クエリログとナビゲーション履歴からの探索意図抽出による協調探索支援

川本　淳平†　　田中　克己††　　田島　敬史††

† 京都大学工学部情報学科 〒606-8501 京都市左京区吉田本町
†† 京都大学大学院情報学研究科社会情報学専攻 〒606-8501 京都市左京区吉田本町
E-mail: †kawamoto@dl.kuis.kyoto-u.ac.jp, ††{ktanaka,tajima}@i.kyoto-u.ac.jp

あらまし　協調探索において，探索領域が狭くなってしまう問題を支援するために，各々のユーザが探索している領域に関する情報を共有し視覚化するシステムを提案する．本研究では，ユーザの探索情報を表すものとしてクエリログに着目した．クエリログとは，検索に使用したクエリが時系列順に記録されているだけのものであり，そのままでは各クエリがどのような文脈で用いられたのかを判断することは困難である．そこで，ナビゲーション履歴を基にクエリログを木構造として表現し直すことで，どのような領域の探索を行っているのかという探索意図の抽出を行っている．また，作成した木構造を基に，未だ探索されていない領域の発見と提示手法についても述べる．

キーワード　情報検索，協調探索，視覚化，周辺情報

1. はじめに

膨大な Web ページ群の中を探索する場合，一人で探索するよりも複数人で探索する方が適している場合がある．例えば，旅行のプランニングや，ある事柄についての情報を網羅的に集めたい場合などである．前者の場合，旅行に参加するユーザはそれぞれ興味や知識（歴史的建造物に詳しい人，郷土料理に詳しい人など）が異なるため，分担して各ユーザが得意分野を探索する方が良い．また後者の場合，関連あるページを出来る限り多く探索することが求められるため，人海戦術が有効であると言える．

しかし，ただ単純に人数分の端末を用意して探索を行うと，各ユーザが似た検索クエリを使用し，似た検索解を得，それらを上位から順に検討していくことになりやすい．これでは複数人で探索を行っているのに，幅広い情報を吟味することができない．そこで本研究では，このようなユーザ同士がお互いに連絡を取り合って意思決定を行うような協調探索において，探索範囲が狭くなってしまうという問題を防ぐために，各ユーザ間での探索情報の共有を支援するとともに，どのような事柄が既に探索されているのか，また探索されていないのかを視覚化して提示するシステムを提案する．

対象とする協調探索は，次のような条件で情報探索を進めるという状況を想定している．

- 参加ユーザは一斉に探索を開始する
- Yahoo!検索 [1] のようなサーチエンジンを用いて検索を行う
- 検索解ページからリンクを辿り探索を行い，任意で推薦するページにはマーキングを行う

幅広い情報の探索を目指す協調探索においては，各ユーザの探索情報をいかに共有するかが重要となる．共有する情報が多すぎる場合，それらを吟味することに時間が使われてしまい探索がなかなか進まなくなってしまう．逆に少なすぎる場合，前述のように探索範囲が狭くなる問題が発生することになる．

そこで，本論文で提案するシステムでは，ユーザの検索履歴を表すクエリログに着目し，ナビゲーション履歴を基に，クエリ間の詳細関係や兄弟関係を俯瞰できるように木構造として表現し共有することで，情報量を増やし過ぎずに，どのような話題領域の探索が行われているかを判りやすく視覚化する．

本論文の 2 章では関連研究について述べ，3 章では本提案手法にて作成する QueryLogTree の概要について述べる．4 章では 3 章にて紹介した QueryLogTree の作成手順について，5 章で協調探索への利用法について述べる．6 章でシステムの実装と実験結果の考察について，最後に 7 章でまとめと今後の課題について述べる．

2. 関連研究

2.1 コンテクスト依存型 Web ブックマーク

上田ら [5] は協調探索において，ユーザが推薦したページがどのような探索過程で選ばれたページであるかを視覚するために，検索結果ページから選ばれたページまでの閲覧履歴を基にコンテクスト依存型 Web ブックマークを定義している．コンテクスト依存型 Web ブックマークは，

- 多くのページを閲覧した中から選ばれたページのランク値は高い
- 閲覧ページ群の内容が，選ばれたページと近ければランク値は高い

という観点から定義されている．即ち，たくさんの似た内容のページ群から選ばれたページはランク値が高いことになる．

コンテクスト依存型 Web ブックマークでは，検索結果ページから推薦ページに辿り着くまでの過程を対象としている点が，

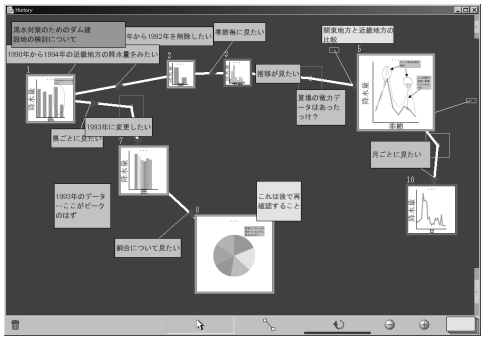

図 1　InTREND の履歴表示画面

検索アクションの意図を推定している本研究との違いである．

2.2　探索履歴の利用

探索履歴を利用した研究としては，松下ら [6] の InTREND がある．InTREND ではユーザが自分の探索履歴を俯瞰するためのインターフェイスとして図 1 のような履歴表示画面を採用している．この探索履歴では，クエリだけでなく，閲覧ページのサムネイルやユーザ自身が入力したアノテーションなど多くの情報が含まれている．

このように，探索履歴を利用した研究では，探索履歴に様々な情報を付加してユーザの探索を支援する研究が多い．しかし協調探索においては，前述のように共有する情報が多くなると，それらの吟味に時間を取られるため好ましいとは言えない．そのため本研究では，どのページを見たのか，どういったリンクを辿ったのかという詳細な情報ではなく，どの領域の探索が既に行われたのかを俯瞰できることを目指している．

