データベースシステム 119−17
（1999. 7. 21）

World Wide Webにおける文脈を利用した同一実体の発見手法

森嶋厚行† 北川博之† 高野智††
†筑波大学 電子・情報工学系 ††筑波大学 第三学群 情報学類

World Wide Webの普及により，分散情報源中の情報が，ハイパーテキストリンクによって相互に関連づけられるようなった．そのような関連付けの重要なクラスとして，現実世界の同一の実体(Entity)を表す情報間の関連付けがある．しかし，通常，Webにおいては，同一の実体を表す情報のすべてが明示的に関連づけられている訳ではない．したがって，同一の実体を表す情報は，Webサーチエンジンやページ内検索などを組み合わせて調べる必要がある．本稿では，Webにおいて，同一の実体を表す情報の出現を検索する手法を提案する．本手法の特徴は次の通りである．(1) 検索の単位が，同一実体を表す情報の出現である (2) 検索の基準が「同一の実体を表す可能性が高いか否か」である．(3) 判断の手がかりとして，実体を表す情報が出現する文脈を利用する．本稿ではさらに，簡単な実験を用いて，本手法の有効性の検証を行った．その結果，いくつかの場合については，高精度の同一実体発見が可能であることが分かった．

Context-based Entity Identification in the World Wide Web

Atsuyuki Morishima†, Hiroyuki Kitagawa†, and Satoshi Takano††
†Institute of Information Sciences and Electronics, University of Tsukuba
††College of Information Sciences, Third Cluster of Colleges, University of Tsukuba

In the World Wide Web, hypertext links are used to relate various data objects to each other. Those hypertext links are important which relate data occurrences representing the same entity in the real world. However, such occurrences are not nesessarily related by the links. Therefore, finding data occurrences for the same entity requires users to use Web search engines, to retrieve particular strings in pages, and so on. This paper proposes a retrieval method for identifying data occurrences representing the same entity in the real world. The features are as follows. (1) A retrieval unit is *a data occurrence* representing the same entity. (2) The criteria for retrieval is *whether the data occurrence is likely to represent the entity or not.* (3) *Contexts of data occurrences* are used as clues for retrieval. This paper also shows the result of a preliminary experiment. It shows that the proposed method can give high presicion in some cases.

1 はじめに

World Wide Web(以下 Web) の普及により，分散情報源中の情報が，ハイパーテキストリンクによって相互に関連づけられるようなった．関連付けされているデータ間の関係には，様々なものがある．例えば，チュートリアルサイトなどでは，閲覧の順序関係を表す関連付け (Preveous, Next など) がある．また，ニュースサイトなどでは，より詳細な情報や関連情報への関連付けがある．そのような様々な関連付けの中でも，重要な関連付けのクラスとして，現実世界における同一の実体 (Entity) に関するデータ間の関連付けがある．例えば，研究会のプログラムサイトでは，論文の著者名から，その人のホームページへの関連付けがある．これは，論文の著者と，そのホームページの人物が，現実世界における同一の実体であることから，関連づけられたものである．通常，Webにおいては，同一の実体に関する情報のすべてが，明示的に関連づけられている訳ではない．むしろ，明示的には関連づけられていないもの方が多い．したがって，現在は，ある実体に関する情報を求めたいにもかかわらず，明示的な関連付け (リンク) が存在しない場合には，Webサーチエンジンを使うのが一般的である．しかし，Webサーチエンジンは，同一の実体を表す情報を発見する目的で設計されているわけではない．その目的は，欲しいWebページの内容を表すキーワード集合を与えて，それに近い内容を表すと考えられるページを発見することである．その結果，通常のサーチエンジンを利用して，ある実体を表す情報を検索したい場合には，次のような問題が生じる．

- 検索結果の単位がWebページに固定されている．一般には，ある実体に関する情報の単位は，ページと一致するとは限らない．したがって，検索結果から，さらにページ内検索などを行って，欲しい情報を得る必要がある．
- 検索の基準が,「同一の実体である可能性が高いかどうか」ではない．現実には，ある実体を表す単語や単語列 (例えば著者名) が多義語であることもある．しかし，実体を表す単語や単語列を入力して検索すると，上位にランクされるページは，単に単語の出現頻度が高いページであったり，単語がタイトルタグに出現するようなページである．そのようなペー

