情報学基礎　30−1
（1993．7．26）

全文検索によるCD-ROMガイドブックの内容検索

伊藤 史朗，　酒井 桂一，　山田 雅章，
小森 康弘，　上田 隆也，　藤田 稔

キヤノン(株) 情報システム研究所

文書の一部をフィールドに分け，その属性と型をテキストモデルにより定義して，文書の内容を反映した全文検索を行なう手法を提案する．本手法は，文書の一部に対するデータモデルを構築していると考えることができ，フィールドの型に応じて，整数の大小関係など検索語以外による条件を使った検索を文書を加工することなく行なえる．本手法では，オブジェクト指向言語による再利用可能な型別の検索クラスを用意して，インデクス検索，フルテキストサーチ，後処理を効率的に組み合わせている．また，CD-ROMで電子出版されている旅行ガイドブックを本手法により検索して，旅行情報の質問応答を行なう音声対話システムTARSANを作成した．

A Full-text Retrieval Method for the Contents of Guidebooks on CD-ROMs

Fumiaki ITOH,　Keiichi SAKAI,　Masayuki YAMADA,
Yasuhiro KOMORI,　Takaya UEDA,　Minoru FUJITA

Information Systems Research Center, Canon Inc.
890-12 Kashimada, Saiwai-ku, Kawasaki-shi, Kanagawa 211, JAPAN

A full-text retrieval method for the contents of semi-structured documents is proposed. The document is divided to fields, whose attributes and types are defined with "text model". The conventional full-text search is expanded in order to retrieve the field with not only keywords but other conditions according to its type. It is considered that a partial data model is constructed for the text. An efficient integration of index search, full-text search and post-processing is realized with reusable classes written in object-oriented language. The effectiveness is demonstrated with TARSAN, a speech conversation system for retrieving travel information from guidebooks on CD-ROMs.

1 はじめに

文書を電子化して扱う環境の発展に伴って，電子化文書の作成，流通，蓄積が進んでいる．これらの文書は，記録メディアの大容量化に伴って全文データとしての蓄積が進む一方であり，これを全文データベースとして扱うことも増えている．なお，一般に全文データベースには図表などのデータも含まれるが，現段階ではテキストデータだけを対象にすることが多い．そこで，本稿でもテキストデータだけから構成される文書を対象として扱う．

文書は古くから情報を表現してきた．今でも，情報の伝達，蓄積のために重要な役割を果たしている．いわば，文書は人間に最も親しまれている情報の表現形式である．情報を電子化するにあたっても，文書をそのまま電子化することも多い．なぜならば，文書はそのままの形で人間が見ることができ，作成や修正も簡単である．また，プレーンテキストであれば，特定のアプリケーションに依存したデータフォーマットではないので流通がしやすい．このような文書の特長を考えると，文書を直接利用しながら，そこに記述されている情報を利用できる枠組みを実現することが重要である．

文書に対する検索を考えた場合，従来いわゆる情報検索が行なわれてきた．情報検索では，文書が表現している ill-form な情報を検索することが中心となっている．言い換えると，データベースでいうところのデータモデルが (完全には) 構築できない情報の検索であり [1]，あるいはデータではなく概念の検索である [2] とも言える．そのため，情報検索では，文書自体が検索の単位になり，その中に記述されている細かい情報を検索することは行なわれてこなかった．全文データに対する情報検索である全文検索でも同じである．検索システムは，人間が望んでいる情報が記述されているであろう文書を取り出すだけであり，取り出された文書を人間が読んで解釈して，その中に記述されている細かい情報を取り出す．

我々は，文書であっても，その一部に対して部分的なデータモデルを作ることができるのではないかと考えた．すなわち，文書全体の検索だけではなく，文書の内容を細かい情報に分けて，それぞれの内容に応じた検索を行なうことを考えた．今回，我々は CD-ROM で出版されているガイドブックに注目した．ガイドブックでは，ある事柄を説明する定型的な情報が記述されていることが多い．旅行のガイドブックでは，名所などの場所，入場料などが記述されている．映画のガイドブックでは，映画の制作年，監督などが記述されている．これらの部分については，データモデルを構築することができる．このようなデータモデルを構築できる部分に対して文書の内容を検索する手法を提案する．

