Webインテリジェンスに基づく
ユーザの知的活動支援に関する研究

A Study on Structuring Web Contents for Web Intelligence

2013年 1月

佐 野 博 之

論文要旨

本研究ではWebインテリジェンス技術の応用によって，ユーザの知的活動を支援することを目的としている．ここでのWebインテリジェンス技術とは主に，Webページを Webコンテンツ単位へと分割するためのWebページ分割手法，および，Webコンテンツへのアノテーションなどの，Web情報の構造化技術を差す．構造化されたWeb情報を利用し，Webコンテンツの再利用という観点からユーザの支援を行う．

Web情報の構造化に関連して，Web上にはハイパーリンクと呼ばれるネットワーク構造が存在する．Webにおける既存のハイパーリンク構造はWebページ制作者の観点に基づく構造である．既存のWebコンテンツ間に対してWeb閲覧者が自由にハイパーリンクを張ることが可能な機構を実現することによって，Web上に存在する既存のハイパーリンク構造とは異なる，Web閲覧者同士の新たなハイパーリンク構造の実現が期待できる．Web情報の構造化に関する研究は既に多数存在するが，閲覧者の観点に着目して構造化を試みた研究は少ない．

Web情報を閲覧者の観点から構造化するために，閲覧者がWeb上の情報に対して自由に自身の観点を記述できるようなシステムを試作した．具体的には，Webページ中に存在するWebコンテンツに対して閲覧者が付箋によるアノテーションを行い，Webコンテンツ間に対してハイパーリンクを作成することが可能なシステムとなっている．本システムの実現において，任意のWebコンテンツに対してアノテーションを行うための技術や，アノテーション間の双方向リンクモデルについての研究を推進した．本システムについては，3章で詳しく述べる．

アノテーションの対象となったWebコンテンツを特定するために，WebページをWebコンテンツ単位へと分割するための手法に関する研究を行った．Webページは一般的に複数のコンテンツから構成される．Webページを記述するためのHTMLは半構造化文書である．HTML文書中には各コンテンツの明確な区切りは記述されていない．高精度なWebページ分割により検索エンジンの精度向上など多くの利点が指摘されており，研究の余地がある．Webページ分割に関する研究は既に多数存在するが，面積や子ノード数など，コンテンツ量に依存する情報を用いる．その結果，同一Webサイト内の同じレイアウトのWebページから異なる分割結果が得られるという問題が存在した．本研究ではWebコンテンツの見出し部分をWebコンテンツ間のセパレータとして利用し，Webページ分割を行う手法を提案した．見出し部分の抽出のために，J4.8アルゴリズムによる決定木学習によって分類器を生成した．評価実験により，分類器の分類精度はF値77.8%，89.3%であることを確認した．得られた見出し部分を用いて最小ブロックの結合を行った結果，ニュースサイトのニュース記事部分に着目した場合，96.1%の精度でコンテン

ツ量に依存しない同一の分割結果が得られることを確認した．複数の Web ページ上で提案手法を用いた Web ページ分割を行い，実験対象とした全ての Web ページで 1000 ミリ秒以内に処理が完了することを示した．本手法の詳細は，4 章で説明する．

3 章，および 4 章で提案した手法を組み合わせることで，Web 情報を閲覧者の観点から構造化することが可能となる．提案手法によって新たに構造化された Web の有用性を実証するために，ユーザの知的活動を支援するためのアプリケーションを試作した．Web ページを Web ブロックへと分割し，Web ブロックをクラウド環境上へと保存することによって Web 情報の再利用性を向上させるためのシステムを試作した．また，Web 情報を元に議論を進めるための議論支援システムや，タブレット端末向けの会議支援システムなどを試作した．提案システムを通じて，本研究で確立した技術が既存の Web コンテンツの再利用性を向上させることを示した．

以下に，本論文の構成を述べる．1 章ではまず，本研究の学術的背景について述べる．研究項目を設定し，それぞれの研究項目の目的について述べる．2 章では本研究を進める上で必要となる基盤技術について述べる．また，既存研究について言及し，本研究の位置づけを行う．3 章では付箋アノテーションシステムの実装について述べる．システムの構成図やシステム実行例のスクリーンショットを用いて全体像についての説明を行った後，DOM ツリーに基づく Web コンテンツの同定手法，および Web コンテンツ間へのハイパーリンク作成について詳しく述べる．4 章では Web ページ分割手法について述べる．分割手法を 3 ステップに分けて詳しく説明を行う．評価実験によって本手法の有効性を示す．5 章では Web 情報を利用してユーザの知的活動を支援するためのアプリケーションについて言及する．最後に 6 章で今後の課題を述べるとともに，本研究をまとめる．

Abstract

The goal of the research is to support Web users in their intellectual activities based on Web intelligence technologies. The research mainly focused on Web structuring technology, such as annotation to the Web contents and Web page segmentation method that divides a Web page into Web contents. The structured Web enables users to easily reuse Web contents.

The most famous structure on the Web is the hyperlink network. The network is based on Webmasters' viewpoints. Webmasters generate hyperlinks to get higher ranking score on major Web search engines, to increase the number of their sites' visitors, to improve affiliate sales on their sites, and so on. If there is a system that enables Web visitors to generate new hyperlinks between existing Web contents, a new hyperlink network will be created by Web viewers. That new network based on Web visitors' viewpoints will completely differ from the existing hyperlink network. Many researchers try to restructure Web information, but there is little research that focuses on Web visitors' viewpoints.

In Chapter.3, the thesis proposes a Web annotation system that enables Web visitors to describe their viewpoints to Web Information, to restructure Web information based on their viewpoints. The Web annotation system enables users to place stickies on Web contents in any existing Web pages. The stickies provided by the system enable users to point out specific contents on Web pages, as well as to generate bidirectional links between the stickies referencing the content. The position of the stickies in the system must correspond to the relevant content in such a way. Related systems decide the position of the stickies by using absolute coordinates to equal the position of the stickies to the content. However, if the absolute coordinates are used, a problem occurs that a sticky is not displayed at the precise position of the information that a user references with the sticky, which in turn presents a problem when a user shares stickies with other users. The thesis suggests a new method for displaying stickies, which ensures that each sticky is always displayed at the corresponding place. An agent adds bidirectional links between the stickies in order to cross-reference similar contents in the system. The agent monitors the stickies which users placed, and generates bidirectional links between the stickies that were placed on similar contents.

Chapter.4 describes a new Web page segmentation method to extract the semantic structure from a Web page. A typical Web page consists of multiple elements with different types of features, such as main content, navigation panels, copyright and privacy notices, and advertisements. Web page segmentation is the division of the page into visually and semantically cohesive pieces. The proposed method is comprised of three steps. First, it divides the page

into minimum blocks. Second, it classifies the blocks into two classes, title blocks or non-title blocks. Third, it assembles groups of these blocks into Web content blocks. While the minimum blocks can play many roles, this study focused on blocks that are the titles of various Web content bits. Decision tree learning is used with nine parameters for each minimum block to extract title blocks from Web pages. Experimental results showed that the decision tree generated by the J4.8 algorithm is the most suitable for this type of extraction, and the segmentation method based on title blocks can divide Web pages that are collected from the news site with 96.1 percent accuracy, independently of amount of content. The results also describes that the method can divide all Web pages that are used in the experiment less than 1000 milliseconds.

Three applications are implemented with Web intelligence technologies to support Web users' intellectual activities. The applications are described in Chapter.5. The first system is a Web contents extraction system. The system enables users to extract Web blocks from Web pages and save the blocks in a cloud computing environment to reuse extracted blocks. The second system is an application to support the discussion of regional issues based on an e-Participation Web platform O_2. The third is a meeting support system for tablet computers. These systems show validities of the technologies proposed in the thesis. Users can easily reuse Web contents in existing Web pages for intellectual activities by using the systems.

目 次

図 目 次

表 目 次

第1章　序論

本章ではまず，本研究の背景について述べる．Web の誕生と進化の概略について，適度に関連研究を引用しつつ説明を行い，“Web インテリジェンス” と呼ばれる研究分野の重要性について言及する．次に研究項目を 3 つ設定し，それぞれの研究項目の目的について述べる．さらに本研究の貢献について言及した後，最後に本論文の構成を示す．

1.1　背景

Web は当初，Web ページの制作者から閲覧者に対して情報伝達を行うためだけの情報配信ツールであった．近年の Web は，ソーシャル・ネットワーキング・サービスに代表されるように，複数のユーザ間でコミュニケーションを行い情報共有をするための，社会的な情報基盤へと進化しつつある．本節では Web の歴史と Web に関する研究を概観することで，“Web インテリジェンス” というキーワードで表現される，知的な Web プラットフォームを実現するための研究の必要性について述べる．

1990 年に Tim Barners-Lee が World Wide Web（以下，Web）を提唱し [43]，世界初の Web サーバ，および Web ブラウザを実装してから，既に 20 年以上が経った．1990 年代半ばまではインターネットの利用は軍事利用や教育機関での利用が主であり，Web の普及率は高くなかった．1995 年に Microsoft によって発売された Windows 95 では，OS 標準の機能として TCP/IP をサポートした．それ以降，一般ユーザでも簡単にインターネットへ接続することが可能となり，Web のユーザ数が爆発的に増加する．それとともに Web ページ数は常に増加を続け，今や Web は人類史上最大の情報源となった．インターネットのモニタリングサービスを展開している Pingdom によれば，2011 年の時点で 5 億 5000 万の Web サイトが存在するという [34]．Web はインターネットの応用技術の一つであるが，今では “インターネット=Web” であると誤解している人も少なくない．以下に，Web がこれほどまでに普及した理由を 3 つ挙げる．

1 つ目に，Web ページが簡単に作成でき，誰もが世界に向けて情報公開することが可能である点が挙げられる．Web ページの実態は HTML と呼ばれる半構造化文書である．HTML 文書のタグは木構造を構成するが，それぞれのタグの中身は非構造化データであるテキストである．HTML タグの知識とテキストエディタさえあれば，誰でも Web ページを作成することができる．作成した Web ページの公開には Web サーバが必要となるが，インターネット・サービス・プロバイダが各契約者に対して Web サーバのスペースを無料で提供するといったサービスを用意している．したがって自前の Web サーバを用

意する必要はなく，作成した HTML ファイルをプロバイダの Web サーバに対してアップロードするだけで Web ページの公開が完了する．

2 つ目は，ハイパーリンク機能である．HTML ではアンカーと呼ばれるタグ (a タグ) を利用して他のドキュメントの URL を指定することで，他のドキュメントへのハイパーリンクを作成することが可能である．ハイパーリンクとは，HTML 文書間を結びつける機能のことである．ハイパーリンク機能によって，ユーザはマウスをクリックしていくだけで，次々と Web ページ間を遷移することが可能となっている．ハイパーリンクによって，Web 上に存在するドキュメントは複雑なクモの巣のような網構造を形成する．ハイパーリンク情報を利用した Web 構造マイニングに関する研究も盛んに行われている．Web 上のハイパーリンク情報を用いて Web ページのランキングを行う HITS アルゴリズム [25] や PageRank アルゴリズム [27] がある．

3 つ目は，全てのリソースが URL によって参照可能な点である．Web 上では HTML ファイルはもちろん，画像や動画など，Web サーバ上に配置されたファイルは全て URL によって一意に特定が可能である．例えば `img` タグの `src` 属性に対して画像の URL を指定することで，Web ページ中に画像を表示することが可能である．1993 年に Marc Andreessen によって開発された “NCSA Mosaic” は，画像表示に対応した初の Web ブラウザである．NCSA Mosaic の登場以前は，Web ブラウザ上で表示可能なデータはテキストデータのみであった．NSCA Mosaic の登場によって Web の表現力が飛躍的に向上した．現在では動画や Flash などといった画像より遥かに表現力が高いメディアも，Web ページ上で表示することが可能となっている．

近年 Web は大きな変革期を迎えている．2005 年には Tim O’Reilly によって Web2.0 が提唱された [44]．Web が誕生してから 2000 年代前半までの Web は，閲覧者がただ Web ページを取得し閲覧するだけの Web であった．Web2.0 は閲覧者参加型の Web であると言える．

2000 年代半ばになると，Google や Amazon などの代表的な Web 企業が，積極的に Web API を公開するようになった．Web API とは，Web 経由で Web サーバが保持しているデータや機能へとアクセスするためのプログラミングインターフェイスのことである．例えば Google が公開している Google Maps API を利用することで，Google が保有している地図データを HTTP プロトコル経由で外部プログラムから利用することが可能となる．Web API が公開されたことにより，閲覧者が Web ページをただ単に閲覧するだけでなく，Web の情報を加工して組み合わせ，新たなコンテンツを生成するといったことが可能となった．複数の Web API を組み合わせて新たなコンテンツを生成する技術は，マッシュアップ [82] と呼ばれている．開発者が容易にマッシュアップを行うための仕組みに関しても研究が行われている [31]．

mixi[1] や Facebook[2] など，ソーシャル・ネットワーキング・サービス（以下，SNS）と呼ばれるサービスの普及も記憶に新しい．SNS では，各ユーザが自身のプロフィールを記

[1] ソーシャル・ネットワーキング サービス [mixi(ミクシィ)], `http://mixi.jp/`
[2] Facebook, `http://www.facebook.com/`

述する Web ページ（以下，プロフィールページと呼ぶ）を持つ．SNS では友人関係がハイパーリンクによって表現される．交流があるユーザ間では，プロフィールページ間にハイパーリンクが作成される．すなわち，SNS のプロフィールページ間に存在するハイパーリンクは，人間関係の可視化手段と言える．これらのハイパーリンクに着目し，SNS から人間関係を抽出する研究も盛んに行われている [96,100]．また，SNS では各ユーザが日記を記述するための Web ページも存在する．交流のある他のユーザが書いた日記に対してコメントするといったことも可能である．SNS 内では各ユーザが情報の発信源となり，サービスを盛り上げていく．

Web へアクセスするためのツールも大きく変化しつつある．従来 Web にアクセスするためのツールと言えば，Microsoft の Windows や Apple の Mac OS を搭載したパーソナルコンピュータが大半であった．しかし近年では，スマートフォンと呼ばれる，インターネットにアクセス可能な携帯電話の普及が目覚ましい．2007 年 6 月にアメリカで発売された iPhone は，携帯電話のあり方を全くと言っていいほど変えてしまった．スマートフォンの登場により，我々は，携帯電話の電波が届く場所であればいつでもどこでも，インターネットへアクセスして Web を利用できる環境を手にした．総務省が平成 24 年 1 月〜2 月に行った調査によると，平成 23 年の日本のインターネット人口普及率は 79.1%，スマートフォンの普及率は 16.2%である [83]．スマートフォンの普及率は今後も増え続けることが予想される．

現在，2014 年の正式勧告を目指して，HTML のバージョンアップが行われている．5 回目のバージョンアップであるため，HTML5 と呼ばれている．HTML5 では，Web ページの構造を機械可読にするためのタグが導入されることになっている．動画や音声再生のための API や，GPS や Web カメラなどといったハードウェアリソースにアクセスするための API などの導入も予定されており，Web をアプリケーションプラットフォームとして利用することが可能となる．本節で述べたような Web の誕生と進化に関するタイムラインが，HTML5 を利用した Web アプリケーションとして公開されている [9,21]．これらの Web アプリケーションでは，HTML5 の表現力を十分に味わうことができる．

これまで述べてきたように，Web は単なる情報配信のプラットフォームから，社会インフラへと進化しつつある．Web 上に存在する膨大な情報の利用性を向上させることによって，人々の暮らしがより向上することが期待できる．人工知能技術を Web 上のデータやアプリケーションへ応用し，Web をベースとした知的なプラットフォームを構築しようとする研究が盛んである．そのような研究分野は，“Web インテリジェンス” と呼ばれている．各種学術会議で，Web インテリジェンスに関する研究や議論が盛んに行われている [33,99]．World Wide Web に対して Wisdom（知恵）をもたらすものとして，World Wide Wisdom Web というものも提唱されている [18]．Web インテリジェンス技術の応用により，Web 情報をより高度に活用することが可能になると期待され，研究の余地がある．

文献 [97] では，図 1.1 に示すようなインテリジェント Web インタラクションの概念モデルが提唱されている．図 1.1 における “現実世界の情報” とは，“Web 上の情報” を意

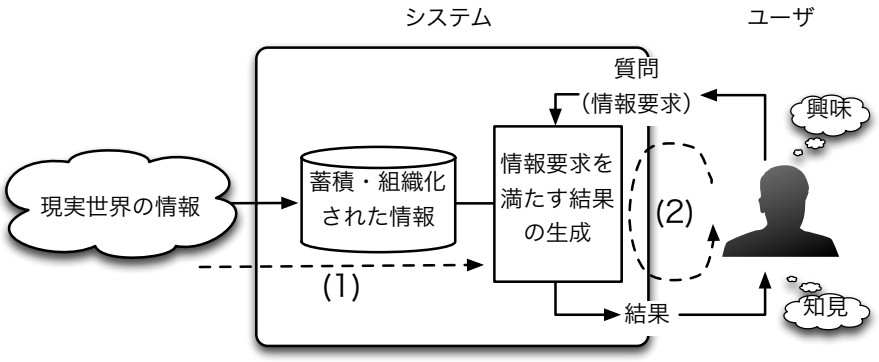


図 1.1: インテリジェント Web インタラクションの概念モデル [97]

味する．このモデルは大きく分けて，(1) と (2) の，2 つのフェーズから構成される．(1) は，Web 上に存在する情報を収集し，計算機にとって扱いやすい形で整理・集約するためのフェーズである．本研究では (1) のフェーズを，“Web 情報の構造化” と呼ぶ．(2) は，ユーザがシステムとのインタラクションを通じて知的活動を行うフェーズである．

本研究では，Web インテリジェンス技術の中でも特に，Web 情報の構造化技術に焦点を当てる．Web 情報構造化によって，Web 情報の利用性向上を目指す．Web 上に存在する情報は大多数が半構造化データである．計算機を用いて情報の統合や検索をするためには，多くの課題を解決する必要がある．既存の Web 情報を利用してユーザの知的活動を支援するために，Web 情報を構造化することが必要である．Web 情報構造化に関して，Web ページを閲覧者の観点から Web コンテンツ単位へと分割するための技術や，Web コンテンツに対するアノテーション技術の研究を行う．実際に Web 情報を利用してユーザの知的活動を支援するためのアプリケーションを構築し，本研究の有用性を示す．

1.2　研究項目，および目的

本研究の大目標は，Web 利用者の知的活動を支援することである．以下に示す 3 項目を，本論文の研究項目として定める．

研究項目 1. Web コンテンツへのアノテーション技術，およびアノテーション間への双方向リンクモデルの設計

研究項目 2. Web ページ分割手法

研究項目 3. Web 情報の利用に基づく知的活動支援アプリケーションの設計

研究項目 1 と研究項目 2 は，Web 情報構造化技術に関する研究である．研究項目 1 と研究項目 2 は，図 1.1 の (1) のフェーズに相当する．研究項目 3 は，Web インテリジェンス技術に基づいたユーザの知的活動支援に関する研究である．研究項目 3 は，図 1.1 の

(2) のフェーズに相当する．研究項目 3 では，研究項目 1 と研究項目 2 で確立した技術も応用する．それぞれの研究項目と目的について，以下に詳細を述べる．

研究項目 1. Web コンテンツへのアノテーション技術，およびアノテーション間への双方向リンクモデルの設計

Web ページ中に存在する任意の Web コンテンツに対して，Web 閲覧者がアノテーションを行うための技術を確立する．アノテーションによって既存の Web コンテンツが手動構造化される．次に，アノテーション対象となった Web コンテンツ間の関連性を可視化するための双方向リンクモデルを提案し，ユーザの Web 閲覧支援へと応用する．

Web ページのレンダリング結果は Web ブラウザに設定してあるデフォルトのフォントサイズや，Web ブラウザのウィンドウサイズに依存する．レンダリング結果が異なると，Web コンテンツが表示される位置も変化する．既存システムが採用する絶対座標を用いたアノテーション表示手法では，Web ページのレンダリング結果が変化した際に，アノテーション対象となったコンテンツからアノテーションがずれてしまうという問題点がある．本研究ではまず，レンダリング結果に頑健なアノテーション位置決定手法を確立する．本技術により Web コンテンツを一意に特定可能となる．

作成したアノテーション間に対して Web 閲覧者がハイパーリンクを作成可能とする機構を実現することで，閲覧者が任意の Web コンテンツ間に疑似的にハイパーリンクを作成することを可能にする．Web 上に存在する既存のハイパーリンク構造は，Web ページ制作者の観点に基づくものである．Web ページ閲覧者が各 Web コンテンツ間に対して自由にハイパーリンクを作成可能な機構を実現することによって，Web ページ制作者の観点とは異なる，新たなネットワーク構造の構築が期待できる．

ユーザの Web 閲覧支援を行うために，アノテーション間の双方向リンクモデルを提案する．関連する Web コンテンツ間に対してエージェントが自動的に双方向リンクを作成するための機構について研究を行う．実際にシステムを試作し，評価実験を通じて有用性を評価する．

研究項目 2. Web ページ分割手法

研究項目 1 では Web コンテンツへのアノテーションを実現した．研究項目 1 の実施において，Web ページ中から，アノテーションの対象となった Web コンテンツを閲覧者にとって意味的にまとまりのある単位で特定するための技術が必要となった．研究項目 2 では，Web ページを Web コンテンツ単位へと分割するための手法に関する研究を推進する．

Web ページは複数のコンテンツから構成される．Web ページを記述するための HTML は半構造化文書である．HTML 文書中には各コンテンツ間の明確な区切りは記述されていない．高精度な Web ページ分割により検索エンジンの精度向上など多くの利点が指摘

されている [38]．既存研究で提案されている Web ページ分割手法では，コンテンツ量によって異なる分割結果が得られるという問題点があった．本研究では新たな Web ページ分割手法を確立し，上記課題の解決を目指す．

本研究は，“Web ページの制作者は，Web ページの可読性向上のために各 Web コンテンツに対して見出しを付ける傾向にある” というヒューリスティクスに基づく．予備実験を行い，本ヒューリスティクスが Web ページ分割へ利用可能であることを明らかにする．次に，Web コンテンツの見出し部分を利用した，新たな Web ページ分割手法を提案する．提案手法では，見出し部分を機械学習によって抽出し，それらをセパレータとして Web ページ分割を行う．

提案手法の有用性を示すために，評価実験によってコンテンツ量に依存しない分割結果が得られることを示す．分割を行うために要する時間の測定を行い，実用的な時間内で処理が完了することを示す．

研究項目 3. Web 情報の利用に基づく知的活動支援アプリケーションの設計

研究項目 3 では，Web 情報を利用してユーザの知的活動支援を行うためのアプリケーションを構築する．本研究では以下の 3 つのシステムを試作する．

1 つ目は，Web ページから Web コンテンツを抽出してクラウド環境へと保存することにより，Web 情報の再利用を支援するシステムを実現する．本システムの目的は，簡易的な Web ラッパーとして利用可能とすること，及び，容易に Web コンテンツのマッシュアップを可能とすることである．Web ページから特定の Web コンテンツを抽出するためには，Web ページの HTML や DOM 構造に目を通し対象となる HTML タグ部分を特定する必要がある．情報リテラシーが高くない Web ユーザにとっては困難である．Web ラッパーを自動構築するための研究も盛んに行われている [63, 89] が，適用できる Web ページに制限が多く，精度も十分であるとは言い難い．本システムではマウス操作によって抽出対象範囲をドラッグ&ドロップで指定するだけで，特定の Web コンテンツを抽出可能とする．本システムによって抽出した Web コンテンツを，“Web ブロック” と呼ぶ．Web ブロックをクラウド環境上へと保存し，他の Web システムから容易に再利用可能とする．

2 つ目は，Web 上で配信されているニュース記事やツイートなどを議論の種として議論を行うための，議論支援アプリケーションを試作する．本システムは Web アプリケーションとして実装する．本システム上で入力された議題や意見などは，RDF[3] サーバ上へと保存することによって構造化を行う．議論情報を構造化することによってコンサーンアセスメントを促進することが目的である．

3 つ目は，タブレット端末上で対面会議を支援するための会議支援システムを試作する．Web 上に会議資料となる PDF ファイルをアップロードし，その PDF ファイルを会

[3] Resource Description Framework の略．Web 上のデータに対してメタデータを記述するためのフレームワーク．

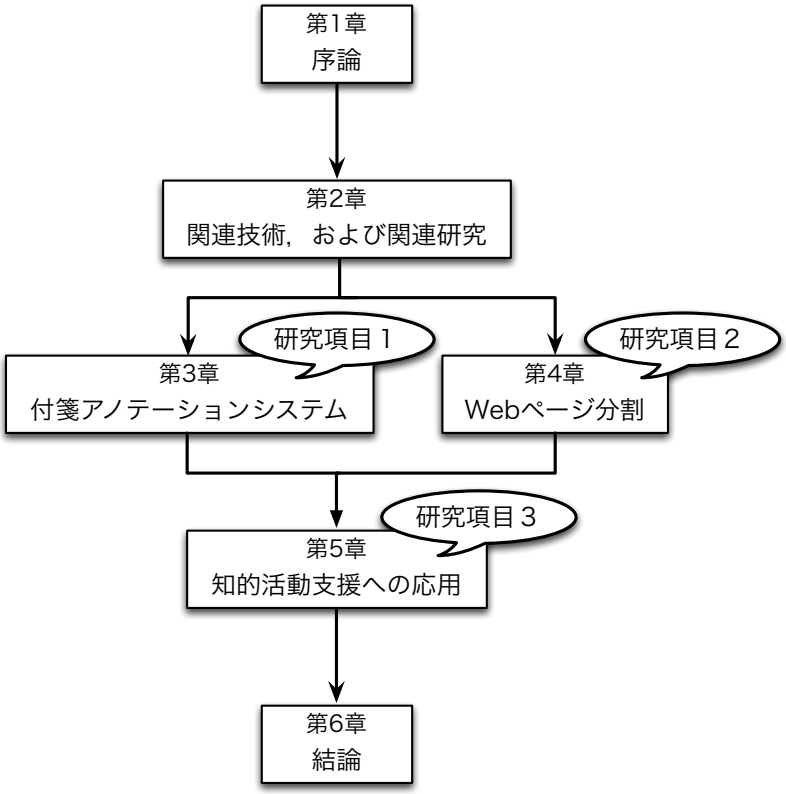


図 1.2: 本論文の章構成，および各章の関連

議資料として利用することによって会議のペーパーレス化を促進する．本システム内部に Web の表示環境を構築することによって，本システムから上述の Web ブロックを利用することや，議論支援アプリケーションを実行することも可能とする．本システムによって PDF の配布資料以外にも，Web ブロックを会議資料として利用することが可能となる．LOD サーバに蓄積された豊富なニュース記事などを会議資料として利用可能とする．複数のタブレット端末間で会議資料の表示同期やポインタ情報の表示同期を行い，円滑な会議進行を支援する．

1.3 論文構成

図 1.2 は，本論文の章構成と，各章の関連性を可視化したものである．

2 章では，本研究を進める上で重要となる技術についての解説を行った後，関連研究について言及し，本研究の位置付けを行う．

3 章では，本研究で実装した付箋アノテーションシステムの設計を示す．任意の Web コンテンツに対してアノテーションを行うための技術や，アノテーション間の双方向リンクモデルについて述べる．システムの構成図やシステム実行例のスクリーンショットを用いてシステムの全体像についての説明を行った後，DOM ツリーに基づく付箋アノ

テーションの表示手法，および付箋アノテーション間への双方向リンク作成手法について詳しく述べる．

4 章では，Web ページを閲覧者にとって意味的にまとまりのある単位へと分割するための，Web ページ分割手法について述べる．Web ページの制作者が Web ページの可読性向上のために各 Web コンテンツに対して見出しを付ける傾向に着目し，見出し部分に基づいた Web ページ分割手法を提案する．

5 章では，Web 情報を利用してユーザの知的活動を支援するためのアプリケーションについて述べる．まずは，Web ブロックをクラウド環境へ保存し，Web 情報の再利用性を向上させるためのシステムについて述べる．次に，Web 情報を議論の種として議論を行うための議論支援システムや，電子会議を行うための会議支援システムについて述べる．これらのアプリケーションにおいて，3 章，および 4 章で確立した Web 情報構造化技術を応用する．

最後に 6 章で，これまでに述べてきた結果を総括し，本研究の成果，および今後の課題を示す．

第2章　関連研究

本章ではまず，本研究の基盤となっている技術を示す．Web インテリジェンスの基盤技術に関して説明を行い，それらの技術が本論文でどのように利用されているかについて簡単に述べる．次に，アノテーションや Web ページ分割に関する既存研究や既存システムについて述べる．それら既存研究の問題点について考察を行い，本研究の位置付けを行う．

2.1　Web インテリジェンスのための基盤技術

ここでは，Web インテリジェンスを実現するために重要となる技術について解説を行う．Web インテリジェンスは Web 技術と人工知能技術の上に成り立つものであるが，その中でも特に，本研究を進めていく上で不可欠な技術に焦点を当てる．まずは，動的な Web アプリケーションを開発するための技術について述べる．本技術は，3 章と 4 章の研究を進める上で必須である．さらに，任意の Web ページ上で JavaScript で記述された Web サービスを実行するための技術についても説明を行う．MiSpider と呼ばれる，付箋アノテーションシステムで使われている Web エージェント技術の解説を行う．次に，4 章の研究を進める背景となる，Cascading Style Sheet について述べる．HTML 文書に対して Cascading Style Sheet の適用した例を示し，文書構造と文書表現が分離できることを示す．最後に，5 章で述べる議論支援システムの基盤インフラとなる住民参画 Web プラットフォームについて説明をする．

2.1.1　動的な Web アプリケーション開発技術

2000 年代前半までは，Web アプリケーションは画面遷移を伴うことが当たり前であった．Web アプリケーションの実行結果はサーバ内に実装された CGI によって Web ページとして出力される．Web ブラウザはその Web ページを表示することによって，ユーザとインタラクションを行う．ユーザがアプリケーション上で何か操作を行うたびに Web ページの遷移が発生していた．

しかし，Ajax の登場により，Web ページの遷移を発生することなしに，1 つの Web ページ内で Web アプリケーションの実行結果を確認できるようになる．Ajax とは “Asynchronous JavaScript ＋ XML” の略であり，Web ブラウザ内部で Web アプリケーションのインターフェイスの構築や非同期通信などを行う技術の総称である．技術自体は Ajax という言葉が発生する前から存在していたが，文献 [17] で Jesse James Grarrett により Ajax

と命名された．特別に新しい技術を用いるのではなく，既存の技術の組み合わせであることが特徴である．Grarrett は文献 [17] の中で，Ajax を以下のように定義している．

1. standards-based presentation using XHTML and CSS;
   XHTML と CSS を利用した標準規格の表現である
2. dynamic display and interaction using the Document Object Model;
   Document Object Model を利用した動的な表示，およびユーザとの対話を行う
3. data interchange and manipulation using XML and XSLT;
   XML と XSLT[1] を利用してデータのやり取りを行う
4. asynchronous data retrieval using XMLHttpRequest;
   XMLHttpRequest を用いて非同期にデータを受信する
5. and JavaScript binding everything together.
   上記全てを JavaScript によって統合する

Ajax では，JavaScript の XMLHttpRequest クラスを用いて Web ブラウザと Web サーバ間で通信を行う．JavaScript とは Web ブラウザ上で動作するスクリプト言語である．Web ページ内で XMLHttpRequest を発行し，非同期にレスポンスデータを取得する．非同期通信とは，Web ブラウザから Web サーバに対してリクエストを発行した後，レスポンスデータの受信完了を待つことなしに，Web ブラウザが他の動作を継続することを意味する．同期通信では，Web ブラウザから Web サーバに対してリクエストを発行すると，レスポンスデータを受信完了するまで Web ブラウザの動作は停止してしまう．Web アプリケーションの構築において同期通信を利用した場合，通信の応答時間の間はユーザが Web アプリケーションを操作できない．すなわち，ユーザエクスペリエンスが低下してしまう．Ajax では非同期通信を利用するため，通信の応答時間内においてもユーザは Web アプリケーションを操作可能である．

Ajax が登場した当時は，XMLHttpRequest のレスポンスデータは XML 形式が主流であった．しかしレスポンスデータの形式は XML である必要はない．XML の代わりに JSON なども多用される．JSON は “JavaScript Object Notation” の略であり，JavaScript におけるオブジェクトの表記法に基づくデータフォーマットである．JSON の実態はテキストデータであるが，JavaScript 内部で eval 関数を利用することにより JavaScript のオブジェクト形式へと変換可能である．したがって，Ajax との親和性が非常に高いと言える．

レスポンスデータの受信が完了すると，JavaScript の Document Object Model API を通じて，Web ページを構成する HTML ファイルの一部を動的に書き換える．Document Object Model API(以下，DOM）とは，HTML 及び XML ドキュメントに対してアクセスするための API（Application Program Interface) である．DOM では，ドキュメントを木構造として扱うことが可能である．例えば図 2.1 の上部に示すような HTML ドキュメン

[1] XML Transformation：XML によって記述された文書を他の XML 文書に変換するための簡易言語

```
<html>
  <head>
    <title>あいさつ</title>
  </head>

  <body>
    <div id='morning'>おはよう</div>
    <div id='noon'>こんにちは</div>
    <div id='night'>こんばんは</div>
  </body>
</html>
```

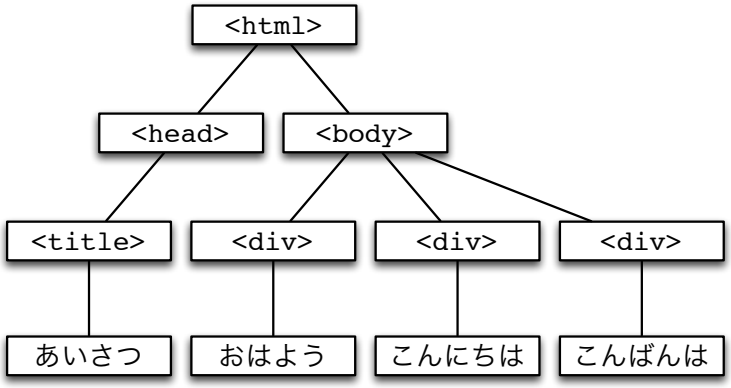


図 2.1: HTML ドキュメントと，それを木構造表記したもの

トは，図 2.1 下部のような木構造 (DOM ツリー) で表現される．DOM によって，HTML ドキュメント上のあらゆるタグ要素に容易にアクセスすることが可能となる．例えば図 2.1 において “こんにちは” というテキストが格納されている DIV タグに対してアクセスするためには，

```
var n = document.getElementById('noon');
```

というコードを書けばよい．上記のコードを適用することにより，目標とする DIV タグがオブジェクト化された上で変数 n に格納される．“こんにちは” を “Hello” へと変更する場合には，

```
n.innerText = 'Hello';
```

とするだけである．ここでは説明の簡略化のために “Hello” へと書き換えたが，Ajax では XMLHttpRequest のレスポンスデータに基づいて HTML の書き換えを行う．

上記が Ajax における処理の流れである．Ajax によってユーザはページ遷移することなしに，1 つの Web ページ内部で Web アプリケーションの実行結果を確認することが