ジでありさえすれば，実際には違う実体に関する情報を含むページであっても上位にランクされる．逆に，同一の実体の情報を含んでいるページであっても，ページの片隅にひっそりとあるようでは，下位にランクされる．

本稿では，Web において，同一の実体を表す情報の出現を検索する手法を提案する．本手法の特徴は，以下の通りである．

- 検索の単位が Web ページではない．**Web ページに含まれている，実体を表す情報の出現 (Occurrence) が検索される．**同一の Web ページに複数含まれている場合は，それらがそれぞれ別の検索結果として返される．
- **検索の基準が，「同一の実体を表す可能性が高いか否か」である．**同一の実体を表す情報の出現である可能性が高いと判断されれば，どんなにページの片隅に存在していても，その出現は上位にランクされる．
- 「同一の実体を表す可能性が高いか否か」を判断する手がかりとして，**実体を表す情報が出現する文脈 (Context) を利用する．**これにより，実体を表す情報が多義語であっても，高精度の検索を実現する．

2 同一実体発見手法の応用

本節では，World Wide Web における同一実体発見手法 (Entity Identification Method) の応用例について述べる．次のような応用が考えられる．

(A) **ブラウジングフェーズと融合された実体検索システム**: 通常の Web ブラウザでは，ブラウジング時に発見したある実体に関連する情報を求めたい場合には，次のような操作を行う必要がある．(1)Web サーチエンジンのサイトにいく (2) その実体に関連するキーワード群を投入し，ページ群を求める (3) 検索結果のページ内をさらに検索する．それに対して，本システムでは，ブラウジング時にある実体を表す情報の上でマウスボタンをクリックすると，その実体を表す情報の出現を検索し，その出現へのリンク群が返される (図 1)．

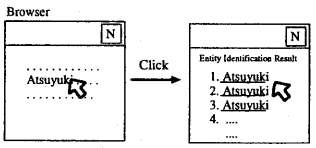

図 1. ブラウジングと融合された実体検索システム

(B) **リンク自動生成による Web サイト統合支援システム**: 複数の独立して開発された Web サイトが存在するとする．これらは，互いに関連した情報を持つものの，それぞれ独立に開発されたため，明示的なリンクが張られていない．同一実体発見手法を用いることにより，リンクを自動生成し，これらのサイトの統合を支援する (図 2)．

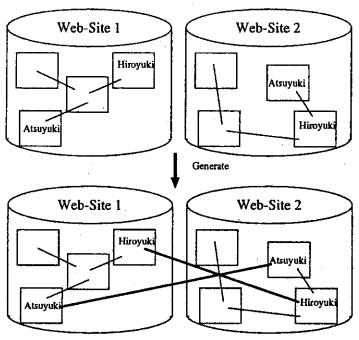

図 2. Web サイト統合支援システム

(C) **データ編集支援システム**: エディタなどによって Web ページなどの文書を作成，編集するときに，その入力に連動する Information Window を用意する．そこには，同一実体発見手法を用いて，現在入力した情報が表す実体の関連情報を，自動的に表示する．利用者は，この情報を参考にしたり，カットアンドペーストすることによって，文書作成の労力を軽減する (図 3)．

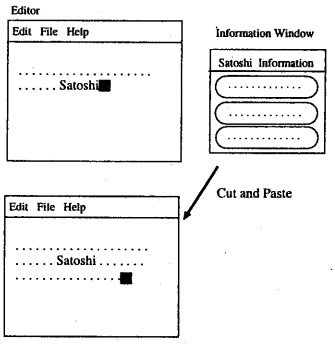

図 3. データ編集支援システム

3 文脈を利用した同一実体発見

本節では，Web における同一実体発見の一手法を提案する．本手法の特徴は，実体を表現する情報が現れる文脈 (Context) を利用することである．

3.1 Web における同一実体発見問題

まず，ある実体を表す情報の *Occurrence* を，次のようにモデル化する．ある Occurrence occ は，3 つ組 (*Representaton*, *ContainingPage*, *PositionInPage*) で表現される．ここで，*Representaion* は，ある実体を表す連続単語列 $w_1 w_2 \ldots w_p$('Univ. of Tsukuba' など)，*ContainingPage* は，その出現を含むページの URL，*PositionInPage* は，その出現の先頭単語 w_1 の，ページ内での位置を，それぞれ表す．一般には，一つの Web ページ内に同一実体を表す複数の Occurrence が存在し

