情報検索技術を用いたXML部分文書の検索手法

波多野 賢治[†] 渡邉 正裕[††]
吉川 正俊[†,†††] 植村 俊亮[†]

インターネット上でデータ交換やデータの配布が行われるようになり，ネットワーク上における構造化文書の利用に注目が集まっている．XMLはネットワーク上におけるデータ交換の標準フォーマットとなりつつあり，我々はXMLを利用することで様々な文書を効率良く記述し，ネットワーク上を流通させることができる．しかし，XML文書の検索手法としてこれまで提案されてきた手法は，XML問合せ言語を用いた文書の論理構造を意識した方法もしくはデータベースに格納後SQLを用いた方法であり，情報検索技術を用いた手法は数少ない．そこで本論文では，XML文書の検索に情報検索技術を用いた新しい検索手法の提案を行い，利用者がXML文書の構造を意識せずに問合せができ，その問合せを受けた検索システムが文書構造と利用者の問合せの内容を考慮することで，利用者の問合せに合致したXML部分文書を検索することができるシステムの実現を目指す．また，実際に検索システムの実装を行い，その有効性の検証を行う．

A Retrieval Method for XML Subdocuments Based on Information Retrieval Techniques

Kenji Hatano,[†] Masahiro Watanabe,[††]
Masatoshi Yoshikawa[†,†††] and Shunsuke Uemura[†]

As XML is becoming as the standard format for data exchange on the Internet, the use of structured documents draws a greater deal of attention. Techniques for retrieval of strucrued documents can be classified into two categories — database-based approach and IR-based approach. Until now, much attention has been paid on the database-based approaches such as XPath and XQuery. However, IR-based retrieval techniques for structured documents are not matured enough. In this paper, we propose a new retrieval technique for XML documents based on IR-based retrieval technique. With our technique, users can extract partial documents relevant to users' query without knowing the structure of the documents. Furthermore, we examined the difference of the accuracy between the present techniques and the technique we are proposing and verify the validity of our proposed technique.

1. はじめに

インターネット技術の発展は，構造化文書の一種であるHTML（HyperText Markup Language）の利用と進歩を促し，その進化としてXML（Extensible Markup Language）[4]の提案が行われた．XMLを文書の記述に利用することで文書の可読性を上げたり，目的に応じてデータ構造を柔軟に表現できるなど，我々は多くの点でその恩恵を受けることができる．現に電子文書のデータ交換・再利用・配布の目的で広範囲にわたってXMLが利用され始めており，今後，膨大な量のXML文書が出現するようになるのは間違いない．したがって，これらの生成，格納，検索を効率的に行うことは大変重要な技術となる．しかし，XML文書を検索する手法として提案されている方法は，XPath（XML Path Language）[7]やXQL，XQueryのようなXML問合せ言語を用いた方法[9),22)]，もしくはXML文書をその構造を利用してデータベースに格納し，SQLを用いて問合せを行うという手法[1),11),25),26)]が主流であり，この手法では，利用者がXML文書の構造をあらかじめ理解しておかなければ，問合せに最適な部分文

† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology (NAIST)
†† 独立行政法人国立特殊教育総合研究所情報教育研究部
Department of Educational and Information Technology, The National Institute of Special Education (NISE)
††† 国立情報学研究所ソフトウェア研究系
Software Research Division, National Institute of Informatics (NII)

書を検索することができないという欠点を持っている．

そこで本論文では，こうした問題点を解決するために，XML 文書の検索に XML 問合せ言語を用いたり，データベースの検索機能を利用するのではなく，情報検索技術を利用して XML 文書の内容解析を検索システムで行うことで，利用者が XML 文書の構造を意識せずに最適な部分文書を検索できるような検索システムの提案を行う．提案する検索システムでは，利用者が検索システムに入力した問合せに類似する XML 文書の要素を抽出し，問合せに対する要素の類似度と文書構造を利用して，その要素がグループ化され部分文書を形成するか否かを判定する．そうして得られた XML 文書の要素もしくは部分文書を情報検索単位としてランキング表示し，それを検索結果として利用者に提示する．

以下，2 章では，本論文の基本事項として XML 問合せ言語と関連研究について，3 章では，情報検索技術を用いた XML 文書の検索システムの概要について，4 章は，提案システムの有効性を示すための実験の手法とその結果の考察を述べ，最後に 5 章において，本提案手法のまとめと今後の課題について述べる．

2. 基本的事項と関連研究

2.1 XML 問合せ言語

W3C（World Wide Web Consortium）の勧告である XPath は，もともとスタイルシートのワーキンググループとリンクのワーキンググループで作成していた言語を統合してできたものであり，データベースの観点から見れば，問合せ言語に必要な機能の一部を実現している．XPath の基本的な考え方は，XML 文書の木構造中において，指定された部分構造が与えられたパターン，すなわち経路式にマッチする要素集合を返すというものである．しかし，XPath では関係データベースでいう結合に相当する演算や新規文書の作成といった機能を持っていないため，XML の問合せ言語の必要性の認識の声が高まり，現在，制定のための活動が行われている段階である．現在のところ，具体的に提案されているものは 1 章でも述べた XQL と XML-QL である．XQL は XSL（Extensible Stylesheet Language）パターンを拡張した構文，すなわち先に述べた XPath に近い形式で木構造のあらゆる要素を特定することが可能である．一方，XML-QL は XML 文書自体に特化した問合せを考慮しており，XQL とは違って問合せ結果の再構成や変換が可能となっている．しかし，現在は，XQL 自身も能力が拡張され，両者の表現能力の差はほとんどない．さらに，近年になって，XQuery と呼ばれる XML 問合せ言語が注目を浴びている[5]．XQuery は，Quilt[6]を基にしており，W3C の XML Query ワーキンググループで提案されている問合せ言語である．これらの問合せ言語は，これまで提案された XML 問合せ言語の長所を取り入れ，人間が理解しやすい問合せ構文を採用するなど工夫されている．しかし，いずれの問合せ言語を用いても，利用者が XML 文書に問合せを行う場合，その文書の構造をあらかじめ把握しておく必要があるという問題点が生じる．なぜなら，現在のインターネット上に氾濫する文書の状況を考えれば，膨大な XML 文書の量と，それにともなう様々な種類の文書構造についての知識を，検索の際に利用者に求めるのは困難だからである．利用者が容易に XML 問合せ言語を利用できるようになるためには，問合せインタフェースなどで専門知識を必要としないよう工夫が必要である．