本手法では，全文検索の一手法であるフルテキストサーチを利用している．フルテキストサーチでは，文書の全文情報を直接利用して検索を行なう．そのために，文書中の情報を取り出して関係データベースなどで扱う方法よりも計算コストが高くつく．しかし，計算機の処理速度の進歩やフルテキストサーチ手法の進展により，実用的な時間内で文書を直接処理できるようになってきている [3]．本手法では，従来の全文検索を拡張して文書の内容検索を実現するために，テキストモデルで文書の内容を定義している．文書の構造を利用して文書の一部をフィールドに分け，フィールドの属性と型をテキストモデルにより定義する．フィールドの型に応じて，整数の大小関係など検索語以外による条件を使った検索を行なっている．本手法の実現にあたっては，オブジェクト指向言語を用いて型ごとに検索クラスを作成することで，インデクスサーチ，フルテキストサーチ，後処理を効率的に組み合わせている．

上記のように，本手法では，いわばデータモデルを構築できる部分に対する検索とその他の部分に対する従来からの情報検索を同時に実現している．しかも，文書からデータベースを作成するなどの加工を行なうことなく，文書を直接用いて検索を実現できる．実際，本手法の適用例である旅行情報の質問応答を行なう音声対話システム TARSAN では，CD-ROM 上の旅行ガイドブックを直接利用している．このシステムは，文書を直接利用できる本手法の有効性を示している．

以下，2 章では本手法が利用している全文検索について説明する．3 章では，本手法で実現されている文書の内容検索について説明する．4 章では，本手法の実現方法について説明する．5 章では，本手法の適用例である音声対話システム TARSAN を説明する．

2 全文検索の各種手法

全文データの蓄積が進むにつれて，全文の情報を使った全文検索が行なわれるようになってきた．全文データを用いた検索を行なう時，検索条件と検索対象の文書との間で，単語[1]のマッチングを取る場合と意味のマッチングを取る場合がある．前者は，検索者が指定した単語が現れる文書が，検索者が意図している文書であるという考えに基づいている．後者は，それでは不十分であっ

[1] 意味を持たない文字列の場合もある．

て意味を考慮しなければならないとする考えに基づいている．しかし，現状では，対象領域固有の知識の必要性や意味解析の不完全さなどから後者は現実的な方法ではない．

前者の検索を実現する方法として，インデクス方式とフルテキストサーチ方式がある．

インデクス方式は，文書中の単語あるいは文字列のインデクスを作成して，インデクスによる検索を行なう方式である．インデクス方式は，検索速度が速い反面，インデクス作成という付加処理が必要であり，文書に加えてインデクスも保持しなければならない．なお，インデクス方式だけでは，不適切な文書も検索してしまう場合が多いので，フルテキストサーチ方式と組み合わせて結果を絞り込むことも多い．日本語は，簡単には単語に区切れないため，様々なインデクスが考案されている．まず，1文字，2文字などの部分文字列をキーとしたインデクスがある [4, 5]．キーの長さを越える文字列で検索する場合には，複数のインデクスを組みあわせて検索を行なう．また，文字列に変換をかけた上でキーを作成することも行なわれている [6]．インデクスの値は文書自体がほとんどであるが，文書中の位置まで保持する方法もある [7]．この場合は，インデクスだけで位置を考慮した検索条件の判定も可能になる．

フルテキストサーチ方式は，パターンマッチングにより直接全文データをサーチして，単語あるいは文字列が存在するかどうかを調べる方法である [8]．フルテキストサーチ方式では，事前にデータを加工する必要がなく，インデクスなどの付加データも必要がない．従って，文書が変更されても特に処理を必要としない．しかし，本質的に検索時間が文書量に比例するので，大量のデータの検索には向かない．検索速度をあげるために，パターンマッチングを行なう専用ハードウェアも作られている [9, 10, 11]．しかし，ハードウェアの能力を十分に生かすためには，記憶媒体にメモリやディスクアレイを用いなければならず，現状の計算機環境には適さない．