できる．Google が Google Maps[2] に Ajax を利用したことで，Ajax が世界中から着目された．Google Maps の登場以前においては地図のスクロールや縮尺の変更を行う度にページ遷移が発生していたが，Google Maps ではページ遷移することなしにそれらの操作が可能となった．

既存の Web ブラウザ上で動作するため，ユーザは新たにソフトウェアをインストールすることなしに Ajax を用いて作成された Web アプリケーションを実行することが可能であるが，開発者は各種 Web ブラウザ間の DHTML，JavaScript などの実装の違いを考慮したコードを書く必要があった．近年ではブラウザの互換性を吸収するための Ajax 用アプリケーションフレームワークも登場し（Prototype JS，jQuery など），これらフレームワークを用いて開発を行うことで，ブラウザの互換性に関する問題を解決することが可能となっている．

本研究で実装した付箋アノテーションシステムでは，Ajax を用いて付箋の貼り付けを実現している．ユーザがマウスをクリックすると，マウスカーソル周辺の HTML からテキストデータを取得し，XMLHttpRequest によってサーバへ送信する．サーバ側では受信した結果を形態素解析し，解析結果を Web ブラウザへと返す．Web ブラウザはレスポンスデータを形態素ごとに `span` タグで括った上で，オリジナルの HTML を書き換える．さらに，付箋アノテーションを表現するための HTML タグを挿入する．

2.1.2　任意の Web ページ上でのプログラム実行

任意の Web ページ上で JavaScript で記述されたプログラムを実行するための技術として，(1)HTTP プロキシサーバを用いる方法 [67]，(2) ブックマークレットを用いる方法 [86] が知られている．近年では，(3)Web ブラウザの機能拡張として実装する方法も存在する．

(1)HTTP プロキシサーバを用いる方法

HTTP プロキシサーバとは学校や企業などの内部ネットワークとインターネットの間に存在する計算機である．直接インターネットに接続できない内部ネットワークに存在する計算機に変わって，代理としてインターネット上の Web サーバとの接続を行う．ネットワークに出入りするアクセスを一元的に管理し，内部ネットワークから外部ネットワークへの特定の種類の接続のみを許可，外部ネットワークからの不正なアクセスを遮断するために用いられる．ユーザは Web ブラウザに対して，HTTP プロキシサーバを使用することを明示的に設定する必要がある．

ただ単に HTTP プロトコルの中継地点として HTTP プロキシサーバを利用するのではなく，HTTP プロキシサーバに対して様々な機能を実装することも可能である．例えば，HTTP プロキシサーバの中には，外部ネットワークと接続する際の回線の負荷を軽減す

[2] Google マップ - 地図検索, `https://maps.google.co.jp/`

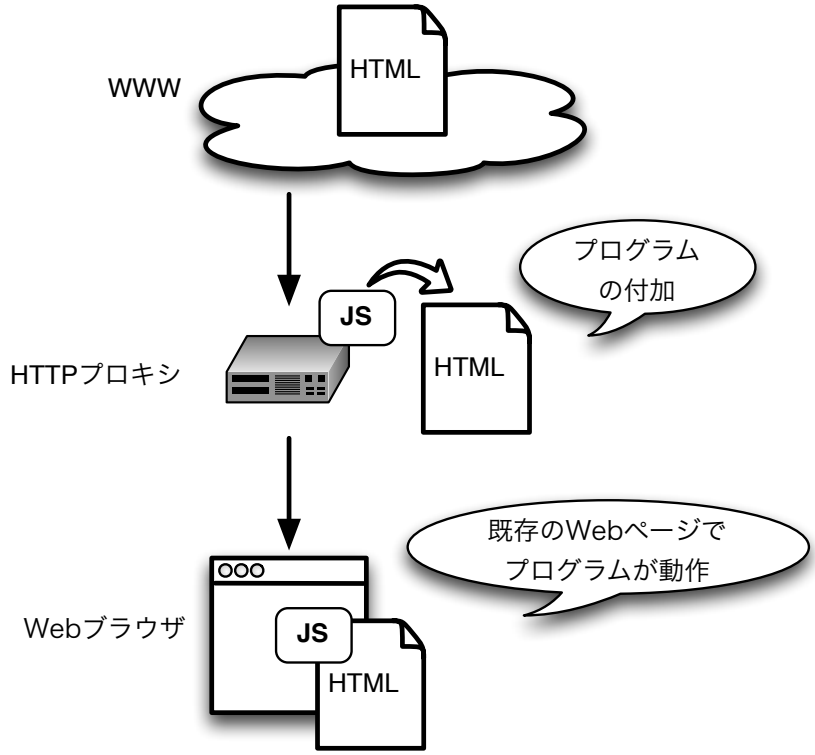


図 2.2: HTTP プロキシを利用した Web サービス (JavaScript) の付加

るために，一度読み込んだファイルをしばらくの間保存しておく，キャッシュ機能を持つものもある．Web サーバから取得したファイルから，バナー広告，JavaScript，ポップアップウィンドウといったものを除去するなど，コンテンツのフィルタリングを行ってから Web ブラウザに送信することも可能である．コンテンツの除去を行うのではなく，それとは逆に，コンテンツの付加も行うことが可能である．図 2.2 に，HTTP プロキシ経由で Web ページを取得する例を示す．図 2.2 では，HTTP プロキシサーバ上で JavaScript を付加した上で，クライアントの Web ブラウザへ Web ページを送信している．HTTP プロキシサーバを介して Web サーバに接続することによって，取得する全ての Web ページに対して JavaScript を付加することが可能である．

(2) ブックマークレットを用いる方法

Web ブラウザにはブックマーク機能が存在する．ブックマークとは登録してある Web ページを表示するための機能である．ブックマークに対して JavaScript のコードを登録しておき，Web ページを表示中にそのブックマークを呼び出すことによって，表示中の Web ページ上で登録してある JavaScript を動作させることが可能である．本技術をブックマークレットと呼び，任意の Web ページ上でプログラムを動作させることが可能な技

術として幅広く利用されている．例えば Amazon では，Web ブラウザ上に表示中の商品を，Amazon の欲しい物リストへ追加するためのブックマークレットが公開されている．Evernote では，Web ブラウザに表示中の Web ページを，Evernote のノートブックへ追加するためのブックマークレットが公開されている．

(3)Web ブラウザの機能拡張として実装する方法

近年では Safari や Chrome のように，Web ブラウザの機能拡張を JavaScript で実装することが可能となっている [2,8]．しかし，プラグインとしてパッケージ化するためにはそれぞれのブラウザ独自の構造で JSON や XML ファイルを用意する必要があり，開発者の手間となる．JSON や XML ファイル作成の手間を省くという点でも，プロキシサーバやブックマークレットには優位性がある．

本研究では任意の Web ページに対して JavaScript を付加するためにプロキシ技術を用いた．ブックマークレットを採用しなかった理由として，ユーザが Web ページ読み込みの度にブックマークレットを起動する必要があり面倒である点が挙げられる．Web ブラウザの機能拡張として実装しなかった理由として，Web ブラウザごとにプラグイン化するための設定ファイルを記述する必要があり，開発の手間になるからである．上記 2 点を考慮した結果，本研究では HTTP プロキシ技術を利用してプログラムの付加を行うこととした．ユーザが Web ブラウザに対して本研究で実装したプロキシサーバを設定することで，任意の Web ページ上で，付箋アノテーション機能を実行することが可能となる．

2.1.3　MiSpider: Web エージェントモデル

MiSpider [40,51] とは Web ブラウザ上で動作するエージェント（本稿では Web エージェントと呼ぶ) モデルである．MiSpider の目的は Web ブラウザ上でエージェント環境を実現し，Web サービスにおける能動的なサービス提供を可能にすることである．MiSpider を利用することによってユーザは既存のブラウザ上でエージェントを利用できるため，インターネットに接続されている環境であれば，世界中からエージェントを利用することができる．

MiSpider におけるエージェントは，ユーザが閲覧ページを移動した時に，セッション間で内部情報を保持したまま移動できる永続性を持つ．また，Web 上の任意のエージェント間で通信を行うためのメッセージパッシング能力を持つ．このようなエージェントの機能を利用することで，ユーザにさまざまな Web ブラウジング支援を提供することができる．

MiSpider はベースエージェントとページエージェントから構成される．ページエージェントは Web ブラウザ上で動作するエージェントであり，JavaScript で記述されている．ベースエージェントはサーバ上で動作する CGI である．ページエージェントはベー

スエージェントのセンサとして動作する．ページエージェントはユーザの Web ブラウザ上で収集した情報をサーバ上のベースエージェントに送信する．ベースエージェントはそのデータを元にサーバ側で知的な処理を行い，結果をページエージェントに送信する．MiSpider ではこのようなサーバ・クライアント型のエージェントモデルを導入することで，サーバの豊富なリソースを使用することが可能となっている．

本論文の 3 章で言及する付箋アノテーションシステムは，Web エージェント技術に基づくものである．Web エージェントモデルとして，MiSpider を採用した．ユーザが Web コンテンツに対して作成した付箋アノテーションがページエージェントとして動作する．ページエージェントは付箋アノテーションの対象となった Web コンテンツのテキストデータをベースエージェントへと送信する．ベースエージェントは受け取ったテキストデータを元に付箋アノテーションをクラスタリングし，類似した内容の Web コンテンツに対して作成された付箋アノテーション間に対して双方向リンクの作成を行う．

2.1.4 文書構造と表現の分離

文書の構造とその文書の表示結果を分離するために，スタイルシートというものが提案されている．構造化文書によって文書の構造や内容を記述し，スタイルシートによってその文書の表示形式を記述する．構造化文書によって記述された内容は，スタイルシートに従って表示され，閲覧者の目に触れることとなる．スタイルシートの中でも特に，HTML や XML の表示形式を決定するために用いられるものが Cascading Style Sheets（以下，CSS）である．スタイルシートの中でも CSS の普及率が圧倒的であるため，単にスタイルシートと言えば CSS を差すことが多い．同じ HTML 文書に対して異なる CSS を適用させると，見た目が全く異なる Web ページとして表示させることが可能となる．

CSS の例を図 2.3 に示す．1 行目では，`div` タグによって記述された要素の表示形式を指定している．ここでは，`div` タグ内に記載されたテキストを 32px のフォントサイズで表示するように指定している．2 行目以降では，`noon` という `id` 属性が指定された要素の表示形式を指定している．CSS では#記号を用いて `id` 属性を指定する．3 行目でフォントサイズを 24px に変更している．4 行目で要素の表示位置を絶対座標で決定するように指定し，5 行目で Web ブラウザの上から 100px の場所に表示するよう指定している．

図 2.1 で示した HTML を Web ブラウザで表示すると，図 2.4(a) のような Web ページとして表示される．次に，図 2.3 で示した CSS を図 2.1 の HTML に適用させると，図 2.4(b) のような Web ページとして表示される．CSS を適用させるためには，HTML の head タグ内部に

```
<link rel="stylesheet" href="CSS のファイル名" type="text/css">
```

と記述すればよい．

図 2.4(a) では上から順に「おはよう」「こんにちは」「こんばんは」と表示されているが，図 2.4(b) では,「おはよう」「こんばんは」「こんにちは」と表示されている.「こんにち

```
1: div { font-size : 32px; }
2: #noon {
3:  font-size : 24px;
4:  position : absolute;
5:  top : 100px;
6: }
```

図 2.3: CSS の例

(a) CSS 適用前　　(b) CSS 適用後

図 2.4: CSS 適用による見た目の変化

は」と「こんばんは」が入れ替わっていることに注意したい．CSS を適用させる HTML ファイルの BODY タグ内は一切変更していないため，HTML ファイル中では,「こんにちは」の後に「こんばんは」が記述されたままである．CSS によって，HTML の DOM ツリー上では隣接している要素が，Web ページとして表示された結果上では隣接していないことが発生する．これが，Web ページ分割を困難にしている理由のひとつでもある．HTML の DOM 要素のみに着目して Web ページ分割を行う研究も多数存在するが，CSS も考慮する必要がある．本研究では Web ページを一度 Web ページをレンダリングし，その結果を元に Web ページ分割を行う．

2.1.5　決定木学習

決定木とは分類問題を解くための予測モデルである．決定木のノードは各変数を表わし，子ノードへの枝はその変数の取り得る値を示す．葉ノードは目的変数の予測値を表わす．事前に与えられた訓練データから決定木を作成する機械学習手法は，決定木学習と呼ばれる．すなわち決定木学習とは機械学習手法の一つである．データマイニング分野においても機械学習は有用である．

ここでは，決定木学習の代表的なアルゴリズムとして，ID3 アルゴリズムについて説

明をする．次に，ID3 アルゴリズムを拡張した C4.5 アルゴリズムについて説明する．

ID3 (Iterative Dichotomiser 3)

ID3 (Iterative Dichotomiser 3) アルゴリズム [22] はオッカムの剃刀の原理に基づいた決定木学習手法である．全ての属性に対して属性値を決定した際の平均情報量 [3] を求め，情報利得が最大となるような属性をノードのラベルに設定する操作を再帰的に行うことによって木構造を構築する．

サンプルデータ集合 C の平均情報量 $H(C)$ は，以下の式で求められる．

$$H(C) = -\sum_{x \in \Omega} P_x(C) \log_2 P_x(C)$$

ここで，x は出力変数，Ω は出力変数の集合，$P_x(C)$ は C において出力が x である確率，である．

C がある属性 $F_i (i = 1, 2, \cdots, n)$ の値に従って m 分割された時の条件付き平均情報量 $H(C|F_i)$ は以下の式で求められる．

$$H(C|F_i) = \sum_{j=1}^{m} \frac{|C_{ij}|}{|C|} H(C_{ij})$$

情報利得 $IG(C, F_i)$ は平均情報量と条件付き平均情報量を用いて，以下の式で求められる．

$$IG(C, F_i) = H(C) - H(C|F_i)$$

全ての属性に対して情報利得を計算し，F_k を以下の式によって決定する．

$$F_k = \underset{F_i}{\operatorname{argmax}}\, IG(C, F_i)$$

F_k でデータを分離し，分離後のデータについても再帰的に処理を行う．平均情報量が 0 になった場合，そのノードを葉ノードとする．

ID3 の問題点として，多くの枝に分離する属性ほど情報利得が大きくなりやすいため，学習を行うことができない可能性がある点が指摘されている．例えば訓練データの属性に各データを識別するための ID が含まれている場合，ID によって分岐をするだけの無意味な決定木が作成される．また，連続値を持つ属性を扱えないという問題点や，訓練データの中に欠損値を持つデータが存在する場合 ID3 では学習を行うことができないという問題点が存在する．

C4.5

C4.5 アルゴリズム [23] は ID3 の問題点を改善した決定木学習手法である．

ID3 では情報利得を用いてデータを分離するための属性を決定するが，C4.5 では情報利得の代わりに情報利得比を用いる．情報利得比とは，分割平均情報量を 1 とした時の情報利得の値である．

C がある属性 $F_i(i = 1, 2, \cdots, n)$ の値に従って m 分割された時の分割平均情報量 $SI(C_i)$ は以下の式で求められる．

$$SI(C_i) = -\sum_{j=1}^{m} P(C_{ij}) \log_2 P(C_{ij})$$

情報利得比 $GR(C, F_i)$ は以下の式で求められる．

$$GR(C, F_i) = \frac{IG(C, F_i)}{SI(C_i)}$$

全ての属性に対して情報利得比を計算し，F_l を以下の式によって決定する．

$$F_l = \underset{F_i}{\operatorname{argmax}}\, GR(C, F_i)$$

連続値を持つ属性を処理するために，C4.5 では離散化処理を行う．連続値を持つ属性区間を閾値によって分割した後，属性を 2 値の特徴へと分割する．この処理を再帰的に繰り返し，連続値の離散化を行う．

C4.5 では属性値の欠損を許す．他のデータから確率的に欠損部分の推定を行い，欠損値が含まれる場合でも学習を可能とする．

過学習を防止するために，C4.5 では枝刈りを行う．過学習とは，訓練データに含まれる偏りやノイズなど，本来であれば学習すべきではないものまで学習してしまうことである．過学習した決定木は，訓練データ以外の未知の事例に対する予測精度が低下する．

C4.5 は，機械学習ツールである WEKA [29] 内で J4.8 として実装されている．本論文の 4 章では J4.8 によって決定木学習を行い，Web ページ中から Web コンテンツの見出しとなるような箇所を抽出するためのルールを発見する．

2.1.6　住民参画 Web プラットフォーム O_2

住民参画 Web プラットフォーム O_2 [59] とは，地域住民がそれぞれの地域問題について話し合うための情報共有基盤である．O_2 では地域の社会問題に関連する情報を RDF 化した上で蓄積し Linked Open Data(LOD) として公開することにより，住民参画のための情報共有基盤を目指している．住民参画とは住民の意見を集約し意思決定に反映させる取り組みである．様々な問題に関する議論を進めるためには，住民の問題意識や懸念事項をまとめる構造化と，その共有が必要である．O_2 の一部として，意見アーカイブとしての LOD データセットを構築した．O_2 における LOD データセットを SOCIA と呼ぶ．

図 2.5 に，O_2 の概要を示す．O_2 には 3 つのステップが存在する．(1) 議論の基となる情報の収集，(2) 収集した情報の構造化と蓄積，(3) 構造化した情報の活用，である．

(1) 議論の基となる情報の収集

エージェントが Web 上をクローリングし，Web に公開されているニュース記事や地方議会の議事録，ツイートを収集する．ニュース記事の収集，および地方議会の議事録の収集には，専用のクローラー，およびラッパーを開発した．対象とするニュース配信サイトは，朝日新聞[3]，読売新聞[4]，産経新聞[5]，毎日新聞[6]，である．地方議会の議事録は，名古屋市の議事録[7] を対象とした．

(2) 収集した情報の構造化と蓄積

(1) で収集した情報を構造化し，相互可用性の高い Linked Open Data(LOD) に基づくデータセット SOCIA として蓄積する．構造化する際に，Web 上のニュース記事をイベントとして構造化し，各イベントとツイートとの関連度を計算し，類似したイベントツイートを関連付ける [93]．機械学習によって作成した地域分類器を用いて，収集した各 Web コンテンツを 47 都道府県へと分類し関連付ける [59].

(3) 構造化した情報の活用

(3) は，(2) において構造化した情報を利用し，地域住民が議論や意見入力を行うフェーズである．(3) のために，構築した LOD データを活用するためのシステムを複数開発した．タブレット端末向けの議論支援システム，会議支援システム，コンサーン・アセスメント支援ツールを開発した．本論文では特に，議論支援システムである citispe@k と会議支援システムについて述べる．

O_2 では図 2.6 に示すように，Web 上の情報を利用して，それらを構造化し意見入力を促すことによって，意見入力に関するサイクルを想定している．議論の基として収集した Web 上の情報を構造化し提示することで，ユーザから意見を入力してもらい，入力された意見を同様に構造化する．この一連の動作を繰り返すことで，問題に関する議論を進める．Web 上の新しい関連情報が提示されることで，意見入力も促される．構造化された情報を分析することによるコンサーン・アセスメントへの応用も期待できる．

Web 上の情報を利用することの利点は，地域に関する社会問題を自動で抽出できる点にある．地域に関する問題は，問題が大きければニュース記事になる可能性も大きくなり，Twitter などでも話題になることが多くなると考えられる．そのため，O_2 のユーザによる意見入力がない状態でも，議論の基となる情報の提供は可能となる．仮に意見入力が行われなくても，地域での問題が把握できるが，この場合はニュース記事や Web 上の情報をまとめただけの情報となる．提供された情報を利用して議題を設定することで，

[3]朝日新聞デジタル, http://www.asahi.com/
[4]ニュース 速報 YOMIURI ONLINE（読売新聞）, http://www.yomiuri.co.jp/
[5]MSN 産経ニュース, http://http://sankei.jp.msn.com/
[6]毎日 jp, http://mainichi.jp/
[7]名古屋市会 会議録の検索と閲覧, http://www.gijiroku.jp/gikai/c_nagoya/index.html

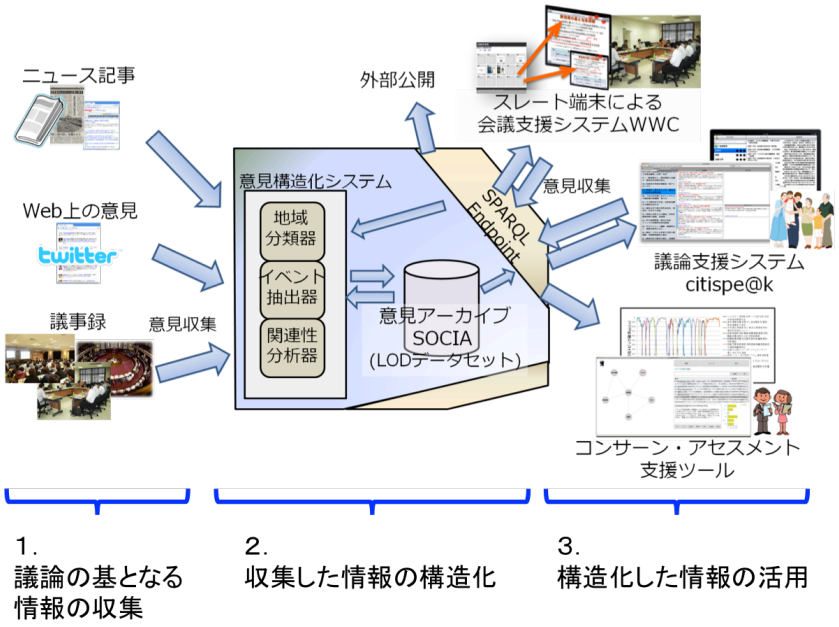


図 2.5: 住民参画 Web プラットフォーム O_2 の概要

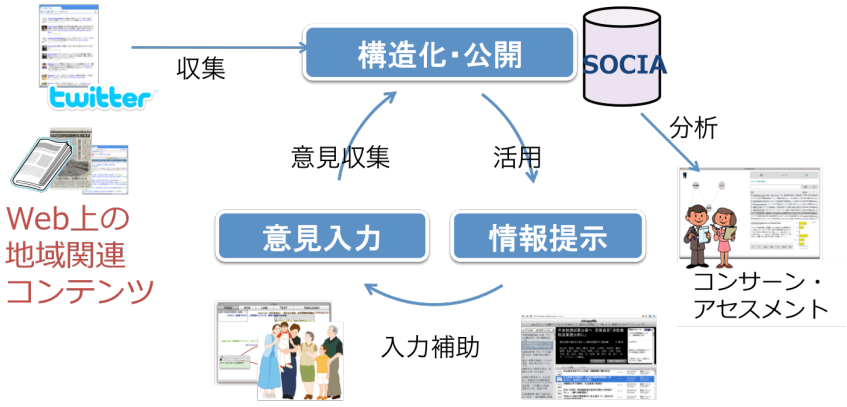


図 2.6: Web 上の情報を利用した住民参画のサイクル

新規に議論に参加するユーザにとっても，議論の対象の把握が容易になる．Web 上の情報を利用するため，新しい情報を基に議論を進行可能となることが期待できる．

2.2　アノテーションに関する既存研究，およびシステム

3 章では任意の Web コンテンツへのアノテーションを実現している．本節ではアノテーションに関連して，学術的研究や，実際に Web 上で公開されている既存のアノテーションシステムについて述べる．

Web ページに存在するテキストデータに対してユーザがアンダーラインを引くことが可能なシステムが，松岡らによって開発されている [95]．文献 [95] では，アノテーションによって付与された原文への 2 次情報（メタデータ）を使って，どのようなサービスに応用できるかについて分析している．このシステムではアノテーション形式の中でも特に，アンダーラインに着目している．松岡らは明治大学の齋藤孝教授が提唱する三色ボールペン読書法を用いて Web 文書に色付きのアンダーラインを付与できるシステムを開発した．実際に第 19 回 人工知能学会全国大会で運用し，評価実験を行った. 三色ボールペン読書法とは，客観的および主観的に重要な部分に色付きのアンダーラインを付与しながら読書をする方法である．分析の結果，ユーザが引いたアンダーラインは文章内の重要語を多く含むことや，ユーザ全体から見るとアンダーラインの色の効果を見出すことができない，ユーザ個人に目を向けると色による個人の特性があることが分かった．松岡らはこれらの実験結果から，アンダーラインから得られる情報は文書要約やユーザプロファイルの作成等に応用できると述べている．

Web ページに対してアノテーションを付加するのみならず，Web ページ内に文章を追加する機能や，Web ページ内の文章を変更する機能を持つ，Web ページの内容を動的に変更できるシステム [66] がある．また，そのアノテーションが自律的に Web ページ上を伝搬していくシステム [85] がある．文献 [66] では，ユーザによる注釈付けをもとにページ内容を動的に再構成する Web 注釈付けシステムを提案している．このシステムによって成長する Web ページが実現し，一つの Web ページから複数のページのバリエーションを提供することが可能となる．また，ページの再構成機能と注釈制御機能を組み合わせることで実現できる Web アプリケーションについて例を挙げている．文献 [85] では，Web アノテーションを拡張した “伝搬するアノテーション” のシステムを提案している．これまでのアノテーションが特定の Web ページに対して行われるという静的なものであることに着目し，竹原らは，アノテーション自身が Web ページを伝搬していき，アノテーションに適する Web ページまで自動的にたどり着くというシステムを構築した．伝搬するアノテーションを実現するためのルールファイルと制御方法について考察を行い，システムの具体的な利用例について述べている．伝搬するアノテーションによってアノテーションが意味をなさない Web ページに居座り続けるのを防ぐことができ，ユーザがよりアノテーションしやすい環境を作ることができる．今回提案するシステムではこのような Web ページの動的な変更やアノテーションの伝搬には対応していないが，Web ページに対して自由にアノテーションを付加することが可能であるという点で関連している．

Annotea [24, 28] とは，WWW コンソーシアム（W3C) で開発されている，Web でアノテーション機能を実現するためのフレームワークである．Annotea を用いることにより，掲示板のような書き込みの機能がない Web ページに対しても，ユーザはアノテーションを書き込むことができ，他のユーザが書き込んだアノテーションを閲覧することもできる．RDF [45] に基づいた記述をすることにより，アノテーションをメタデータとして使用することが可能となるため，Annotea はセマンティックウェブを実現するための取り

組みであると言える．ユーザは，専用の拡張機能 Annotea Ubimarks [8] をインストールした Firefox や，W3C が開発したオープンソースの Web ブラウザである Amaya [9] などから，Annotea を利用することができる．

ソーシャルタギングとは，インターネットの Web サイト上で，投稿されている写真や映像などのコンテンツに対して，そのコンテンツの内容や属性等を一言で記述する索引語やキーワード（以下，タグと記述する）を自由に追加して，検索などに役立てるシステムである．既存の検索エンジンとは対照的に，ボトムアップ的にウェブ上の情報を組織化しようという試みである．多数のユーザがタグを作成しそれを他のユーザ間で共有することができるために，多数のユーザの視点からコンテンツが整理されることが特徴である．インターネット上の情報の集合体の中から目指す情報や隣り合った情報を，より探しやすく，見つけやすく，たどり着きやすくするために考え出された方法である．

ソーシャルタギングを代表するものとして，ソーシャルブックマークが存在する．既存の Web ブラウザのブックマークが，ユーザの使用している Web ブラウザの中にページの URL を保存しておくのに対して，ソーシャルブックマークでは，インターネット上でブックマークを公開し，共有するものである．公開されたブックマークに対しては，全てのユーザが自由にタグを付けることが可能である．タグが多数付加された Web ページは，多くのユーザによって閲覧されている Web ページということになる．ユーザによって付加されたタグを共有し，検索に用いることで，他のユーザが注目している Web ページを発見することを可能とする．ソーシャルブックマークの有名なものとして，Delicious[10] (図 2.7(a)) や，はてなブックマーク[11]（図 2.7(b)) がある．ソーシャルブックマークはブックマークに対してタグを付加するが，写真に対してタグを付加する flickr[12]（図 2.7(c)) や，動画に対してタグを付加するニコニコ動画[13]などがある．

2.3　Web ページ分割に関する既存研究

本論文の 4 章では，Web ページ分割手法を提案している．Web ページ分割とは，Web ページを入力として，閲覧者にとって意味的にまとまりのある単位へと分割することである．高精度な Web ページ分割は，PC で閲覧することを目的として作成された Web ページを携帯電話の小さなディスプレイで閲覧するための Web ページへと変換するための基盤技術，複数の Web コンテンツが含まれている Web ページから閲覧者にとって重要であるメインコンテンツのみを抽出するための基盤技術として利用可能である．

本節では Web ページ分割に関する既存研究について言及をする．既存の研究は，DOM 構造のみに着目した分割手法と，DOM 構造だけでなくレイアウト情報にも着目した分

[8] Annotea Ubimarks :: Add-ons for Firefox, https://addons.mozilla.org/ja/firefox/addon/annotea-ubimarks/

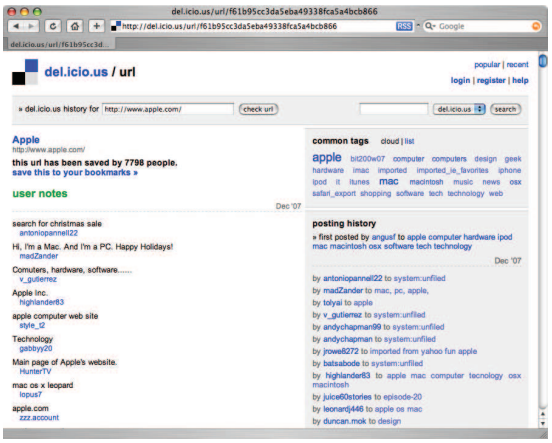
[9] Amaya Home Page, http://www.w3.org/Amaya/

[10] Delicious, http://delicious.com/

[11] はてなブックマーク, http://b.hatena.ne.jp/

[12] Welcom to Flickr - Photo Sharing, http://www.flickr.com/

[13] niconico, http://www.nicovideo.jp/



(a) del.icio.us

(b) はてなブックマーク

(c) flickr

図 2.7: 既存のソーシャルタギングシステム

割手法が存在する．分割結果の粒度も異なる．Web ページをテンプレート単位へと分割するだけの手法や，更に細かく，Web コンテンツ単位まで分割するものなどが存在する．

2.3.1 節では，DOM 構造を用いた分割手法に関する研究について述べる．2.3.2 節では，レイアウト情報を用いた分割手法に関する研究について述べる．

2.3.1　DOM 構造を用いた分割手法

文献 [53] は DOM 構造とリンク数を利用して Web ページをテンプレート単位へと分割する手法を提案している．

文献 [50] は，タグの階層構造を利用して Web ページを Web コンテンツ単位へと分割する手法を提案している．提案手法では，タグの階層構造を辿っていく途中で，Web ページをテンプレート単位へと分割することも可能となっている．

文献 [92] では，Web ページを部分的に解析して得られるタグの数や深さ等の HTML タグの相対的な構造を利用する．算出したコンテンツ間距離に基づき，大小 2 種類の閾値を利用して Web ページの分割点を導出する．文献 [13] では，特定の DOM ノードを根とした部分木を生成し，そこから葉ノードまでのパスのエントロピーを用いて，Web ページ中の意味のあるブロックを抽出している．文献 [52, 55] では，Web コンテンツを携帯電話上の小さなディスプレイで表示するために，DOM 構造を用いた Web ページ分割を行っている．

DOM 構造に基づく Web ページ分割の問題点として，視覚的には隣接している要素でも DOM 構造上隣接していない要素は，同じブロックへ分割することができないという点がある．HTML4 の特徴として，文章の内容と表現の分離が挙げられる．図 2.4 で示したように，視覚情報は HTML ファイルの DOM 構造で表現できるものではなく，CSS によって表現される．したがって，Web ページを閲覧者の観点から分割するためには，HTML の DOM 構造を解析するだけでなく，HTML をスタイルシートと共にレンダリングして得られるレイアウト情報も用いる必要がある．

2.3.2　レイアウト情報を用いた分割手法

文献 [41, 56] は，サポートベクターマシンによる学習を行うことによって，レイアウト情報に基づいた Web ページ分割を行う研究である．Web ページがあるテンプレートに従って作成されているという仮定に基づき，いくつかのテンプレートに分類した後，レイアウト情報を用いてブロックに分割する．Web ページは，ユーザが一目でどこに何が配置されているか把握できるようにデザインされており，そのレイアウトテンプレートの種類は多くない．ここで，Web ページの構成要素をヘッダ（V_h），フッタ（V_f），左メニュー（V_l），右メニュー（V_r），中央記事（V_c）と定義し，それらの組み合わせで作成可能なテンプレートを分類の対象とした．文献 [7] で提案されている手法は，文献 [92] の手法を改良した手法である．

この手法を適用して得られる分割結果の粒度は非常に粗い．Web ページ中からメインコンテンツのみを抽出するためには，上記手法を適用して得られる分割結果をさらに細かく分割する必要がある．

文献 [4] [5] では，VIPS アルゴリズムと呼ばれるレイアウト情報を利用したヒューリスティクスに基づく Web ページ分割手法が提案されている．フォント情報，面積，背景色，座標など，レイアウトに関する様々なパラメータを用いた 12 個のルールを HTML タグごとに使い分けることで，Web ページをコンテンツ単位へと分割する．文献 [36] では，各 DOM ノードの座標情報をパラメータとして決定木を用いた機械学習を行うことによって，Web ページを 9 つのブロックに分割する手法が提案されている．

文献 [4] [5] [36] で提案されている手法では，Web ページを一度非常に細かいブロックまで分割した後，2 つのブロックにおけるフォントや背景色の違い，Web ページのレンダリング結果におけるブロックの面積やブロック間の距離などを利用し，フォントが同じである場合や距離が小さい場合に 2 つのブロックを結合している．しかし，それらフォントの違いやブロック間の距離が Web コンテンツの切れ目を明確に表している Web ページは少ない．また，長い文章の段落ごとには一定間隔の距離を空ける事も多いが，実際にはそれらの段落がまとまって 1 つの Web コンテンツを示している．ブロックの面積は，そのブロック内部に存在するテキストの量や画像の解像度などによって大きく変化する．そのため，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に，異なった Web ページ分割結果が作成されるという問題がある．

2.4 議論支援・会議支援に関する既存研究

議論の構造化という観点から関連研究について言及する．ここでは，IBIS（Issue Based Information System）モデル [47]，DRL（Decision Representation Language）モデル [19]，QOC（Questions, Options, and Criteria）モデル [1] について述べる．

IBIS モデルでは，Issue（問題点），Position（解決策），Argument（意見），Other（その他）という 4 種類のノードを用いて議論を木構造として表現する．木のルートは Issue である．Issue は子ノードとして 1 つ以上の Position を持つ．各 Position は子ノードとして 1 つ以上の Argument を持つ．木構造の親子関係において，それぞれのノード間は，responds-to（回答），supports（賛成），objects-to（反対）などの 9 種類のリンクによって接続される．これらのリンクには，接続するノードの制約を持つ．例えば，responds-to は Issue と Position の接続のみに用いられ，Issue と Argument を接続することはできない．supports と objects-to は Position と Argument の接続のみに用いられる．

これまでに，gIBIS [15]，Debategraph [42]，Deliberatorium [30] など，IBIS モデルに基づく議論支援システムが複数構築されてきた．IBIS は単純なモデルで議論を構造化することができるという利点があるが，表現力が乏しく，複雑な議論の構造化が困難であるという問題点があった．

