Vol.2009-DBS-149 No.4
2009/11/20

挟みこむ検索: 2オブジェクトからの補間オブジェクト発見

旭　直人[†1]　山本岳洋[†1,†2]
中村聡史[†1]　田中克己[†1]

本論文では，ユーザが入力した2つのオブジェクトの間にあたるオブジェクトを発見する手法を提案する．例えば，桶狭間の戦いと本能寺の変の間に起こった出来事を知りたい，2つの知っている本の中間の難易度を持つ本を発見したい，といったような状況は良くある．しかし，従来の検索エンジンでは，こうしたオブジェクトを発見することは難しい．そこで本研究では，2つの入力の間に位置するようなオブジェクト（補間オブジェクト）を発見するシステムについて述べる．また，検索エンジンを利用し，語の出現位置に注目することで補間オブジェクトを自動的に発見する手法を提案する．最後に，評価実験により提案手法の有用性を示す．

Finding Intermediate Objects between Two Objects

Naoto Asahi,[†1] Takehiro Yamamoto,[†1]
Satoshi Nakamura[†1] and Katsumi Tanaka[†1]

We propose a method for finding intermediate objects between two objects that a user inputs. For example, there are many situations such that he/she wants to know an event between “the Battle of Okehazama” and “Honnoji Incident”, or that he/she wants to find a book that has intermediate level between two books he/she knows. However, it is difficult to find such intermediate objects by conventional search engines. First, we describe a system that find intermediate objects between two inputs. Second, we propose find intemediate objects automatically using positions of words. Finally, we show the results of our experiments and evaluate the effectiveness of our method.

†1 京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Graduate School of Informatics, Kyoto University
†2 日本学術振興会特別研究員（DC1）

1. はじめに

近年，検索エンジンは必要とする情報を得る上で欠かせないツールとなってきている．検索エンジンの進歩に伴い，ユーザが明確な検索意図を持っており，的確なクエリを作ることができれば，ユーザは自分の求める情報へ容易にたどり着くことができるようになった．しかしながら依然として，明確な検索意図を持っているにも関わらず，的確なクエリを思いつくことができなければ，求める情報へたどり着くことは困難である．このような状況はさまざま考えられるが，その中でもそもそもクエリをキーワードとして表現することが困難であるものが存在する．例えば以下のような場合がある．

- 戦国時代の出来事といえば，“桶狭間の戦い” と “本能寺の変” があった．しかし，その間に何があったかを思い出すことができない．この2つの出来事の間にあった出来事を知りたい．
- 両親を夕食に誘いたい．大学の近くにはたくさん飲食店がある．○○レストランでは安すぎて両親は満足しないだろう．しかし，××レストランだと高すぎる．前者よりはよいが，後者よりは安いといったようなレストランを見つけたい．
- 友達が Java に関する本を2冊貸してくれた．1冊は私にとって簡単すぎたが，もう1冊は難しすぎて理解できなかった．この2冊の間のレベルの本を探して購入したい．
- 友達から結婚式の招待状が届いた．参列することを決めたのだが，招待状を受け取ってから結婚式に参列するまでに何をすればいいのだろう．
- クイックソートのプログラムを書いている．最初と最後のコードは何を書けばいいのか分かるのだが，その間には何を埋めればいいかわからないため調べたい．

これらの状況では，ユーザはすでに分かっている2つのもの，出来事，プロセスの間にあたる何かを見つけたいと考えている．このような検索はある種の補完検索と考えられる．しかし，従来の検索エンジンではこうした情報を直接的に得ることは困難である．現在の検索エンジンが指定されたキーワードを文書が含むかどうかという基準で結果を返すが，ユーザは知りたいものの名前をそもそも知らないため，直接その名前を用いてクエリを生成できない．そのため，ユーザは欲しい情報に直接到達できないことが原因である．このような情報を得るためにはクエリを工夫したり，似たようなクエリを何度も作り，多くのページを閲覧する必要がある．

そこで本論文では，既知の2つのオブジェクトをクエリとして受け取り，ある観点でオブジェクトを順序づけた場合に，その2つのオブジェクトの間に存在するようなオブジェクト

をユーザに提示する検索手法を提案する．ここでいうある観点とは，先の例であれば，時間，価格，難しさ，順番といったものになる．ユーザが入力した 2 つのオブジェクトの間に位置する最適なオブジェクトを本論文では “補間オブジェクト” と定義する．

提案手法では，ユーザが入力した 2 つのオブジェクトと補間オブジェクトについて Web ページで言及する場合，補間オブジェクトはその 2 つのオブジェクトの間に記述するのではないか，という仮定に基づいて語の位置に着目し，候補語の抽出を行う．その後，出現頻度や出現位置に基づき候補語をランキングし，ユーザに補間オブジェクト候補として提示する．

最後に，提案手法の有用性を示すためにプロトタイプを実装し実験を行った．実験では正解が一意に判定できるようなものを対象とし，正解セットを用意して，補間オブジェクトの抽出精度を求めた．また，クエリを構成する 2 つのオブジェクトのメジャー度の違いや 2 つのオブジェクト間の距離によって精度が変わるかといった実験も行った．そして一意に正解が決まらないような場合の抽出も行い，提案手法の可能性を示した．

