修 士 論 文

メタデータを活用した
Web コンテンツの分類と活用法

Classification and application for web contents utilizing Metadata

2008 年 1 月 29 日

指導教員　相田 仁 教授

東京大学大学院
新領域創成科学研究科 基盤情報学専攻

66325　佐 伯 嘉 康

内容梗概

インターネットが普及し，そのアプリケーションの1つであるWeb上には様々な者達が記述し公開している，様々なコンテンツが存在している．

利用者がWebコンテンツを発見するためには，個人または団体が設置した検索エンジンを用いて，Webコンテンツに含まれるであろう単語を検索クエリィとする．しかし，人手により検索データベースを構築しているディレクトリィ型検索エンジン，エージェントにWebを巡回させ検索データベースを構築するロボット型検索エンジンのいずれも，膨大な量となったWebコンテンツを全て網羅しているとは言えず，Web上に公開した情報，知識が等しく適切に活用されるためには，新たな機構が必要となっている．

そこで本研究では，WebコンテンツのAuthorityとHubに分割するHITSのアルゴリズムを改良し，WebコンテンツをWebコンテンツに付与したメタデータを活用して分類，集約する事で利用効率の向上を目指すための，メタデータマイニングを提案し開発した．その結果，改良無しHITSよりも，提案手法が効果がある事を証明した．また，提案したメタデータマイニングを活用し，効率的にWebコンテンツを活用するための環境の例として，Webコンテンツをメタデータマイニングの結果に応じて配信し，利用者がWebコンテンツを活用する際の支援となるシステムを提案した．

目次

図目次

表目次

第1章

序論

概要

本章では，本研究の動機のための背景と問題提起，そして本論文の構成について述べる．

1.1　研究の背景

インターネット上で構築されるWorld Wide Web(WWW，以下Web)は，Tim Berners-Lee[1]が1990年に提案し実装して公開して以降，日常生活に於いても学術研究に於いても，欠かす事の出来ない技術(インフラストラクチャ)となっている．

世界的に見れば，ITU(International Telecommunication Union)[2]による2006年の調査『WORLD TELECOMMUNICATION/ICT INDICATORS』 [9]の中の『Internet indicators: subscribers, users and broadband subscribers』によると，世界のインターネット利用者人口は約11億3659万人(人口普及率約17.47%)となっており，また，日本に於ける総務省[3]による平成19年版『情報通信白書』(情報通信に関する現状報告) [16]によれば，インターネット利用者人口は年々の上昇を続け，平成18年(2006年)には約8754万人(人口普及率約68.5%)となっている．この利用者全てがWebを利用していると想定しても良いだろう．また，同報告書によれば，個人の解説したblog [4]を閲覧するユーザは調査対象の約40%となっている．つまり，インターネットの普及により，Webは個人の情報発信の場を敷居を低くして提供し，それを利用する者も，ある程度の利用価値を見出しながら，Webを積極的に利用していると考えられる．

しかし，Webは当初の想定から僅かに外れた道を歩みながら，つまり知識の集約の場としての機能が上手く働かず，不特定多数の作成者による文書やファイルが氾濫した事により，所謂，情報爆発[5]という現象が起こった．爆発的に増大し続けるWeb上のデータは，作成者と利用者の間の橋渡しが未完のまま，そのデータの持つ意味とは関係無く，何度も再利用されるもの，限りなく発見不可能の状態となるものに分けられている．多くのWeb上のデータの発見には，Google[6]やYahoo![7]等の検索エンジンが用いられており，プログラムのクローリング，あるいは人による検索データベースへの登録が他者への発見のために必要不可欠になっている．

これは，Web上のデータの淘汰による繁栄と駆逐による分類をもたらすものであ

[1] “Tim Berners-Lee”, http://www.w3.org/People/Berners-Lee/
[2] “International Telecommunication Union”, http://www.itu.int/net/home/index.aspx
[3] “総務省”, http://www.soumu.go.jp/
[4] Weblogとも，時系列的に更新される日記や特定テーマについての個人またはグループのWebサイト
[5] “info-plosion 情報爆発”, http://www.infoplosion.nii.ac.jp/info-plosion/
[6] “Google”, http://www.google.com/
[7] “Yahoo!”, http://www.yahoo.com/

り，ハイパリンクによってあらゆる情報をリンクし，Web上のデータの再利用を積極的に促すWebの本質では無い．Web上のデータが等しく，利用者に積極的に利用される様な環境を，Webの中あるいはWebの外から提供する必要があると考える．

1.2　研究の目的

本論文で述べる研究は，研究の背景として述べた問題点を解決するための一手法として，Webのデータ(以下，Webコンテンツ)にメタデータ(データの意味を記述したデータ)を付加し，WebコンテンツのWeb上での流通の促進のための道具として利用する事を目指す.

加えて，新たなWebとして実装が進んでいる，Semantic Webについても言及し，その意味と問題点を考察する．

1.3　本論文の流れ

本論文は以下の様に展開する．

第2章に於いてインターネットアプリケーションの1つであるWebの社会的基盤，技術的基盤としての概説と基礎技術について述べる．そして，Webの再定義，応用としての，Semantic Webについて述べる．そこでは，Semantic Webの中心技術である，メタデータ，メタデータを記述するための構造を定義したResource Description Framework(RDF)についても解説する．

第3章では，データを活用する方法として，データマイニングの概要と方法について述べる．そして，Web上に存在するデータを活用するための方法として，Webデータマイニングについて述べる．併せてデータマイニング，Webデータマイニングの先行研究について述べる．

第4章では，自身の研究として，メタデータを用いたWebデータマイニング(メタデータマイニング)について述べる．ここでは，メタデータマイニングの活用例として，メタデータを付与したWebデータ(Webコンテンツ)のPush配信システムを作成する．

第6章では以上の研究に対する評価，考察，発展，問題点を述べる．

最後に，第7章で，本論文の結論を述べる．

第2章

WebとSemantic Web

概要

本章では，今や生活や研究の中に浸透し，必須の技術となっているWebについて社会的側面，技術的側面から解説する．そして，Webの新たな進化形であるSemantic Webについて言及する．

2.1　社会基盤としてのWeb

第1章でも述べたが，インターネットアプリケーションとしてのWebは日本のみならず世界中にあらゆる分野で浸透している．

ところで，日本では総務省が提案した政策である“u-Japan”[1]を中心に浸透しているユビキタス社会は,「神は遍在する」の意味を持つUbiquitousの語源であるラテン語のUbiqueの意味から「いつでも，どこでも(、だれでも)」というキャッチフレーズを頻用し，コンピュータ同士が自立的に連携して動作する事で，人々の生活を支援する技術による環境作りを目指している．ここで使われるコンピュータというのは，センサなどの小型デバイスを含め，PDAや携帯電話等のモバイル機器が主となっている．つまり，会社や家にあるコンピュータでネットワークに接続し，利用するのではなく，個人の持つ情報機器がネットワークの要となる．

この様に，個人が台頭するネットワークの始まりは，Webである．Webの浸透は，まず大学内ネットワークから始まった．1993年末頃から学生が個人のWebサイトを作成し始め，Webコンテンツは次第に増加し，商用，または個人が作成した検索エンジンが必要不可欠となり，Web上のWebコンテンツを全て利用する事は誰にも不可能になっていった．不特定多数の個人が，爆発的にWebコンテンツを増大させた．

Webは個人主義の元で，全ての他の個人と同じレベルに立つ事が出来，個人の情報を分け隔て無く発信し，個人の責任の下で，あるいは責任を限りなく小さくした匿名者として，時間的，または空間的制約から解放され，クリエイティブな発信者になる事が出来る．Webは，そういう場を構築し提供した．

2.2　技術基盤としてのWeb

Webはインターネットのアプリケーションの1つである．クライアントサーバモデルに基づき，Webコンテンツというリソースのある場所をURI(Uniform Resource Identifier)によって指定し，主にプロトコルとしてHTTP(Hypertext Transfer Protocol)を用いて，要求と応答によってリソースが転送される．

[1] “u-Japan 政策”, http://www.soumu.go.jp/menu_02/ict/u-japan/index.html

多くのWebコンテンツはHTML(HyperText Markup Language)の系統の書式で記述され，単方向のハイパリンクを持っている．リソースにリンクを張る際にリソースの管理者に連絡する必要はなく，これによりWebを構成するWebサーバやWebブラウザの実装が容易になっている．その反面，リソースの相互管理が不可能になり，移動や削除等に伴う消失に関して対応できないという問題はある．

Webは進化を続け，HTMLファイルのみならず，あらゆるフォーマットのファイルを扱う様になった．また，JavaScript[2] (ECMAScript[3]) がクライアント側のWebブラウザに実装された事によりDHTML(Dynamic HTML)によるより柔軟なインタフェースや構造を持つWebコンテンツがWeb上に存在出来る様になった．

2005年中頃から "Web 2.0"[4] というバズワードと共に，Webの進化形が問われ，Ajax(Asynchronous JavaScript + XML)と呼ばれるWebブラウザがWebサーバと非同期に通信する事で，感覚的にシームレスなインタフェースを提供し，WebのRIA(Rich Internet Application)を既存技術の範囲で開発者側にも利用者側にも容易に実現した．また，"Web 2.0" は "Long Tail"(塵も積もれば山となる)，"Data中心"，"参加型Web" などの考えを広め，一時の流行を作った．

Webは誕生当時から，先進的な技術の試験運用，新たな技術分野の誕生の場として貢献し続けている．

2.3 Semantic Web

以上で述べたWebには，現在様々な視点から問題点が見える．

まず第一に，多数のWebコンテンツはHTML等の書式を利用し最低限の構造のみを持ったコンテンツである．一般的にデータベースに分類されるシステムでは，システムがデータの構造をある程度の意味として管理し，制御する事が出来る．しかし，Webはその仕組みを持たず，構造は最早何の意味も持っていない．

第二に，ハイパリンクの単方向性のために，全てのリソースへのアクセスが不可能になっている現実がある．後出のリソースは先出のリソースへのハイパリンクを張ったとしても，先出のリソースから後出のリソースへアクセス出来ないために，利用者の利用行動を制限する事になる．解決策として，Trackback[5]という参照を示すための仕組みがあるが，ハイパリンクを双方向にする様な効果は得られず，リ

[2] "Core JavaScript 1.5 Reference - MDC", http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Reference

[3] "Index of Ecma Standards", http://www.ecma-international.org/publications/standards/Stnindex.htm#Software

[4] "O'Reilly – What Is Web 2.0", http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[5] "TrackBack Technical Specification", http://www.sixapart.com/pronet/docs/trackback_spec

ソース到達不可能排除，発見支援には至っていない．

以上等の現状のWebに於ける問題点を解決するために，あるいはWebを見直し，より良くするために，Webの考案/開発者であるTim Berners-LeeがSemantic Webを提唱した [2].

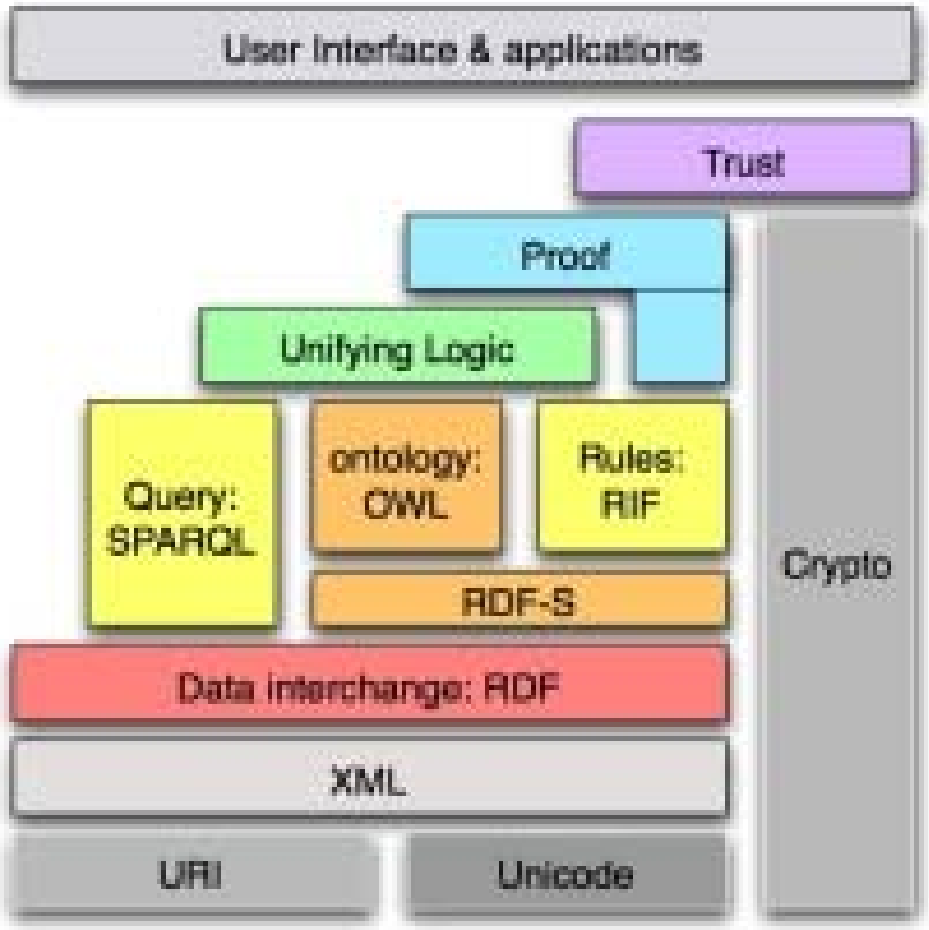

図2.1: Semantic Web Layer(2006)