うる．このとき，Webにおける同一実体発見問題を，次のように定義する．

定義 Webにおける同一実体発見問題とは，あるOccurrence occ_0 が与えられたとき，occ_0 が表現している現実世界の実体と同じ実体を表現している全てのOccurrence $occ_1, \ldots, occ_n$ を，Web中から発見することである．

3.2 アルゴリズム

一般には，Webにおける同一実体発見問題を完全に解くことは不可能である．本節の手法では，Occurrence occ_o に対して，同一実体を表すと推測されるOccurrence群 $occ_1, \ldots, occ_m$ と，「同一であると推測される度合い」$r_1, \ldots, r_m$ を計算する．ここで $0 \leq r_i \leq 1$ である．r_i は，1に近いほど，同一実体を表す可能性が高いことを表す．

本アルゴリズムでは，あるOccurrence occ の *Context* を，occ のまわりの特定の領域（ここでは *Extent* と呼ぶ）に存在する *Concept* の集合であるとモデル化する．ここで，Conceptとは，現実世界のある概念を表すテキスト表現である．例えば，ある大学の研究室のホームページに，'DBLAB' というRepresentationを持つOccurrenceがあるとする．このOccurrenceのContextが，例えば，{ 'Institute', 'Information', 'Univ. of Tsukuba', 'Member', 'Kitagawa', 'Ishikawa', 'Morishima' } であるということがあり得る（図4）．図に示すように，Extentは複数のページにまたがっていても良い．

本アルゴリズムでは，内部的にContextを利用して，同一実体であると推測される度合いを計算する．アルゴリズムを図5, 6, 7に示す[1]．図5は，本アルゴリズムの最も上位の関数C-EI(Context-based Entity Identification)である．関数C-EIは，あるOccurrence occ_0 を与えられたとき，同一の実体を表す可能性があるOccurrence occ_i と，それが occ_0 と同一の実体を表すOccurrenceであると推定される度合いの組の集合，すなわち $\{(occ_1, r_1), \ldots, (occ_m, r_m)\}$ を返す．

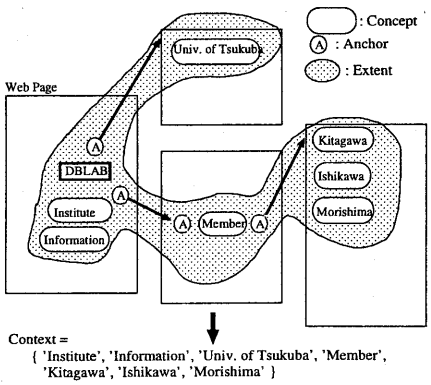

図4. Context

[1]これらに含まれる型宣言，変数宣言のうち，自明なものは省略している．

関数C-EIの概要は次の通りである．まず，occ_0 のRepresentationを含むWebページ群を求める．次に，それらページ群に含まれている全てのRepresentaionのOccurrenceの集合を求める（$\{occ_1 \ldots occ_m\}$ とする）．続いて，occ_0 を含む全ての $occ_i (0 \leq i \leq m)$ について，Context c_i と，それに基づくベクトル $\boldsymbol{v_i}$ を求める．さらに，$\boldsymbol{v_0}$ と $\boldsymbol{v_i}$ のコサイン測度を，occ_0 と occ_i が同一の実体を表すと推測される度合いである r_i とする．このように，Contextを用いた同一実体発見を行う．

```
function C-EI(occ_0): set of (Occurrence, Real)
type
    Context: set of Concepts;
var
    U: set of URLs;
    OCC: set of Occurrences;
    c_i: Context;
    AC: set of Concepts;
begin
    w_1 w_2 ... w_p := Representation(occ_0);
    U := retreivePages(w_1 w_2 ... w_p);
    OCC := ⋃_{u∈U} getOccurrences(u, w_1 w_2 ... w_p);
    let OCC = {occ_1, occ_2, ..., occ_m};

    let c_i = getContext(occ_i, ContainingPage(occ_i));
    let v_i = getVector(c_i, AC) where AC = ⋃_{i=0}^{m} c_i;
    let r_i = (v_0 · v_i) / (|v_0||v_i|);
    return {(occ_1, r_1), ..., (occ_m, r_m)};
end;
```

