文書構造を利用した情報検索と利用者支援システムに関する提案

佐々木 貴文† 上島 紳一‡

†‡関西大学大学院総合情報学研究科 〒569-1095 大阪府高槻市霊仙寺町 2-1-1

E-mail: †fb2m124@edu.kutc.kansai-u.ac.jp, ‡ueshima@res.kutc.kansai-u.ac.jp

現在，大量の電子文書の普及に伴いそれらの中から必要とする情報を検索する手法が重要となっている．しかしサーチエンジンなどの全文検索では検索が文書単位に行われるため使いにくい欠点を持つ．本研究では情報を利用者の興味の範囲に応じて文書内から抽出し集約して１つの情報として見ることを目的とするシステムの提案を行う．ここではこのような情報の集約と情報の粒度の調整を行うシステムを次の３段階の方法を用いて行う．（１）各文書の持つ章，節，段落などの構造とキーワードの位置関係を考慮した全文検索により，文書の意味による構造化を行う．（２）全文検索結果となる文書からキーワードに関する情報を含む部分のみを取り出し検索結果として利用者に提示する．（３）提示された結果に対して利用者の興味に応じた情報の粒度の調整を可能とするための GUI による操作機能の提案．これらを実現するため本研究ではＸＭＬにより文書構造を明確化し処理に利用している．

キーワード 全文検索,情報検索,XML

Proposal of A User Support System with a Text Search Method using Document Structure

Takafumi SASAKI† Shinichi UESHIMA‡

†‡Graduate School of Informatics, Kansai University 2-1-1 Ryozenji, Takatsuki-City, Osaka, 569-1095 Japan

E-mail: †fb2m124@edu.kutc.kansai-u.ac.jp, ‡ueshima@res.kutc.kansai-u.ac.jp

Recently text search method plays crucial roles in information systems. Search engine retrieval results provide document wise for given queries while they do not answer enough information for their appropriate aggregation. In this paper, we propose a user support system for text retrieval that helps a user to find relevant information pieces, and aggregate them by adjusting their granularity via user operation through GUI. This system has (1) the full-text search function considering the position of related keyword in documents structure generated from chapters and paragraphs, (2) function to extract only a portion including the information about a keyword from the document, which brings a full-text search results as a reference.,(3) GUI that supports user to operate results to adjust their granularity according to his needs. We here employ XML to handle document structure explicitly.

Keyword Full Text Search, Information Retrieval, XML

1. はじめに

現在，大量の情報が電子データ化されており検索システムにより必要な情報の検索が行われる．

その際，利用者が行う作業は次のようになる．

（１）サーチエンジン等の全文検索システムを用いキーワードを含む文書の検索を行う

（２）検索結果となる文書を開きキーワードに関する記述を探す

（３）２の作業を繰り返し複数の文書に分散し記述されている情報を集約する

これらの作業を繰り返すことにより検索システムの利用者は必要な情報を得ることができる．

しかしこれらは誰にでも簡単に行うことができるわけではない．その理由は検索システムに次のような問題点があるためである．

・検索結果に不必要な文書が多く含まれる

・全文検索における検索結果は文書名とキーワードの前後数文字である場合が多く必要な情報を得るためには文書を開いてその中からキーワードに関係する記述を探す必要がある．

本研究ではこれらの問題を解決しより必要な情報の検索を容易に行うためのシステムの提案を行う．システムの流れは実際に利用者が検索を行う際の手順に基づき次のようになる．

