、入力された語句を多くの種類含んでいる事を重視して評価する。』

これらの単語群は「できるだけ多く (as much as possible)」の対象である事を、記号では以下の様な関数を用いて定義する事にする。なお、$\mathbf{X}$ は検索語句群を表すものとする。

AP($\mathbf{X}$)

この時、

$\phi$（$\alpha$,AP($\mathbf{X}$))

この結果としてはどの様なものが適切であるか、どの様な演算関数をこの関数に対して当てはめるのが適切か、その条件を挙げると

・検索語句群 $\mathbf{X}$ のうち、一つでも文書$\alpha$に含まれている場合には、スコアとして 0 より大きな値を返す。

・検索語句群 $\mathbf{X}$ を or で結合した場合に比べ、多くの種類の検索語句群を含む程重点的にスコアが高くなる必要がある。

次に、「ひとつは」という概念を以下のように言葉で定義する。

『検索のために入力された語句は一つ以上含んでいる必要があるが、評価する際、入力された語句を多くの種類含んでいる事も考慮するが、文書中での検索語句の出現回数を重視して評価する。』

これらの単語群は「ひとつは (at least)」の対象である事を、記号では以下の様な関数を用いて定義する事にする。

AL($\mathbf{X}$)

この時、

$\phi$（$\alpha$,AL($\mathbf{X}$)) この結果としてはどの様なものが適切であるか、その条件は

・検索語句群 $\mathbf{X}$ のうち、一つでも文書$\alpha$に含まれている場合には、スコアとして 0 より大きな値を返す。

・検索語句群 $\mathbf{X}$ を or で結合した場合に比べ、多くの種類の検索語句群を含んでも、それ程スコアが高くならない必要がある。

以上の AP($\mathbf{X}$) と AL($\mathbf{X}$) の事をまとめて曖昧記号と表現する事にする。なお、本研究では、$\mathbf{X}$ の中には検索語句群のみが入る事とする。

### 3.3 従来手法との検索結果の違い

従来手法と大きく違う点は、ブーリアン記号による検索ではその演算子の対象となる語句はあくまで隣接する 2 個の要素のみを対象とした演算の組み合わせでしか無かったのに対して、曖昧記号による検索では、対象となる語句の数は決まっておらず、3 個以上の単語を対象とする事ができる点が大きく異なる。

この様な概念を導入する事により、今までできなかったどのような検索が可能になるか例を示す。

例えば以下の 3 種類の文書があったとする。数字はその文書に出現する回数を表す。なお、どの語句も検索する文書全体では同じ位の数存在するため、ここでは、単純に出現回数がそのまま、その文書におけるその語句のスコアであると仮定する。

- 文書 A（北海道:20, 東京:20, 沖縄:10）
- 文書 B（北海道:50, 東京:10, 沖縄:0）
- 文書 C（北海道:60, 東京:0, 沖縄:0）

従来の AND 検索、OR 検索によってどの様な検索が可能であるか考えてみると、「北海道 東京 沖縄」の 3 つの検索語句に関して、AND 検索した場合には、文書 A のみ検索の条件に該当し、文書 B,C は該当しないため、検索結果から排除される。ここで、検索語句の出現回数が多い文書 B,C も検索結果に活かしたいと考えた場合、従来の AND 検索 OR 検索だけでは解決する事ができない。

そこで、「できるだけ多く」という視点で「北海道 東京 沖縄」の 3 つの検索語句を検索したとすると、多くの検索語句を含んでいるものを高く評価する一方で、検索語句を 1 種類でも含むものは検索結果の対象となるため、文書 A,B,C 全てが出力され、ランキングとしては文書 A,B,C の順となる検索が可能となる。

また、OR 検索した場合には、文書 A,B,C どれも条件を満たすため、全て検索結果として出力される。そして、単純に出現回数の多い順にランキングされるため、検索結果は検索語句の出現回数の合計が同じである文書 B と C が 1 位であり、その後に文書 A の順で出力される事となる。ここで、検索語句の出現回数が同じ場合には、含んでいる検索語句の種類が多い方をより上位にしたいと考えた場合、従来の AND 検索 OR 検索だけではやはり解決する事ができない。