我々は，文書を加工せずに直接使って検索を行なうという目的から，ソフトウェアによるフルテキストサーチ方式を採用した．

3 ガイドブックの内容検索

3.1 テキストモデル

ガイドブックの内容検索を実現するために，テキストモデルを作成して文書の内容を定義する．以下，テキストモデルについて説明する．ここでは，例として旅行情報のガイドブックである「旅蔵」[17] のテキストモデルを用いている．

ドメイン

一つのガイドブック中に，異なる種類の対象についての記述が複数あることも多い．対象が異なると記述項目に違いが生じるので，内容の定義は個々に行なう．対象の違いによるグループをドメインと呼ぶ．「旅蔵」の例では，ドメインとして，温泉，ゴルフ場，博物館などがある．これ以後に説明する情報の定義は，全てドメインごとに行なう．

レコード

ガイドブックでは，ドメインに属する事象について様々な説明がなされている．一つの事象についての説明を行なっている部分をレコードとして区切る．例えば，温泉ドメインでは一つ一つの温泉についての説明がそれぞれレコードになる．レコードは，特定の文字列で区切る．テキストモデルで，レコードの区切り文字列を定義する．図1に，レコードの例を示す．

湯ノ花沢温泉
所在地：神奈川県箱根町
名称ヨミ：ユノハナザワオンセン
交　通：小田原駅バス５０分
宿泊施設 (軒数)：１
宿泊施設 (人数)：２４０
概　要：芦ノ湯の上にあり展望絶好。硫化水素泉５５～８５度。標高９４０m。
効　能：皮膚病
利用者数 (年間)：６１年 (１月～１２月) ２１，５０９人

図 1: レコードの例

フィールド

内容検索において，検索の単位となるのがフィールドである．フィールドは，レコードの中の部分文字列であり，次の二つの方法で識別する．

- 特定の文字列 (フィールドキー) から改行コードまたは別のフィールドキーまで
- レコードの先頭からの特定行数目の行

図1のレコードに対するフィールドの定義例を図2に示す．

属性

フィールドに記述されているデータが表現している内容を属性として定義する．属性には任意の語句を使用

フィールド 1	1行目
フィールド 2	「所在地：」から改行コードまで
フィールド 3	「名称ヨミ：」から改行コードまで
フィールド 4	「交　通：」から改行コードまで
フィールド 5	「宿泊施設 (軒数)：」から改行コードまで

図 2: フィールドの定義例

することができる．フィールドの内容を的確に表す語句が望ましい．ただし，属性を指定するとフィールドが一意に定まるようにする必要がある．検索にあたっては，条件の指定と内容の取り出しを属性を用いて指定する．なお，個々のレコードにおいて，ある属性が示すフィールドの値をその属性の属性値と定義する．図 2に示したフィールドに対する属性の定義例を図 3に示す．

フィールド 1	名称
フィールド 2	所在地
フィールド 3	名称読み
フィールド 4	行き方
フィールド 5	施設数

図 3: 属性の定義例

型

同一の属性に対する属性値は，同じ型のデータになる．ここでいう型とは，文字列，整数といった型ではなく，意味まで考慮した型である．例えば，「所在地」属性は「地名」という型のデータで記述されると考える．型は，予め用意されている型から選ぶ．また，新しく型を定義してもよい．この場合は，その型に対して，後述する検索クラスをインプリメントしなければならない．図 3に示した属性に対する型を図 4に示す．