DRL モデルでは，Alternative（案），Goal（目標），Claim（主張），Question（質問），Procedure（手順）という 5 種類のノードと，Is-A-Sub-Decision-Of，Is-A-Goal-For，Is-A-Subgoal-Of などの 18 種類のリンクを用いて意思決定の過程をグラフ表現することによって議論を構造化する．また，複数のノードをひとまとまりにした Group というノードも存在する．用意されているリンクの種類が多く複雑な議論も正しく構造化することが可能であり，高い表現力を持つという利点がある反面，DRL による記述は初心者にとっては困難であるという問題点がある．文献 [20] では，DRL モデルに基づく議論支援システムが提案されている．

QOC モデルでは，Question（問題），Option（解決策），Criteria（評価基準）の 3 種類のノードを用いて議論の構造化を行う．QOC モデルでは IBIS モデルと同様に，木構造を用いて議論を構造化する．木のルートは Question である．Question は子ノードとして 1 つ以上の Option を持つ．各 Option は子ノードとして 1 つ以上の Criteria を持つ．Option と Criteria は Assessment（評価）と呼ばれるリンクで接続される．Assessment には Positive Assessment と Negative Assessment の 2 種類が存在する．QOC モデルでは，Positive Assessment のリンクを実線で表現し，Negative Assessment のリンクを破線で表現する．

QOC モデルは評価基準の要素を持つため，代替案と比較検討した結果を容易に把握できるという利点が存在する．しかし文献 [37] において QOC の表現能力の限界が指摘されている．この問題に対し，文献 [61] において，QOC モデルと AHP(Analytic Hierarchy Process) を組み合わせるという手法が提案されている．

次に，会議支援という観点から関連研究について述べる．会議支援に関する研究は CSCW(Computer Supported Cooperative Work) の分野において盛んに行われている．大きく分けて，対面会議を支援する研究と，遠隔会議を支援する研究の 2 つに分類できる．対面会議とは，会議参加者が同じ場所に集まって議論を進める会議方式である．遠隔会議とは，会議参加者がそれぞれ異なる場所からネットワーク経由でリアルタイムに情報を共有し，議論を進める会議方式である．住民参画 Web プラットフォーム O_2 でサポートする会議は，対面会議を想定している．ここでは対面会議支援に関する研究について言及する．

文献 [98, 102] では多言語会議支援システムについての研究について述べられている．日本で開催される会議の参加者の大多数が日本人であるが，そのような会議において，日本語に流暢ではない外国人が参加している場合も存在する．文献 [98, 102] では多数の日本人参加者が少数の外国人参加者の会議内容理解を支援することを “All for one” と呼び，All for one 型の多言語会議支援を行うためのシステムについて構築・評価実験を行っている．All for one 型会議支援システムの実装においては “支援者同士の作業情報共有” を一つの目的としており，それを実現するために，アウェアネス情報の共有機能を開発している．アウェアネス情報の共有機能として，具体的には，(1) 色による支援者識別機能，(2) 翻訳完了ラベル機能，(3) テレポインタ機能，(4) 外国人からのフィードバック機能，の 4 つが実装されている．本研究では多言語会議の支援は想定していないが，文

献 [98,102] における (1) 色による支援者識別機能，(3) テレポインタ機能，は対面会議において意思疎通を図るにあたっては大変興味深い機能であり，本研究の遂行においても参考とした．

2.5 本研究の位置づけ

本研究では，任意の Web ページ中に存在する Web コンテンツに対してユーザが自由にアノテーションを行うことにより，Web 情報を閲覧者の観点から再構造化することを目的とする．更には，構造化された Web 情報を利用して，ユーザの知的活動を支援することを目的とする．

既存のソーシャルタギングのサービスでは，専用の Web サイトに登録されたコンテンツに対してのみタギング可能である．例えば，flickr であれば，ユーザが flickr のサイトに画像を登録する必要がある．flickr ユーザがタグを付加できるコンテンツは，flickr に登録された画像のみである．

Annotea であれば，Web ページに存在する全てのコンテンツに対してアノテーションを与えることが可能になるが，Annotea を利用するためには，ユーザは専用のクライアントを導入する必要がある．アノテーションを新たなサービスに応用するためには，ユーザから多くのアノテーションを獲得することが重要である．Annotea では，ユーザが使い慣れたブラウザを利用することができないため，多くのユーザからアノテーションを獲得することは困難である．

本研究では 2.1.3 節で解説した MiSpider と呼ばれるエージェントモデルを用いたアノテーションシステムを提案する．これにより，インターネットに接続可能な Web ブラウザさえあれば任意の Web ページに対して付箋アノテーションを行うことが可能となる．ユーザは使い慣れたブラウザに対してプロキシサーバを設定するだけで，本システムを用いて任意のコンテンツに対して付箋アノテーションを行うことができる．MiSpider のベースエージェントによって，ユーザが作成した付箋アノテーションを用いた知的処理を行うことができる．本研究ではエージェントが付箋アノテーションを分類し，似たような内容に対して貼り付けられた付箋アノテーション同士に対して双方向のリンクを作成する．これにより関連が深いと思われる付箋アノテーションが Web におけるハイパーリンク構造とは異なる新たなリンク構造を形成し，そのリンク構造は情報推薦などに使用することが可能になると期待できる．

付箋アノテーションの対象となった Web コンテンツをエージェントが抽出するために，Web ページ分割を行う必要がある．本研究で目的としている Web ページ分割は，テンプレート単位では粒度が粗すぎる．Web ページをレンダリングして得られるレイアウト情報を利用し，Web コンテンツ単位へと Web ページ分割を行うための手法を確立する必要がある．本研究で提案する Web ページ分割手法は，DOM 構造の隣接関係を利用しない．レイアウト情報に着目した既存研究の問題点として，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に，

異なった Web ページ分割結果が作成されるという問題があった．コンテンツ量に依存しない分割結果が得られる Web ページ分割手法の確立が課題であり，本研究では，閲覧者が Web ページをどのように認識し，分割しているかをシミュレートする．そのために，Web コンテンツの見出し部分に着目した分割を行う．見出し部分に着目した分割を行うことにより，既存研究の問題点を解決することが可能となることを示す．

住民参画 Web プラットフォーム O_2 のための議論支援システムに関しては，SOCIA 上に蓄積された Linked Open Data を利用することによって，関連情報提示に基づく議論支援を行う．会議支援システムでは，対面会議において参加者間の意思疎通を支援するために，ポインタ表示や発表資料表示の同期機能を実装する．ポインタ表示の同期機能は文献 [98, 102] におけるテレポインタ機能と同様の機能であるが，テレポインタ機能は会議参加者のマウスポインタの位置を共有するものであり，システム利用中は常に表示され続ける．すなわち，会議参加者がテレポインタに着目すべきタイミングを把握しづらいという問題点が挙げられる．本システムで提案するポインタ機能はユーザが任意のタイミングで呼び出す機能であり，普段はポインタ表示は行われていない．ポインタ表示の同期を行うためにはシステムをポインタモードと呼ばれるモードへと状態遷移させた後に，画面上の着目して欲しい箇所をタップする必要がある．したがって，ポインタが画面に表示された時には，他の会議参加者の誰かが画面に着目して欲しいという意思表明をしていることになる．

会議参加者全員が常に同じ資料を見ているとは限らないため，ただ単にポインタ表示の同期を行うだけでは異なる会議資料に対してポインタを表示してしまう可能性がある．本システムでは会議資料表示の同期機能も実装することで，上記課題を解決した．会議参加者は，発表者と聴講者に二分される．発表者は本システムを発表者モードへと状態遷移させた上で会議へ参加し，聴講者は本システムを聴講者モードへと状態遷移させた上で会議へ参加する．聴講者モードとなっている端末では，発表者モードの端末に表示されている会議資料が常に表示されるような仕様とした．

2.6　結言

本章ではまず，本研究の基盤となっている技術について述べた．Web インテリジェンスを実現するための基盤技術に関して言及し，それらの技術が本論文でどのように利用されているかについての説明を行った．次に，アノテーションや Web ページ分割，議論支援，会議支援に関する既存研究や既存システムについて述べた．アノテーションに関しては，学術的研究だけでなく，実際に Web 上で公開されているサービスについても言及し，それらの問題点を示した．Web ページ分割は，DOM 構造に着目した分割手法，レイアウトに着目した分割手法，という 2 つの観点から既存研究を分類し，それぞれの研究について簡単な説明を交えつつ紹介した．それら既存研究の問題点について考察を行い，本研究の位置付けを行った．

第3章　付箋アノテーションシステム

3.1　序言

本章では本研究で実装した付箋アノテーションシステムについて述べる．

Web ページ内には複数の話題に関連する文章や画像などのコンテンツが含まれている．本研究では Web ページ内の複数のコンテンツの中から任意のコンテンツを特定するための手段として，Web ページ上のコンテンツに付箋アノテーションを行うためのシステムを実現した．これによりユーザが指し示したいコンテンツをブックマークやソーシャルタグよりも正確に指し示すことが可能になる．ブックマークやソーシャルタグは閲覧者にとって Web ページを特定する手段としては利用可能であるが，Web ページ中の一部分を特定する手段としては不十分である．フラグメント識別子によって Web ページ内のコンテンツを特定可能であるが，Web ページ閲覧者が指し示したいコンテンツにフラグメント識別子が設定されていない場合には利用不可能である．

本研究では Web ページの閲覧者が Web ページ中の任意の箇所に対して付箋によるアノテーションを行い，作成された付箋アノテーションにはフラグメント識別子を与える．閲覧者がコンテンツを特定可能な付箋アノテーションを実現することで，コンテンツ間に双方向のリンクを張ることも可能になる．閲覧者がコンテンツ間に双方向リンクを張ることにより，ブックマークやソーシャルタグ以上の情報が得られると期待できる．本システムの実現には画面上での付箋アノテーションの位置と対象のコンテンツの画面上での位置が一致していることが重要である．ユーザが意図したコンテンツとアノテーションの位置を一致させるために，既存のシステムでは絶対座標を用いてアノテーションの位置を決めていた．

絶対座標を用いてアノテーションを表示すると，ユーザの Web 閲覧環境が変化した際にアノテーションが対象コンテンツからずれるという問題が発生する．すなわち複数のユーザ間におけるアノテーションの共有に支障がある．本研究ではユーザの Web 閲覧環境が変化しても対象コンテンツからずれることのない付箋アノテーションの実現方法を示す．本システムでは付箋アノテーション付きのコンテンツ間に存在する関連をユーザ間で共有するために，付箋アノテーション間に双方向リンクを作成する．ユーザが作成した付箋アノテーションを監視し，関連が高いと思われるコンテンツに対して作成された付箋アノテーションに対して自動的に双方向リンクを作成するエージェントを実現する．

本章ではまず，システムの構成を示した後，スクリーンショットを交えながらシステムの動作例を示し，付箋アノテーションシステムの説明を行う．次に，ユーザの Web 閲覧

環境に依存しない付箋アノテーションの表示手法について解説を行う．付箋アノテーション間の双方向リンクモデルを提案し，双方向リンクを作成するための biLink エージェントについて述べる．実験を行い本システムの評価を行った後，本章をまとめる．

3.2　付箋アノテーションシステムの概要

人が書物を参照するときには，何度も参照する頁に対して付箋を貼付けるという動作を行う．WWW 上の情報を参照するときには，Web ブラウザのブックマーク機能を用いて，頻繁にアクセスする Web ページをブックマークに登録する．書物に対して貼付ける付箋が貼付ける場所によって頁内の特定の場所を指定することが可能であるのに対し，ブックマーク機能は Web ページのタイトルと URL のみを保存しておくだけのものである．ブックマーク機能は Web ページ内の特定の場所を指定することを目的としていない．近年，Web ページのマルチコンテンツ化が進んでいる．Web ページのマルチコンテンツ化とは，1 つの Web ページ内にヘッダやフッタ，広告，複数の記事など，多様な内容が含まれることを意味する．マルチコンテンツの Web ページにおいてユーザが目的の情報を閲覧するためには，ブックマークから Web ページを開いた後に手作業で目的の情報を探す必要があり，煩雑である．Web ページの一部を参照するためのしくみが求められている．

Web ページ内のコンテンツを指定する方法として，HTML には `id` 属性や `name` 属性が存在する．`id` 属性や `name` 属性は，フラグメント識別子として使用することが可能である．通常これらの属性は Web ページの作成者によって指定されるものである．本研究では，Web ページの閲覧者が既存の Web ページ内のコンテンツに対して，自由にフラグメント識別子を指定することが可能なシステムを提案する．ユーザ自身がフラグメント識別子を指定することにより，Web ページのタイトルと URL のみならず，ユーザが Web ページ上のどのコンテンツに着目しているかを保存することを可能とする．本研究におけるコンテンツとは，Web ページ内で提供されるテキストや画像，企業広告などの，各種情報のことである．本システムでは，付箋を貼付けるというインターフェイスを用いて，ユーザがフラグメント識別子を指定することを可能にする．

書物に対して付箋を貼付ける場合，その付箋に対してメモ書きを行うことがある．それと同様に，本システムでは，貼付けた付箋に対してユーザが自由にコメントを付けられる機能を用意する．これにより，ユーザが Web ページ上のコンテンツに対して付箋を貼付け，その付箋を用いて，コンテンツに対してアノテーションを与えることを可能にする．付箋という形でアノテーションを可視化することを，本論文では付箋アノテーションと呼ぶ．ユーザは，Web ページ内に含まれるテキストデータや画像など，任意のコンテンツに対して付箋によるアノテーションを行うことができる．

3.2.1 システム構成，およびシステム利用の流れ

図 5.1 は本システム構成の概略と，システム利用の流れを示している．まずは，本システムの構成についての説明を行う．本システムは MiSpider と呼ばれるエージェントモデルに基づき，Web ブラウザ上で動作するページエージェントと，サーバ上で動作するベースエージェントから構成される．ページエージェントは付箋アノテーションシステムのフロントエンド（付箋クライアント）として動作する．すなわち，ユーザに対して付箋アノテーションを作成するためのインターフェイスを提供するシステムである．付箋クライアントは JavaScript で実装されており，Web ブラウザ上で動作する．実装にあたっては，クロスブラウザ対応を強く意識した．したがってユーザは，JavaScript に対応した Web ブラウザさえあれば，特別なプラグインをインストールすることなく本システムを利用することができる．付箋クライアントはマウスイベントとキーイベントを監視している．ユーザは付箋アノテーションを行いたいコンテンツにマウスポインタを合わせ Shift キーを押しながらマウスをクリックすることにより，目的のコンテンツに対して付箋アノテーションを作成することができる．付箋サーバは本システムのバックエンドであり，プロキシモジュール，付箋データベース，および双方向リンク作成のためのベースエージェントから構成される．

次に，本システム利用の流れについて述べる．付箋アノテーションシステムを利用するための前準備として，ユーザは Web ブラウザのプロキシサーバに，付箋サーバのホスト名を入力しておく必要がある．ユーザが本システムに Web ページ取得の要求を行うと，図 5.1(1)(3) に示すように，Web ブラウザは付箋サーバ内のプロキシモジュール経由で Web ページを取得する．付箋サーバ内部には，過去に付箋アノテーションを作成した Web ページの URL と HTML が保存されたデータベース（以下，付箋データベース）がある．プロキシモジュールは付箋データベースにアクセスし，ユーザがアクセスしようとしている Web ページの HTML が付箋データベースに保存されているかどうかを問い合わせる．付箋データベースに HTML が保存されている場合には，その HTML を Web ブラウザに送信する．保存されていない場合には，Web サーバから新たに HTML を取得する．新たに取得した HTML に対して，図 5.1(2) に示すように，付箋クライアントを付加してから Web ブラウザに送信する．その際，文字コードを UTF-8 へと変換する．プロキシサーバで文字コードを変換する理由については，3.3.2 節で述べる．付箋クライアントを付加するためには，HTML の `head` タグ直下に，以下の `script` タグを埋め込むだけでよい．

`<script type='text/javascript' src=`付箋クライアントの `JavaScript` ファイルの `URL />`

既存の Web ページに対して Web サービスを付加する方法として，プロキシを利用する手法 [67] と，ブックマークレットを利用する手法 [86] がある．JavaScript で作成したプログラムを特定の方法でパッケージ化することで，ブラウザの機能拡張として実装することも可能である [2, 8]．本システムではプロキシを用いて Web ページに対して付箋

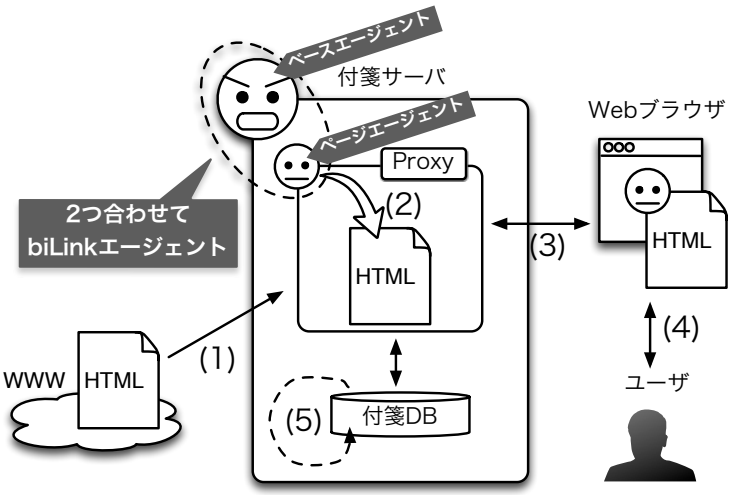


図 3.1: 付箋アノテーションシステム利用の流れ

クライアントを付加することとした．ブックマークレットを採用しなかった理由として，ユーザが Web ページ読み込みの度にブックマークレットを起動する必要があり，面倒である点が挙げられる．Web ブラウザの機能拡張として実装しなかった理由として，Web ブラウザごとにプラグイン化するための設定ファイルを記述する必要があり，開発の手間になるからである．

付箋サーバのプロキシモジュールを経由することによって，ユーザの Web ブラウザに対しては付箋クライアントが付加された Web ページが送信される．ユーザは付箋クライアントとのインタラクションを行い（図 5.1(4) 参照），Web ページ中の任意のコンテンツに対して付箋アノテーションを行うことが可能となる．

Web ページの遷移が発生した時やユーザが Web ページを閉じた時，付箋クライアントは `onunload` イベントを取得して，ユーザが作成した付箋アノテーションの保存を試みる．本システムでは，ユーザが付箋アノテーションを作成した際に，Web ページの HTML を動的に書き換え，HTML タグの追加を行っている．付箋データベースには，Web ページの URL と，HTML タグを追加した結果の HTML 全てを保存する．このように，本システムでは HTML 全てを保存するが，その HTML にはすでに `script` タグが付加されている．過去に付箋アノテーションを作成した Web ページを Web ブラウザに送信する場合，即ち，付箋データベースから HTML を取得して Web ブラウザに送信する場合には，`script` タグを付加する必要はない．

本節では図 5.1(5) に関しての説明を省略した．図 5.1(5) はユーザから獲得した付箋アノテーションを利用してベースエージェントが知的処理を行うフェーズであり，3.4 節で詳細を述べる．

3.2.2 実行例

実際に本システムを用いてWebページ内部に付箋アノテーションを行った直後のスクリーンショットを図3.2に示す．図3.2の上はApple社のWebブラウザであるSafariを使用してGoogleニュース[1]を開き，ニュース記事2つに対して付箋アノテーションを作成した画面である．

ユーザは図3.2の上のようにコンテンツに対して付箋アノテーションを行うことができる．作成した付箋アノテーションをダブルクリックすることにより，その付箋アノテーション情報の詳細をウィンドウ(以下，付箋詳細ウィンドウ)で閲覧可能である．この実行例では2つの付箋アノテーションを作成している．ここではその2つの付箋アノテーションのうち，右上の付箋アノテーションの付箋詳細ウィンドウが表示されている．付箋詳細ウィンドウを拡大したものを図3.2の下に示し，(1)〜(6)の部分について説明する．(1)は付箋アノテーションを行った日時の表示である．(2)は付箋アノテーションの色を選択するために使用する．(3)はコンテンツに対する注釈であり，コンテンツに対するタグとして用いることが可能である．ここに入力された注釈は付箋アノテーションの画像の上にも重ねて表示される．(4)は関連する付箋アノテーションへのリンク表示であり，表示されている文字列をクリックすることにより，付箋アノテーションが行われたコンテンツと関連しているコンテンツに対して作成された付箋アノテーションを開くことができる．なお，(5)は付箋アノテーションを削除するためのボタンであり，(6)は付箋詳細ウィンドウを閉じるためのボタンである．

図3.2ではWebページ上のテキストデータに対して付箋アノテーションを行っているが，本システムではテキストデータだけではなく，画像やFlashファイルなど，Webページ上の任意のコンテンツに対して付箋アノテーションを行うことが可能となっている．図3.3に，画像に対して付箋アノテーションを行った例を示す．画像に対して付箋アノテーションを行う場合，その画像の左上からの相対座標を用いて表示しているため，画像の中の好きな箇所に対して付箋アノテーションを行うことが可能となっている．図3.3では，NISSANが開発したスポーツカーGT-Rのボンネットに対してユーザが付箋アノテーションを行い，コメントとしてGT-Rのエンジンの名前を記述している．

最後に，図3.4に双方向リンクの例を示す．図3.4の上では，Webブラウザを用いて“日航機無断滑走、「防氷液」の効果切れ迫り機長らに焦りか”というタイトルがつけられたニュース記事を閲覧している．この図の中では，ユーザがニュース記事に対して付箋アノテーションを行っている．付箋詳細ウィンドウには，2つのリンクが表示されている．2つのリンクの上側は“「テークオフ」を誤解？完成指示復唱せず　新千歳空港”というタイトルのWebページに対して作成された付箋アノテーションへのリンク，下側は“JAL国際線、軽食用カートをトイレに収納 食事は提供”というタイトルのWebページに対して作成された付箋アノテーションへのリンクである．ここでユーザが上側のリンクをクリックした場合，図3.4の下図に示すように，対象のWebページに対して作成さ

[1]Googleニュース, http://news.google.co.jp/

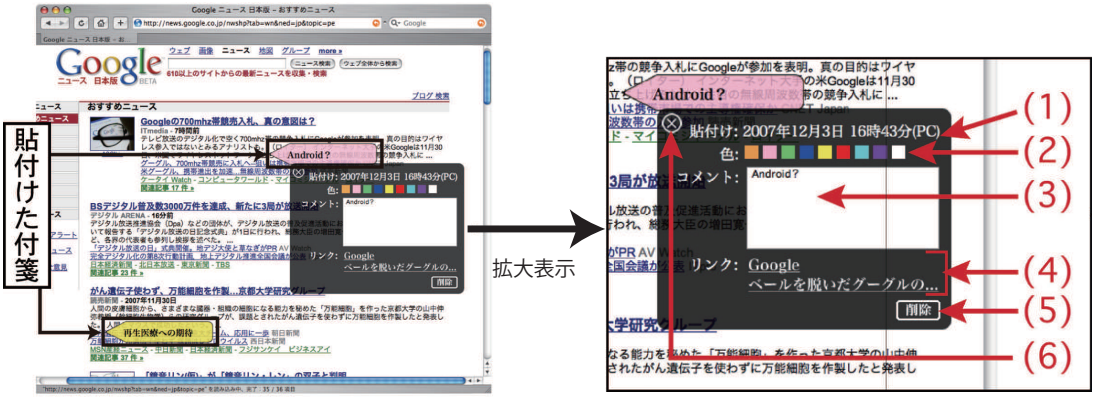


図 3.2: 付箋アノテーションシステムの実行例

図 3.3: 画像に対して付箋アノテーションを行ったスクリーンショット

図 3.4: 双方向リンクを辿った例

れた付箋アノテーションが Web ブラウザの一番上に表示されるようにリンク遷移をする．このようにして，エージェントが付箋アノテーションに対して自動的に作成したリンクをクリックすることで，似たようなコンテンツに対して作成された付箋アノテーションへとページ遷移することが可能となっている．本リンクは双方向のリンクとなっているため，図 3.4 の下図の付箋アノテーションで付箋詳細ウィンドウを開いた場合，もとの Web ページ（“日航機無断滑走、「防氷液」の効果切れ迫り機長らに焦りか” というタイトルの Web ページ）へのリンクが表示される．そのリンクをクリックすることで，さっきとは逆に，図 3.4 の下図の付箋アノテーションから図 3.4 の上図の付箋アノテーションへとページ遷移することが可能となっている．このようにして本リンクを辿ることにより，ユーザは関連するコンテンツ間を相互にトラバースすることが可能となり，Web 閲覧の効率が高まることが期待できる．本システムの応用として，他のユーザと付箋アノテーションを共有することも考えられる．エージェントが他のユーザとの付箋アノテーション間に双方向リンクを作成するようにすることにより，他のユーザが着目している，内容が類似したコンテンツを発見することが可能となる．

3.3 Web コンテンツの同定

3.3.1 絶対座標を用いた付箋位置決定手法の問題点

付箋アノテーションの実現において，ユーザの閲覧環境が異なったとしても同一の付箋アノテーションが同一のコンテンツを示すことが重要である．既存のシステムでは絶対座標を用いてアノテーションの表示位置を決定している．しかしコンテンツの表示位置は，Web ブラウザのレンダリング結果に依存する．Web ブラウザのウィンドウサイズやフォントサイズなどが変更された際には，コンテンツの表示位置も変化する可能性がある．図 3.5(a) は，Web ブラウザである Firefox を用いて，Google ニュースを開いた画面を左右に並べたものである．ウィンドウサイズは左右同じであるが，右側の表示では左側よりもフォントサイズを大きくしている．左右のスクリーンショットを比較すると，同じ Web ページを表示しているにも関わらず，コンテンツの表示位置が大きく異なっている．コンテンツの表示位置が変化するにも関わらず絶対座標を用いてアノテーションの表示位置を決定すると，アノテーションが元々指し示していたコンテンツからずれてしまうという問題がある．

絶対座標を用いて付箋アノテーションを表示した例を，図 3.5(b) に示す．これは図 3.5(a) で表示しているニュース記事に対して付箋アノテーションを作成しているが，図 3.5(b) の左と右では，付箋アノテーションが同じニュース記事を指し示していないのが確認できる．これではアノテーションの意味を成さない．本システムでは付箋アノテーションを表示する Web ページ内の位置情報として絶対座標を使用しない．コンテンツの DOM ノードに対して付箋アノテーションを行うという手法を採用した．これにより，コンテンツの表示位置が変化しても，付箋アノテーションの表示位置も Web コンテンツに追従

(a) コンテンツの表示位置が異なる例

(b) 絶対座標に基づいて付箋を表示

(c) DOM ツリーに基づいて付箋を表示

図 3.5: 絶対座標に基づく表示手法と DOM ツリーに基づく表示手法の比較

図 3.6: テキストノードの分割

して移動する．図 3.5(c) は，本手法を用いて付箋アノテーションを表示したスクリーンショットである．図 3.5(b) では付箋アノテーションが対象コンテンツからずれていたが，図 3.5(c) では同じコンテンツを指し示していることが確認できる．

3.3.2 DOM ツリーに基づく付箋位置決定手法

本システムでは Web ページの左上を基準とした絶対座標を用いて付箋アノテーションを表示するのではなく，Web ページを構成する HTML の DOM ノードに対して付箋アノテーションの HTML タグを付加し，表示する．ユーザが作成した付箋アノテーションの情報は付箋データベースに保存される．DOM ノードに対して付箋アノテーションの HTML タグの付加を行っているため，ユーザの閲覧環境が異なったとしても，同一の付箋アノテーションが同一のコンテンツを示す．ユーザが過去に付箋アノテーションを作成した Web ページが更新された場合には，付箋アノテーションの対象となった DOM ノードのパスを比較することにより，付箋アノテーションの追従を試みる．

ただ単に先ほど述べた手法を適用しただけでは，長いテキストノードに対して付箋アノテーションの作成を行う場合に，ユーザが意図したところに付箋アノテーションを作成することができないという新たな問題が発生する．テキストノードの左上箇所にしか付箋アノテーションを行うことができず，ユーザは自由なアノテーションを行うことができない．本研究では，テキストノードに対してのアノテーションであれば，付箋アノテーションを作成する前にそのテキストを細かい span 要素へと分割する．本システムの初期の実装段階において，テキストノードを一文字ずつ span 要素へと分割し，文字単位での付箋アノテーション作成を可能にすることを試みた．これによりユーザの意図したところに正確に付箋アノテーションを行うことが可能となる．しかし一文字ずつ span 要素に分割すると，Web ブラウザのレンダリングエンジンの実装上の都合により，テキストの折り返しが行われないという問題を確認した．これは，Web ブラウザは連続するインライン要素を左右に並べて表示するためである．具体例を図とともに示す．図 3.6 上部は，Yahoo!ニュースで配信されていたニュース記事の Web ページである．図 3.6 上部の赤枠部分はニュース記事本文であるが，ニュース記事本文に対して span 要素へ分割せずに付箋アノテーションを行うと，常に赤枠の左上部分に付箋が表示されてしまう．このままではニュース記事本文中のユーザが意図した箇所に対して付箋アノテーションを行うことが不可能であるため，ニュース記事部分を span 要素へ分割する．図 3.6 上部のニュース記事部分を一文字ずつ span 要素へと分割した結果が，図 3.6 左下部である．図 3.6 上部と左下部のニュース記事部分を比較すると，折り返しの位置が大幅に変化していることが確認できる．図 3.6 左下部では折り返しが行われておらず，ニュース記事のテキストが右の広告部分に重なって表示されてしまっている．

上記課題を解決するために，以下の 2 つの手法が考えられる．1 つ目は，span 要素を包含するブロックレベル要素のスタイルの word-wrap プロパティに対して “break-all” を設定し，ブロックレベル要素の領域に合わせて強制的に折り返しを行うという手法である．しかしスタイルを変更すると，Web ページによっては Web デザイナーが意図したレイアウトと大幅に異なるレンダリング結果が得られる可能性がある．本システムによって既存の Web ページのスタイルを変更することは好ましくない．2 つ目は，テキストノードを形態素ごとに span 要素へと分割する手法である．span 要素を包含するブロックレベル要素の右端に 2 文字以上文字を含む span 要素が配置されると，span 要素の途中で折り返し表示される．本手法では，折り返し箇所は元の Web ページから少しずれてしまうというデメリットがある．図 3.6 右下部は，図 3.6 上部のニュース記事部分を形態素ごとに span 要素へと分割した結果である．図 3.6 上部と図 3.6 右下部を比較すると，折り返し箇所に少々ずれはあるが，レイアウトが然程崩れていないことが確認できる．

本システムでは 2 つ目の手法を採用した．本システムの応用で要求される付箋アノテーション作成の精度は，形態素単位で十分である．分割の結果生成された要素に対して付箋アノテーションの HTML タグを付加することで，テキストノードに対しては形態素単位でのアノテーション作成を可能とした．形態素ごとに分割するメリットとして，形態

素ごとに付箋アノテーション可能となり，本システムの応用においてアノテーション対象となった単語をユーザの嗜好を表わすキーワードとして利用するといったことが可能となる．形態素解析ごとに付箋アノテーションを作成できるということは，ユーザの知的活動支援を行うために有用である．

英語やフランス語といった言語ではわかち書きが行われるため，スペースをセパレータとしてテキストノードを span 要素へと分割すればよい．しかし，日本語，中国語，朝鮮語，タイ語といった言語ではわかち書きが行われず，単語間の区切りが明示されない．わかち書きが行われない言語では，テキストノードの形態素解析を行う必要がある．本システムが対象とする Web ページの言語は英語と日本語であり，日本語のテキストノードに対して付箋アノテーションを行う場合には，本システムでは付箋サーバ側で形態素解析を行う．本システムでは形態素解析のために MeCab [64] を用いているため，日本語以外の言語には対応していない．サーバ側で言語を判定しそれぞれの言語に対応した形態素解析システムを動作させることによって，多言語対応も可能である．

図 3.7 は，付箋アノテーション作成の際に行われる，Web ブラウザと付箋サーバ間のデータ送受信の例である．図 3.7 では，Web ページ中に存在する “いい天気です。” というテキストデータの “天気” にマウスカーソルを合わせ，付箋アノテーションを行ったというシチュエーションを想定している．ユーザが利用している Web ブラウザにおいて，ページエージェントは，ユーザがクリックしたテキスト “いい天気です。” を XMLHttpRequest を利用して付箋サーバ側にポストする．付箋サーバ側ではベースエージェントが MeCab を用いて形態素解析を行う．形態素解析を行うに当たっては文字コードの問題が存在するが，本システムではプロキシサーバで全ての HTML ファイルを UTF-8 へと変換している．したがって，サーバ側で受け取ったテキストデータは全て UTF-8 であるという前提で形態素解析を行えばよい．ベースエージェントは形態素解析の結果を，XMLHttpRequest のレスポンスとしてページエージェントへと送信する．ページエージェントは XMLHttpRequest のレスポンスを元に，テキストデータを形態素ごとに span 要素へと分割する．分割が完了した後に，付箋アノテーションを表現するための HTML タグを挿入する．

ページの遷移が発生した時や，ユーザがページを閉じた時，付箋クライアントは onunload イベントを取得して，ユーザが作成した付箋アノテーションの保存を試みる．本システムではユーザが付箋アノテーションを行った際に HTML タグの追加を行っている．従って今回は，Web ページの URL と HTML タグを追加した結果の HTML を，付箋データベースに保存することとした．

3.3.3 DOM ツリーの変化に対する付箋の追従

過去に付箋アノテーションを行った Web ページが更新された場合には，付箋アノテーションを行った DOM ノードのパスを更新前と更新後の HTML ファイルで比較することにより，付箋アノテーションの対象となったコンテンツの追跡を試みる．ここではパスの比較を行い，ユーザが付箋アノテーションを行ったコンテンツの追跡を行うアルゴリ

