複数文書閲覧時の文書間の意味的関係の抽出と提示による文書ナビゲーション

内藤　稔† 大島　裕明†† 高橋亜希子†† 田中　克己††

† 京都大学工学部情報学科　〒 606–8501 京都府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 京都府京都市左京区吉田本町
E-mail: †{naito,ohshima,akiko.t,tanaka}@dl.kyoto-u.ac.jp

あらまし　本稿では，複数の文書を閲覧している際の，文書間の意味的な関係を抽出し，これを視覚的に提示する手法を提案する．現在の多くの Web ブラウザはタブブラウザと呼ばれるもので，タブを切りかえることで多くのページを閲覧することができるようになっている．しかし，同時に閲覧しているページが増えると，関連するページがどこにあるのか，なぜそのページを開いていたのかといった，ページ間に存在する意味や関連が分からなくなってしまう．これは，同時に開いている文書の関係を示す情報が不十分であることに起因すると考えられる．このような問題を解決するため，本稿では複数文書を同時に閲覧する際の，文書間の意味的な関係を抽出・提示するシステムを提案する．これにより，Web ページやオフィス文書の同時閲覧の意味を把握しやすくなり，過去のタスクの想起も容易になる．
キーワード　付箋インタフェース, 文書ナビゲーション, 関係抽出

1. はじめに

近年，コンピュータやネットワークの発展によって数多くの文書から多くの情報を得ることができるようになった．その反面，得られる情報量や情報源が肥大化し，タスクを行う上で閲覧する文書の数が増え，また同時に複数のタスクを持つことも多くなった．そのため，情報源となる文書の管理が難しくなってしまう事がしばしばある．

Web 情報に限定しても，例えばデジタルカメラの商品情報の比較をしている際に，商品ページばかりではなくメーカーの製品情報ページや，ユーザによるレビューページを同時に閲覧することがある．この時，現在の主流となっているタブブラウザでは，開いているタブが多くなりすぎてしまう．それにより，どのタブにどの商品の情報が記述されているのかが分からなくなってしまう問題がある．

また，タスク状況を保つために，たくさんのタブを開いた状態ででセッションを保存していたのだが，いざ復元した時には，過去の自分がどのようなタスクを行っていたかを忘れてしまっていることもある．そのため，各ページの関連や，そのページを開いていた意味が全く分からなくなっており，結局復元したセッションから情報を得るためにすべてのタブを見直す必要がでてくる問題がある．

このような問題が起こってしまう大きな要因は，現在のブラウザでは同時に開いているページでもこれを単体で扱い，その関連に着目した機能がないためである．また，現在のブラウザではローカルドキュメントとの連携が難しく，特定のドキュメントファイルと同時に開いていた Web ページ，といった検索が難しくなっている問題もある．

先程のデジタルカメラの例で言うと，デジタルカメラの詳細を Web ページから調べながら，その内容を Excel ファイルにまとめて比較したりしていた場合を考える．この時，後からその Excel ファイルを開いた時に，同時に開いていた Web ページ，即ち調べていたデジタルカメラの情報ページを復元したいというニーズがあると考える．

これらの問題を解決するべく，我々は閲覧した文書間にある意味的な関係を抽出・提示することで，各文書間の関連や意味をユーザに想起させる手法を提案する．

本手法によって解決すべき問題は，以下のような問題が挙げられる．

- 同時に閲覧している文書が多過ぎて欲しい情報を有している文書がどれか分からない．
- 文書間の関係が分からないために，各文書を開いていた意味が分からない．
- 特定の文書と同時に開いていた文書がどこにあるのか分からない．

これらの問題は，各ページを単一に扱う現状のタブブラウザ環境やブックマーク機能だけでは解決が難しい．しかし，文書間の意味的な関係・関連を提示することで可能となる擬似的なタスクの俯瞰によって，ユーザを億劫なタブ間の移動や膨大な履歴情報の参照といったストレスから開放できる．

ここで，文書間の意味的関係があるとは，比較関係・詳細関係・補完関係などといった文書間に意味のある関連があること

である．また，そのような静的な関連だけではなく，同じ検索クエリによって辿りついた文書や，同時に開いていた文書といった，ユーザによる動的な関連も考えられる．

このような文書間の関係を抽出するために，本研究では以下の 4 つの関連を用いた．これらについて，以下に定義を与える．

構造的関連

構造的関連とは，文書そのものが持つ構造的な関連であり，ハイパーリンクによる関連や，URL のドメインによる関連などがこれに当たる．

内容的関連

内容的関連とは，文書同士の内容的な関連であり，文書の情報を用いた，文書間の類似度などによる関連がこれに当たる．

共通ノード関連

共通ノード関連とは，個々のユーザにとっての文書間の関連であり，同じクエリや同じリンク元など，ユーザの行動における共通の要素を持った関連である．

共使用関連

共使用関連とは，ユーザが同時に使用していたということによる文書間の関連である．

これらの関連は大きく静的な関連とユーザによる動的な関連の 2 つに分けられる．構造的関連と内容的関連は，共に文書間の静的な関連である．よって文書が同じであれば，ユーザが変わっても同じ関連が得られる．一方で共通ノード関連，及び共使用関連は文書の内容ではなく，ユーザの行動に着目した関連である．このため，文書の情報からだけでは関連が得ることができない．また，得られる関連も個々のユーザ毎に異なるものである．