図5. 関数 C-EI

図6に関数getContextを示す．これは，引数として与えられるOccurrence occ のContextを求める．すなわち，occ のExtentに含まれるConceptを探し，その集合を結果として返す．getContextの引数としては，occ の他に，Conceptを探索する起点ページ *page* をとる．処理の概要は次の通りである．変数 c は，結果を格納するための変数である．まず，Webページ *page* の中で，occ のExtentに含まれるテキスト部分を，t に格納する．関数extentについては後述する．続いて，関数getConcepts(これも後述)を用いて，変数 c に，t に含まれるConcept集合を格納する．もし，t 内にハイパーテキストリンク（アンカー）がある場合には，そのリンクが指すページを起点としてgetContextを行い，その結果のConcept集合を c に追加する．

```
Funcion getContext(occ, page): Context
type
    Context: set of Concepts;
var
    c: Context;
    t: Text;
begin
    t := getText(extent(occ), page);
    c := getConcepts(occ, t);
    for all anchor a_i containedIn t do
        c := c ∪ getContext(occ, referencedPage(a_i));
    return c;
end;
```

図6. 関数 getContext

図7に関数getVectorを示す．これは，Context c のベクトル表現を計算する．引数として，c の他に，ベクトル空間を表すための，Conceptの集合 AC をとる．本アルゴリズムでは，AC として，検索対象となるいずれかのContext c_i に含まれるような，全てのConceptの集合が与えられる．結果は，AC 中のそれぞれのConceptに対応する次元を持つ，$|AC|$ 次元のベクトルである．各次元は，Conceptが存在する場合には，関数weightで重み付けされた値をとる．存在しない場合は，0をとる．

```
function getVector(c, AC): Vector
begin
    let AC = {concept_1, ..., concept_n};
    return (s_1, ..., s_n) where
        if concept_i ∈ c then s_i = weight(concept_i)
            else s_i = 0;
end;
```

図7. 関数 getVector

3.3 関数 extent, getConcepts, weight

3.2節のアルゴリズムでは，関数extent, getConcepts, weightが未定義である．最も簡単なものとしては，例えば，次のような定義が考えられるが，定義によって同一実体発見の精度が左右されると考えられるため，適切な定義を行うことが重要である．

extent Occurrenceから，k 語以内の語に含まれる範囲をExtentとする．リンクをたどる場合は，l 語分進んだと解釈する[2]．例えば，$k = 20, l = 5$ など．

getConcepts Extentに現れるストップワード以外の単語を全てConceptとして解釈する．また，"..."で囲まれた文字列も，Conceptと解釈する．

weight idfを利用する．現実には，Web全体に対するidfは計算不可能であるので，例えば，Webサーチエンジンを用いて以下のような疑似idfを求める．

$$weight(concept_i) = log(\frac{MAX\_HitCount}{HitCount(concept_i)})$$

ここで，$HitCount(concept_i)$ は，$concept_i$ を表す文字列をWebサーチエンジンに投入したときに返される，ヒットページ数を表す．$MAX\_HitCount$ は，計算対象となるContextに現れる全てのConceptのヒットページ数のうち，最大の数を表す．

4 実験

本稿の手法を用いて簡単な実験を行った．本実験では，関数extent, getConcepts, weightは，3.3節で説明した定義を用いた．extentのパラメータとしては，$k = 20, l = 5$ を用いた．また，関数retrievePagesとweightの実装に，gooを利用した．

図8に示すような，3つの実験を行った．実験1, 2は，特定の研究者のOccurrenceを求める．これらについては，比較的本手法が効果的であると想定される．その理由は，研究者の名前は，論文の共著者の名前や，研究テーマに関連するキーワードと共に出現することが多いからである．実験3は，実験1, 2に比べ，出現のContextがバラエティに富んでいると考えられるため，精度が落ちることが想定される．

