DEIM Forum 2010 A1-4

部分木を利用した適合部分検索手法の一考察

楊　　斐† 孫　　一† 清光 英成† 大月 一弘† 森下 淳也†

† 神戸大学大学院国際文化学研究科 〒 657–8501 神戸市灘区鶴甲 1–2–1

あらまし 検索システムに対して複数のキーワードが入力されるとき，それらの間の関連によってユーザが得たい情報は異なる．検索システムへ自然言語の文を入力とすることは表現力に富む反面，実現が容易でない．また，キーワードの列挙によりユーザが意図する情報を探し出すことにも限界がある．そこで，検索対象の論理構造や主題などのメタ情報を利用するとともにユーザが意図するキーワード間の係り受けを簡便に記述することで，適切なコンテンツを適切な分量で検索結果として得る方法を考察する．

キーワード キーワード感の関連, 半構造データ, グラフデータ

Fei YANG†, Yi SUN†, Hidenari KIYOMITSU†, Kazuhiro OHTSUKI†, and Jun-ya MORISHITA†

† Graduate School of Intercultural Studies, Kobe University
Tsurukabuto 1–2–1, Nada-ku, Kobe, 657–8501 Japan

1. ま え が き

ネットワーク速度の向上とセンサー技術の発達により単位時間内に収集できるデータ量が大幅に増加している．また，計算機や蓄積装置の性能が向上し処理・格納可能なデータ量が飛躍的に増加すると共に，生産されるデータ量も加速的に増加している．これらにより歴史・文化資産をデジタル化して保存・蓄積・活用するデジタルアーカイブもコンテンツの高精細化や提供方法を多様化することができるようになってきた．このような現状で，収集されたデータから必要なデータをを効率よく発見あるいは検索する要求が日増しに高まり，これらの要求を満たすための研究開発が盛んに進められている．情報爆発時代のデータには量的な問題だけでなく質的な問題もあり，その本質が明らかになりつつある．種類も規模も著しく違うデータをいかにして統一的に扱えるようにするのか，信頼性の不確かな大量のデータから利用可能で必要なデータだけを取り出す，或いは大量のデータ中に埋没している未知の知識を発見するなどである．

デジタルアーカイブに格納される資料はあるテーマに関するあらゆるものを網羅的に収集するため，その形態が多様性に富んでいる．また，ひとつの資料の中に個別の資料として扱うことが可能な構成要素を重層的に含むような複合型資料が多いという特徴がある．そこで，我々はこのような不定形複合型コンテンツに適したデータ格納手法ならびにその検索手法の開発を行ってきた．

本研究は，資料の任意の部分にメタデータを与え、ユーザが必要とするであろう部分のみを検索結果として提供することを目的とする．従来の検索方法は Web 検索であれば検索結果の単位が Web ページ，図書検索であれば検索結果の単位が図書であったが，資料の規模や形態がまちまちなアーカイブを検索してユーザが必要とするのは資料中のある部分のみであることが考えられるからである．本稿では特に，複数のキーワードに対してその間の関連を簡単に指定したときに有効な検索結果を求める関係と演算とを整理する．

今日，キーワード集合として列挙された全ての語句を含む Web ページや文書を検索するシステムが一般的になってきた．膨大なデータから必要とするデータの候補を選出する方法としての有用性は否定しない．しかしながら，必ずしもユーザの検索意図を反映した結果が得られるとはいえない．それは，全ての語句を同等に扱うためで，ユーザの検索意図を反映した結果を提供するためには，語句を接続する助詞的な要素を考慮した解釈が必要である．また，規模が著しく異なるデータを扱うため，ユーザに提供する検索結果が量的に柔軟性を持つ必要がある．