2.2 関 連 研 究

2.1 節で述べたように，XML 問合せ言語を用いた文書検索は，文書の構造を指定した要素および部分文書の検索は可能であるが，利用者が文書の構造を知らない場合は，結局，すべての要素に対して文字列検索をしなければならないという問題点が生じる．こういった問題点を解決するためには，情報検索技術と XML 問合せ言語の統合が必要となってくる．

このような情報検索技術と XML 問合せ言語の統合に関する研究は，近年，数多く行われるようになった．Fuhr らは，XQL において情報検索技術を利用できるように拡張した検索言語 XIRQL を提案している[13]．この論文では，XQL と情報検索技術を統合するために必要な要素技術として，出現単語に対する重み付けの手法や問合せに対する類似度を基にした検索の実現手法が必要であるとし，XQL に対してこれらの手法の拡張を行っている．また，Florescu らは，XML 問合せ言語 XML-QL を拡張し，XML 文書の各要素において文字列検索を実現するための手法の提案を行っており，XML 文書の構造を知らない利用者でも，異なる DTD（Document Type Definition）を持つ XML 文書の検索を容易にできるよう工夫を行っている[12]．我々の先行研究においても情報検索技術と XML 問合せ言語 XPath との統合利用を考え，文字列検索とベクトル検索の統合を行うことで実現する研究を行っており，XML 文書の検索手法に対して文書構造を意識した情報検索技術の適用を行っている[28]．

さらに，こうした技術を適用した XML 検索エンジンもいくつか登場しており，GoXML[31]や XYZfind[10]などは情報検索技術と XML 問合せ言語の統合した手

法で該当 XML 文書を検索することが可能である．

本論文で提案する手法は，これらの手法のように単に XML 問合せ言語と既存の情報検索技術の統合を図るだけではなく，XML 文書の各要素の内容☆や文書構造を考慮し，利用者が検索システムに入力する問合せに相応しい部分文書を検索できるよう工夫している．この点が，従来の構造化文書の検索手法とは大きく異なる．

3. 情報検索技術を用いた構造化文書の検索

本章では，情報検索技術を用いた XML 文書の検索を行ううえで必要な要素技術の提案と，その技術を利用した検索システム実現の方法について述べる．

3.1 XML 文書の情報検索単位

本研究の目的は，利用者が文書構造の指定をせずに，検索システムが問合せに相応しい XML 文書要素もしくは連続したそれら文書要素で構成された部分文書を検索結果として返すことである．本研究では，このような問合せの内容に相応しい検索結果を XML 文書の情報検索単位と呼ぶ．

XML 文書からこの情報検索単位を抽出するために，本手法では文書を要素単位に分割する手法をとる．文書をある単位に分割しそれを検索の単位とする手法は，従来からパッセージ検索（passage retrieval）[23]として行われており，これまで多くの研究が行われている．しかし，パッセージ検索が検索結果を分割した単位もしくは文書全体としているのに対し，本手法は，問合せに対する文書要素の類似度と文書の論理構造を利用し，利用者の問合せに類似し，かつ隣接した文書要素であればそれらの要素をグループ化して検索結果として返す点が異なる．さらに，本手法では，ノードの意味関係に注目し，あるノードがその兄弟要素や子要素の内容を要約したものである場合は，そのノードの特徴量を兄弟要素や子要素の特徴量に反映させる手法をとっている．このような文書ノード間の関係を本論文ではノード間意味関係と呼んでいるが，この手法を適用している点もパッセージ検索にはない点である．

ノード間意味関係を**図 1** の文書を例に説明すると，`chapter` 要素の `title` 属性の値や `title` 要素の要素内容は，後に続く兄弟要素や子要素の内容を要約したものであると考えることができる．つまり，その親要素である `chapter` を構成する子要素 `section` に特徴量を反映させ，検索の対象とはならないようにすべきである．したがって，図 1 の灰色の要素間には意味関係が存在すると考えられるため，これらの要素が検索対象となる．

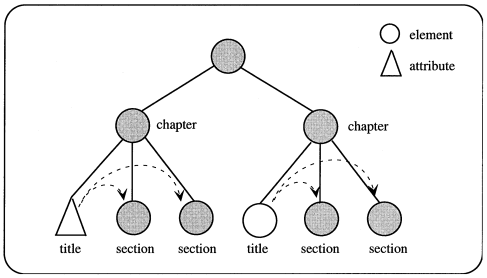

図 1 ノード間意味関係
Fig. 1 Context relationship among documents' nodes.

さて，利用者の問合せに相応しい情報検索単位を検索できるようにするためには，文書要素から特徴を抽出し，その特徴量を解析することが必要となる．2.1 節で述べた XML 問合せ言語をを用いれば，確かに文書要素の検索は可能である．しかし，これは利用者があらかじめ XML 文書の構造を理解しているからであり，文書構造に関する知識がなければ，XML 問合せ言語を使って文書要素を検索することはできない．また，情報検索単位を検索する際に，XML 問合せ言語のように要素の内容を考慮せずに単に文書の論理構造と要素内容に対するキーワードマッチングだけで検索することは問題となる．

このような問題点を解決するには，XML 文書の検索手法に情報検索技術を適用し，XML 文書を構成する要素の内容とノード間意味関係，すなわち XML 文書の文脈を考慮する検索システムを提案する必要がある．文書構造と内容を考慮した XML 文書検索システムは著者らが知る限りは存在しないが，Tajima らによる World Wide Web における情報検索システムは，利用者が入力した問合せから HTML 文書の持つ内容ととそれらを結び付けたリンク構造をともに考慮した検索結果を得ることができる[27]．

本論文では，利用者が検索システムに入力した問合せに相応しい XML 文書の要素もしくはそれら要素で構成された部分文書を XML 文書の情報検索単位とし，その情報検索単位で検索結果を返すことが可能な情報検索システムの提案を行う．以下の節では，具体的にその検索システムの実装の手法について述べる．

3.2 検索システムの実装

従来の文書検索システムとは異なり，XML 文書の検索システムでは，前節で説明した情報検索単位を検索結果とすべきであるため，XML 文書の最小単位で

☆ 本論文では，文書を成り立たせている実質や意味のことを指して単に内容という．XML では開始タグと終了タグで挟まれた部分のことを内容というが，ここでは区別のために要素内容と呼ぶ．