2. 補間オブジェクト検索

ある軸上において，ユーザの与えた 2 つのオブジェクトの間にくる適切なオブジェクトを補間オブジェクトと定義する．ここでいう軸とは，2 つのオブジェクトの間にどのオブジェクトがくるべきかを決める観点のことである．例えば，“ナイル川” と “長江” が与えられた時に，“長さ” という観点で考えれば，その間に来るのは “アマゾン川” である．しかし，軸として考えられるのは “長さ” だけでなく，“流域面積” や “深さ”，といったものも考えられる．この選択されるべき軸はユーザの意図によって変わる．

本論文では，ユーザがシステムに対して入力するオブジェクトはテキストでの表現を想定する．入力されるオブジェクトの種類としては，もの，人，組織，集団，場所，出来事，概念，プロセス，階級といった様々なものが考えられる．ここでユーザによって入力される 2 つのキーワードは，何かしらの同じトピックに属しており，求めている補間オブジェクトも同じトピックに属しているという前提で処理する．例えば，ユーザが「ブルース・リー」と「ジャッキー・チェン」を入力として与えた場合，ユーザはこの 2 人（カンフー映画スター）の間に位置するあるカンフー映画スターを求めていると考えられる．

補間オブジェクトを求めるシステムの要件は以下の通りである．

(1) ユーザの入力した 2 つのオブジェクトの間に存在する軸をリストアップする
(2) 取得した軸の中から適切な軸を選択する

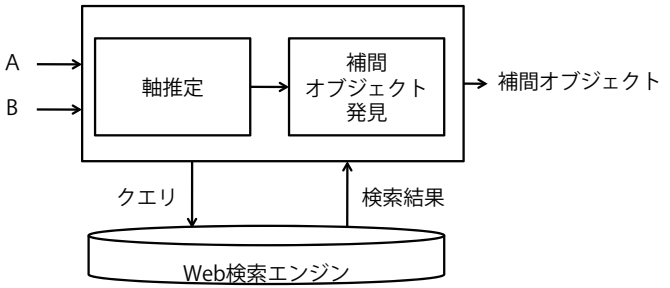

図 1 システム概要

(3) 選択された軸上で補間オブジェクトの候補を取得する
(4) 得られた補間オブジェクト集合をランキングし，ユーザに提示する

これを図示すると図 1 のようになる．

本論文では，(1) 及び (2) については Web 上で最もよく現れていると考えられる軸を対象とする．これは，与えられたオブジェクトの間に最も現れている語を集めることにより行う．(3) と (4) は 4 章で説明する．(1) と (2) に関しては将来的にはユーザが自分の並び替えを行いたい軸を指定することを可能にしたいと考えている．

3. 関 連 研 究

軸は 2 つのオブジェクトの間にくるオブジェクトを決める観点であるが，これは比較検索との関連が深い．Liu ら[1),2)] の手法では，入力として 2 つの Web サイトを与えることで，サイトの構造をツリー状にして提示し，2 つの Web サイト間の差分を発見する．[3)] では，Web サイト間を比べるためのブラウザを開発している．そのシステムではひとつのサイトからページを指定することで，他のサイトから類似するページを発見し，ユーザに提示する．これらの研究では，入力として Web ページや Web サイトを与え，類似点，相違点を比べたり，類似ページを発見して，比較するといったことを目的としている．それに対し，Sun ら[4)] のシステムでは，入力として比較したい 2 つのものをクエリとして与えると、システムはそれぞれのクエリで検索エンジンから検索結果を取得し，類似する検索結果要素をペアにしてユーザに提示することでユーザが指定した 2 つのものの比較を容易にしている．また，クラスタリングやキーフレーズ抽出を行うことにより，様々な観点から検索結果を眺めることが可能になっている．

一方，2 つのエンティティ間に存在する関係性を発見することに重点を置いた研究も行われている．2 つのエンティティの間にある関係性を見つけるクエリとはどのようなものかが[5] で述べられている．例えば，“フランス，ドイツ” というエンティティの間に存在する関係性は “ヨーロッパの国” といったようなものであり、その語の関係性を発見するためのクエリである．こういった関係性を見つけるタスクを予め用意したテキスト集合をソースとして解決しようとする研究に，Zang や Zhai ら[6),7)] の comparative text mining がある．この研究では，2 つのテキスト集合に対してその集合間に存在する共通点や、一方にのみ存在する特徴を発見する．また、同様の問題を解決するために Web ページをソースとするものには[5),8)] があるが，Web ページをソースとして用いた際には，クエリとして入力された 2 エンティティが両方記述されたページがなければ関係を抽出できないという問題がある．それに対し，Luo ら[9)] の手法では，ユーザから 2 つのクエリを受け取り，それぞれのクエリに関して Web 検索を行う．そして，それぞれのクエリで得られた検索結果集合からページペア集合を作成し，ページペアの類似度順に並べ替えを行い，それとともに 2 つのエンティティをつなぐと思われる語をユーザに提示する．この手法では，直接的に 2 エンティティ間の関係性が述べられていなくても，類似する表現を発見することで，2 エンティティの関係を表す語が取得できる．