属性	型
名称	基本
所在地	地名
行き方	交通手段
施設数	整数

図 4: 型の定義例

以上のテキストモデルは，既定のフォーマットで記述されたテキストファイルとして保持される．テキストモデルの作成は，現状では人手で行なっている．

3.2 検索条件と検索結果

本手法による内容検索の検索条件と検索結果について説明する．テキストモデルを定義することで，テキストモデルに基づいて以下のような検索を実現できる．

検索はドメインごとに行なう．検索条件は，属性単位で指定する．また，属性に対する条件の AND/OR 条件を指定することができる．属性に対してどのような条件を指定できるかは型によって異なる．基本型の場合は，フルテキストサーチと同じで検索語を指定する．指定された検索語の論理関係，距離関係などの条件も指定できる．地名型の場合は地名を指定する．地名は都道府県名，市町村名の他に地域名も指定できる．例えば，「湘南」という地域名を指定すれば，「湘南」地域に該当する市町村のレコードを検索できる．整数型の場合は，整数値の大小関係を条件とすることができる．条件の指定方法も数式だけではなく，「10 以上 20 未満」や「10 ～ 20」といった自然言語による指定もできるようになっている．また，文書中の全角，半角コードの違いや「,」の有無などの表記法の違いも吸収する．なお，フィールドとして取り出せない部分に対しては，全体という属性を定義することで，いわゆる情報検索としての検索も可能になる．

検索結果として，検索されたレコードを特定するポインタの他，指定した属性の属性値を取り出すことができる．

検索条件と検索結果の入出力は，検索テーブルと呼ぶ構造体を用いて行なっている．一つのテーブルで，1 ドメインに対する検索条件を記述し，検索結果を同じ構造体で返す．図 5に条件テーブルと結果テーブルの例を示す．

R	
C	分類	温泉
C	所在地	神奈川県
C	施設数	10以下
R	名称	
R	所在地	

条件テーブル

R	xxxxxxx
C	分類	温泉
C	所在地	神奈川県
C	施設数	10以下
R	名称	湯ノ花沢温泉
R	所在地	神奈川県箱根町

結果テーブル

図 5: 検索テーブルの例

先頭行は，テーブルフラグとポインタを表している．テーブルフラグは，検索テーブルの役割を示すフラグであり，検索を行なう時は “R” を用いる．その他に，どのようなドメインや属性があるかの問い合わせを行なうことができる．それ以降の行は，各属性に対する検索条件と結果要求を表している．先頭のアルファベットは，属

性に対する検索条件の指定か結果の要求かを示すフラグである．“C” が条件指定を，“R” が結果要求を示している．次に属性を指定する．ここで,「分類」となっている行だけは，属性ではなくドメインの指定に用いられている．次に，条件指定の場合は検索条件が入る．結果要求の場合は，取り出された属性値が入る．

3.3 ブロックインデクス

フルテキストサーチでは，検索時間が検索対象である文書の量に比例するので，実用時間内で検索ができるように検索対象を絞り込むことが必要である．このとき，ある時間内で検索できる量まで絞り込めればよく，過剰に絞り込む必要はない．このことを考慮して次のブロックインデクスを考えた．

一般に，文書中のレコードは，文書の著作者が考えた分類に従って並んでいる．同じ分類に属するレコードは，連続して並んでいる．これを，ブロックと呼ぶことにする．ブロック自体もある基準に従って順序付けられて並んでいる．旅行ガイドブックの例では，各地域に関する情報を記述したレコードがまとまって，地域ごとにブロックを構成することが多い．そして，地理的な順序関係に従ってブロックが並べられる．北から順に並べていくことなどはよくある例である．ブロックに分ける際に用いられた分類は，著作者が最も重要と考えた分類であると考えられる．ガイドブックを作成する時には，一番よく使われる分類に従って記述することが多い．旅行ガイドブックは，旅行先の地域を限定して調べることが一番多いと考えるから上記のような構成になる．

この分類は，文書のいずれかの属性を反映していることが多い．その属性について検索条件を指定した場合には，条件に該当するブロックだけを検索すれば検索時間の短縮につながる．そこで，属性値をキーとしてブロックを値とするブロックインデクスを考えた．既存の属性に関係なくブロックが構成されている場合は，ブロックを分ける際に用いられた分類を疑似属性として扱うこともできる．

テキストモデルで，必ず条件として指定しなければならない属性を定義することもできる．この属性を必須属性と呼ぶ．ブロックインデクスの対象となっている属性を必須属性とすると，検索の高速化に有効である．