本研究は，蓄積されたデータからユーザの意図を反映した検索方式の実現を目的とし，ユーザは，

- データ構造の知識を持たなくても良い
- 思いついたキーワードを入力する以外の操作は最小限
- 言語や習慣に依存しない表現方法

であるような問合せ方式を目指している．そのため，part-of 木のようなユーザが容易に表現でき，かつシステムが変換しやすい抽象度の問合せスキームを提案する．

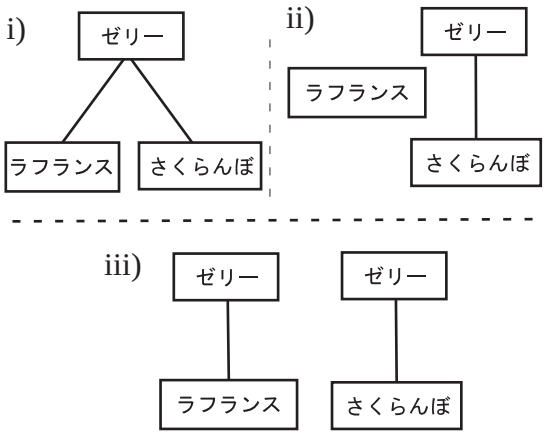

図 1　問合せ意図の part-of 関連
Fig. 1　A part-of Relationships on a Query Example

2. 問合せ式

「a と b」あるいは「a の b」を探すような問合せは “a と b” あるいは “a の b” というフレーズを見つけることで精度の高い検索結果候補を得ることができる．しかしながら，文書タイトルの文字列中に a を含み，文書コンテンツ中に b を含んで「a と b」あるいは「a の b」を表現しようとしているリソースは検索結果候補とはならない．このような文書構造を利用した意味表現に対してもキーワード間の関連性を反映した検索手法を実現できれば有用と考える．そこで，ユーザの検索意図を簡潔かつ明示的に表現できる問合せ式を考察する．

例えば，“ラフランス”, “さくらんぼ”, “ゼリー” というキーワードが入力された場合，全ての語句を含む web ページや文書を検索することで，ユーザの検索意図を反映した検索結果が提供できるとは限らない．このようなキーワードの入力に対して助詞を補うとすれば,「ラフランスとさくらんぼのゼリー」が自然であるが，ユーザの検索意図を確実に把握できたとは言えない．この自然言語文も

i)　ラフランスとさくらんぼとが共に入っているゼリー
ii)　ラフランスと「さくらんぼのゼリー」
iii)　(希に) ラフランスのゼリーとさくらんぼのゼリー

という解釈が容易にできるからである．ユーザの検索意図は上記のいずれかではないかと推測することは可能であるが，システムが勝手に決めつけて検索処理を始めることが良いとは思わない．ユーザが検索意図を容易かつ明示的に表明する方法がないことが問題である．上記 i)〜iii) の違いを part-of 関連に基づいて図示すると，図 1 のようになる．

問合せ意図内の part-of 関連を表現するための記号は

- 左辺と右辺に接続なし
- 左辺が右辺の下に接続
- 左辺と右辺が共通の上位ノードに接続

を表現できればよいので，それぞれ $RAND$, $DAND$, AND という記号で記述することにする．3 番目の「接続なし」は右辺と左辺に関係がないことを表現すると同時に，両方が同じ情報単位内に存在することを表現しようとしている．図 1 の問合せ意図を上記の記号で表現すると，

i)　“ラフランス” AND “さくらんぼ” $DAND$ “ゼリー”
ii)　“ラフランス” $RAND$ “さくらんぼ” $DAND$ “ゼリー”
iii)　“ラフランス” $DAND$ “ゼリー” $RAND$ “さくらんぼ” $DAND$ “ゼリー”

となる．

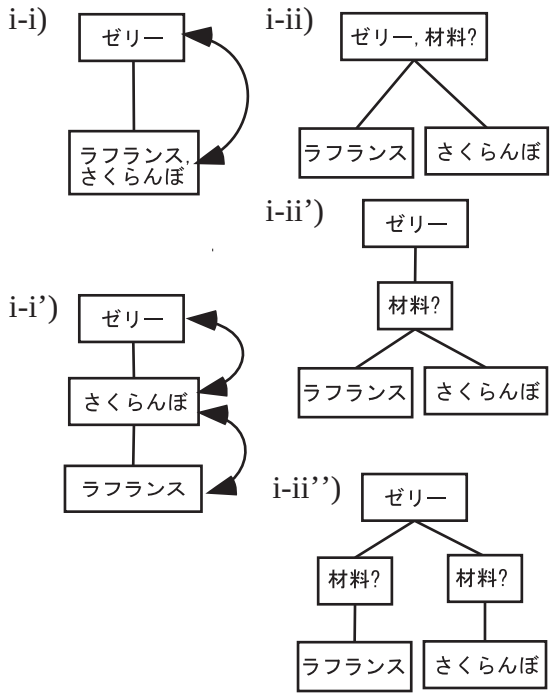

図 2　変換された構造木の適合パターン 1

3. 問合せ木

図 2 に図 1 の i) の意図に合う可能な構造部分木を示した．構造部分木の矩形はノードを，矩形内の文字列はノードを代表する語句或いは特徴語（以下代表語）を表している．それぞれの構造部分木は，