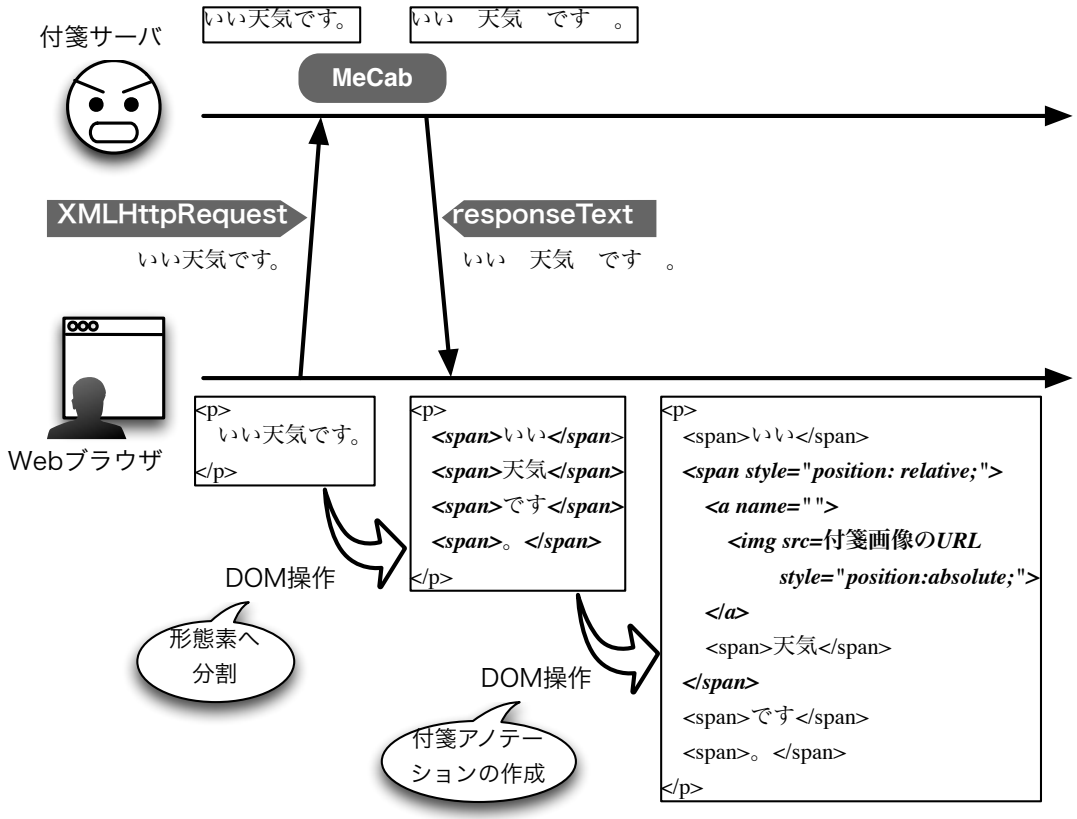


図 3.7: 形態素への分割，及び付箋アノテーション用 HTML タグ挿入の流れ

ズムについて考察を行う．

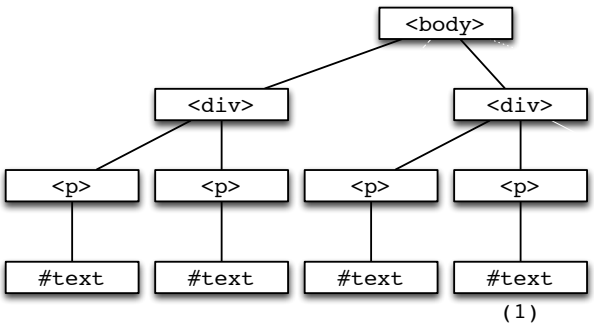
図 3.8(a) は DOM ツリーに基づき HTML の構造を図示したものである．この HTML において (1) はテキストノードである．今，(1) のノードに対して付箋アノテーションを行ったと仮定する．(1) のノードからボトムアップ式に親ノードを辿っていくことにより，(1) のノードのパスを得ることができる．XML Path Language(XPath) の式を用いて記述すると，(1) のノードのパスは，

```
/body/div[2]/p[2]/#text
```

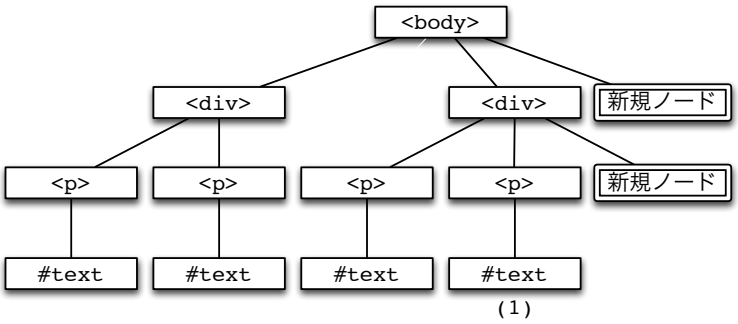
となる．

ここで HTML の更新があり，図 3.8(b) のように新たなノードが追加されたとする．この場合には (1) のパスに変化はないため，影響はない．つまり，DOM ツリーにおいて，ルートから目的のノードへのパスよりも右側へノードが追加された場合には目的のノードへのパスは変化しないため，特別な処理を必要としない．

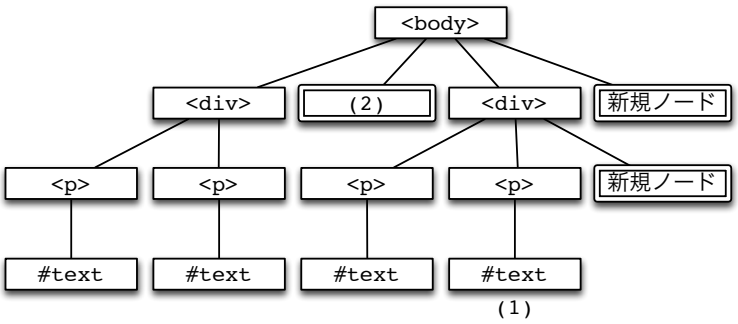
次に，ルートから目的のノードへのパスよりも左側へノードが追加された場合を考える．図 3.8(c) はノード追加後の構造を図示したものであり，新たなノードとして (2) に`<div>`が追加されたとする．この場合，(1) のノードへのパスは，



(a) ベースとなる DOM ツリー



(b) ノード追加後 その 1



(c) ノード追加後 その 2

図 3.8: HTML 更新による DOM ツリーの変化

```
/body/div[3]/p[2]/#text
```

となり，ユーザが付箋アノテーションを行った時とはパスが変化している．しかし，図 3.8(c) の (2) に 以外のノードが追加された場合には，(1) のノードのパスは，

```
/body/div[2]/p[2]/#text
```

のままであり，ユーザが付箋アノテーションを行った時からパスは変化していない．従っ

て，ルートから目的のノードへのパスよりも左側へノードが追加された場合には，追加されたノードに注意する必要がある．

ルートから目的のノードへのパスよりも左側へノードが追加された場合は，DOM ノードのパス間の類似度や，テキストマッチングを用いてコンテンツの追跡を行えばよい．先ほどの DOM ツリーを例に考えると，

```
/body/div[*]/p[*]/#text
```

のパスのノードを全て取得し，それらのノードにおいてテキストマッチングを行うことで，付箋アノテーションの対象となったコンテンツを発見することができる．

しかしテキストによっては，マッチングの対象が本来のものとは異なってきてしまう問題がある．例として，日記のような Web ページを考える．Web ページ中に,「今日は晴れだった.」というテキストノードがあるとする．このテキストノードを (A) とする．ユーザが (A) に対して付箋アノテーションを行うことを考える．Web ページが追加更新され,「今日は晴れだった.」という新しいテキストノードが追加されたとする．この新しいテキストノードを (B) とする．この場合,「今日は晴れだった.」というテキストを用いてテキストマッチングを行うと，(A) と (B) の両方に成功してしまい，元々付箋アノテーションの対象となっていたコンテンツを一意に特定することができない．本システムでは，付箋アノテーションの対象コンテンツの候補として，マッチングが成功したコンテンツ全てをユーザに提示することを考える．複数の対象とマッチングが成功した場合には，それらのコンテンツに対して付箋の表示ではなく色付けを行う．色付けを行うことにより，ユーザにおおまかな場所を知らせることが可能である．

3.4　Web コンテンツ間へのリンク作成

3.4.1　付箋間の双方向リンク

ブログにはトラックバックという機能が存在する．トラックバックとは別のブログへリンクを張った際に，リンク先のブログに対してリンクを張ったことを通知する仕組みのことである．ブログ作者が別のブログの記事を参照して自身のサイトにコメントを掲載するような場合，元の記事へのリンクを張るのが一般的だが，単にリンクしただけでは元の記事の作者はどの Web ページからリンクされているのか容易に知ることはできない．トラックバックすることにより，リンク元記事の URL やタイトル，内容の要約などがリンク先のブログに対して送信される．トラックバックによってブログに双方向性が生まれる．本システムではトラックバックの考え方に基づき，関連するコンテンツ間を相互にトラバース可能なリンクを自動生成する．このリンクを本論文では双方向リンクと呼ぶ．

本システムでは，biLink エージェントと呼ばれる Web エージェントがユーザの作成した付箋アノテーションを管理している．biLink エージェントは，MiSpider [40, 51] と呼ばれる Web エージェントモデルに基づき，ページエージェントとベースエージェントか

ら構成される．ページエージェントは，ユーザが付箋アノテーションを行ったコンテンツの内容をベースエージェントに送信する．ベースエージェントでは，ページエージェントから受信した情報から付箋アノテーションの分類を行い，関連が高いと思われるコンテンツに対して作成された付箋アノテーション同士に双方向リンクを作成する．次節では biLink エージェントについて詳しく述べる．

3.4.2 biLink エージェント

biLink エージェントは付箋アノテーションに対して双方向リンクを与える機能を持つ．biLink エージェントはユーザの付箋アノテーションの作成を監視している．ユーザが付箋アノテーションを行うと，ページエージェントは付箋アノテーションの対象となったコンテンツを抽出する．抽出されたコンテンツは，ベースエージェントに送信される．ユーザが付箋アノテーションを行ったコンテンツが画像や Flash ファイルなどの場合，ページエージェントは，DOM ツリーに基づき付箋アノテーションの対象となったコンテンツ周辺のテキストをベースエージェントへと送信する．これは，画像や Flash 周辺には，その説明となるテキストが多いというヒューリスティクスに基づいている [54]．ベースエージェントは受信したコンテンツの内容を解析し，付箋アノテーションの分類を行う．付箋アノテーションの分類手法に関しては 3.5 節で詳しく述べる．そして，関連性が高いと思われるコンテンツに作成された付箋アノテーション同士に，自動的に双方向リンクを作成する．関連研究として，アンカー前後のテキスト情報を用いて Web ページの自動分類をする研究 [6] や，ページ間の内容の類似度を用いて Web ページへのリンクを自動生成する研究 [88] が挙げられる．文献 [6] では，アンカー前後の文章がリンク先の Web ページのキーワードを含むということに着目した．アンカー前後のテキスト情報が Web ページの分類に使用できると考え，実際にシステムを実装して評価実験を行っている．文献 [88] では，リンク元がリンク先と関連する部分となるように関連リンクを構築することを目的としている．キーワード検索システムによる最上位の検索結果をリンク元とし，検索対象 Web ページ群を分類した後の各クラスタの任意の Web ページをリンク先ページとするリンクを自動的に構築する．

図 3.9 は，biLink エージェントの動作の例である．この例では，ユーザが本システムを用いて，(1)〜(6) の計 6 枚の付箋アノテーションを行ったとする．(A) は付箋の全体集合であり，biLink エージェントは，これらの付箋アノテーションの分類を行う．図 3.9 で，矢印は付箋アノテーションの分類を表す．これらの付箋アノテーションが，(a) = {(2), (4)}，(b) = {(1)，(5)，(6)}，(c) = {(3)} の 3 つに分類されたとする．

biLink エージェントはこの分類の結果をもとに，付箋アノテーション間に双方向リンクを作成する．(a) には (2) と (4) の付箋アノテーションが存在するため，これら 2 つの付箋アノテーションを双方向リンクで結ぶ．(b) には (1) と (5) と (6) の付箋アノテーションが存在するため，これら 3 つの付箋アノテーションを双方向リンクで結ぶ．(c) には (3) の 1 つの付箋アノテーションしか存在しないため，リンクを作成する必要はない．

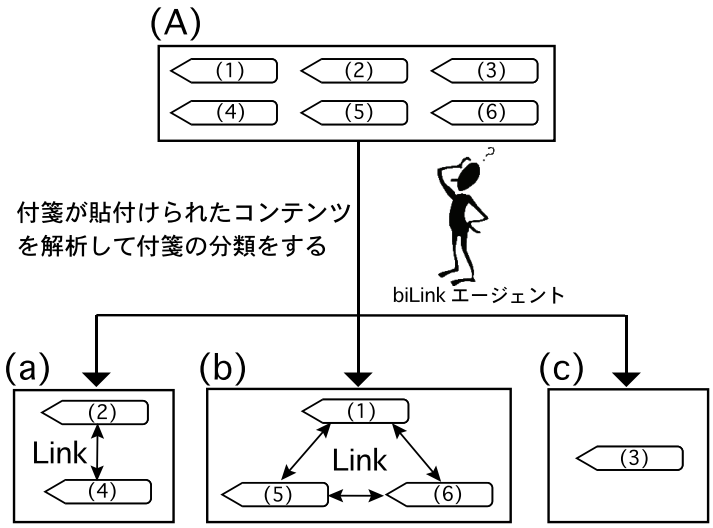


図 3.9: biLink エージェントの動作例

3.5　付箋の分類

biLink エージェントは付箋アノテーション間に双方向リンクを作成する際に，アノテーション対象となったコンテンツをもとに付箋アノテーションをクラスタに分類する必要がある．付箋アノテーションの分類は，以下のような流れで行う．

biLink エージェントはまず，ユーザが初めて作成した付箋アノテーションの前後の文章を解析し，文書ベクトルを計算してそれを 1 つのクラスタとする．次に新しく作成した付箋アノテーションの前後の文章を解析し，文書ベクトルを計算する．先ほど求めた文書ベクトルと，既存のクラスタの文書ベクトルとの距離を比較する．既存のクラスタとの距離が小さいのであれば付箋アノテーションをそのクラスタに追加し，クラスタのベクトル更新をする．さらに，クラスタ内部で双方向リンクを作成する．既存のクラスタとの距離が大きければ，新しくクラスタを作る．以降付箋アノテーションが行われるたびに文書ベクトルを計算し，上記の作業を繰り返す．

ここでは文書ベクトルの作成と付箋の類似度計算について詳しく述べる．

biLink エージェントが付箋アノテーションの分類を行う手法について述べる．biLink エージェントは付箋アノテーションが存在する Web ページを，MeCab を用いて形態素解析して文書の索引語を決定する．本研究では実装を簡単にするために，索引語の評価値として $tf \cdot idf$ の値を用いた．$tf \cdot idf$ は文書中の特徴的な単語を抽出するためのアルゴリズムである．$tf \cdot idf$ は単語の出現頻度 tf と逆出現頻度 idf の 2 つの指標を掛け合わせたものとなる．tf，idf は以下の式を用いて計算できる．

$$tf_i = \frac{n_i}{\sum_{k=1}^{n} n_k}$$

$$idf_i = \log \frac{|D|}{|\{d : d \ni t_i\}|}$$

procedure *classifyStickies(fusen, G, D)*
fusen: 付箋アノテーション
$G = \{d_1, d_2, .., d_n\}$: 既存のクラスタ集合
$D = \{d_1, d_2, .., d_n\}$: 既存のクラスタのベクトル

1: $d_f \Leftarrow documentVector($ *fusen* $)$
2: // 既存のクラスタ数が 0
3: **if** $n = 0$ **then**
4: // *fusen* を一つ目のクラスタとする
5: $g_1 \leftarrow \{$ *fusen* $\}$
6: $d_1 \leftarrow d_f$
7: $G \leftarrow \{g_1\}$
8: $D \leftarrow \{d_1\}$
9: **else**
10: $max \Leftarrow -\infty,\ s \Leftarrow 0$
11: // コサイン類似度が最も大きいクラスタを探す
12: **for all** $d_i \in D$ **do**
13: **if** $max < cos(d_i, d_f)$ **then**
14: $max \Leftarrow cos(d_i, d_f),\ s \Leftarrow i$
15: **end if**
16: **end for**
17: **if** $max > \theta$ **then**
18: // クラスタに追加，ベクトル更新
19: $g_s \leftarrow g_s \cup \{$ *fusen* $\}$
20: $d_s \leftarrow averageVector(g_s)$
21: **else**
22: // *fusen* を新たなクラスタとする
23: $g_{n+1} \leftarrow \{$ *fusen* $\}$
24: $d_{n+1} \leftarrow d_f$
25: $G \leftarrow G \cup \{g_{n+1}\}$
26: $D \leftarrow D \cup \{d_{n+1}\}$
27: **end if**
28: **end if**

図 3.10: 付箋分類アルゴリズム

n_i は単語 i の出現頻度，$|D|$ は全文書数，$|\{d : d \ni t_i\}|$ は単語 i を含む文書数である．idf は特定の文書にしか出現しない単語の重要度を上げる役割を果たしている．

本研究において tf を求める際のベースとなる文書は，付箋アノテーションの対象となったコンテンツの存在する Web ページである．本システムは Web をベースとしたアプリケーションであるため，idf を求める際の全文書数として，Yahoo! API [48] で検索可能な全ての Web ページの数を用いた．idf を求める際の索引語の登場する文書数として，その索引語の Yahoo!API での検索結果の数を用いた．

付箋アノテーションの分類を行う際の文書の類似度は，ベクトル空間モデルに基づき 2 つの文書ベクトルのコサイン尺度から求める．文書ベクトルは，各次元に索引語を割り当て，各成分に索引語の評価値を割り当てたものとするが，付箋アノテーションの対象となったコンテンツに対して重みをかけて計算する．ここでのコンテンツとは，実際に付箋アノテーションが行われた DOM 要素から DOM ツリーを親ノードへたどっていき，直近のブロックレベル要素とする．

本システムにおける文書ベクトルの計算式は次のようになる．

$$d_i = (w_{i1}, w_{i2}, \cdots, w_{iM})^T + \alpha(v_{i1}, v_{i2}, \cdots, v_{iM})^T$$

ここで，w_{ij} は i 番目の文書における語 t_j の評価値 ($j = 1, 2, \cdots, M$)，v_{ij} は i 番目の文書の付箋アノテーション対象となったコンテンツにおける語 t_j の評価値，M は文書集合に含まれる異なる語の数，α は重みである．本システムでは付箋アノテーションの対象となったコンテンツに対して適度な重みをかけて文書ベクトルを計算している．予備実験では

$$d_i = (w_{i1}, w_{i2}, \cdots, w_{iM})^T$$

という一般的な文書ベクトルの計算式を使用したが，これでは複数の話題が含まれている Web ページでは適切な分類を行うことができなかった．したがって本研究では，付箋アノテーションの対象となったコンテンツに対して適度な重みをかけて文書ベクトルを計算し，その結果生成された文書ベクトルを用いて分類を行うことで，複数の話題が含まれている Web ページにおいても適切な分類を行うことが可能となった．

次に，2 つのベクトルを d_1，d_2 とした場合，コサイン尺度は次の式で表現される．

$$cos(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1|| \cdot ||d_2||}$$

コサイン尺度は 2 つのベクトルの角度が小さい場合に大きな値を取る．付箋アノテーションを分類した結果生じたクラスタについては，クラスタに含まれる Web ページの文書ベクトルの平均を求め，それをクラスタの文書ベクトルとする．すなわち，クラスタ C の文書ベクトル d_C は，

$$d_C = \frac{\sum_{i=1}^{n} d_i}{n}$$

の式で与えられる．

biLink エージェントが付箋アノテーションの分類を行うために用いる手続き *Classify-Stickies* を，図 3.10 に示す．*ClassifyStickies* では，ユーザが初めて付箋アノテーションを行ったコンテンツを解析し，文書ベクトルを求め，それを 1 つのクラスタとする．以降，新しく付箋アノテーションの対象となったコンテンツを解析し，文書ベクトルを求め，その文書ベクトルと，既存のクラスタの文書ベクトルとの類似度を計算する．類似度が閾値 θ よりも大きければ付箋アノテーションをそのクラスタに追加し，クラスタのベクトル更新をする．類似度が閾値 θ を下回っていれば新しくクラスタを作成し，付箋アノテーションをそのクラスタに分類する．

重み α と類似度の閾値 θ は，人間が作成した正解例に基づき決定した．ランダムに収集した Web ページから，正解例と同様の分類結果になるような重み α と類似度の閾値 θ を決定した．結果，α が 25，θ が 0.15 となった．

ユーザが本システムの使用を開始した時点では，付箋のアノテーションの数は 0 であるため，クラスタ数も 0 である．ユーザが付箋アノテーションを 1 つ行った場合に，初めてクラスタが発生する．その後，付箋アノテーションが増えるたびに，biLink エージェントはクラスタリングを行い，同じクラスタに属する付箋が発生した場合には，双方向リンクの作成を行う．双方向リンクを作成する際には，付箋アノテーションをノードとしてグラフで表現した場合，完全グラフとなるように双方向リンクで結ぶ．ただし，HTML でリンクを実現するためのアンカータグの `href` 属性では，リンク先を一つしか指定できない．このため，本システムでは，3.2.1 節で示した図 3.2 の (4) の付箋の詳細表示モードにおいて，全てのリンクを羅列することにより解決を図っている．

3.6 評価実験・考察

本研究では対象コンテンツからずれない付箋アノテーションの実現のために，形態素解析と Web ページの DOM 構造を用いた付箋アノテーションの作成手法を採用した．ユーザから獲得した付箋アノテーションの応用として，エージェントによる付箋アノテーションの分類を行い，付箋アノテーション間の双方向リンクモデルを提案した．本研究で実装したシステムを用いて 2 つの実験を行い，付箋アノテーション作成にかかる時間と，双方向リンクの妥当性を評価した．本実験結果によって，ブラウジング環境が変化してもずれない付箋アノテーションが高速に作成可能であり実用可能であることを示すとともに，エージェントによって作成される双方向リンクが人間の観点と似通っていることを示す．

3.6.1 付箋アノテーション作成に要する時間

本システムにおける付箋アノテーション作成の実行速度を評価し，本システムが妥当な時間で付箋アノテーションを生成できることを示す．

表 3.1: 形態素解析，および付箋表示にかかった時間

$\boldsymbol{length}$(byte)	$\boldsymbol{t_1}$(msec)	$\boldsymbol{t_2}$(msec)
200	97	121
400	99	124
600	101	130
800	104	133
1000	111	138
1200	114	145
1400	114	147
1600	122	159
1800	123	157
2000	129	167

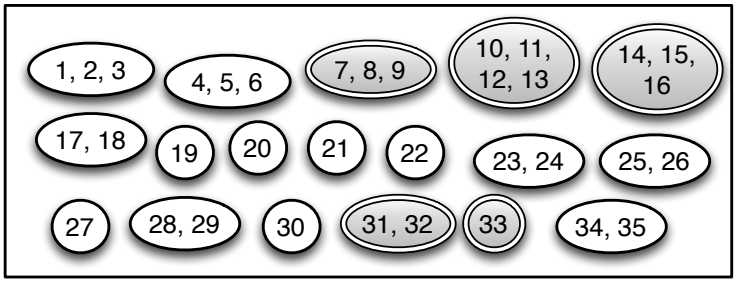
本実験では与えられたテキストの文字数の増加に伴う処理時間の変化を調査する．ここでは 200 文字から 2000 文字までの間を，200 文字刻みで評価する．各文字数あたり 10 回の付箋アノテーション作成を行い，マウスをクリックした瞬間から，付箋サーバの解析結果を Web ブラウザが受信完了するまでの時間，および，付箋アノテーションが Web ブラウザ上に表示されるまでの時間をそれぞれ計測した．

実験環境として，ネットワークの遅延の影響を避けるために，ベースエージェントとページエージェント，すなわち付箋サーバと Web ブラウザを同一計算機上で実行した．実験に用いた計算機は，CPU として Core Duo 1.83GHz，メモリ DDR2 SDRAM 1.5GB を搭載する MacBook Pro(Early 2006) であり，OS として Mac OS X 10.5.3，Web ブラウザとしては Safari 3.1.1 を用いた．

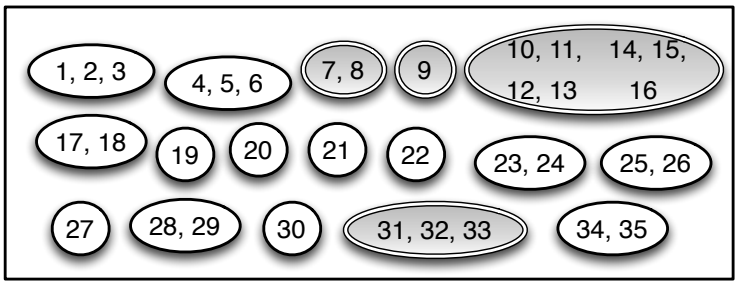
実験結果は表 3.1 のようになった．表 3.1 において $length$ とは実験に用いたテキストの長さである．t_1 はユーザがマウスをクリックしてから Web ブラウザがサーバから形態素解析の結果を受け取るまでにかかった時間である．t_2 はユーザがマウスをクリックしてから Web ブラウザ上に実際に付箋が表示されるまでにかかった時間である．

表 3.1 より，文字数の増加に伴い，処理時間も緩やかに増加していることがわかる．2000 バイトでも約 170 ミリ秒で処理が完了していることから，十分に高速であるといえる．本システムが対象とするコンテンツは Web ページの一部であるため，2000 バイトのテキストは十分な長さのテキストである．

先ほど述べたように，今回の実験では，ネットワーク遅延の影響を避けるために付箋サーバと Web ブラウザを同一計算機上で実行している．付箋サーバと Web ブラウザを異なる計算機で実行した場合には，Web ブラウザから付箋サーバへオリジナルのテキストが送信される時間，付箋サーバから形態素解析した結果のテキストが送信される時間が余分に必要となってくる．しかしこれらのテキストデータを送受信したとしても，HTTP



(a) エージェントによる分類結果 ($\alpha = 25, \theta = 0.15$)



(b) 人手による分類結果

図 3.11: 分類結果

のリクエストヘッダ，レスポンスヘッダ，TCP/IP の IP ヘッダなどを含めても高々10KB ほどのデータであると予想される．FTTH 100Mbps の理論値で計算した場合，10KB のデータは 0.8 ミリ秒，ISDN 64Kbps の理論値で計算した場合には 800 ミリ秒で転送が完了する計算となる．したがって転送を含めた処理全体としても 1 秒以下で完了することとなり，十分に実用可能である．

本実験により，ブラウジング環境が変化してもずれない付箋アノテーションが高速に作成可能であり，実用可能であることを示した．

3.6.2 双方向リンクの妥当性

双方向リンク作成のための付箋アノテーション分類手法について評価した．関連するコンテンツに対して作成された付箋アノテーション間で適切なリンクが生成されることを確認するために，適切な付箋アノテーションの分類が行われるか評価した．ブログ記事やニュース記事などのそれぞれ異なる Web ページに対してランダムに付箋アノテーションを 35 個作成し，それら 35 個の付箋アノテーションを実験対象とした．実験対象の付箋アノテーションを，エージェントと人間がそれぞれ分類した結果を調べた．付箋アノテーションの対象となった Web ページは，アノテーションに関する研究者のブログや携帯電話に関するニュースなど，17 の話題に渡っている．なお，今回の実験に使用した Web ページは全て日本語で記述されている．

図 3.11 は本実験によって得られた分類結果である．図 3.11(a) はエージェントによる分類結果である．図 3.11(b) は人手による分類結果である．図 3.11 における数字は付箋アノテーションの ID を表しており，同じ円の中に存在する番号の付箋アノテーションが同じクラスタに分類されたことを表している．例えば，付箋アノテーション $\{1,2,3,4,5,6\}$ に関しては，エージェントと被験者となったユーザともに，$\{1,2,3\}$，$\{4,5,6\}$ の 2 つのクラスタに分類した．同様に，付箋アノテーション $\{7,8,9\}$ に関しては，エージェントはこれらの付箋アノテーションを 1 つのクラスタに分類したが，被験者となったユーザはこれらの付箋アノテーションを $\{7,8\}$，$\{9\}$ の 2 つのクラスタに分類したことを示している．

図 3.11 より，期待していた分類と類似した結果が得られていることが分かる．エージェントによって 35 枚の付箋アノテーションが計 18 個のクラスタに分類されており，そのうちの 13 個が人間によって期待される分類結果と完全に合致している．つまり正解率は 72.2%である．完全に合致しなかった残りの 5 個についても，結果が類似している．

実験結果により，エージェントによる付箋アノテーションの分類結果が人手による分類結果と似ていることが確認できた．本システムではこの分類結果をもとに，付箋アノテーション間に対して双方向リンクを生成する．したがって，エージェントが作成した双方向リンクを辿ることで，ユーザは似たようなコンテンツに対して作成された付箋アノテーションを閲覧することが可能であると言える．

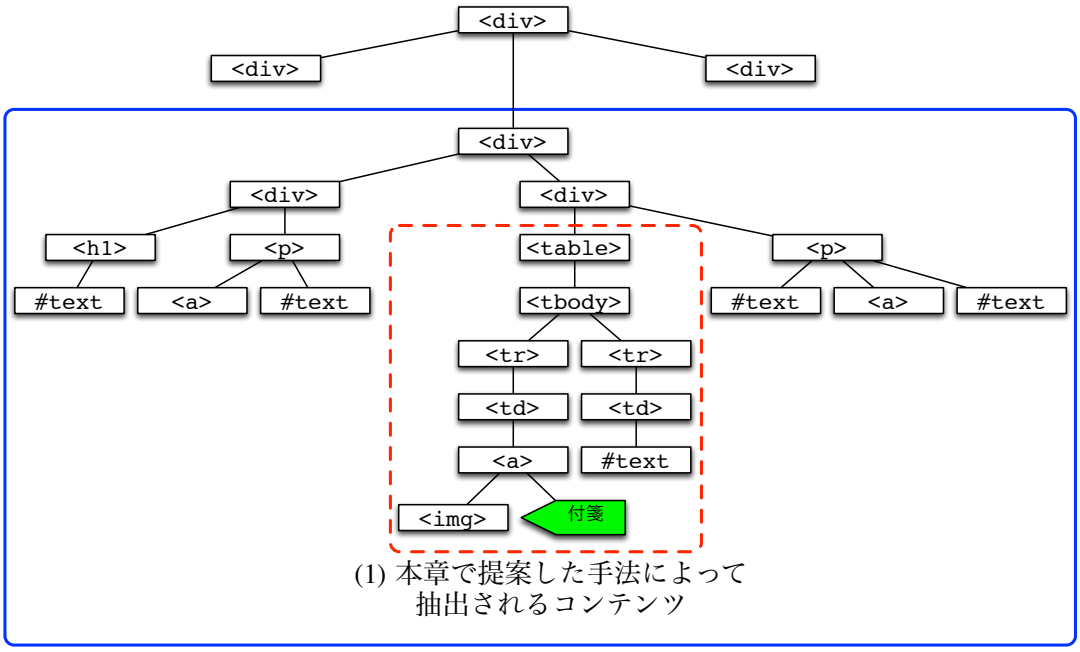
3.7　結言

本章では Web エージェントを用いた Web コンテンツへの付箋アノテーションシステムを提案し，その試作を示した．システムの概要，構成を示し，スクリーンショットを示しながら付箋アノテーションについての説明を行った．

閲覧環境に依存しない付箋アノテーションの位置決めが課題であり，本研究では本課題への解決方法として，付箋アノテーションの画像を DOM ノードに対して付加する手法を示した．これによりユーザの Web 閲覧環境が変化しても，付箋アノテーションはコンテンツを正確に差し示すことが可能になった．実験により，本手法は実用的な速度で動作することを示した．本システムでは付箋アノテーションの対象となったコンテンツ間に存在する関連性を示すために，付箋アノテーションに対して双方向リンクを作成するエージェントを実現した．関連が高いと思われるコンテンツに対して作成された付箋アノテーション間にエージェントが自動的に双方向リンクを作成することで，ユーザは関連するコンテンツ間を相互にトラバース可能となる．本双方向リンクには，検索エンジンとは異なる観点からの Web ページの発見に貢献できる．多数の Web コンテンツ間の双方向リンクはソーシャルタグ以上の情報を持つと考えられ，今後はこれらの情報を用いた新たな情報検索や推薦についての研究が必要である．

本章で実装したエージェントは，DOM 構造に基づきアノテーション対象の Web コンテンツを抽出し，付箋アノテーションのクラスタリングを行うためのテキストデータを

ニュース記事中の画像に対して付箋アノテーションを作成



(1) 本章で提案した手法によって抽出されるコンテンツ

(2) ニュース記事部分
(閲覧者にとって意味的にまとまりのある単位)

図 3.12: 適切な単位でコンテンツを抽出できない例

取得する．DOM ツリーの構造によっては適切な単位で Web コンテンツを抽出することができない．図 3.12 に例を示す．図 3.12 上部のスクリーンショットは，ニュース記事の画像部分に付箋アノテーションを行った例である．図 3.12 下部はその DOM ツリーである．3.5 節で述べたように，文書ベクトル作成の際に重みを与えるコンテンツは，付箋アノテーションが付与された DOM 要素から親ノードへたどっていった，直近のブロックレベル要素である．図 3.12 では，赤い波線で括った箇所に相当する．しかしニュース記事部分は青い実線で括った箇所であり，青い実線部分が閲覧者にとって意味的にまとまりのある単位である．エージェントの付箋アノテーションの分類精度向上のために，青い実線部分のコンテンツに対して重みを与えた上で文書ベクトルを作成することが好ましい．

Web ページ中からのメインコンテンツ抽出に関する研究が盛んに行われている [16,32,39,87,101]．付箋アノテーションの分類精度向上のために，これら既存研究で提案されている手法を導入するというアプローチも考えられる．しかし，Web ページの中でユーザにとって有用であるコンテンツはユーザごとに異なり，文献 [16,32,39,87,101] で想定しているようなメインコンテンツが付箋アノテーションの対象となるとは限らない．文献 [16,32,39,87,101] では，ニュース記事ページやブログ記事ページなどから，ニュース記事本文，ブログ記事本文を抽出するための研究が大半である．ニュース記事ページやブログ記事ページに代表されるような Web ページでは，Web ページ中のメインコンテンツ（記事部分）とサブコンテンツ（ヘッダやフッタ，広告など）が明確である．しかし，複数のメインコンテンツが配置されている Web ページも多数存在する．例えば Yahoo! のようなポータルサイトのトップページでは，主要なニュース記事一覧やサービス一覧など，ナビゲーション用コンテンツが複数存在する．文献 [16,32,39,87,101] で提案されている手法は，このような Web ページに対しては有効ではない．

本研究では計算機が Web ページを意味的にまとまりのある単位へと分割するための Web ページ分割技術を確立し，付箋アノテーションの対象となったコンテンツを特定可能とすることを試みた．次章では本研究で新たに確立した，Web ページ分割手法について言及する．

第4章　Webページ分割

4.1　序言

本章で述べる研究は，Web ページ分割に関する研究である．Web ページ中に存在する，閲覧者にとって意味的にまとまりのある単位のことを，本研究では Web ブロックと呼ぶ．Web ページ分割とは，計算機によって Web ページを Web ブロック単位へと分割することである．

Web ページ分割アルゴリズムを確立することにより，様々な Web 技術の精度向上が期待できる．図 4.1 は Yahoo!ニュースのスクリーンショットである．この Web ページの中にはメインコンテンツ（実線）である記事の他に，サイトロゴや広告，サイトメニュー，関連記事などのサブコンテンツ（破線）が含まれている．1 つの Web ページには多種多用な情報が記載されており，その中で閲覧者が必要としている情報はわずかである．Web ページ検索システム，コンテンツフィルタリングシステム，情報抽出システム等でこのような Web ページを処理対象とする場合，メインコンテンツ以外のテキスト情報がノイズとなり，システムの精度が低下してしまう [26]．システムの精度を向上させるためには，処理対象の Web ページを Web ブロックへと分割し，Web ページ中の主要な Web ブロックのみをシステムの処理対象とすればよい．3 章で述べた付箋アノテーションシステムと Web ページ分割手法を組み合わせることによって，Web エージェントが付箋貼り付けとなった Web コンテンツを適切な単位で抽出することが可能となり，双方向リンク作成の精度が向上することが期待できる．