図2.1は，Semantic Webを構成する技術をレイヤケーキとして表現したものである．図は2006年7月にボストンで行われた“AAAI Conference”にて，Tim Berners-Leeが発表したスライド [3] に使われた図である．

下位の基盤技術として，Webコンテンツを記述するための言語(文字コード)として“Unicode”[6]，Webコンテンツの場所を指し示すための“URI(Uniform Resource Identifier)”[7]，Webコンテンツを記述する書式としての“XML(Extensible Markup Lan-

[6] “Unicode Home Page”, http://www.unicode.org/

[7] “RFC 3986 Uniform Resource Identifier (URI): Generic Syntax,” http://tools.ietf.org/html/rfc3986

guage)”[8]，XMLにより記述したWebコンテンツを分類するための “Namespace(名前空間)”[9]が定義されている．

上位の応用技術として，“User Interface(ユーザインタフェース)” や “Application(応用)” の他，Webコンテンツに対する “Trust(信頼性)” や “Proof(証明)” を保証するための技術も，Semantic Webレイヤケーキには組み込まれている．また，信頼性を支える技術として，“Crypt(暗号)” がある事も見て取れる．

ここで注目したいのは，中位の技術として定義されている，“Data interchange: RDF” が支えている，メタデータに関する技術である．次節2.4ではこのRDFについて解説をする．

2.4　メタデータ構造記述枠組み: RDF

2.4.1　RDFの概要

Semantic Webを支える最も重要な技術は，メタデータに関する技術である．Semantic Webで扱うWebコンテンツには全てメタデータが付与されている事が前提になっている．

RDF(Resource Description Framework) [20]は，Semantic Webに採用されている，メタデータを記述するための枠組み，意味を表現するためのモデルである．

RDFはメタデータをTriple(3つ組)のグラフの集合として表現する．Tripleの葉はSubject(主語)，Object(目的語)に分けられ，枝はPredicate(述語，語彙)となる．それぞれの要素にはResource(リソース，URIによって指し示される全てのもの)を定義する事が出来る．またObjectには，定数値やEmpty(空)のデータを定義する事が出来る．

図2.2: RDFの三つ組グラフ

図2.2は，Tripleを図として表現したものである．このRDFは自然文で書くと「[Subject]の[Predicate]は[Object]である」となる．

[8] “Extensible Markup Language (XML) 1.0 (Fourth Edition)”, http://www.w3.org/TR/xml/, “Extensible Markup Language (XML) 1.1 (Second Edition)”, http://www.w3.org/TR/xml11/

[9] “Namespaces in XML 1.0 (Second Edition)”, http://www.w3.org/TR/xml-names/, “Namespaces in XML 1.1 (Second Edition)”, http://www.w3.org/TR/xml-names11/

2.4.2　RDF と Namespace

RDF は，そのデータの内外に多種多様な言語，ジャンルの語彙が存在する事を前提とし，保証している．

その保証の為に Namespace(名前空間) の技術が用いられている．名前空間と語彙を結び付ける事によって，語彙が同じであっても異なる名前空間である事を明示出来る．

これは，Web 等の分散環境，不特定多数の作成者が多種多様なデータを記述する環境に於いて，RDF データを構築するために必要な技術である．

2.4.3　RDF の記述

RDF は，メタデータの記述方式ではなく，意味の表現モデルである．従って，RDF によるメタデータを記述する方法には幾つかある．

1 つは RDF/XML [18] である．コンピュータ同士の通信に使われるデータ記述言語として広く普及している，HTML などと同様のマークアップ言語である XML を用いた記述方式である．RDF/XML で記述した RDF データの例として．図 2.3 を示す．

別の記述方式として，N-Triples がある．N-Triples は Notation3 という RDF を記述するための方式をさらに簡略化し．RDF/XML の冗長性を排除し，見易く (書き易く) するための記述方式である．「<主語> <述語> <目的語>.」の 1 行で 3 つ組のグラフを記述する．N-Triples で記述した RDF データの例として，図 2.4 を示す．

図 2.3 と図 2.4 は共に，図 2.5 を記述したものである，主語を URI“`http://example.com/Alice`” によって示される “Alice”，述語を URI“`http://example2.com/family#father`” によって示される “father”，目的語を URI“`http://example.com/Bob`” によって示されるリソースで名前が “Bob” である，とすると，単純な自然文で表すと「Alice の父は Bob である」となり，メタデータに関する情報は “`http://example2.com/family`”，または “`http://example2.com/person`” に記述されているとする．

両者記述方式を比較すると，N-Triples は RDF/XML より幾らか簡潔に RDF によるメタデータを記述する事が出来る事が分かる．RDF の記述方式は他にも幾つかある．記述方式の使い分けは，その RDF メタデータを用いるアプリケーション (プログラム) がその記述方式によるデータを処理できるかに大きく依存する．一般的に見て，XML による記述は，多くのプログラミング言語がその処理に対応している事から，広く用いる事が出来ると考えられる．

もう 1 つの記述方式として，RDFa(RDF/A とも，RDF with attributes) [19] を挙げる．

上に挙げた記述方式 (RDF/XML，N-Triples) は，現在の Web に最も多く流通している (X)HTML によって記述されたデータとの親和性が低いものである．また，エディ

```
<?xml version="1.0" ?>
<rdf:RDF
 xmlns:rdf=http://www.w3c.org/1999/02/22-rdf-syntax-ns#>
  <rdf:Description rdf:about="http://example.com/Alice">
    <http://example2.com/family#father>
      <rdf:Description rdf:about="http://example.com/Bob">
        <http://example2.com/person#name>
          Bob
        </http://example2.com/person#name>
      </rdf:Description>
    </http://example2.com/family#father>
  </rdf:Description>
</rdf:RDF>
```

図 2.3: RDF/XML による RDF データ例

```
<http://example.com/Alice> <http://example2.com/family#father>
         <http://example.com/Bob>.
<http://example.com/Bob> <http://example2.com/person#name> "Bob".
```

図 2.4: N-Triples による RDF データ例

タなど記述支援が不足している事もあり，これら記述方式を採用するには大なり小なり敷居がある事が分かっている．従って，これらの記述方式による RDF データを流通させるためには，(X)HTML によるデータを置き換えるか，(X)HTML によって記述されたデータを編集しメタデータとしての RDF データとのリンクを張る必要がある．Semantic Web の普及を進めるために，また，現在の Web の良い状態を維持するためにも，これらは得策ではない．従って，(X)HTML データに直接，RDF メタデータを記述するための埋め込み記述方式が提案された．それが RDFa である．

RDFa は，RDF データを XHTML の文書データの Tag(タグ) に Attribute(アトリビュート) として記述する．

名前空間の宣言として，RDF と同じく「`xmlns:PREFIX="IDENTIFIER"`」を用い，メタデータは,「`property="metadata"`」「`rel="metadata"`」「`role="metadata"`」などを用いて記述する．RDFa を埋め込んだ XHTML 文書は，XHTML 1.x または 2 の文書として解釈する事が出来る．名前空間を宣言する事が出来るので，RDF と同じく語

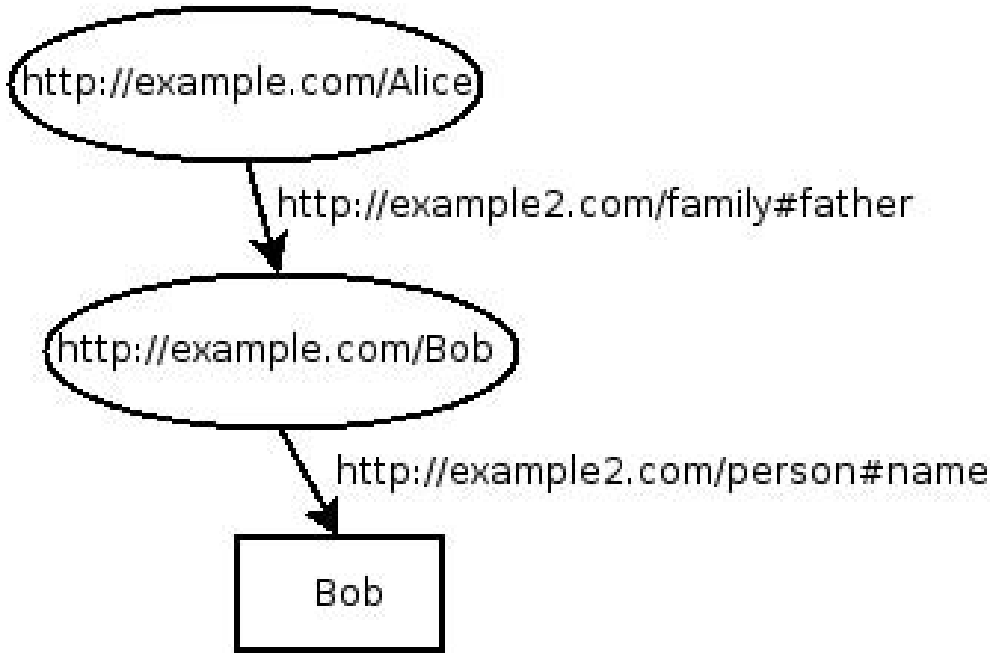

図 2.5: RDF グラフ例

彙の衝突や矛盾を防ぐ事が出来る．

RDFa で記述した RDF データの例として，XHTML 文書データの一部を図 2.6 に示す．また，この文書データは Web ブラウザで読み込むと 2.7 の様に表示される．これは現在の Web にある (X)HTML によって記述された文書データと同様に，RDFa によって記述された RDF データが変わりなく文書として表示されているが，データを見ればメタデータが埋め込まれている事を示している．

同様の，XHTML のアトリビュートにメタデータを記述する手法として，W3C ではなく有志の団体 microformats.org によって開発された microformats [10] がある．特に，検索エンジン大手の Yahoo![11] や Microsoft [12]，ブログ検索エンジンで知られる Technorati [13] が注目している技術である

メタデータの付加に対して似たアプローチに見える RDFa と microformats の大きな違いの 1 つに，RDFa が RDF と同じく名前空間により語彙の衝突を回避しているのに対し，microformats は名前空間を持たず，小規模か，人がデータを理解している事を前提とした環境で用いられることを想定している，という事がある．言い換えれば，RDFa は RDF との互換を想定し，microformats は想定していない，という事である．

Web で用いられることを想定した場合，複数の Web コンテンツを組み合わせて用

[10] “microformats”, `http://microformats.org/`
[11] “Yahoo!”, `http://www.yahoo.com/`
[12] “Microsoft Corporation”, `http://www.microsoft.com/`
[13] “Technorati: Front Page”, `http://www.technorati.com/`

```
<div
 xmlns:exf="http://example2.com/family#"
 xmlns:exp="http://example2.com/person#"
 about="http://example.com/alice">
  Alice についての記述.
  Alice の父は
  <a href="http://example.com/Bob" rel="exf:father">
   この人
  </a>
  で，名前は
  <span
   about="http://example.com/Bob" property="exp:name">
    Bob
  </span>です.
</div>
```

図 2.6: RDFa による RDF データ例

Alice についての記述.
Alice の父は この人 で，名前は Bob です.

図 2.7: RDFa による RDF データ例の Web ブラウザによる表示

いる場合が多い．その時，語彙の衝突などによる矛盾や混乱は適切に回避する必要がある．従って今後，Web から Semantic Web に移行する時期に於いて，RDFa を積極的に用いる事が必要であり，適切であると考える．

本研究では，以上で述べた利点から，RDF によって表現されるメタデータを用いている．

第3章

データマイニングとWebデータマイニング

概要

本章では，データの集合を解析し，知識を獲得するための手法としてのデータマイニングと，その技術をWebのデータを対象に特化したWebマイニングについて解説する．

3.1　データマイニング

データマイニング(Data Mining)は，データの集合からデータの性質(モデル，パターン)を表す，構造，規則，関係を発見するための手法である．

データマイニングは，データベースによる知識発見という過程の1つのステップとして定義出来る．知識発見の過程を以下に示す.