ブロックインデクスは，元々のガイドブックの目次に相当する．また，テキストモデルは，ガイドブックの凡例に相当する．人間がガイドブックを読む時にも，目次で読む範囲を定めて，どのような内容が書いてあるかを凡例から判断する．本手法は，この行為を計算機上で実現したものと考えることもできる．

4 全文検索による内容検索の実現

3 章で述べた内容検索を実現するために，フルテキストサーチによる検索とそれ以外の検索を我々はオブジェクト指向言語を用いて組みあわせる手法を考案した．本手法では，型ごとに検索クラスを用意し，フルテキストサーチの検索条件を設定するオブジェクトと，フルテキストサーチで得られた結果に対して後処理で条件を判定する後処理オブジェクトとを検索条件に従って生成する．

図 6 に検索手法の概要を示す．これらは全て Objective C 言語により記述している．以下，各部について説明する．

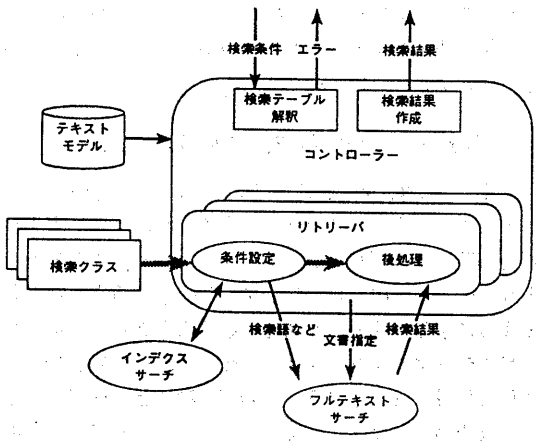

図 6: 内容検索の実現手法

4.1 コントローラ

コントローラは，全体の制御を行なうオブジェクトで，検索条件と検索結果の入出力も行なう．コントローラは，条件テーブル中のドメインや属性の指定が正しいかどうかのチェックを行なう．正しい場合は，ドメインを指定してリトリーバオブジェクトを生成する．正しくない場合は，エラーテーブルを返す．エラーテーブルでは，検索者に対して検索条件の変更を要求できる．必須属性が定義されている場合は，必須属性の条件が指定されているかどうかのチェックも行なう．必須属性が指定されていない場合は，必須属性に対する条件を指定するように要求するエラーテーブルを返す．検索の結果，検索条件を満足するレコードが検索された場合は，条件テーブルで指定された属性の属性値を取り出して，結果テーブル

を作成する．

4.2 リトリーバ

リトリーバは，指定されたドメインのテキストモデルを参照して，属性ごとに検索条件の処理を行なう．

まず，属性の型に従って，その型の検索クラスから条件設定オブジェクトを生成する．条件設定オブジェクトは，ブロックインデクスが存在するときは，条件に合致するブロックを取り出しリトリーバに渡す．条件設定オブジェクトはさらに，フルテキストサーチの検索語，AND/OR 条件などを設定する．

属性ごとの条件を全て処理した後は，リトリーバは，渡されたブロックから，各属性の AND/OR 関係を考慮して，フルテキストサーチにかけるブロックを決定する．そして，そのブロックに対してフルテキストサーチを行なう．フルテキストサーチの結果，レコードが検索されると後処理オブジェクトが残りの条件を満足しているかどうかのチェックを行なう．

4.3 検索クラスと条件設定・後処理オブジェクト

検索クラスは，条件設定オブジェクトと後処理オブジェクトを生成するクラスである．定められた型に対する条件を解釈して，ブロックインデクスサーチ，フルテキストサーチ，後処理のいずれか適する処理で検索を行なえるようにする．検索クラスを作成するときに，対象とする型のデータのシンタックス，セマンティクスを考えて，上記の処理に振り分ける．いわば，型ごとに検索を行なう知識を検索クラスとしてインプリメントしていると考えられる．