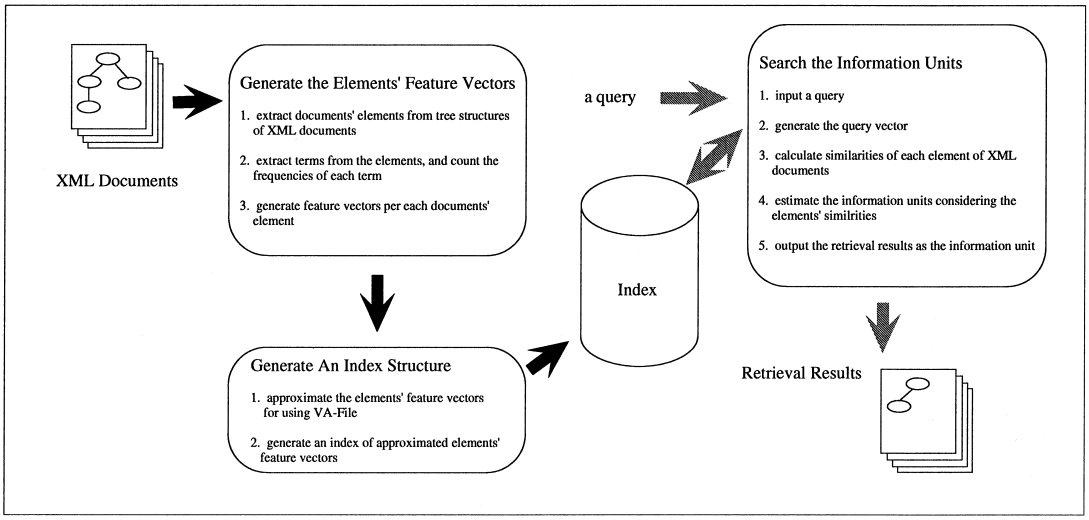

図2 XML文書検索システムの概略図
Fig. 2 Our information retrieval system for XML documents.

ある要素を文書の特徴量抽出の単位とする必要がある．検索システムにおける処理の流れを**図2**に示す．図に示されているように提案する検索システムは，要素特徴ベクトル生成，索引づけ，情報検索単位の検索の3つのモジュールに分かれている．以下の項では，これら3つのモジュールについて，さらに詳しく述べる．

3.2.1 要素特徴ベクトルの生成

文書検索で利用される文書の特徴量の表現方法，すなわち検索モデルは，これまで数多く提案されているが，本研究ではベクトル空間モデルを利用する[24]．このベクトル空間モデルを採用する際に問題となるのは，

(1) XML文書中に出現する単語の種類が非常に多くなる，すなわちベクトルの次元が大きくなる点．つまり，問合せとXML文書の要素との類似度を計算する際の計算コストが大きくなる点．
(2) 問合せに対して全XML文書の要素の類似度を計算するため，問合せと類似した要素を探索する際の計算コストが大きくなる点．
(3) ベクトル空間を表現するときに，XML文書に出現する単語をベクトルの基底とするが，文書に出現する単語には類似語，類義語も含まれているためにベクトル空間の各軸の直交性が保証されない点．

があげられる．(1)および(2)の問題点を解決するために，本研究では不要語リストを利用した不要語の除去処理および接辞処理を行うことで，文書中に出現する単語の種類を削減，すなわち，ベクトルの次元の削減を行った．また，次項で説明する索引づけにVA-File（Vector Approximation File）[29]を用いることで問合せ中のキーワードを1つも含まないようなXML文書の要素については類似計算を行わないようにし，類似した要素を探索する際の計算コストの削減を行った．しかし，(3)の問題点に関しては，LSI（Latent Semantic Indexing）[8]や意味の数学モデル[19]がその解決策として提案されているが，これらの処理のための計算コストや検索精度を考慮した結果，直交性の保証がされていなくても本検索システムはある程度の検索精度を得ることができると判断し問題点の解決を行わなかった．

以上のことを考慮したうえで，XML文書の要素特徴ベクトルの生成について説明していく．要素特徴ベクトルを生成するためには，まず，XML文書の各要素における特徴量の抽出を行わなければならない．以下にその手順を示す．

(1) **XML文書の木構造の解析**
まず最初にXML文書 D_i (i は1以上の整数) を要素に分割する．このとき，各要素 e_{ij} (jは1以上の整数) には文書の識別子である DID（Document IDentifier），要素の種類（タグ）の識別子である GID（General element IDentifier）および要素の木構造の中での位置を表す UID（Unique element IDentifier）を割り振る．このとき，ノード間意味関係を考慮して検索対象すべき文書要素を選定しておく必要がある．なぜなら，文献15)における予備実験において，ノード間意味関係を考慮して検索

対象となる文書要素を選別し，かつその要素に含まれる単語の数がある程度大きな値☆ をとらなければ検索精度が悪化するという知見が得られているからである．

(2) **XML 文書の各要素から出現単語を抽出**
XML 文書の要素 e_{ij} ごとに単語を抽出し，その出現頻度を数える．XML 文書 D_i の要素 e_{ij} から単語 $w_1, w_2, \ldots, w_{n'}$ (n'は 1 以上の整数) が抽出され，それぞれの出現頻度が $N^{e_{ij}}_{w_1}, N^{e_{ij}}_{w_2}, \ldots, N^{e_{ij}}_{w_{n'}}$ であるとすると，以下のようにベクトル表現される．

$$\boldsymbol{V}(e_{ij}) = (N^{e_{ij}}_{w_1}, N^{e_{ij}}_{w_2}, \ldots, N^{e_{ij}}_{w_{n'}}) \quad (1)$$

ただし，抽出された単語 $w_1, w_2, \ldots, w_{n'}$ は，すでに不要語処理および接辞処理は行われているものとし，ノード間意味関係は単語の出現頻度に反映されているものとする．