ここで「ひとつは」という視点で「北海道 東京 沖縄」の 3 つの検索語句を検索すると、全ての文書が出力される。そしてランキングは、より多くの種類の検索語句を含むものという観点では、文書 A,B,C の順となるが、単純に検索語句の出現回数では、文書 B と C が 1 位で,A が 3 位となるため。その両者の重み付けにより最終的なランキングは変わってくる事になる。ここでは検索語句の出現回数を最優先し、その上で含む検索語句の種類を評価するように重み付けしたとすると、検索結果と

しては文書 B,C,A の順に出力させる事ができる。

この様に「できるだけ多く」「ひとつは」という視点から検索する事が可能となれば、今まで検索する事ができなかった、ユーザの微妙なニュアンスを含んだ条件について検索する事が可能となる。

## 4. 実 現 方 法

前章で単語群による演算子の有用性を説明した。それを実際の全文検索システムにどの様に活用するか、そしてどの様に実装するかについて本章で述べたいと思う。

単語群による演算子を実現するための方法としていろいろな方法が考えられるが、本論文では、自分が検索したい概念を単語群という形で定義し入力、その単語群のべき集合を用意し、それぞれを検索・スコアリングをし、それらの結果を統合する方法によって、単語群による演算子を実現する事を提案する。本論文では単語群に対して「フォルダ」というメタファを用い、フォルダに単語群を入れるというイメージにより、「できるだけ」フォルダおよび「ひとつは」フォルダという「できるだけ多く」「ひとつは」といった柔軟な検索に対応できる検索方法を提案する。

なお、べき集合を用いた理由は、２つ以上の要素に関する tf-idf の値をスコアリングに盛り込む事により、より適切な結果を出力できると考えたからである。

以下では、このフォルダの実装の仕方について述べていく。

### 4.1 フォルダの定義

本論文で提案するフォルダについて定義する。

その前に 2 要素以上の検索語句を含んだ条件式に対する idf の値を以下の様に定義する。

idf は検索の対象となっている全ての文書の数 N と、条件式 T を満たす文書の数 df(T) によって

$$idf(T) = \log \frac{N}{df(T)}$$

と定義する。

「できるだけ」フォルダを以下のように定義する。

1. フォルダ内の検索語句を 1 つ以上含む文書の集合を取り出す。
2. 対象となる検索語句を要素とする、べき集合を考える。この際、２つ以上の要素を含むものについてはそれらを and で結合したもの（T）を考える。そしてそれぞれについて idf の値を求める。$idf(T) = \log \frac{N}{df(T)}$
3. 文書 1 つ 1 つに対して、それらのべき集合各々の tf の値を求め、tf-idf の値を求める。この際、2 要素以上の検索語句に対する tf の値は、各要素の tf の値を求め、その中で最小の値を tf の値とする。$tf(d, t_1 and t_2) = min\{tf(d, t_1), tf(d, t_2)\}$
4. 各文書について、べき集合各々の結果を全て足し合わせ、その値をその文書のスコアとする。ただし、べき集合のうち、全く要素の含まれないものについてはスコアは 1 で固定する。

このように定義する事により、「できるだけ」フォルダでは、多くの種類の検索語句を含む文書に対して、重点的に重み付けがなされる事になると考えられる。

「ひとつは」フォルダを以下のように定義する。

1. フォルダ内の検索語句を 1 つ以上含む文書の集合を取り出す。
2. 対象となる検索語句を要素とする、べき集合を考える。この際、２つ以上要素を含むものについてはそれらを or で結合したもの（T）を考える。そしてそれぞれについて idf の値を求める。$idf(T) = \log \frac{N}{df(T)}$
3. 文書 1 つ 1 つに対して、それらのべき集合各々の tf の値を求め、tf-idf の値を求める。この際、2 要素以上の検索語句に対する tf の値は、各要素の tf の値を求め、それらの合計の値を tf の値とする。$tf(d, t_1 or t_2) = tf(d, t_1) + tf(d, t_2)$
4. 各文書について、べき集合各々の結果を全て足し合わせ、その値をその文書のスコアとする。ただし、べき集合のうち、全く要素の含まれないものについてはスコアは 0 で固定する。