4.2　予備実験：タイトルブロックの有無の調査

提案手法では，Web ページの製作者が Web ページ内に複数のタイトルブロックを配置するという仮定のもとで，Web ページ分割を試みる．Web ページ内にタイトルブロックが複数存在し Web コンテンツ間のセパレータとなっていることを確認するため，予備実験を行った．複数の Web ページを収集し，それら Web ページの中にタイトルブロックが存在するかどうかを調査した．

本研究では，直下に存在する Web コンテンツの見出しとなるような最小ブロックを，タイトルブロックと定義する．最小ブロックとは，“子ノードとしてブロックレベル要素を持たないブロックレベル要素” のことである．最小ブロックへの分割に関しては，4.3.1 節で詳しく述べる．

図 4.1: 1 つの Web ページが複数の Web コンテンツを含む例

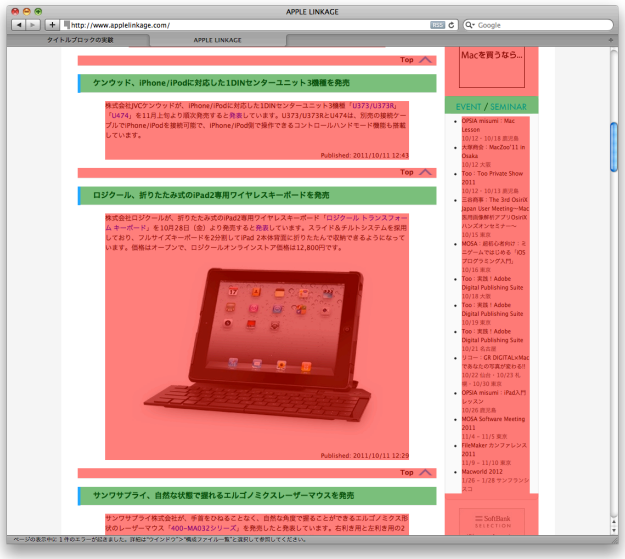


図 4.2: タイトルブロックを指定した後のスクリーンショット

4.2.1 実験方法

予備実験用に簡単なシステムを実装した．本システムは JavaScript を用いて，ブックマークレットとして実装されている．被験者がブックマークレットを実行すると，Web ページが最小ブロックへと分割され，最小ブロックが赤色に着色される．

ブックマークレットを起動後，被験者がタイトルブロックだと思ったブロックをマウスでクリックすると，ブロックが赤色から緑色へと変化する．被験者には実験対象の Web ページ内に存在する全てのタイトルブロックをクリックして緑色にするよう依頼した．図 4.2 はタイトルブロックをクリックした後のスクリーンショットである．被験者がクリッ

クした情報はサーバに送信される．サーバ側で取得した情報は，タイトルブロック有無の調査に利用するだけでなく，最小ブロックをタイトルブロックとそれ以外のブロックへと分類するための分類器を作成するための機械学習において，教師信号として利用する．分類器作成のための機械学習に関しては 4.3.2 節で詳しく述べる．

実験対象の Web ページは以下のようにして決定した．2011 年 9 月 7 日 15 時時点での Google トレンド[1]上位 10 件のキーワードをクエリーとして Google で検索を行い，それぞれのクエリーに対して検索結果の上位 5 件の Web ページを取得した．クエリーは 1 位から順に，“日向燦”，“武田邦彦”，“築地銀だこ”，“深頚部膿瘍”，“担々麺本舗 辣椒漢”，“王座戦”，“NOPOPO”，“科学技術館”，“渡辺美優紀”，“MH3G” であった．取得した 50 件の Web ページにおいて，本研究で実装したシステム（ブックマークレット）が動作しなかった Web ページが 4 件存在したため，それら 4 件の Web ページは実験対象から除外した．すなわち実験対象とした Web ページは全部で 46 件である．システムが動作しなかった原因として，Web ページの JavaScript で宣言されていた変数と本システムで宣言した変数の衝突が挙げられる．

これら 46 件の Web ページの中には，企業ページ，ブログ，Wikipedia，2 ちゃんねる，ニュースサイトのような典型的なデザインを有する Web ページだけでなく，個人が Web オーサリングツールを用いて作成した Web ページも含まれていた．実験対象の Web ページをシステムによって最小ブロックへと分割した後に，最小ブロックを被験者 7 名に提示した．被験者にはそれら最小ブロックがタイトルブロックであるか否かの分類を依頼した．分類を行ったのは名古屋工業大学の情報工学科に所属する学部 4 年の学生 7 名である．同じ最小ブロックであっても評価者によってタイトルブロックか否かの判定が分かれることがあった．その場合は多数決によってタイトルブロックか否かを決定した．

4.2.2 実験結果

実験対象とした Web ページ 46 件の中には，平均して 17 個/ページのタイトルブロックが含まれていることを確認した．タイトルブロックを全く含まないと判断された Web ページはわずか 1 件のみであった．その Web ページは，ページ全体が Flash で構成されていた．上記の実験結果により，タイトルブロックを Web コンテンツ間の区切りとして Web ページ分割するというヒューリスティクスが利用できると判断した．

4.3 タイトルブロックに着目した分割手法

コンピュータ・ビジョンでは領域分割と呼ばれる研究が存在する．領域分割とは 1 枚の画像を同じ特徴を持つ複数の領域に分けることである．領域分割を行うために，文献 [35] では分割等合法と呼ばれる手法が提案されている．分割統合等は分割フェーズと統合フェーズに分けられる．分割フェーズでは，入力された画像は小さな矩形へと分割

[1]Google トレンド, http://www.google.co.jp/trends

される．統合フェーズでは，隣接する矩形の特徴量を考慮しつつ矩形の結合が行われる．統合フェーズを終えた時点で，入力された画像は似たような特徴を持つ複数の領域へと分割される．

Web ページ分割アルゴリズムを設計するにあたっては，分割等合法を参考とした．既存の Web ページ分割手法においても，分割統合法と同様，Web ページを非常に細かい単位まで分割した後に視覚情報や DOM 構造を利用してそれらを結合し，意味的にまとまりのある単位である Web コンテンツへと分割している．本論文で提案する手法においても既存の手法と同様，一度細かい単位 (最小ブロックと呼ぶ) まで分割した後に，最小ブロックを結合する点では同じである．本手法では結合の際にタイトルブロックに着目した結合を行っており，これが既存研究との差分である．タイトルブロックとは，最小ブロックの中でも特に，直下の Web コンテンツの見出しとなる最小ブロックのことである．

タイトルブロックに着目した理由を 2 つ述べる．1 つ目の理由として，Web コンテンツが多数配置されている Web ページには，人が閲覧したときに読解しやすいよう，Web コンテンツの上部にタイトルブロックが配置されていることが多いことが挙げられる．すなわち，タイトルブロックは複数の Web コンテンツ間の仕切りとして利用することが可能であると言える．4.2 節で示したように，予備実験を行い，大半の Web ページに含まれる Web コンテンツが，タイトルブロックとそれに続く本文・画像から構成されていることを確認した．

2 つ目の理由として，最小ブロックのコンテンツ量に非依存な結合が可能になることが挙げられる．既存手法ではブロックの結合のために，2 つのブロックにおけるフォントや背景色の違い，Web ページのレンダリング結果におけるブロックの面積やブロック間の距離などを利用する．フォントが同じである場合や，ブロック間のユークリッド距離が閾値を下回った場合に，それぞれのブロックを結合する．しかし，フォントの違いやブロック間の距離が Web コンテンツの切れ目を明確に表している Web ページは少ない．長い文章の段落ごとには一定間隔の距離を空ける事も多いが，実際にはそれらの段落がまとまって 1 つの Web コンテンツを示している．また，ブロックの面積は，そのブロック内部に存在するテキストの量や画像の解像度などによって大きく変化する．そのため，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に，異なった Web ページ分割結果が作成されるという問題がある．タイトルブロックを使った結合を行うことにより，このような問題を解決することが可能となる．4.4.3 節で実験を行い，高い精度でメインコンテンツ量に依存しない Web ページ分割が可能になったことを示す．

Web ページ分割の流れは以下のようになる．まずは，Web ページ全体を，最小ブロックへと分割する．次に，分類器を用いて，最小ブロックをタイトルブロックと，それ以外のブロックへと分類する．最後に，タイトルブロックと，その直下に続くタイトルブロック以外のブロックを結合していく．全てのタイトルブロックに対して，上記の作業を繰り返す．4.3.1 節では，最小ブロックへの分割について述べる．4.3.2 節では，タイトルブロックの分類器について述べる．4.3.3 節では，タイトルブロックに基づいて最小ブ

procedure $isBlockLevelNode(n)$
$BT \leftarrow$ ['p', 'blockquote', 'pre', 'div', 'noscript', 'hr', 'address', 'fieldset', 'legend', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'ul', 'ol', 'li', 'dl', 'dt', 'dd', 'table', 'caption', 'thead', 'tbody', 'colgroup', 'col', 'tr', 'th', 'td'];

```
1: if isValidNode(n) is false then
2:     return false;
3: else if displayStyle(n) is 'block' then
4:     return true;
5: else if tagName(n) ∈ BT then
6:     return true;
7: else
8:     return false;
9: end if
```

図 4.3: ブロックレベル要素判定アルゴリズム

ロックを結合するためのアルゴリズムについて述べる．

4.3.1 最小ブロックへの分割

最小ブロックへの分割には，W3C によって定義されているブロックレベル要素 [46] を用いる．ブロックレベル要素は Web コンテンツの配置やまとまったレイアウトを指定するために使われることが多い．ブロックレベル要素は Web ページ上で矩形領域を確保し，子供の要素をその領域内に描画する．Web ページの全体的なレイアウトは，入れ子構造になったブロックレベル要素によって決定される．本論文では最小ブロックを，“子ノードとしてブロックレベル要素を持たないブロックレベル要素” と定義する．ただしインライン要素であっても，最小ブロックの兄弟ノードである場合には，そのインライン要素も 1 つの最小ブロックとして抽出する．これにより，Web ページ上にレンダリングされる全ての要素がいずれかの最小ブロックに属することとなる．DOM ノード n がブロックレベル要素であるか否かは，図 4.3 に示すアルゴリズムによって判定する．

図 4.3 に示す $isBlockLevelNode$ 関数は，単純な if-then ルールに基づいている．1 行目では関数 $isValidNode$ によって，引数のノードが “有効ノード” か否かの判定をしている．DOM ノードの中には Web ブラウザで Web ページをレンダリングした際に，実際に Web ページ上に表示されるものとされないものが存在する．表示されるノードを本論文では “有効ノード” と呼び，表示されないノードを “無効ノード” と呼ぶ．$isValidNode$ 関数内では以下の 4 つの式の判定を行い，引数として与えられた DOM ノードが有効ノー

ドか否かを判定する．

$$\begin{cases} width(n) \geq 1, height(n) \geq 1 & (4.1) \\ top(n) + height(n) > 0, left(n) + width(n) > 0 & (4.2) \\ displayStyle(n) \neq \texttt{'none'} & (4.3) \\ visibilityStyle \neq \texttt{'hidden'} & (4.4) \end{cases}$$

4 つの式が全て成り立つとき，引数の DOM ノード n は有効ノードとなる．式 (4.1) は，DOM ノードの横幅 (pixel) と縦幅 (pixel) が共に 1 以上であることである．どちらかの幅が 0 になってしまった場合，そのノードは Web ページ上に表示されない．式 (4.2) は，DOM ノードの右下の座標 (x, y) が，$x > 0$ かつ $y > 0$ を満たすことである．DOM ノードの座標は Web ページを平面とした直交座標系で表現される．その際，Web ページの左上を原点とし，x 軸は水平方向に右の方向を正，y 軸は垂直方向に下の方向を正とする．式 (4.3) は，DOM ノードの `display` スタイルが “`none`” でないことである．`display` スタイルとは要素の表示形式を指定するためのスタイルであり，“`none`” が指定された要素は Web ブラウザ上に表示されない．`display` スタイルに “`none`” が指定された要素は Web ブラウザ上に表示されない．最後に，式 (4.4) は，`visibility` スタイルが “`hidden`” でないことである．`visibility` スタイルとは要素の表示・非常時を指定するためのスタイルであり，“`hidden`” が指定された要素は Web ブラウザ上に表示されない．`visibility` スタイルに “`hidden`” が指定された要素は Web ブラウザ上に表示されない．ただし，`display` スタイルに “`none`” を指定したときとは異なり，要素を表示するための領域は確保される．

次に，図 4.3 の 3 行目と 5 行目についての説明を行う．3 行目ではノード n の `display` スタイルによる判定を行っている．5 行目ではノード n のタグが，ブロックレベル要素の記述に用いられるタグかどうかを判定している．BT は，ブロックレベル要素の記述に用いられるタグの配列である．DOM ノードに CSS で付加されるスタイルのうち，`display` スタイルではインライン要素とブロックレベル要素の指定が可能である．そのため，BT に示したタグであっても，`display` スタイルを “`inline`” に指定した場合，インライン要素となる．したがって，5 行目でノードのタグ名を判定する前に，3 行目で `display` スタイルの判定を行う必要がある．

上記のルールに従い，Web ページを最小ブロックへと分割する．図 4.4 は Google で “名古屋工業大学 新谷研究室” と検索した結果の Web ページから，検索結果の上位 3 件の部分を切り取ったスクリーンショットである．この図の中には実線で囲った 12 個の最小ブロックが存在する．

4.3.2　タイトルブロックの抽出

図 4.4 の中には 3 つのタイトルブロックが存在する．“新谷研究室” というテキストを持つ最小ブロック，“新谷研への道順” というテキストを持つ最小ブロック，“名古屋工業

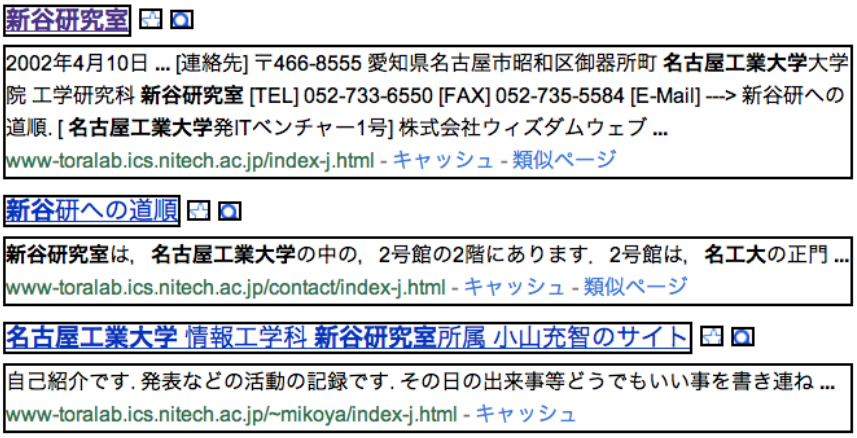


図 4.4: Web ページから抽出された最小ブロックの例

表 4.1: タイトルブロック判定パラメータ

特徴量	詳細
F_1	テキストノード長
F_2	テキストノードの面積 / ノード全体の面積
F_3	画像ノードの面積 / ノード全体の面積
F_4	ブロックの横幅 / 高さ
F_5	下隣接ブロックの面積がノードの面積より大きいかどうか
F_6	<H1>, <H2>, <H3>, <H4>, <H5>, <H6>, <DT>タグかどうか
F_7	同じ HTML タグ名が上隣接方向に連続している数
F_8	同じ HTML タグ名が下隣接方向に連続している数
F_9	下位 DOM ノードの合計数

大学 情報工学科 新谷研究室所属 小山充智のサイト” というテキストを持つ最小ブロックの 3 つである．これら 3 つの最小ブロックは直下に存在するブロックのタイトルを表しているため，タイトルブロックとみなすことができる．図 4.4 のスクリーンショットは，これらのタイトルブロックを区切りとして意味的に 3 つに分割できる．

本研究で提案する Web ページ分割アルゴリズムでは，最小ブロックをタイトルブロックとタイトルブロック以外のブロックへと分類する必要がある．本研究では機械学習によって分類器を生成した．機械学習にはレイアウトに基づく特徴量と，HTML のタグおよび DOM 構造に基づく特徴量を用いた．F_1 から F_9 の，9 つの特徴量を用いた．

それぞれの特徴量の簡単な説明と，これらの特徴量を導入した理由を述べる．

F_1 から F_5 はレイアウトに基づく特徴量である．タイトルブロックは下隣接ブロックの内容を簡潔に表すキーワード・文章で構成されるため，テキストノード長は短くなり

(F_1)，ブロック内部でテキストノードの占める面積の割合が大きくなる (F_2)．同時に，画像が占める面積の割合は小さくなる (F_3)．画像をほとんど含まず主にテキストノードで構成されるため，タイトルブロックは高さに比べて横幅が大きくなる (F_4)．タイトルブロックは下隣接ブロックの内容を簡潔に表すキーワード・文章であるため，下隣接ブロックよりも面積が小さくなる (F_5)．

F_6 から F_9 は HTML のタグおよび DOM 構造に基づく特徴量である．H1, H2, H3, H4, H5, H6 タグは見出しを記述するために定義されたタグである．DT は Definition Term の略であり，DD タグとセットで利用される．DT タグの中に定義語を記述し，DD タグの中にはその用語の説明を記述する．つまり，DT タグは DD タグに記述した内容のタイトルを表していると言える (F_6)．上下隣接方向に同じ HTML タグが連続することは，そのブロック自身が隣接するブロックと並列関係にあることを意味する．タイトルブロックは直下に存在するコンテンツの見出しとなるブロックであり，タイトルブロック自身が連続して出現することはない．したがって，同じ HTML タグが隣接して連続する可能性は低い (F_7, F_8)．タイトルブロックは背景色やフォントで装飾するだけの HTML で記述される傾向にあるため，タイトルブロックが持つ DOM ノードの下位ノード数は少なくなる (F_9)．

これらの特徴量を用いて機械学習を行い，タイトルブロックの分類器を作成する．タイトルブロックの判定は，ブロックが“タイトルブロックである” もしくは“タイトルブロックでない” の 2 クラス分類問題であり，また，枝刈りによる過学習の防止が行えるという理由から，決定木学習によって分類器を生成した．比較対象として，サポートベクターマシンによる分類器も生成した．訓練データは，予備実験において，人手で最小ブロックをタイトルブロックとそれ以外のブロックに分類したものである．訓練データの作成方法や作成した分類器の性能については，4.4.1 節の評価実験で詳しく述べる．

4.3.3　最小ブロックの結合

タイトルブロックを用いて最小ブロックをコンテンツブロックへと結合する．Web ページの閲覧者が Web ページ中で認識する意味的にまとまりのある単位のことを Web コンテンツと呼ぶが，コンテンツブロックとは，Web コンテンツを形成する最小ブロックの集合である．図 4.5 に最小ブロックの結合アルゴリズムを疑似言語で示す．本アルゴリズムに対して最小ブロックの集合を入力すると，それぞれの最小ブロックをコンテンツブロック単位へとまとめ，コンテンツブロックの集合を返す．結合の基本的なパターンは，タイトルブロックと下方向に隣接する一般ブロックを，タイトルブロックが出現するまで繰り返し結合していくというパターンである．

アルゴリズムの各ステップに対して詳細な説明を行う．入力された最小ブロックの集合からタイトルブロックを 1 つ取り出し（2, 3 行目），まずはそのタイトルブロックを一時的なコンテナへと格納する（4 行目）．以降，幅優先探索によって，結合するブロックを決定していく．下方向に隣接している最小ブロックを取得し（19 行目），取得した最

procedure $makeContentBlocks(MB)$
$MB = \{mb_1, mb_2, \cdots, mb_n\}$: Web ページ中に存在する最小ブロックの集合

```
 1: CB ⇐ {};
 2: for all mb ∈ MB do
 3:   if isTitleBlock(mb) is true then
 4:     Container = {mb};
 5:     left ⇐ ∞; top ⇐ top(mb);
 6:     x ⇐ −1; y ⇐ −1;
 7:     Q ⇐ {mb};
 8:     while Q.length > 0 do
 9:       b ⇐ shift(Q);
10:       if left > left(b) then
11:         left ⇐ left(b);
12:       end if
13:       if x ≥ left(b) + width(b) then
14:         x ⇐ left(b) + width(b);
15:       end if
16:       if y < top(b) + height(b) then
17:         y ⇐ top(b) + height(b);
18:       end if
19:       Belows ⇐ belowBlocks(b);
20:       Q ⇐ Q ∪ Belows;
21:       flag ⇐ false;
22:       for all b2 ∈ Belows do
23:         if isTitleBlock(b2) is true then
24:           flag ⇐ true;
25:           break;
26:         end if
27:       end for
28:       if flag is false then
29:         Container ⇐ Container ∪ Belows;
30:       else
31:         for all added ∈ Container do
32:           delete added from MB;
33:         end for
34:         for all mb2 ∈ MB do
35:           if inRect(left, top, x − left, y − height, mb2) is true then
36:             Container ⇐ Container ∪ mb2;
37:             delete mb2 from MB;
38:           end if
39:         end for
40:         CB ⇐ CB ∪ {Container};
41:         break;
42:       end if
43:     end while
44:   end if
45: end for
46: return CB;
```

図 4.5: 最小ブロックの結合アルゴリズム

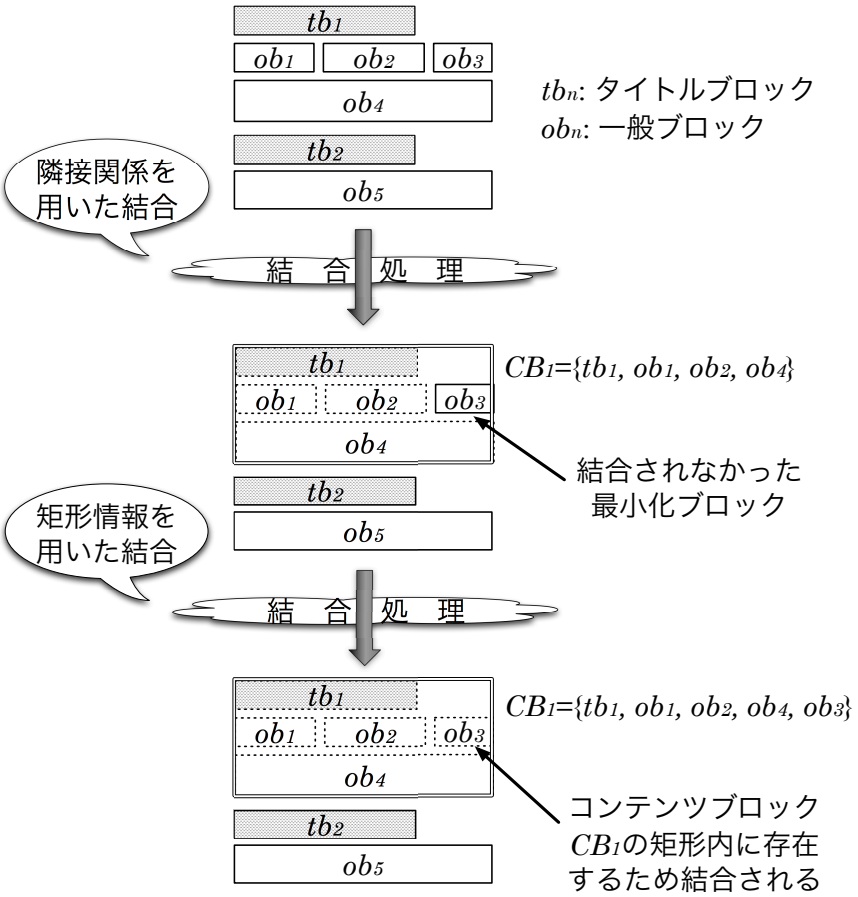


図 4.6: 結合ステップ

小ブロックの中にタイトルブロックが含まれているかどうかをチェックする（22〜27 行目）．タイトルブロックが含まれていなければ，それら最小ブロックをコンテナへ追加する（28, 29 行目）．タイトルブロックが含まれていれば，コンテナへの最小ブロック追加を終了する．その時点でコンテナに格納されている最小ブロックの集合が，コンテンツブロックということになる．コンテナに格納されている最小ブロックを最小ブロックのリストから削除する（31〜33 行目）．コンテナに格納されている最小ブロックの集合が作る矩形の中に存在する最小ブロックを，コンテナに追加する（34〜39 行目）．コンテナをコンテンツブロックの集合へ追加し，幅優先探索を終了する（40, 41 行目）．上記の処理を，全てのタイトルブロックに対して行う．

本アルゴリズムでは，隣接関係を用いた結合処理の後に，矩形情報を用いた結合処理（34〜39 行目）を行っている．具体例を図 4.6 に示す．図 4.6 の上部のように，タイトルブロックが 2 個 (tb_1, tb_2)，一般ブロックが 5 個 ($ob_1, ob_2, \cdots, ob_5$) の，合計 7 個の最小ブロックが存在する場合を考える．タイトルブロック tb_1 に着目した場合，まずは tb_1 がコンテナへと格納される．すなわち，$Container = \{tb_1\}$ である．tb_1 の下に隣接する最小

ブロックは，ob_1 と ob_2 である．これらは 2 個とも一般ブロックであるため，結合処理を進める．ob_1 と ob_2 がコンテナへと格納され，$Container = \{tb_1, ob_1, ob_2\}$ となる．ob_1 の下に隣接する最小ブロックは，ob_4 である．ob_4 は一般ブロックであるため，結合処理を進める．ob_4 がコンテナへと格納され，$Container = \{tb_1, ob_1, ob_2, ob_4\}$ となる．ob_4 の下に隣接する最小ブロックは，tb_2 である．tb_2 はタイトルブロックであるため，隣接関係を用いた結合処理を終了する．この時点で tb_1, ob_1, ob_2, ob_4 の 4 つの最小ブロックの結合を完了した（図 4.6 中央参照）．ob_3 は tb_1 と隣接していないため，上記の処理を行っただけでは ob_3 は結合されないという問題が発生する．この問題を解決するために，次のステップとして，結合された最小ブロックの集合が形成する矩形内部に位置する最小ブロックも，同一のコンテンツブロックへ結合する．図 4.6 では，コンテンツブロック CB_1 の矩形内に ob_3 が存在する．したがって ob_3 も CB_1 に結合した後に，結合処理を終了する（図 4.6 下部参照）．

図 4.6 では簡略化のためにコンテンツブロックが 1 段組の図を示したが，本アルゴリズムでは，Web ページが 1 段組であることを仮定しない．コンテンツブロックが複数の段組でレイアウトされた Web ページには，タイトルブロックも複数の段組で存在する．それぞれのタイトルブロックに対して直下に存在する一般ブロックを結合していくという処理を繰り返し行っていくため，全てのタイトルブロックに対して処理を完了した時には，Web ページが複数の段組へと分割される．

4.4 評価実験・考察

4.4.1 タイトルブロックの抽出精度

提案手法によって得られる Web ページ分割結果の精度は，タイトルブロックの判定精度によって左右される．J4.8 アルゴリズムによる決定木学習，ランダムツリーアルゴリズムによる決定木学習，サポートベクターマシンによって 3 種類の分類器を作成した．分類器作成のために使用した機械学習アルゴリズムは全て，教師あり学習である．4.2 節の予備実験で収集したデータを訓練データとして用いた．

タイトルブロック判定精度を分類器生成時の 10 分割交差検定で測定する．評価基準は，タイトルブロックを正しく判定した数 (a)，一般ブロックを正しく判定した数 (b)，タイトルブロックを一般ブロックと判定した数 (c)，一般ブロックをタイトルブロックと判定した数 (d) の 4 つで行う．タイトルブロックの判定精度 P_{tb}，一般ブロックの判定精度 P_{ob}，タイトルブロックの再現率 R_{tb}，一般ブロックの再現率 R_{ob} を以下の式で求める．

$$P_{tb} = \frac{a}{a+d},\ P_{ob} = \frac{b}{b+c}$$

$$R_{tb} = \frac{a}{a+c},\ R_{ob} = \frac{b}{b+d}$$

それぞれの F 尺度 F_{tb}，F_{ob} を以下の式で求める．

$$F_{tb} = \frac{2 \cdot P_{tb} \cdot R_{tb}}{P_{tb} + R_{tb}},\ F_{ob} = \frac{2 \cdot P_{ob} \cdot R_{ob}}{P_{ob} + R_{ob}}$$

表 4.2: タイトルブロックの判定精度と再現率

	J4.8	RT	SVM
a:タイトルブロックを正しく判定した数	588	593	424
b:一般ブロックを正しく判定した数	1401	1338	1395
c:タイトルブロックを誤判定した数	194	189	358
d:一般ブロックを誤判定した数	141	204	147
P_{tb}: タイトルブロックの判定精度	0.807	0.744	0.743
R_{tb}: タイトルブロックの再現率	0.752	0.758	0.542
F_{tb}: タイトルブロックの F 尺度	0.778	0.751	0.627
P_{ob}: 一般ブロックの判定精度	0.878	0.876	0.796
R_{ob}: 一般ブロックの再現率	0.909	0.868	0.905
F_{ob}: 一般ブロックの F 尺度	0.893	0.872	0.847

表 4.2 に，人手で判定した結果を訓練データとして分類器で学習した際の 10 分割交差検定の結果を示す．学習器に入力したブロック数はタイトルブロックが 782 個，一般ブロックが 1542 個の，合計 2324 個である．表 4.2 の “J4.8”, “RT”, “SVM” はそれぞれ，J4.8 アルゴリズムによる決定木学習，ランダムツリーアルゴリズムによる決定木学習，サポートベクターマシンを表している．

生成された決定木を観察することで，タイトルブロックの判定には表 4.1 に示した特徴量 F_6，F_5 が大きな影響を与えることが分かった．決定木の根に最も近いところでは，特徴量 F_6，つまり，`H1`，`H2`，`H3`，`H4`，`H5`，`H6`，`DT` タグで記述されているかどうかによって，分岐していた．企業や団体などの Web サイトや，個人ユースでのブログ管理には，コンテンツマネジメントシステムが用いられる．コンテンツマネジメントシステムによって作成された Web サイトではタイトルブロックとしてこれらのタグを用いて記述される場合が大半であるが，個人が作成した Web ページでは，スタイルによってフォントサイズを変更することや，HTML4 では非推奨とされている `FONT` タグを用いて直下に存在するブロックよりもフォントサイズを大きくすることによってタイトルが表現されていることがあった．そのような場合はタイトルブロックであるにも関わらず一般ブロックであると誤判定されやすい．今回，特徴量 F_5 で下に隣接するブロックとの面積サイズの大小関係を考慮したが，面積サイズだけではなく，フォントサイズの大小も比較するべきであったと言える．下に隣接するブロックのフォントサイズとの大小関係という特徴量を考慮することにより，精度・再現率が改善する可能性がある．

4.4.2　Web ページ分割結果

4.4.1 節で作成した決定木によってタイトルブロックを抽出し，それらを利用して Web ページ分割を行った．実行例を図 4.7 に 4 つ示す．赤い矩形が分割結果を表している．矩形

(a) Yahoo!ニュース

(b) Ameba ブログ

(c) amazon.co.jp

(d) 新谷研究室

図 4.7: 提案手法による Web ページの分割結果

が存在しないところは，システムによって分割されなかったところである．図 4.7 の (a) は Yahoo!ニュース[2]のニュース記事，(b) は Ameba ブログ[3]のブログ記事，(c) は Amazon.co.jp のトップページ[4]，(d) は新谷研究室のトップページ[5]である．(a), (b) は分割が成功していると判断された結果，(c) はほぼ成功していると判断された結果，(d) は失敗と判断された結果である．

[2]Yahoo!ニュース, http://headlines.yahoo.co.jp/hl
[3]Ameba (アメーバ)｜無料ブログ・コミュニティ&ゲーム SNS, http://ameblo.jp/
[4]Amazon.co.jp： 通販 - ファッション、家電から食品まで【無料配送】, http://www.amazon.co.jp/
[5]新谷研究室, http://www-toralab.ics.nitech.ac.jp/index-j.html

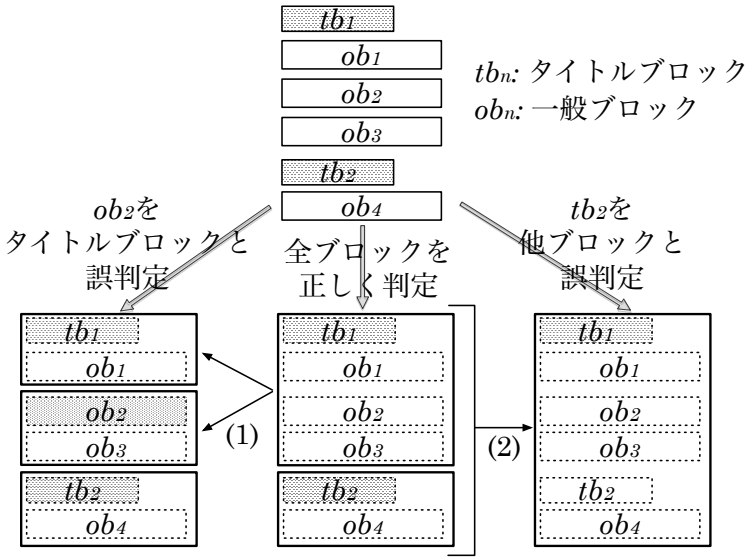


図 4.8: 誤判定によって意図しないコンテンツブロックが生成される例

決定木学習アルゴリズムには J4.8 アルゴリズムを採用した．一般ブロックをタイトルブロックと誤判定した場合，本来であれば上に隣接するブロックと結合されるべきであるが，図 4.8 の (1) に示すように，結合されずに別々のコンテンツブロックとして分割されてしまう．逆に，タイトルブロックを一般ブロックと誤判定した場合，本来であれば上に隣接するブロックとは別のコンテンツブロックとして分割されるべきであるが，図 4.8 の (2) に示すように，分割されずに一つのコンテンツブロックとして結合されてしまう．Web の閲覧者は，図 4.8(1) のように分割すべきではないところで細かいブロックに分割されると明らかに間違った分割結果であると判断するが，図 4.8(2) のように 1 つの大きなブロックとして分割された場合には間違った分割結果であるとは判断しないことが多い．そのような分割を行うためには，一般ブロックをタイトルブロックとして誤判定する数を減らすことが重要である．つまり，タイトルブロックの判定精度 P_{tb}，および一般ブロックの再現率 R_{ob} が高くなるようなアルゴリズムを採用すればよい．表 4.2 に示した実験結果により，J4.8 アルゴリズムを採用することとした．

提案手法では，Web ページ中のタイトルブロックが存在しないところではコンテンツブロックへの結合処理が行われないため，Web ページ中に分割されない領域が発生する．(d) では，タイトルブロックが 2 つしか抽出されなかったため，コンテンツブロックも 2 つしか生成されず，Web ページのほぼ大半が分割されなかった．(d) の Web ページ右上には，画像を用いて表現されているタイトルブロックが 4 つ存在するが，本研究で生成した分類器ではこれらのタイトルブロックを一般ブロックと誤判定した．これは決定木学習で利用した訓練データの中に，画像を用いて表現されているタイトルブロックがあまり含まれていなかったことが原因であると考えられる．

