ウェブ検索エンジンのための
テキスト種別特定に関する研究

2014 年 3 月

吉田 光男

ウェブ検索エンジンのための
テキスト種別特定に関する研究

吉田 光男

システム情報工学研究科
筑波大学

2014 年 3 月

目次

第 1 章　はじめに　1
1.1　本研究の背景及び目的 1
1.2　ウェブ検索エンジン利用者に対する本研究の貢献 5
1.3　本論文の構成 8

第 2 章　関連研究　11
2.1　コンテンツ抽出に関する研究 11
2.2　検索クエリの分類に関する研究 18

第 3 章　ウェブページのコンテンツ並びにポスト及びコメントの自動特定　21
3.1　本章の概要 21
3.2　コンテンツ及びポスト・コメントの定義 22
3.3　提案手法 22
3.4　評価実験 34
3.5　本章のまとめ 50

第 4 章　集中的に検索されるクエリを対象としたカテゴリ特定　51
4.1　本章の概要 51
4.2　集中的に検索されるクエリの例 52
4.3　提案手法 54
4.4　評価実験 60
4.5　本章のまとめ 68

第 5 章　おわりに　69

参考文献　71

図目次

1.1 ウェブで流通するテキスト量の変化 2
1.2 ウェブページに占めるコンテンツの例（破線部分） 3
1.3 ウェブ検索エンジンにおける本研究の位置付け 4
1.4 ウェブ検索及びニュース検索で「朝日新聞」と検索した例 6
1.5 検索スニペット（サマリー）の例 7
1.6 ウェブ検索で「小保方晴子」と検索した例 8
1.7 ウェブ検索で「eclipse」と検索した例 9
1.8 Google における「eclipse」の検索頻度の変化（2012 年） 9

2.1 ルールを用いる手法の概要 12
2.2 経験則による手法の概要 14
2.3 機械学習による手法の概要 15
2.4 ウェブページ集合を用いる手法の概要 16

3.1 ExtractUniqueBlock 法の着目点 24
3.2 分離抽出手法の着目点 25
3.3 リスト 3.1 の HTML コードを DOM ツリーで表現した結果 27
3.4 図 3.3 の一部 27
3.5 図 3.4 の DOM ツリーを簡素にレンダリングした結果 27
3.6 図 3.4 の DOM ツリーから 2 つのブロックを抽出した結果 28
3.7 図 3.6 の抽出結果を簡素にレンダリングした結果 28
3.8 DOM ツリーから 5 つのブロックを抽出した例（破線部分） 29
3.9 親ノード優先と兄ノード優先との違い 32
3.10 ブロック識別子の付与処理の例 1 32
3.11 ブロック識別子の付与処理の例 2 33

3.12 各ブロックにブロック識別子を付与した例 34
3.13 ポストとコメントの分離抽出の例 35
3.14 正解データ作成用のアノテーションツール 37
3.15 連続するコンテンツ部分の抽出失敗例（破線部分） 46
3.16 実験結果のブログ記事ページ例（分離抽出手法） 47
3.17 ページ数と処理時間及びコンテンツ抽出性能との関係 48
3.18 コサイン類似度の閾値とコンテンツ抽出性能（%）との関係 48

4.1 Google トレンドで調査した検索頻度（2012 年） 52
4.2 クエリ「カレログ」の検索頻度及び外部リソースの状況（2011 年） . . . 53
4.3 Twitter におけるツイートの例 55
4.4 クエリとユーザの関係 55
4.5 クエリとハッシュタグの関係 57
4.6 構築されるグラフ G の形状 59
4.7 はてなキーワードの項目ページ例（カレログ） 61
4.8 使用するクエリのカテゴリ分布 61
4.9 実験方法 62
4.10 パラメータ ρ の調整 (QueryGraph) 65
4.11 パラメータ α の調整 (QueryTwitterGraph, $\rho = 0.2$) 65
4.12 クエリとユーザ及びハッシュタグとのグラフ構造（UserGraph） 67
4.13 クエリ間のグラフ構造（QueryGraph） 68