これらの研究では，2 者間に存在する関係性や比較観点を発見することに重点が置かれており，本研究の軸がこれにあたると考えられる．しかし，これらの研究では与えられた 2 者間での比較，関係性発見を目的としており，その間に当たるものを発見することや，その間のものと比較することは目的ではない．

ユーザが入力した 2 オブジェクトの間にくる候補語としては、そのオブジェクトの同位語が考えられる．同位語とは，“トヨタ” や “日産” に対する “ホンダ”，“ダイハツ” といった共通の上位語をもつ語のことである。この同位語を発見する研究についても多数行われている．Ghahramani ら[10)] の Bayesian Sets では，語の共起テーブルのような大規模なデータに対し，ベイズ推定を用いることで同位語を取得する．山口ら[11)] は，検索時において同位語同士では似たようなクエリが用いられることに注目し，クエリログから同位語を発見する手法を提案している．Lin ら[12)] の提案では，係り受け解析が行われている大規模コーパスを用いて，類似する語のクラスタを発見する．Shinzato ら[13)] の提案では，HTML 構造を用いて同レベルに記述されている語を同位語の候補として取得し，相互情報量，共起度が高いものを同位語として評価する．大島ら[14)] は同位語が記述される際の助詞に着目し，検索エンジンの返すタイトルやスニペットを利用して巨大なコーパスを持たずとも同位語を

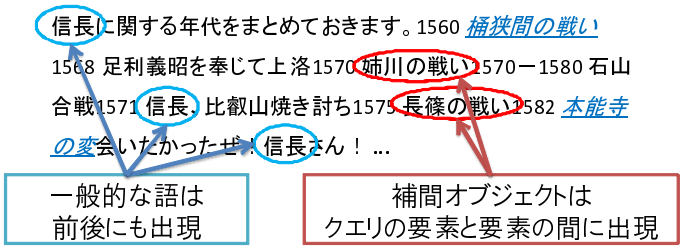

図 2 クエリ：“桶狭間の戦い”,“本能寺の変” の場合

発見する手法を提案している．Web サービスでは，複数の語を与えることで同位語のセットを返す Google Sets[15)] がある．これらの手法は，クエリとして与えたものの同位語が取得できるが，なんらかの観点で見た場合の同位語間の順番といったものは考慮していない．本研究では，なんらかの観点で見た場合に、その同位語がクエリとして与えられた 2 オブジェクトの間に当たるかどうかが重要となる．

4. 手　　法

本論文では，検索エンジンを用いて補間オブジェクトを発見する手法を提案する．ユーザが入力した 2 つのオブジェクトと補間オブジェクトについて Web ページで言及する場合，補間オブジェクトはその 2 つのオブジェクトの間に記述するのではないか，と考えられる．また，a と b の間の補間オブジェクトを求めたいような場合，文章中で a の前や b の後ろでも頻繁に現れるような語は，そのトピックでの一般的な語である可能性が高い．そこでそうした語は適切な補間オブジェクトではないという仮定をおいた．例えば,「桶狭間の戦い」,「長篠の戦い」,「本能寺の変」という 3 つのイベントを考えた場合，これらのイベントはこの順番でページに記載されることが多いと考えられる．そして「信長」や「今川」といったような言葉はページのいろいろな箇所で出現する（図 2 ）．つまり，与えられた 2 つの入力 a, b に対する補間オブジェクト t はウェブページ中に現れる 2 つの入力 a, b の記述の間にのみ出現しやすいのではないかと考えられる．この仮定に基づき，補間オブジェクトの抽出を行う．

4.1 候補語集合の取得

まず，システムは 2 つのオブジェクト名 a と b をユーザからの入力として受け取る．ここで，問題を簡単にするために，ある軸上で a は b より小さいものとする．次に，システム

はクエリ “$a \wedge b$” を作り，検索エンジンにクエリを投げる．システムは検索結果を得た後、検索結果からスニペットを取り出す．システムはそれらのスニペットから a と b の間に出現する語を補間オブジェクトの候補語として収集する．ただし，収集の対象とする候補語の品詞は名詞と名詞句に限る．以降，収集した候補語集合を $C = \{t_1, t_2, \cdots, t_n\}$ とする．

4.2 語の出現頻度と出現位置に基づくランキング

本章冒頭で述べた仮説に基づき a と b の間での語の出現頻度と，a の前や b の後ろで語がどの程度出現しているかという尺度を用いて候補語のランキングを行う．まずは C に含まれるそれぞれの語 t に対し，以下の出現位置による尺度 $F_{a,b}(t)$ を求める．

$$F_{a,b}(t) = \frac{tf_{bet(a,b)}(t)}{tf_{bet(a,b)}(t) + (tf_{pre(a)}(t) + tf_{suf(b)}(t))}$$

$$0 < F_{a,b}(t) \leq 1$$

$tf_{bet(a,b)}(t)$ は a と b の間での t の出現頻度である．$tf_{pre(a)}(t)$ は，a の前での語 t の出現頻度，$tf_{suf(b)}(t)$ は，b の後での語 t の出現頻度である．a の前や b の後に t が多く出現すればするほど，$F_{(a,b)}(t)$ の値は小さくなる．t が a と b の間にだけ現れる場合，$F_{(a,b)}(t)$ の値は最大値 1.0 をとる．候補語集合それぞれの語の出現頻度 $tf_{bet(a,b)}(t)$ の中で最大値を $tf_{max(a,b)}$ とし，$F_{a,b}(t)$ と $tf_{bet(a,b)}(t)$ を用いて以下のようにそれぞれの語のスコアを求める．

$$Rank(t) = \alpha \cdot \frac{tf_{bet(a,b)}(t)}{tf_{max(a,b)}} + \beta \cdot F_{a,b}(t)$$