これらの特徴を活かして，ユーザに対する文書ナビゲートに有用である関連を抽出し，それぞれを独立して提示する事で，ユーザに負担の少ない文書ナビゲーションを行うのが本手法の目的である．

本稿の構成は以下の通りである．2 節では関連研究について述べ，本研究が取りくむ問題について明らかにする．3 節では文書間にあると考えられる関連の種類とその抽出アルゴリズムについて述べ，4 節では 3 節によって抽出した関連を可視化する手法についてを述べる．5 節ではその実装方法について詳しく述べ，6 節では実装したシステムの実際の運用例を挙げる．7 節では実際にユーザにシステムを使用してもらい，観察できた事象について述べる．最後に 8 節では結論として，この本研究による成果をまとめる．

2. 関 連 研 究

2.1 文書間の静的な意味的関係

佐野ら [1] は，Web ページ内の DOM 内に付箋を埋め込むことでクロスブラウザで動作する付箋アノテーション付与するシステムを考案した．このシステムでは各付箋が貼られた文書同士の類似度を自動で求め，グループ分けすることで，付箋同士が相互に関係を持つ機能が用意されている．

また，豊田ら [2] は，アーカイブのリンク関係からページの相関図を表すグラフを時系列毎に抽出して，発展過程を動的に見るシステムの考案をしている．この研究は初期シードページのリンク関係から関係するページを自動的に補完していくものである．

このように，本手法でいうところの文書間にある静的な意味的関係を抽出している研究はあるが，本手法ではこれに加え，ユーザ意図を考慮した動的な意味的関係も同時に抽出・提示する．これにより，ユーザにタスクを想起させ，より直感的な文書閲覧の支援を目指している．

2.2 ユーザ意図による関係

日野ら [3] は，アプリケーションを超えた知識として small knowledge を定義し，その理想的な蓄積および利用方法を提案している．

暦本 [4] は，デスクトップ環境を時間軸に記録していき，作業当時の環境を回復できるシステムを考案している．このシステムでは各アプリケーションが保持する時間情報を連携させて，クロスアプリケーションでの共使用関係が実装されている．

定免ら [5] は，ファイルの閲覧遷移記録や共起時間を用いて共使用関係を抽出するシステムを考案した．なお，このシステムはローカルドキュメントでの使用を想定されている．

大澤ら [6] は，ユーザの操作ログや履歴情報を用いて関連の高いオブジェクトを検索する “俺デスク” システムを考案している．このシステムでは共使用関係を用いて関連の深いファイルの検索を行っている．

渡辺ら [7] は，ファイルのテキスト類似性や作成時間，共起時間からファイルの関係性を抽出し，多次元でクラスタリングを行い，ファイル検索を行う “FileSearchCube” システムを考案した．これは多角的な関係性からファイルの関係を抽出する手法である．

このように，ユーザの意図を考慮したファイル間の関係の取得のために共使用関係を用いた研究は数多くある．本手法では Web ページとローカルドキュメントの連携，及び文書間の共使用関係を主眼に置いた共使用関係の抽出を目指す．また，同時に Web ページの閲覧におけるユーザ行動にも着目し，共使用関係と独立に抽出することで，共使用関係だけでは捉えられない文書間の意味的関係の抽出も目指す．

3. 文書間の意味的関係

本節では，文書間にある意味的関係について，本研究における定義と，そのために抽出する文書間の関連の種類，及びその抽出法について述べる．

3.1 ユーザが扱う文書

ここでは，想定される状況でユーザが扱う文書について述べる．

ユーザは Web 上のページを閲覧するが，この時通常の html ページのほかに，pdf ファイルや Word ファイルやプレーンテキストを読み込む可能性がある．また，ローカルのドキュメントとして pdf ファイルや Word ファイル等の Office 文書を Web と連携して運用する事も十分考えられる．これらの文書を等しく扱うことができればアプリケーションに依存しない文書間の意味的関係を取る事ができる．

3.2 文書間の意味的関係

文書間に意味的関係があるとは，意味のある関連が文書間にあることを言う．

例えば，デジタルカメラの商品についてのページを比較している場合，これらのページが全く違う通販サイトのページ同士であったり，辿ったページの経緯が全く違ったりしても，これらのページは互いに比較対象であるという意味的な関係を持っている．他にも詳細関係，補完関係など，文書間にある意味的な関係が考えられる．

これらの意味的関連を持つ文書は，共通するトピックを持つと考えられる．先に挙げた例では，比較しているページ群はデジタルカメラの商品ページであるという共通点を持っている．これが意味的関係を捉えるトピックである．

このような意味的関係は文書内容に依存するもので，ユーザの行動に関係なく，同じ文書間からは一意に関係が抽出される．

このような静的な意味的関係に対し，ユーザの意図による動的な意味的関係も考えられる．論文の執筆中に調べていた文献は個々の内容に意味的関係がなくても，ユーザの意図として同じ論文に対して参照されていた文献という意味で関係がある．このユーザ意図による意味的関係を持つ文書は，共通のタスクを持っていると考えられる．