i-i)　ラフランスとさくらんぼを共に代表語として持つノードとゼリーを代表語として持つノードが接続
i-i')　それぞれの代表語を持つノードが接続
i-ii)　ゼリーを代表語として持つノードの下位に他の代表語を持つノードが接続
i-ii')　ゼリーを代表語として持つノードの下位にあるノードを挟んで他の代表語を持つノードが接続
i-ii'')　i-ii') の変形

であることを表している．また，両矢印は上下が入れ替わっても良いことを表している．これは，木構造の作り方によって上位・下位（包含・被包含）の表現が違うことを吸収するためである．i-i), i-i') は最下位ノードが，i-ii), i-ii'), i-ii'') は最上位ノードが「ラフランスとさくらんぼが共に入っているゼリー」の適合ノードであるといえる．i-i) 型は下位ほど事象を特定する表現であり，i-ii) 型は部品展開を表現しているからである．

ここで，接続しているノード同士が直接接続するという制約を課したとすると，それによって問合せ方式の柔軟性が損なわ

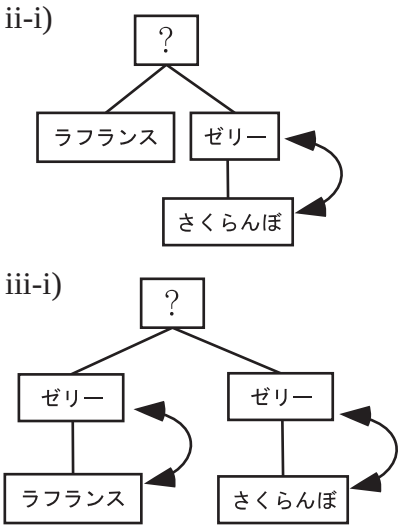

図 3　変換された構造木の適合パターン 2

れてしまう可能性がある．また，再現率の低下が推測できる．例えば，季節のゼリーにラフランスとさくらんぼの両方入ったゼリーがあるとすると，キーワードとして指定されない語句 ”季節” を代表語として持つノードが間に存在する可能性が容易に想像できるからである．そのため，本研究が提案する問合せ木は入力キーワードを代表語として持つノード同士の距離を 1 に限定しないという柔軟性を持たせることにした．i-ii) 型に材料?を代表語として持つノードを例示しているが，議論を簡単にするため単に想定が可能なパターンを漏らさずに示すことが本意である．外部からの知識として挿入することができれば適合率（或いは精度）の向上が期待できるが本稿では議論しない．

図 1 の ii), iii) の例の直観的な適合パターンを図 3 に示した．ii-i) はラフランスと「さくらんぼのゼリー」を共に含む情報単位を求めるので，最上位ノードが適合ノードである．同様に，iii-i) も「ラフランスのゼリー」と「さくらんぼのゼリー」を共に含む情報単位を求めるので，最上位ノードが適合ノードである．本研究では，図 2 の i-i) 型や「材料?」を除いた i-ii) 型を ii-i) 型，iii-i) 型の問合せ木も考慮して考察をすすめる．

3.1　演　　算

ここで，資料の構造を木構造にモデル化して議論をすすめることにする．あるノード v_i が存在するツリーのルートから v_i への経路上に存在するすべてのノードの集合を $path(v_i)$ とする．

親戚演算子 *RAND*

RAND は，左辺と右辺のキーワードをそれぞれ代表語としてもつノードが構造木中に存在し，共通の祖先を持つかどうかを調べて共通の祖先のうち最も経路が長いノードを解とする．

$$v_i\ RAND\ v_j = v_k$$
$$(\text{ただし，}\ path(v_k) = path(v_i) \cap path(v_j))$$

n 項親戚演算

n 個のノード $v_1, \cdots, v_n$ の RAND 演算は，親戚関係にある $v_1, \cdots, v_n$ の共通で最も近い祖先を解とする．

$$v_1\ RAND \cdots RAND\ v_n = v_k$$
$$(\text{ただし，}\ path(v_k) = path(v_1) \cap \cdots \cap path(v_n))$$