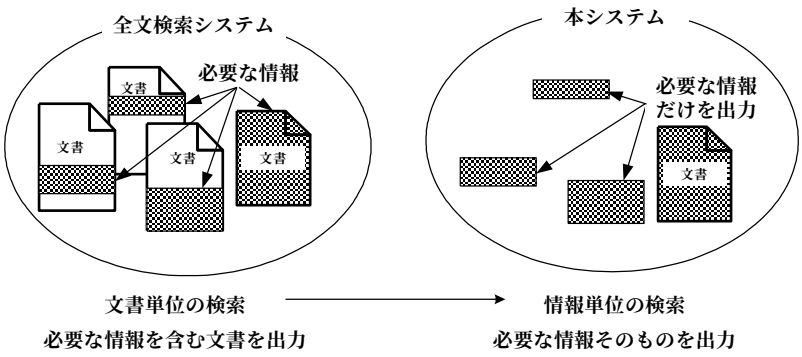

図 1　情報単位での検索

（1）検索対象となる文書を XML 化し文書の持つ構造を明確にする．
（2）文書構造を利用した全文検索の索引の作成
（3）文書構造を利用した全文検索
（4）検索結果文書からの情報の抽出と提示
（5）検索結果の情報粒度調整のためのインタフェースの提供

また既存の全文検索システムでは複数のキーワードが入力されるとそのキーワードを含む文書の文書名とキーワード前後数文字が表示される場合が多い．しかし利用者が必要とするのは文書全体ではなくキーワードに関する情報，つまり文書中のキーワードに関する記述のある部分のみである．そこで本システムは検索の対象を文書単位ではなく文書に記述された情報単位とすることを目標とした（図１）．

これらの詳細について２で検索の準備と文書構造を用いた全文検索について，３で全文検索結果となる文書内からの情報の抽出と出力方法，利用者による情報集約支援について述べる．

また関連研究として絹谷らによる XML 部分文書の抽出などがある．この研究では XML を DTD や要素名などの構造を利用し文脈の境界を求めあらかじめ分割を行っている．本研究ではこれとは異なり最小単位の要素を集約する形で情報の抽出を行う．

2.　文書構造を利用した情報の検索

2.1.　本システムの流れ

本システムは全文検索[1]を基としておりシステムの流れは図 2 で示したように準備と検索の２つに分けることができる．

2.2.　XML の利用

2.2.1　対象となる文書

本システムは検索対象を XML 文書とする．これは文書の構造を利用するため対象となる文書が明確でならなければならないからである．

なお本システムでは DTD などの XML 文書に対する構造定義は利用しない．つまり整形式 XML 文書が検索対象となる．ただし１つの要素に含まれるテキストデータが極端に大きい場合や，逆に小さすぎる場合は検索が正しく行われない場合がある．