このように，文書同士が同一のトピックを持っていたり，同一のタスクに属している場合に，意味的な関係があるとする．また，意味的関係の強さは，共有するトピックやタスクの数に依存するものとする．

このような文書間の意味的関係を抽出することで，現在閲覧している文書と関係性のある文書へのナビゲーションをユーザに対して提供できる．これにより，開きすぎた文書から必要な文書を探したり，文書を開いていた意味を類推したりといった，現在の環境では難しい問題を解決することができる．

これらの文書間の意味的関係は，文書間にある様々な関連によって抽出できる．例えばハイパーリンクによる文書間の構造的な関連は，ハイパーリンクの性質上，共通するトピックを持つ文書同士にある関連だと考えられる．そのため，ハイパーリンクによって関連付けられた文書の間には意味的な関係があると考えられる．以下では，このように文書間の意味的関係を捉えるために有用であると考える文書間の関連について述べる．

3.3 文書間の関連

意味的関係を捉えるために有用なものとして以下の 4 つの関連を考える．

- 構造的関連
- 内容的関連
- 共通ノード関連
- 共使用関連

このうち構造的関連，内容的関連は，文書間の静的な関係を得るために有用であり，共通ノード関連，共使用関連はユーザ意図による動的な関係を得るために有用であると考える．以下に各関連の定義と抽出法について述べる．

3.3.1 構造的関連

構造的関連とは，文書そのものが持つ静的な関連であり，文書自体の構造的な関係を示すものである．

文書を構造的に配置する場合，その配置には意味がある事が多い．そのため，文書間の意味的関係を得るためにこの関連を抽出する意義は十分にあると考える．

構造的関連として，以下のようなものが考えられる．

直接ハイパーリンク関連

リンクを持っているページはリンク先であるページと共通するトピックを持っている事が多い．これはその関連を用いた関連で，文書 A が文書 B へのリンクを持っていた場合に A → B の直接ハイパーリンク関連があるとする．

この関連の強さはページによらず一定だが，B → A の関連もある場合は，双方向の関連があるということで強い関連となる．

なお，この関連の方向性は関連の抽出の段階では双方向に関連があるか否かにしか興味が無いため，特に意味はない．

間接ハイパーリンク関連

同じリンクを持っているページ同士は，リンク先のページを通して共通するトピックを持っている事が多い．これはその関連を用いた関連で，文書 A と文書 B が共通のリンクを持っていた場合に AB の間に間接ハイパーリンク関連があるとする．

この関連の強さは，持っている共通リンクの強さに依存する．共通リンクの数が多ければ多いほど，強い関連となる．

また，リンク先だけではなく，共通のリンク元を持っていればその数も加算する手法も考えられるが，今回は見送っている．

ページ内で二つのトピックを扱っており，それぞれの詳細として挙げられたページ同士は共通の元リンクを持ってはいるが，それぞれ別のトピックに属している．このように，共通のリンク先ほど関連性が信頼できないうえ，後述する共通遷移リンク関連によって，これに近い関連を抽出しているため，抽出に用いていない．

ドメイン関連

この関連は複数文書間の関連を取るもので，各文書の URL を比較して，包含関係にあった場合に関連があるとして，関連文書をツリー構造として抽出する．

この関連の強さは一定である．

これらのうち，特にハイパーリンクによる関連は，その特性から共通するトピックを持つ文書との関連を表現しやすく，文書間の意味的関係の抽出に有用であると考える．

また，この構造的関連を抽出することで，ユーザに関連性の高い文書同士を報せることができるため，開いている文書が多くなっても，ユーザが近い内容の文書を探す際の負担を軽減させる事ができると考える．

3.3.2 内容的関連

内容的関連とは，文書そのものが持つ静的な関連であり，文書自体の内容的な関係を示すものである．

文書の内容的な関連は，そのまま意味的な関係へと直結するため，この関連を抽出する意義は十分にあると考える．

内容的関連として，以下のような関連が考えられる．

内容類似度関連

この関連は 2 文書間の関連を取るもので，文書 A と文書 B のコサイン類似度を取り，その値が閾値以上なら内容類似度関連

があるとする．

実際のコサイン類似度の計算法は後述するが，それによって得られたスコアがこの関連の強さとなる．

文書の特徴ベクトルが似ていれば，文書同士が共通するトピックを持っている可能性が高いため，この関連を抽出は有用であると考える．

また，この関連を取る事で，リンク関係とは関係なく内容的に近い文書同士を報せることができるため，ユーザへの文書ナビゲーションを可能にする．

その他の内容的関連

その他，文章の内容的な関係として，対立する文書や，比較文書，詳細文書などが考えられる．

これらの関係は抽出できると，ユーザに各文書の内容を想像させるのに十分な関係で非常に興味深い関係なのだが，抽出が難しい上に，関係が抽出できたところで，本手法で目指す可視化では表現が難しいため，これらの関係の抽出はしない．