このように定義する事により、「ひとつは」フォルダでは、検索語句を一種類でも含む文書に対してある程度の重み付けがなされ、複数の種類の検索語句を含んでいても、それ程重点的には重み付けがなされない事が考えられる。

### 4.2 スコアの計算例

検索対象とする文書が全体で 100 あり、文書に現れる単語として A,B,C があるとする。そして以下のような状況を考える。

| | |
|---|---|
| A を含む文書の数 | 60 |
| B を含む文書の数 | 50 |
| C を含む文書の数 | 40 |
| A と B を含む文書の数 | 30 |
| A と C を含む文書の数 | 20 |
| B と C を含む文書の数 | 10 |
| A と B と C を含む文書の数 | 5 |

この時、A が５回、B が３回、C が０回出現する文書について、「できるだけ」フォルダ、「ひとつは」フォルダによってスコアをそれぞれ算出してみると、

| 「できるだけ」 | | 「ひとつは」 | |
|---|---|---|---|
| 要素 | スコア | 要素 | スコア |
| {A,B,C} | - | {A,B,C} | 8 × 0.07=0.56 |
| {A,B} | 3 × 1.7=5.1 | {A,B} | 8 × 0.32=2.56 |
| {A,C} | - | {A,C} | 5 × 0.32=1.6 |
| {A} | 5 × 0.7=3.5 | {A} | 5 × 0.73=3.65 |
| {B,C} | - | {B,C} | 3 × 0.32=0.96 |
| {B} | 3 × 1.0=3 | {B} | 3 × 1.0=3 |
| {C} | - | {C} | - |
| {} | 1 | {} | 0 |
| 合計 | 12 | 合計 | 12 |

この計算過程を見てみてみると、「できるだけ」フォルダで

は含まれている検索語句の種類が多いものに対して重点的に重み付けがなされる事、「ひとつは」フォルダでは、検索語句を一種類でも含む文書に対してある程度の重み付けがなされるため、結果として複数の種類の検索語句を含んでいても、それ程重点的には重み付けがなされない事が確認できる。

### 4.3 スコアの正規化について

AND,OR 検索によるスコアとフォルダによる検索のスコアでは、フォルダによるスコアが非常に大きくなってしまいうため、単純に AND,OR 検索とフォルダによる検索を組み合わせるのでは、検索結果にフォルダによって算出された結果だけが大きく影響してしまう事になる。そのため、「できるだけ」フォルダ及び「ひとつは」フォルダによる結果は以下の様に補正した。なお、補正後の「できるだけ」フォルダ及び「ひとつは」フォルダをそれぞれ $\phi'(\alpha,\mathrm{AP}(\mathbf{X}))$, $\phi'(\alpha,\mathrm{AL}(\mathbf{X}))$ と書く事にする。

$$\phi'(\alpha,\mathrm{AP}(\mathbf{X}))=\phi\ (\alpha,\mathrm{AP}(\mathbf{X}))/2^n$$
$$\phi'(\alpha,\mathrm{AL}(\mathbf{X}))=\phi\ (\alpha,\mathrm{AL}(\mathbf{X}))/2^n$$

ここで、n はフォルダの中に入っている検索語句の数である。この補正式は、単語群検索をフォルダを使って計算する際、べき集合を用いているため、最大でその回数分のスコアが足されると考えられ、スコアの最大値が AND,OR 検索したものと同じ位になるようにという考えのもと、上記の式とした。この部分はより適切な正規化の方法を研究する必要があるといえる。