(3) **XML 文書の要素ごとに特徴ベクトルを生成**
検索対象となる全 XML 文書中に出現した単語の種類とその出現頻度を利用して，文書要素ごとに特徴量を計算，すなわち特徴ベクトルを生成する．さらに，全 XML 文書に出現する単語の総出現頻度を利用して正規化を行う．全 XML 文書から単語 $w_1, w_2, \ldots, w_n$ (n は 1 以上の整数かつ $n' \leq n$) が合計 $N_{w_1}, \ldots, N_{w_n}$ 個抽出され，それぞれの単語が要素 e_{ij} 中に $N^{e_{ij}}_{w_1}, \ldots, N^{e_{ij}}_{w_n}$ 回出現したとすると，要素 e_{ij} の要素特徴ベクトル $\boldsymbol{F}(e_{ij})$ は，

$$\boldsymbol{F}(e_{ij}) = \left(\frac{N^{e_{ij}}_{w_1}}{N_{w_1}}, \frac{N^{e_{ij}}_{w_2}}{N_{w_2}}, \ldots, \frac{N^{e_{ij}}_{w_n}}{N_{w_n}} \right) \quad (2)$$

となる．

3.2.2 情報検索単位の検索

3.2.1 項の式 (2) から計算された要素特徴ベクトル $\boldsymbol{F}(e_{ij})$ は，通常 5,000〜20,000 次元となる．これら要素特徴ベクトルから問合せに類似した特徴ベクトルを探索する際には，その高速化を図るために索引ファイルを生成するのが普通である．本研究では，その索引づけに VA-File を用いている．

VA-File は近似ファイルを用いた多次元ベクトル空間索引機構である．この手法では，ベクトル空間をセルに区切り，各セルにビット列を割り当てる．そして，セルに格納されるデータオブジェクトのベクトルはセルによって近似され，その幾何的な近似は VA-File として作成される．探索時には，VA-File を走査して探索すべきオブジェクトの候補点を作り，その後オブジェクトの空間ベクトルが近傍点であることを確認するために，ベクトルファイルにおける候補集合に関する部位だけを参照する．3 次元程度の空間では，探索すべき候補点を絞ることが困難であり十分な性能を得ることはできないが，多次元であればあるほど VA-File による効果がある．文献 29) では，6 次元以上のベクトル空間における探索で VA-File が X-tree や R*-tree を上回る性能を発揮すると報告されている．

3.2.3 情報検索単位の提示

利用者がキーワードによる問合せを検索システムに入力したとき，検索システムはまず入力されたキーワードから問合せ特徴ベクトルを生成する．そしてその問合せ特徴ベクトルと要素特徴ベクトルの類似度を計算し，類似度の高いものから順に検索結果として返す．この際，類似度を計算する際の計算コストを削減するために 3.2.2 項で述べた VA-File による索引を利用して類似度を計算すべき要素特徴ベクトルの絞り込みを行っている．

問合せ Q が与えられたとき，検索システムは以下のような手順で利用者に検索結果を返す．

(1) **問合せ特徴ベクトルを生成**
まず，XML 文書の要素特徴ベクトルと同じ基底を持つ問合せ Q の特徴ベクトル $\boldsymbol{q}$ を生成する．具体的には，要素特徴ベクトル $\boldsymbol{F}(e_{ij})$ が検索対象である全 XML 文書中に出現する単語の出現頻度を基に計算されているため，$\boldsymbol{q}$ もキーワード中にそれら出現単語 $w_1, \ldots, w_n$ が出現しているか否かで決定される．すなわち，問合せ特徴ベクトル$\boldsymbol{q}$ は，

$$\boldsymbol{q} = (q_{w_1}, q_{w_2}, \ldots, q_{w_n}) \quad (3)$$

と表現される．ただし，q_{w_k} $(k = 1, 2, \ldots, n)$ は，

$$q_{w_k} = \begin{cases} 1 & \text{問合せに } w_k \text{が含まれている場合} \\ 0 & \text{問合せに } w_k \text{が含まれていない場合} \end{cases}$$

である．

(2) **問合せベクトルと要素特徴ベクトルとの類似度の計算**
3.2.2 項で述べたように VA-File による索引を利用して類似度計算を行う要素特徴ベクトルの絞り込みを行う．そして，絞り込まれた要素特徴ベクトル $\boldsymbol{F}(e_{ij})$ と問合せベクトル $\boldsymbol{q}$ との類似度を計算する．類似度の計算はベクトル空間モデルでは通常，次式のような 2 つのベクトル

☆ 単語数にはこれといって基準があるわけではない．しかし，文脈の変化を把握するために必要な文の数として，3〜5 文，すなわち 30〜50 単語は必要であるといわれている[16]．