3.3.3 共通ノード関連

共通ノード関連とは，ユーザにとっての文書間の意味的な関連であり，文書の内容ではなく，リンクの遷移や検索クエリといったユーザの行動に依存した共通のノードを持つ，個々のユーザにとっての意図による関連である．

これらのユーザの行動は，ユーザの行っているタスクを表現している事が多い．そのため，文書間の動的な意味的関係を得ためにこの関連を抽出する意義はあると考える．

共通ノード関連として，以下のような関連が考えられる．

共通遷移リンク関連

この関連は元リンクとなる文書を中心として，そこから遷移した文書群のグルーピングとして表現される関連である．

ある文書 A の元リンクが文書 B であった場合，文書 A は文書 B の子ノードとなる．ただし，この親子関係は文書 B を中心とした遷移関係のグループの中での関係であり，文書 B から文書 A にも遷移していた場合，文書 A を中心とした遷移関係の中では，文書 B が文書 A の子ノードになる．

各関係の強さは一律であるが，両方向に遷移関係があった場合は片方向の遷移関係よりも強くなる．

この関連は，3.2.1 節で述べた直接ハイパーリンク関連と共通する部分が多いが，ハイパーリンク関連が文書間にある静的な関係であるのに対し，こちらはユーザが実際にページを辿ったという動的な関係であり，そのユーザにとってより意味のある関連であると言える．逆に，ハイパーリンク関係があり，しかし遷移リンク関連がない場合は，クリックスルーがなされたという解釈もできる．

共通クエリ関連

この関連は文書に辿り着いた際の検索クエリを中心として，その検索クエリによって辿られた文書群のグルーピングとして表現される関連である．

ある検索クエリ A を用いて検索を行い，そこから辿り着いた文書 B は，クエリ A の子ノードとなる．また，各関係の強さは，検索結果ページから遷移したページの数に依存する．例えば検索結果から文書 B →文書 C →文書 D と遷移した場合，文書 D とクエリ A の間の関係の強さは，文書 B や文書 C とクエリ A との間の関係の強さよりも小さくなる．これは，検索結果ページからリンクを辿れば辿るほど，元の検索クエリから関係のないページに遷移する可能性が高くなるためである．

共通タグ関連

この関連は文書にユーザがタグ付けをすることで，ユーザが意図的に文書群のグルーピングを行うための関連である．

この関連は自動抽出ではなく，ユーザが意図して文書群をグルーピングをするという，機能的な関連であるが，このような機能も必要であると考えたため，関連の一つとして挙げている．

このような共通ノード関連を抽出することで，ユーザに過去に自分が辿ったタスクを想起させることができるため，時間が経って意味の分かり辛くなった文書群に対しても，閲覧しやすくなる．

3.3.4 共使用関連

共使用関連とは，文書の共使用関係に着目した関連であり，共通ノード関連と同じくユーザの行動に着目したものである．この関連は，共通ノード関連よりもユーザ側に踏み込んだタスク依存的な関連であり，タスクの俯瞰には重要な関連であると考える．そのため，共通するタスクによって表現される文書間の動的な意味的関係の抽出にも有用な関連であると考える．なお，ここでいう共使用関係とは，ユーザが同時に開いていた文書間にある関係のことである．

共使用関係の抽出するために有用だと思われるものとして，共起回数，共起間隔，共起時間，ユーザの操作ログなどが挙げられる．これらを使った抽出のアルゴリズムは次項で詳述する．

また，共使用関連を抽出することで，ユーザはアプリケーションの壁をこえた文書間の関連を知ることができ，また過去の作業環境を復元するなど，過去のタスクの想起にも十分役立つ情報が得られる．

3.4 各関連の抽出アルゴリズム

ここでは，3.2 節で述べた各関連について，どのようなアルゴリズムで抽出できるかを詳述する．

3.4.1 直接/間接ハイパーリンク関連

各文書の保有するハイパーリンクは，文書の解析を行う事によって容易に取得できる．

html 文書であった場合は，$< a >$ タグを抽出し，その中のリンク情報を取得する．それ以外の文書であった場合,「http://」や「ftp://」などの文字列による抽出を行うことでリンク情報の取得が可能である．

リンク情報の取得ができれば，直接ハイパーリンクであれば，リンク情報と対象文書のパスを，間接ハイパーリンクであれば，それぞれの文書のリンク情報を照合して，関連の有無を調べる．

3.4.2 ドメイン関連

各文書のパス（URL）を取得し，その包含関係を捉える．アルゴリズムとしては，パスを「/」文字によって分割し，パスの共通部分から包含関係を捉える．

たとえば，文書 A のパスが「http://kakaku.com/pc/note-pc/」，文書 B のパスが「http://kakaku.com/pc/」として取得できた場合，文書 A は文書 B の子ノードとなり，文書 C の