画像を用いたタイトルブロックの抽出精度向上のためには，訓練データを見直し，画

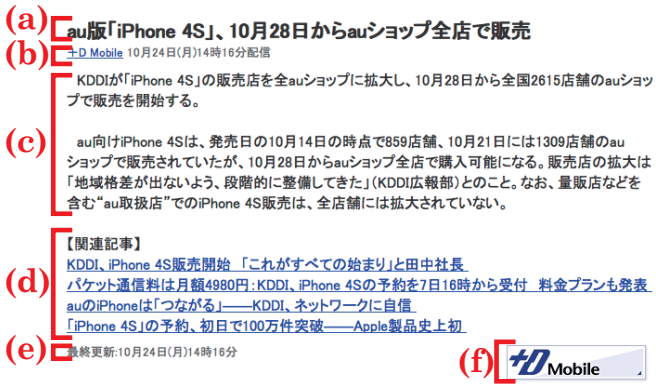


図 4.9: Yahoo!ニュースのニュース記事部分の構成

像を用いて見出しを表現しているタイトルブロックを訓練データに多数含める必要がある．表 4.1 で設定した特徴量に関しても見直しが必要である．画像を用いたタイトルブロックでは，特徴量 F_1 の値が常に 0 となってしまう．しかし，`img` 要素の `alt` 属性に設定されたテキストの長さを特徴量 F_1 として利用することで，F_1 が 0 となることを回避可能である．一般的に，Web ページ中に画像を表示するための `img` 要素の `alt` 属性には，画像の代替テキストが設定される．画像の上にマウスカーソルを乗せた時に，`alt` 属性に設定したテキストがポップアップ表示される．`alt` 属性に対しては，画像データで表現された文字のテキストデータが設定されることが多いため，上記手法が有用であると期待できる．

また，本研究で利用した決定木学習のための J4.8 アルゴリズムのメリットとして，2.1.5 節で述べたように，特徴量の欠損を扱うことが可能な点が挙げられる．`alt` 属性に対してテキストデータが設定されていなかった場合でも，特徴量 F_1 の値を 0 ではなく，欠損値として扱い決定木学習を行うことが可能である．特徴量 F_2 に関しても同様である．画像を用いたタイトルブロックではテキストノードの面積が常に 0 となるため，特徴量 F_2 の値も 0 となってしまうが，タグ名が `img` の場合には特徴量 F_2 を欠損値として扱い決定木学習を行えばよい．

訓練データを見直し新たに評価実験を行い，上記仮説を検証する必要がある．画像を用いたタイトルブロックの抽出精度が改善されることによって (d) のような Web ページの分割精度を上げることが可能であり，本研究の今後の課題である．

4.4.3 コンテンツ量に依存しない分割結果

既存研究で提案されている分割手法の問題点として，同じ Web サイト内に存在する同一レイアウトの Web ページでさえも，メインコンテンツのテキスト量が変化した場合に異なった分割結果が得られるという問題があった．提案手法によってメインコンテンツのテキスト量が変化しても同一の分割結果が得られることを確認するために，実験を

行った．

本実験では Yahoo!ニュースを実験対象の Web サイトとした．Yahoo!ニュースでは，“国内”，“海外”，“経済”，“エンターテインメント”，“スポーツ”，“テクノロジー”，“地域”の 7 ジャンルにおいて，アクセスランキングが提供されている．2011 年 10 月 24 日 2 時の時点での，それぞれのジャンルのアクセスランキング上位 20 件のニュース記事ページを対象に実験を行った．すなわち実験対象としたニュース記事ページは合計 140 件である．Yahoo!ニュースで配信されているニュース記事ページにおいて，ニュース記事部分は図 4.9 に示す (a) から (f) の 6 つから構成されている．(a) はニュース記事のタイトル，(b) は配信日時，(c) はニュース記事本文，(d) は関連記事，(e) は最終更新日時，(f) は 1 次配信元サイトのロゴである．提案手法を用いて Web ページ分割を行い，これら 6 つが 1 つのコンテンツブロックへと結合されるかどうかを確認した．ただし，(d) が存在しないニュース記事も存在するため，その場合は (d) を除く 5 つを対象とする．6 つが 1 つのコンテンツブロックへと結合された場合を分割成功とし，2 つ以上のコンテンツブロックへと結合された場合を分割失敗とする．他の最小ブロックが対象のコンテンツブロックへと結合された場合も，分割失敗とみなす．

実験を行った結果，140 件中 135 件の Web ページで分割に成功した．精度は 96.4%であり，十分に実用的であると言える．分割に失敗した 5 件では，以下のような分割が行われていた．ジャンルが経済のニュース記事ページでは，(f)1 次配信元サイトのロゴの下に，ニュース記事と関連する株価を表す最小ブロックが配置されているページがあった．これらのページでは，(a) から (f) に加え，株価を表す最小ブロックも同一のコンテンツブロックに分割された．このような失敗は 3 件存在した．

ニュース記事本文の中に画像と画像のキャプションが配置されるニュース記事ページもあるが，そのようなページの中で，キャプションがタイトルブロックと判定されたニュース記事ページが存在した．画像のキャプションをセパレータとして，2 つのコンテンツブロックへと分割された．このような失敗は 2 件存在した．4.4.1 節で述べたようにタイトルブロックの抽出精度を改善することにより分割精度を向上させることが可能であり，本研究の今後の課題とする．

4.4.4　Web ページ分割にかかる時間

実験を行い，提案手法の実行速度を計測した．実験に用いた計算機は，CPU として Core i3 3.2GHz，メモリ DDR3 SDRAM 8GB を搭載する iMac(Mid 2010) であり，OS として Mac OS X 10.6.8，Web ブラウザとしては Safari 5.1.5 を用いた．この環境上で 100 個の Web ページの分割を行い，分割にかかった時間を計測した．実験対象とした Web ページは，2012 年 4 月 23 日 11 時の時点で Yahoo!ニュースで配信されていた最新のニュース 100 件[6]である．それぞれの Web ページで 10 回分割を行い，分割にかかった平均の時間を測定した．

[6] `http://backnumber.dailynews.yahoo.co.jp/?t=l&c=top` からアクセス可能

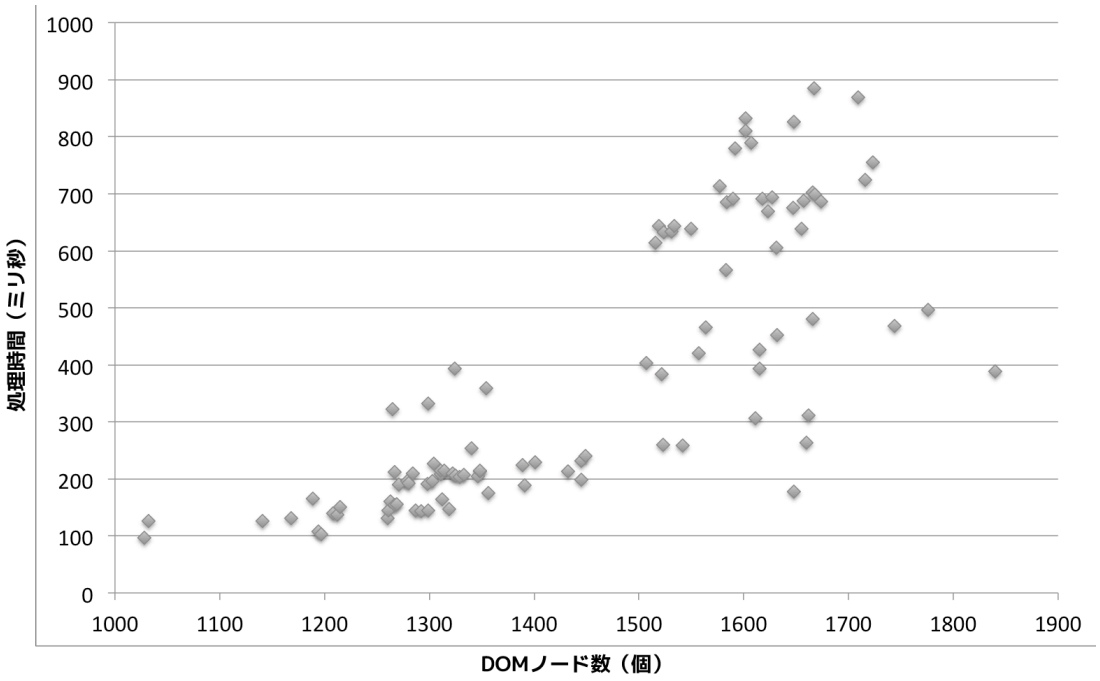


図 4.10: Web ページ分割にかかる時間

図 4.10 に，本実験の結果を示す．横軸には Web ページの DOM ノード数を，縦軸には Web ページ分割処理にかかった時間（ミリ秒）を示す．DOM ノード数だけでなく，それぞれのノードの入れ子構造の複雑さも処理時間に影響を与えると予想されるが，本研究では単純に DOM ノード数と処理時間のみを調査した．実験対象とした全ての Web ページで，1000 ミリ秒以内に分割を終えることを確認した．最も時間がかかった場合でも 885 ミリ秒で処理を終えており，その Web ページの DOM ノード数は 1667 であった．本実験により，十分実用的な時間内で提案手法による Web ページ分割が行えることを確認した．

4.5 結言

本章で提案した Web ページ分割手法では，Web ページを最小ブロックという単位まで分割した後に，Web コンテンツの見出しとなるようなブロック (タイトルブロック) に着目して最小ブロックの結合を行うことにより，Web ページを意味的にまとまりのある単位へと分割する．既存の Web ページ分割手法の多くが，面積や子ノード数など，コンテンツ量に依存する情報を用いて結合を行っていた．その結果，同一 Web サイト内の同じレイアウトの Web ページから異なる分割結果が得られるという問題が存在した．提案手法ではコンテンツ量に非依存な結合を行うために，タイトルブロックとそれに続くタイトルブロック以外のブロック（一般ブロック）を結合していく．そのためには，計算機によるタイトルブロックの抽出が課題となる．計算機によるタイトルブロックの自動抽出

を行うために，機械学習によって分類器を生成した．J4.8 アルゴリズムによる決定木学習によって生成した分類器により，F 値 77.8%，89.3%でタイトルブロックと一般ブロックの抽出に成功した．得られたタイトルブロックを用いて最小ブロックの結合を行った結果，ニュースサイトのニュース記事部分に着目した場合，96.1%の精度でコンテンツ量に依存しない同一の分割結果が得られることを確認した．複数の Web ページ上で提案手法を用いた Web ページ分割を行い，実験対象とした全ての Web ページで 1000 ミリ秒以内に処理が完了することを示した．

本システムの応用として，Web 情報の編纂が考えられる．文献 [94] では情報編纂技術に関する研究の必要性が議論されている．Google は現在，iGoogle[7] と呼ばれるサービスを提供している．iGoogle とは，Google のトップページに対してユーザが自由にコンテンツを追加できるサービスである．iGoogle では既に用意されているニュースウィジェットや天気ウィジェットなどを配置できる他，ユーザが URL を指定して自由にコンテンツを追加することも可能となっている．このような情報編纂を行うために，Web ページを Web ブロックへと分割し，各 Web ブロックに対して URL を発行する必要がある．5.2 節では Web ページ分割のアプリケーションとして，情報編纂のための Web ブロック再利用機構を実装する．

[7] iGoogle, `http://www.google.co.jp/ig`

第5章　知的活動支援への応用

5.1　序言

本章ではWebインテリジェンス技術に基づき，ユーザの知的活動支援を行うためのアプリケーションについて述べる．

まずは，Web情報を様々な形で再利用可能にするための，Webブロック再利用機構を試作する．本システムは，Webページから Webブロックを抽出してクラウド環境へと保存することにより，Web情報の再利用を支援する．抽出したコンテンツをクラウド環境上へと保存し，他の Webシステムから容易に再利用可能である点で有用である．

次に，地域社会の問題についての住民間での議論を支援するための，議論支援システム citispe@k を試作する．地域社会への住民参画支援を目的とした Webプラットフォーム O_2 [59] では，意見アーカイブとしての LODデータセット SOCIAが構築されている．citispe@k では，SOCIA に蓄積された関連情報を地域住民に対して提示する．これにより議題の背景知識の理解が深まり，住民からの意見入力の促進が期待できる．citispe@k は意見入力機能だけでなく，議論の手動構造化機能を持つ．入力された意見に質問・アイデア・ツッコミなどのタグを付与し，RDFサーバ上において住民の議論を構造化することを可能とする．

最後に，対面会議を支援するための電子会議システムをタブレット端末上に実装する．Web上に PDFの会議資料をアップロードし，タブレット端末上で会議資料を閲覧することにより，会議資料配付の手間を省くことを目的とする．PDFだけでなく，Webページや様々な動画ファイルなどといった Web情報を会議資料として利用可能にするために，本システム内部に Webの表示環境を構築する．端末上に表示する資料の表示やポインタに関して複数の端末間で画面の同期機能を実現し，会議参加者の円滑な意思疎通を支援する．

5.2　Webブロック再利用機構

Webページから抽出した Webブロックをクラウド環境へと保存することによって，Webブロックを再利用することが可能なシステムを実装した．本研究ではクラウド環境として，Evernote [1] を用いた．

[1] Evernote — Remember everything with Evernote, Skitch and our other great apps., http://evernote.com/

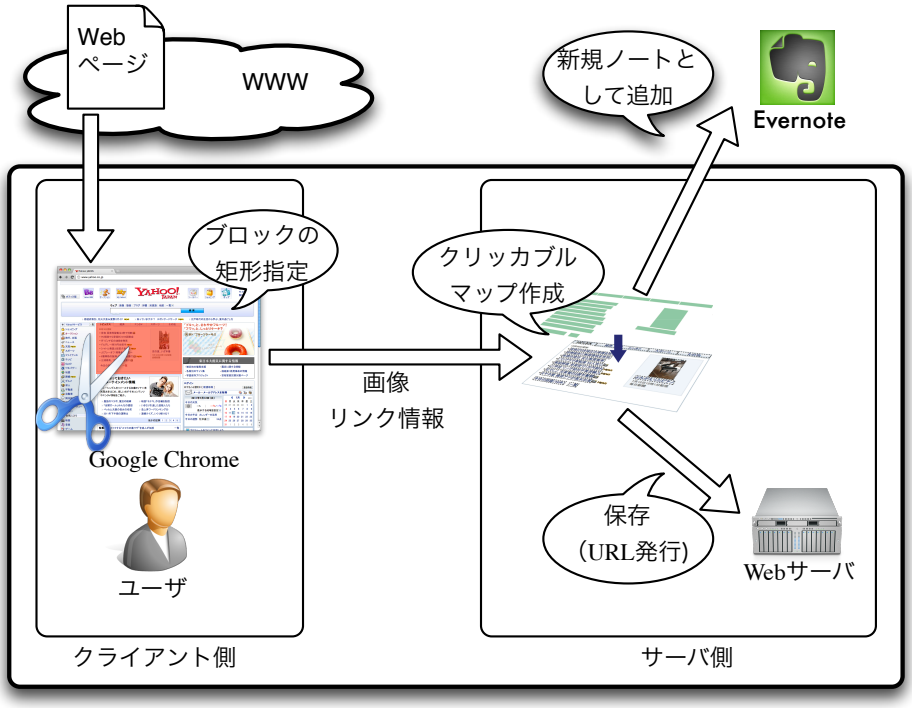


図 5.1: Web ブロック再利用システムの流れ

図.5.1 に本システム利用の流れを示す．本システムはクライアントサーバモデルに基づく．クライアント側のソフトウェアは，Web ブラウザの機能拡張として実装した．具体的には，Google Chrome の上で動作する Google Chrome Extension として実装した．ユーザはまず，Google Chrome で Web ページをロードする．次に，Extension を起動する．本 Extension を用いることで，ユーザはマウスのドラッグ&ドロップによって，Web ページ中の切り抜きたい範囲を指定することが可能となる．ここで，ユーザが切り抜くことができるのは最小ブロック以上の粒度に限定する．ユーザが座標を指定する他にも，4 章で述べた Web ページ分割手法を適用して得られる Web ブロック単位での抽出が可能となっている．

文献 [65] ではモバイル端末向けに Web ページを画像化して配信するシステムを提案している．このシステムでは画像化と同時に，Web ページの DOM 情報から Web ページ内に含まれているリンク情報（リンクが張られている要素の座標や，リンク先の URL など）を取得する．画像化した Web ページとリンク制御用のプログラムを Flash ファイルとしてラップして，モバイル端末へと配信する．文献 [65] ではサーバ側で Web ページのレンダリング，画像化，リンク抽出を行っている．本システムにおいて同様の作業をサーバ側で行うと，ユーザの Web ブラウザ上での表示と異なる結果が得られるという問題が発生する．本システムでは HTML5 技術に基づき，同様の作業をクライアント側，すなわちユーザの Web ブラウザ上で行う手法を提案する．

5.2.1 Web ページ画像化

切り抜いた部分は Web ブラウザ上で画像化され，サーバに送信される．画像化する理由は，ただ単に Web ページ中から対象の HTML コードを抽出しただけでは，元の Web ページのレイアウトを再現できない可能性があるからである．HTML4 では HTML ファイルに文書を記述し，CSS ファイルにスタイルを記述する．HTML ファイルに記述されたノードは，CSS ファイルに記述されたスタイルに従ってレンダリングされた後，Web ブラウザ上に表示される．CSS ファイルに記述されたスタイルに従って Web ページ全体をレンダリングした後に対象部分を画像として抽出することによって，上記の問題を解決することが可能となる．

画像化のために Chrome Extensions API を利用した．Chrome Extensions API には，Web ページを画像としてキャプチャするためのメソッドが用意されている．Chrome Extensions API で用意されている `chrome.tabs` オブジェクトは Chrome ブラウザのタブを表す．`chrome.tabs` オブジェクトは選択中のタブで表示している Web ページのキャプチャ画像を取得するための，`captureVisibleTab` メソッドをもつ．`captureVisibleTab` メソッドによって得られる画像は Web ページ全体の画像である．ユーザによって指定された範囲に従い，得られた画像をトリミングする必要がある．

画像のトリミングのために HTML5 の Canvas API を利用した．Canvas API とは，Web ブラウザ上で JavaScript を用いてグラフィックスを描画するための API である．`canvas` のグラフィックスコンテキストが持つ `drawImage(image,sx,sy,sw,sh,dx,dy,dw,dh)` メソッドを利用することによって，引数として与えられた画像の使用する範囲を指定することが可能である．ここで，第 1 引数 `image` は描画するイメージである．第 2，第 3 引数 `(sx, sy)` は使用範囲の開始座標である．第 4，第 5 引数 `(sw, sh)` は使用範囲の幅と高さである．第 6，第 7 引数 `(dx, dy)` は描画する画像を配置するための，`canvas` 内における開始座標である．第 8，第 9 引数 `(dw, dh)` は `canvas` 内において画像を描画する幅と高さである．

本システムの実装において，第 1 引数 `image` に対しては `captureVisibleTab` メソッドによって得られた Web ページの画像オブジェクトを指定する．第 2，第 3 引数 `(sx, sy)` はユーザによって指定された矩形の開始座標，第 3，第 4 引数 `(sx, sy)` はユーザによって指定された矩形の幅と高さを指定する．第 6，第 7 引数 `(dx, dy)` には $(0,0)$ を指定し，第 8，第 9 引数 `(dw, dh)` には再度，ユーザによって指定された矩形の幅と高さを指定する．上記のように引数を与えることによって，Web ページ全体の画像からユーザがマウスで指定した領域のみをトリミングし，`canvas` へと描画することが可能となる．次に，`canvas` の `toDataURL()` メソッドを呼び出すことによって，`canvas` 内に描画されている画像の data URL を取得する．data URL scheme とは，バイナリデータを base64 形式でエンコードすることによって，HTML や JavaScript のようなテキストデータ内にリソースを埋め込むためのものである．本システムでは `toDataURL()` メソッドによって得られた画像データを XMLHttpRequest を利用してサーバ側へポストし，サーバ側で受け取ったデータを png 形式の画像へとデコードしている．

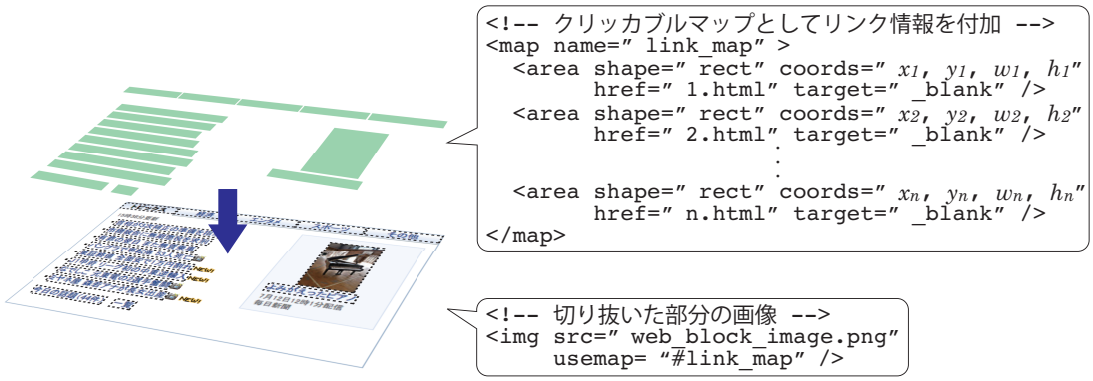


図 5.2: Web ブロックのクリッカブルマップ化

5.2.2　リンク情報の抽出

画像化によりレイアウトが崩れてしまうという問題点を解決することが可能となったが，画像化してしまうと，Web ページのリンク情報が失われてしまうという問題がある．本研究では図 5.2 のようにクリッカブルマップとしてリンク情報を画像に重ねることによって，リンクを再現した．クリッカブルマップとは，Web ページ中に配置された画像の特定の箇所に対してハイパーリンクを作成する機能である．クリッカブルマップを作成するためには，対象画像となる `img` 要素の `usemap` 属性に対して，`map` 要素の `name` 属性を指定する．`map` 要素内では，`area` 要素を用いて領域指定とリンク先のドキュメントの指定を行う．1 つの `map` 内には複数の `area` 領域を配置することが可能である．

リンク情報取得アルゴリズムを図 5.3 に示す．1 行目ではまず，Web ページ中のアンカータグを全て取得し，配列 A へと格納している．2 行目から 5 行目では，得られたアンカータグの座標を，ユーザによって指定された矩形の左上を $(0, 0)$ とした相対座標へと変換している．次に，6 行目から 10 行目では，ユーザによって指定された矩形と交わらないものを配列 A から削除している．最後に 11 行目から 18 行目で，ユーザによって指定された矩形と交わっている部分のみを残している．

5.2.3　システムのメリット

本論文ではこのクリッカブルマップの HTML をクリッカブル Web ブロックと呼んでいる．作成されたクリッカブル Web ブロックを Web サーバに保存し，クリッカブル Web ブロックに対して URL を発行する．クリッカブル Web ブロックを Evernote API を通じて，Evernote に対して新規ノートとして追加する．ユーザは以降，図 5.4 に示すように，Evernote にアクセスすることによって，切り取った Web ブロックにアクセスすることが可能となる．

ユーザが本システムを利用するメリットとして，簡易的な Web ラッパーとして利用可能な点と，容易に Web コンテンツのマッシュアップが可能な点が挙げられる．ユーザは

procedure $getLinks(d, x, y, w, h)$

d : Web ページ

x : ユーザによって指定された矩形の左上頂点 x 座標

y : ユーザによって指定された矩形の左上頂点 y 座標

w : ユーザによって指定された矩形の幅

h : ユーザによって指定された矩形の高さ

1: $A \Leftarrow$ all <A> nodes in d
2: **for all** $a \in A$ **do**
3: $left(a) = left(a) - x$;
4: $top(a) = top(a) - y$;
5: **end for**
6: **for all** $a \in A$ **do**
7: **if** $left(a) > w \mid\mid top(a) > h \mid\mid left(a) < -width(a) \mid\mid top(a) < -height(a)$ **then**
8: delete a from A;
9: **end if**
10: **end for**
11: **for all** $a \in A$ **do**
12: **if** $left(a) + width(a) > w$ **then**
13: $width(a) = x + w - left(a)$;
14: **end if**
15: **if** $top(a) + height(a) > h$ **then**
16: $height(a) = t + h - top(a)$;
17: **end if**
18: **end for**
19: **return** A;

図 5.3: リンク情報取得アルゴリズム

図 5.4: Web ブロックを Evernote 上で管理

図 5.5: Web ブロックのマッシュアップ

マウス操作のみで Web ページから Web コンテンツを抽出し，情報編纂を行うことが可能となる．すなわち，本システムは簡易的な Web ラッパーとして機能することを意味する．Web ページ中から特定の Web コンテンツを抽出するためには Web ページの HTML や DOM 構造に目を通し対象となる HTML を特定する必要があり，情報リテラシーが高

くない Web ユーザにとっては困難であった．本システムではマウス操作によって抽出したい範囲をドラッグ&ドロップで指定するだけで，特定のコンテンツを抽出することを可能とした．

本システムのもう一つのメリットとして，様々な Web ページから抽出したコンテンツを自由に組み合わせて情報編纂を行い，新たなコンテンツとすることが可能である[2]．本研究では，本技術を “Web ブロックのマッシュアップ” と呼んでいる．図 5.5 では，(1) から (4) の 4 つのブロックを任意の場所に配置した新たな Web ページを作成している．これら 4 つのブロックは，異なる Web ページから収集したものである．(1)(2) は Yahoo! JAPAN [3]のトップページから切り取ったナビゲーションである．(3) は朝日新聞社のニュースサイト[4]から切り取ったニュース記事本文である．(4) は ASCII.jp[5]のトップページから切り取ったアクセスランキングである．図 5.5 に示した Web ページは，4 つの `iframe` を用意し，`iframe` の `src` 属性に対してそれぞれの Web ブロックの URL を表示するだけで実現可能である．

5.3 Web 情報に基づく議論支援システム citispe@k

住民参画 Web プラットフォーム O_2 において構造化した情報を活用するために，citispe@k と呼ばれるアプリケーションを試作した．citispe@k は，Web 上のニュース記事や Twitter を参考にして地域の社会問題について話し合ったり整理したりするための議論支援システムである．citispe@k は Web アプリケーションとして実装されており，PC，タブレット端末を問わず，Web ブラウザから利用可能である．

citispe@k は大きく分けて，関連情報の提示，議論の構造化，の 2 つの役割を持つ．それぞれ 5.3.1 節，5.3.2 節で詳細を述べる．

5.3.1 関連情報の提示

Web ブラウザを用いて citispe@k の URL にアクセスすると，図 5.6 に示すように，イベントや関連情報が提示される．左側には，最近のイベントの一覧が表示され，地域のフィルタを選択することで，特定の地域に関するイベントのみの提示が可能となる．そして，特定のイベントを選択すると，右側にイベントの情報が提示される．それぞれは，イベント自体の情報，イベントとの関連ニュース記事やツイート，他の関連イベントへのリンクである．

リンクをクリックすると，関連イベントの場合は，そのイベントが選択された状態状態となり，図 5.6 と同様の状態となる．ニュース記事やツイッターの場合は，その URL の Web ページを表示する．図 5.7 にニュース記事を選択した状態の例を示す．

[2]私的利用を対象としており，著作権の問題はない．
[3]Yahoo! JAPAN, `http://www.yahoo.co.jp/`
[4]朝日新聞デジタル, `http://www.asahi.com/`
[5]ASCII.jp, `http://ascii.jp/`

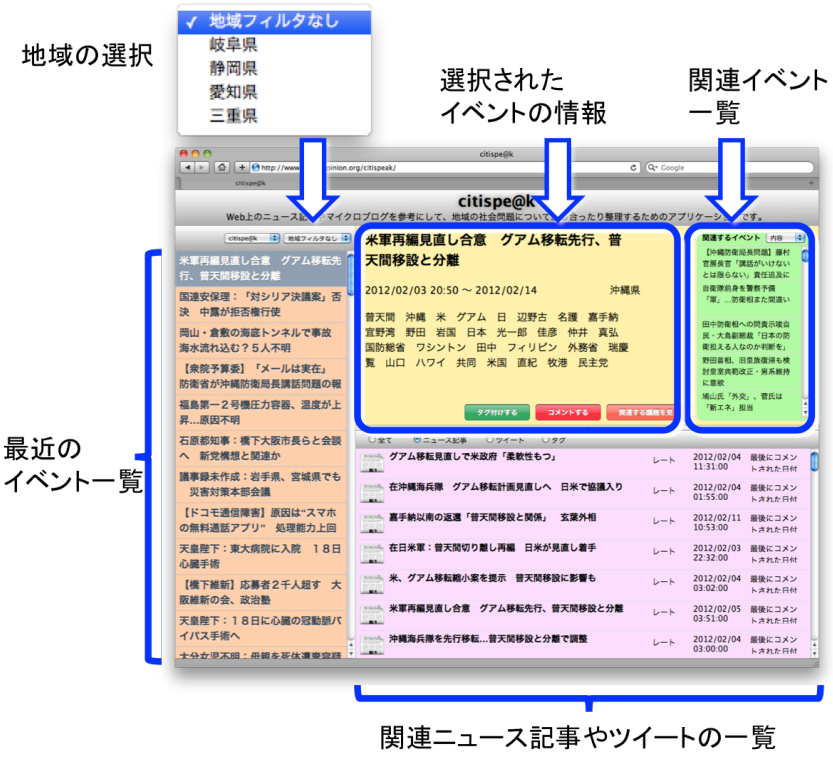


図 5.6: イベント一覧と関連情報の提示例

図 5.7: ニュース記事に対するコメント入力

図 5.8: 議題の作成

ヘッダー部分にボタンが追加された状態で Web ページが表示され，ヘッダーの“コメントする”ボタンを押すことで，そのニュース記事に対する意見を入力できる．ここで入力された意見は LOD サーバに登録され，Twitter にも投稿される．現在は特定のアカウントでの投稿となっているが，今後は任意のアカウントでの投稿にも対応する予定である．

本システムを用いて議論を行う場合，ユーザはまず，議題を作成する必要がある．図 5.8 に議題作成のインターフェースを示す．

イベントを選択し，「関連する議題を見る」ボタンを押すと，ユーザが作成した議題の一覧と，「新規議題の設定」ボタンが表示される．ここで「新規議題の設定」ボタンを押すと，議題のタイトルと詳細を入力するためのビューが表示される．タイトルと詳細を入力した後に「OK」ボタンを押すと，閲覧中のイベントと関連付けられた状態で新たな議題が作成される．

5.3.2　議論の構造化

citispe@k における議論はグラフ構造として表現される．本グラフ構造では，イベント，SOCIA 上に蓄積された Web コンテンツ，citispe@k 上で作成された議題がノードに相当する．各 Web コンテンツに対してタグを付与し，Web コンテンツ同士をリンクで接続していくことによって議論の構造化を目指す．

本システムでは，イベント，SOCIA 上に蓄積された Web コンテンツ，作成した議題に対して，(1) タグ付けを行う，(2) コメントする，(3) 資料を追加，の 3 つのアクションが可能である．議題に関しては，上記 3 つのアクション以外にも，ユーザが削除することが可能となっている．

(1) のタグ付けを行うことによって，各ノードに対して属性を与えることができる．citispe@k では，各ノードが複数の属性を持つことを許可する．2.4 節で述べたモデルにおいては，各ノードの属性は 1 つに限定されている．例えば IBIS モデルでは，1 つのノードが Issue かつ Position となることを許可しない．ある Issue I_1 に対して提案された Position P_1 が新たな Issue を含む場合，IBIS モデルでは新たに I_2 を定義し，I_1 と I_2 をリンクで接続する必要がある．しかし citispe@k では，1 つのノードに対して複数のタグを与えることを可能とした．本システムで事前に用意したタグは，ノードの種類を表すタグとして，アイデア，非難，ファシリテーション，トリビア，ツッコミ，質問，の 6 つを用意した．評価基準のタグとして，文化，経済，日本経済，治安，環境，教育，を用意した．評価基準タグはネガティブ，ポジティブ，ニュートラルの極性を持つ．例えば，日本経済に関してポジティブなノードに対しては，日本経済 + といったタグが付与される．これらの評価基準タグは，SOCIA の LOD サーバ上では図 5.9 のような形で定義される．評価基準タグは socia:polar プロパティを持ち，極性が positive の場合は+1, negative の場合は-1 の値をとる．図 5.10 はタグ付けのためのユーザインターフェースである．

上記のようなタグが意見やニュース記事などに多く付与されれば，コンサーン・アセスメントを行うための補助となる．分析は専門家や行政が行う場合があるが，タグ付けには，議論中においても住民にとっての利点がある．タグによって評価基準と極性が明示されていれば，ある議論の意見を分類して閲覧することが可能になる．これにより，その時点での他ユーザの意見や議論の状況の把握が容易となり，議論支援につながる．他のユーザがタグ付けされた意見やニュース記事などを発見しやすくなるため，タグを付ける行為自体の利点もある．付与可能なタグに関しては，ユーザが自由に追加可能としている．仮に専門家がタグを設定する場合は，評価基準などの適切性が期待されるが，イベント全てに対する十分な種類のタグの設定や，住民視点での評価基準を設定することが困難である．ただし，ユーザが自由に追加する場合は，タグの種類が増加したときに，タグ付与時の選択の負荷が増加する．そのため，タグの検索補助や整理の仕組みなどを検討する必要がある．

(2) は住民参画を促進するための，各 Web コンテンツに対して意見を投稿するための機能である．本機能を用いて，地域住民が社会問題について自分の意見を述べることを想定している．ここで入力されたコメントは SOCIA に登録されると同時に，Twitter に

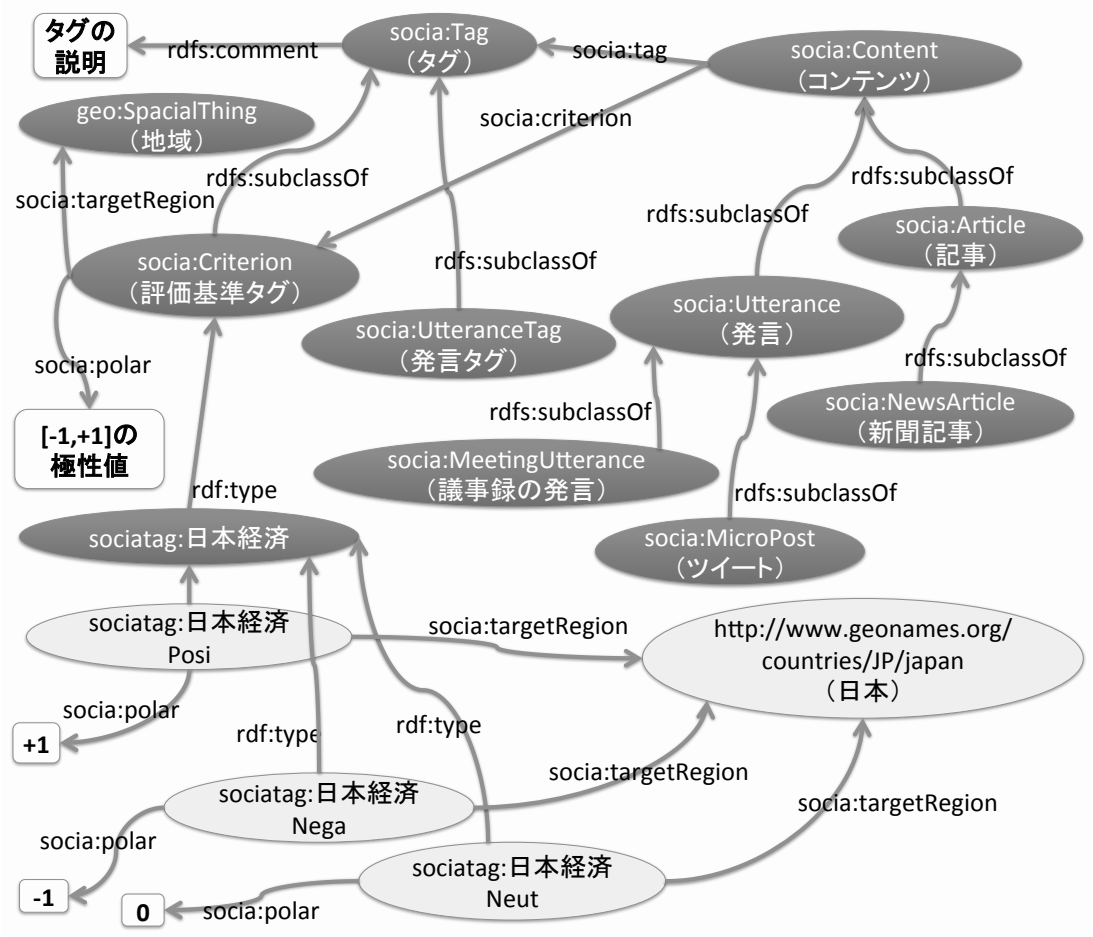


図 5.9: SOCIA における評価基準タグの定義

も投稿される．現在は特定のアカウントのみでの投稿となっているが，今後は任意のアカウントでの投稿に対応する予定である．ここで入力されたコメントに対しても (1) のタグ付けを行い構造化をすることが可能となっている．