2.3　KeyGraph

大澤ら [4] は，文章の内容を推定するために，文章中の重要な単語を抽出し，それらを単語同士の関係に基づいたグラフとして表示するシステムを提案している．KeyGraph では，対象が文章であり，文章中の語を抽出しグラフ化する．対して本研究では，文章中の語の関係ではなく，ユーザが使用したクエリの関係を可視化している．また，本研究では，協調探索というリアルタイムで進んで行く探索に合わせて木を作成しなければならないため，極力簡単な計算で作成できるよう努めている．

3.　QueryLogTree

本研究では前述のように，クエリログを各検索の意図，文脈を俯瞰できるように木構造に変換し共有する．本章では，探索情報として共有する木構造 QueryLogTree の概要について紹介する．

探索意図や文脈を俯瞰するためには，どのような話題が探索されたのか，また関連のある話題にはどのようなものがあるのかが判別できる必要がある．しかし，クエリログは，検索に使われたクエリが時系列順に記録されているだけのものであり，目的達成のためには次のような問題がある．

（1）　どこからどこまでが同一話題の検索であったのかを判断することが難しい

（2）　絞込み検索を行っているのか，言い換え表現による検索を行っているのかという各クエリ間の関係を読み取ることが難しい

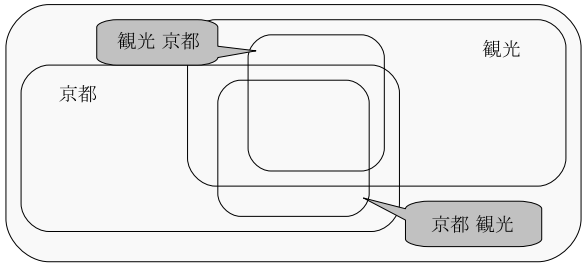

図 2　話題キーワードの順序による探索領域の違い

こうした問題を解決するため，クエリログを木構造として表現しなおすことで各クエリによって得られる話題領域の関係を視覚化する．

以下では，まず QueryLogTree で利用する概念や定義について説明し，その後 QueryLogTree を定義する．

3.1　クエリを表すキーワード式

検索に用いられるクエリは，話題を表すキーワードを詳細化する順にスペースで区切って入力し AND 検索を行うものとする．OR 検索は本研究では扱わない．キーワード間の関係が対等である場合は，より主話題であると思える方を先に記述するか別々のクエリとして検索を行うものとする．例えば，図 2 に示すように「京都 観光」というクエリは京都という主話題に対し観光という副話題の検索を表し,「観光 京都」というクエリは観光という主話題に対し京都という副話題の検索を表すとする．

以降では，あるクエリ Q が「$a\ b$」であることを，$Q = a\ b$ と書くこととする．また，特に断りの無い限りキーワード式中の小文字アルファベット $a, b \cdots$ はキーワードを表すこととする．

3.2　探 索 単 位

ユーザがあるクエリによる検索を行ってから，次のクエリによる検索を行うまでの一連の行動を探索単位と呼ぶことにする．探索単位内では，ユーザは検索結果ページからリンクを辿り，ページを閲覧し任意に推薦するページにマーキングを施すという操作を繰り返すことが出来る．

以降では，i 番目に実行された探索単位のことを $Unit_i$ と書き，最も新しい探索単位で，QueryLogTree に追加されようとしているものを $Unit_{new}$ と書くことにする．

3.3　探索情報を表す木構造

3.3.1　クエリ間の親子関係

QueryLogTree では，話題の絞込みを可視化するために，親クエリの話題を探索過程として詳細化するクエリを子として接続する．例えば,「Q = 京都観光」の子として「Q = 京都観光 慈照寺」を接続する．

しかし，上のような単純な木構造に変換しただけでは，言い換え表現等による検索を判別することが出来ない．例えば，慈照寺の通称「銀閣寺」を用いた「Q = 京都観光 銀閣寺」も，別内容の探索を目的とした「Q = 京都観光 金閣寺」も，共に「Q = 京都観光」の子として接続される．そのため,「Q = 京都観光 慈照寺」と「Q = 京都観光 銀閣寺」は同じ話題の探索で

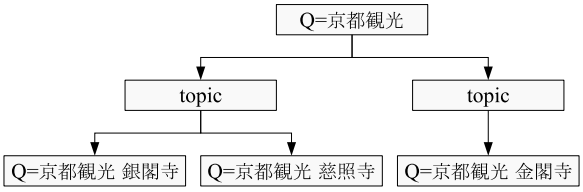

図 3　topic の追加

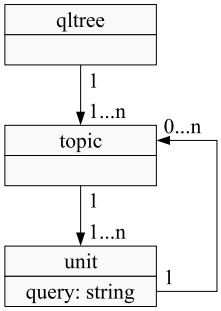

図 4　QueryLogTree のスキーマ

あり「Q = 京都観光 金閣寺」は異なる話題の探索であることを判別できない．

そこで，木に topic という節点を追加し，共通話題を扱うクエリを topic で束ねて親クエリへと接続することで共通話題の可視化を図る．先ほどの例の場合，図 3 に示すように,「Q = 京都観光 慈照寺」と「Q = 京都観光 銀閣寺」を同じ topic 要素の子とすることで共通話題であることを表し，別の topic 要素の子である「Q = 京都観光 金閣寺」とは異なる話題の探索であることを表す．