パスが「http://kakaku.com/camera/」だった場合，B と C は兄弟ノードとなる．

3.4.3 内容類似度関連

文書 A と文書 B に形態素解析を行い，取得した特徴ベクトルに TF-IDF で重み付けしたベクトルを用いてスコアを算出する．

各文書の特徴ベクトルをそれぞれ，

$$\boldsymbol{v}_A = tf_A \cdot idf \tag{1}$$

$$\boldsymbol{v}_B = tf_B \cdot idf \tag{2}$$

とすると，文書 A と文書 B の類似度は，コサイン類似度を使って，

$$cos_{AB} = \frac{\boldsymbol{v}_A \cdot \boldsymbol{v}_B}{|\boldsymbol{v}_A||\boldsymbol{v}_B|} \tag{3}$$

として計算できる．

この cos_{AB} が設定した閾値を越えた場合，関連があると判定し，その強さは cos_{AB} に依存する．

3.4.4 共通ノード関連

各文書に，遷移元リンク，検索クエリやタグといった情報を保持させ，それを照合することで実現する．

遷移元リンク・検索クエリ・タグといった情報をグループノードと表現する．共通ノード関連を抽出するには，まず関係を取る文書全てから，各文書が保持するグループノードを取得する．取得したグループノードを親ノードとし，これに属する文書を子ノードとしてグルーピングを行っていく．なお，1 つの文書が複数の親ノードに属しても構わないものとする．

こうして得られた親子ノードによるグループが，共通ノード関連を表現するものである．

3.4.5 共使用関連

2 節で述べた通り，共使用関係を抽出するアルゴリズムは既にいくつか考案されており，共起回数，時間，感覚，操作ログを用いたアルゴリズムのいくつかは関連研究として挙げた文献で紹介されているため，ここでの説明は省く．

ここでは本手法のために考案した，共起回数，共起間隔を文書の閲覧履歴から抽出し，共使用関係を抽出するアルゴリズムを記す．

関連の抽出に際して，基本的な考え方は履歴内での共起間隔が近ければ関係が強く，また共起回数が多くても関係が強いだろうというものである．

まず，各文書は，自分以外の文書との共使用に関するスコアを持っており，文書 A の文書 B に対するスコアを $Co_{A->B}$ とする．このスコアを，参照履歴が更新された際に更新し，各文書間の共使用関連のスコアとする．

参照履歴が更新された場合，まず新たに開かれた文書のスコア $Co_{0->i}$ を更新する．なお，ここで i は $0 < i \leqq N$ であり，文書 i とは履歴の文書の最近のものから N 件が割り当てられる．$Co_{0->i}$ のスコアは，

$$Co_{0->i} = Co_{0->i} + K(N - i) \tag{4}$$

の式によって加算されていく．ここで K は $K > 0$ のパラメータである．なお，この計算は共起間隔をスコアとして与えているものであるため，既に閉じられた文書や閲覧されていた時間が極端に短い文書のスコアは加算するべきではないため，加算を行う条件を設定する必要がある．即ち，

$$Open?(D_i) \ \texttt{\&\&} \ Time(D_i) > T \tag{5}$$

この条件式が真となる場合のみ，文書 i との共使用スコアを加算する．なお，$Open?(Di)$ は文書 i が現在開かれているかどうか，$Time(Di)$ は文書 i が閲覧されていた時間を表現している．また，T は $T > 0$ なるパラメータである．

このように文書 0 の各文書に対する共使用スコアを計算した後，今度は履歴の新しいものから N 件の文書 i の，文書 0 に対する共使用スコアの加算を行う．ただし，文書 i に対して，(6) 式が偽である場合はスコアの更新は行わない．なお，更新するスコアは，

$$Co_{i->0} = Co_{i->0} + K(N - i) \tag{6}$$

と表現できる．

このようにスコアを加算していくことで，文書 A と文書 B の間の共使用スコアは

$$Co_{A->B} = \sum K(Len(A, B)) \tag{7}$$

として求められる．ここで $Len(A, B)$ は文書 A と文書 B の閲覧履歴上での間隔を表している．式の通り，これは共起回数及び共起間隔に依存したスコアである事が分かる．

以上のように各関連を定義付け，抽出するアルゴリズムを挙げたが，次節ではこれらのアルゴリズムを元に抽出した関連を可視化する手法について詳述する．

4. 関係の可視化

本節では前節までによって抽出した関係を可視化する目的，手法，実装について述べる．

4.1 可視化の目的

得られた関係を可視化する目的は，文書間の意味的な関係をユーザに提示する事で，タブやタスクバーに並ぶばかりで分からなくなりがちな文書同士の関係や，行っていたタスク内容を少しでも分かりやすく俯瞰できるようにすることである．

ここで，本手法で定義している 4 つの関連は，それぞれに意味と興味深さを持ってはいるが，それらを独立に可視化するだけでは十分ではないと考える．これらの関連はそれぞれ，文書を特定の側面から見た際の関係でしかないため，これらをマージした総合的な関係を可視化しなければならない．そうすることで，初めて本手法の目的である効果的な文書ナビゲーションが達成できると考える．

このマージには，各関連の優先度を決めた上で各関連のスコアをうまくとっていくことで総合的な関係を取得するといった手法も考えられる．しかし本手法ではそうではなく，各関連をそれぞれ独立且つ同時に可視化する手法を用いる事にした．各