### 4.4 フォルダシステムの表記法

「できるだけ」フォルダおよび「ひとつは」フォルダを以下のように表記すると定める。

- 「できるだけ」フォルダ

「できるだけ」フォルダに入れたい検索語句群を〈〉で挟む事によって定義する。

例： < A B C >

- 「ひとつは」フォルダ

「ひとつは」フォルダに入れたい検索語句群を [] で挟む事によって定義する。

例： [ A B C D ]

フォルダシステムを利用した検索として以下のような検索が可能となる。

例：[ 牛肉 豚肉 鶏肉 馬肉 羊肉 さば カツ ] and 〈 にんじん じゃがいも たまねぎ なす ピーマン 豆 〉 and カレー and レシピ

ユーザがカレーを作りたいと思った時に、野菜がいろいろな種類入っていて、とにかく何か 1 つは肉が入っているカレーのレシピを見てみたいと思った場合、まず、野菜、肉といった場合にどの様なものが自分は思い浮かぶか、挙げてみた後に、上記の様に検索する事により（この場合、ユーザは野菜として「にんじん じゃがいも たまねぎ なす ピーマン 豆」が思い浮かび、肉として「牛肉 豚肉 鶏肉 馬肉 羊肉 さば カツ」が思い浮かんだ)、ユーザの要求により近いレシピから表示される事となる。

### 4.5 フォルダシステムの実現方法

本システムを実装するにあたり、日本語全文検索として広く利用されている Namazu を利用し、そこに関数を組み込む形で「できるだけ」フォルダと「ひとつは」フォルダの機能を実装した。Namazu はインデックス検索を利用しているため、あらかじめ作成されたインデックスに対して、スコアリングの計算・検索を行う事になる。
以下で実装のやり方を説明する。

検索式を左側から処理する過程で、〈 があった時点で「できるだけ」フォルダに関する関数が呼び出され処理が行われる。また、[ があった時点で「ひとつは」フォルダに関する関数が呼び出され処理が行われる。

- 「できるだけ」フォルダにおける処理のプロセス

1. 〉が現れるまで、入力された検索語句を認識していき、〈 と 〉に挟まれた語句を「できるだけ」フォルダの検索語句の対象とする。
2. 入力された検索語句を 1 つ以上含む文書の集合を作る。
3. べき集合それぞれの idf の値を求める関数を呼び出す。
4. 関数を呼び出し、各々の文書に対して、各べき集合に対して tf の値を求め、3. で求めた idf の値を掛け合わせる。そして、それらの結果を足し合わせる。
5. 以上の処理が終わり、検索条件を満たしている文書 id とその文書のスコアをデータとして含んでいる、検索結果を関数の戻り値として返す。

- 「ひとつは」フォルダにおける処理のプロセス

1. ] が現れるまで、入力された検索語句を認識していき、[と] に挟まれた語句を「ひとつは」フォルダの検索語句の対象とする。
2. 入力された検索語句を 1 つ以上含む文書の集合を作る。
3. べき集合それぞれの idf の値を求める関数を呼び出す。
4. 関数を呼び出し、各々の文書に対して、各べき集合に対して tf の値を求め、3. で求めた idf の値を掛け合わせる。そして、それらの結果を足し合わせる。
5. 以上の処理が終わり、検索条件を満たしている文書 id とその文書のスコアをデータとして含んでいる、検索結果を関数の戻り値として返す。

## 5. 実 行 結 果

### 5.1 実 行 環 境

今回実行を行ったマシンのスペックは、
CPU:Celeron 1.7GHz
MEM:DDR-SDRAM 256M × 2
HD :Ultra ATA/100
OS :Vine Linux 2.6r1
である。
使用したソフトウェアのバージョンは Namazu 2.0.12 である。日本語解析ツールには KAKASI を使用した。Perl のバージョンは 5.006001 である。
また、実行は全てプロンプト上で行った。