よく使われる型に対しては，検索クラスを予め用意することで，テキストモデルで型を指定するだけで検索が実現できる．すなわち，一度作成した検索クラスは簡単に再利用ができる．新たに検索クラスを作成する場合でも，オブジェクト指向言語のインヘリタンス機能により，型に固有の部分をインプリメントするだけでよい．もちろん，新たに作成した型の検索クラスも再利用可能になる．

4.3.1 ブロックインデクスサーチ

検索クラスからは，まず条件設定オブジェクトが生成される．条件設定オブジェクトは，対象としている属性に対する検索条件を渡される．その属性にブロックインデクスがある場合には，どのようなキーがインデクスに用意されているかを調べて，条件に応じたキーを生成する．

例えば，地名型の検索クラスで，ブロックインデクスのキーが都道府県名であると仮定する．検索条件が，市町村名で指定された場合には，地名シソーラスを用いて上位の都道府県名をインデクス検索のキーとする．その結果，フルテキストサーチでは，指定された市町村以外のレコードも検索することになる．しかし，インデクスで完全に絞り込まなくても，フルテキストサーチが許容時間内ですむだけの量に絞り込めればよいので，問題はない．

4.3.2 フルテキストサーチ条件設定

条件設定オブジェクトは，フルテキストサーチにより検索できる条件に対して，検索語と検索語の AND/OR 関係などの条件をフルテキストサーチオブジェクトに対して設定する．

フルテキストサーチオブジェクトは，AC 法 [12] の日本語テキスト用の拡張である SA 法 [13] をベースにしたパターンマッチングにより検索語の検出を行なう．また，この方法では，レコード，フィールドの切り出しを行なう文字列のパターンマッチングも同時に行なうことができるので，一回のサーチでレコードやフィールドの切り出しと検索語のパターンマッチングを行なっている．検索語間の AND/OR 条件などの判定は，AND/OR ノードのネットワークを作成し，見つかった検索語をトークンとしてネットワークに流すことで高速な判定を実現している．検索対象のテキストの保持にはリングバッファを用いているので，後処理条件の判定や結果の属性値の取り出しなどのときに，必要なデータをバッファから取り出すことができる．

4.3.3 後処理条件判定

指定された条件の判定を，全てフルテキストサーチで行なうことができる場合は，条件の設定を行なった後，条件設定オブジェクトは消滅する．フルテキストサーチだけでは判定できない場合は，条件設定オブジェクトは，後処理オブジェクトに置き換わる．後処理オブジェクトは，フルテキストサーチにより検索されたレコードに対して，対象としている属性値を取り出して，条件の判定を行なう．例えば，整数型の属性値の大小関係の判定は，属性値を整数に変換して判定を行なう．

生成された後処理オブジェクトはリトリーバによって管理される．リトリーバは，フルテキストサーチによって検索されたレコードから，後処理オブジェクトが必要とする属性値を取り出し，後処理オブジェクトに条件の判定を行なわせる．そして，複数の後処理オブジェクト

の判定結果をもとに，そのレコードを最終的な検索結果とするかどうかを決定する．

5 CD-ROMガイドブックの検索事例 —音声対話システムへの応用—

5.1 TARSANの概要

本手法による検索の適用例として，TARSAN[2]を説明する．TARSANはCD-ROMで電子出版されている旅行情報ガイドブックの全文データを検索して，その情報をもとに質問応答を行なう音声対話システムである[14]．本手法を用いることで，特別なデータベースを作成することなく，音声対話の情報源を提供している．