ただし，

$$\alpha + \beta = 1$$

$$0 < Rank(t) \leq 1$$

この $Rank(t)$ の降順により，候補語のランキングを行い，ユーザに提示する．

5. 実　　験

提案手法の有用性を評価するため，我々はシステムのプロトタイプを実装し，実験を行った．

5.1 補間オブジェクト抽出精度の測定

まず，どの程度正確に補間オブジェクトを抽出できるかどうかを評価するために実験を行った．実験にあたり，“人物”，“出来事”，“出世魚”，“場所’，“作品” の 5 つのカテゴリを用意し，それぞれのカテゴリにおいて，2 つのサブセットとそれをどのように並べるかを決める軸を予め設定した．例えば，“人物” のサブセットである「徳川将軍」では，将軍の代数という軸で，初代 “家康”，2 代 “秀忠”，3 代 “家光”，...，15 代 “慶喜” という順序が正しいとした．自動的に正解，不正解を判定するために，サブセットから少なくとも 1 つ以上の補間オブジェクトを含むようにして 2 つの要素を全て組み合わせてクエリを生成し，そのクエリをプロトタイプ（図 3 ）へ入力するような実験用システムを用意した．先の “徳川将軍” の例では，15 人の歴代将軍から任意の 2 人を取ってくる組み合わせから初代 “家康”，2 代 “秀忠” といったような将軍の代数という軸で考えた場合に，間にくるべきオブジェクトがない組み合わせを除いたものが用いるクエリとなる．実験用システムはプロトタイプから返ってきた結果を正解セットに基づき自動で正誤を判定する．最後に，それぞれのトピックで上位 k 件の候補語に少なくとも正解が 1 つは含まれているクエリの割合を求める．この割合を @k と表す．それぞれのサブセットが持つ要素数，及びそれより生成されたクエリ数，サブセットから作られたクエリ例及びそのクエリが入力された場合の正解例を表 1 に示す．

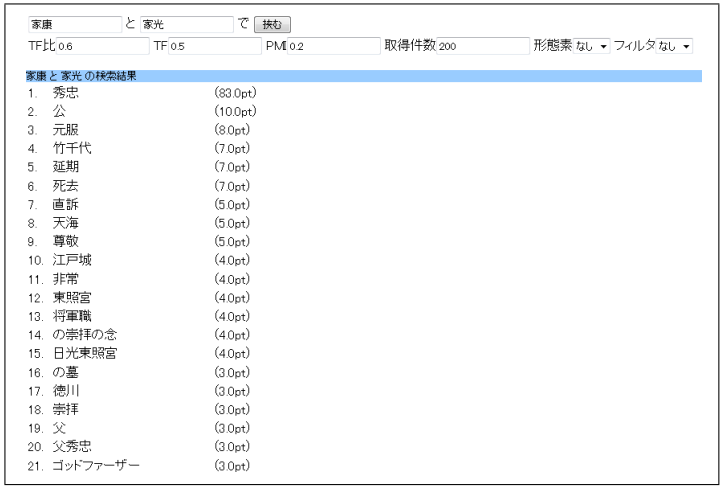

図 3 実装したプロトタイプ

各種パラメータは $\alpha = 0.8$，$\beta = 0.2$ と設定した．検索結果の取得には Yahoo Web Search API[16] を用い，それぞれのクエリについて上位 200 件を取得した．

実験の結果，それぞれのサブセットで上位 k 件の候補語集合が少なくとも 1 つの正解である補間オブジェクトを含む割合は表 2 のようになった．この結果を見ると提案手法は，“徳川将軍家”，“天皇”，“ボラ”，“京都の通り”，“村上春樹”，“伊坂幸太郎” においては上位

Vol.2009-DBS-149 No.4
2009/11/20

表 1 実験で用いたクエリ

カテゴリ	サブセット	要素数	設定した軸	クエリ数	入力例	出力例
人物	徳川将軍	15	将軍の代数	91	家康, 吉宗	家光
	天皇	10	天皇の代数	36	神武天皇, 崇神天皇	開化天皇
出来事	戦国時代	15	発生日時	91	桶狭間の戦い, 本能寺の変	長篠の戦い
	第二次世界大戦	10	発生日時	36	満州事変, ポツダム宣言	ミッドウェー海戦
出世魚	ブリ	6	名の移り変わり	10	ワカナゴ, ブリ	ハマチ
	ボラ	6	名の移り変わり	10	ハク, トド	スバシリ
場所	京都の通り	24	北からの順序	276	御池, 五条	四条
	阪急京都線	26	梅田からの駅の順序	300	茨木市, 高槻市	総持寺
作品	村上春樹	12	発表順	55	ノルウェイの森, 海辺のカフカ	ねじまき鳥 クロニクル
	伊坂幸太郎	9	発表順	28	オーデュボンの祈り, 重力ピエロ	ラッシュライフ