関連を同時に可視化することで，例えば構造的関連で拾えなかった関係も共通ノード関連によって関係が結ばれる等，提示する結果上で自動的にマージされる．これにより，ユーザが文書間の総合的な関係を容易に理解できるため，この手法でも目的は十分に達成できると考える．

4.2 各関係の出力

結果の出力について，本手法ではグラフでの可視化を選んだ．各文書をノードとし，その間にある関連をエッジで表現することで，文書間の関係の有無や強さを直感的かつ容易にユーザに提示できるためである．

ここで，各関係をグラフで表現するにあたり，可視化の手法として，

- 2 ノード間にエッジを張る（エッジ中心）
- 元ノードから派生ノードにエッジを張る（ノード中心）

が考えられる．

本システムの場合，構造的関連と内容的関連は文書間の静的な関係であるため，ノード同士のエッジを張る方法で表現でき，共通ノード関連と共使用関連は中心となる要素や文書を中心にグルーピングを行う関係であるため，後者の派生ノードにエッジを張る方法で表現できる．

なお，間接ハイパーリンク関連，内容類似度関連，クエリ関連，共使用関連に関してはスコアを持つため，エッジにその値を渡す事で，エッジの長さで関係の強弱を表現することができる．この手法については実装の項で詳しく述べる．

次節では，これまで述べた手法をもとに実際に構築したシステムの実装に関して詳述する．

5. 実　　装

本節では，提案手法である各文書の関係を抽出する手法を実際にシステムとして構築した際の実装方法について述べる．文書の管理方法や，前節で定義及び抽出アルゴリズムを与えた文書間の関連について，システムとして必要な関連を選別し，実際に実装したものについて，その実装方法について詳しく述べる．

5.1 対象とする文書

本システムを実際に稼動させたときに対象となる文書について述べる．

理想は全ての文書を等質に扱う事なのだが，簡単のため，本システムでは Web ページと，ローカルドキュメントとして Office 文書のみを扱うものとする．

5.2 文書の管理

文書の管理方法として，本システムでは Web ページについてはローカルプロクシとして，Office 文書に対しては各アプリケーションをフックするシステムとして動作する．これによりユーザの文書閲覧行為に介入し，以下の情報を各文書に対して保持する．

- ページの URL/ドキュメントのフルパス
- 遷移元ページの URL
- 検索クエリ
- タイムスタンプ
- 共使用スコア

以下で，これらの情報を用いて実際に文書間の関連を抽出するシステムの実装方法について詳述する．

5.3 文書間の関連抽出システムの実装

ここで，システムが 3 節で述べた文書間の関連の抽出をするための，各関連のスコア算出の実装方法として選んだ手法について述べる．

5.3.1 構造的関連

構造的関連に対して，本システムはハイパーリンク関連を用いてスコアを算出する．

なお，ドメイン関連を用いなかったのは，3 節でも述べたとおり，本システムでは木構造を伴わないグラフで表現するため相性が良くないうえ，同じドメイン間の関係はハイパーリンク関係からでも十分に強い関係として取れるためである．

文書内のハイパーリンク要素は，$< a >$ タグのコレクションを取得し，各タグの URL を抜き出し，そのページの持つリンクリストデータとして抽出している．

このようにして得られたリンクリストから，それぞれの URL に対してハイパーリンク関連があるかどうかを判定している．

5.3.2 内容的関連

内容的関連について，本システムでは各文書の TF-IDF を用いた特徴ベクトルのコサイン類似度を用いて関係の強さを抽出する．文書から, [8]MeCab の形態素解析を用いて名詞，動詞，形容詞を抽出する．これから各文書の特徴ベクトルを作成し，全ての文書に対して特徴ベクトルを作成し終えたらそれを基に idf を算出する．

これらを用い，関係を取りたい文書の特徴ベクトルのコサイン類似度を計算して，文書間の内容的関連のスコアとしている．

5.3.3 共通ノード関連

共通ノード関連について，本システムでは共通遷移リンク関連，共通クエリ関連，共通タグ関連を用いて抽出する．

これらの情報については各文書に共通する元リンクやクエリ，タグを抽出し，各文書をグルーピングし，関連とする．

まず，関連を取る文書群から，対象となるグループノード（クエリ・元リンク・タグ）を全て抽出する．その後，各グループノードに対して，属する文書を探してグループノードにグルーピングを行う．

このグループノードと属する文書との間にエッジを持つような関連を持たせる．

5.3.4 共使用関連

共使用関連に対して，本システムでは 3 節にも記した，付箋の参照履歴を用いてスコアリングを行うアルゴリズムを用いて実装を行う．

共使用関係の抽出には共起時間が用いられる事が多いが，この手法ではユーザが複数のタスクを同時に行っている場合や，文書を閉じずに次のタスクに移ってしまった場合に関係のない共使用関係が出てしまう問題がある．そして，本システムではそのような状況を想定したものであるため，共起時間を用いず，参照履歴内での共起回数と共起間隔を用いた共使用関係の抽出

を行っている．

ここで，スコアの算出を実装する際に必要な情報として参照履歴があるが，これを取得するため，参照付箋とアクティブタイムを持つ参照データをクラスとして実装し，これをリストとして保持する事で参照履歴とする．