1. データ洗浄: データベースからノイズや一貫性を取り除く
2. データ統合: 必要に応じて複数のデータソースを統合する
3. データ選択: データベースから分析対象のデータを選択(検索)する
4. データ変換: マイニングに適したデータ構造に変換する
5. データマイニング: パターン抽出
6. パターン評価: ある尺度(関心度)に従って興味のあるパターンを同定する
7. 知識表現: マイニングされた知識を効果的にユーザに提供する

3.1.1　相関ルール

Association Rule(相関ルール)とは，データの関連性から，例えばあるデータ(Antecedent，条件)から別のデータ(Consequent，帰結)を導く事が出来るための，Confidence(信頼度)，Support(支持度)を求め，そのデータの重要度を測るものである．

事実を規則として表し，ある事象が発生した対象は別の事象も発生する(であろう)という予測を導き出す事に使われる．

相関ルールの定義は，以下の様になる．

相関ルールを求めるためには，データに含まれる頻出アイテム(セット)を求める必要がある．アプリオリ・アルゴリズムはその方法の1つで，アイテムセット$_k$からアイテムセット$_{k+1}$を求める．そして，アイテムセットが揃ったところで，計算式に基づきアイテムセットの信頼度，重要度を求める．

3.1.2　分類

Classification(分類)とは，データに対応するカテゴリィを予想するものである．カテゴリィは既知のものであり有限である．

分類作業は，学習(Step 1)によってデータの分類概念を蓄積し，実際の(狭義の)分類(Step 2)を行う．学習にはベイズ分類，多数決，遺伝的アルゴリズムが用いられ，データ構造として決定木などが用いられる．

3.1.3　クラスタリング

Clustering(クラスタリング，グループ化)とは，データの集合をある似た属性からクラスタというグループに分けるものである．分類と異なり，クラスタは既知のものではない．従って，分類と区別を分かり易くするために分割と呼ぶ事もある．

クラスタリングは類似度という概念を用いる．類似度は距離を用いる場合が殆どである．距離には，各座標からの単純な距離であるユークリッド距離，各座標の差の総和を距離とするマンハッタン距離，上記2種の距離の一般化であるミンコフスキー距離がある．

他にも，クラスタの密度，セルやグリッド構造に基づくもの，ニューラルネットワークに基づくものなどがある．

3.2　Webマイニング

以上に述べた，データマイニングの手法を，Webのデータ解析に当てはめたものが，Webデータマイニングである．

対象となるデータは，Webに特化して多く存在する(X)HTMLによって記述された文書であり，構造がある程度決まっている分，関連性をハイパリンク，URLや文書に含まれる単語，属するドメインなどによって効率的に，また正確に抽出する事に重点が置かれる．

Webマイニングは，ここでは大きく3つの分類に分ける．Webコンテンツの集合全体を1つあるいは複数のネットワークと見なし有向グラフなどでモデル化するWeb構造マイニングと，Web上に存在するWebコンテンツの集合の構造を内容の解析と共にマイニングするWeb内容マイニングと，Webコンテンツを利用する利用者の行動を分析した内容の解析と共にマイニングするWeb履歴マイニングである．

3.2.1　Web構造マイニング

Web構造マイニングは，Webコンテンツの集合の構造を，有向グラフなどでモデル化し，マイニングに活かすものである．有向リンクはWebコンテンツが持つハイパーリンクに対応させて考える事が出来る．

先行研究例として，IBM他の発表 [7]，Andrei Broderの研究 [5] がある．これによると，1999年に収集した約2億のWebページと15億のハイパリンクの調査をしたところ，Webは4つの領域「IN(origination)」21%,「SCC(strongly-connected core)」29%,「OUT(termination)」21%,「TENDRIL(tendril)」21%に加え「DISC(disconnected)」8%に分かれているという調査結果が出た．

INに存在するWebページははSCCに存在するWebページにアクセスしており，SCCに存在するWebページはOUTに存在するWebページにアクセスしている．その逆の繋がりは弱い．TENDRILはCOREに存在するWebページにアクセスせず，INから，またはOUTへアクセスしているWebページである．DISCはどこにもハイパーリンクが張られていないWebページである．これらの連結構造は図3.1の様な蝶ネクタイ構造に例えられる (蝶ネクタイ理論).

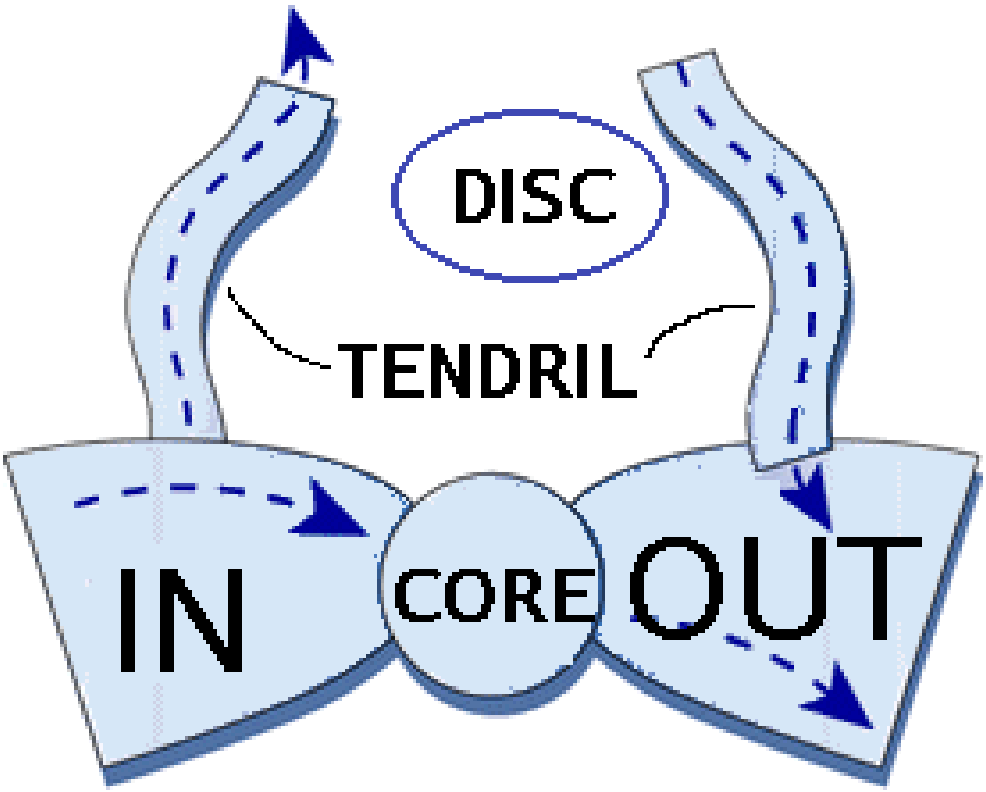

図3.1: Webの蝶ネクタイ構造

この調査研究は2000年のものであり，過去の結果として現在のWebの構造には

合致しない可能性もある．外国で「Weblog」が流行し始めた1999年から，日本での「ウェブログ(ブログ)」ブームが始まった2002年以降，特に，2002年にSix Apart社によって作られたブログツール「Movable Type」[1]にTrackBackシステムが実装，仕様が公開され，一般的に使われ始めて以降，ハイパーリンクの構造は確実に変化したと考えられる．加藤らによる研究[10]によれば，2006年に収集した120億のWebページを調査したところ，COREが2000年の調査よりも巨大化している事が分かった．

この様に，Webコンテンツの内容に頼らず，ハイパリンクの解析のみでも，Webコンテンツの関係性からWebの構造を発見する事は可能である．

3.2.2　Web内容マイニング

Web内容マイニングは，Webコンテンツを収集し，通常のデータマイニングと同様に，内容からWebコンテンツの特徴語などを抽出し，検索やWebコンテンツの分類に活用される．

3.2.3　Web履歴マイニング

Web履歴マイニングは，Webコンテンツの利用者の利用履歴などの分析から，Webコンテンツの収集や再設計に活用される．

3.3　Webコンテンツの分類に関する先行研究

Webコンテンツに限らず，テキストの分類に関する先行研究は幾つもあるが，ここでは，Web構造マイニングに属し，Webの特徴であるハイパリンクを解析したWebコンテンツ分類について解説し，その問題点を挙げる．

あるWebコンテンツAが他のWebコンテンツBへハイパリンクを張るという事は，WebコンテンツBがWebコンテンツAと何らかの関係があり，もしWebコンテンツBが多くのWebコンテンツからのハイパリンクを張られているとすれば，WebコンテンツBが人気(ポピュラリティ)が大きい，つまり有用である，と判断する事が出来る．

社会科学に於いて，その様な相互関係ネットワークに関する研究は早くから展開されており[21]，個人や組織をノードと置き，エッジを社会的相互作用として，ノードに対するエッジ(リンク)が増える事は，ノードの社会的水準，社会的人気，社会的信頼に関する指標が増えるとしている．

同様に，論文や学術図書をノード，論文の引用をリンクと見なし，論文の有用性

[1] http://www.sixapart.com/movabletype/

を測る研究がある [6]．この研究は，増え続ける学術図書のデータベースの中で，今後，如何に効率的に適切なものを発見する為に必要なものが何かを考え，その結果として，論文の引用を記録するためのインデックス(索引)として，“Science Citation Index”，“Social Science Citation Index”，“Humanities Index” を作成し，被引用数の大きい(インパクトファクタ [15] の大きい)学術図書を対象とする “Web of Science” というデータベースにまとめた．現在は，この様な引用追跡が可能なデータベースは “Google Scholar”[2]等幾つかあるが，当時は画期的で唯一であった．

しかしWebに於いて，社会科学と同様にリンクの数だけでWebコンテンツの水準や信頼を測る事は，必ずしも正確にはならない．また，学術図書と異なり，被リンクWebコンテンツがリンク元となるWebコンテンツを発見する事は，ハイパリンクが持つ単方向性から不可能に近い．また，学術図書は記述され公開されればそれで多くの場合変更は不可能になり，不変のコンテンツとなるが，Webは記述，公開後も更新する事が可能であり，引用と被引用の時間的制約が破綻し，他のリンク構造とは明らかに異なるものとなる．そこで，Webマイニング等の研究の評価指標には人間による主観的な認識と判断，妥当性と品質の考え方を取り入れる方法が積極的に取られている．

3.3.1 HITS

HITS(Hyperlink Induced Topic Search) [11] は，Webコンテンツの特徴であるハイパリンクを重視するために，Webのハイパリンク構造をグラフと見なしたWebコンテンツクラスタリング手法である．

HITSはWebコンテンツを，“Authority”(特定のトピックに関して権威のあるWebコンテンツ)と，“Hub”(特定のトピックに関して連絡網として権威がある，つまりAuthorityへのハイパリンクを持つWebコンテンツ)の2つの特徴を持つクラスタに分類する．

HITSの基礎アルゴリズムを以下に示す．

1. 検索したいクエリィに関係すると思われるWebコンテンツを一定数r個収集し，root setとする
2. root setを参照またはroot setから参照の距離(パス長)が1であるWebコンテンツの集合をWebコンテンツ1つにつき最大d個収集し，root setに追加する．これをWebコンテンツn個からなるbase setとする
3. base setに含まれる全てのWebコンテンツに対し，Authority，Hubとして評価する重み付けをする

[2] “Google Scholar”, `http://scholar.google.com/`

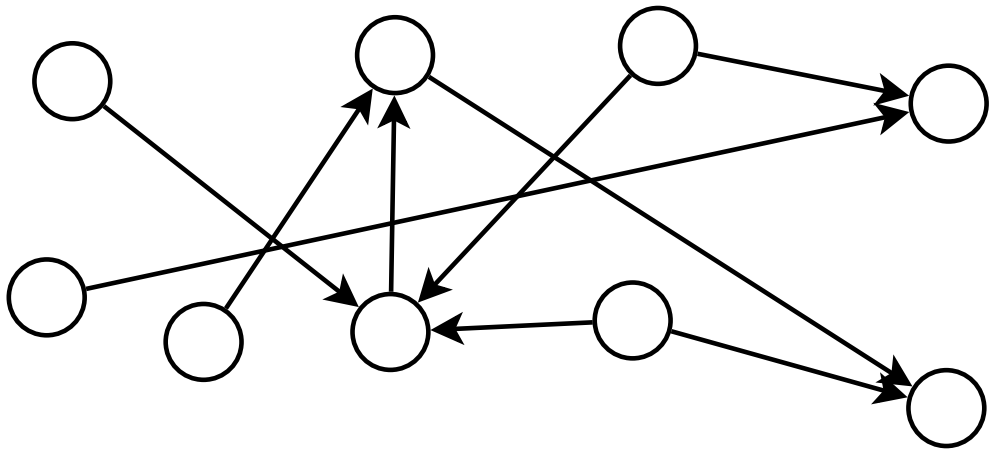

図3.2: ハイパリンクを持ったWebコンテンツ

各Authority，Hub評価値の算出は，以下のアルゴリズムによる．

各Webコンテンツpについての，Authority値(ベクトル)をA_p，Hub値(ベクトル)をH_pとする．また，隣接行列をEとする．

隣接行列Eは以下の様に定義される．Webコンテンツが持つ他のWebコンテンツに関するハイパリンクの参照被参照の情報を隣接行列として表す.

$$E = \begin{pmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & & \vdots \\ e_{n1} & \cdots & e_{nn} \end{pmatrix}$$

$$e_{ij} = \begin{cases} 1 \ if \ web \ content \ i \ points \ to \ web \ content \ j \\ 0 \ otherwise \end{cases}$$