3.3.2　QueryLogTree のスキーマ

前節までの定義を踏まえて，本論文では QueryLogTree を図 4 のスキーマを持つ XML データで表現する．各要素の内容は次のとおりである．

qltree

ユーザ毎のルート要素である．子要素には探索した話題を持つ．

topic

探索内容の話題表す．子要素には，この話題に関わる探索単位集合を持つ．

unit

探索単位を表す．子要素には，詳細化する話題集合を，属性としてキーワード式を持つ．

3.4　QueryLogTree の例

例えば，クエリログが

（1） 広島 郷土料理

（2） 美酒鍋

（3） 山ふぐ

（4） 広島 交通機関

である探索の場合，図 5 の様な QueryLogTree を作成する．

この探索では，広島についてのグルメに関する情報と交通に関する情報を探索しており，ルート要素直下に 2 つの topic 要素が来ている．グルメ情報の topic 要素の子要素には「郷土料理」が，郷土料理を詳細化する話題として,「美酒鍋」「山ふぐ」

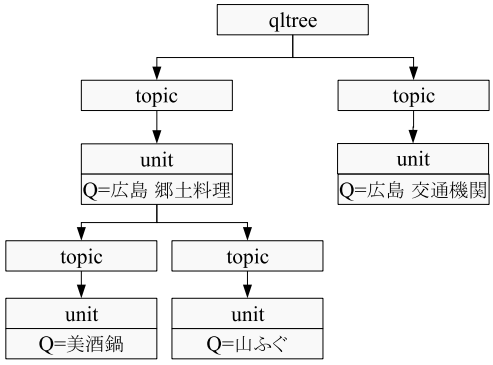

図 5　QueryLogTree の例

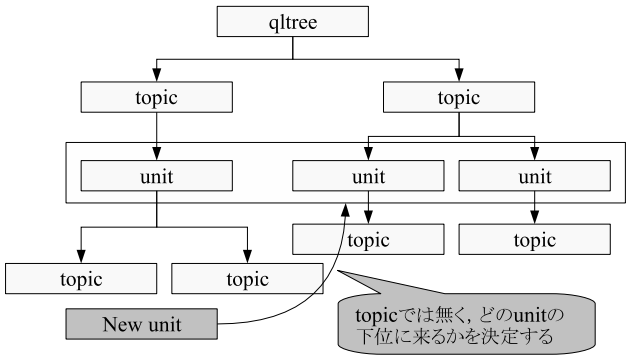

図 6　探索単位の親子関係の決定

と言った広島の郷土料理に関する探索が接続されている．

ここで，広島の郷土料理を詳細化する話題として接続されている 2 つのクエリは，他のクエリと異なり「広島」を含んでいない．しかし,「広島」を含まなくても「美酒鍋」「山ふぐ」と言ったクエリにより得られる話題は「広島 郷土料理」をクエリとして得られる話題の部分集合になっており，子として接続される必要がある．

4.　QueryLogTree の作成

本章では，実際に探索で得たクエリログから，3 章にて紹介した QueryLogTree を構築するアルゴリズムを説明する．QueryLogTree は，新たな探索単位が実行されるたびに拡張していく．ここで重要となるのが，QueryLogTree は協調探索実行時に，探索情報を共有するために用いられるということである．即ち，ユーザがリアルタイムで探索を行うのと同程度の計算時間で作成できなければならない．また，本来の探索を妨げないように，利用するマシンパワーも出来るだけ抑える必要がある．これらの条件を踏まえ，QueryLogTree の構築を行う．

4.1　QueryLogTree の拡張手順

QueryLogTree の拡張は,

（1） 新しい探索単位がどの探索単位の子であるのかの判定

（2） 同一話題に対する探索のグルーピング

という二段階で行う．

つまり，第一段階では，図 6 に示すようにどの探索単位を詳細化するクエリであるかの判定を行い．第二段階では，図 7 に示すように，兄弟要素間で共通話題の探索か否かを判定する．

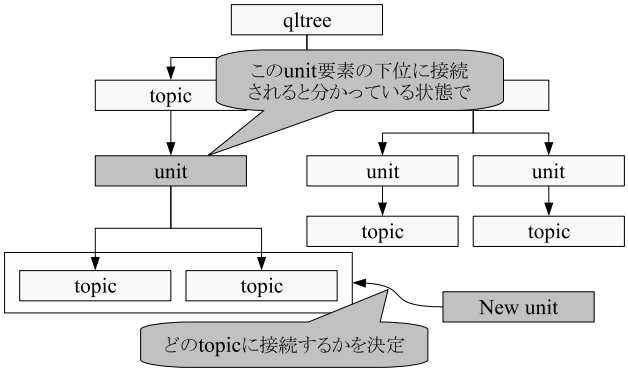

図 7　トピックの決定

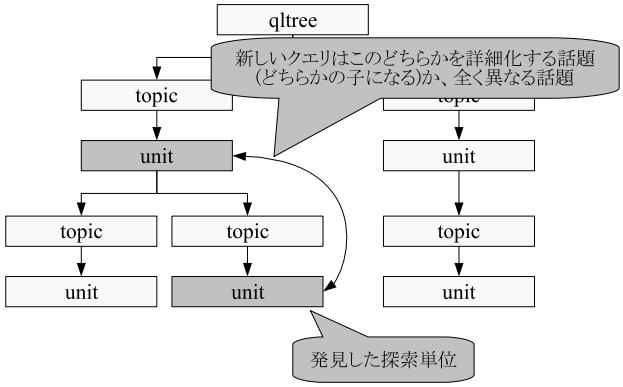

図 8　見つかった探索単位と新しいクエリの関係

4.2　探索単位の親子関係の決定

クエリに $Q_{new} = ab\cdots$ を持つ unit 要素 $Unit_{new}$ を追加する場合を考える．この場合，次の 3 つの可能性が考えられる．