表目次

3.1 データセットの詳細 38
3.2 正確度（Accuracy）による分離抽出性能の比較 40
3.3 F 値による分離抽出性能の比較 41
3.4 提案手法によるポストとコメントの分離抽出性能詳細 42
3.5 分離抽出処理のみの性能比較（F 値） 44
3.6 コンテンツ再抽出の有無による平均抽出性能の比較（マイクロ平均） . . 45

4.1 提案手法によるクエリ分類性能（%） 64
4.2 クリッキングログとの比較実験結果（%） 66

第 1 章

はじめに

1.1　本研究の背景及び目的

ウェブ検索エンジンは，ウェブで流通するコンテンツの増加に伴い，インターネットを利用する際には欠かせないサービスとなっている。総務省による調査では，図 1.1 が示すようにウェブで流通するテキスト量は 2003 年から 2009 年までの 6 年間で 6 倍にまで増加している [東川 09]。また，ウェブ検索エンジンのクローラがアクセス可能なサーフェース・ウェブの量は，約 2,600 万ページ（1998 年）から約 1 兆ページ（2008 年 7 月）まで増加している [Alpert 08]。膨大なウェブページの中から求めるウェブページを見つけるにはウェブ検索エンジンが必要であり，日本で実施された 2008 年の調査では，インターネット利用者の 93.7% が一般的なウェブ検索エンジンを日常的に利用しているという結果が出ている [東川 09]。

ウェブで流通するコンテンツの増加は，ブログなどの利用者（消費者）がコンテンツを作成するメディア（CGM: Consumer Generated Media）の普及に一因がある。日本では，2004 年から 2005 年ごろにかけてブログコンテンツが急増しており，現在も増加傾向が見られる [佐伯 09]。従来，利用者は企業から提供される情報，商品，サービスを消費するだけの存在であり，ウェブで流通するコンテンツは企業や行政機関の公式サイトが中心であった。CGM の普及により，企業や行政機関のみならず，一般のインターネット利用者もコンテンツを作成するようになり，ウェブで流通するコンテンツは多様化している。流通するコンテンツが多様化するのに伴い，ウェブ検索エンジン利用者（コンテンツを探す者）の検索要求も多様化している。今後，ウェブ検索エンジンは，ブログなどにあらわれる新しい種類のテキストに対応した上で，利用者の様々な検索要求に応えていく必要がある。