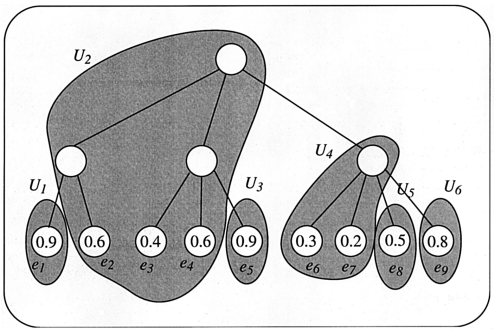

図 3 部分文書の生成
Fig. 3 Forming subdocument.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT spec
          (title, authlist, abstract, status, body, back)>
<!ATTLIST spec w3c-doctype (cr|note|pr|rec|wd)       #IMPLIED
               other-doctype CDATA                   #IMPLIED
               status (int-review|ext-review|final) #IMPLIED>
<!ELEMENT title (#PCDATA) >
<!ELEMENT authlist (author+) >
<!ELEMENT author (name, affiliation?, email?) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT affiliation (#PCDATA) >
<!ELEMENT email (#PCDATA) >
<!ELEMENT abstract (p+) >
<!ELEMENT status (p+) >
<!ELEMENT body (div1+)>
<!ELEMENT div1 (div2*|p+)>
<!ATTLIST div1 id NMTOKEN #REQUIRED>
<!ELEMENT div2 (div3*|p+)>
<!ATTLIST div2 id NMTOKEN #REQUIRED>
<!ELEMENT div3 (p+)>
<!ATTLIST div3 id NMTOKEN #REQUIRED>
<!ELEMENT back (div1+)>
```

図 4 テストコレクション XML 文書の DTD
Fig. 4 DTD of our reference collection's XML documents.

の余弦によって計算される．

$$sim(\boldsymbol{q}, \boldsymbol{F}(e_{ij})) = \frac{\boldsymbol{q} \cdot \boldsymbol{F}(e_{ij})}{|\boldsymbol{q}||\boldsymbol{F}(e_{ij})|} \qquad (4)$$

(3) **部分文書の形成**

問合せベクトルに対する要素特徴ベクトルの類似度が計算されると，その類似度の値および XML 文書の論理構造を基にその要素が部分文書を形成するか否かを判断する．部分文書を形成するか否かの判断の手法は様々な手法が考えられるが，本論文では，最も単純な手法である隣接要素の類似度の最大値からある閾値 α 以内であれば 1 つの部分文書を形成すると定義した．つまり，要素 e_r $(r = 1, 2, \ldots)$ がある部分文書 D_s に含まれるためには，

$$sim_{\max} - (\boldsymbol{q}, \boldsymbol{F}(e_r)) \leq \alpha$$

である必要がある．ただし，$sim_{\max}$ は部分文書を構成する要素の類似度の中で最大の値，要素 e_r は部分文書 D_s を構成する連続した文書要素である．たとえば，ある問合せに対し各要素の類似度が**図 3** のように計算されたとする．$\alpha = 0.2$ であるとすると，要素 e_2，e_3，e_4 および e_6，e_7 が部分文書を形成する．結局，U_1 〜U_6 の 6 個の情報検索単位がこの XML 文書から抽出される．

(4) **情報検索単位の提示**

最終的に検索結果として返される XML 文書の要素やその部分文書が情報検索単位として得られるので，それらの類似度を計算する．情報検索単位が文書要素の場合は，問合せに対する類似度が情報検索単位の類似度となるが，部分文書の場合は，部分文書を構成する文書要素の類似度から再計算をする必要がある．部分文書の類似度の計算方法は様々な方法が考えられるが，本研究では次の 2 つの手法を提案する．

(a) 部分文書を構成する末端要素の特徴ベクトルと問合せ特徴ベクトルの類似度の平均値を部分文書の類似度とする方法．

(b) 部分文書の末端要素の特徴ベクトルの和と問合せ特徴ベクトルとの類似度を計算し，それを部分文書の類似度とする方法．

こうして計算された類似度を基にその値の高いものから順に並べ，利用者に検索結果として返す．

4. 評価実験

本章では，提案・実装した XML 文書検索システムの有用性を評価するために行った実験について述べる．

4.1 実験方法

本実験は，情報検索単位を基に検索する手法が，従来の要素を基に検索するパッセージ検索と比較してどれほど有効かを検証するために行った．

実験に使用したデータは，W3C の Technical Reports and Publications の Web ページ[30]から得られる HTML 文書を，**図 4** に示す DTD に従って XML 文書に変換したものを利用した．この DTD は W3C が記述する XML の DTD である XMLspec[3]のサブセットである．これは，3.1 節で述べたノード間意味関係と 3.2.1 項で述べた知見を考慮し，かつ文書要素内に比較的多くの単語を含んだ要素だけを扱った DTD となっている．具体的には，生成された XML 文書の `abstract`，`body`（本文），`back`（付録）要素を検索の対象とし，また，章全体を表す `div1` 要素，節全体を表す `div2` 要素，項全体を表す `div3` 要素の要素名は考慮していない．しかし，見出しを表現しているそれら要素の属性値は，文書要素の内容を要約していることから，その文書要素に子要素がある場合はすべての子要素に，また子要素がない場合はその要素自身に出現する単語として扱っている．

実験環境には米国サン・マイクロシステムズ Ultra Enterprize 450（UltraSPARC-II 296 MHz × 2，主記憶 1 GByte）を用いた．この環境下で，XML 文書からの特徴量抽出やその特徴ベクトルの生成，索引づけ，そして問合せに対する検索結果の提示のプログラムの開発を perl5（patchlevel: 5，subversion: 3）を用いて実装した．

独自に検索対象データを作成した理由は，検索システムの評価に一般的に用いられるテストコレクションが，本実験で用いるような XML 文書のものがいまだ公開されていないからである．したがって，我々の所属している研究室で独自のテストコレクションを作成した上で，検索システムの評価のために適合率および再現率を計算した．生成した XML 文書ファイルは 17 個，用意した問合せ/解答セットは以下の 3 つである．

- **問合せ/解答セット 1**
  質問文 XHTML の互換性の問題は将来どう解決されるのか？
  問合せキーワード XHTML compatible issue future.
  解答 `REC-xhtml1-20000126.xml` の 5 章および 6 章．
- **問合せ/解答セット 2**
  質問文 XML のエンティティの文字コードは UTF-8 のほかに何が利用できるのか？
  問合せキーワード XML entity character encoding UTF-8.
  解答 `REC-xml-19980210.xml` の 2.2 節および 4.3.3 項，付録 F 章と，`REC-xml-20001006.xml` の 2.2 節および 4.3.3 項，付録 F.1 項．
- **問合せ/解答セット 3**
  質問文 XML の要素型名や属性名に使える文字には何があるのか？
  問合せキーワード attribute element type name character code