（1） Q_i のキーワード全てを Q_{new} が含む様な $Unit_i$ が存在する

（2） Q_{new} と Q_i には共通キーワードと共通で無いキーワードの両方ある様な $Unit_i$ が存在する

（3） 共通キーワードを持つ unit 要素が存在しない

それぞれの場合の接続方法は次の様になる．また，親となりうる要素が複数存在する場合については後述する．

4.2.1　Q_i のキーワード全てを Q_{new} が含む様な $Unit_i$ が存在する場合

この場合の接続規則は，

見つかった要素の子として接続する．

とする．なぜなら，Q_{new} は Q_i にキーワードを追加したものであり，絞込検索を行っていると言えるからである．

4.2.2　Q_{new} と Q_i には共通・非共通キーワードの両方ある様な $Unit_i$ が存在する場合

この場合の接続規則は，

見つかった要素の兄弟として接続する．

とする．なぜなら，Q_{new} と Q_i には共通キーワードがあり，そこに異なるキーワードを追加している．これは，この 2 つのクエリがある話題を異なる方向に詳細化していると言え，同一 topic 要素の子として接続すべきだからである．

4.2.3　共通キーワードを持つ unit 要素が存在しない場合

この場合，上記 2 つの場合ほど簡単に接続先を決定できない．そこで，

新たなクエリは探索の過程で発見・想起される

と言う事を仮定する．なぜなら，ユーザが探索途中に新たにクエリを想起するのは，探索の過程で得た情報に因るところが大きく，何の情報も無く突然新たなクエリを思いつくことは少ないと考えられるからである．全く別の話題に関して探索を始める場合は突然思いつくこともあるが，現在の探索を詳細化するようなクエリに関しては，追加キーワードは今までに見た単語を利用することが多いと言える．

そのため，各探索単位 $Unit_i$ において，ユーザが閲覧したページに含まれる単語集合 $Word_i$ を作成し，クエリと閲覧ページに含まれる単語の関係を利用する．単語の抽出には，名詞句まで考慮した [8] で用いられている手法を利用する．

クエリ Q_{new} と各 $Unit_i$ の閲覧ページに含まれる単語集合 $Word_i$ に対して，次の条件を満足するものを探す．

$$\forall a \in Q_{new} \text{ に対し } a \in Word_i$$

$Unit_i$ が見つからなかった場合，$Unit_{new}$ は今までの探索とは全く別の話題に関する探索であると考え，新しいクエリはルート要素に接続する．

$Unit_i$ が見つかった場合，新しいクエリは，$Unit_i$ または上位の探索単位を詳細化するクエリであると考える（図 8）．なぜなら，キーワードがある探索単位で見つかったと言うことは，少なくともその探索内容に関係のあるキーワードであると考えられる．しかし，どの話題を詳細化するキーワードであるかは現時点では判定できない．例えば,「広島 郷土料理」というクエリによる探索中に，郷土料理の「水軍鍋」の記事から広島の水軍として有名な「村上水軍」と言うキーワードを得て検索したとする．この場合，新しい探索単位は「広島 郷土料理」では無く上位の探索単位である「広島」を詳細化する話題の探索であると言え，そちらに接続しなければならない．そのため，発見した $Unit_i$ から上位要素へとさかのぼって適切な要素を探す必要がある．

ここで，共通キーワードが無いにも関わらず話題を詳細化する場合とはどんな場合かを考える．それは，新しいクエリによる探索領域が上位話題による探索領域の内部に収まっており，上位話題に関するクエリを追加しなくても判別できる場合であると言える．例えば,「牡蠣の土手鍋」というクエリに関して，牡蠣の土手鍋は広島の郷土料理であり，この場合,「広島 郷土料理 牡蠣の土手鍋」と書かなくとも「牡蠣の土手鍋」だけで十分と言える．

そこで，$Unit_{new}$ 内で出現ページ数の多い単語，つまり DF 値の高い単語を抽出し，単語集合 $HighDFWord_{new}$ とする．そして，先程発見した $Unit_i$ からルート要素に向かって，次の条件を満足するような探索単位を探す．

$$\forall a \in HighDFWord_{new} \text{ に対し } a \in Word_i$$

こうして発見された探索単位に対し，$Unit_{new}$ は十分条件になっていると考え，その子として接続する．条件を満足する探索単位が見つからなかった場合は，どの話題の十分条件にもなっていないとみなし，ルート要素に接続することにする．

今回は $Unit_{new}$ の中で 60%以上のページに出現している語の集合を $HighDFWord_{new}$ とした．

4.2.4 接続先競合の解決

上記規則だけでは，接続先となる要素が複数存在する可能性がある．そのような場合，先ず接続先候補として挙げられた unit 要素各々に対し，閲覧ページの tf·idf 特徴ベクトルと $Unit_{new}$ における閲覧ページの tf·idf 特徴ベクトルとのコサイン類似度が一定値以下のものを対象から除外する．次に，

> 対象となる unit 要素が複数存在する場合は最も新しく作成された要素を対象とする

という規則を適応する．これは，先述の「新たなクエリは探索の過程で発見・想起される」という仮定に因っている．

つまり，新しいクエリが 2 つ以上の話題を詳細化する可能性がある場合でも，ユーザの意図としては，より新しい探索に関係しているだろうと考えている．

4.3 接続先話題の決定

クエリの文字列としては異なっていても，同一話題を探索するクエリが存在する．例えば言い換え表現などである．本節では，前節までで作成した木に対し，同一話題を扱う要素をグルーピングする手法を説明する．対象は兄弟要素である．