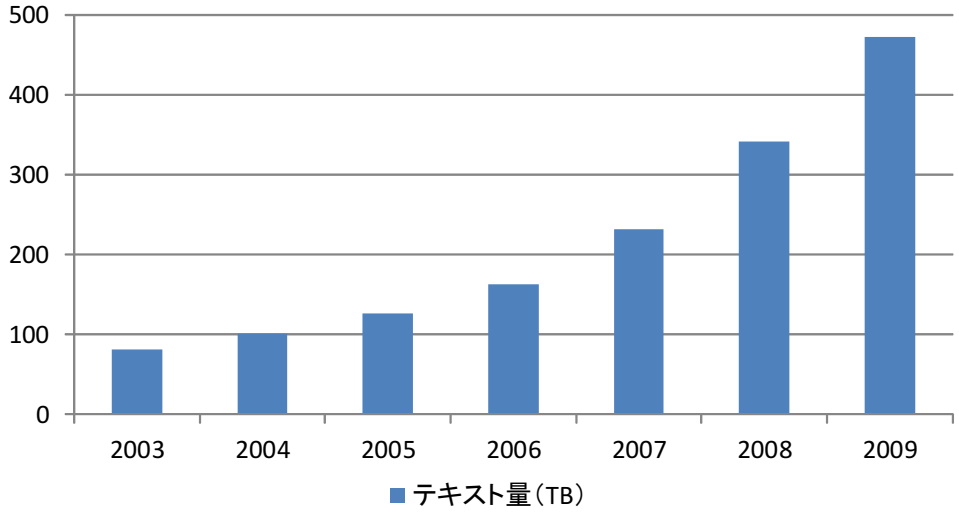

図 1.1　ウェブで流通するテキスト量の変化

一般利用者がコンテンツを作成するようになったのは，ウェブページのコンテンツ管理を支援する CMS（Content Management System）の普及にある。従来，コンテンツを作成し，ウェブ上に公開するには専門的な知識が必要であったが，CMS によって専門的な知識無しで公開できるようになった。現在では，ブログサイトはもちろんのこと，ニュースサイトや企業のサイトも CMS によって管理されるようになってきている。CMS の中心的な機能は，あらかじめ設定したテンプレートに基づきウェブページを生成する機能である。この機能により，ウェブページの作成が容易になったが，その一方で，各ウェブページにメニューや広告を過剰に付加されるようになり，ページに占めるコンテンツは縮小している。例えば，図 1.2 に示すニュースページ[*1]には，ヘッダ，メニュー，広告，関連記事リストなど不要部分が多々存在することによりページに占めるコンテンツの割合が低いことがわかる。そのため，ブログをはじめとするウェブページのコンテンツを適切に検索させるためには，ウェブページのテキストの中からコンテンツを特定する必要がある。

ウェブ検索エンジン利用者の検索要求が多様化し，それらの検索要求に “適した” ウェブページを提示するためには，入力された検索クエリが含まれるウェブページを提示するだけでは不十分になってきている。この要因は，検索クエリに含まれる語の多義性と未知性にある。例えば，「アップル」はある文脈では「コンピュータ関連企業の Apple Inc.」を指し，また別の文脈では「果物のアップル」を指す。これが多義性の問題であり，検索クエリとして「アップル」が入力されたとき，ウェブ検索エンジンはいずれの意味（カテ

[*1] 左: `http://www.47news.jp/CN/201308/CN2013081301002124.html` (accessed 2013-12-27)
右: `http://www.47news.jp/CN/201311/CN2013111901002327.html` (accessed 2013-12-27)

図 1.2　ウェブページに占めるコンテンツの例（破線部分）

ゴリ）であるかを特定した上で，適したウェブページを提示する必要がある。未知性の問題とは，「ねこねこりんごのいぬ*2」のように一般に知られず，検索クエリが含まれるウェブページが存在しない検索クエリに対し，どのようなコンテンツを提示するかという問題である。この場合，利用者はアニメの情報を探していると推察され，何らかのアニメ情報を提示することが考えられる。これらのように，検索要求が多様化している今日，検索クエリのカテゴリを分類（識別）することで，ウェブ検索エンジンの性能向上を目指していく必要がある。

以上を踏まえ，図 1.3 が示すように，本研究は，検索対象及び検索入力の両面に対してその種別を特定することにより，ウェブ検索エンジンの性能向上を目指す。検索対象及び検索入力のそれぞれに対し，新たな手法を提案する。

*2 筆者が作成した架空のアニメキャラクター。

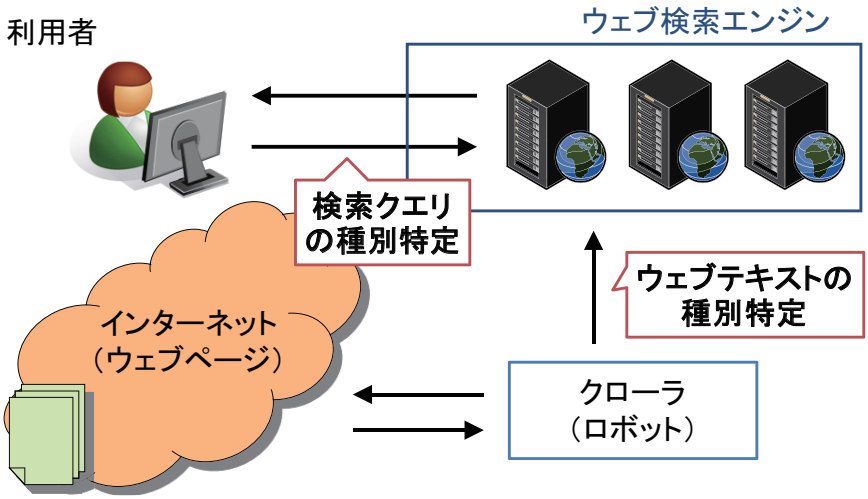

図 1.3　ウェブ検索エンジンにおける本研究の位置付け

研究内容 1: ウェブテキストの種別特定
ウェブページのコンテンツ並びにポスト及びコメントの自動特定

ウェブページの内容はコンテンツと不要部分に大別でき，機械的な処理を行う前処理として，コンテンツ抽出を行う必要がある。例えば，Yi ら [Yi 03] は，ウェブニュース記事ページのクラスタリングを行う際，事前にコンテンツ抽出を行うことにより，性能が大幅に向上することを報告している。