図9は，各実験によって得られた，有効Occurrence数と正解数を示す．有効Occurrence数とは，retrievePages (goo) によって検索されたページ群に含まれるOccurrenceのうち，英文のページに含まれるOccurrenceのみの総数である．今回は，これらだけを対象として同一実体発見を行った．正解数は，有効Occurrence群のうち，occ_0 と同じ現実世界の実体を表しているOccurrenceの数である．これは，実際に人手でチェックを行った．

	実体	Representaion	occ_0 の位置
実験1	森嶋厚行	Atsuyuki	ゼミメンバー紹介ページ
実験2	北川博之	Kitagawa	ゼミメンバー紹介ページ
実験3	太陽	Sun	太陽系の説明ページ

図8. 実験内容

	有効 Occurrence 数	正解数
実験1	93	10
実験2	527	95
実験3	1872	133

図9. 有効 Occurrence 数と正解数

図10, 11, 12は，本手法の精度を検証するために，各実験について，有効Occurrenceに含まれる正解集合に対するrecallと，実験結果のprecisionの関係を示したものである．予想通り，実験1と2の精度が相対的に高いことが示された．

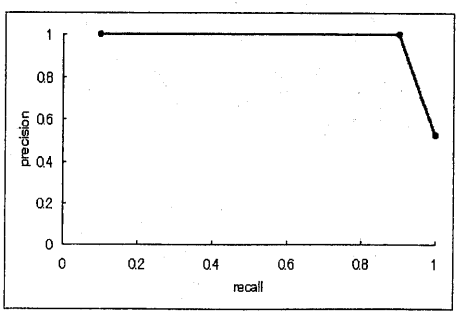

図10. 実験1

[2]もし，ハイパーテキストリンクが，ページ中の特定のアンカーを指す場合は，その位置から語数をカウントする．そうでない場合には，ページの先頭位置からカウントする．

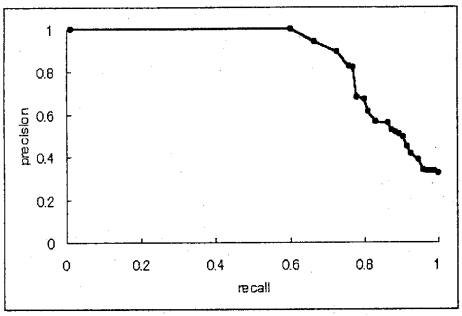

図 11. 実験 2

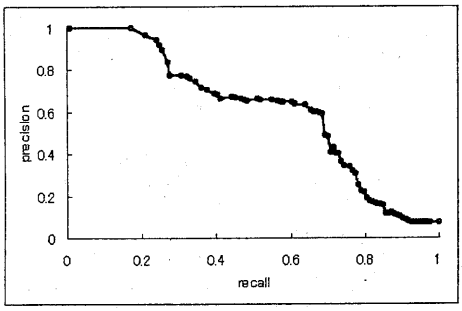

図 12. 実験 3

5 議論

実験から，本手法が，比較的単純な extent と getConcept の定義を用いても，少なくともある種の実体に関しては，高精度の同一実体発見を実現することが分かった．しかし，3.3 節で述べたとおり，一般的には，本手法の精度は，extent, getConcepts, weight の定義に大きく左右されると考えられる．次に，それぞれに関する議論を示す．

(A) **関数 extent に関する議論**: (1)Extent の範囲の決定について: 本稿の実験では，3.3 節の関数 extent の定義を使い，さらにパラメータとして $k = 20, l = 5$ という比較的狭い範囲を Extent とした．人間が Context を利用して同一の実体を発見する際には，一般的には，より広い範囲が Extent として利用されると考えられる．しかし，単純に Extent を拡大すると，実体と関連度の低い Concept が Context に含まれる可能性が高くなる．したがって，これらのトレードオフを検証する必要がある．また，より効果的な Extent の決定をおこなうためには，リンクのセマンティクスなどを利用することも考えられる．(2)Extent が広がりうる場所について: 本アルゴリズムでは，Extent が広がる場所は，与えられた Occurrence が現れるページと，そのページからハイパーテキストリンクをたどって到達可能なページ群の中に限定されている．もし，逆に，Occurrence が現れるページに対してハイパーテキストリンクを張っているページにも，Extent が広がりうるとすれば，より高精度の検索が可能になると考えられる．現在は，特定の Web サーチエンジンでは，あるページを指すページ群を求めることができるので，これを使えば実現可能である．