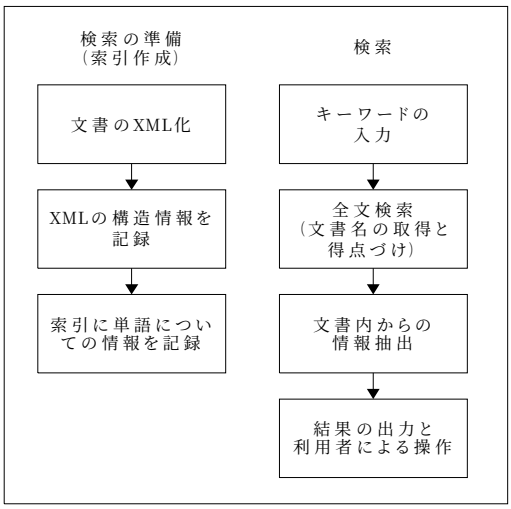

図 2　システムの流れ

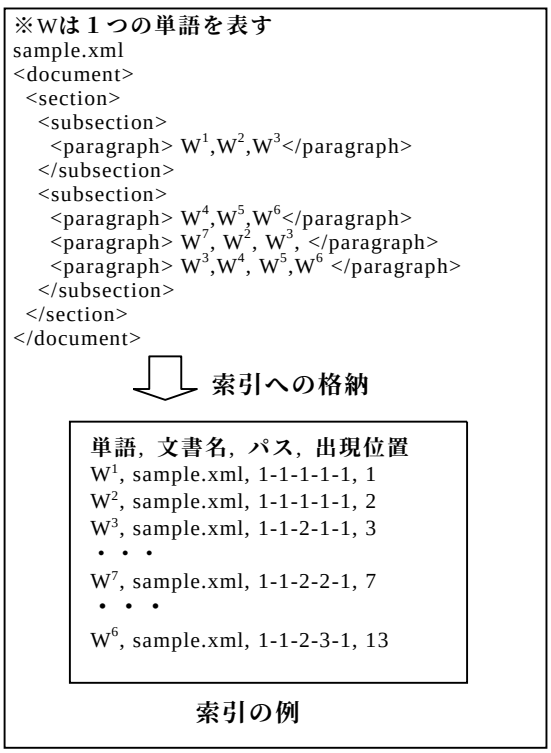

図 3　索引への格納

2.2.2　文書の XML 化

2.2.1 で述べたように本システムは XML 文書を対象としている．そこで XML 以外の文書を XML 化する必要がある．文書の XML 化については以前に実験を行っている[2]．

2.3.　索引の作成と文書への得点づけ

本システムでキーワードが入力されるとまず行わ

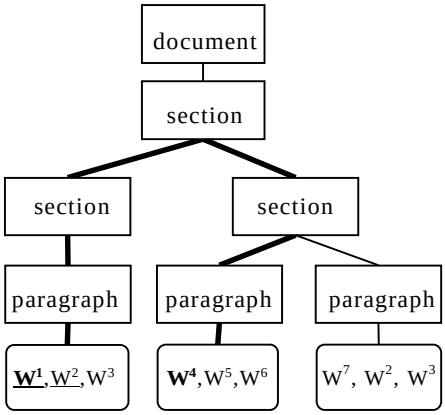

図4　XML文書への得点付け

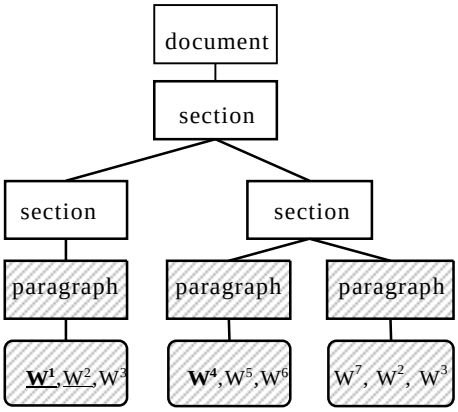

図5　抽出単位

れるのは全文検索となる．そのための索引作成と結果に対する得点付けについて述べる．

2.3.1　索引の作成

検索対象となるXML文書に予め索引の作成を行う．

図3の下側にある索引の例は上のXML文書を索引化した例である．索引には次の情報を記録する．

（１）単語を含む文書名（数値コード化）

（２）単語のXML文書内でのパス（その単語に至るまでの経路）

（３）単語の文書内での出現位置

単語のパスとは，各単語がXML構造上の階層の位置を示すものであり，本システムではルート要素から見て何番目の子であるかを数値で表す．例えば W^7 はルート要素から見ると１番目の子の１番目の子の２番目の子の２番目の子の１番目の子となる．そこでそのパスである,1-1-2-2-1を索引に記録する．なおこのパスの表現方法はdewey orderとして知られており，XMLの結合やデータの追加を効率的に行える[2]．

また単語の出現位置は文書の先頭から見てその単語が何番目に出現したかを表す．W^7 は７番目に出現した単語なので７が記録される．

2.3.2　文書への得点付け

本システムではキーワードが入力されると索引からキーワードに関する情報を取得する．これによりまずキーワードがどの文書に含まれているか判明する．そしてこれらの文書に対して得点づけが行われる．

得点付け方法についてはTF/IDF法やPage Rankなど様々な方法が存在するが本研究ではXMLの構造を利用する方法を提案する．なおキーワードは２個以上入力されることを前提としている．