Authority値を求めるには式3.1，Hub値を求めるには式3.2を用いる．

$$A_p = E^T H_p \tag{3.1}$$

$$H_p = E A_p \tag{3.2}$$

この2式を相互に代入すると，以下の式3.3，式3.4を求める事が出来る．

$$A_p = E^T E A_p \tag{3.3}$$

$$H_p = E E^T H_p \tag{3.4}$$

つまり，良いAuthorityは複数のHub値の高いHubにハイパリンクが張られ，良いHubは複数のAuthority値の高いAuthorityにハイパリンクを張っている，という事を(再帰的に)表現している．

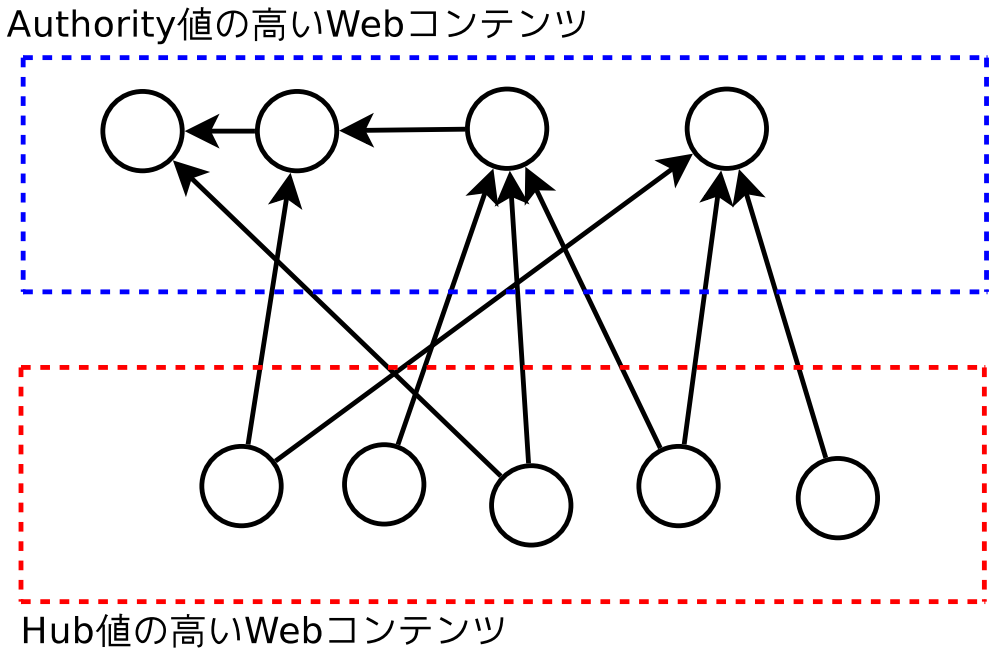

図 3.3: HITS によりクラスタリングされた Web コンテンツ

A_p, H_pが変化しなくなるまで，以下の計算を繰り返す．A_p, H_pの初期値を$(1, 1, ..., 1)^T$とする．$\|X\|_1$ は X の行列ベクトルの 1 ノルム (Norm) を表す．

$$
\begin{aligned}
A_p = H_p = (1, 1, \cdots, 1)^T & \\
do\{ & \\
& A' \leftarrow E^T H_p \\
& H' \leftarrow E A_p \\
& A_p \leftarrow A'/\|A'\|_1 \\
& H_p \leftarrow H'/\|H'\|_1 \\
\}until\|A_p + H_p\| < \varepsilon &
\end{aligned}
$$

即ち，上記アルゴリズムに於いて t 回の繰り返しが続いた場合，次の式が求められる．

$$A_p = (E^T E)^{t-1} E^T 1 \tag{3.5}$$

$$H_p = (E E^T)^t 1 \tag{3.6}$$

即ちこれは，A_p は E^TE の最大固有値に対する固有ベクトル，H_p は EE^T の最大固有値に対する固有ベクトルとなっている．

3.3.2　HITSに関する問題点

第3.3.1小節の計算から分かる様に，HITSはリンクの重みを初期値として全て同じとしている．従って，内容を見た場合，関係のあるWebコンテンツも関係の無いWebコンテンツも，高い評価値を得る事がある．これに対する対策として，ハイパリンクに重みを導入する等が考えられる．

また，HITSは，ハイパリンクを関連のあるWebコンテンツ同士を結び付けるものであると前提として定義している．しかし，実際に於いてHITSの前提と異なり，内容の関連性が低いWebコンテンツがハイパリンクに依って結び付いている事は多々ある．例えば最近のWebコンテンツにはハイパリンクの付いた広告(Advertisement)がよく張られている．Webコンテンツの内容に応じた広告が広告提供サービスに依って提供される事もあるが，それらは基本的にWebコンテンツの内容とは直接関係が無く，Webコンテンツの再利用の際に不要なものである．この様に，適切でないWebコンテンツが密なハイパリンク構造を持った場合，HITSのアルゴリズムに依り重要度が高いWebコンテンツとして評価される．これをTopic Drift問題と一般に呼ぶ．

またWebコンテンツの持つハイパリンクには様々な意味がある．それを解釈せずWebコンテンツを分類する事は，Webコンテンツの再利用に於いて不適当な結果を得，便利を享受する事が出来ない．

3.3.3　ハイパリンクによるWebコンテンツの分類に関する先行研究

村田 [13] [14] による研究では，Kumar等によるWeb Trawling [12] の手法に近似した方法で，数個のURLからそのURLを含むWebコンテンツに関する完全2部グラフ(ハイパリンク元をfans，リンク先をcentersとしている)を発見する事を目標としている．この完全2部グラフがWebコミュニティとなる．

ここでも，HITSと同様に，Webコンテンツの発見には既存の検索エンジンを用いて行っている．コミュニティの洗練のために，URLの出現回数(多数決)から，内容の関連度を測定する．1つのWebコンテンツから全てのWebコンテンツに，ハイパリンクを辿る事で到達する事はほぼ不可能なため，検索エンジンから発見出来るWebコンテンツを研究の対象にする事は，Webに関する研究では比較的よく用いられる．

3.3.4　先行研究に対する問題点

以上に挙げた先行研究には，他にWebに特有のあるレベルの問題がある．

1つは測定の誤りを導くための偽装行為である．あるWebコンテンツにハイパリンクを張るために幾つもWebコンテンツを作り，ハイパリンクを偽造するなど，SPAM行為がその測定の誤りを導く原因となる．これに対する対策は，人，または学習フィルタリングによる評価等が挙げられる．

また，そもそも，アルゴリズムが算出した測定結果に対する評価も，最終的には人間の判断により定義されるものがほとんどであり，完全にコンピュータに結果を任せる事は不可能であるとしている．

3.4　本研究での先行研究の活用

そこで本研究では，Webコンテンツの持つこのハイパリンクを分析対象とし，Webコンテンツの内容に深く踏み込まず解析を可能とする比較的単純で拡張の目安が立ち易いHITSアルゴリズムを用いて分類を行う．先行研究と異なる点は，Semantic Webの基盤技術として用いられているRDFによって表現されるメタデータ技術を用い，Webの機能を活かした，より効率的で効果的な分類を目指すという点である．

第4章

HITSの改良によるメタデータマイニングの提案

概要

本章では，第3章で解説したデータマイニングの技術を用い，Semantic Web等で使われるメタデータを含んだWebコンテンツをマイニングする為に特化した技術として，メタデータマイニング技術を提案する．

4.1　メタデータを付与したWebコンテンツ

ここに述べるメタデータマイニングで用いる対象となるデータは，Web上に存在するWebコンテンツであり，かつ，RDFによって表現されるメタデータを持っているとする．

Webコンテンツが持つ，メタデータRDF語彙の種類は以下のものとする．

- title: Webコンテンツタイトル
- creator: Webコンテンツ制作者
- publisher: 属する他のWebコンテンツへのハイパリンク
- description: Webコンテンツの内容
- created: Webコンテンツ作成日時
- updated: Webコンテンツ更新日時
- related-: 他Webコンテンツへのハイパリンクに関する情報(以下参照)

“related-” には，以下の種類がある．

- related-self: 自己のWebコンテンツを示す
- related-reference: 参考のWebコンテンツを示す
- related-common: 同一のWebコンテンツを示す
- related-postscript: 追記のWebコンテンツを示す
- related-reedit: 修正のWebコンテンツを示す

本文中ではメタデータの意味を自然言語(日本語)で示し説明している．メタデータは広義ではRDFで表現出来るもので無くても良い．しかし，メタデータの意味を共有する，つまりメタデータを利用するシステムがメタデータの意味を参照するためのアクセス方法と場所をメタデータ内に記述する事が，Web上等分散環境でメタデータを用いる場合，便利のために必要である．尚，ここで定義したメタデータはURI⟨`http://rdf.aida.k.u-tokyo.ac.jp/rdfs/webc`⟩によって参照する事が出来る．メタデータの意味を記述する上で，Dublin Core Metadata Initiative[1]が定め

[1] “Dublin Core Metadata Initiative (DCMI)”, `http://dublincore.org/`

ている，WWWのリソースや書誌情報を管理するためのRDF語彙である，Dublin Core[2]を参考にした．

データマイニングでは，複数のデータの関連を導く事が重要である，ハイパリンクの意味をメタデータにより明示する事により，後でWebコンテンツの関係をマイニングに活用する際に，適切な表現からより精度の高い結果を得られると考える．

4.2　メタデータを用いたWebコンテンツのクラスタリング

第3章で述べた，Webコンテンツの分類には限界がある事が分かっている．

そこで，本研究では提案するメタデータマイニング技術の1つとして，以上で述べたWebコンテンツに付与するためのメタデータを用いて，Webコンテンツの分類を行うための手法について述べる．

4.2.1　分類手法

クラスタリング対象であるWebコンテンツは，他のWebコンテンツに対するハイパリンクを持っているWebコンテンツとする．かつ，ハイパリンクは，第4.1節で述べたメタデータを持つものとする．

クラスタリングの結果を表示するために，あるWebコンテンツがクエリィとして与えられ，同クラスタだと判別されたWebコンテンツの集合を返すシステムを示す．

クラスタリングには，第3章第3.3.1小節で挙げた先行研究を参考にし，Webコンテンツの持つメタデータを解析して活用するために，以下のアルゴリズムを設計した．HITSと異なる点は，“Hub” や “Authority” の発見による有用なWebコンテンツの発見ではなく，Webコンテンツがどの Webコンテンツと同分類に属するかの発見である．Webコンテンツの評価はアルゴリズムに於いて目指していない．

分類の距離

まず，クラスタリングのために，Webコンテンツ間の距離を定義する．即ち，第3章で示したHITSのアルゴリズムに於いて，各Webコンテンツのハイパリンクにメタデータによる種類を導入する事で，区別，重みの情報を付与する．

HITSのアルゴリズムは，ハイパリンクの意味を解釈せず，全てリンク重みを1としている．Webコンテンツにとって，ハイパリンクによる参照には，様々な意味が

[2] “RFC 5031 The Dublin Core Metadata Element Set”, http://www.ietf.org/rfc/rfc5013

ある事は前述した通りであり，従って，分類に於いても，この意味を解釈する必要がある．

Webコンテンツへのハイパリンクには前述のメタデータと，ハイパリンク先の情報を持っている．このハイパリンク先もまた，分類解析対象のWebコンテンツであり，解析はWebコンテンツをハイパリンク経由で辿る様に実行される．従って，当然だが，Webコンテンツが1つの場合はクラスタリングは行われない．

メタデータはハイパリンクのアンカテキスト[3]よりも明確にハイパリンクの意味を示す事が出来，参照構造を抽出する上で洗練する事が出来る．

クラスタリングに名前は無く，集合への距離重み付けの大きさ(大きい程関連度が近い)によって判別される．同クラスタであると定義される距離の閾値は，メタデータによって定義した．具体的には，各ハイパリンクメタデータに対して以下の距離としての値を持つとする．

- related-self: 距離重み付け 0
- related-reference: 距離重み付け 1
- related-common: 距離重み付け 2
- related-postscript: 距離重み付け 3
- related-reedit: 距離重み付け 4

これに従い，HITSのアルゴリズムの改良を行い，Webコンテンツの分類を行う．

分類アルゴリズム

HITSのアルゴリズムは第3章第3.3.1節に示している．

隣接行列Eについて，全て1としていた係数を，上記の距離重み付けを導入する．つまり，改良したHITSアルゴリズムは，式3.1，式3.2，式3.3，式3.4，式3.5，式3.6と合わせて，以下の様になる．

$$e_{ij} = \begin{cases} 0\ if\ i\ points\ to\ j\ as\ self\ or\ otherwise \\ 1\ if\ i\ points\ to\ j\ as\ reference \\ 2\ if\ i\ points\ to\ j\ as\ common \\ 3\ if\ i\ points\ to\ j\ as\ postscript \\ 4\ if\ i\ points\ to\ j\ as\ reedit \end{cases}$$

各Webコンテンツpについての，Authority値(ベクトル)をA_p，Hub値(ベクトル)をH_pとする．以下のアルゴリズムは，改良無しHITSのアルゴリズムから変更していない．

[3] HTMLに於けるaタグで囲まれたテキスト部

$$
\begin{aligned}
A_p = H_p = (1, 1, \cdots, 1)^T \\
do\{ \\
A' \leftarrow E^T H_p \\
H' \leftarrow E A_p \\
A_p \leftarrow A' / \|A'\|_1 \\
H_p \leftarrow H' / \|H'\|_1 \\
\}until \|A_p + H_p\| < \varepsilon
\end{aligned}
$$

以下でこのアルゴリズムを用いた実験とその結果を示す.