各スコアは参照リストの変更と同期して値を修正される．

求まった共使用スコアを用いて，共使用関係を求める．共使用関係を求めたい文書 A の共使用スコアを参照し，スコアの高い順にスコアリストをソートして，$Co_{A->i} \leqq K$ となる上位 N 件の文書を文書 A と共使用関係にある文書として提示する．なお，K, N は任意の定数とする．

5.4 可視化システムの実装

この節では，前節までの実装により抽出した関連を可視化するために構築したシステムの実装方法について述べる．

グラフの挙動は，物理演算を用いてノードやエッジ間の距離や位置をリアルタイムで変更できるライブラリを用いた．

これを用いて，各関連を色の違うエッジで表現することで文書間の関連を視覚化している．また，エッジは関連の有無を表現するだけではなく，その強さによって収縮もする．そのため，文書間の関係の強さも視覚的に捉えることができるものである．このようなエッジを各関連毎に独立に張ることで，文書はそれぞれの関連による引力を受け，意味的関係に即した位置関係を取る事ができる．

6. システムの入出力と運用イメージ

ここでは，システムの入出力についてや，実際に起こりえると考えられるタスクを実行した際に得られた結果を紹介する．

6.1 システムの入出力

入力については，任意の数の文書群か，1 つのシード文書となる．

これに対して出力は，入力が文書群であった場合はその文書群の関係を，シード文書であった場合は，その文書と共使用関係にあった文書を抽出し，その文書群の関係を出力とする．

これにより，単なるタブブラウザや履歴，ブックマークではできない，それを閲覧していた時のタスクを想起させるような文書群の提示が可能となる．

6.2 システムの運用例

以下にシステムの運用例として「デジタルカメラの購入検討」の例を挙げ，システムの出力について考察する．

ユーザはまず検索エンジンに「デジカメ」と入力して検索を行う．そこで結果として出てきた価格比較サイトにて人気のデジタルカメラ数点を閲覧，気に入ったものについて各メーカーのページを検索してから覗き，商品について調べた．

以上のようなタスクを本システム上で終えた後にページの可視化を行うことで，図の結果が得られた．

ハイパーリンク関連や共通クエリ関連から価格比較サイトで閲覧した商品ページ群，カシオやパナソニックのメーカーのページ群がそれぞれで固まって表示されている．また，各クエリを入れた検索ページが中央に位置しており，適切に関係が提示されていると言える．

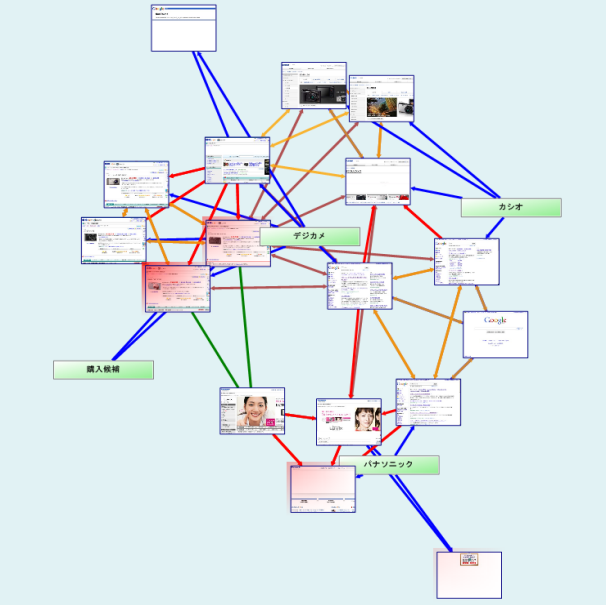

図 1 出力結果

7. 評価実験

システムの評価実験として，タブブラウザとの比較を行った．以下に実験の概略，及びどのような事象が観測できたかを記す．

7.1 実験概略

タブブラウザとの比較を行うため，ユーザには予め用意したページ集合について，そのページ集合が作られるまでを追体験してもらい，そのページ集合から具体的な情報を見つけ出すようなタスクを設定した．

ページ集合については,「デジタルカメラの購入検討」という想定のもと開かれた 26 文書,「金沢旅行で巡る観光スポットの検索」という想定のもと開かれた 31 文書の 2 つの状況を用意し，それぞれ「カシオ EXILIM ZOOM EX-Z7330 の値段を調べる」「伏見寺の営業時間を調べる」といったタスクを，タブブラウザを用いて 2 つずつ，本システムを用いて 2 つずつ行ってもらった．

これらの各タスクについて，情報を発見するまでに開いたページの数，情報を発見するまでにかかった時間を計測し，評価とした．

なお，ユーザを 2 グループに分け，タブブラウザで行うタスクと本システムを用いて行うタスクを入れかえて行ったため，各状況に対し，4 つのタスクについて実験結果が得られた．