図6にTARSANの構成を示す．本手法は，この中の全文検索部で用いられている．

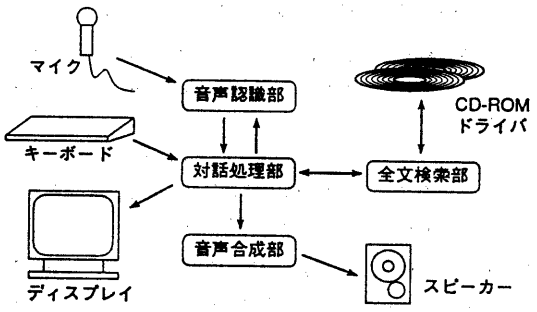

図7: TARSANの構成

TARSANは，マイクから入力される音声を音声認識により文字列に変換する[15]．キーボードから直接文字列として入力することもできる．音声認識できる語彙が限られているので，音声認識ができない語句はキーボードから入力する．対話処理部では，入力文を解析して全文検索部に対する検索条件を生成する．入力が不完全であったり，全文検索部からエラーテーブルが返ってきた場合は，問い返しを行なう．全文検索部がCD-ROM上の文書を直接検索し結果を返す．検索結果に基づき対話処理部が応答文を生成する．応答文はディスプレイに表示するとともに，音声合成部に送られ音声出力を行なう[16]．ハードウェア構成は，3台のNeXTワークステーションで音声認識部を実現し，1台のNeXTで残りの部分を実現している．

[2]Travel Assistant system composed of information Retrieval, Speech conversation Access and Natural-language processing

音声認識ができる単語数には限りがあるので，TARSANでは対話の局面に応じて認識可能な語彙を動的に切り替えている．例えば，当初地名は地方名と都道府県名しか認識できない．都道府県を指定して検索を行なった後は，検索結果が存在する市町村名が限られるので，それらの市町村名を認識語彙に加えることを行なっている．

TARSANでの対話例を図8に示す．

```
Usr: おはよう。
Sys: おはようございます。TARSANです。
Usr: 神奈川県にある温泉について教えて。
Sys: 箱根町に16件，厚木市に4件，横須賀市に2件，
     湯河原町に2件など，合計で31件あります。
Usr: 箱根町では。
Sys: 芦ノ湖温泉，木賀温泉，底倉温泉，宮城野温泉など，合計で16件あります。
Usr: その効能を示せ。
Sys: 病後回復・ストレス解消が1件，皮膚病が1件あります。
Usr: 効能が皮膚病では。
Sys: 他の条件，所在地が箱根町，を残しますか。
Usr: はい。
Sys: 湯ノ花沢温泉です。
Usr: 湯ノ花沢温泉の行き方は。
Sys: 湯ノ花沢温泉の行き方は小田原駅バス50分です。
```

図8: TARSANでの対話例

TARSANは，既存の文書が音声対話システムの情報源になり得ることを示している．それまでの音声対話システムの情報源は，実験規模の限られたものあった．それは，大量の情報をデータベース化するにはコストがかかるからである．TARSANでは，本手法により文書を直接利用するので，簡単にデータを利用することができる．TARSANが扱う情報は約8万件である．

5.2 TARSANでの全文検索

TARSANでは，「旅蔵」[17]と「JTB宿泊情報」[18]の二つの旅行ガイドブックのテキストモデルを作成して検索を行なっている．この二つのガイドブックから，温泉，ゴルフ場，ホテルなど24種類のドメインの情報を検索している．

上記の二つの文書とも所在地に従ってレコードを配列しているので，TARSANでは所在地を必須属性にしている．旅行情報の問い合わせでは，利用者に所在地を限

定してもらうことは自然である．そこで，都道府県ごとのブロックインデクスを作成して，検索時間を削減している[3]．CD-ROM の場合は，データ転送速度に比べてシーク速度がかなり遅いので，ブロックインデクスにより連続したレコードを検索できることは，高速化の上で大きな効果がある．1 都道府県を条件に指定した場合の検索時間は，平均 2 秒程度である．

また，音声認識語彙を動的に切り替えるために，検索結果の市町村名を取り出す必要がある．そこで，地名型のデータを取り出すにあたって，地名を解析して市町村名だけを取り出すこともできるようにしている．

6 おわりに

文書の一部に対してデータモデルを構築して，文書を加工することなく文書の内容検索を実現する手法を提案した．オブジェクト指向に基づき，データの型に応じて，フルテキストサーチとそれ以外の検索手法を使い分ける検索クラスを設計することで，よく利用される型に対する検索手法を再利用できる．また，CD-ROM により電子出版されている旅行ガイドブックの検索を音声対話を通して行なうシステム TARSAN を作成した．