4.2.2　シミュレーションによる分類の実験

ここでは，シミュレーション上で仮想のWebを構築し，分類の効果を測定するための実験を行う．即ちシミュレーション上でWebコンテンツと想定するノードを100個生成し，ランダムにハイパリンクを生成する．結果は，HITSに従い，Authority値とHub値で表現する．

実験のためのシミュレーションプログラムの実装には，付録Cで示した環境を用いた．

まず，隣接行列からランダムに2ノードを選出し，またランダムにハイパリンクメタデータの種類を選出し，隣接行列を書き換える．これを1000回繰り返す.

ランダムに生成したWebコンテンツノードとハイパリンクの配置例を，図4.1に示す.

また，オリジナルのHITSとの結果の比較のため，ハイパリンクの重みを全て1とした隣接行列も併せて生成する．これは完全なランダムWebを生成して提案したアルゴリズムを適用した実験である．

プログラムは，付録Dリスト D.1に示している．

まず，シミュレーション上で，これまでの改良を加えないHITSアルゴリズムを用いた場合の，生成WebコンテンツノードのAuthority値(付録A表A.1)とHub値(付録A表A.2)を示す.

次に，同シミュレーション上で，改良を加えたHITSアルゴリズムを用いた場合の，生成WebコンテンツノードのAuthority値(付録A表A.3)とHub値(付録A表A.4)を示す.

HITSアルゴリズムと提案した改良HITSアルゴリズムの比較を行うために，各ノードのAuthority値とHub値を比較する．単純に，各Webコンテンツノードの

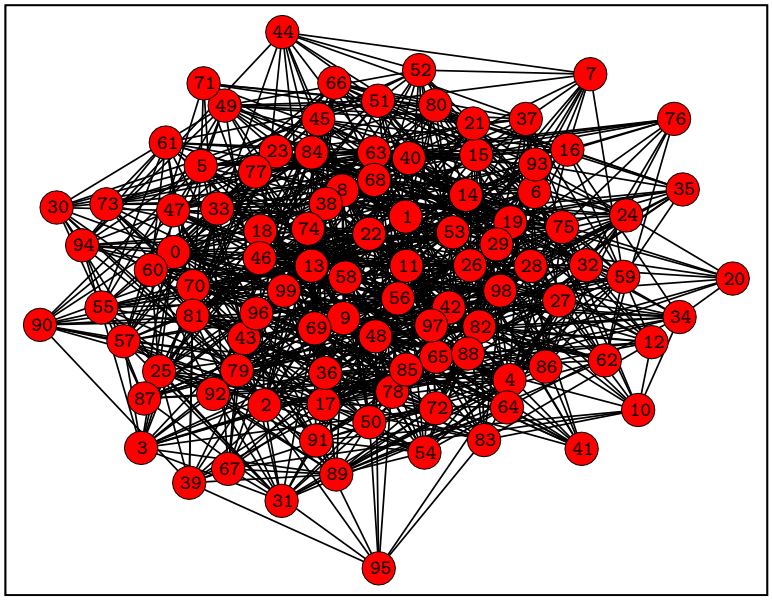

図 4.1: ランダムに生成した Web コンテンツノードとハイパリンク例

Authority 値と Hub 値それぞれについて，改良 HITS アルゴリズムによるものから改良無し HITS アルゴリズムとの差を算出する．比較を見易くするために，各値を 100 倍して示す (付録 A 表 A.5，付録 A 表 A.6).

この比較から分かる項目として以下の点が挙げられる．

- 殆どの Web コンテンツノードはそれが持つ Authority 値と Hub 値の比較差の符号が異なっている
  つまり，Authority 値が大きくなれば Hub 値は小さくなり (または Authority 値が小さくなれば Hub 値は大きくなり)，Hub 値が大きくなれば Authority 値は小さくなる (または Hub 値が小さくなれば Authority 値は大きくなる).
- Authority 値の比較差の絶対値が Hub 値の比較差の絶対値の 2 倍から 4 倍 (もしくはそれ以上)，または Hub 値の比較差の絶対値が AUthority 値の比較差の絶対値の 2 倍から 4 倍 (もしくはそれ以上) の値を示している Web コンテンツノードが多い
  これは，改良 HITS アルゴリズムでは，ハイパリンクにメタデータを付与し，各メタデータに関して，ハイパリンクに 2 から 4 の重み付けを行ったためであると考えられる．これにより，各 Web コンテンツの Authority 値と Hub 値の幅が大きくなり，分類の閾値を定め易くなると考えられる．

このシミュレーションは，完全にランダムに，Webコンテンツのハイパリンクを生成した．その様なWebに於いても，この提案した改良HITSアルゴリズムは，コミュニティを発見する上で有用である事が確認出来る．しかし，現実のWebに於いては，大きいAuthority値を持つWebコンテンツは大きいAuthority値を維持しようとし，また，大きいHub値を持つWebコンテンツは大きいHub値を維持しようとする．このWebコンテンツの成熟がWebコミュニティを生成し構築する．Webコミュニティに於いて，Hubの機能を持つWebコンテンツの存在はコミュニティの基盤となるものである．つまり，Webコンテンツの性格によって，Webコンテンツは目指す方向へと発達し，Webコミュニティが成熟する．従って，ランダムなハイパリンクは現実的とは言えず，実際のハイパリンクは，ある時点での各値に基づき，ハイパリンクを作為的に調整する必要がある．

第5章

メタデータマイニング応用システムの提案: WebコンテンツPush配信

概要

本章では，第4章で解説したメタデータマイニングを活用した事例として，Webコンテンツの Push配信システムを提案する．

5.1　Push配信の意義

昨今，雑誌等紙媒体の衰退に関する報道をよく目にする．パソコンや携帯電話によるWebの活用の浸透や電子メディアを利用するための環境の普及等に伴い，多くの人達が電子メディアに比べ，保存コストが掛かり耐久度の低い雑誌を必要としなくなり，多くの雑誌が休刊，あるいは廃刊に追い込まれているという．

雑誌等紙媒体が，最早絶滅の道を歩むであろう事は明白だと考えているが，紙というハードウェア制約を取り除いて，雑誌というメディアを見直した場合，現在のWebに無い利点，あるいは魅力が見て取れる．

例えば，雑誌等に掲載された広告等は一般に，雑誌を見るであろう者達，購読者層の特徴を考慮し，有意義であると考えている，それは必ずしも購読者にとって不可欠なものでは無いにしろ，何らかの利益をもたらす事も想定している．つまり興味を持つであろうカテゴリィの商品やサービスの広告を提供する．例えば，読者が雑誌等に掲載され，かつ自身が知り得ていない情報を発見した時，得をしたと感じた事も少なからずあるだろう．

また，雑誌というのは一般的に，媒体やメディアに分割されている同じカテゴリィに属するテーマを集約し，提供する役割を果たす．つまり分類を行っている．従って，購読者は雑誌に掲載されたコンテンツを，横断的に閲覧し，仮に集約されていない状態で提供された時に見落とす可能性のある情報も，集約されているという前提の下で，時間を節約して，取捨選択する事が出来る．

これらの雑誌等が持つ特徴は，主に雑誌を作る者達，つまり編集者の手に依るところが大きい．Webでも，あるWebサイト管理者による，雑誌の様な，複数のWebコンテンツを集約して，提供するサービスがあるが，Webという環境を活かす上で，コンピュータ，プログラムによる自動化が出来ていないというのが現状である．

また序でに，分類について少し言及する．生物学の基礎の一分野として分類学がある．即ち，生物の特徴から生物の種類を分類し，体系的に理解する事を目的とした学問である．初期の分類学に於いては，人為分類という人間の生活上，思考上に都合が良く，実用的に便利な分類が行われた．即ち人間の認知に依存した分類であった．また教育の分野でも，生徒に何らかのトピックの分類を行わせる事には，情報教育の発展のために効果的であるという考えがある．川喜多二郎の『発想法』やマインドマップなども，この分類による人間の認知能力の向上を狙ったものであ

ると考えられる．即ち，分類とは，人間の情報処理能力を向上させるために，効果的であり，例えばWeb等，多種多様なコンテンツが存在している場に於いて，コンピュータの分類の支援は，必ず人間のコンテンツ利用の効率の向上に貢献すると考える事が出来る．

そこで本研究では，雑誌等の様に，Webコンテンツを集約し，提供するシステムを開発する．そのシステムでは，Webコンテンツに付与されたメタデータを活用し，より精度の高い結果を目指す.

5.2　実装

システムは，解析部，提供部の2つに機能を分割する事が出来る．

まず，解析部ではメタデータを付与されたWebコンテンツを解析し，第4章で挙げた手法を用いて分類を行う．ユーザは1つ以上のWebコンテンツを利用対象とし，解析部に於いて，そのWebコンテンツに同分類とする複数のWebコンテンツをユーザに，提供部によって提供する．

尚，第4章で述べた解析可能なメタデータが付与されているWebコンテンツが存在している事を前提とし，以下，Webコンテンツとは全てメタデータを持っているものであるとする．この前提に関する問題点は，後の今後の課題を述べる第6章で言及する．

システムの利用者は，ハイパリンクを持つWebコンテンツを生成あるいは取得する．そのWebコンテンツを解析部に渡す．解析部では，受け取ったWebコンテンツから張られたハイパリンクを辿り，隣接Webコンテンツをn個取得する事を，探索と再帰を繰り返す事で行う．そして，第4章で提案した改良HITS アルゴリズムを用い，Webコンテンツを分類する．

提供部では，分類されたWebコンテンツの集合から，Authority値の高いWebコンテンツ，Authority値の低いWebコンテンツ，及びHub値の高いWebコンテンツをシステムの利用者に提供する．ここで，Authority値が低いWebコンテンツを提供する理由は，一般に，Webコンテンツが生成，流通した直後の段階に於いてAuthority値は低いが，Authority値あるいはHub値の高いWebコンテンツを参照しており，早い段階で利用者に提供する事は有意義であると考えている．

実装案を明確に表すために，ここでも，第4章第4.2節で作成したシミュレーション上で，仮想のWebを構築し，その実験を行う.

まず，メタデータを付与したWebコンテンツの例を2つ示す(図5.1，図5.2)．これらのWebコンテンツは共に，RDFの構造に従ったメタデータなので，今回の解析の対象になる事が出来る．

既に，Webコンテンツが存在しているものとし，利用者は1つのWebコンテンツ

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:webc='http://rdf.aida.k.u-tokyo.ac.jp/rdfs/webc#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
>
...
 <rdf:Description>
  <webc:related-reference>http://example.com/3</webc:related-reference>
 </rdf:Description>
...
</rdf:RDF>
```

図5.1: メタデータを付与したWebコンテンツ例1(RDF/XML)

を解析部に渡す．そして，そのWebコンテンツを改良HITSアルゴリズムに従って解析し，求められたAuthority値とHub値から，提供部によって関連したWebコンテンツの集約と配信を行う．

ここで再び第4章と同じく，シミュレーションを用いて生成したWebコンテンツの集合に於ける，利用者側とシステム側の配信までの流れを示す．使用したシミュレーションプログラムは付録Dリスト D.2に示している．

仮に，システムは100個のWebコンテンツを把握し管理しているものとする．

利用者から解析部に渡されたWebコンテンツを，解析部は101個目のWebコンテンツとして解釈し，そのメタデータから改良HITSアルゴリズムを適応する．

利用者から渡されたWebコンテンツがHubとして機能した場合(図5.3)，Authorityとして機能した場合(図5.4)について，それぞれAuthority値とHub値を示す(付録B表B.1，表B.2，及び付録B表B.3，表B.4)．

システムの流れを図5.5に示す．

このシミュレーションに於けるシステムの問題点として，Webコンテンツの存在が全てシステムに既知となっており，1回の解析で全てのWebコンテンツに対して処理を行っているという点がある．Webコンテンツの数が数百億個であると調査されている現実のWebに於いて，この様なアルゴリズムを適用するのは適切でない．従って，Webコンテンツの数が限られている段階で解析を行い，予めある程度の解析による分類を行い，利用者によって与えられたWebコンテンツと共に解析するWebコンテンツを最小限にする等の対策が必要となる．