### 5.2 使用データ

今回の実行では日本情報処理学会のサイトの「電子図書館」にある、論文・雑誌記事を対象とした。

情報処理学会のサイトに対して wget を行い、それによって集まったデータを利用した。その際、目次のページについてはインデックスを作成する前に排除した。アブストラクト全体の容量は 341M であった。それに対して作られたインデックスに関するデータは Total Documents:77,884,Total Keywords:290,357 であった。

### 5.3 評価実験の方法

今回は 2 種類の観点から本システムによる検索結果の評価を行う。1 つ目の方法として、あらかじめ、ユーザの期待する文書を 1 つ決めておき、検索条件によりどの様にその文書のスコア及び順位が変化するかを見る事によって評価を行う。また 2 つ目の方法として、その時の検索結果全体の分布や上位 5 個を見る事により評価を行う。

本論文では、まず始めに 1 つ文書を選び、その文書の中にある単語の 3 語を使った検索結果と、その中からの 4 語と含まれていない 1 語を加えた計 5 語を使った検索結果について、and、「できるだけ」、「ひとつは」、or で結合したそれぞれの場合を示す。

### 5.4 評 価 実 験

ユーザが期待しているページが「日本語教育素材作成のための日本語分析ツールの開発」についてのアブストラクトであったとする。なお、この文書は検索結果では「article004.html」と表示されている。

以下に検索式と検索結果を示す。

- ./namazu "( 教育 日本語 分析 )" .

参考ヒット数: [ 教育: 1224 ] [ 日本語: 1081 ] [ 分析: 2042 ]

検索式にマッチする 5 個の文書が見つかりました。

1. 外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案- (スコア: 591)
2. 理工系留学生のセミナーでの対話にみられるパラフレーズから理解に至る過程のプロトタイプ化の試み (日本語教育) (スコア: 228)
3. article004.html (スコア: 118)
4. article002.html (スコア: 43)
5. article002.html (スコア: 43)

対応するグラフは図 1 である。

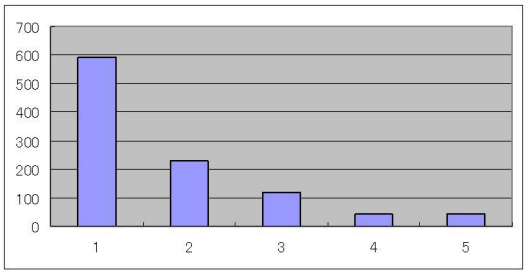


図 1 "( 教育 日本語 分析 )" の検索結果

- ./namazu "⟨ 教育 日本語 分析 ⟩" .

検索式にマッチする 4046 個の文書が見つかりました。

1. 外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案- (スコア: 1,132)
2. 表現教育における効率性と確実性の追及 日本語教育システムの開発を例として (スコア: 424)
3. 日本語→欧語機械翻訳のための日本語の分析 (スコア: 382)
4. 留学生を対象とした CAI 日本語教育 (スコア: 357)
5. 理工系留学生のセミナーでの対話にみられるパラフレーズから理解に至る過程のプロトタイプ化の試み (日本語教育) (スコア: 354)

対応するグラフは図 2 である。なお、「article004.html」のスコアは 198 であった。

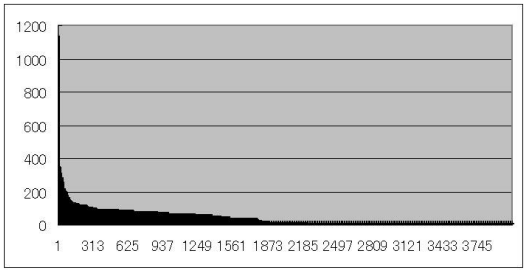


図 2 "⟨ 教育 日本語 分析 ⟩" の検索結果

- ./namazu "[ 教育 日本語 分析 ]" .