(3) は，現在閲覧中のコンテンツに対して，他の Web コンテンツを参考資料としてリンクさせるための機能である．資料の追加ボタンを押すと，閲覧中のコンテンツと関連性が高い Web コンテンツの一覧が表示され，その一覧の中からコンテンツを選択することで，資料を追加することが可能となっている．関連性の計算手法は，文献 [93] で述べられている．システムによって提示された一覧の中に適切な資料が存在しない場合には，ユーザが Web コンテンツの URL を入力することによって，新たな資料として追加することも可能である．ここでリンクされた Web コンテンツは，5.3.1 章で述べたように，関連情報として住民に対して提示される．この機能によってグラフ構造を構築し，議論を構造化していくことが citispe@k の目的である．

5.4　タブレット端末を利用した会議支援システム

タブレット端末上で会議資料や Web 情報を参照しながら会議を行うための会議支援システムを試作した．本システムで想定している会議は対面会議である．本システムでは会議資料の配付と画面同期によって意思疎通の支援を行う．

5.4.1　会議資料の配付

会議資料をペーパーレス化し電子資料として配布することで，情報漏えいの防止，および印刷費用の削減へ貢献した．

日本ネットワークセキュリティ協会では，個人情報漏えいに関して，漏えいした組織の業種、漏えい人数、漏えい原因、漏えい経路などの調査を行っている．文献 [91] によると，2011 年に漏えいした個人情報の 68.7%が，紙媒体によって漏えいしたとされている．本調査結果は個人情報漏えいに関する結果であるが，個人情報だけでなく，企業の研究成果や開発途中の製品情報といった機密情報の大半も紙媒体によって漏えいしていると考えられる．会議資料を紙に印刷して会議参加者に対して配布することは，セキュリティ上好ましくない．会議資料を電子化し適切なサンドボックスを設計することで，情報漏えいの防止が期待できる．印刷による人件費やコピー費を削減するためにも，会議のペーパーレス化に対する需要が高まっている．

本システムで実現する会議資料配付機能の要件として，会議終了後に，会議資料に対して適切なアクセス権限を設定できる必要がある．これまでに会議資料をネットワーク経由で配布するシステムは多数開発されているが，会議終了後に会議資料を回収する仕組みに着目したシステムは存在しない．

本システムはサーバ・クライアントモデルに基づく．サーバでは，会議資料の管理を行う．図 5.11 は会議資料の管理ツールである．会議資料の管理ツールは，Web アプリケーションとして実装されている．枠内に会議資料をドラッグ&ドロップすることで，サーバへ会議資料がアップロードされる．会議資料として利用可能なファイル形式は，PDF, PowerPoint，画像，動画（H.264 + AAC の mp4 など，iOS で再生可能な形式）である．クライアントでは，会議資料を表示し，同期パケットの送受信を行う．本研究ではクライアントのタブレット端末として，iPad を採用した．iPad ではアプリケーションがサンドボックス化される．iPad 上のアプリケーションは，他のアプリケーション上で保存されたデータに対してアクセスすることができない．すなわち，本システム以外のアプリケーションから会議資料を保護することが可能となり，本研究で想定している会議資料配付機能の要件を満たす．

資料を登録すると，資料のサムネイルが表示される．図 5.11 では，4 つの資料が登録されている．サムネイルの左下には「削除」ボタン，右下には「公開」チェックボックスが表示される．削除ボタンはサーバから資料を削除するために利用される．公開チェックボックスに対してチェックを入れると，iPad への資料配布が許可される．チェックボックスからチェックを外すと，クライアント端末から資料が削除される．すなわち，会議参

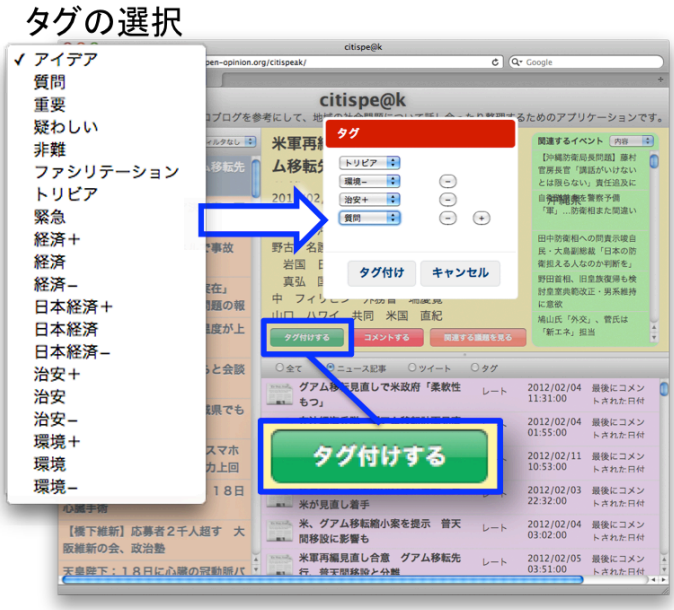


図 5.10: ニュース記事や意見，イベントに対するタグ付け

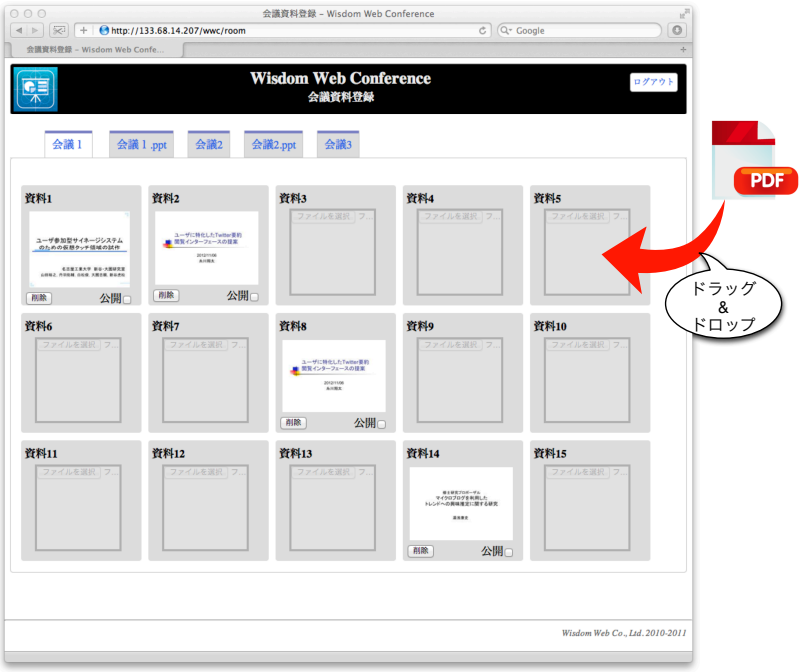


図 5.11: 会議資料の登録

加者から資料を回収したことを意味する．公開チェックボックスを利用することで，サーバから資料を削除することなしに，クライアント端末上での資料閲覧の可否を決定することが可能となる．

図 5.12 はクライアント上で会議資料を表示したスクリーンショットである．図 5.12 で

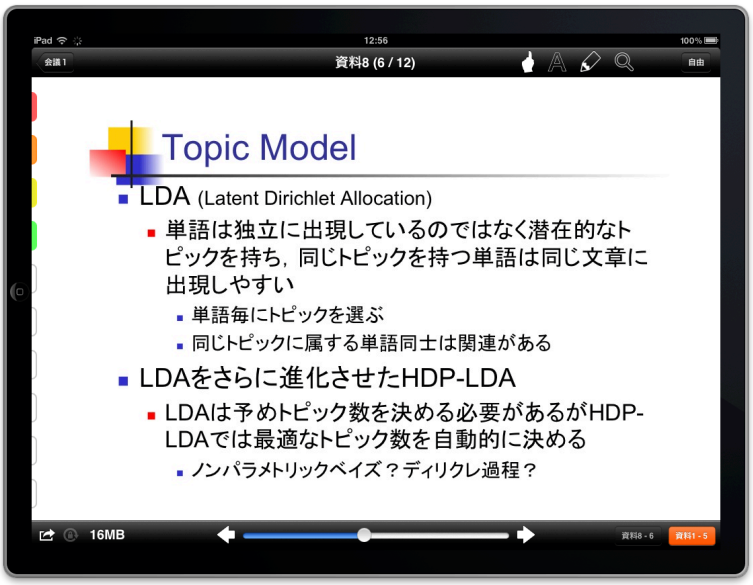


図 5.12: 会議資料の閲覧

図 5.13: citispe@k を会議資料として利用

は，PDF 資料を表示している．Web 情報を会議資料として利用可能にするために，会議資料表示画面に対して Web 情報の表示機構を実装した．Web ページの URL が記載されたテキストファイルをアップロードすることで，その Web ページを会議資料として利用することも可能である．これにより，5.2 節で抽出した Web ブロックを会議資料として利用可能となる．ここでの Web ページとは，動的な Web ページを含む．すなわち Web

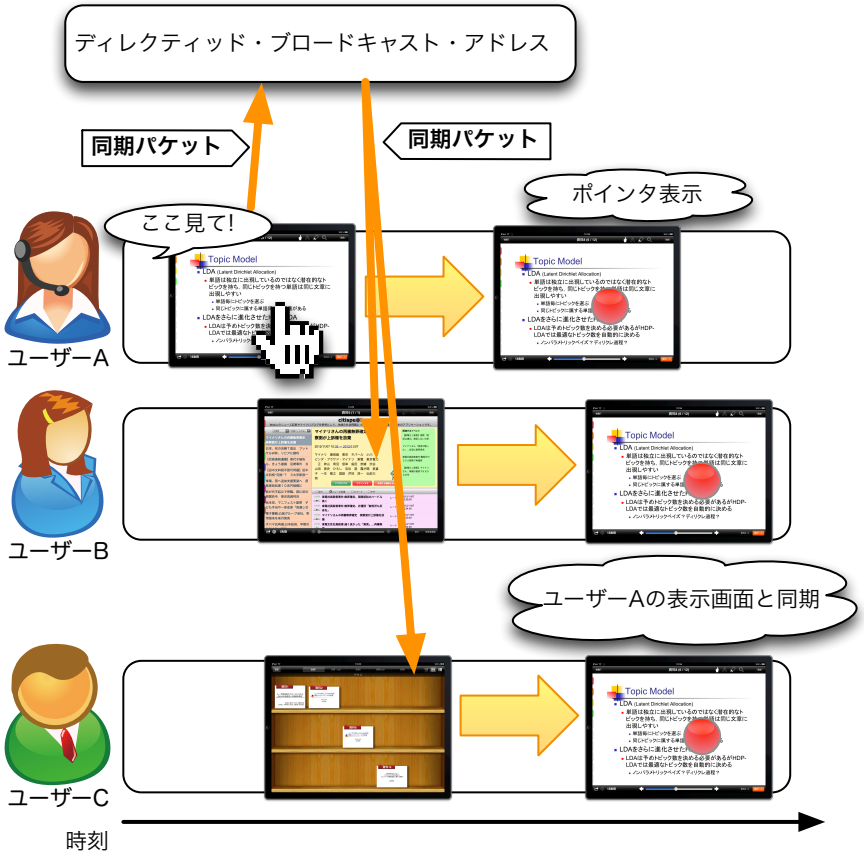


図 5.14: 画面表示の同期

アプリケーションも実行可能であり，本研究では 5.3 節で実装した citispe@k を呼び出して利用することを想定している．図 5.13 は本システムから citispe@k へアクセスしているスクリーンショットである．本システム上での動作結果が，5.3 節で示した Web ブラウザ上で実行しているスクリーンショットと一致していることが確認できる．本システムから citispe@k を呼び出すことにより，対面で発言するだけでなく，意見を入力し，議論を進めていくことが可能である．議論は LOD サーバ上で構造化されるため，会議終了後に議題を整理するために有用である．

5.4.2 画面同期による意思疎通の支援

発表者は聴講者に対して同期パケットを送信するか否かを選択することができる．同期パケットを送信することにより，発表者と聴講者のタブレット上に同じ画面が表示され，発表者・聴講者間での意思疎通が促進される．

本システムにおける画面の同期は，大きく分けて 2 つある．会議資料表示の同期と，ポインタ表示の同期である．会議資料表示の同期とは，タブレット上で表示する会議資料

を同期するための機能である．発表者が説明している会議資料の同じページが表示されることが好ましい．発表者がタブレット上で会議資料を選択すると，聴講者のタブレット上でも同じ資料が表示される．発表者がタブレット上で会議資料のページをめくると，聴講者のタブレット上でも同じようにページがめくられる．

発表者がタブレット上で会議資料の説明している箇所をタップすると，タップした位置にマーカーが表示され，聴講者のタブレット上でも同じ位置にマーカーが表示される．画面上で指を動かすことで，マーカーを指の軌跡に従ってアニメーションさせることが可能である．本マーカー機能は，プレゼンテーションでスライドを指し示すために利用されるレーザーポインターに準えて，ポインタ機能と呼ぶ．

会議資料表示の同期，およびポインタ表示の同期は，ブロードキャストを用いて実現している．ブロードキャストとは，あるネットワーク内に存在する全ての計算機に対してデータを送信することである．IP アドレスのホスト・アドレス部のビットを全て 1 にしたものは，“ディレクティッド・ブロードキャスト・アドレス” と呼ばれる．例えば 192.168.1.0/24 というネットワークにおけるディレクディッド・ブロードキャスト・アドレスは，192.168.1.255 である．本アドレスに対してデータを送信すると，同一ネットワークに接続されている全ての計算機に対して同じデータが送信される．同期パケットをディレクティッド・ブロードキャスト・アドレスに対して送信することによって会議参加者の全てのタブレットに対して同期パケットが送信され，全てのタブレット上で同一画面が表示される．

図 5.14 は，ポインタ同期機能を利用した時のイメージ図である．ユーザ A，B，C が互いに異なる資料を閲覧しているときに，発表者であるユーザ A がポインタ同期機能を利用したとする．ユーザ A の端末から同期パケットがディレクティッド・ブロードキャスト・アドレスへと送信された後，ユーザ B，C の端末へと送信される．同期パケットを受信したユーザ B，C の端末ではユーザ A と同じ会議資料が表示され，さらにユーザ A がタップしている箇所がユーザ B，C の端末上にも表示される（図 5.14 の赤い○）．

ブロードキャスト可能であるのは UDP パケットのみである．TCP をブロードキャストすることは不可能である．TCP は通信の信頼性が保証されるが，UDP ではパケット喪失の恐れがある．ポインタ同期のためのパケットが一部欠落した場合はポインタのアニメーションが滑らかにならないだけであるため，大きな問題にはならない．しかし会議資料同期のためのパケットが欠落してしまった場合には，発表者と聴講者の端末間で異なる資料が表示されてしまい，混乱の元となる．本システムでは，会議資料同期のためのパケットを複数回繰り返し送信することで，冗長性を確保する．実験を行い，同じパケットを 3 回繰り返し送信することで，20 台の iPad で問題なく動作することを確認した．

5.5 結言

Web 情報を知的活動支援に利用するために，Web インテリジェンス技術に基づいたアプリケーションを設計した．

まずは，Web 情報を様々な形で再利用可能にするための，Web ブロック再利用機構を試作した．本システムは，Web ページから Web ブロックを抽出してクラウド環境へと保存することにより，Web 情報の再利用を支援する．本システムは簡易的な Web ラッパーとして利用可能であり，容易に Web コンテンツのマッシュアップを可能とする．Web ページ中から特定の Web コンテンツを抽出するためには Web ページの HTML や DOM 構造に目を通し対象となる HTML を特定する必要があり，情報リテラシーが高くない Web ユーザにとっては困難であったが，本システムではマウス操作によって抽出したい範囲をドラッグ&ドロップで指定するだけで，特定のコンテンツを抽出可能となった．抽出したコンテンツをクラウド環境上へと保存し，他の Web システムから容易に再利用可能である点で有用である．

次に，地域社会の問題についての住民間での議論を支援するための，議論支援システム citispe@k について述べた．地域社会の問題についての住民間での議論を支援するために，SOCIA を利用した議論支援システム citispe@k を開発した．citispe@k では，SOCIA に蓄積された関連情報を地域住民に対して提示する．これにより議題の背景知識の理解が深まり，住民からの意見入力の促進が期待できる．また，citispe@k は議論の手動構造化機能を持つ．入力された意見に質問・アイデア・ツッコミなどのタグを付与し，RDF サーバ上で住民の議論を構造化することを可能にした．

最後に，対面会議を支援するための電子会議システムをタブレット端末上に実装した．Web 上に PDF の会議資料をアップロードし，タブレット端末上で会議資料を閲覧することにより，会議資料配付の手間を省くことが可能となっている．本システム内部には Web の表示環境を構築することによって，本システムから Web ブロックを利用することや，議論支援アプリケーションを実行することを可能とする．本システムによって PDF の会議資料以外にも，Web ブロックを会議資料として利用することが可能となる．UDP パケットを利用して画面の同期機能，およびポインタ表示の同期機能を実装し，会議参加者の間で円滑な意思疎通ができるようなシステムを実現した．

第6章　結論

6.1　まとめ

本研究では Web 情報を利用してユーザの知的活動を支援するために，Web インテリジェンス技術を利用した．本節では，本研究で得られた成果を，各章ごとにまとめる．

3 章では，閲覧者が任意の Web ページ内のコンテンツに対して，付箋によるアノテーションの付与を可能にする，Web アノテーションシステムを実現した．

既存の Web ページへのアノテーションシステムとして，様々なシステムが存在する．ユーザのアノテーションを活用し新たなサービスに応用しようとする点で，それらのシステムと本システムの目的は一致している．アノテーションを新たなサービスに応用するためには，ユーザから多くのアノテーションを獲得する必要がある．既存のシステムを利用する場合，ユーザは専用の拡張機能をインストールする必要や，専用の Web ブラウザを使用する必要がある．システムの利用に関して，新たな機能拡張や Web ブラウザを導入することは，ユーザにとって好ましいこととは言えない．したがって，既存のシステムでは，多くのユーザからアノテーションを獲得することが困難である．本システムは，ユーザが使い慣れた既存の Web ブラウザに対して，プロキシを指定するだけで使用可能である．付箋という見慣れた形でアノテーションを行うことが可能であるため，ユーザからの積極的なアノテーションが期待できる．このように本システムは，既存のシステムと比較して，ユーザビリティが高いと言える．

貼付けた付箋は元々指し示していた位置からずれることがないため，ユーザ自身が既存の Web ページへ付箋を貼付け，重要な箇所を示すことにより，Web 閲覧の効率を上げることができる．本システムを用いてコンテンツに対して付箋を貼付けることにより，エージェントが，それらのコンテンツを付箋を通じてリンクで結んでゆく．ユーザが重要だと思った箇所に付箋を貼付けることにより，コンテンツ同士が結びつき，ユーザが次に同じページを閲覧する際の情報収集の効率を改善することが可能である．

他のユーザが貼付けた付箋と自分が貼付けた付箋の位置を比較することで，人気のあるコンテンツの発見を容易にすると思われる．WWW のユーザは，情報収集をする際に検索エンジンを使う．通常の検索エンジンでは，サーチエンジン独自の検索アルゴリズム [62] を用いてページをランキングしている．ユーザは検索結果の上位にランクされた Web ページを重要なものとみなし，上位にランクされた Web ページは積極的に開くが，下位にランクされた Web ページを開くことは少ない．たとえ検索結果が下位にランクされる Web ページであっても，本システムでは，ユーザ自身が Web ページの重要だと思ったコンテンツに積極的にアノテーションを行うことにより，その Web ページはより一層

価値を増す．貼付けた付箋を多数のユーザ間で共有することで，検索エンジンに頼らない情報収集を行うことが可能である．

本システムでは既存の Web ページに対してアンカー要素の name 属性を埋め込むことが可能であるため，利用例として，外出前に PC で Web ページを途中まで閲覧し，読んだところまで付箋を貼付け，外出先からスマートフォンを用いてその Web ページの続きを読むといったことも考えられる．PC で貼付けた付箋へのリンクがたどれるため，PC とスマートフォン間でブックマークの共有をしているとも言える．

3 章の初期の成果は各種学術会議，および研究会において 5 本 [68–71, 73] の発表を行った．さらに議論をまとめ，その研究成果は，国際会議に 2 本 [10, 11]，和文論文誌に 1 本 [72]，論文が採録された．

4 章では，新たな Web ページ分割手法を確立した．既存の Web ページ分割手法の多くが，面積や子ノード数など，コンテンツ量に依存する情報を用いて結合を行っていた．その結果，同一 Web サイト内の同じレイアウトの Web ページから異なる分割結果が得られるという問題が存在した．提案手法ではコンテンツ量に非依存な結合を行うために，タイトルブロックとそれに続くタイトルブロック以外のブロック（一般ブロック）を結合していく．具体的には，Web ページを最小ブロックという単位まで分割した後に，Web コンテンツの見出しとなるようなブロック (タイトルブロック) に着目して最小ブロックの結合を行うことにより，Web ページを意味的にまとまりのある単位へと分割する．計算機によるタイトルブロックの抽出が課題であり，計算機によるタイトルブロックの自動抽出を行うために，機械学習によって分類器を生成した．J4.8 アルゴリズムによる決定木学習によって生成した分類器により，F 値 77.8%，89.3%でタイトルブロックと一般ブロックの抽出に成功した．得られたタイトルブロックを用いて最小ブロックの結合を行った結果，ニュースサイトのニュース記事部分に着目した場合，96.1%の精度でコンテンツ量に依存しない同一の分割結果が得られることを確認した．複数の Web ページ上で提案手法を用いた Web ページ分割を行い，実験対象とした 100 件の Web ページ全てにおいて，1000 ミリ秒以内に処理が完了することを示した．本章の初期の成果は各種学術会議，および研究会において 6 本 [74–79] の発表を行った．さらに議論をまとめ，その研究成果は，英文論文誌に 1 本 [12]，和文論文誌に 1 本 [80]，論文が採録された．

5 章では，Web インテリジェンス技術を知的活動支援へと応用するためのアプリケーションについて述べた．Web ページから Web ブロックを抽出してクラウド環境へと保存することにより，Web 情報の再利用を支援するシステムを実現した．本システムは簡易的な Web ラッパーとして利用可能であり，容易に Web コンテンツのマッシュアップを可能とする．Web ページ中から特定の Web コンテンツを抽出するためには Web ページの HTML や DOM 構造に目を通し対象となる HTML を特定する必要があり，情報リテラシーが高くない Web ユーザにとっては困難であったが，本システムではマウス操作によって抽出したい範囲をドラッグ&ドロップで指定するだけで，特定のコンテンツを抽出可能とした．抽出したコンテンツをクラウド環境へと保存し，他の Web システムから容易に再利用可能とした．その研究成果を学術会議，および研究会において 2 本 [76, 77]

発表した．

総務省の戦略的情報通信研究開発推進制度（SCOPE）における地域 ICT 振興型研究開発の一環として，2 つのシステムの研究開発を推進した．まずは，Web 上で配信されているニュース記事やツイートなどを議論の種として議論を行うための，議論支援アプリケーション citispe@k を試作した．本システム上で入力された議題や意見などは，RDF として構造化した上でサーバ（LOD サーバ）へと保存する．議論情報を構造化することによって，コンサーンアセスメントを促進する．その研究成果を学術会議において 1 本 [81] 発表した．また，“Linked Open Data チャレンジ Japan 2011[1]” において，チャレンジデー賞を受賞した．

次に，タブレット端末上で対面会議を支援するための電子会議システムを試作した．本システム内部に Web の表示環境を構築することによって，本システムから Web ブロックを利用することや，議論支援アプリケーションを実行することを可能とした．本システムによって PDF の配布資料以外にも，Web ブロックを会議資料として利用することが可能となった．LOD サーバに蓄積された豊富なニュース記事なども会議資料として利用可能とした．

6.2 貢献

本研究の貢献を，分野ごとにまとめる．本研究は Web インテリジェンスに関する研究であるが，その中でも特に，アノテーション技術・集合知，Web 情報分類，半構造データ処理，セマンティック Web，協調作業支援の分野において貢献がある．

アノテーション技術・集合知

3 章で述べる付箋アノテーションシステムは，アノテーション技術の分野に貢献がある．付箋アノテーションシステムは，Web ブラウザに対してプロキシサーバを設定するだけで，任意の Web ページ上に存在する Web コンテンツに対して付箋を用いたアノテーションを行うことが可能になる．本システムの実現には画面上での付箋の位置と対象の Web コンテンツの画面上での位置が一致していることが重要である．Web コンテンツの表示位置は Web ブラウザのレンダリング結果に依存するため，絶対座標を用いたアノテーション表示位置決定手法を利用することはできない．本研究ではユーザの Web 閲覧環境が変化しても貼付けたコンテンツからずれることのないアノテーションの表示方法を確立した．

付箋アノテーションを複数のユーザ間で共有することにより，集合知の分野に貢献できる．他のユーザが作成した付箋アノテーションと自分が作成した付箋アノテーションの位置を比較することで，人気のあるコンテンツの発見を容易にすることが期待される．

[1] Linked Open Data Challenge Japan 2011, http://lod.sfc.keio.ac.jp/challenge2011/index.html

WWW のユーザは，情報収集をする際に検索エンジンを使う．通常の検索エンジンでは，サーチエンジン独自の検索アルゴリズムを用いてページをランキングしている [62]．ユーザは検索結果の上位にランクされた Web ページを重要なものとみなし，上位にランクされた Web ページを積極的に開くが，下位にランクされた Web ページを開くことは少ない．たとえ検索結果が下位にランクされる Web ページであっても，本システムでは，ユーザ自身が Web ページの重要だと思った Web コンテンツに対して積極的にアノテーションを行うことにより，その Web ページはより一層価値を増す．作成された付箋アノテーションを多数のユーザ間で共有することで，検索エンジンに頼らない情報収集を行うことが可能になる．

Web 情報分類

3 章で述べる付箋アノテーションシステムで実装した biLink エージェントは，Web 情報分類の分野に貢献がある．biLink エージェントは，関連が高いと思われる Web コンテンツに対して貼り付けられた付箋アノテーション間に双方向リンクを作成する．本双方向リンクによって Web 情報のクラスタリングが行われ，ユーザの Web 閲覧支援へと繋がる．

本研究で提案したクラスタリング手法では，付箋アノテーション付きのコンテンツに対して重みを与え，文書ベクトルを作成するというモデルを提案した．評価実験を行い，提案モデルによって適切に分類が行えることを確認した．

付箋アノテーションシステムを運用することで構成される双方向リンク構造は，既存のハイパーリンク構造とは異なるものである．既存のハイパーリンク構造を対象とした研究がこれまでに盛んに行われている [25, 27]．文献 [25, 27] で提案されている手法を，付箋アノテーション間の双方向リンク構造へと適用させることにより，新たな知見が得られる可能性がある．すなわち，本研究成果は，Web 構造マイニングに関して新たな研究分野を切り開くことが期待できる．

半構造データ処理

4 章で提案する Web ページ分割は，半構造化文書である HTML を処理するための前処理として有用であり，半構造データ処理の分野において貢献がある．半構造化文書である HTML とその見た目を定義する CSS を処理することで，Web ページ分割を行う．HTML5 では header 要素や section 要素など，Web ページの構造を記述するためのタグが新たに定義されているが，バージョン 4 以前の仕様で記述された HTML 文書中には Web ページの構造を明確に記述するための仕組みが用意されていない．Web 上に存在する Web ページの大半が HTML4 以前の仕様で記述されている．計算機を用いて効果的な情報検索や情報推薦を行うために，前処理として Web ページ中のノイズを削除する必要がある．本研究で提案した Web ページ分割手法を行い Web ブロックに対して適切な重

み付けを行うことによって，Web ページ中から主要なコンテンツのみを抽出する技術が実現可能となる．

セマンティック Web

5 章で述べる議論支援システム citispe@k は，セマンティック Web の分野に貢献がある．citispe@k は，Web 上のニュース記事や Twitter を参考にして地域の社会問題について話し合ったり整理したりするための議論支援システムである．citispe@k は大きく分けて，関連情報の提示，議論の構造化，の 2 つの役割を持つ．住民参画 Web プラットフォーム O_2 において RDF サーバ上で構造化された情報を議論の種として，議論を支援する．citispe@k 上で入力された情報も RDF サーバ上に構造化して蓄積される．

議論データを RDF 化することのメリットとして，コンサーン・アセスメントへの応用が挙げられる．RDF タグによって，蓄積された意見データの計算機による知識処理が可能となる．RDF サーバ上で構造化されたデータを利用することにより，Web ユーザに対して関連情報や背景情報の提示を行うといったことが実現可能となり，更なる議論の発展に繋がることが期待される．

協調作業支援

5 章で述べる議論支援システム，および会議支援システムは，協調作業支援の分野に貢献がある．議論支援システムでは地域問題に関する情報提示を行い意見入力を促進することによって，議論支援を行う．会議支援システムでは会議資料の表示同期，ポインタの表示同期を行うことによって，対面会議での議論支援を行う．これらのアプリケーションによって，ユーザの協調作業が円滑に進むことが期待できる．会議支援システムは会議だけでなく教育機関での講義に利用することも可能であり，e-Learning の分野においても有用である．会議支援システムを利用して講義を行うことで，資料配付の手間を省き，同期機能によって講師が受講者に対してより効果的に説明を行うことが可能になる．

6.3 今後の課題

最後に本研究の今後の課題を示す．

付箋アノテーションシステムの利用において，他のユーザが貼付けた付箋を確認できることは他のユーザとの知識共有を可能にする．しかし全てのユーザの付箋を表示すると，ブラウザ内が他のユーザが貼付けた付箋で埋め尽くされてしまい，Web 閲覧の効率が低下してしまうという問題が考えられる．これはベースエージェントが付箋を分類し，分類結果からユーザの嗜好を学習し嗜好が似通った他のユーザの付箋のみを表示するようにすることで解決可能である．

本システムにおける付箋の利用方法はユーザによって異なり，例えば，個人のメモ書きとしてコメントを与えた場合にはその付箋は共有する必要はない．他のユーザのそういった付箋を共有すると，その付箋がノイズとなってしまい，Web 閲覧の効率を下げてしまう可能性も考えられる．付箋にパーミッションの情報を持たせることによって，共有すべき付箋と共有すべきでない付箋を明示することが可能にする必要がある．

アノテーションを共有する際に，表記の揺れの問題がある．表記方法に関しては制限がないため，多数のユーザがアノテーションを行うと，同じ意味で表記の異なるアノテーションが生じる．これは，類似したアノテーションのクラスタリングを行うことで解決可能である [90].

多数のユーザの間で付箋を共有する場合，付箋がリアルタイムに共有される方が好ましい．プッシュ型情報配信を用いて，あるユーザの付箋の貼付けを，他のユーザに配信することにより，オンライン上でリアルタイムな議論を行うことが可能となる．近年，Web でのプッシュ型情報配信技術に関する議論が盛んに行われるようになってきた．その中でも特に，WebSocket [14] や WebRTC[2]は大変興味深い技術である．WebSocket は Web ブラウザと Web サーバが双方向通信を行うことを可能とする．WebRTC は Web ブラウザ間でリアルタイムにコミュニケーションを行うことを可能とする．付箋アノテーションのリアルタイムな共有のためには，これらの技術を本システムへと応用すればよい．

動的な Web ページへの対応も今後の課題である．本論文で述べたシステムは HTML をデータベースに保存するため，静的な Web ページに付箋を貼付けることを前提としている．しかし今日では，掲示板や Weblog など，同じ URL でも閲覧する時刻によって内容が異なる Web ページが多数存在する．本システムを用いてこのような Web ページに付箋を貼付けた場合，後から再びその Web ページを表示しても，表示される内容は付箋を貼付けた時と同じ内容のものであり，更新が反映されない．これは新しい情報を得る際の妨げとなる．

Web ページ分割に関しては，タイトルブロック分類器の精度を向上させる必要がある．分類器作成のための訓練データの中に，画像を用いて表現されているタイトルブロックがあまり含まれていなかったため，画像で表現されたタイトルブロックの抽出精度に問題があった．画像を用いたタイトルブロックの抽出精度向上のためには，訓練データを見直し，画像を用いて見出しを表現しているタイトルブロックを訓練データに多数含める必要がある．本研究で決定木学習のために利用した特徴量に関しても見直しが必要である．画像を用いたタイトルブロックではテキスト長の値が常に 0 となってしまう．`alt` 属性に対しては，画像データで表現された文字のテキストデータが設定されることが多い．代わりに `img` 要素の `alt` 属性に設定されたテキストの長さを利用すればよい．

また，本研究で利用した決定木学習のための J4.8 アルゴリズムのメリットとして，特徴量の欠損を扱うことが可能な点が挙げられる．`alt` 属性に対してテキストデータが設定されていなかった場合でも，欠損値として扱い決定木学習を行うことが可能である．訓練データを見直し新たに評価実験を行い，上記仮説を検証する必要がある．

[2]WebRTC,`http://www.webrtc.org`

本研究で提案した手法では，Web ページ中のタイトルブロックが存在しない箇所の分割ができないという問題がある．同一 Web ページの中でも，タイトルブロックが存在する Web コンテンツと，存在しない Web コンテンツが配置されている．そのような Web ページにおいては，よい分割結果が得られないことを評価実験を通じて確認できた．本研究で提案した手法と既存の Web ページ分割手法が採用している手法を組み合わせることにより，分割精度が更に向上することが期待できる．具体的には，DOM ツリーにおける要素間の距離や要素の背景色やフォントサイズを利用した最小ブロックの結合処理を行えばよい．

謝辞

本研究を遂行し学位論文としてまとめるにあたり，多くの方のご支援ご協力を賜りました．この場で皆様に感謝の言葉を申し上げます．

まず，名古屋工業大学 大学院工学研究科 情報工学専攻の新谷 虎松教授に厚く御礼申し上げます．新谷教授には研究活動に限らず，様々な場面で懇切丁寧にご指導して頂き，数多くの貴重な体験をさせて頂きました．ここに心より感謝致します．

同専攻の内匠 逸教授には，本論文の副査となることを快く承諾して頂きました．専攻内審査会において頂戴いたしましたご指摘は，本論文の改善に対して大いに有益でありました．ここに心より感謝致します．

同専攻の大囿 忠親准教授には，研究や実装に関して親身にご指導して頂き，多くの有益なご意見を頂きました．大囿先生の助言により本研究が飛躍的に進展しました．ここに心より感謝致します．

同専攻の白松 俊助教には，研究活動の中で数多くの御指導を頂きました．特に論文執筆において，研究に関する有益な御指導を頂きました．ここに心より感謝致します．

名古屋工業大学 大学院工学研究科 情報工学専攻 新谷研究室の同期生である平田 紀史君とは研究室配属から 6 年間，Swezey Robin 君とは博士後期課程の 3 年間，苦楽を共にしてきました．その中で，皆の頑張り，励ましに助けられることも多く，充実した日々を過ごすことができました．ここでみなさんに感謝の意を表しつつ，今後のご活躍をお祈り申し上げます．

実験の実施に際しては，新谷研究室の後輩の皆様のご協力を頂きました．その他，新谷研究室の関係者の方々をはじめ，多数の方々に御支援を頂きました．ここで合わせて御礼申し上げます．

最後に日々の生活を支えてくれた私の家族に心より感謝します．

2013 年 1 月
佐野 博之

参考文献

[1] Allan MacLean, Richard M. Young, Victoria M.E. Bellotti and Thomas P. Moran, “Questions, Options, and Criteria: Elements of Design Space Analysis,” Journal of Human-Computer Interaction, Vol.6, pp.201–250, 1991.