```
<html>
...
<div
  xmlns:webc='http://rdf.aida.k.u-tokyo.ac.jp/rdfs/webc#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
>
 ...
 <a href="http://example.com/3" rel="webc:related-reference">参考</a>
 ...
</div>
...
</html>
```

図 5.2: メタデータを付与した Web コンテンツ例 2(RDFa)

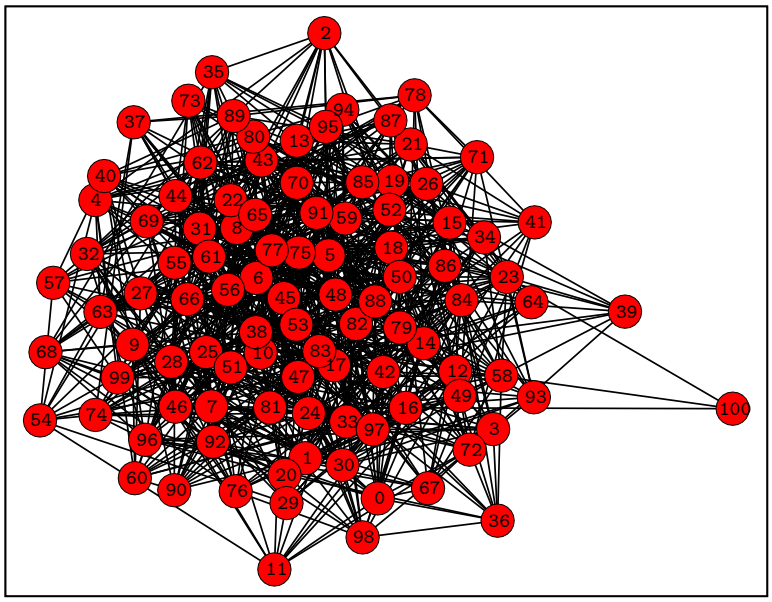

図 5.3: 利用者からの Web コンテンツが Hub として機能した例

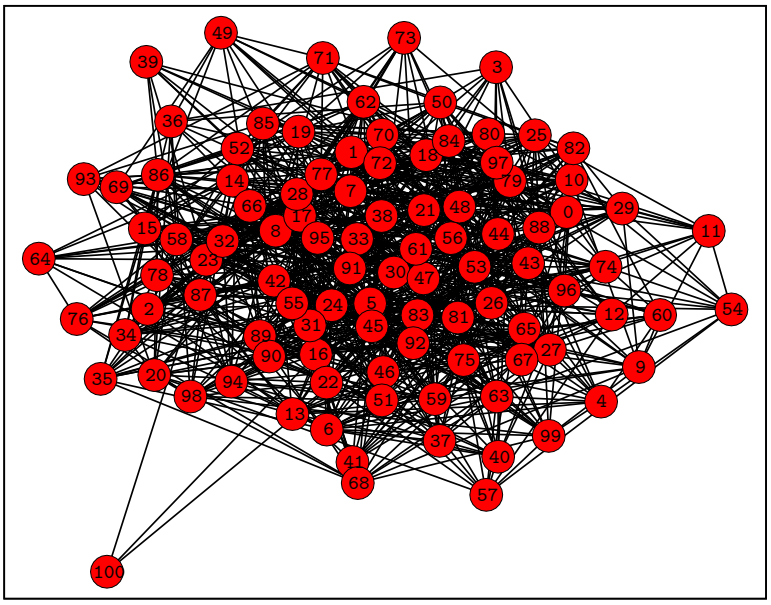

図 5.4: 利用者からの Web コンテンツが Authority として機能した例

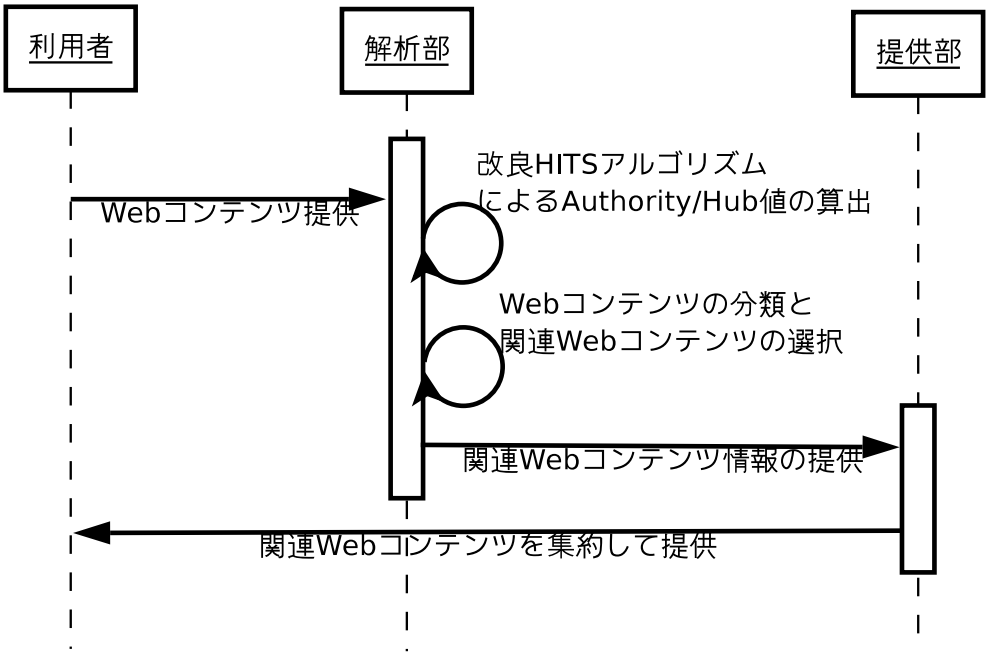

図 5.5: Push 配信システムの流れ

第6章

評価と考察

概要

本章では，第4章，第5章で提案，実装した技術について評価，考察を述べる．

6.1　評価

今回の実験は，全て作成したシミュレーション上で行った．多くの先行研究では，既存の検索エンジンやWebコンテンツを収集するために開発したクローラを用い，研究の評価をしている．

しかし，本研究の対象は現実に存在する実Webコンテンツではなく，メタデータの付与を必須とした，現在のWebコンテンツに置き換わるものであり，シミュレーション上での実験が適切だと考えた．また，数値のみではあるものの，既存のアルゴリズムとAuthority値とHub値を適切に比較する事で，提案したメタデータマイニングの効果を示す事が出来た．

6.2　考察

本研究では，ハイパリンクにRDFによって表現されるメタデータを付加するという行為のみで，ハイパリンクを持つWebコンテンツの属する分類を，HITSのアルゴリズムを用いて，Authority値とHub値を活用する事により判断出来る事を示した．

RDFを採用した事で，メタデータの意味は本研究で提案したものに止まらない．URI⟨`http://rdf.aida.k.u-tokyo.ac.jp/rdfs/webc`⟩で公開しているメタデータの意味は，適切な方法で誰でも参照可能であると同時に，メタデータの語彙を追加したり，意味を変更して，別の場所に公開する事で異なるメタデータマイニングの結果を導く事も可能となる．

現在広くWeb上で普及している検索エンジンの多くは，Webコンテンツの内容をハイパリンクを含めて文書の内容を取得，形態素解析等をして，単語の関係をWebコンテンツの関連の解析に用いている．つまり，人間が見るものをコンピュータが進化する事で，コンピュータが理解に努めるという手法を取っている．勿論，この手法が進化すれば，本研究より優れた結果を導く事も可能かも知れない．

しかし，本研究で述べた様に，Webコンテンツの解析に必ずしも，Webコンテンツの内容，つまり人間が読むデータを，そのままで理解出来ないコンピュータが読む必要は無い．メタデータというコンピュータが読むためのデータを生成し，かつメタデータの意味を参照可能の方法で指定する事により，人間とコンピュータが読むデータを分離する事(図6.1)が，今後，Webコンテンツ利用環境の効率を向上さ

せるためにも，Semantic Web の普及を推進させるためにも，必要不可欠な方針であると考える．

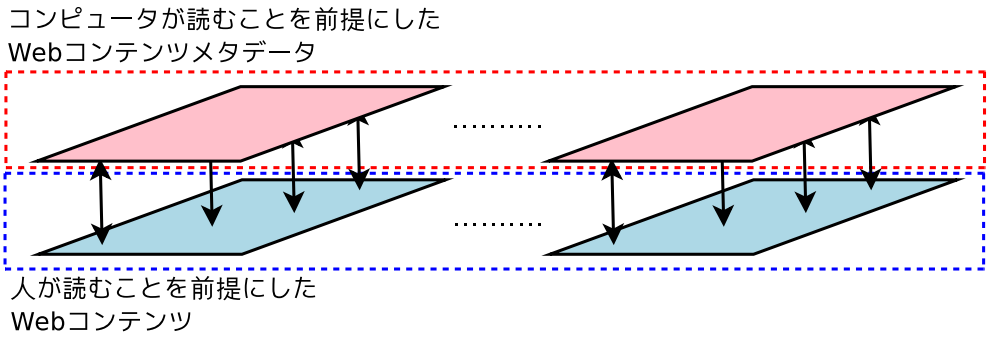

図 6.1: メタデータと Web コンテンツ

第7章

結論

概要

本章では，以上の章で述べた提案に対し，結論を述べる．

7.1　結論

以上に於いて，本論文では現在のWebの問題点から新しいWebとしてのSemantic Webについての説明，Semantic Webの基盤技術であるメタデータ技術，RDFについて解説した．

7.2　今後の課題

今回は，ハイパリンクの持つメタデータのみを解析の対象とし，解析と分類を行った．メタデータの種類も，ここで想定する5種類に関する重み付けしかしていない．しかし，メタデータはその語彙と意味を誰でも自由に記述し，共有する事が出来る．Webとはそういう場である．従って，今回述べたメタデータの解析は，1つの結果であって，メタデータの種類，あるいは解析アルゴリズムが異なる事で違う結果を見せる事もあり得る．

また，先行研究の他のWebコンテンツの分類手法として，ハイパリンクのアンカテキスト(自然言語)を解析し，その単語の関連や頻出度から分類を行う方法がある．また，Googleで用いられているPageRankは，URIに含まれる文字列，(X)HTMLに於けるタグの重要度を考慮し，各検索語(クエリィ)に対してランキング評価という尺度でWebコンテンツの分類を行っている．

WebコンテンツPush配信については，現在のWebコンテンツ生成システム(Weblog，Wiki等)には，このメタデータを正しく生成するための仕組みはほぼ無い．従って，このシステムを利用するためには，メタデータを記述するための支援を行うための仕組み，例えば，エディタや自動生成の仕組みが求められる．

今後は，これらの技術を併せて利用し，より精度の高い分類が行われる事で，Webコンテンツ利用者の利用効率向上を促進させる事が出来ると考える．

謝辞

本研究を進めるにあたり，多くの方々のご協力をいただきました．

相田仁教授には，修士課程に入学し相田研究室に配属して以来，研究に関する意見や有益なアイデアを幾つも頂きました．また，研究を行う上でも研究室生活を過ごす上でも非常に素晴らしい環境を与えていただきました．ここに心から御礼申し上げます．

また，日々の研究室生活に於いて様々な面で何度もお世話になりました，技術専門職員の千葉新吾氏，助教の藤枝俊輔氏，秘書の中山早百合氏に感謝致します．

そして，同期のアピラックウィリヤ・ウィッタヤー氏，杉谷心氏，藤原直弘氏には，日々の研究室生活や研究を行う上で，互いに切磋琢磨し，時に支え合い，時に励まし合いながら，2年間を過ごす事が出来た事を嬉しく思います．

最後に，研究室の皆様と，遠方ながら様々な時点で心の支えになった家族に感謝します．

2008年1月29日

参考文献

[1] Pierre Baldi, Paolo Frasconi, Padhraic Smyth, 水田正弘, 南弘征, 小宮由里子: “確率モデルによるWebデータ解析法”, 森北出版, 2007/05.

[2] Tim Berners-Lee: “Challenges of the second decade”, http://www.w3.org/Talks/1999/05/www8-tbl/, 1999/05, Accessed: 2008/01.

[3] Tim Berners-Lee: “Artificial Intelligence and the Semantic Web”, http://www.w3.org/2006/Talks/0718-aaai-tbl/, 2006/07, Accessed: 2008/01.

[4] Sergey Brin, Lawrence Page: “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, http://infolab.stanford.edu/~backrub/google.html, 1998, Accessed: 2008/01.

[5] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, Janet Wiener, “Graph structure in the web”, http://www9.org/w9cdrom/160/160.html, 2000/5

[6] Gene Garfield, Chairman Emeritus: “THE USE OF JOURNAL IMPACT FACTORS AND CITATION ANALYSIS FOR EVALUATION OF SCIENCE”, 1998/04.

[7] IBM, “AltaVista, Compaq and IBM Researchers Create World’s Largest, Most Accurate Picture of the Web - United States” http://www-03.ibm.com/press/us/en/pressrelease/1733.wss, 2000/5

[8] 石川博: “次世代データベースとデータマイニング”, CQ出版社, 2005/5.

[9] ITU: “International Telecommunication Union - BDT”, http://www.itu.int/ITU-D/icteye/Indicators/Indicators.aspx, 2007/08, Accessed: 2008/01.

[10] 加藤真, 山名早人, “Fact of the Web –30億ページのウェブの解析–”, DEWS 3B-i6, http://www.db.soc.i.kyoto-u.ac.jp/DEWS2006/doc/3B-i6.pdf, March 2006

[11] Jon M. Kleinberg: “Authoritative Sources in a Hyperlinked Environment”, http://www.cs.cornell.edu/home/kleinber/auth.pdf, 1998, Accessed: 2008/01.

[12] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, “Trawling the Web for Emerging Cyber-Communities”, 8th International World Wide. Web Conference, http://www8.org/w8-papers/4a-search-mining/trawling/trawling.html, 1999.

[13] 村田剛志: “参照の共起性に基づくWebコミュニティの発見”, 人工知能学会論文誌 16巻3号B pp.316-323, 2001.

[14] 村田剛志: “ハイパーリンクのグラフ構造に基づくWebコミュニティの洗練”, 人工知能学会論文誌 17巻3号SP-I pp.322-329, 2002.

[15] Per O Seglen: “Why the impact factor of journals should not be used for evaluating research”, 1997/02.

[16] 総務省: “総務省：情報通信統計データベース：情報通信白書”, http://www.johotsusintokei.soumu.go.jp/whitepaper/ja/cover/index.htm, 2007/07, Accessed: 2008/01.

[17] W3C: “RDF Test Cases”, http://www.w3.org/TR/rdf-testcases/, 2004/02.

[18] W3C: “RDF/XML Syntax Specification (Revised)”, http://www.w3.org/TR/rdf-syntax-grammar/, 2004/02, Accessed: 2008/01.

[19] W3C: “RDFa Primer 1.0”, http://www.w3.org/TR/xhtml-rdfa-primer/, 2007/03, Accessed: 2008/01.

[20] W3C: “Resource Description Framework (RDF) / W3C Semantic Web Activity”, http://www.w3.org/RDF/, 2007/1, Accessed: 2008/01.

[21] Stanley Wasserman, Katherine Faust: “Social Network Analysis: Methods and Applications”, 1994.

発表文献

[22] 佐伯嘉康, 相田仁: “セマンティック・ウェブ技術を用いたWebコンテンツ配信”, 情報処理学会第69回全国大会, 第69回(平成19年)全国大会 講演論文集(分冊1) pp.441-442(2S-3), 2007/03.

[23] 佐伯嘉康, 相田仁: “RDFaによるメタデータを活用したWebコンテンツ配信技術”, FIT2007第6回情報科学技術フォーラム, FIT2007第6回情報科学技術フォーラム 一般講演論文集第1分冊 pp.159-160(B-034), 2007/09.

付録 A

シミュレーション上での Web コンテンツの Authority 値/Hub 値

第 4 章第 4.2.2 小節で実験をし測定した，ランダムな Web コンテンツノードの Authority 値及び Hub 値を示す.

4 つの表はそれぞれの 1 つのセルが 1 つの Web コンテンツノードに対応しており，その Web コンテンツノードが持つ値を示している．順序は関係が無い.

表 A.1: 改良無し HITS アルゴリズムによる測定 (Authority 値)

0.01226638	0.00590034	0.00962637	0.0064064	0.01363476
0.01144131	0.0117308	0.00868902	0.01126576	0.01227569
0.0041151	0.0121785	0.00635308	0.02873724	0.00534397
0.01231196	0.0134811	0.01343407	0.01329727	0.01115332
0.00680683	0.00891011	0.01374922	0.01376323	0.00591868
0.00395906	0.00872843	0.0081948	0.01549995	0.01279338
0.00879323	0.00916984	0.01065846	0.01431937	0.00536017
0.00876681	0.00551607	0.01475499	0.01615065	0.00738988
0.0079845	0.00759662	0.01303409	0.00839608	0.00822598
0.00776165	0.01428184	0.01008377	0.00657265	0.00680001
0.01006357	0.01184268	0.00657047	0.01132617	0.01200649
0.0052381	0.01921105	0.00824515	0.00680576	0.00647539
0.00811789	0.00864527	0.00864298	0.01346907	0.00797354
0.01545677	0.01296935	0.00595596	0.00954484	0.01658741
0.00805776	0.00372529	0.00908991	0.00429712	0.01351314
0.01366595	0.00467676	0.01011544	0.00817024	0.01084939
0.0106452	0.01363493	0.01961261	0.00607383	0.01353241
0.01228972	0.00441953	0.00379573	0.01075304	0.00720541
0.00751123	0.00878799	0.01136767	0.01028633	0.00487296
0.00418326	0.01234341	0.00920275	0.01444168	0.01014865

表 A.2: 改良無し HITS アルゴリズムによる測定 (Hub 値)

0.01253162	0.01590761	0.00679611	0.01121917	0.00914347
0.00988482	0.01200896	0.0044786	0.01023423	0.01326717
0.00727071	0.01534411	0.00736798	0.01385742	0.01443215
0.01031313	0.00684814	0.00659188	0.00712316	0.00590252
0.00346328	0.01168274	0.01157261	0.00290425	0.01231605
0.01391453	0.01296887	0.01848895	0.00969331	0.01602503
0.00447645	0.01048789	0.0068932	0.00879972	0.01044262
0.00468132	0.01908408	0.01133282	0.0043223	0.00512867
0.00933098	0.00703927	0.01478021	0.00967771	0.00908522
0.01189836	0.01346995	0.00460864	0.0116249	0.01473776
0.00900269	0.00782324	0.01178346	0.01090412	0.00886132
0.01009915	0.01348645	0.01189159	0.00799588	0.01444172
0.01145806	0.00647259	0.00713937	0.01556039	0.01051471
0.00736237	0.01082325	0.00354229	0.00698172	0.01472469
0.00721227	0.00698143	0.01077917	0.00772601	0.01840989
0.00795747	0.00676703	0.00934527	0.01265102	0.00823312
0.01053737	0.01075934	0.0114482	0.00904933	0.01262831
0.01141184	0.01437895	0.01428763	0.01160207	0.01016891
0.0085863	0.00731713	0.00653518	0.00928755	0.00977655
0.00403967	0.00650617	0.01155639	0.00805457	0.0096821

表 A.3: 改良 HITS アルゴリズムによる測定 (Authority 値)

0.01411913	0.00645898	0.01253178	0.0050794	0.01017087
0.01505069	0.01162461	0.00835375	0.00790509	0.01286639
0.0056479	0.01078605	0.00483596	0.0347845	0.00668803
0.00897965	0.01454582	0.01480931	0.01357043	0.01097467
0.00694178	0.00570649	0.01483464	0.01162468	0.00600732
0.00516025	0.00770666	0.00572353	0.01798656	0.01180406
0.00989965	0.01075421	0.01294843	0.01377935	0.00631665
0.00910119	0.00505623	0.01674628	0.01768744	0.00711832
0.00996421	0.00796115	0.01090966	0.00616983	0.00964061
0.00603705	0.01158899	0.0131944	0.00862379	0.00652229
0.00955292	0.00953246	0.00919757	0.00893674	0.00979288
0.0061667	0.02144953	0.00683306	0.00673163	0.00450167
0.01012772	0.00981067	0.00883487	0.01386497	0.01082324
0.01450113	0.01230555	0.00451643	0.00807867	0.01543923
0.00897787	0.00439123	0.00813613	0.00392471	0.01128645
0.01515578	0.00471264	0.00844762	0.00853278	0.0080105
0.00551087	0.01290445	0.01818271	0.00684441	0.01458267
0.01461288	0.00352583	0.00188123	0.01262608	0.00549118
0.00867887	0.01045837	0.01350281	0.00820092	0.00439399
0.00300875	0.01279355	0.0098781	0.01722432	0.00982592