  解答 `REC-xml-names-19990114.xml`の 2，3，4 章および付録 A.3 項と `REC-xml-19980210.xml` および `REC-xml-20001006.xml` の 2.2，2.3，3.0（3 章の枕詞の部分），3.1 節および付録 B 節．

4.2 実験結果

4.1 節で用意した問合せ/解答セットを検索システムに入力すると，テストコレクションの XML 文書から 1,137 個の文書要素が抽出され，検索結果として XML 文書の要素もしくは部分文書が問合せに対する類似度順に返された．このランキング結果を利用すると，検索システムを評価するために使用される再現率–適合率グラフが得られる．

問合せ/解答セットを以下の (1)〜(3) に示す検索システムに入力した際に得られる再現率–適合率グラフを**図 5**，**図 6**，**図 7**☆（図 7 のみ適合率の軸のスケールが異なる）に示す．なお，これらグラフを生成する際に使用した部分文書を形成時に利用する閾値 α の値は，後で述べるように実験から求めた 0.1 としている．

(1) XML 文書の要素を検索結果とするパッセージ検索を利用した検索システム
(2) 本研究で提案した情報検索単位を検索結果とし，その類似度の計算を 3.2.3 項で述べた (a) の手法で計算する検索システム
(3) 本研究で提案した情報検索単位を検索結果とし，その類似度の計算を 3.2.3 項で述べた (b) の手法で計算する検索システム

図 5〜7 のグラフを見れば分かるように，パッセージ検索をそのまま XML 文書に適用した手法 (1) に比べ，我々が提案した情報検索単位で XML 文書を検索する手法 (2) の方が検索精度が高い．我々の提案したもう一方の手法である手法 (3) の検索精度が低くなるのは，情報検索単位の評価値の計算に文書要素の評価値を用いずに行ったためであろうと考えられる．なぜなら，先行研究 14) において，「XML 文書の検索は全文検索による手法よりもパッセージ検索を適用した方が検索精度が良い」という報告を行っているが，この報告から部分文書全体の特徴ベクトルと問合せの類似度を計算するよりも，あらかじめ各要素特徴ベクトルから類似度を計算してから部分文書全体の類似度を求めた方が検索精度が上昇すると考えることができるからである．この実験結果は，それを裏付ける形となったと考えられる．

以上をまとめると，我々の提案手法である手法 (2) を適用すれば，検索結果のランキングの上位に，利用者の問合せに相応しい要素もしくは部分文書が多く含まれていると考えることができ，我々の提案手法の有効性がこれで確認することができたといえる．

ところで，部分文書を決める際の α の値を変化させたときの適合率–再現率グラフは**図 8** のようになった（検索手法 (2) を実装した検索システムにおいて問合せ/解答セット 2 を用いた場合）．図 5〜7 のグラフを描くときに用いた閾値 α の値は，図 8 のグラフを

☆ この再現率–適合率グラフは，文献 2) で説明されている Interpolated precision at 11 standard recall levels に従っている．

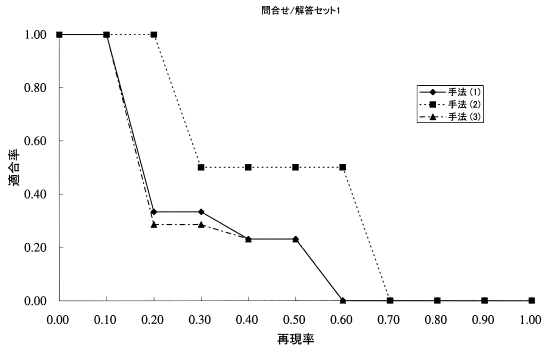

図 5 再現率–適合率グラフ（問合せ/解答セット 1）
Fig. 5 Recall-Precision graph (query/answer set 1).

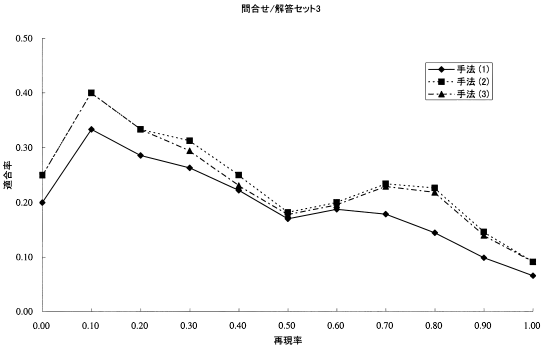

図 7 再現率–適合率グラフ（問合せ/解答セット 3）
Fig. 7 Recall-Precision graph (query/answer set 3).

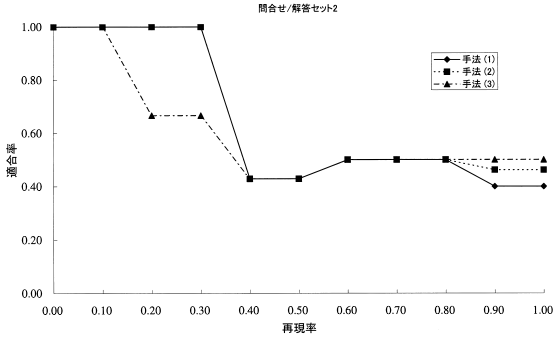

図 6 再現率–適合率グラフ（問合せ/解答セット 2）
Fig. 6 Recall-Precision graph (query/answer set 2).

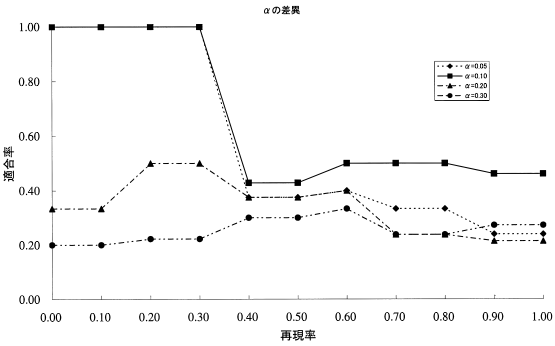

図 8 再現率–適合率グラフ（α を変化させた場合）
Fig. 8 Recall-Precision graph if we use different value of α.

得るための実験から決めたものである．しかし，この閾値 α は，テストコレクションを構成している文書セットが異なれば変化するものである☆．したがって，様々な文書セットに対応できるよう，閾値 α の値を自動で計算する手法を提案する必要があると思われる．

これらの実験結果から，今後の文書検索は文書，構造化文書の違いに関係なく検索を行う際には，文書をより細かな単位で分割し，その分割単位で文書の特徴抽出を行ったうえで，抽出された特徴を基に行われるであろうと予想できる．また，検索対象が構造化文書の場合は，その論理構造だけではなくノード間意味関係を考慮して文書から特徴抽出をしなければならず，本研究はそれらの知見を実証した一研究と位置づけることができる．

5. おわりに

本論文では，XML 文書を検索する際に生じる，あらかじめ利用者が XML 文書の構造を理解しておかなければ，利用者の問合せに最適な部分文書を検索することができないという問題点に注目し，XML 文書の検索に情報検索技術を適用することでその解決を図る提案を行った．また，提案手法を適用した検索システムを実装し，テストコレクションを利用した検索実験を行うことで提案手法の有効性を実証した．

本論文で提案した手法の利点としては，

(1) 現在の Web 検索エンジンの問合せ入力のように，キーワードを利用した問合せが可能であるため，XML 問合せ言語を利用する場合のように XML 文書の構造をあらかじめ理解しておく必要や，SQL や XQL など専門的な知識を必要とせずに利用者が検索システムを容易に利用できる．

(2) 巨大な XML 文書に対して検索を行う場合，検索結果が情報検索単位で返されるため，全文検索の手法に比べ，利用者が検索したい内容が記述されている部分を直接検索することが可能である．また，情報検索単位は，XML 文書要素の隣接関係やノード間意味関係を考慮して計算

☆ 文献 15) で行った予備実験では，テストコレクションに 8 文書利用したが，その際の閾値 α の値は 0.2 であった．

されているので，利用者は問合せの意味により類似した検索結果を得ることができる．

(3) 文書要素を検索単位としたパッセージ検索と比較した場合，検索結果に類似した要素が多く含まれていても類似している要素がグループ化されるので，それらすべてを1つずつ閲覧せずに済む．すなわち，利用者が検索結果全体を短時間に把握しやすい．

(4) XML 文書を要素ごとに分割して情報検索を行うために，検索結果自体の検索精度の向上が図られる．

などの点があげられる．

また，今後の課題として取り組むべき事項として以下のものがあげられ，これらについて今後取り組んでいく必要がある．

- 提案手法の評価を行うために，今回はテストコレクションを独自に生成し，それを基に評価実験を行ったが，4.1 節で述べたように，利用した要素はノード間意味関係や，比較的多くの単語を含む必要があるという知見を考慮，すなわち，章，節，項に限定している．しかし，検索対象とならなかった文書要素の中には，本来は検索対象となるような内容が存在するのも確かである．具体的にいえば，実験に使用した XML 文書の `author` 要素は検索対象に含まれるべきだと思われる．したがって，XML 文書検索の際に利用される文書要素の選別アルゴリズムを提案する必要がある．この手法の提案は，XML 文書の要素名を検索に利用するか否かを判定するアルゴリズムの実現にも深く関係があり，さらにノード間意味関係を解析するための重要なアルゴリズムであると考えている．また，XLink や XPointer も考慮し，文書内の構造だけではなく，文書外の構造に関しても考慮した検索手法を提案する必要もある．