CD-ROM で電子出版されているガイドブック，リファレンスブック 11 種について本手法が適用できるか検討してみた．その結果，8 種について本手法が適用できることがわかった．残りのうち，2 種はフィールドの切り出しができないなどの理由で本手法が適用できなかった．また，1 種はテキストがフィールドに分けるような構造になっておらず，テキストモデルを定義できなかった．

TARSAN で示したように，大規模なデータでなければ，文書を加工せずに直接検索しても十分な検索速度が得られるといえる．現在，CD-ROM 文書の検索速度は 150KB/s であり，これは CD-ROM ドライブのデータ転送速度と同じであり，I/O ボトルネックとなっている．最近は，転送速度が 2 倍，4 倍である CD-ROM ドライブもあるので，それらのドライブでも評価してみたい．

今後は，テキストモデル定義のサポート環境の整備，フィールドの切り出しの拡張などを行なって，多様な文書に対して本手法が適用できるようにしていきたい．

参考文献

[1] K Parsaye, M. Chignell, S. Khoshafian, and H. Wong: Text Management and Retrieval, *In Intelligent database*, Wiley, New York, 1989.

[2] 細野：情報検索理論・技法の問題点とその解決の方向, 情処研報, 情報学基礎, 24-5, 1991.

[3] 菊地, 小川, 高橋, 杉本, 金田：全文検索の技術動向とシステム事例, 情処研報, 情報学基礎, 92-FI-25, 1-8, 1992.

[4] 畠山, 浅川, 加藤: ソフトウェアによるテキストサーチマシンの実現, 情処研報, 情報学基礎, 92-FI-25, 19-25, 1992.

[5] 岩崎, 小川: 文字成分表による文字列検索の実現と評価, 情処研報, データベースシステム, 93-DBS-92, 1-10, 1993.

[6] 高田: 高速全文不完全一致検索システムの実現, 情処研報, 情報学基礎, 92-FI-25, 27-32, 1992.

[7] 菊池: 日本語文書用高速全文検索の一手法, 情処研報, 情報学基礎, 92-FI-25, 9-16, 1992.

[8] 有川他：テキストデータベース管理システム SIGMA とその利用, 情処研報, 情報学基礎, 14-7, 1989.

[9] 加藤, 藤澤, 大山, 川口, 畠山：大規模文書情報システム用テキストサーチマシンの研究, 情処研報, 情報学基礎, 14-6, 1989.

[10] 高橋, 山崎, 西塚, 本村：テキスト DB マシンと検索プロセッサのアーキテクチャ, 情処研報, 計算機アーキテクチャ, 91-ARC-90, 47-54, 1991.

[11] 菅野, 安藤, 伊藤, 田村, 鶴林, 早川：ワークステーション内蔵型フルテキストデータベースプロセッサ SDP, 情処研報, 計算機アーキテクチャ, 91-ARC-90, 55-62, 1991.

[12] A.V.Aho, and M.J.Corasick: Efficient string matching: An aid to bibliographic search, *Comm. ACM, 18,* 333-340, 1975.

[13] 篠原, 有川: 日本語テキスト用の Aho-Corasick 型パターン照合アルゴリズム, 情処研報, 自然言語処理, 52-4, 1985.

[14] 藤田他：音声対話と全文検索を利用した電子ガイドシステム (1) ーシステム概要ー, 情報処理学会全国大会, 46, 2E-4, 1993.

[15] 山田, 伊藤, 酒井, 小森, 大洞, 藤田：音声対話 CD-ROM 情報検索システムー対話中の未知語部の再評価アルゴリズムの提案ー, 信学技報, 音声, 発表予定, 1993.

[16] 酒井, 山田, 伊藤, 小森, 池田, 藤田：音声ガイドシステムにおける全文検索結果の利用法, 情処研報, 自然言語処理, 93-NL-95, 117-124, 1993.

[17]「旅蔵」電子ブック, (社) 日本観光協会, 1990.

[18] 中嶋編：JTB の「宿泊情報」(電子ブック), JTB 日本交通公社出版事業局, 1992.

[3]所在地として全国を指定することも可能である．