表 A.4: 改良 HITS アルゴリズムによる測定 (Hub 値)

0.00944888	0.01485219	0.00501414	0.01085561	0.00978106
0.01213363	0.01364813	0.00716838	0.00586013	0.00866625
0.00427947	0.01757602	0.00663831	0.00961418	0.01420279
0.00957433	0.00647888	0.00781908	0.00636965	0.00579585
0.00371354	0.01018099	0.01588226	0.00159947	0.00993628
0.0137171	0.01643594	0.01915976	0.00854732	0.01667911
0.00412388	0.0103244	0.00894622	0.01109676	0.01446884
0.0035807	0.01600033	0.01040532	0.00544694	0.00485301
0.00800949	0.00663685	0.01460106	0.01043683	0.00813339
0.00982241	0.01764392	0.00290563	0.00827841	0.01916599
0.00572544	0.00970667	0.01357101	0.01000139	0.00928309
0.00825211	0.00808418	0.01338816	0.00456792	0.01465481
0.0157369	0.00936571	0.00800176	0.01691398	0.01249507
0.00738703	0.00806766	0.00189012	0.00897662	0.01703688
0.00521183	0.00634803	0.01393672	0.00580087	0.01586476
0.00472272	0.00694008	0.0100377	0.01568603	0.00490947
0.0090856	0.00998438	0.01283519	0.00681981	0.01558551
0.01079207	0.01777508	0.01051306	0.01359582	0.00753146
0.01055614	0.01007112	0.00744402	0.01105537	0.01550165
0.00421307	0.00847666	0.01081735	0.00625362	0.0120192

表 A.5: 改良無し HITS アルゴリズムと改良 HITS アルゴリズムの値の比較 (Authority 値)

0.18527551	0.05586445	0.29054132	-0.13270054	-0.34638941
0.3609383	-0.01061928	-0.03352717	-0.33606674	0.05906963
0.15328006	-0.1392451	-0.1517118	0.60472614	0.13440663
-0.33323041	0.1064724	0.13752401	0.02731677	-0.01786502
0.0134953	-0.32036135	0.10854269	-0.21385521	0.00886346
0.12011897	-0.10217673	-0.24712715	0.24866118	-0.09893138
0.11064196	0.15843675	0.22899693	-0.05400216	0.09564817
0.03343713	-0.04598399	0.19912915	0.15367837	-0.02715564
0.1979719	0.03645303	-0.21244328	-0.22262505	0.14146278
-0.17245923	-0.26928467	0.31106299	0.20511352	-0.02777161
-0.05106491	-0.23102218	0.26271066	-0.23894322	-0.22136039
0.09285981	0.22384748	-0.14120903	-0.00741268	-0.19737247
0.20098309	0.11654043	0.01918933	0.03958965	0.28497001
-0.09556355	-0.06638024	-0.14395235	-0.14661721	-0.11481787
0.09201115	0.06659421	-0.09537807	-0.03724134	-0.22266909
0.14898315	0.00358868	-0.16678202	0.03625479	-0.28388912
-0.51343226	-0.07304788	-0.14299051	0.07705825	0.10502526
0.23231561	-0.0893702	-0.19145016	0.18730456	-0.17142316
0.11676425	0.16703793	0.21351415	-0.2085408	-0.04789663
-0.11745115	0.04501346	0.06753458	0.27826415	-0.03227267

表 A.6: 改良無し HITS アルゴリズムと改良 HITS アルゴリズムの値の比較 (Hub 値)

-0.30827356	-0.10554135	-0.17819755	-0.03635641	0.0637582
0.22488092	0.16391659	0.26897818	-0.43741045	-0.46009178
-0.29912475	0.22319102	-0.07296645	-0.424324	-0.02293557
-0.07388012	-0.03692608	0.12272017	-0.07535068	-0.01066758
0.0250258	-0.15017584	0.43096559	-0.13047826	-0.23797714
-0.0197427	0.34670733	0.06708175	-0.11459883	0.06540816
-0.03525726	-0.01634842	0.20530202	0.22970406	0.40262111
-0.11006244	-0.30837459	-0.09274945	0.1124639	-0.02756616
-0.13214872	-0.04024227	-0.01791515	0.07591202	-0.09518317
-0.20759535	0.41739672	-0.1703009	-0.33464862	0.44282293
-0.32772527	0.18834219	0.17875503	-0.09027308	0.0421772
-0.18470439	-0.5402275	0.14965747	-0.3427968	0.021309
0.42788418	0.28931191	0.08623836	0.13535931	0.19803572
0.00246626	-0.27555905	-0.16521695	0.19948989	0.23121827
-0.20004407	-0.06333958	0.31575568	-0.19251379	-0.25451229
-0.3234754	0.01730447	0.06924285	0.3035007	-0.33236528
-0.14517696	-0.07749557	0.13869884	-0.22295197	0.29572049
-0.06197751	0.33961284	-0.37745725	0.19937532	-0.26374455
0.19698388	0.27539864	0.09088387	0.17678207	0.57250941
0.01733963	0.19704887	-0.0739046	-0.18009482	0.23370945

付録B

Push配信シミュレーション上での Webコンテンツの Authority値/Hub値

第5章第5.2節で実験をし測定した，WebコンテンツノードのAuthority値及びHub値を示す.

4つの表はそれぞれの1つのセルが1つのWebコンテンツノードに対応しており，そのWebコンテンツノードが持つ値を示している．順序は関係が無い.

表B.1: 利用者のWebコンテンツがHubとして機能した場合(Authority値)

0.00646427	0.00827398	0.00765148	0.00515145	0.0042879	
0.02367755	0.01611377	0.0080609	0.01167006	0.00602162	
0.01055942	0.00800856	0.01067154	0.00481681	0.00829101	
0.01175122	0.02897922	0.01339635	0.00803137	0.0078599	
0.00791062	0.00964894	0.00928068	0.0079236	0.01401214	
0.01630116	0.00749897	0.01719174	0.01013189	0.01146576	
0.01402613	0.0135774	0.01198401	0.00897345	0.00734986	
0.0079411	0.00638854	0.00439166	0.00443039	0.00221997	
0.00639627	0.0037944	0.01342772	0.01053057	0.0093033	
0.01463311	0.01080057	0.01519564	0.02104276	0.00452312	
0.01239902	0.01633574	0.01465259	0.01828675	0.0038137	
0.01265186	0.02036472	0.00501474	0.00777866	0.0107784	
0.00223527	0.01439987	0.01062351	0.01266353	0.00301984	
0.01053617	0.01332518	0.00545855	0.0059578	0.0098342	
0.01030473	0.0133544	0.01122211	0.00617866	0.00494489	
0.01440604	0.00617361	0.00485286	0.00407701	0.01195539	
0.0124959	0.01081719	0.01484632	0.01442577	0.01131926	
0.00510249	0.01268565	0.00672013	0.00571429	0.0120447	
0.00454513	0.00822175	0.00934001	0.00432068	0.00788043	
0.00675647	0.01400016	0.01244936	0.0020483	0.01463236	0

表B.2: 利用者のWebコンテンツがHubとして機能した場合(Hub値)

0.00648446	0.00801088	0.00820658	0.00568581	0.00975965	
0.00964071	0.00930283	0.01482608	0.02288017	0.00733055	
0.00896151	0.00788778	0.00472008	0.01007785	0.01409511	
0.00855254	0.00324596	0.00748685	0.00808047	0.01233089	
0.00728691	0.00419557	0.00991365	0.01080954	0.01205778	
0.00306217	0.01638385	0.00639181	0.01223257	0.01165403	
0.00983522	0.00658106	0.00889587	0.01015072	0.00941165	
0.00627186	0.0065961	0.01263975	0.02223314	0.00631735	
0.00905009	0.00909869	0.01078129	0.01629804	0.01330812	
0.01366395	0.0149378	0.00794136	0.00873454	0.00476483	
0.01097319	0.00773002	0.00328351	0.00778199	0.00497743	
0.00792012	0.01157758	0.00693152	0.01163071	0.01338069	
0.01073133	0.01166259	0.01188451	0.00988929	0.00821756	
0.01470906	0.01160679	0.00801348	0.0066617	0.00681694	
0.01166881	0.00852471	0.00419848	0.00712379	0.01129864	
0.01031653	0.00644822	0.01762149	0.00810648	0.00794786	
0.00859667	0.01554991	0.00747957	0.01581708	0.01489334	
0.00947679	0.00875398	0.00998654	0.01803647	0.00565891	
0.006657	0.01655055	0.01314866	0.00720515	0.00740512	
0.01656837	0.00817555	0.01906964	0.01361668	0.00324478	0.00540818

表B.3: 利用者のWebコンテンツがAuthorityとして機能した場合(Authority値)

0.00641927	0.00830437	0.0076452	0.00511599	0.00426275	
0.02366484	0.01617632	0.00803882	0.01163708	0.00601161	
0.01059977	0.00795918	0.01062845	0.00480797	0.00825433	
0.01173327	0.02811738	0.01338054	0.00804301	0.00783698	
0.00790732	0.00961361	0.00922891	0.00768576	0.01397345	
0.01632296	0.00750356	0.01719268	0.01007675	0.0114863	
0.0139903	0.01355627	0.01196294	0.00893949	0.00734807	
0.00792331	0.0063468	0.00437438	0.00442962	0.00222715	
0.0063829	0.00379331	0.01338134	0.0105258	0.00929759	
0.01462949	0.01086576	0.01513306	0.02098491	0.00449226	
0.01242585	0.0163044	0.014608	0.01821855	0.00380209	
0.01267085	0.02032081	0.00499084	0.00776199	0.01077204	
0.00223743	0.01439058	0.01059893	0.01263413	0.00302943	
0.0105733	0.013258	0.00544813	0.00593996	0.00983068	
0.01031735	0.01334563	0.01115315	0.00616821	0.00494552	
0.01437072	0.00617818	0.00484386	0.00405667	0.012052	
0.01244636	0.01083045	0.01483338	0.01348101	0.011283	
0.00508111	0.01262844	0.00670876	0.00571297	0.01201804	
0.00455588	0.00828358	0.00933559	0.00431318	0.00788466	
0.00674144	0.01398105	0.01241175	0.00204394	0.01466693	0.00329806

表B.4: 利用者のWebコンテンツがAuthorityとして機能した場合(Hub値)

0.00653399	0.00807396	0.008238	0.0057436	0.0097927	
0.00971092	0.00933896	0.01481967	0.02304849	0.00739726	
0.00901876	0.00790594	0.00474852	0.01010331	0.01406983	
0.00857892	0.00363781	0.00753786	0.00813867	0.01243616	
0.00721254	0.00422756	0.00998454	0.01102596	0.01202774	
0.00307768	0.01651243	0.00643812	0.01222318	0.01163864	
0.00990925	0.00662348	0.0089492	0.01021904	0.00946775	
0.00631964	0.00653759	0.01270592	0.02230729	0.00621989	
0.00911698	0.00907593	0.01087049	0.01642136	0.0134169	
0.01363067	0.01497776	0.00800926	0.00880985	0.00479681	
0.01104069	0.0077546	0.00330221	0.00785196	0.0050083	
0.00797955	0.01167029	0.00697106	0.01172633	0.01348522	
0.01068399	0.01173965	0.01196862	0.00997599	0.00818078	
0.01483508	0.01161341	0.00806671	0.00671436	0.00686194	
0.01172202	0.0085817	0.00422369	0.00717481	0.01130747	
0.01038492	0.00650274	0.01773977	0.00808424	0.00801739	
0.00867126	0.01550119	0.00754603	0.0164731	0.0148867	
0.0095524	0.00881499	0.01005796	0.01807763	0.00566113	
0.00670408	0.01647828	0.01327411	0.00723962	0.00746365	
0.01670138	0.00824014	0.01900517	0.01358226	0.00327267	0

付録C

実験環境

本研究の実験には2台のコンピュータを使用した．その構成を以下に示す．

- Plat'Home 1U / CPU: Pentium 3 Dual 852MHz / Memory: 514MB
  - OS: CentOS 4.5 [1]
  - Python 2.5 [2]
    - Numerical Python (NumPy) 1.0.5.dev [3] // 行列演算用ライブラリ

- DELL dimension 8400 / CPU: Pentium 4 2.8GHz / Memory: 1GB
  - OS: Debian GNU/Linux 4.0 etch [4]
  - Python 2.4.4
    - Numerical Python (NumPy) 1.0.1
    - NetworkX (NX) 0.36 [5] // 複雑ネットワーク処理用ライブラリ
    - Matplotlib 0.87.7 [6] // 2D 描画用ライブラリ

[1] "www.centos.org - The Community ENTerprise Operating System", http://www.centos.org/
[2] "Python Programming Language – Official Website", http://www.python.org/
[3] "Numpy Home Page", http://numpy.scipy.org/
[4] "Debian – The Universal Operating System", http://www.debian.org/
[5] "NetworkX", https://networkx.lanl.gov/wiki
[6] "Matplotlib / pylab - matlab style python plotting (plots, graphs, charts)", http://matplotlib.sourceforge.net/

付録D

作成/使用プログラム一覧

本研究で作成及び使用したプログラムを以下に示す.

リスト D.1: シミュレーション用プログラム