(B) **関数 getConcepts に関する議論**: Stemming を行うことや，辞書を使って複合語を Concept とすれば，より高精度の検索が実現可能と考えられる．

(C) **関数 weight に関する議論**: 今回の実験では，Web 全体に対する疑似 idf を用いた．これは，重要度を表す尺度として適切であると考えたからである．本手法では，retreivePages によって検索された Web ページ群が，Occurrence を求めるための対象となる．しかし，通常の文献検索とは異なり，これらの中での idf を利用することは，不適切であると考えられる．なぜなら，この Web ページ群には偏りがあるからである．例えば，このページ群に含まれるある実体の出現の割合が多い場合，特徴的な単語であるにも関わらず，それらの全てに含まれているために，重要度が下がる．逆に，たまたま含まれていた数少ない単語の重要度が高いと判断される．通常の文献検索と異なるもう一つの点は，文献検索では一般的に利用される単語の頻度情報が，同一実体発見においては，それほど重要ではないと考えられることである．

さらに，実験の結果から，次のことが観察された．すなわち，正解にも関わらず低くランキングされる Occurrence のいくつかについては，その Context が，occ_0 の Context に対しては重なりは少ないものの，上位に存在する正解の Occurrence の Context との重なりがある程度存在するということである．したがって，精度を上げるために，Relevance Feedback や Query Expansion などの手法が有効であると考えられる．

6 関連研究

Web における検索については，実用レベルから研究レベルまで，様々なものが存在するが，検索単位の観点からは，次のようなものに分類される．(1) ページ検索:Yahoo や goo などの通常の Web サーチエンジンなどがこれに当てはまる．(2) 関連するページ群の検索:[4][11][15] などの研究がある．(3) 画像などの特定の型のオブジェクトの検索:Lycos がサポートしている．研究レベルでは [7] がある．また，以上のような Web 検索に限定せず，文書検索の研究一般に範囲を拡大すれば，文書よりも小さな単位を検索する Passage Retrieval[13] の研究がある．これらに対し，本研究の検索単位は，ある実体を表すデータ表現の出現である．

検索に文脈情報を用いることについては，次のような関連研究が存在する．[6] では，Database をグラフ構造で表現し，ノード検索を行う行う手法を提案している．そこでは，ある基準で選択されたノード群を，近傍に現れて欲しいノード群との物理的位置に基づいて，ランキングして出力する．[7] では，Web 中の画像検索のために，画像のまわりのテキスト情報を用いたインデックス

付けを行うことを提案している．[9] では，検索対象となる画像に，内容に関する特徴語でインデックス付けを行い，連想検索を行う仕組みを提案している．ここでの焦点は，連想検索における類似度の解釈が，利用者に指定された文脈 (単語群) によって，動的に決定されることである．本研究がこれらと大きく異なる点は，利用者が文脈を意識しないことである．文脈は自動的に抽出され，暗黙に利用される．文脈を暗黙に利用するものとしては，[5] がある．これは，ハイパーテキストデータベースに格納されている記事を検索するシステムにおける Relevance Feedback に関する研究である．記事検索の結果に対してユーザがブラウジングを行う際に，たどったリンクをシステムが監視し，そのリンクの周辺情報を利用して，Relevance Feedback を行う．また，[2] は，ハイパーテキスト中の文書を分類する際に，リンクによって関連付けされている他文書の情報を利用する．

Entity Identification は，情報源統合の分野における重要な概念である．以前から Multidatabase の分野で様々な研究が行われてきた．Sheth ら [12][14] は，Multidatabase におけるオブジェクト間の Semantic Similarity を定式化し，Semantic Similarity を判定する上での，Context の重要性を主張している．しかし，我々の知る限り，データが出現する Context を暗黙の手がかりとして，自動的に Entity Identification を行う手法の提案や実験の報告は存在しない．[3] では，各 Entity を表すテキスト間の類似性を尺度とした Entity の判定方式を提案している．[1][8] では，グローバルビューレベルのオブジェクトへのマッピング規則を，人手で指定することによって，Entity Identification の問題を解決する．[10] では，Extended Key Equivalence を用いた手法を提案している．この手法でも，Instance Level Functional Dependency という知識を人手で作成する必要がある．