さらに，近年増加しているブログのコンテンツは，ポストと呼ばれるブログの書き手によるコンテンツと，コメントと呼ばれるブログの読み手によるコンテンツに大別できる。ポストとコメントの存在はブログの特性の 1 つであり，ブログのコメントを利用する研究が行われ始めるなど [Hu 07, Hu 08, Bhattarai 09, Parapar 10, Ma 12]，ブログのコンテンツ抽出はポストとコメントを分離抽出することが期待される。例えば，Li ら [Li 07] はブログ記事ページをクラスタリングするタスクにおいて，ポストとコメントを分割し，コメントにウェイトをかけることでクラスタリング精度が向上することを報告している。

本研究では，あるウェブページのコンテンツは他のウェブページに出現しないという仮説をもとに，ウェブページ集合を与えさえすれば，全自動でコンテンツを抽出する手法を提案する。さらに，複数のブログ記事ページを含むあるブログサイトにおいて，ポストは全ての記事ページに出現する一方で，コメントは一部の記事ページにしか付かない傾向があることに着目し，ポスト及びコメントの分離抽出手法も提案する。

研究内容 2: 検索クエリの種別特定
集中的に検索されるクエリを対象としたカテゴリ特定

入力される検索クエリの頻度に着目すると，恒常的に検索されるクエリ，周期的に検索されるクエリ，集中的に検索されるクエリに大別できる。従来のクエリ分類手法では主に恒常的に検索されるクエリを対象としており，集中的に検索されるクエリを対象とした研究は行われていなかった。集中的に検索されるクエリは未知の語である可能が高く[*3]，ウェブ上にも適した情報が少ないが故に関連する情報も提示する必要がある。その一貫として，適切なクエリ分類が望まれる。

本研究では，集中的に検索されるクエリを対象としたクエリ分類手法を提案する。このようなクエリと対象とする場合，分類に必要な素性が欠乏するという問題があるが，即時性の高いソーシャルメディアのデータを利用することで，この問題を解決する。

1.2　ウェブ検索エンジン利用者に対する本研究の貢献

本研究はウェブ検索エンジンの性能向上を目的としている。本節では，取り組む研究課題の成果がウェブ検索エンジン利用者にどのような影響を与えるかについて述べる。

1.2.1　ウェブページのコンテンツ並びにポスト及びコメントの自動特定

ウェブページのコンテンツの特定により，検索結果の改善が期待できる。図 1.4 はグーグル社のウェブ検索エンジン及びニュース検索エンジンで「朝日新聞」と検索した結果である[*4]。左側がウェブ検索の結果であり，最上位に朝日新聞（asahi.com）のトップページが提示されている。一方，右側がニュース検索の結果であり，上位 10 件の全てに朝日新聞社が発信した記事ページが提示されている。

検索エンジンを利用する目的（検索意図）は，次の 3 種類にわけられる [Broder 02]。

Navigational　あるウェブサイトにたどり着きたいという意図
Informational　ある事柄に関する情報を知りたいという意図
Transactional　ある事柄に関するモノが欲しいという意図（例：動画）

ニュース検索を行う利用者は，ある事柄に関する情報を知りたいという検索意図である

[*3] 例えば，ある新製品が発表された場合，未知である新製品名の検索が急増する。
[*4] `https://www.google.co.jp/`（2013 年 12 月 30 日に検索）

図 1.4　ウェブ検索及びニュース検索で「朝日新聞」と検索した例

Informational な検索を行う割合が高いという報告があり [Li 08b]，先の検索結果を検討すると，ウェブ検索の結果は利用者の意図通りの結果であるものの，朝日新聞に関する情報（ニュース）を含まないニュース検索の結果は利用者の意図に反している可能性がある。ニュース記事本文（コンテンツ）を適切に抽出し，本文のみを検索できるようにすれば，ニュース検索も利用者の意図したとおりの検索結果を提示できると考えられる。