RAND 演算は各ノードへの経路上に存在するノード集合の積をとるので，入力されるノードの順に依存しない．

直系演算子 *DAND*

DAND 演算は，キーワード間の修飾関係「の」に対応し，直系関係にある v_i, v_j の下階層側を解とする．

$$v_i\ DAND\ v_j = \begin{cases} v_i, & \text{path}(v_i) \supset \text{path}(v_j) \\ v_j, & \text{path}(v_i) \subset \text{path}(v_j) \end{cases}$$

ここで，解を下階層側としたのは，構造の下階層に下るにしたがって内容が特化していくことを考慮したからで，細項目側となる下階層側を解とするのが無難と考えたからである．

n 項直系演算

n 個のノード $v_1, \cdots, v_n$ の DAND 演算は，直系関係にある $v_1, \cdots, v_n$ の最も下層のノードを解とする．

$$v_1\ DAND \cdots DAND\ v_n = v_k$$
$$(\text{ただし，}\ path(v_k) = path(v_1) \cup \cdots \cup path(v_n))$$

RAND 演算と同様に DAND 演算は各ノードへの経路上に存在するノード集合の和をとるので，入力されるノードの順に依存しない．

3.2　関　　数

RAND 演算を実行する rand() 関数と DAND 演算を実行する dand() 関数を以下のように定義する．

***rand*() 関数**

rand() 関数は，任意のノード v_i, v_j を引数とし，v_i, v_j が親戚関係にあるとき，$path(v_i) \cap path(v_j) = path(v_k)$ であるようなノード v_k を戻り値とする．v_i, v_j が親戚関係にないときと，空値であるとき ϕ を戻り値とする．

n 項の RAND 演算

$v_1\ RAND\ v_2\ RAND \cdots RAND\ v_n$

を *rand*() 関数で表現すと，

$$rand(\cdots(rand(v_1, v_2)\ldots), v_n)$$

である．これは，RAND 演算を左から順に実行することを意味し，戻り値と次の引数とを引数として *rand*() 関数の適用を $n-2$ 回行う．

***dand*() 関数**

dand() 関数は，任意のノード v_i, v_j を引数とし，v_i, v_j が直系関係にあるとき，$path(v_i) \cup path(v_j) = path(v_k)$ であるようなノード v_k を戻り値とする．v_i, v_j が直系関係にないとき