（１）検索語を2語1組にし，それぞれが存在する要素間のパスの数を計算する．図4で示した構造を持つXML文書に対してキーワードが W^1 と W^4 で検索された場合，パスは太線で示したものとなりパス数は6となる．

（２）2語が同じ要素内に存在する場合2語の出現位置を比較し差を得点とする．図4でキーワードが W^1 と W^2 の場合がこれに該当する．この場合，W^1 は文書の先頭から見て１番目，W^2 は文書の先頭から見て2番目に出現しており出現位置の差は2-1=1となる．

（３）上記のいずれかで求まった値に重みをかけ文書の得点とする．重みの値は（２）より（１）に掛けるものの方が大きい．

本システムでは上記の方法で計算した点数の低いものが上位となる．

検索対象となる単語が文書内で複数ある場合はそれぞれの組合せすべてで得点を計算し最も得点の低いものを採用し文書に対する得点とする．

検索語がｎ語（ｎ≧３）の場合，${}_nC_2$ 通りの組合せについて上記の計算を行い，その総和を得点とする．例えば検索対象語が「W_1」「W_2」「W_3」の場合(W_1,W_2), (W_1,W_3), (W_2,W_3)の3通りについて得点を求めその和をその文書に対する得点とする．

なおこの方法を用いてTF/IDF法との比較実験を行っている[3]．

3.　情報の抽出と利用者支援

3.1.　文書内からの情報の抽出

2では本システムにおける全文検索方法について述べた．ここまでで得られるのはキーワードに関する情報を含む文書の文書名である．そこで次に行うことは文書内からの情報の抽出となる．

3.1.1　抽出対象の選択

情報抽出を行うにはどの文書からどの部分を取り出すかを決定する必要がある．

【抽出元】

まずは抽出元の決定は２の全文検索結果の上位数件（任意または固定）とする．本システムでは単語の位置情報を全文検索で利用しているため検索結果上位ならばある部分にキーワードがまとまって存在するこ

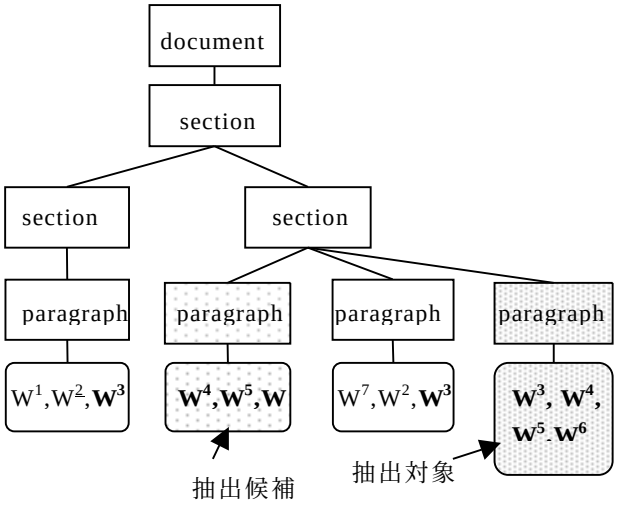

図 6 抽出対象の決定

ととなる．そのため抽出部分を明確にしやすい利点がある．

【抽出単位】

次に入力されたキーワードに対して文書中のどこがキーワードに関する情報が記述されている部分であるかの決定を行う．

本システムでは情報を抽出する単位を要素値としてテキストデータを含む要素（最小部分木，図 5 の斜め線部分）とする．これは本システムがあらかじめ文書を分割しておきそれらに対して検索を行うのではなくキーワードを含む最小限の要素を組み合わせて動的に抽出部分を決定するためである．

このように抽出部分を動的に決定することによりキーワードにより必要となる部分が異なることへの対応と抽出部分にキーワードが集中することにより見出しの役割をさせることが可能となる．