話題語の抽出には共起度を利用する [9] ことが多いが，本研究にとっては前述のように計算時間のかかる手法を利用することは好ましくない．また，グルーピング手法としては，ページ内の単語の出現頻度を特徴ベクトルとしてベクトル間の類似度を利用する手法が多いが，言い換え表現による検索で得られたページ間では，同じ単語が出現することは少なく類似度も小さくなってしまう．例えば,「Q = 京都観光 銀閣寺」による検索結果のページには「銀閣寺」という語は多くても，言い換え表現の「慈照寺」という語は少ない．これは，2 つの表現を併記しているページが少ないためである．

しかし，併記してるページは少数ではあるが存在する．本手法では，その情報を用いて同一話題の判定を行う．兄弟関係にある unit 要素 $Unit_i$, $Unit_j$ とそのクエリ $Q_i = k_{i,1}\, k_{i,2} \cdots k_{i,n}$, $Q_j = k_{j,1}\, k_{j,2} \cdots k_{j,m}$ に対する判定は，次の様に行う．

（1） $k_{i,n} = k_{j,m}$ ならば同一話題とする

（2） $Q =$ “$k_{i,n}(k_{j,m})$” でフレーズ検索を行い，検索結果に “$k_{i,n}(k_{j,m})$” をというフレーズがあるかを調べる．フレーズが存在する場合，$Unit_i$ と $Unit_j$ は同一トピックとする

（3） $Q =$ “$k_{j,m}(k_{i,n})$” でフレーズ検索を行い，検索結果に “$k_{j,m}(k_{i,n})$” をというフレーズがあるかを調べる．フレーズが存在する場合，$Unit_i$ と $Unit_j$ は同一トピックとする

3.1 節で記したキーワード式の仮定より，$Unit_i$ における探索領域を最も代表するキーワードは $k_{i,n}$ であり，$Unit_j$ における探索領域を最も代表するキーワードは $k_{j,m}$ である．そのため，兄弟要素である $Unit_i$，$Unit_j$ の最終キーワードが同じである場合，近い領域を探索していると考えられる．そこで（1），では 2 つのクエリの最終キーワードを比較している．(2)，(3) では，$k_{i,n}$ と $k_{j,m}$ を併記しているページを検索し，2 つのキーワードが言い換え表現であるかを調べている．例えば，銀閣寺と慈照寺の場合，Q =“銀閣寺 (慈照寺)”で検索を行い，検索結果に「銀閣寺 (慈照寺)」というフレーズが含まれているか調べ，同様にして Q =“慈照寺 (銀閣寺)”を用いても調べている．

(2), (3) で用いた括弧を使った検索によって得られる語の関係は，言い換え表現だけでは無い．例えば,「関係記事一覧（五十音順)」という文がある場合,「関係記事一覧」と「五十音順」が言い替え表現であると判定されてしまう．しかし，同一話題探索の判定を行う対象は，兄弟要素であり，親子関係判定によって同一話題を詳細化していると判断された探索単位のみである．そのため，上記の様な不適切な語に対して判定を行う場合は少ないと考えられる．

5. 協調探索への利用

本章ではこれまで述べてきた QueryLogTree の協調探索への利用について述べる．協調探索においては，1 章で述べたように誰がどの分野の探索を行っているのか，まだ探索されていない分野はどこであるかを可視化することが重要である．

5.1 既に探索された領域の提示

同じ分野の探索を抑える方法としては，すべてのユーザが閲覧したページ URL を共有し，既に閲覧済みのページを他のユーザが閲覧できない様にすることが考えられる．1 つのページを閲覧できる人数を制限してしまえば，狭い範囲の探索になることは無いと言える．

しかし，あるユーザにとって価値の薄いページであっても他のユーザにとっては重要なページであることも多い．ユーザ間で興味や知識量に差がある場合はなおさらである．逆に，複数ユーザが閲覧し推薦できる内容だと判断したページは，協調探索を行っているグループにとって，より価値のあるページであるとも考えられる．

そこで，本研究では QueryLogTree を用いて，ユーザ毎の探索クエリをユーザに提示することで大まかに探索されている領域を知らせる方法を取っている．他のユーザが閲覧したページや推薦したページについての情報は提示するだけに留め，どのページに関しても閲覧を妨げることは行わない．

提示の方法は，自分の QueryLogTree に他ユーザのそれを結合し一つの木構造にして表示している．

5.2 未だ探索されていない領域の提示

5.2.1 協調探索における 2 つのフェーズと提示する領域

協調探索の状態には，拡大フェーズと縮小フェーズがあると考えられる．前者は，どういったものをグループとして求めているのかが定まっておらず，幅広い情報を収集し比較検討するために探索領域を拡大して行っている状態を指し，後者は，グループとして求める情報が明確となり，より精度の高い詳しい情報を求めていたり，同じ方向性の情報の中で比較検討を行う

ために探索領域を縮小し深く探索を行っている状態を指す．

旅行のプランニングを例にとると，グループで旅行に行くことだけは決まっているが，目的地等が決まっていない場合，まずは観光地やグルメ，レジャーといった多くの情報を収集し目的地等を決めることから始めるであろう．この状態は拡大フェーズと言える．目的地等が決まってくると，その中でよりグループに適した宿泊場所であったり行程であったり，狭い領域を詳細に探索し，自分達に適した情報を求めていくと考えられる．この状態は縮小フェーズと言える．現実的にはこの様にすんなりと物事が決まることは珍しく，一旦決まりかけていた話が，ちょっとしたきっかけで振り出しに戻るという事は多々起こりうる．即ち，フェーズは流動的である．