検索式にマッチする 4046 個の文書が見つかりました。

1. 外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案- (スコア: 1,484)
2. 表現教育における効率性と確実性の追及 日本語教育システムの開発を例として (スコア: 937)
3. 日本語→欧語機械翻訳のための日本語の分析 (スコア: 871)
4. 日本語構文解析システム「KNP」のハングル化とそれを用いた日本語から韓国語への対照分析 (スコア: 802)
5. 手続き的教材教育のための ITS における学生モデル構築と教育戦略 (スコア: 729)

対応するグラフは図 3 である。なお、「article004.html」のスコアは 293 であった。

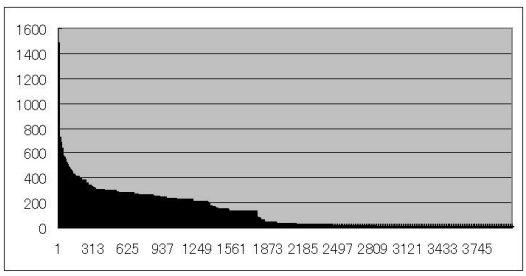


図 3 "[ 教育 日本語 分析 ]" の検索結果

- ./namazu "( 教育 or 日本語 or 分析 )" .

検索式にマッチする 4046 個の文書が見つかりました。
1．外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案-（スコア：591）
2．表現教育における効率性と確実性の追及 日本語教育システムの開発を例として（スコア：370）
3．日本語→欧語機械翻訳のための日本語の分析（スコア：349）
4．日本語構文解析システム「KNP」のハングル化とそれを用いた日本語から韓国語への対照分析（スコア：321）
5．手続き的教材教育のための ITS における学生モデル構築と教育戦略（スコア：289）
対応するグラフは図 4 である。

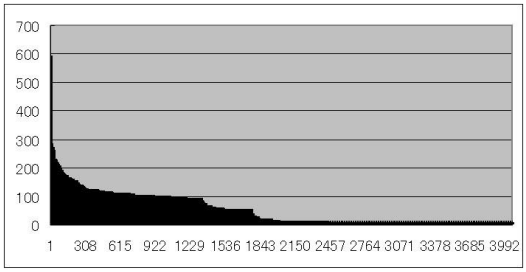


図 4 "( 教育 or 日本語 or 分析 )"の検索結果

- ./namazu "( 教育 日本語 分析 ツール テキスト )" .

参考ヒット数: [ 教育: 1224 ] [ 日本語: 1081 ] [ 分析: 2042 ] [ ツール: 1229 ] [ テキスト: 840 ]
検索式にマッチする文書はありませんでした。

- ./namazu "⟨ 教育 日本語 分析 ツール テキスト ⟩" .

検索式にマッチする 5637 個の文書が見つかりました。
1．外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案-（スコア：579）
2．日本語分析資料およびツールの調査（スコア：209）
3．日本語教育・学習支援システムのテキストデータベースについて（スコア：133）
4．article004.html（スコア：107）
5．表現教育における効率性と確実性の追及 日本語教育システムの開発を例として（スコア：106）
対応するグラフは図 5 である。なお、「article004.html」のスコアは 107 であった。

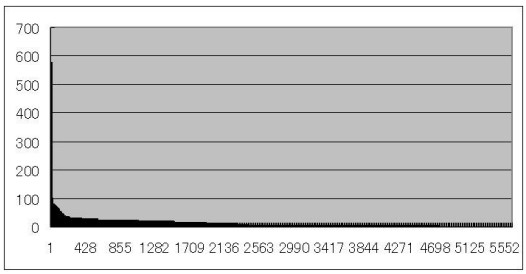


図 5 "⟨ 教育 日本語 分析 ツール テキスト ⟩"の検索結果

- ./namazu "[ 教育 日本語 分析 ツール テキスト ]" .