また，本実験は，情報検索には慣れているものの，本システムを扱うのは初めてである 4 人の被験者に実施してもらった．

7.2 実験結果

実験の結果として，表 1，2 のような結果が出た．なお，いずれも 4 つのタスクの平均値を取っている．

情報を得るまでに開いたページは減少しているが，かかった時間がタブブラウザと比べて長くなってしまうという結果が出た．

表 1　情報を発見するまでにかかった時間（単位は秒）

状況	タブブラウザ	本システム
デジカメ購入	23.1	54.8
金沢旅行	25.9	36.8

表 2　情報を発見するまでに開いたページ数

状況	タブブラウザ	本システム
デジカメ購入	7.8	3.4
金沢旅行	11.1	2.6

7.3　考察と今後の課題

情報を得るまでに開いたページは確実に減少しているため，ページ閲覧の効率自体の上昇には成功している．

問題はタスクにかかる時間が多くかかってしまっている点だが，実験時のユーザの行動を観測していると，タブブラウザでのページ閲覧では，該当するページに当たるまで無根拠にページを開いているのに対し，本システムを使っている時は，該当するページを意味的に推論しながらグラフ上を探すという行動を取っていた．この行動の中で，商品比較サイト上での商品ページなどは，周辺に似たような商品のページが多く，サムネイルからではうまく判別できず，ページの同定に時間がかかってしまうような状況がよく見られ，これがタスクに時間がかかってしまった大きな理由だと考える．また，どのユーザも本システムを使った最初のタスクを行う時に，システムの利用に慣れていないために時間を多く取られてしまっており，これがデジカメ購入のタスクにおいて，タスク遂行時間が大幅に伸びてしまった原因である．

このため，今後の課題として，よりページの特徴を捉える複雑な関係の抽出・提示や，ユーザインタフェースを改善するなどして，個々のページを同定しやすくすることが挙げられる．

また，今回の実験ではタブブラウザと本手法の比較を行ったが，両手法の間には文書の可視化と文書間の関係抽出という 2 ステップの差異があるため，関係抽出の有意性を示すために，文書の可視化だけを行った手法との比較実験等も今後行っていきたい．

これらの課題を解決し，タスク遂行時間においてもタブブラウザを上回り，よりストレスのない文書閲覧の提供を目指したい．

8.　結　　論

本稿では，複数の閲覧文書の間にある意味的な関係を抽出・提示することによる文書ナビゲーションの手法について論じた．

文書間には様々な関係がある事が考えられるが，本手法ではその内の，ハイパーリンク関連，内容類似度関連，共通遷移リンク関連，共通クエリ関連，共通タグ関連，共使用関連を用いて各文書間の関連を求め，それを可視化した．

これらの関係を独立に，かつ同時に可視化することで，文書を独立に扱う現状のブラウジングやドキュメント管理に比べ，ユーザにタスクの想起を促し，文書の閲覧効率を向上させることができた．

今後の課題として，相反する文書や比較文書といったより複雑な文書間の関係や，関係間の関係を抽出するなどして，より高度な関係を扱うことや，可視化のインタフェースについて，よりユーザにとって直感的で理解のしやすいインタフェースを実現するなどの課題が挙げられる．これらの課題についても，今後解決していきたい．

謝辞　本研究の一部は，京都大学 GCOE プログラム「知識循環社会のための情報学教育研究拠点」，および，文部科学省科学研究費補助金（課題番号：18049041，21700105），および，独立行政法人情報通信研究機構の高度通信・放送研究開発委託研究「電気通信サービスにおける情報信憑性検証技術に関する研究開発 課題ア Web コンテンツ分析技術」（研究代表者：田中克己）によるものです．ここに記して謝意を表します．

文　　献

[1] 佐野博之, 大園忠親, 新谷虎松:“ 付箋アノテーションを用いた情報共有システムの試作 ”, 第 22 回人口知能学会全国大会, 2G1-03, (2008).

[2] 豊田正史，喜連川優：日本のウェブアーカイブにおけるコミュニティ発展過程の詳細分析，第 14 回データ工学ワークショップ DEWS ’03, 2-P5, (2003).

[3] 日野亜希子，日野洋一郎，中村聡史，田中克己：ウェブ上の文書を利用した small knowledge の自己組織的関連付け手法，第 19 回データ工学ワークショップ DEWS 08，B6-6，(2008)

[4] Jun Rekimoto, Time-machine computing: a time-centric approach for the information environment, Proceedings of the 12th annual ACM symposium on User interface software and technology, p.45-54, (1999)

[5] 定免睦昌, 國島丈生, 横田一正: 複数タスク環境におけるユーザ操作に基づくファイル間関連度の導出, 第 8 回日本データベース学会年次大会, D1-3, (2010)

[6] 大澤亮, 高汐一紀, 徳田英幸: 俺デスク:ユーザ操作履歴に基づく情報想起支援ツール, 情報処理学会第 47 回プログラミング・シンポジウム, (2005).

[7] Yousuke Watanabe, Kenichi Otagiri and Haruo Yokota: FileSearchCube: A File Grouping Tool Combining Multiple Types of Interfile-Relationships, Web-Age Information Management11th International Conference, p.386-397, (2010)

[8] “MeCab: Yet Another Part-of-Speech and Morphological Analyzer”, http://mecab.sourceforge.net/