両フェーズでは，探索において利用する情報は異なっており，提示する領域もフェーズによって異なる．拡大フェーズでは，目標とする情報が定まっていないため，できるだけ広い範囲の探索が行えるように探索領域を提示する．この場合，探索に関連するが，まだ誰も探索していない領域を提示する方が良いと言え，その中で最も大きい領域を提示することにする．縮小フェーズでは，現在探索中の領域について，より詳細化した情報を提示するために，対象領域を細分化する領域の提示を行う．

5.2.2 未探索領域を表すキーワードの発見手法

未だ探索されていない領域を発見する方法として検索エンジンインデックスからの同位語発見技術 [7] および話題語発見技術 [10] を使用する．

検索エンジンインデックスからの同位語発見技術では，助詞の「や」を用いて同位語を検索する．キーワード k の同位語が X である時,「k や X」「X や k」というフレーズが多く文章中に現れることに注目し,「$Q =$ “k や”」,「$Q =$ “や k”」でフレーズ検索を行い検索結果ページのサマリ集合から X に当てはまる語を取り出している．また，話題語発見技術では，連体修飾助詞の「の」を用いて話題語を検索する．キーワード k による探索領域の，ある部分領域を代表する語が X であるとする時,「k の X」というフレーズが多く文章中に現れることに着目し,「$Q =$ “k の”」でフレーズ検索を行い，得られた検索結果ページのサマリ文章集合から X に当てはまる語を取り出している．さらに，これらの手法について，使用するクエリに $Q = topic$ を連言として加えることで $topic$ という主話題の下での同位語や話題語を得ることが出来る．

例えば，広島という主話題の下で郷土料理の同位語を検索する場合,「$Q =$ 広島 “郷土料理や”」,「$Q =$ 広島 “や郷土料理”」を使用し,「特産品」「温泉」「特産物」「地酒」などを得ることができ，京都観光という主題語の下で銀閣寺の話題語を検索する場合,「$Q =$ 京都観光 “銀閣寺の”」を使用し,「紅葉」「雪」「参道」などを発見することができる．

以上の性質より，QueryLogTree において，あるクエリ Q_i から同位語 X_{coord} を求めた場合，X_{coord} は Q_i の兄弟要素，即ち探索領域を広げるようなクエリとなる．一方，話題語 X_{topic} を求めた場合，X_{topic} は Q_i の子要素，即ち探索領域を詳細化するようなクエリとなる．しかし，Q_i の親要素に対して話題語 X'_{topic} を求めた場合，Q_i の兄弟要素 X'_{topic} を得ることが出来る．このため話題語発見を同位語の検索にも利用できるように思えるが，話題語発見では探索領域を細分化する話題語すべてを検索するのに対し，同位語発見では数ある話題語のうちユーザが使用したクエリに関連のある語のみを求めることができるため，より現在の探索に関係のある語を得ることが出来る．

よって，本提案システムでは，前述の拡大フェーズに対しては，探索領域を広げるような語の発見方法として同位語発見技術を用い，縮小フェーズに対しては，探索領域を詳細化するような語の発見方法として話題語発見技術を利用することにする．また，両フェーズは流動的であるため，フェーズ間の遷移が起こっても良いように常に両方のフェーズを支援することとし，拡大フェーズ用，縮小フェーズ用 2 種類のキーワードを提示する．以下に，両フェーズのための提示するキーワード発見アルゴリズムを記す．

5.2.3 拡大フェーズのための同位語発見

（1） 探索単位集合 **U** から $Unit_i$ を選び，そのクエリが $Q_i = k_1 k_2 \cdots k_n$ だとする

（2） 「$Q = k_{n-1}$ “k_nや”」と「$Q = k_{n-1}$ “や k_n”」より最も出現頻度の高い同位語 X_i を得る

（3） 1〜2 を繰り返し **U** の全要素について X_i とそのランク値 $Rank(X_i)$ を求る

（4） 最もランク値の高い同位語 X_{max} を $Unit_{max}$ の兄弟として接続する

まず，3.1 節で仮定した様にクエリは話題を詳細化する順に記述されているので，手順 2 に対し k_n を用いているのは，$Unit_i$ における探索領域を最も代表するキーワードが k_n と言えるからである．また k_{n-1} を連言として追加することで k_{n-1} という主題語の下での同位語を求めている．見つかった同位語の評価には 2 つの指標がある．1 つは，その同位語から得られる探索領域の大きさで，本研究ではその同位語の検索結果数の常用対数で近似している．もう 1 つは現在の探索目的との関連性で，同位語の検索に使用したクエリによる探索単位の閲覧ページ数で近似している．一般に，検索結果数は閲覧ページ数に比べ非常に大きな値を取る．そのため，両方の指標を同程度考慮に入れるために対数を取っている．また，閲覧ページ数で近似しているのは，探索目的に合った検索であったのならば検索結果からリンクをたくさん辿り，期待していた結果が得られないような検索であったのならば，少しのページを閲覧しただけで次の検索に移るだろうという仮定に依っている．検索結果一覧には各ページのサマリが含まれており，閲覧ページにはこのサマリも含めるため，リンクを辿って閲覧したページ数を $View_i$ とすると $View_i + 1$ を閲覧ページとして用いる．以上よりランク値として次の値を用いる．