検索式にマッチする 5637 個の文書が見つかりました。
1．外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案-（スコア：1,590）
2．表現教育における効率性と確実性の追及 日本語教育システムの開発を例として（スコア：845）
3．日本語→欧語機械翻訳のための日本語の分析（スコア：787）
4．日本語分析資料およびツールの調査（スコア：772）
5．日本語構文解析システム「KNP」のハングル化とそれを用いた日本語から韓国語への対照分析（スコア：723）
対応するグラフは図 6 である。なお、「article004.html」のスコアは 324 であった。

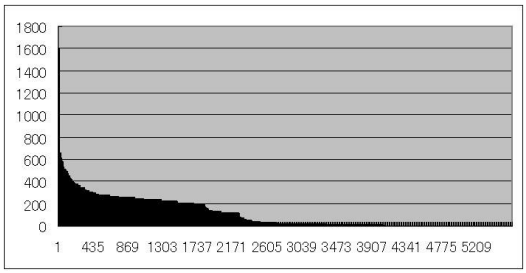


図 6 "[ 教育 日本語 分析 ツール テキスト ]"の検索結果

- ./namazu "( 教育 or 日本語 or 分析 or ツール or テキスト )" .

検索式にマッチする 5637 個の文書が見つかりました。
1．外国人のための日本語教育教材の問題点について-日本語分析ツールを応用した教育素材の提案-（スコア：699）
2．表現教育における効率性と確実性の追及 日本語教育システムの開発を例として（スコア：370）
3．日本語→欧語機械翻訳のための日本語の分析（スコア：349）
4．日本語分析資料およびツールの調査（スコア：346）
5．日本語構文解析システム「KNP」のハングル化とそれを用いた日本語から韓国語への対照分析（スコア：321）
対応するグラフは図 7 である。

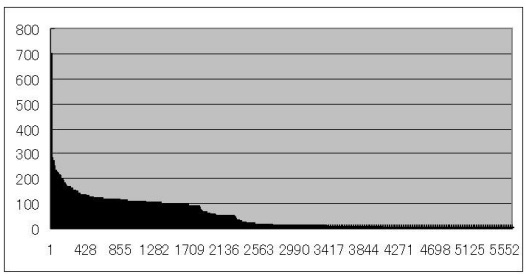


図 7 "( 教育 or 日本語 or 分析 or ツール or テキスト )"の検索結果

## 6. 評　　価

対象とする文書をのについて見てみると、どの検索においても、複数の検索語句による絞り込みにより上位の方に来ている

事が確認できる。なお、2 つのフォルダを比較すると、検索語句を増やした時の順位の上がり方は、「できるだけ」の方が順位の変化の仕方が急であった。これは「できるだけ」フォルダは検索語句の種類をどれだけ多く含んでいるかが重要であるのため、絞込みを行う事による効果がより現れていると言える。

また、対象とした文書には含まれていない語句を含む方の検索結果では、対象の文書は and 検索を行った場合には、出力の対象から外れてしまっていた。しかしながら、フォルダシステムの場合には、検索語句のうち欠けているものがあったとしても、その事を配慮した上で、検索結果の候補からは外さないため、どちらのフォルダを利用した場合も検索結果として出てきている。

次に全体の分布について見てみる。まず始めに、正規化についてであるが、実際のところあまり上手く機能していなかった。その原因としては、荒く $2^n$ で割った事に無理があったと思われる。これ以下の議論では、相対的な値、分布、順位といったものについて着目して議論を行う。

それぞれの検索方法から出力された上位 5 個の文書を見てみると、どの検索方法を用いても大体同じ結果が得られているが、「できるだけ」フォルダによる検索の結果は、文書中の検索語句の出現回数を考慮しつつも、含んでいる検索語句の種類を重視している傾向が見られる。一方、「ひとつは」フォルダによる検索の結果の場合には、含んでいる検索語句の種類を考慮しつつも、出現回数を重視している傾向が見られた。

次にグラフについて見てみると「できるだけ」フォルダの結果は「ひとつは」フォルダの結果と or 検索の結果を比較してみると、「できるだけ」フォルダの結果のグラフは他の２つに比べ、下に凸の具合が非常に急であると言える。これは、検索語句を１つでも多く含むと急激にスコアが高くなるようにした事が反映されている結果であり、提案概念に従っているといえる。