表 2 候補語集合の上位 k 件 (@k) が 1 つでも正解を含む割合

	a1	a3	a5
徳川将軍家	94.51 %	97.80 %	97.80 %
天皇	75.00 %	97.22 %	97.22 %
戦国時代	13.19 %	23.08 %	29.67 %
第二次世界大戦	25.00 %	38.89 %	47.22 %
ブリ	50.00 %	80.00 %	80.00 %
ボラ	90.00 %	100.00 %	100.00 %
京都の通り	80.43 %	88.41 %	92.39 %
阪急京都線	33.33 %	53.00 %	60.33 %
村上春樹	80.00 %	90.91 %	92.73 %
伊坂幸太郎	89.29 %	92.86 %	96.43 %
平均	57.77 %	70.31 %	75.03 %

1 件のみを見た場合でも，75% 以上の精度で補間オブジェクトを抽出できていることが分かる．それに対し，“戦国時代”，“第二次世界大戦”，“阪急京都線” では精度が低くなってしまっている．“戦国時代”，“第二次世界大戦” で精度が低くなってしまった原因としては，どこまでを正解とするかという粒度の問題が考えられる．今回は正解順序をある観点で決めたサブセットに基づきシステムが自動的に正誤判定を行ったが，実際には正解である候補語まで不適であると判断してしまっている可能性がある．例えば，“長篠の戦い” と “本能寺の変” の間には “安土城建設” があったが，予め設定したサブセットには含まれていないため，“安土城建設” が抽出されても不正解として扱われてしまった．このように，どこまでを正解と扱うかという問題が精度を下げてしまった一因として考えられる．全体的に精度を下げてしまった要因としては，クエリが一般的な語であればあるほど様々な話題の中でその語が出現するので，その語のコンテキスト（話題）が限定されず，不適合な候補語が大量に抽出されてしまったということが挙げられる．例えば，“阪急京都線” で “(南茨木, 水無瀬)” というクエリの場合に，“新築マンション” という語が抽出されていた．これは阪急の駅名というコンテキストに限定されていないため，沿線のマンションの話題が検索されてしまったことが原因であると考えられる．しかし，今回はこの場合の軸を “阪急の駅の順番” としているため不正解であるが，南茨木駅と水無瀬駅の間にこの新築マンションがあり，軸は位置的なものと考えた場合，正解と考えることもできる．よって，今後軸をユーザがうまく指定できる仕組みを考えていくと同時に，その語が抽出されたのは，こういうコンテキストにおける場合であったということを示せるような仕組みを作っていく必要があると考えられる．

また，クエリを構成する語の検索ヒット数が多いか少ないかといったことも考えなければならない．検索ヒット数が多くなるほど様々なコンテキストがあると考えられる．この違いについては 5.2 節で考察を行う．一方，ユーザが入力する 2 つのオブジェクトによって正解となる補間オブジェクトの個数は異なる。つまり、その数によっても精度が変わってくるのではないかと考えられる．ここで、ユーザが入力する 2 オブジェクト間に含まれる補間オブジェクトの個数が増えれば増えるほどコンテキストに広がりが出てくるため，精度が落ちていくのではないかと予想される．これについては 5.3 節にて検討する．

5.2 入力オブジェクトの記載量による抽出精度の違い

5.1 節より，クエリを構成する 2 つの要素それぞれの検索ヒット数の度合いによって結果が変わる可能性について言及した．そこで，クエリを構成する 2 要素の検索ヒット数の違いにより，精度がどの程度変わるのかを調べた．正解セット $R = \{t_1, t_2, \ldots, t_N\}$ に対し，以下を満たす時，t_n はメジャーであると定義する．逆に以下を満たさない時，t_n はマイナーであるとする．

$$HitCount(t_n) \geq \mu \cdot \frac{1}{N} \sum_{t \in R} HitCount(t)$$

$HitCount(t)$ とは，クエリ t に対する検索エンジンの返す検索ヒット数である．今回は，$\mu = 0.3$ とした．クエリがメジャーな語とメジャーな語で構成されている場合，メジャーな語とマイナーな語で構成されている場合，マイナーな語とマイナーな語で構成されている場合のそれぞれにおいて表 1 で用いたクエリを用い，その正答率を求めた．その結果，表 3 のようになった．