$$Rank(X_i) = \log_{10}(n_i) \times (View_i + 1)$$

5.2.4 縮小フェーズのための話題語発見

（1） QueryLogTree から最も新しいクエリ $Q_{new} = k_1 k_2 \cdots k_n$ を選ぶ

（2） 「$Q = k_{n-1}$ “k_nの”」を用いて出現頻度の大きい話題語 $X_{new,j}$ を得る

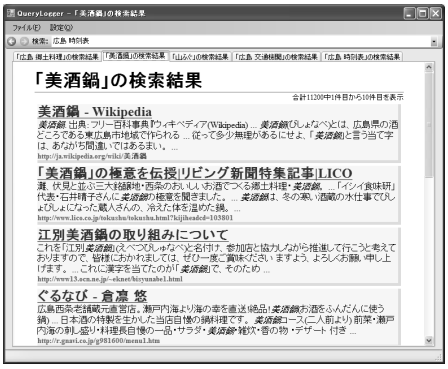

図 9　探索画面

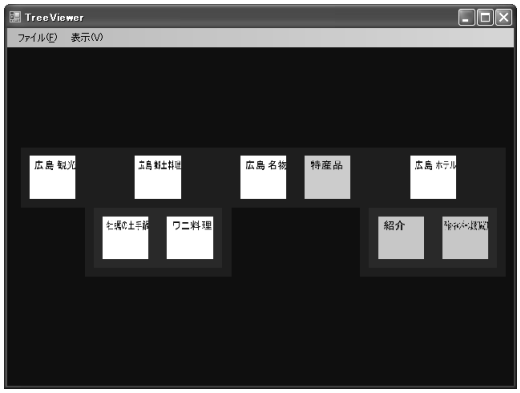

図 10　QueryLogTree 表示画面

（3）　$X_{new,j}$ を QueryLogTree に接続する

手順 2 で k_n および k_{n-1} を使用しているのは同位語発見の時と同じ理由である．詳細化クエリについては，部分領域へ分割するために複数の語を取り出し接続する．

5.2.5　発見したキーワードの提示

発見した両フェーズ用のキーワードは，既に探索された領域の提示方法と同様に全ユーザの QueryLogTree を統合した木に追加し提示する．ただし，実際に探索を行うまではどの topic 要素に属する語であるか判定できないため独立に扱う．

6.　システムイメージと実験

6.1　システムの概観

本論文で提案した手法を用いた探索システムのプロトタイプについて説明する．図 9，図 10 にシステムの実行画面を示す．

図 9 は Web ページの表示画面である．通常のブラウザと異なり，ユーザは直接 URL を入力することはできず，クエリを入力し検索を行うか，ページのリンクを辿ることで探索を進めていく．図 10 は QueryLogTree を表示させた画面である．「Q = 広島 ホテル」が最も新しい探索単位で，その子要素として「紹介」「予約キャンペーン」といった話題語が別の色で追加されている．また，「Q = 広島 名物」の同位語として「特産品」が追加されている．

6.2　システムの動作

ユーザは，1 章で設定した様に次の操作を行い探索を進めることができる．

- クエリを入力し検索を行う

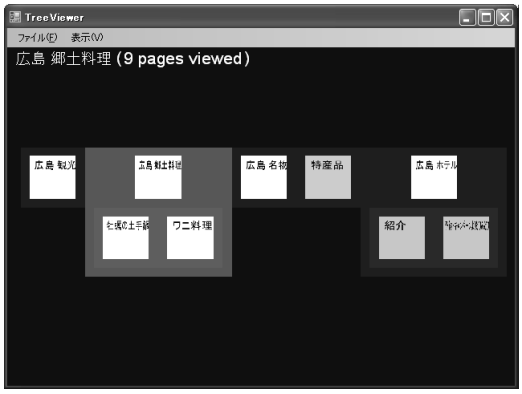

図 11　QueryLogTree 表示画面 (付加情報の表示)

- 検索結果一覧からリンクを辿ってページを閲覧する
- 気に入ったページがあればマークを付ける

システムは，ユーザが新たなクエリを入力し検索を行うと，その前のクエリによる探索が終了したと判断し，その探索単位を 4 章にて説明した手順により QueryLogTree に追加する．ユーザが QueryLogTree の表示を求めた場合，次の手順により表示用 QueryLogTree を作成する．

（1）　最新の QueryLogTree のコピーを作成する
（2）　未追加の現在実行中の探索単位を追加する
（3）　他のメンバーの QueryLogTree を取得し追加する
（4）　未探索領域を表す語を検索し追加する

最新の QueryLogTree は現在の探索単位を含んでいないため，(2) において追加している．(4) では 5 章にて説明した手法を用いて未探索領域を表すクエリを検索し追加している．

6.3　表示する QueryLogTree

メニューから「探索情報の表示」を選択すると QueryLogTree を表示する．図 10 の例では，「広島 郷土料理」の子要素として「牡蠣の土手鍋」「ワニ料理」が配置されている，子要素の場合は図 11 に示すように親要素にマウスポインタを合わせるとポップアップされる．また，マウスポインタを合わせている要素については何ページ閲覧したかという追加情報が表示されるようになっている．

6.4　QueryLogTree の作成実験と考察

提案手法を検証するために．QueryLogTree の作成実験を行った．

6.4.1　親子関係の判定

話題を詳細化しているクエリの抽出が正しく行われるかの実験を行った．使用したクエリログは次の通りである．

（1）　広島 郷土料理
（2）　牡蠣の土手鍋
（3）　ワニ料理
（4）　広島 観光
（5）　広島 旅館

ただし，「牡蠣の土手鍋」や「ワニ料理」といったキーワードは，「広島 郷土料理」に関する探索を行い，閲覧ページしたページから発見されたものである．

結果は，図 12 に示すように，「ワニ料理」や「牡蠣の土手鍋」

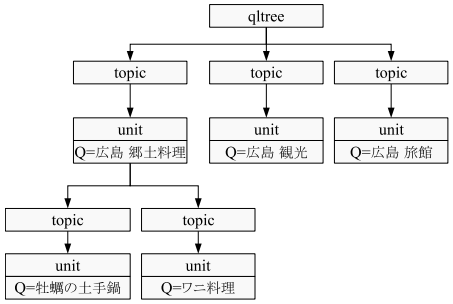

図 12　親子関係の判定実験結果

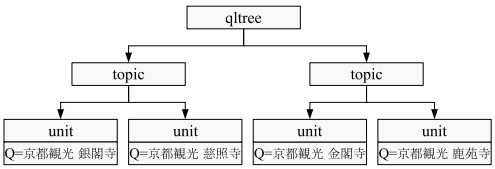

図 13　同一トピックの判定実験結果

探索内容	クエリ数	親子関係判定 正解率	同一話題判定 正解率
技術情報の探索 1	16	87.5%	87.5%
ショッピング情報の探索	22	90.9%	95.4%
ある話題に関する情報収集	9	77.7%	100%
技術情報の探索 2	27	85.1%	96.3%