7 結論と今後の課題

本稿では，Web において，同一実体を表す情報の出現を検索する手法を提案した．本手法の特徴は，次の通りである．(1) 検索の単位が，同一実体を表す情報の出現である (2) 検索の基準が，「同一の実体を表す可能性が高いか否か」である．(3) 判断の手がかりとして，実体を表す情報が出現する Context を利用する．

本稿ではまた，簡単な実験の結果を示し，本手法の有効性の検証を行った．その結果，いくつかの場合については，高精度の同一実体発見が可能であることを示した．

今後の課題，研究の展開としては，次のようなものがある．(1) より詳細な実験評価．多様な実体を検索対象としたり，様々な Context 作成法を用いて行う．(2) 本手法の改良．Extent が広がりうる場所の拡張などが考えられる．(3)Relevance Feedback や Query Expansion との組合せの検証．(4) 異種情報源統合への応用．Web 以外の情報源に対象を拡大する．(5)Authority 情報の優先など，「同一実体を表す可能性」以外の基準の導入．(6) 多量に得られる結果の提示法の開発 (クラスタリング，要約，など)．(7) 応用システムの構築と評価．(8) 同じ実体を表すか否かの問合せを受け付ける，Entity Identification Server の開発．

参考文献

[1] I. Chan and D. Rotem. Integrating Information from Multiple Independently Developed Data Sources. *Proc. CIKM'98*, pp. 242-250, 1998.

[2] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. *Proc. SIGMOD'98*, pp. 307-318, 1998.

[3] W. W. Cohen. Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity, *Proc. SIGMOD'98*, pp. 201-212, 1998.

[4] 段一為, 佐野綾一, 波多野賢治, 田中克己. 極小部分マッチグラフを基本単位とした Web 文書群の検索機構. *Proc. DEWS'99*, 1999.

[5] G. Golovchinsky. What the Query Told the Link: the Integration of Hypertext and Information Retrieval. *Proc. ACM Hypertext'97*, pp. 67-74, 1997.

[6] R. Goldman and N. Shivakumar. Proximity Search in Databases. *Proc. VLDB'98*, pp. 26-37, 1998.

[7] V. Harmandas, M. Sanderson, and M. D. Dunlop. Image Retrieval by Hypertext links. *Proc. SIGIR'97*, pp. 296-303, 1997.

[8] W. Kent, R. Ahmed, J. Albert, M. Ketabchi, and M. Shan. Object Identification in Multidatabase Systems. *Interoperable Database Systems (IFIP Trans. A-25, Proc. DS-5)*, pp. 313-330, 1993.

[9] Y. Kiyoki, T. Kitagawa, and T. Hayama. A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning. *ACM SIGMOD Record*, Vol. 23, No. 4, pp. 34-41, 1994.

[10] E. Lim, J. Srivastava, S. Prabhakar, and J. Richardson. Entity Identification in Database Integration. *Proc. ICDE'93*, 294-301.

[11] W. Li and Y. Wu. Query Relaxation by Structure for Web Document Retrieval with Progressive Processing. *Proc. Advanced Database Symposium'98*, pp. 19-25, 1998.

[12] A. Ouksel and A. Sheth. Semantic Interoperability in Global Information Systems. *SIGMOD RECORD*, vol. 28, no. 1, pp. 5-12, 1999.

[13] G. Salton, J. Allan, and C. Buckley. Approaches to Passage Retrieval in Full Text Information Systems. *Proc. SIGIR'93*, pp. 49-58, 1993.

[14] A. Sheth and V. Kashyap. So far (Schematically), yet So Near (Semantically). *Interoperable Database Systems (IFIP Trans. A-25, Proc. DS-5)*, 1993.

[15] K. Tajima, Y. Mizuuchi, M. Kitagawa, and K. Tanaka. Cut as a Querying Unit for WWW, Netnews, and E-mail. *Proc. ACM HyperText'98*, pp. 235-244, 1998.