平均をとると，それぞれの結果に有意な差は見られなかったが，個々の結果を見ていくと

大きな差が見られるものがあった．“天皇” では，メジャー・メジャーの組み合わせに比べると，メジャー・マイナー，及びマイナー・マイナーのほうが精度がとても良い．“阪急京都線’ でも同様の傾向が見受けられる．これは，メジャーな語同士であると様々なコンテキストが考えられるため，ノイズとなる候補語が多く含まれてしまう．しかしマイナーな語を含むと，コンテキストが限定されやすく，また一覧で載っているようなページがヒットしやすくなるために精度が向上すると考えられる．このようにクエリを構成する語がマイナーなためにコンテキストを限定する働きをする場合に精度の向上が見られることが分かった．

5.3 正解補間オブジェクト数による抽出精度の違い

5.1 節においてクエリを構成する 2 つのオブジェクトに挟まれる補間オブジェクトの数により，精度に違いが出るのではないかという可能性について言及した．そこで，2 つのオブジェクトに挟まれる正解補間オブジェクト数で分類し，その精度の違いを比較した．正解補間オブジェクトの数は $1, 2, 3, \ldots, 9$ の場合と 10 個以上の場合の 10 通りに分類した．それぞれの場合における精度は図 4 の通りである．

図 4 から，正解補間オブジェクトの数が増えるに従い精度はやや増加していき，ほぼ横ばいになることが分かる．これは，正解となる補間オブジェクトの数が少ない間は，正解となる数が少ないために精度が低くなるが，正解補間オブジェクトの数が増えるに従い，正解を含む率が上がるが，クエリを構成する語と語のコンテキストが様々なものを含むようになり，連続して他の補間オブジェクトと記述されていることが少なくなるために，精度が低下し，結果としてほぼ横ばいになったのではないか，と考えられる．しかし，まだ十分な実験数であるとは言えないため，今後，実験数を増やしてより正確な精度を確かめていきたいと考えている．

5.4 正解が一意に決められないものの抽出

これまでの実験では，徳川将軍の順序，京都の通りを北から順番に、といったような正解が一意に決まるようなものを扱った．この節では正解が一意には決まらないようなものについて実例を示しながら説明する．

棋士　入力：(大山康晴，羽生善治)

60 年代のトップ棋士だった大山と 90 年代前半のトップ棋士になった羽生の間にあてはまる強い棋士を見つけようという意図で “大山康晴” と “羽生善治” をクエリとして与えたとする．システムがこのクエリによって抽出したものは “中原誠”,“米長邦雄”,“谷川浩司” であった．実際にこの 3 人は有名であり，大山と羽生がそれぞれトップ棋士であった時期の間の時期に彼ら 3 人はトップ棋士であった．この結果により，中原は 2 人の間の時期に最強

表 3 クエリのメジャー・マイナーの組み合わせを考慮した際の上位 k 件の少なくとも 1 つ正解を含む割合

トピック	クエリの組み合わせ	a1	a3	a5
徳川将軍家	メジャー・メジャー	93.8 %	96.9 %	96.9 %
	メジャー・マイナー	97.8 %	97.8 %	97.8 %
	マイナー・マイナー	84.6 %	100.0 %	100.0 %
天皇	メジャー・メジャー	66.7 %	88.9 %	88.9 %
	メジャー・マイナー	68.4 %	100.0 %	100.0 %
	マイナー・マイナー	100.0 %	100.0 %	100.0 %
戦国時代	メジャー・メジャー	16.7 %	27.8 %	44.4 %
	メジャー・マイナー	14.0 %	22.0 %	26.0 %
	マイナー・マイナー	8.7 %	21.7 %	26.1 %
第二次世界大戦	メジャー・メジャー	33.3 %	44.4 %	44.4 %
	メジャー・マイナー	26.3 %	31.6 %	42.1 %
	マイナー・マイナー	12.5 %	50.0 %	62.5 %
ブリ	メジャー・メジャー	なし	なし	なし
	メジャー・マイナー	66.7 %	83.3 %	83.3 %
	マイナー・マイナー	25.0 %	75.0 %	75.0 %
ボラ	メジャー・メジャー	100.0 %	100.0 %	100.0 %
	メジャー・マイナー	83.3 %	100.0 %	100.0 %
	マイナー・マイナー	なし	なし	なし
京都の通り	メジャー・メジャー	76.5 %	86.3 %	96.1 %
	メジャー・マイナー	82.3 %	88.7 %	90.1 %
	マイナー・マイナー	79.8 %	89.3 %	94.0 %
阪急京都線	メジャー・メジャー	19.0 %	42.9 %	53.6 %
	メジャー・マイナー	34.6 %	50.6 %	57.1 %
	マイナー・マイナー	50.0 %	73.3 %	78.3 %
村上春樹	メジャー・メジャー	0.0 %	0.0 %	0.0 %
	メジャー・マイナー	70.6 %	82.4 %	88.2 %
	マイナー・マイナー	86.5 %	97.3 %	97.3 %
伊坂幸太郎	メジャー・メジャー	100.0 %	100.0 %	100.0 %
	メジャー・マイナー	84.6 %	92.3 %	100.0 %
	マイナー・マイナー	91.7 %	91.7 %	91.7 %