検索結果の改善以外に，検索結果に提示されるスニペット（サマリー）の改善も期待できる。図 1.5 のように，検索結果にはウェブページのタイトルとともにスニペットも表示される。検索結果で提示されるスニペットは指示的（indicative）なものであり，利用者はそのスニペットを読んでページを訪問するかどうかを判断する [奥村 04]。このスニペットは，検索クエリが出現する周辺のテキストを表示しているに過ぎず，必ずしもそのページの内容そのものを表しているわけではない。これまでに，ページ内に出現するポスト及びコメントの情報を用いて，適切な要約を行う手法が提案されており [Ma 12, Parapar 10, Hu 08]，本論文で提案する自動抽出手法と組み合わせれば，検索スニペットの内容を改善できる可能性がある。

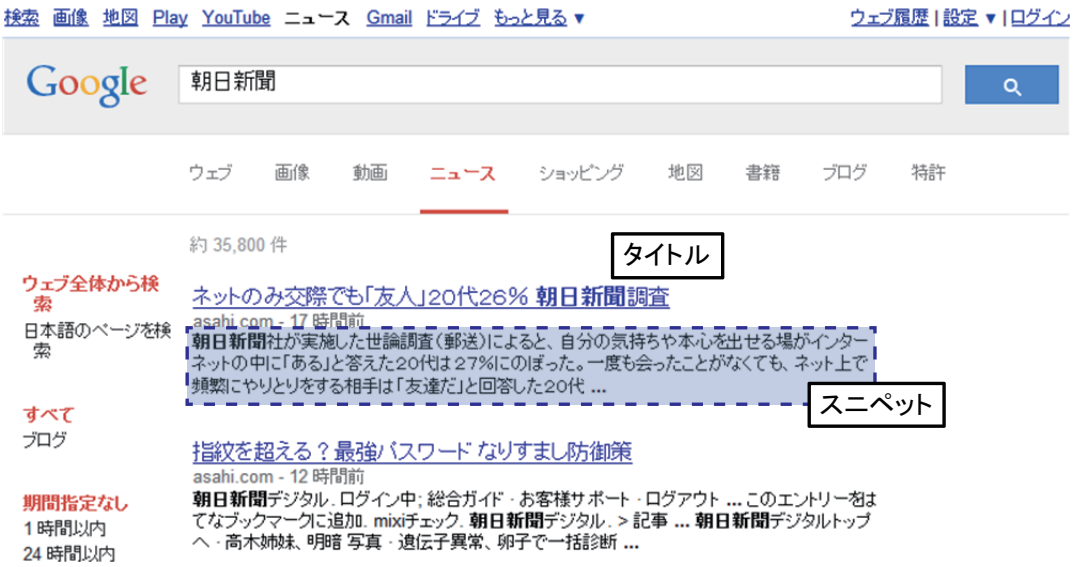

図 1.5　検索スニペット（サマリー）の例

1.2.2　集中的に検索されるクエリを対象としたカテゴリ特定

現在の検索エンジン利用者は，広告もコンテンツの一部として認識しており [Broder 08]，適切な広告を提示することは，利用者の利益にかなう。図 1.6 は Yahoo! JAPAN のウェブ検索エンジンで「小保方晴子」と検索した結果である[*5]。Yahoo! JAPAN の検索結果では，通常，右部分（図の破線部分）に検索クエリに適合する広告が表示される。しかし，本検索結果では，検索クエリに合致する広告が見つからず，空白になっている。クエリのカテゴリを特定し，そのカテゴリに応じた広告などの付加情報を提示できれば，利用者の利便性が向上すると考えられる。

集中的に検索されるクエリは，同じクエリであっても恒常的に検索されているときと異なる目的で検索される場合がある。例えば，Google で「eclipse」と検索すると，図 1.7 のように上位には総合開発環境 “Eclipse” のダウンロードページ[*6]が提示される[*7]。この検索結果が適切であるかどうかは，検索する時期によって異なる。図 1.8 が示すように 2012 年 5 月 20 日，21 日には「eclipse」が集中的に検索されているが[*8]，この時期，金環日食が発生しており，総合開発環境を調べる目的とは異なる目的で検索されていると推察