```
#!/usr/local/bin/python

import random      # 乱数処理
import pprint      # ログ保存
import numpy       # 行列演算
import networkx    # 複雑ネットワーク処理
import pylab       # 支援networkx

METADATA = {       # メタデータの種類に応じた色の定義
    1: "red",
    2: "blue",
    3: "green",
    4: "yellow",
    }

def hits(e):       # アルゴリズムHITS
    et = numpy.transpose(e)                    # 転置行列
    a = numpy.transpose(numpy.ones((100))) # 値初期化Authority
    h = numpy.transpose(numpy.ones((100))) # 値初期化Hub
    k = 0
    while k < 1000:
        ad = numpy.dot(et, h)
        hd = numpy.dot(e, a)
        a = numpy.divide(ad, numpy.sum(numpy.absolute(ad)))
        h = numpy.divide(hd, numpy.sum(numpy.absolute(hd)))
        k = k + 1
    return (a, h)

e1 = numpy.zeros((100, 100))    # 隣接行列改良(用HITSの初期化)
e2 = numpy.zeros((100, 100))    # 隣接行列改良無し(用HITSの初期化)

webgraph = networkx.XGraph()    # グラフ定義
pos = networkx.spring_layout(webgraph)

svfile = open("randomwebgraph.txt", "w+")  # ログファイル

for i in xrange(1000):                     # 改良隣接行列生成HITS
    e1[random.randint(0, 99)][random.randint(0, 99)] = random.randint(1, 4)
print e1
#pprint.pprint(e1, svfile)

for i in xrange(100):                      #改良無し隣接行列生成HITS
    for j in xrange(100):
        if e1[i][j] != 0:
            webgraph.add_edge(i, j, METADATA[e1[i][j]])
            e2[i][j] = 1
```

```
47 print e2
48 #pprint.pprint(e2, svfile)
49
50 (a1, h1) = hits(e1)                          # 改良HITS
51 print "a1: ", a1
52 pprint.pprint(a1, svfile)
53 print "h1: ", h1
54 pprint.pprint(h1, svfile)
55
56 (a2, h2) = hits(e2)                          # 改良無しHITS
57 print "a2: ", a2
58 pprint.pprint(a2, svfile)
59 print "h2: ", h2
60 pprint.pprint(h2, svfile)
61
62 asub = numpy.subtract(a1, a2)                # 比較のための差
63 print asub
64 pprint.pprint(asub, svfile)
65 hsub = numpy.subtract(h1, h2)                # 比較のための差
66 print hsub
67 pprint.pprint(hsub, svfile)
68
69 svfile.close()
70
71 networkx.draw(webgraph)                      # グラフの描画
72 pylab.savefig("randomwebgraph.png")
73 pylab.savefig("randomwebgraph.eps")
74 pylab.show()
```

リスト D.2: Web コンテンツ Push 配信シミュレーション用プログラム

```
1  #!/usr/local/bin/python
2
3  import random      # 乱数処理
4  import pprint      # ログ保存
5  import numpy       # 行列演算
6  import networkx    # 複雑ネットワーク処理
7  import pylab       # 支援networkx
8
9  METADATA = {       # メタデータの種類に応じた色の定義
10     1: "red",
11     2: "blue",
12     3: "green",
13     4: "yellow",
14     }
15
16 def hits(e):       # アルゴリズムHITS
17     et = numpy.transpose(e)                  # 転置行列
18     a = numpy.transpose(numpy.ones((101)))   # 値初期化Authority
```

```
19      h = numpy.transpose(numpy.ones((101))) # 値初期化Hub
20      k = 0
21      while k < 1000:
22          ad = numpy.dot(et, h)
23          hd = numpy.dot(e, a)
24          a = numpy.divide(ad, numpy.sum(numpy.absolute(ad)))
25          h = numpy.divide(hd, numpy.sum(numpy.absolute(hd)))
26          k = k + 1
27      return (a, h)
28
29  e1 = numpy.zeros((101, 101))    # 拡張隣接行列提供(コンテンツWebHubの初期化)
30  e2 = numpy.zeros((101, 101))    # 拡張隣接行列提供(コンテンツWebAuthorityの初期化)
31
32  webgraph1 = networkx.XGraph()    # グラフ定義
33  webgraph2 = networkx.XGraph()
34  pos = networkx.spring_layout(webgraph1)
35  pos = networkx.spring_layout(webgraph2)
36
37  svfile = open("randomwebgraph2.txt", "w+")  # ログファイル
38
39  for i in xrange(1000):                      # ランダムなハイパリンク生成
40      r1 = random.randint(0, 99)
41      r2 = random.randint(0, 99)
42      e1[r1][r2] = random.randint(1, 4)
43      e2[r1][r2] = e1[r1][r2]
44      webgraph1.add_edge(r1, r2, METADATA[e1[r1][r2]])
45      webgraph2.add_edge(r1, r2, METADATA[e1[r1][r2]])
46  print e1
47  print e2
48  #pprint.pprint(e1, svfile)
49  #pprint.pprint(e2, svfile)
50
51  uc = numpy.transpose(numpy.ones((101)))
52  for i in xrange(3):                     # 提供コンテンツにWeb
53      r = random.randint(0,100)           # ランダムなハイパリンク生成
54      uc[r] = random.randint(1, 4)
55      e1[100][r] = uc[r]
56      e2[r][100] = uc[r]
57      webgraph1.add_edge(100, r, METADATA[uc[r]])
58      webgraph2.add_edge(r, 100, METADATA[uc[r]])
59
60  (a1, h1) = hits(e1)                         # 改良HITS
61  print "a1: ", a1
62  pprint.pprint(a1, svfile)
63  print "h1: ", h1
64  pprint.pprint(h1, svfile)
65
66  (a2, h2) = hits(e2)                         # 改良HITS
67  print "a2: ", a2
```

```
pprint.pprint(a2, svfile)
print "h2: ", h2
pprint.pprint(h2, svfile)

svfile.close()

networkx.draw(webgraph1)                    # グラフの描画
pylab.savefig("randomwebgraph2_1.png")
pylab.savefig("randomwebgraph2_1.eps")
pylab.show()

networkx.draw(webgraph2)                    # グラフの描画
pylab.savefig("randomwebgraph2_2.png")
pylab.savefig("randomwebgraph2_2.eps")
pylab.show()
```