最後にユーザの労力について見てみると、まず、検索の入力にかかる時間であるが、フォルダシステムを利用した場合の方が、入力する検索語句の数が多くなる可能性が高いため、入力するのに必要とする時間は多いが、フォルダシステムの場合には、自分が思いついたものから順に次々と入力していく事ができ、すぐ入力できる。従って、短い時間で自分の要求する検索を得られる検索式を導き出せると言える。

次に、検索に関する待ち時間であるが、従来の手法では、検索語句の増加に伴い、線形的に時間が増加するが、フォルダシステムではべき集合を用いているため、指数関数的に増加していく事になる。従って、検索語句が増えるにつれ急激に処理時間が長くなる。

そして、希望する結果にユーザが出会えるまでに必要な回数について考えてみると、従来の検索では、試行錯誤の回数だけ検索を行う必要があるのに対して、フォルダシステムを何度も対話的に検索を行う必要が無いため、本システムを利用する事により、大幅に検索のための労力と時間を省く事が可能である。

以上の考察を統合的に考えると、フォルダシステムを利用する事により、ユーザの検索に対する労力及び時間を減らす事ができると考えられる。

## 7. ま と め

本研究ではユーザが日常使っている曖昧な条件に着目し、全文検索において曖昧な検索を行う事ができるように、従来 and,or といったブーリアン演算子により単語を結合していた検索に対し、「できるだけ」「ひとつは」という概念を導入し、単語群に基づく検索を提案した。これにより、「できるだけ」という概念により、全ての語句を含む文書を優先する一方で検索式の語句の一部を含んでいる文書も結果に反映される検索が可能となり、「ひとつは」という概念により、単純に出現回数が多い文書を優先するだけでなく含まれている語句の種類を考慮に入れるという検索が可能となった。

また、単語群に基づく検索を実現するためにフォルダシステムを実装した。本システムを利用することにより、従来よりも曖昧な条件を検索する事が可能となり、また、なかなか自分の希望する検索結果が得られなかったという点も改善された。その結果として、今までよりも検索のための労力と時間を削減する事ができた。

今後ユーザの検索に対する要望はさらに高度化かつ広範囲化していく事が予想される。そのため、そのニーズに応えることが出来るように、ユーザの検索に対する要望を１つずつ解決していき、より便利な全文検索システムとなるように研究を進めていきたい。

## 文　　献


[1] 全文検索システム Namazu: http://www.namazu.org/

[2] 馬場 肇, 改訂 Namazu システムの構築と活用～日本語全文検索徹底ガイド～, ソフトバンクパブリッシング,2003

[3] G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[4] 松村敦, 高須淳宏, 安達淳,"単語間の係受け関係を用いた情報検索手法の評価"情報処理学会論文誌：データベース,Vol.41,No.SIG1,pp. 22-29,Feb,2000

[5] 宮川祥子, 清木康,"特定分野ドキュメントを対象とした意味的連想検索のためのメタデータ空間生成方式 "情報処理学会論文誌：データベース,Vol.40,No.SIG05-003,1999

[6] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", 1998

[7] 北内啓, 高木徹, 岩城修,"意味情報に基づく検索と全文検索の統合", 情報学基礎研究報告,No.060 - 003,2000

[8] 徳田圭世, 間瀬久雄, 辻洋,"デジタルドキュメントにおける共起データを用いた検索ターム連想支援について", デジタル・ドキュメント研究報告,No.010 - 003,1997

[9] Fabio Crestani, Gabriella Pasi (Eds.) Soft Computing in Information Retrieval – Techniques and Applications

[10] Namazu - 全文検索で文書の山に立ち向かう: http://www.namazu.org/ sat ipsj/

[11] Fox, E. A. Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types. Ph.D. Dissertation, Department of Computer Science, Cornell University (1983).