- 本手法は，XML 文書の論理構造の一部，すなわち要素の隣接関係を意識した検索手法である．しかし，XML 文書の論理構造には隣接関係だけではなく，兄弟関係および親子関係も存在する．したがって，本手法でも要素間の親子関係と兄弟関係双方を考慮した機能を実現し，XML 問合せ言語で現在実現できている機能を，本手法においても実現できるように機能拡張を行う必要がある．具体的には，利用者が検索システムに入力した問合せキーワードをすべて含むような最小の部分文書を検索する XML 文書の検索手法が，Kinutani らによって提案されているが[18]，それぞれの手法の利点はそれぞれの手法の欠点を補うことが可能であるため，両手法を統合した XML 文書検索手法を提案する．このとき，類似度の値が近い値を持った文書要素が隣接していても，文書の構造，たとえば，章をまたがった場合のような場合はグループ化しないようにするアルゴリズムを組み込むことを必要がある．
- 本手法では，部分文書を検索する際に，利用者の問合せに類似した文書要素をグループ化する手法を採用している．しかし，要素が持つ問合せに対する類似度が近く，かつそれらが隣接しているという条件だけで部分文書を形成しているにすぎない．一方，構造化文書の構造から論理的に問合せに相応しい部分文書を抽出する構造化文書検索手法が存在する[20]．したがって，先に述べた新しい XML 文書検索手法を提案する際に，この要素技術を本手法に適用し比較実験を行う必要がある．また，利用者の問合せに対する情報検索単位の類似度の計算手法についても構造化文書の特徴を反映できるようなアルゴリズムを考える必要がある[21]．
- 文献 17) では，パッセージ検索の問題点として以下のものがあげられている．
  (1) 文書の解析，検索の際の計算量の増加
  (2) 文書要素の大きさの正規化
  (3) たまたま低類似度を持った文書要素を持つ部分文書のランキングの低下
  (4) 文書分割のアルゴリズム
  この問題点のうち，(1) および (2) の 2 点は，本手法を適用しても問題点となってくると予想される．提案手法の確立のためにもこれらの問題点についての対処が必要である．

謝辞 本論文で利用した XML 文書のテストコレクション作成を手伝っていただいた，奈良先端科学技術大学院大学情報科学研究科マルチメディア統合システム講座のスタッフ，学生，そして OB 諸氏に感謝いたします．また，論文をまとめるにあたり有益なコメントをしていただいた査読者の方々および DBWeb2000 のコメンテーターの方々に感謝いたします．

本研究の一部は，文部省科学研究費（課題番号：11480088，12680417，12780309），ならびに科学技術振興事業団戦略的基礎研究推進事業（CREST）によるものである．ここに記して誠意を表します．

参 考 文 献

1) Abiteboul, S., Cluet, S., Christophides, V.,

Milo, T., Moerkotte, G. and Siméon, J.: Querying Documents in Object Databases, *International Journal of Digital Libraries*, Vol.1, No.1, pp.5–19 (1997).
2) Baeza-Yates, R. and Ribeiro-Neto, B.(Eds.): *Modern Information Retrieval*, ACM Press (1999).
3) Bosak, J., et al.: Guide to the W3C XML Specification ("XMLspec") DTD, Version 2.1. http://www.w3.org/XML/1998/06/xmlspec-report-v21.htm (2000).
4) Bray, T., Sperberg-McQueen, M. and Paoli, J.: Extensible Markup Language (XML) 1.0. http://www.w3c.org/TR/REC-xml (1998). W3C Recommendation, 10-February-1998.
5) Chamberlin, D., Florescu, D., Robie, J., Siméon, J. and Stefanescu, M.: XQuery: A Query Language for XML. http://www.w3c.org/TR/xquery (2001). W3C Working Draft, 15-February-2001.
6) Chamberlin, D., Robie, J. and Florescu, D.: Quilt: An XML Query Language for Heterogeneous Data Sources, Lecture Notes in Computer Science (2000).
7) Clark, J. and DeRose, S.: XML Path Language (XPath) Version 1.0. http://www.w3c.org/TR/xpath (1999). W3C Recommendation, 16-November-1999.
8) Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R.: Indexing by Latent Semantic Analysis, *J. Am. Soc. Inf. Sci.*, Vol.41-6, pp.391–407 (1990).
9) Deutsch, A., Fernandez, M., Florescu, D., Levy, A. and Suciu, D.: XML-QL: A Query Language for XML. http://www.w3c.org/TR/NOTE-xml-ql/ (1998). Submission to the W3C, 19-August-1998.
10) Egnor, D. and Lord, R.: Structured Information Retrieval using XML, *Proc. ACM SIGIR 2000 Workshop on XML and Information Retrieval* (2000). http://www.haifa.il.ibm.com/sigir00-xml/final-papers/Egnor/index.html.
11) Florescu, D. and Kossmann, D.: Storing and Querying XML Data using an RDMBS, *IEEE Data Engineering Bulletin*, Vol.22, No.3, pp.27–34 (1999).
12) Florescu, D., Kossmann, D. and Manolescu, I.: Integrating Keyword Search into XML Query Processing, *Proc. 9th International World Wide Web Conference* (2000). http://www9.org/w9cdrom/324/324.html.
13) Fuhr, N. and Großjohann, K.: XIRQL: An Extension of XQL for Information Retrieval, *Proc. ACM SIGIR 2000 Workshop on XML and Information Retrieval* (2000). http://www.haifa.il.ibm.com/sigir00-xml/final-papers/KaiGross/sigir00.html.
14) 波多野賢治，森本考弘，吉川正俊，渡邉正裕，植村俊亮：アクセス権を考慮した構造化文書の検索手法の提案，第 3 回インターネットテクノロジーワークショップ論文集，日本ソフトウェア科学会 (2000).