ただし要素に含まれるテキスト量が大きいと（ある文書の章すべてが１つの要素に含まれる場合等）利用者が検索結果に対し再度，文書内検索を行わなければならなくなる．また逆にテキスト量が少なすぎる場合は正しく検索と抽出が行われない事が考えられる．そのためあらかじめ構造の変更が必要となることがある．

【抽出部分の選択】

全文検索結果となる文書内からどの部分を抽出するかを選択する方法について述べる．その流れは以下のようになる．なおこの処理は全文検索の得点計算後に行われる．

（１）文書内の各要素がどのキーワードを含むかの記録を行う．

（２）キーワードをすべて含む要素は抽出対象（検索結果として出力する）とする．

（３）（２）に該当しない文書に対してはキーワードを一定数以上含む場合抽出候補とする．

（４）（３）で決定した抽出候補については抽出対象要素と抽出候補要素を結ぶパスの数，または前後の要素が含むキーワードを考慮し抽出対象にするかの判断を行う．

まず，抽出対象要素と抽出候補要素を結ぶパスの数を用いる方法について述べる．

図 6 においてキーワードを「W^3,W^4, W^5,W^6」，抽出候補要素キーワードを３つ以上含む要素とする．なお W は要素値に含まれる単語を表す．

これによりパスが”1-1-2-3-1”の要素が抽出対象，パスが”1-1-2-1-1”の要素が抽出候補要素となる．ここで両者のパスを比較することにより２つの要素を結ぶパスの数を求めることができる．この例の場合，２つの要素を結ぶパスの数は 4 となる．この値が N（N≧0）以下なら抽出候補を抽出対象とする．例えば N の値が 4 以上の場合なら抽出候補要素は抽出対象となるが N が４未満の場合は抽出対象とならない．

次に前後の要素に含まれるキーワードを考慮した方法について述べる．

この方法では抽出候補である要素の前後の要素に着目する．例の場合，抽出候補要素に含まれるキーワードは「W^4, W^5,W^6」の３つであり含まれないキーワードは「W^3」である．ここで抽出候補要素の前後に当たる要素（”1-1-1-1-1”,”1-1-2-2-1”）を見ると「W^3」を含む．つまり前後の要素に含まれるキーワード要素と抽出候補要素に含まれるキーワードを合わせるとすべてのキーワードを含むこととなる．そこで抽出候補とその前後はキーワードに関するする情報を含むと仮定し抽出対象とする．

これらの方法を用いて抽出候補要素を抽出対象とすることにより結果として出力を行う情報量が増えることとなる．しかし増えた情報の中には不必要なものが含まれる可能性も高くなる．不要な情報の量が少なければ利用者はそれらを読み飛ばすだけでよいがなるべく不要な情報は少ない方が望ましい．そこで抽出候補を決定する際や抽出対象にする際のパラメータの設定が重要であると考えられる．

【検索結果の出力】

本システムで結果として 3.3.1 節の方法で選択された複数の要素の内容をひとつの文書として出力を行う．

図 7 は検索結果を XML 文書として出力した例である．検索結果は全文検索時に抽出元文書につけられた得点に基づいた順で出力される．つまりまず全文検索上位１番目の文書から取り出された要素，次は全文検索２番目の文書から取り出された要素・・・の順に出力されることとなる．