Vol.2009-DBS-149 No.4
2009/11/20

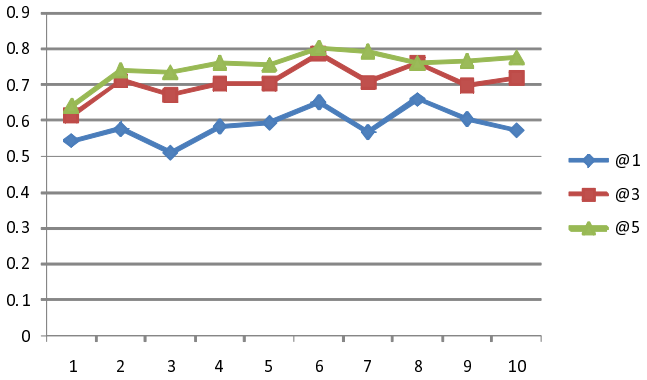

図 4 クエリとクエリの間の補間オブジェクト数ごとによる正解率の違い

棋士であったことがわかる．この結果は概ね正しいのではないかと考えている．

観光地　入力：(金閣寺，銀閣寺)

金閣寺より良くて銀閣寺ほどではない観光地を見つけようと意図で “金閣寺” と “銀閣寺” をクエリとして与えたとする．システムが抽出した結果は “二条城”,“花園”,“平安神宮”,“清水寺” であった．これらは結果としては悪くないが，単に有名な観光地が出ただけ，と捉えることもできる．その観光地がどれくらい人気か，どのくらい古いか，どのくらいアクセスが容易か，といったような軸でも比較することが可能である．つまり，システムは単に候補語を出すだけではなく，なぜその語が選ばれたのかといった根拠も提示する必要がある．

駅　入力：(京都駅，東京駅)

駅と駅の間に位置する主要な駅を求めようという意図で “京都駅” と “東京駅” というクエリを入力したとする．システムが抽出したのは “自由に”,“バス”,“西本願寺” といったようなものばかりであった．一方で，“東京駅” と “京都駅” という順序で入力した場合は，“品川駅” や “新横浜駅” といった良い結果が得られた．これは，結果が a と b の向きに依存しているということである．加えて，それぞれの距離が短いような 2 駅を入力した場合は比較的正しいものが抽出できるが，距離が遠くなると関係性が乏しくなり，結果が悪くなることがあった．これは 5.3 節で考察した通りである．

湖　入力：(ヒューロン湖，オンタリオ湖)

アメリカとカナダにまたがる五大湖のヒューロン湖とオンタリオ湖の間に位置する湖の名前を調べようと思い “ヒューロン湖” と “オンタリオ湖” という 2 オブジェクトをクエリとして入力したとする．システムは “ミシガン湖” をトップ候補として抽出した．ミシガン湖はヒューロン湖とオンタリオ湖の間に位置するので，これは正しい結果である．このように五大湖の名前を入れた際に，システムは “位置” という軸で正しく結果を抽出できたといえる．ここで考えなければならないのは，比較する観点は位置だけに限らず，“表面積” や “水量”，“最大深度” も考えられる．将来的にこういったものもユーザの意図に合わせて自由に抽出できるようにしたいと考えている．

6. 考　　察

実験より，クエリを表す語がどういった話題で使われているのかといったコンテキストを考慮することの重要性がわかった．コンテキストを制限する方法として考えられるのは，明示的にユーザがクエリにコンテキストを限定するキーワードを追加するといった方法がまず考えられる．この方法を用いれば，ユーザの意図に沿ってコンテキストを絞ることができる．しかし，ユーザが的確な絞り込むための語を知っていなければならないといった問題に直面する．別の方法として考えられるのは，軸を表す語，つまり比較観点となるような語をうまく抽出してくることで，その語を用いてユーザが軸の選択を可能にしたり，その語によりコンテキストを絞ることである．本論文では，軸を選択することができなかったが，今後は上で挙げたような方法を用いて軸の選択を可能にするとともに，精度の向上を目指していく予定である．

次に，補間オブジェクト抽出に用いたソースについて考察する．本論文では，補間オブジェクトの抽出に際し，検索結果のスニペットのみを使用した．しかし，スニペットの文書長はきわめて短いため，それほど情報量がない．そのため，我々のシステムが抽出できるものは Web サーチエンジン側のスニペット生成アルゴリズムに依存してしまう．例えば，a と b の補間オブジェクト $t_1, t_2, t_3, t_{4,5}$ が存在するとする．検索エンジンにクエリ “a” と “b” を与えると，検索エンジンはクエリ近傍のテキストを表示するため，a と t_1,t_5 と b のみを含むような検索結果を返す可能性が高いと考えられる．もしこの仮説が正しい場合，スニペットのみを利用するだけでは，t_1 や t_5 以外を得ることが困難である．スニペットを利用する場合は，得られた t_1 や t_5 を新たにクエリとして用いることによって，残る t_2, t_3, t_4 を取得していくといった方法が考えられる．このように最初に与えたクエリに加え，得た補間オブジェクトも利用することにより，多くの補間オブジェクトを得ていくことが可能であると考えられる．また，スニペットだけではなく，スニペットとして表示されている結果の

オリジナルの Web ページ自体にアクセスして全ての情報を取得することで，一気に全ての候補を得ることができる可能性がある．しかしこの場合，スニペットと異なり，文書長が長くなるためアルゴリズムの改良が必要となる．