15) 波多野賢治，渡邉正裕，吉川正俊，植村俊亮：要素特徴ベクトルを基にした部分文書構造の自動抽出，*Proc. DBWeb2000*, 情報処理学会シンポジウムシリーズ，Vol.2000, No.14, pp.213–220 (2000).
16) Hearst, M.: TextTiling: A Quantitative Approach to Discourse Segmentation, Technical Report S2K-93-24, University of California, Berkeley (1993).
17) Kaszkiel, M. and Zobel, J.: Passage Retrieval Revisited, *Proc. 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.178–185, ACM (1997).
18) Kinutani, H., Yoshikawa, M. and Uemura, S.: Identifying Result Subdocuments of XML Search Conditions, *Proc. 2000 Kyoto International Conference on Digital Libraries: Research and Practice*, pp.232–239 (2000).
19) Kiyoki, Y., Kitagawa, T. and Hayama, T.: A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning, *ACM SIGMOD Records*, Vol.23, No.4, pp.34–41 (1994).
20) Lalmas, M.: Dempster-Shafer's Theory of Evidence applied to Structured Documents: Modelling Uncertainty, *Proc. 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.110–118, ACM (1997).
21) Pahlevi, S. and Kitagawa, H.: Similarity Search of XML Documents Based on Tag Structures and Contents, 情報処理学会第 61 回全国大会論文集 CD-ROM (2000).
22) Robie, J., Lapp, J. and Schach, D.: XML Query Language (XQL). http://www.w3c.org/TandS/QL/QL98/pp/xql.html (1998).
23) Salton, G., Allan, J. and Buckley, C.: Approaches to Passage Retrieval in Full Text Information Systems, *Proc. 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.49–58 (1993).
24) Salton, G. and Lesk, M.: Computer Evaluation of Indexing and Text Processing, *J. ACM*,

Vol.15, No.1, pp.8–36 (1968).
25) Shanmugasundaram, J., Tufte, K., Zhang, C., He, G., DeWitt, D. and Naughton, J.: Relational Databases for Querying XML Documents: Limitations and Opportunities, *Proc. 25th International Conference on Very Large DataBases*, pp.302–314 (1999).
26) Shimura, T., Yoshikawa, M. and Uemura, S.: Storage and Retrieval of XML Documents using Object-Relational Databases, *Proc. 10th International Conference on Database and Expert Systems Applications*, LNCS, Vol.1677, pp.206–217, Springer-Verlag (1999).
27) Tajima, K., Hatano, K., Matsukura, T., Sano, R. and Tanaka, K.: Discovery and Retrieval of Logical Information Units in Web, *Proc. 1999 ACM Digital Library Workshop on Organizing Web Space*, pp.13–23 (1999).
28) 渡邉正裕，波多野賢治，吉川正俊，植村俊亮，中村　均：XPath を用いた文字列検索とベクトル検索の統合について，*Proc. DBWeb2000*, 情報処理学会シンポジウムシリーズ，Vol.2000, No.14, pp.349–356 (2000).
29) Weber, R., Schek, H.-J. and Blott, S.: A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces, *Proc. 24th International Conference on Very Large Databases* (1998).
30) World Wide Web Consortium: W3C Technical Reports and Publications. http://www.w3c.org/TR/.
31) XML Global Technologies: GoXML.com—XML Search Engine—Search and index XML documents. http://www.goxml.com/.

(平成 12 年 12 月 20 日受付)
(平成 13 年 4 月 11 日採録)

（担当編集委員　安達 淳）

波多野賢治（正会員）

1995 年神戸大学工学部計測工学科卒業．1999 年同大学院自然科学研究科博士後期課程修了．博士（工学）．同年奈良先端科学技術大学院大学情報科学研究科助手，現在に至る．XML データベース，情報検索に関する研究に従事．ACM 会員．

渡邉　正裕（正会員）

1994 年京都大学理学部宇宙物理学教室卒業．1999 年奈良先端科学技術大学院大学情報科学研究科博士後期課程単位取得退学．同年文部省国立特殊教育総合研究所（現，独立行政法人国立特殊教育総合研究所）入所，現在に至る．情報検索，情報可視化に関する研究に従事．ACM 会員．

吉川　正俊（正会員）

1980 年京都大学工学部情報工学科卒業．1985 年同大学院工学研究科博士後期課程修了．工学博士．同年京都産業大学計算機科学研究所講師．同大学工学部助教授を経て，1993 年より奈良先端科学技術大学院大学情報科学研究科助教授，現在に至る．1989〜90 年南カリフォルニア大学客員研究員．1996〜97 年ウォータルー大学客員准教授．2000 年から国立情報学研究所ソフトウェア研究系客員助教授．XML データベース，多次元空間索引等の研究に従事．電子情報通信学会，ACM，IEEE Computer Society 各会員．

植村　俊亮（正会員）

1964 年京都大学工学部電子工学科卒業．1966 年同大学院工学研究科修士課程修了．同年通産省工業技術院電気試験所（現，電子技術総合研究所）入所．1988 年東京農工大学工学部数理情報工学科教授を経て，1993 年奈良先端科学技術大学院大学情報科学研究科教授，現在に至る．工学博士．1970〜71 年マサチューセッツ工科大学客員研究員．データベースシステム，自然言語処理，プログラム言語の研究に従事．電子情報通信学会，ACM，IEEE 各会員．