ただし，この方法では１つの文書内に重要な部分とそうではない部分が存在する場合，どちらも抽出元が

```
<?xml version="1.0" encoding="SHIFT-JIS" ?>
<!DOCTYPE result [
<!ELEMENT result (keywords,results*)>
<!ELEMENT keywords (keyword*)>
<!ELEMENT result (file,score,text*)>
<!ELEMENT file (#PCDATA)>
<!ELEMENT score (#PCDATA)>
<!ELEMENT text (#PCDATA)>
<!ATTLIST file code ID #REQUIRED> ]>
<results>
<keywords>
<keyword>阪神</keyword><keyword>星野</keyword>
<keyword>監督</keyword><keyword>退任</keyword>
</keywords>
<result>
<file code="6">gs6.xml</file>
<score>10</score>
<text path="1 1 1">「縦じまの重さ、想像絶する」阪神の星野監督が退任会見／プロ野球</text>
<text path="1 2 3 1">阪神は２８日、星野仙一監督（５６）が退任し、後任に岡田彰布（あきのぶ）守備走塁コーチ（４５）が昇格すると発表した。星野監督は今後もアドバイザー的な立場で球団に残り、岡田新監督を支える。〈岡田新監督の「顔」２面〉</text>
</result>
<result>
<file code="9">gs9.xml</file>
<score>34</score>
<text path="1 1 1">プロ野球　阪神・星野仙一監督が退任、岡田彰布氏が後任に－－阪神正式発表</text>
</result>
<result>
<file code="34">gs34.xml</file>
<score>863</score>
</result>
<result>
<file code="23">gs23.xml</file>
<score>877</score>
</result>
<result>
<file code="11">gs11.xml</file>
<score>924</score>
</result>
</results>
```

図 7 結果の出力例

同じ文書となるため重要度に関係なく出力されることとなる．そこで１つの文書から複数の要素が取り出される場合，何らかの方法で取り出される要素のグループ化を行いそれぞれについて再得点付けを行い重要度の高い物から順に出力するのがよいと考えられる．

結果はこのようにひとつの文書とするが抽出する要素の数によってはこの文書自体が非常に大きな物になる可能性がある．そこで出力行う際には一文書から取り出す要素の最大数及び全体での取り出す要素の数を決定しておく．

3.2. 利用者による情報粒度の変更

3.1 で述べたように結果として出力が行われるのは主にキーワードが含まれる複数の要素となる．また結果として出力される情報の粒度（情報の大きさ）はキーワードによって異なることとなる．さらに抽出条件を満たしていても 3.1.2 で述べたような制約により結果に含まれない要素も存在する．

そのため結果として出力されない要素にも利用者によって必要となる情報が含まれる可能性がある．また出力された要素に記述されたテキストを読みその前後を更に読みたいと判断する場合も考えられる．

そこで抽出された部分を見た上で他の部分を表示させるかどうかを任意に選択可能にすることにより利用者の情報検索作業を支援する方法を採用することにした．今回定義した操作は以下である．

（１）ひとつの文書内の条件を満たす要素をすべてもしくは一定数表示
（２）表示されている要素の次または前のテキストデータを含む要素を表示
（３）文書の構造を提示し表示する要素を選択

これらの操作を行うための GUI を作成した例が図 7 である．利用者は出力結果に含まれる要素に対する操作と抽出元となった文書に対する操作を行えるようにした．この場合の操作の流れは次のようになる．

１．操作対象とする要素，もしくは文書を選択
２．どのような操作を行うか決定
３．操作の結果を表示

これらの操作を行う際にはなるべく利用者に XML 文書が検索対象になっていることを意識させない方法とすることを目指している．

図 8　利用者インタフェースの例

4. 実験と考察

本研究では提案手法を用いたプロトタイプシステムを作成し実験を行った．

4.1. プロトタイプシステム

プロトタイプシステムでは次の事を前提とした．

・ 利用者は検索を行うための専用のソフトウエアを使用しない．

・ 検索システムはブラウザなどの Web 端末から利用可能とする．
・ プログラムはすべてサーバ上で稼働する．

このような前提を設定する理由として，本システムが想定する利用者がコンピュータに関する知識を持っているとは限らないからである．そのためソフトウエアの導入や設定，複雑な操作を必要とすることなく容易に使えることを目的とする必要がありプロトタイプシステム作成でも重要項目とした．

4.2. 検索実験

4.2.1 実験対象

今回，検索対象として約 5000 件のニュース記事を１記事１ファイルとして XML 化したものと 50 記事 1 ファイルとして XML 化したものの２通りを用いた．