また，今回は共起を用いて補間オブジェクトの発見を行っているため，取得できるものは共起しているものに限られてしまう．共起しているもの以外を得るためには，その語の周辺語句や比較観点となる語を用いてオブジェクトとオブジェクトをつなぎ，評価を行っていく必要がある．今後，そのような共起を用いただけでは発見できないようなオブジェクトの発見に取り組んでいく予定である．

7. まとめと今後の課題

本論文では Web 検索エンジンを用いて 2 つの入力の補間オブジェクトを発見する手法の提案を行った．提案手法では，まず候補語を収集し，その語の記述位置と出現頻度を用いてランキングを行い，ユーザに提示する．

実験では、提案手法を用いた場合に、2 オブジェクトからどの程度補間オブジェクトが正確に得られるかを示した。その結果，いくつかのクエリにおいては 75% 以上の精度を示し，失敗例からはコンテキストを考慮することが必要であるという知見を得られた．また、2 オブジェクトのクエリとして与えたメジャー度を考慮した場合に抽出精度がどれくらい違うのか，クエリとして与えた 2 オブジェクト間の補間オブジェクトの数によって精度がどの程度違うのかといったことを示した．

今回は適切な補間オブジェクトを 2 つの入力の間に最も現れたものとしたが，場合によっては 2 つの入力の平均をとるものが適切な補間オブジェクト（例えば 2 つの本の値段の平均の値段を持つ本）であるかもしれない．適切な補間オブジェクトとは何かということを今後考えていく必要がある．また，今回は軸（比較観点）を明示的に扱わなかったが，今後は形容詞に着目したり，比較表現 “〜より” を利用することで，軸の発見やユーザの明示的な選択といったことに取り組んでいく予定である．そして，2 つのオブジェクトの内側だけでなく，外側に存在するオブジェクト（大正，明治なら昭和、平成）も発見できるように手法の拡張を考えていく予定である．

謝辞 本研究の一部は，グローバル COE 拠点形成プログラム “ 知識循環社会のための情報学教育研究拠点 ”，計画研究 “ 情報爆発時代に対応するコンテンツ融合と操作環境融合に関する研究 ” (研究代表者 : 田中克己，A01-00- 02, 課題番号 18049041)，未踏 IT 人材発掘・育成事業 2009 年度上期 未踏ユースによるものです．ここに記して謝意を表すものとします．

参 考 文 献

1) B. Liu, Y. Ma, and P. S. Yu, “Discovering Unexpected Information from Your Competitors’ Web Sites”, In Proceedings of KDD ’01, pages 144-153, 2001.
2) B. Liu, K. Zhao, and L. Yi, “Visualizing Web Site Comparisons”, In Proceedings of WWW ’02, pages 693-703, 2002.
3) A. Nadamoto and K. Tanaka, “A Comparative Web Browser (CWB) for Browsing and Comparing Web Pages”, In Proceedings of WWW ’03, pages 727-735, 2003.
4) J. Sun, X. Wang, D. Shen, H. Zeng, and Z. Chen. “ CWS: A Comparative Web Search System ”, In Proceedings of the 15th International Conference on World Wide Web, pp. 467- 476, 2006.
5) D. Mahler, “Holistic Query Expansion Using Graphical Models”, New Directions in Question Answering 2004, pp.203-214.
6) P. Zang, “CTMs: A Comparative Text Mining System”, Master thesis, University of Illinois at Urbana-Champaign, Computer Science Department, 2004.
7) C. Ahai, A. Velivelli, and B. Yu, “A Cross-collection Mixture Model for Comparative Text Mining”, In Proceedings of KDD ’04, pp. 743-748, 2004.
8) S. M. Harabagiu, V. F. Lacatusu, and A. Hickl, “Answering Complex Questions with Random Walk Models”, In Proceedings of SIGIR 2006, pp220-227, 2006.
9) G. Luo, C. Tang, and Y. Tian, “Answering Relationship Queries on the Web”, In Proc. of the 16th international conference on World Wide Web, pp. 561-571, 2007.
10) Z. Ghahramani and K. Heller, “Bayesian sets”, In Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS2005), 2005.
11) 山口 雅史, 大島 裕明, 小山 聡, 田中 克己, “サーチエンジンのクエリログを利用した同位語の発見”, DBSJ Letters, Vol.5, No.2.
12) D. Lin, “Automatic Retrieval and Clustering of Similar Words”, In Proceedings of the 36th annual meeting on Association for Computational Linguistics, pp. 768-774, 1998.
13) K. Shinzato, K. Torisawa, “A Simple WWW-based Method for Semantic Word Class Acquisition”, In Proceedings of the Recent Advances in Natural Language Processing (RANLP05), pp.493-500, 2005.
14) H. Ohshima , S. Oyama, K. Tanaka, “ Searching Coordinate Terms with Their Context from the Web ”, In Proceedings of WISE 2007, pp. 40-47, 2006.
15) Google Sets http://labs.google.com/sets
16) Yahoo! デベロッパーネットワーク http://developer.yahoo.co.jp/webapi/search/