[2] Apple Incorporated, “Safari Extensions Development Guide,” `https://developer.apple.com/library/safari/documentation/Tools/Conceptual/SafariExtensionGuide/SafariExtensionGuide.pdf`

[3] Claude Shannon, “A Mathematical Theory of Communication,” The Bell System Technical Journal, Vol.27, pp.379–423, 623–656, 1948.

[4] Deng Cai, Shipeng Yu, Ji-rong Wen and Wei-ying Ma, “VIPS: A Vision-based Page Segmentation Algorithm,” 2003.

[5] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma, “Extracting Content Structure for Web Pages based on Visual Representation,” The 5th Asia-Pacific Web Conference on Web Technologies and Applications, pp.406–417, 2003.

[6] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock and Gary W. Flake, “Using Web Structure for Classifying and Describing Web Pages,” The 11th International Conference on World Wide Web, pp.562–569, 2002.

[7] Gen Hattori, Keiichiro Hoashi, Kazunori Matsumoto and Fumiaki Sugaya, “Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information,” The 16th International Conference on World Wide Web, pp.361–370, 2007.

[8] Google Incorporated, “Hello There! - Google Chrome,” `http://developer.chrome.com/extensions/docs.html`

[9] Google Incorporated, “The evolution of the web,” `http://evolutionofweb.appspot.com/`, 2011.

[10] Hiroyuki Sano, Taiki Ito, Tadachika Ozono and Toramatsu Shintani, “Building Web Annotation Stickies based on Bidirectional Links,” The 5th Workshop on Semantic Web Applications and Perspectives, 2008.

[11] Hiroyuki Sano, Tadachika Ozono and Toramatsu Shintani, “Generating Bidirectional Links for Web Annotation Stickies,” The 22nd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, pp.682-690, Jun. 2009.

[12] Hiroyuki Sano, Shun Shiramatsu, Tadachika Ozono and Toramatsu Shintani, “A Web Page Segmentation Method based on Page Layouts and Title Blocks,” International Journal of Computer Science and Network Security, Vol.11, No.10, pp.84–90, 2011.

[13] Hui Guo, Jalal Mahmud, Yevgen Borodin, Amanda Stent and I.V. Ramakrishnan, “A General Approach for Partitioning Web Page Content based on Geometric and Style Information,” International Conference on Document Analysis and Recognition, Vol.02, pp.929–933, 2007.

[14] Ian Fette, “ The WebSocket Protocol,” http://tools.ietf.org/html/rfc6455, 2011.

[15] Jeff Conklin and Michael L. Begeman, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion,” The 1988 ACM Conference on Computer-Supported Cooperative Work, pp.140–152, 1988.

[16] Jeff Pasternack and Dan Roth, “Extracting Article Text from the Web With Maximum Subsequence Segmentation,” The 18th International Conference on World Wide Web, pp.971–980, 2009.

[17] Jesse James Garrett, “Ajax: A New Approach to Web Applications,” http://www.adaptivepath.com/ideas/essays/archives/000385.php, 2005.

[18] Jiming Liu, “The World Wide Wisdom Web (W4),” Databases in Networked Information Systems Lecture Notes in Computer Science, Vol.2822, pp.1–4, 2003.

[19] Jintae Lee, “Decision Representation Language (DRL) and Its Support Environment,” MIT Artificial Intelligence Laboratory Working Paper, No.325, 1989.

[20] Jintae Lee, “SIBYL: A Tool for Managing Group Design Rationale,” The 1990 ACM Conference on Computer-Supported Cooperative Work, pp.79–92, 1990.

[21] John Allsopp, “Web history, a timeline”, http://webdirections.org/history/, 2012.

[22] John Ross Quinlan, “Discovering Rules by Induction from Large Collections of Examples,” Expert Systems in the Micro-electronic Age, pp.168–201, 1979.

[23] John Ross Quinlan, “C4.5: Programs for Machine Learning,” Morgan Kaufmann Publishers Inc., ISBN:1-55860-238-0, 1993.

[24] Jos Kahan and Marja-Ritta Koivunen, “Annotea: An Open RDF Infrastructure for Shared Web Annotations,” The 10th International Conference on World Wide Web, pp.623–632, 2001.

[25] Krishna Bharat and Monika R. Henzinger, “Improved algorithms for topic distillation in a hyperlinked environment,” The 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp.104–111, 1998.

[26] Lan Yi, Bing Liu and Xiaoli Li, “Eliminating Noisy Information in Web Pages for Data Mining,” The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.296–305, 2003.

[27] Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” http://ilpubs.stanford.edu:8090/422/, 1999.

[28] Marja-Riitta Koivunen, “Annotea and Semantic Web Supported Collaboration,” The 2nd European Semantic Web Conference, 2005.

[29] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, "The WEKA Data Mining Software: An Update," ACM SIGKDD Explorations Newsletter, Vol.11, No.1, pp.10–18, 2009.

[30] Mark Klein, "The MIT Collaboratorium: Enabling Effective Large-Scale Deliberation for Complex Problems," MIT Sloan Research Paper No.4679-08, 2007.

[31] Marwan Sabbouh, Jeff Higginson, Salim Semy and Danny Gagne, "Web mashup scripting language," The 16th International Conference on World Wide Web, pp.1305–1306, 2007.

[32] Michal Marek, Pavel Pecina and Miroslav Spousta, "Web Page Cleaning with Conditional Random Fields," Workshop on Autonomic Communication, 2007.

[33] Rajendra Akerkar, Pierre Maret and Laurent Vercouter, "Web intelligence and communities,", The 4th International Workshop on Web Intelligence & Communities, ISBN 978-1-4503-1189-2, 2012.

[34] Royal Pingdom, "Internet 2011 in numbers," http://royal.pingdom.com/2012/01/17/internet-2011-in-numbers/, 2012.

[35] Shiuh-Yung Chen, Wei-Chung Lin and Chin-Tu Chen, "Split-and-Merge Image Segmentation based on Localized Feature Analysis and Statistical Tests," Graphical Models and Image Processing, Vol.53, No.5, pp.457–475, 1991.

[36] Shumeet Baluja, "Browsing on Small Screens: Recasting Web-page Segmentation into an Efficient Machine Learning Framework," The 15th International Conference on World Wide Web, pp.33–42, 2006.

[37] Simon Buckingham Shum, "Analyzing the Usability of a Design Rationale Notation," Design Rationale: Concepts, Techniques and Use, pp.185–215, 1996.

[38] Srinivas Vadrevu and Emre Velipasaoglu, "Identifying Primary Content from Web Pages and its Application to Web Search Ranking," The 20th International Conference on World Wide Web, pp.135–136, 2011.

[39] Sunita Sarawagi, "Information Extraction," Foundations and Trends in Databases, Vol.1, No.3, pp.261–377, 2008.

[40] Tadachika Ozono, Toramatsu Shintani and Yujiro Fukagaya, "On a Web Mail System based on Web Agents," International Journal of Computer Science and Network Security, Vol.6, No.5B, pp.9–17, 2006.

[41] Taiki Ito, Hiroyuki Sano, Tadachika Ozono and Toramatsu Shintani, "A Hierarchical Web Page Segmentation Algorithm Using Machine Learning," 2008.

[42] Thoughtgraph Limited, "Debategraph," http://debategraph.org/

[43] Tim Berners-Lee and Robert Cailliau, "WorldWideWeb: Proposal for a HyperText Project," http://www.w3.org/Proposal, 1990.

[44] Tim O'Reilly, "What Is Web 2.0," http://oreilly.com/web2/archive/what-is-web-20.html, 2005.

[45] W3C, "RDF - Semantic Web Standards," http://www.w3.org/RDF/

[46] W3C, “The global structure of an HTML document,” http://www.w3.org/TR/html401/struct/global.html

[47] Werner Kunz and Horst W. J. Rittel, “Issues As Elements of Information Systems,” MIT Artificial Intelligence Laboratory Working Paper, No.131, 1970.

[48] Yahoo Japan Corporation, “Yahoo!デベロッパーネットワーク,” http://developer.yahoo.co.jp/

[49] Yiming Yang, Sean Slattery and Rayid Ghani, “A study of approaches to hypertext categorization,” Journal of Intelligent Information Systems, Vol.18, pp.219–241, 2002.

[50] Yu Chen, Wei-Ying Ma and Hong-Jiang Zhang, “Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices,” The 12th international conference on World Wide Web, pp.225–233, 2003.

[51] Yujiro Fukagaya, Tadachika Ozono, Takayuki Ito and Toramatsu Shintani, “MiSpider: A Continuous Agent on Web Pages,” The 14th International Conference on World Wide Web, pp.1008–1009, 2005.

[52] Yuki Arase, Takuya Maekawa, Takahiro Hara, Toshiki Uemukai and Shojiro Nishio, “A Web Browsing System based on Adaptive Presentation of Web Contents for Cellular Phones,” The 2006 International Cross Disciplinary Workshop on Web Accessibility, pp.86–89, 2006.

[53] Ziv Bar-Yossef and Sridhar Rajagopalan, “Template Detection via Data Mining and its Applications,” The 11th International Conference on World Wide Web, pp.580–591, 2002.

[54] Zheng Chen, Liu Wenyin, Feng Zhang and Mingjing Li, “Web mining for Web image retrieval,” Journal of the American Society for Information Science and Technology, Vol.52, pp.831–839, 2001.

[55] 荒瀬由紀, 前川卓也, 原隆浩, 上向俊晃, 西尾章治郎, “携帯電話を用いた Web 閲覧のためのコンテンツ適応的提示システム (モバイルアプリケーション, < 特集 > ユビキタス時代を支えるモバイル通信と高度交通システム),” 情報処理学会論文誌, Vol.47, No.12, pp.3149–3164, 2006.

[56] 伊藤太樹, 浅見昌平, 大囿忠親, 新谷虎松, “SVM に基づくテンプレートを考慮した Web ページの分割手法について,” 電子情報通信学会技術研究報告 人工知能と知識処理, Vol.108, No.119, pp.81–86, 2008.

[57] 伊藤正詩, 大囿忠親, 新谷虎松, “ハイパーリンクの多機能化を目的とした BAC-Link システムの試作,” 合同エージェントワークショップ＆シンポジウム 2006, 2006.

[58] 江木啓訓, 石橋啓一郎, 重野寛, 村井純, 岡田謙一, “協同記録作成を基にした対面議論への参加支援環境の構築 (< 特集 > コラボレーションの「場」とコミュニティ」の編集にあたって)(会議支援),” 情報処理学会論文誌, Vol.45, No.1, pp.202–211, 2004.

[59] 大囿忠親, 白松俊, 新谷虎松: 地域コミュニティにおける議論活性化のための住民参画 Web プラットフォームについて, 第 22 回 Web インテリジェンスとインタラクション研究会, AI2011-28, pp.65–70, 2011.

[60] 加藤健太, 佐野博之, 大囿忠親, 新谷虎松, “Web ページへの付箋アノテーションを用いたニュース記事閲覧支援システム,” 第 71 回 情報処理学会全国大会, 3N-2, 2009.

[61] 小林 祐介, 櫨山 淳雄, “QOC と合理的意思決定の組合せによる設計根拠獲得手法,” 情報処理学会論文誌, Vol.49, No.7 pp.2258–2264, 2008.

[62] 兼宗進, “検索エンジンの検索アルゴリズム (< 特集 > インターネット検索エンジン),” 情報の科学と技術, Vol.54, No.2, pp.78–83, 2004.

[63] 楠村幸貴, 土方嘉徳, 西田正吾, “テンプレートの交差と DOM 構造の解析による情報抽出手法の提案 (データマイニング),” 電子情報通信学会論文誌, Vol.J90-D, No.9, pp.2495–2509, 2007.

[64] 工藤拓, “MeCab: Yet Another Part-of-Speech and Morphological Analyzer,” `http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html`

[65] 近藤圭佑, 浅見昌平, 大囿忠親, 新谷虎松, “モバイル端末のための Web コンテンツ閲覧支援環境とその応用,” 電子情報通信学会技術研究報告 人工知能と知識処理, Vol.108, No.325, pp.13–18, 2008.

[66] 齋藤哲生, 清光英成, 田中克己, “ページの動的再構成を行う Web 注釈付けシステムの提案,” 情報処理学会研究報告 データベース・システム研究会報告, Vol.2001, No.70, pp.265–272, 2001.

[67] 坂本暁, 北英彦, 高瀬治彦, 林照峯, “Proxy 技術を利用した Web サービスのためのプラットフォームの提案,” 情報処理学会研究報告, Vol.2002, No.031, 2002.

[68] 佐野博之, 大囿忠親, 新谷虎松, “DOM ツリー解析に基づく Web ページへの付箋貼付けシステム,” 第 6 回 情報科学技術フォーラム, 第二分冊 pp.357–358, 2007.

[69] 佐野博之, 浅見昌平, 大囿忠親, 新谷虎松, “Web エージェントを用いた Web コンテンツへの付箋アノテーションの実現,” 合同エージェントワークショップ&シンポジウム 2007, 2007.

[70] 佐野博之, 近藤圭佑, 浅見昌平, 大囿忠親, 新谷虎松, “オンライン Web ページに基づく付箋アノテーションシステムとその応用,” 第 70 回 情報処理学会全国大会, 5R-6, 2008.

[71] 佐野博之, 大囿忠親, 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 人工知能学会全国大会, 2G1-03, 2008.

[72] 佐野博之, 浅見昌平, 大囿忠親, 新谷虎松, “Web エージェントを用いた Web コンテンツへの付箋アノテーションの実現,” コンピュータソフトウェア, Vol.26, No.3, pp.69–77, 2009.

[73] 佐野博之, 伊藤太樹, 柿元宏晃, 平田紀史, 白松俊, 大囿忠親, 新谷虎松, “Web 閲覧者の視点を考慮した付箋アノテーション間のリンク構造に基づく情報推薦モデルの提案,” 第 72 回 情報処理学会全国大会, 3Q-7, 2010.

[74] 佐野博之, 土井達也, 白松俊, 大囿忠親, 新谷虎松, “役割に基づく Web ページの分割手法とその応用について,” 電子情報通信学会 人工知能と知識処理研究会, Vol.110, No.301, pp.61-66, 2010.

[75] 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “確率モデルを用いた Web ブロックの役割推定手法とその応用,” 第 25 回 人工知能学会全国大会, 3F-14, 2011.

[76] 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “閲覧者の観点に基づく Web ページ分割手法設計のための事例収集システムの実現,” 日本ソフトウェア科学会 第 28 回大会, 1E-1, 2011.

[77] 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “閲覧者の観点に基づく Web ページ分割のための事例収集エージェントについて,” 合同エージェントワークショップ&シンポジウム 2011, 2011.

[78] 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “Web ブロック間のリンク構造に基づく閲覧者の観点の構造化システムの試作,” 情報処理学会 知能システム研究会, Vol.2012-ICS-165, No.6, pp.1–6, 2012.

[79] 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “Web ブロック間のリンク構造に基づく閲覧者の観点の構造化について,” 平成 24 年 電気学会全国大会, 3-032, 2012.

[80] 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “Web ページ分割のための決定木学習を用いたタイトルブロック抽出,” 電子情報通信学会論文誌, Vol.J95-D, No.4, pp.909–918, 2012.

[81] 佐野博之, 平田紀史, Robin Swezey, 白松俊, 大囿忠親, 新谷虎松, “住民参画 Web プラットフォーム O2 における関連情報を用いた議論支援システム,” 第 26 回 人工知能学会全国大会, 3C2-OS-13b-12, 2012.

[82] 新谷虎松, 大囿忠親, “知的 Web のためのマッシュアッププログラミング,” 情報処理 Vol.50, No.5, pp.444–453, 2009.

[83] 総務省, “平成 23 年通信利用動向調査の結果,” http://www.soumu.go.jp/menu_news/s-news/01tsushin02_02000040.html, 2012.

[84] 高崎隼, 佐野博之, 大囿忠親, 新谷虎松, “オフライン Web 技術に基づく付箋アノテーションシステムの試作,” 第 71 回 情報処理学会全国大会, 3N-3, 2009.

[85] 竹原幹人, 小山聡, 角谷和俊, 田中克己, “自律的伝搬機能を持つ Web アノテーションシステム,” 電子情報通信学会 第 14 回 データ工学ワークショップ, 2003.

[86] 田辺正喜, 大囿忠親, 伊藤孝行, 新谷虎松, “ユーザの閲覧ページに合わせた BookMarklet によるドメインへの Web サービス付加システム,” 第 68 回 情報処理学会全国大会, 2006.

[87] 鶴田雅信, 増山繁, “レイアウト情報を用いた Web ページの主要な DOM ノードの抽出法,” 人工知能学会論文誌, Vol.25, No.6, pp.742–756, 2010.

[88] 中谷圭吾, 鈴木優, 川越恭二, “文書間類似度とキーワードを用いた Web リンク自動生成手法,” 日本データベース学会 Letters, Vol.4, No.2, pp.89–92, 2005.

[89] 中野雄介, 山登庸次, 武本充治, 須永宏, “Web アプリケーション Web サービス化ラッパシステムの実装と評価 (ネットワークサービス, < 特集 > 情報洪水時代のネットワークサービス),” 情報処理学会論文誌, Vol.49, No.2, pp.727–738, 2008.

[90] 丹羽智史, 土肥拓夫, 本位田真一, “Folksonomy マイニングに基づく Web ページ推薦システム (エージェント応用システム, < 特集 > マルチエージェントの理論と応用),” 情報処理学会論文誌, Vol.47, No.5, pp.1382–1392, 2006.

[91] 日本ネットワークセキュリティ協会, “2011 年 情報セキュリティインシデントに関する調査報告書 〜個人情報漏えい編〜,” http://www.jnsa.org/result/incident/2011.html, 2012.

[92] 服部元, 松本一則, 菅谷史昭, “タグの深さを利用したコンテンツ間距離に基づく Web ページの自動分割方式,” 日本データベース学会 Letters, Vol.4, No.1, pp.149–152, 2005.

[93] 平田紀史, Robin Swezey, 佐野博之, 白松俊, 大囿忠親, 新谷虎松, “住民参画 Web プラットフォームのためのニュース記事と意見の構造化,” 人工知能学会研究会資料, SIG-SWO-A1101-02, pp.1–6, 2011.

[94] 松下光範, 加藤恒昭, “情報編纂研究促進のための試み,” 人工知能学会論文誌, Vol.24, No.2, pp.272–283, 2009.

[95] 松岡有希, 坂本竜基, 中田豊久, 伊藤禎宣, 武田英明, “論文概要に対する色付きアンダーライン付与システムの運用・分析,” 電子情報通信学会第 17 回データ工学ワークショップ, 2006.

[96] 松尾豊, 安田雪, “SNS における関係形成原理 —mixi のデータ分析—,” 人工知能学会論文誌, Vol.22, No.5, pp.531–541, 2007.

[97] 松下光範, 櫻井茂明, 村田忠彦, 高間康史, “インテリジェント Web インタラクションにおけるファジィ処理の役割 (< 特集 >Web インテリジェンスとインタラクション),” 日本知能情報ファジィ学会誌, Vol.18, No.2, pp.119–128, 2006.

[98] 宮部真衣, 吉野孝, “All for one 型対面会議支援システムのためのワークスペースアウェアネスの効果,” 電子情報通信学会論文誌, Vol.J94-D, No.1, pp.27–36, 2011.

[99] 山田誠二, 村田剛志, 北村泰彦, “知的 Web 情報システム (< 特集 >「Web システムにおける情報獲得支援技術」),” 人工知能学会誌, Vol.16, No.4, pp.495–502, 2001.

[100] 湯田聴夫, 小野直亮, 藤原義久, “ソーシャル・ネットワーキング・サービスにおける人的ネットワークの構造 (事例分析, < 特集 > ネットワーク生態学〜生命現象から社会文化現象の新しいパースペクティブ〜),” 情報処理学会論文誌, Vol.47, No.3, pp.865–874, 2006.

[101] 吉田光男, 山本幹雄, “教師情報を必要としないニュースページ群からのコンテンツ自動抽出,” 日本データベース学会論文誌, Vol.8, No.1, pp.29–34, 2009.

[102] 吉野孝, 井出美奈, “All for one 型多言語会議支援システムの構築と評価 (ヒューマンコミュニケーション基礎, < 特集 > ヒューマンコミュニケーション〜人間中心の情報環境構築のための要素技術〜論文),” 情報処理学会論文誌, Vol.51, No.1, pp.36–44, 2010.

研究業績

第一著者

学術雑誌（査読付き）

1. 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “Web ページ分割のための決定木学習を用いたタイトルブロック抽出,” 電子情報通信学会論文誌, Vol.J95-D No.4, pp.909-918, 2012 年 4 月. **査読付き**
2. Hiroyuki Sano, Shun Shiramatsu, Tadachika Ozono and Toramatsu Shintani, “A Web Page Segmentation Method based on Page Layouts and Title Blocks,” International Journal of Computer Science and Network Security, Vol.11 No.10, pp.84-90, Oct. 2011. **査読付き**
3. 佐野 博之, 浅見 昌平, 大囿 忠親, 新谷 虎松, “Web エージェントを用いた Web コンテンツへの付箋アノテーションの実現,” コンピュータソフトウェア, Vol.26 No.3, pp.69-77, 2009 年 7 月. **査読付き**

国際会議（査読付き）

4. Hiroyuki Sano, Tadachika Ozono and Toramatsu Shintani, “Generating Bidirectional Links for Web Annotation Stickies,” The 22nd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, pp.682-690, Jun. 2009. **査読付き**
5. Hiroyuki Sano, Taiki Ito, Tadachika Ozono and Toramatsu Shintani, “Building Web Annotation Stickies based on Bidirectional Links,” The 5th Workshop on Semantic Web Applications and Perspectives, Dec. 2008. **査読付き**

研究会

6. 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “Web ブロック間のリンク構造に基づく閲覧者の観点の構造化システムの試作,” 情報処理学会 知能システム研究会, Vol.2012-ICS-165 No.6, pp.1-6, 2012 年 1 月.
7. 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “閲覧者の観点に基づく Web ページ分割のための事例収集エージェントについて,” 合同エージェントワークショップ&シンポジウム 2011, 2011 年 10 月.
8. 佐野 博之, 土井 達也, 白松 俊, 大囿 忠親, 新谷 虎松, “役割に基づく Web ページの分割手法とその応用について,” 電子情報通信学会 人工知能と知識処理研究会, Vol.110 No.301, pp.61-66, 2010 年 11 月.
9. 佐野 博之, 浅見 昌平, 大囿 忠親, 新谷 虎松, “Web エージェントを用いた Web コンテンツへの付箋アノテーションの実現,” 合同エージェントワークショップ&シンポジウム 2007, 2007 年 10 月.

国内会議

10. 佐野 博之, 平田 紀史, Robin Swezey, 白松 俊, 大囿 忠親, 新谷 虎松, ”住民参画 Web プラットフォーム O2 における関連情報を用いた議論支援システム,” 第 26 回 人工知能学会全国大会, 3C2-OS-13b-12, 2012 年 6 月.
11. 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “Web ブロック間のリンク構造に基づく閲覧者の観点の構造化について,” 平成 24 年 電気学会全国大会, 3-032, 2012 年 3 月.
12. 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “閲覧者の観点に基づく Web ページ分割手法設計のための事例収集システムの実現,” 日本ソフトウェア科学会 第 28 回大会, 1E-1, 2011 年 9 月.
13. 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “確率モデルを用いた Web ブロックの役割推定手法とその応用,” 第 25 回 人工知能学会全国大会, 3F-14, 2011 年 6 月.
14. 佐野 博之, 伊藤 太樹, 柿元 宏晃, 平田 紀史, 白松 俊, 大囿 忠親, 新谷 虎松, “Web 閲覧者の視点を考慮した付箋アノテーション間のリンク構造に基づく情報推薦モデルの提案,” 第 72 回 情報処理学会全国大会, 3Q-7, 2010 年 3 月.
15. 佐野 博之, 平田 紀史, 白松 俊, 大囿 忠親, 新谷 虎松, “マッシュアップを利用した Web サービス構築支援システムとその応用,” 第 23 回 人工知能学会全国大会, 1B3-3, 2009 年 6 月.
16. 佐野 博之, 柿元 宏晃, 平田 紀史, 大囿 忠親, 新谷 虎松, “携帯電話向け情報編纂システムのためのパソコン用書類変換機構の試作,” 第 7 回 情報科学技術フォーラム, 第二分冊 pp.343-344, 2008 年 9 月.
17. 佐野 博之, 大囿 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 人工知能学会全国大会, 2G1-03, 2008 年 6 月.
18. 佐野 博之, 近藤 圭佑, 浅見 昌平, 大囿 忠親, 新谷 虎松, ”オンライン Web ページに基づく付箋アノテーションシステムとその応用,” 第 70 回 情報処理学会全国大会, 5R-6, 2008 年 3 月.
19. 佐野 博之, 大囿 忠親, 新谷 虎松, ”DOM ツリー解析に基づく Web ページへの付箋貼付けシステム,” 第 6 回 情報科学技術フォーラム, 第二分冊 pp.357-358, 2007 年 9 月.

修士論文

20. 佐野博之, “Web エージェントを用いたオンライン Web コンテンツへの付箋アノテーションに関する研究,” 名古屋工業大学 大学院工学研究科 情報工学専攻, 2010 年 3 月.

卒業論文

21. 佐野博之, “オンライン Web ページに基づく付箋アノテーションシスムとその応用,” 名古屋工業大学 工学部 情報工学科, 2008 年 3 月.

その他・共著

22. Robin M. E. Swezey, Hiroyuki Sano, Shun Shiramatsu, Tadachika Ozono and Toramatsu Shintani, “Automatic Detection of News Articles of Interest to Regional Communities,” International Journal of Computer Science and Network Security, Vol.12 No.6, pp.99-106, Jun. 2012. **査読付き**

23. Norifumi Hirata, Hiroyuki Sano, Robin M. E. Swezey, Shun Shiramatsu, Tadachika Ozono and Toramatsu Shintani, ”A Web Agent Based on Exploratory Event Mining in Social Media,” The 3rd IIAI International Conference on e-Services and Knowledge Management, pp.- , Sep. 2012. **査読付き**

24. Shun Shiramatsu, Norifumi Hirata, Robin M. E. Swezey, Hiroyuki Sano, Tadachika Ozono and Toramatsu Shintani, ”Gathering Public Concerns from Web towards Building Corpus of Japanese Regional Concerns,” The 3rd IIAI International Conference on e-Services and Knowledge Management, pp.- , Sep. 2012. **査読付き**

25. Shun Shiramatsu, Robin M. E. Swezey, Hiroyuki Sano, Norifumi Hirata, Tadachika Ozono and Toramatsu Shintani, ”Structuring Japanese Regional Information on the Web as Linked Open Data towards Supporting Concern Assessment,” The 4th International Conference on eParticipation, pp.73-84 , Sep. 2012. **査読付き**

26. Robin Swezey, Hiroyuki Sano, Norifumi Hirata, Shun Shiramatsu, Tadachika Ozono and Toramatsu Shintani, ”An e-Participation Support System for Regional Communities Based on Linked Open Data, Classification and Clustering,” The 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing(ICCI*CC 2012), pp.211-218, Aug. 2012. **査読付き**

27. Taiki Ito, Hiroyuki Sano, Tadachika Ozono and Toramatsu Shintani, “A Hierarchical Web Page Segmentation Algorithm Using Machine Learning,” The 11th International Conference on Intelligent Systems and Control, Nov. 2008. **査読付き**

28. 白松 俊, 平田 紀史, Robin M. E. Swezey, 佐野 博之, 大囿 忠親, 新谷 虎松, “LOD データセット SOCIA の復興促進への適用に向けた検討,” 第 28 回セマンティックウェブとオントロジー研究会, SIG-SWO-A1202-05, 2012 年 10 月.

29. 平田 紀史, 佐野 博之, Robin M. E. Swezey, 白松 俊, 大囿 忠親, 新谷 虎松, “住民参画 Web プラットフォームのための地域の社会問題に関する LOD 構築支援システム,” 第 22 回 Web インテリジェンスとインタラクション研究会, 2012 年 3 月.

30. 平田 紀史, Robin M. E. Swezey, 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “住民参画 Web プラットフォームのためのニュース記事と意見の構造化,” 第 24 回 セマンティックウェブとオントロジー研究会, pp.1-6, 2011 年 6 月.

31. 白松 俊, 平田 紀史, 佐野 博之, Robin Swezey, 大囿 忠親, 新谷 虎松, ”Linked Data を用いた住民参画 Web プラットフォーム O2,” 第 26 回 人工知能学会全国大会, 3C2-OS-13b-10, 2012 年 6 月.

32. 平田 紀史, 佐野 博之, Robin Swezey, 白松 俊, 大囿 忠親, 新谷 虎松, ”住民参画 Web プラットフォーム O2 における関連情報構造化システム,” 第 26 回 人工知能学会全国大会, 3C2-OS-13b-11, 2012 年 6 月.

33. 白松 俊, 佐野 博之, 平田 紀史, Robin M .E. Swezey, 大囿 忠親, 新谷 虎松, “Linked Open Data を用いたコンサーン・アセスメント支援機構の開発,” 第 45 回土木計画学研究・講演集, 2012 年 6 月.

34. 伊藤 太樹, 柿元 宏晃, 佐野 博之, 平田 紀史, 白松 俊, 大囿 忠親, 新谷 虎松, “階層的 Web ページ分割を用いたサブコンテンツ除去手法について,” 第 72 回 情報処理学会全国大会, 3R-9, 2010 年 3 月.

35. 柿元 宏晃, 伊藤 太樹, 佐野 博之, 平田 紀史, 白松 俊, 大囿 忠親, 新谷 虎松, “操作履歴と DOM 構造に基づく Web 行動分節化システム,” 第 72 回 情報処理学会全国大会, 5R-1, 2010 年 3 月.

36. 平田 紀史, 伊藤 太樹, 柿元 宏晃, 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, “イベントの属性抽出と系列化に基づくニュース記事閲覧支援システム,” 第 72 回 情報処理学会全国大会, 4V-6, 2010 年 3 月.
37. 工藤 聖広, 佐野 博之, 平田 紀史, 高崎 隼, 白松 俊, 大囿 忠親, 新谷 虎松, “スマートフォンを用いた分散共有ワークスペースに基づくプレゼンテーション資料管理システム,” 第 72 回 情報処理学会全国大会, 1Z-4, 2010 年 3 月.
38. 清水 堅, 土井 達也, 佐野 博之, 工藤 聖広, 白松 俊, 大囿 忠親, 新谷 虎松, “スライドシーンに基づくスライド作成支援システムの実現,” 第 72 回 情報処理学会全国大会, 1Z-5, 2010 年 3 月.
39. 柿元 宏晃, 佐野 博之, 大囿 忠親, 新谷 虎松, “Web 行動リプレイシステムに基づく Web アプリケーション動作検証システムとその応用,” 第 23 回 人工知能学会全国大会, 1D1-2, 2009 年 6 月.
40. 加藤 健太, 佐野 博之, 大囿 忠親, 新谷 虎松, “Web ページへの付箋アノテーションを用いたニュース記事閲覧支援システム,” 第 71 回 情報処理学会全国大会, 3N-2, 2009 年 3 月.
41. 高崎 隼, 佐野 博之, 大囿 忠親, 新谷 虎松, “オフライン Web 技術に基づく付箋アノテーションシステムの試作,” 第 71 回 情報処理学会全国大会, 3N-3, 2009 年 3 月.
42. 工藤 聖広, 佐野 博之, 大囿 忠親, 新谷 虎松, “スマートフォンのためのカードモデルを利用したコンテンツ開発環境の実現,” 第 71 回 情報処理学会全国大会, 1Q-8, 2009 年 3 月.
43. 柿元 宏晃, 佐野 博之, 平田 紀史, 大囿 忠親, 新谷 虎松, “カードモデルに基づく情報編纂システムを利用したレシピ検索システムの試作,” 第 7 回 情報科学技術フォーラム, 第二分冊 pp.347-348, 2008 年 9 月.
44. 平田 紀史, 柿元 宏晃, 佐野 博之, 大囿 忠親, 新谷 虎松, “携帯電話向け情報編纂システムのためのコンテンツ作成システムの試作,” 第 7 回 情報科学技術フォーラム, 第二分冊 pp.345-346, 2008 年 9 月.
45. 大囿 忠親, 柿元 宏晃, 佐野 博之, 平田 紀史, 新谷 虎松, “携帯電話における情報閲覧支援のための情報編纂システムについて,” 第 7 回 情報科学技術フォーラム, 第二分冊 pp.341-342, 2008 年 9 月.
46. 兼岩 竜之介, 佐野 博之, 白松 俊, 大囿 忠親, 新谷 虎松, ”Web 上の論文データを利用した学術英文典型度評価システム,” 平成 21 年度 電気関係学会東海支部連合大会, O-046, 2009 年 9 月.
47. 公益財団法人 堀科学芸術振興財団 平成 23 年 第 21 回研究助成内定, 2012 年 3 月.
48. LOD チャレンジ実行委員会 LOD チャレンジ Japan 2011 チャレンジデー賞受賞, 2012 年 3 月.
49. 独立行政法人 日本学生支援機構 大学院第一種奨学金 特に優れた業績による返還免除 (全額), 2010 年 5 月.
50. 名古屋工業大学大学院 平成 21 年度 学術活動部門 副学長賞受賞, 2010 年 2 月.
51. 公益財団法人 NEC C&C 財団 2009 年度前期 国際会議論文発表者助成内定, 2009 年 5 月.