対象データをこのように設定した理由としてはまず本システムの実験に適したテストコレクションが存在しないことが挙げられる．そのため検索対象データの作成と評価方法を適宜する必要がある．しかし個人が作成した HTML のように特定の構造を持たない文書の変換は難しい．そこでデータ作成の効率なども考慮し，同一の構造を持つ HTML で記述されたニュース記事を XML に変換を行い検索対象としている．

4.2.2 実験方法

いくつかの質問文の作成を行い，それらについての情報を得るに適すると思われるキーワードを選択し本システムで検索を行った．なお実験結果の評価項目は次の７項目である．

A)全文検索結果数
B)抽出対象となった文書の数
C)抽出対象文書に含まれる総要素数
D)取り出された要素の数
E)キーワードをすべて含む要素の数
F)質問ついての情報の見出しとなる要素（要素の内容を見ることで前後の要素に書かれている内容が推測可能なもの）
G)質問に関する具体的な情報を含む要素（質問に関する記述を含む要素（補足内容は除く））
Ｆ)キーワードとは関連性のない要素

これらの評価は検索結果として出力された文書に含まれる要素内容を読み，行った．

4.2.3 実験結果と考察

実験を行った結果より本提案手法は次のような特徴を持つことが確認された．

・全文検索結果から余分な情報を省くことができ，利用者が入力したキーワードについての情報を抽出し提示することが可能である．
・抽出された文書内容に連続性がないためある構造以下の内容をすべて必要とする場合には必ず利用者による操作が必要となる．
・情報の粒度は抽出要素決定法により異なる．幅広い情報を得たいときは前者を，より具体的な情報を得たいときは後者を利用するなど使い分けるのが良いと考えられる．
・抽出されない要素に情報が含まれることがあると言えるがそれらは利用者の情報粒度調整操作により表示可能となるため完全に情報が欠落するわけではない．
・キーワードに付いての記述が中心となっている文書もしくは文書中の一部にキーワードについての記述が存在する文書のいずれにおいても情報の抽出は行うことができる．
・検索対象となる文書サイズが大きくなると抽出対象選択にパス数を利用した場合，不必要な情報が抽出される割合がわずかだが高くなる．

5. おわりに

本稿ではコンピュータや検索について詳しくない人でも利用しやすい情報検索システムの提案をした．その具体的な方法として全文検索と XML を利用した要素の抽出を行っている．

そして検索システム内の処理だけではなく GUI により利用者にも操作可能にすることによりすべてをコンピュータに任せるのではなく利用者の意図を検索結果に取り入れることを可能になる．

また，今後の課題として以下のことが考えられる．

本研究では GUI に関する提案も行ったが利用者アンケートなどの実験を行うことができなかった．そこでそれらを実施し結果を反映することにより GUI の構築が課題となる．

今回，評価の対象とはしなかったが全文検索の索引やアルゴリズムについて最適化を行っていないためファイルサイズや計算量など考慮すべき点が存在する．それらについての改善も必要であると言える．

参考文献

[1] 石川佳治,定兼邦彦,北川博之：文書データを対象とした索引技術,情処誌 42 巻 10 号,p.980-987,2001 年 10 月

[2] S. Al-Khalifa, H.V. Jagadish, N. Koudas, J.M. Patel, D. Srivastava, Y. Wu: Structural joins: A Primitive for Efficient XML Query Pattern Matching. ICDE 2002.

[3] 高野哲郎,佐々木貴文,上島紳一:テキストの XML 文書化と全文検索に関する検討,第 64 回情処全大 ,論文集(3) p.161-162,2002 年 3 月

[4] 絹谷弘子,波多野賢治,吉川正俊,植村俊亮：XML 文書の文書構造と内容を用いた部分文書の抽出方法,情処誌 vol.43 No SIG 2(TOD 13),p.80-92,2002 年 5 月