図 14　QueryLogTree 作成実験の結果

といった広島の郷土料理に関するクエリが「広島 郷土料理」の子に接続され，期待通りの結果が得られた．

現時点での問題点は，例えば「広島の観光」について調べたい場合に「広島」だけをクエリにする場合である．その後,「広島 郷土料理」と言うクエリを使った場合，本提案手法では「観光」について探索したことが残らない．探索意図を共有するためには，このように大雑把なクエリからの探索意図抽出に関しても考える必要があると言え，今後の課題である．

6.4.2　同一話題の判定

同一話題の判定実験は，次のクエリログを用いて行った．

（1）　京都観光 金閣寺
（2）　京都観光 鹿苑寺
（3）　京都観光 銀閣寺
（4）　京都観光 慈照寺

結果は，図 13 に示すように，正しく判定することができた．

6.4.3　実際の探索における QueryLogTree 作成実験

Google 検索履歴 [3] から過去に行った実際の探索のクエリと閲覧ページ等の情報を得て提案システムで同じ探索を行い，QueryLogTree を作成した．

探索内容と実験結果は図 14 に示す様であった．探索内容について，技術情報の探索 1 およびショッピング情報の探索はどういったものを探索すれば良いかを予め設定せずに行ったもので，ある話題に関する情報収集および技術情報の探索 2 は設定した話題に関して関連のある情報を網羅的に集めることを目標として行ったものである．どちらのタイプの探索であっても比較的正しく判定できていると言える．

7.　まとめと今後の課題

本研究では，協調探索において，探索参加ユーザが似たようなクエリを利用することが多く探索領域が狭くなってしまう問題を支援するために，クエリログを木構造に変換し QueryLogTree として提示することで各ユーザの探索領域を俯瞰できる様なシステムの設計と実装を行った．QueryLogTree の作成には，クエリに含まれるキーワードの関係と閲覧したページの特徴を利用し，本来の探索の妨げにならない様に複雑な手順を使わずに効果的な結果が得られることを目標とした．また，同位語発見技術や話題語発見技術を利用し未だ探索していないが有益と思える探索領域の発見・提示も行っている．

現在はユーザ間で探索話題が重ならない為の支援を行っているが Web グラフとしても重ならない為の支援も必要であると言える．なぜなら，異なる話題の探索を行っていても，同じページへ辿り着きやすい場合がある．このような場合にも結果的に探索領域は狭くなってしまう．そのため，Web グラフとしての探索領域に関する支援が今後の課題である．

謝辞　本研究の一部は，文部科学省 21 世紀 COE 拠点形成プログラム「知識社会基盤構築のための情報学拠 点形成」（リーダー：田中克己，平成 14〜18 年度）および，文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基 盤技術の研究」，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）の援助を受けており，また，文部科学省科学研究費補助金萌芽研究「AND-OR グラフを用いるデータモデルとその操作系，制約記述系に関する研究」(研究代表者：田島敬史，課題番号：18650021）によっております．ここに記して謝意を表すものとします．

文　献

[1] Yahoo!検索: http://search.yahoo.co.jp/
[2] MeCab: http://mecab.sourceforge.jp/
[3] Google 検索履歴: http://www.google.com/searchhistory/
[4] 大澤幸生，N.E. Benson，谷内田正彦, ”KeyGraph：単語共起グラフの分割統合によるキーワード抽出，”電子情報通信学会論文誌, J82-D1，No.2, pp.391-400,1999.
[5] 上田正明，中島伸介，角谷和俊，田中克己, ”探索アクティビティの共有と視覚化に基づく協調型情報探索, ”第 13 回データ工学ワークショップ (DEWS2002) 論文集, A2-5, 2002 年 3 月.
[6] 松下光範, 白井良成, 桑原和意, ”意思決定のための探索過程の振りかえり支援—探索履歴の空間配置—, ”人工知能学会全国大会論文集, Vol. JSAI03, pp.99-102, 2003 年.
[7] 大島裕明，小山聡，田中克己, ”サーチエンジンのインデックスを利用した同位語検索と同位語コンテキストの発見, ”情報処理学会研究報告，Vol.2006，No.77，2006-DBS-140(II)，pp.161-168, 2006 年 7 月.
[8] 松生泰典, 是津耕司, 小山聡, 田中克己, ”検索結果の概要を表すキーワード式生成による質問修正支援, ”電子情報通信学会データ工学ワークショップ (DEWS2005)，1C-i9，2005 年 2 月 28 日.
[9] 松尾豊，石塚満, ”語の共起の統計情報に基づく文書からのキーワード抽出アルゴリズム, ”人工知能学会論文誌, Vol.17 No.3, pp.217-223, 2002 年.
[10] 野田武史，大島裕明，手塚太郎，小山聡，田中克己，”Web 検索結果のクラスタリングに用いる話題語の質問キーワードからの自動抽出，”電子情報通信学会データ工学ワークショップ (DEWS2006)，2C-i8，2006 年