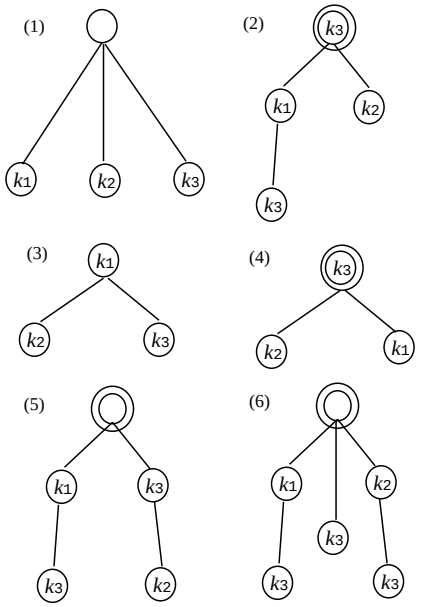

図 4　検索式 [(k_1 と k_2) の k_3] の分配則的解釈

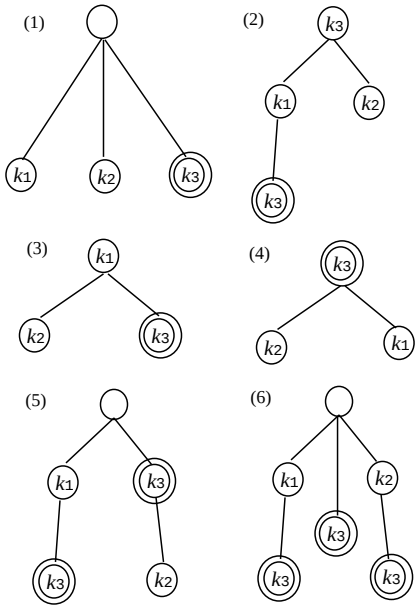

図 5　検索式 [(k_1 と k_2) の k_3] の算数的解釈

と，空値であるとき ϕ を戻り値とする．

n 項の DAND 演算

$v_1\ DAND\ v_2\ DAND\ v_3\ DAND \cdots DAND\ v_n$

を $rand()$ 関数で表現すと，

$dand(\cdots(dand(v_1, v_2), v_3), \ldots), v_n)$

である．これは，DAND 演算を左から順に実行することを意味し，戻り値と次の引数とを引数とした $rand()$ 関数の適用を繰り返して行う．

4. 複合検索

検索キーワード間の関係を複数指定したときの解釈を考察する．

$DAND$ 優先

本稿はここまで root ノードからキーワードを含むノードへの経路に着目して演算を定義して利用してきたが，検索結果として得られる部分資料が含むノードに着目して RAND と DAND の優先を考える．

$v_i\ RAND\ v_j$ は k_i をキーワードとして持つノードを root とする部分資料と，k_j をキーワードとして持つノードを root とする部分資料とを含む部分資料を求める．それぞれの部分資料が含むノードの集合を $V_{k_i}, V_{k_j}, V_{k_i\ RAND\ k_j}$ とすると，

$$V_{k_i} \cup V_{k_j} \subseteqq V_{v_i\ RAND\ v_j}$$

となっていなければならないはずである．同様に，$v_i\ DAND\ v_j$ は k_i をキーワードとして持つノードを root とする部分資料と，k_j をキーワードとして持つノードを root とする部分資料で包含される方の部分資料を求める．それぞれの部分資料が含むノードの集合を $V_{k_i}, V_{k_j}, V_{v_i\ DAND\ v_j}$ とすると，

$$V_{k_i} \cap V_{k_j} = V_{v_i\ DAND\ v_j}$$

となっていなければならないはずである．つまり， RAND に和のような性質があり，DAND に積のような性質があることがわかる．複合演算の便宜上 DAND 優先を採用することにした．

たとえば，$v_1\ RAND\ v_2\ DAND\ v_3$ は

$rand(v_1, dand(v_2, v_3))$

というように，DAND の処理を行った後に RAND の処理を行うことにする．

ところで，ユーザが k_1 と k_2 の k_3 と表現するとき,「k_1」と「k_2 の k_3」を探している場合と 「k_1 と k_2」の「k_3」を探している場合の二種類が考えられるが，本稿では検索式の自然言語的解釈は行わず，前者を

k_1 と k_2 の k_3

後者を

(k_1 と k_2) の k_3

と書いてもらうことにする．

括弧と展開

ここで，

- 検索式の解釈の関数表現を $Q()$
- ノード v のキーワード集合を $key(v)$

とする．$Q()$ の戻り値はノードの集合である．また，見やすさを考慮して，検索式を [ ] でくくることとする．

検索式 [k] の解釈は，

$$Q(k) = \{v \mid k \in key(v)\}$$

である．検索式 [k_1 の k_3 と k_2 の k_3] の解釈は

$$\begin{aligned} &Q(k_1 \text{ の } k_3 \text{ と } k_2 \text{ の } k_3) \\ &\quad = \{rand(dand(v_1, v_3), dand(v_2, v_3)) \\ &\qquad \mid v_1 \in Q(k_1), v_2 \in Q(k_2), v_3 \in Q(k_3)\} \end{aligned}$$

である．しかしながら，この検索式には k_3 が二度指定されている．そこで，括弧を用いて入力を簡略化する．上記はユーザが「k_1 の k_3」と「k_2 の k_3」を探していると考えられるので検索式に括弧を許し，検索式 [(k_1 と k_2) の k_3] を検索式 [k_1 の k_3 と k_2 の k_3] と展開することにする．これは，上述の RAND が持つ和の特徴と DAND が持つ積の特徴を利用して分配則を

適用している．

ここで，検索式 $[(k_1$ と $k_2)$ の $k_3]$ を算数的に

$$Q((k_1 \text{ と } k_2) \text{ の } k_3)$$
$$= \{dand(rand(v_1, v_2), v_3))$$
$$\mid v_1 \in Q(k_1), v_2 \in Q(k_2), v_3 \in Q(k_3)\}$$

と解釈したとしよう．図 4 に検索式 $[(k_1$ と $k_2)$ の $k_3]$ を分配則に基づいて解釈したときの結果を，図 5 に検索式 $[(k_1$ と $k_2)$ の $k_3]$ を算数的に解釈したときの結果をそれぞれ示した．図中の k_1, k_2, k_3 はキーワード k_1, k_2, k_3 をそれぞれ含むノードを示し，二重丸のノードが各解釈での結果である．図 4 と図 5 が全ての場合を網羅しているわけではないが，算数的に解釈した場合，ユーザが意図しない結果を含み，期待した結果を含まないことが考えられる．それは，図 5 で (4) 以外は k_1 あるいは k_2 をキーワードとしてもつノードを含まない部分木の root を結果の材料としているからである．検索式はユーザとシステムとのインタフェースであるので，このような結果は直観的にも正しいと思えない．そのため，現時点では分配則に基づく解釈を採用した．

5. ま と め

従来の資料検索は検索結果の単位が資料であったが，ユーザが必要とするのは資料中のある部分のみであると考えられるため，資料の任意の部分にメタデータを与え，ユーザが必要とするであろう部分のみを検索結果として提供するための検索方法を考察した．特に，複数のキーワードに対してその間の関連を簡単に指定したときに有効な検索結果を求める関係と演算とを整理し，検索式の解釈方法について議論した．キーワード間の関係として並列関係を表す「と」と修飾関係を表す「の」を検索式に指定するため，従来の検索機構が概ね AND, OR の論理演算を実装するのに対して AND を細分化して RAND 演算と DAND 演算を導入した．それは，二つの検索キーワード k_1 と k_2 が「k_1 と k_2」という意図で入力されているのか，それとも「k_1 の k_2」という意図で入力されるのかを区別したいのとともに，ユーザがキーワード間の関係を明示的に指定できれば有用と考えたからである．

また，検索式の解釈方法において集合論的な分配法則を利用できたのは，構造木の部分木をその root ノードからの経路上に存在するノードの集合として議論を進めたことによる．

文　献

[1] Taira Yoda, Hidenari Kiyomitsu, Kazuhiro Ohtsuki, Jun-ya Morishita, "An Extended AND Operations for Retrieving a Flexible Information Unit from Tree Structured Data," Proc. of International Symposium on Applications and the Internet Workshops (SAINTW'07), pp. 53-56, 2007.

[2] 坂口良, 清光英成, 大月一弘, 森下淳也, 依田平, "緩い構造を持つデータに対する問合せ語間の関連を反映する検索方式の提案," 第 18 回データ工学ワークショップ DEWS2007, C1-7, 2007

[3] 依田平，大月一弘，森下淳也，清光英成，" デジタルアーカイブに対する効率的な検索の提案　—神戸大学電子図書館システムを例として— "，情報処理学会シンポジウムシリーズ 18 号　人文科学とコンピュータシンポジウム論文集，pp.259-266，2001.

[4] 依田平，小椋正道，大月一弘，森下淳也，清光英成，" 電子図書館用デジタルアーカイブの検索方法の検討 "，情報処理学会研究報告 70 号，pp.469-476，2001.

[5] 依田平，大月一弘，清光英成，森下淳也，" ツリー型不定形文書からの部分文書の検索手法の検討 "，第 14 回データ工学ワークショップ DEWS2003，2003.

[6] 依田平，渡邊隆弘，大月一弘，鳩野逸生，岩杉大輔，" 多様な資料構造に対応したデジタルアーカイブシステムー神戸大学電子図書館アーカイブ検索システムー "，情報処理学会研究報告，2003-FI-73，pp. 45-52，2003.

[7] K. Tajima, K. Hatano, T. Matsukura, R. Sana and K. Tanaka: Discovery and Retrieval of Logical Information Units in Web, Proc. of WOWS, pp. 13-23, Berkeley, CA, 1999.

[8] 絹谷弘子, 波多野賢治, 吉川正俊, 植村俊亮: 情報検索技術を用いた部分文書構造の自動抽出, 情報処理学会論文誌: データベース, Vol. 42, No. SIG8(TOD10), pp. 36-46, 2001.