[*5] `http://www.yahoo.co.jp/`（2014 年 2 月 3 日に検索）
[*6] `http://www.eclipse.org/downloads/`
[*7] 2013 年 12 月 30 日に検索した。
[*8] Google トレンド（`http://www.google.com/trends/`）による調査。縦軸の値は，最大値を 100 とする相対値である。

図 1.6　ウェブ検索で「小保方晴子」と検索した例

される。この時期においては，金環日食に関するページを上位に表示することが適切であると考えられ，検索クエリのカテゴリを適切に特定できれば，検索結果を目的に応じて変更できる可能性がある。

1.3　本論文の構成

まず，第 2 章では，コンテンツ抽出とクエリ分類について，それらに関係する研究について述べる。第 3 章では，ウェブページからのコンテンツ並びにポスト及びコメントの自動抽出手法を提案し，日本語のブログ記事データを用いた実験により，従来手法よりも高い精度で抽出できることを示す。第 4 章では，集中的に検索されるクエリの分類手法を提案し，応用として想定されうる実用的な実験環境において，提案手法が有効に機能したことを示す。最後の第 5 章にて，本研究の成果をまとめ，今後の課題について述べる。

図 1.7　ウェブ検索で「eclipse」と検索した例

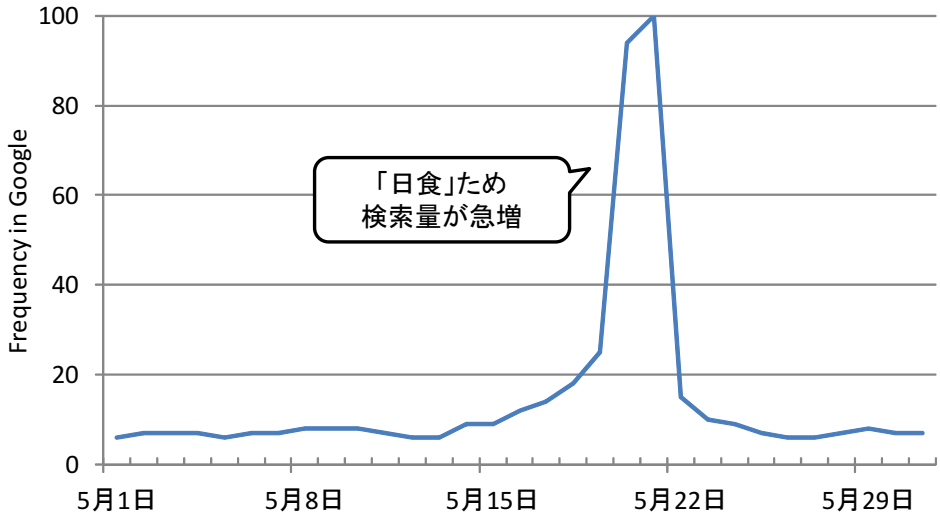

図 1.8　Google における「eclipse」の検索頻度の変化（2012 年）

第 2 章

関連研究

2.1　コンテンツ抽出に関する研究

本研究では，ウェブページからコンテンツを抽出する手法並びにブログのポスト及びコメントを分離抽出する手法を提案するが，コンテンツを抽出する手法の先行研究は多数あるものの，ポスト及びコメントを分離抽出する手法の先行研究は少ない。まず，ポスト及びコメントを区別せず，コンテンツの抽出を行う先行研究について述べる。それらの研究事例を踏まえた上で，ポスト及びコメントを分離抽出する手法について述べる。

コンテンツの抽出を行う研究は，これまでに多数行われている。それらの手法を分類すると，以下の 4 手法に分類できる。

- 人手によって作成されたルールを用いる手法
- 経験則による手法
- 人手によって作成された訓練データを用いた機械学習による手法
- ウェブページ集合から自動的に抽出する手法

2.1.1　人手によって作成されたルールを用いる手法

ルールを用いる手法の概要を図 2.1 に示す。Gupta ら [Gupta 03, Gupta 05b, Gupta 05a] は，不要部分のルールを生成し，このルールを適用することにより不要部分を削除し，コンテンツを抽出する手法を提案している。また，正規表現 [Hemenway 04] や XPath[*1]，ラッピング言語 [澤 08] を用いることにより，コンテンツ抽出プログラムを

[*1] XML Path Language：XML 文書の特定部分を表現する言語 [Clark 99]。

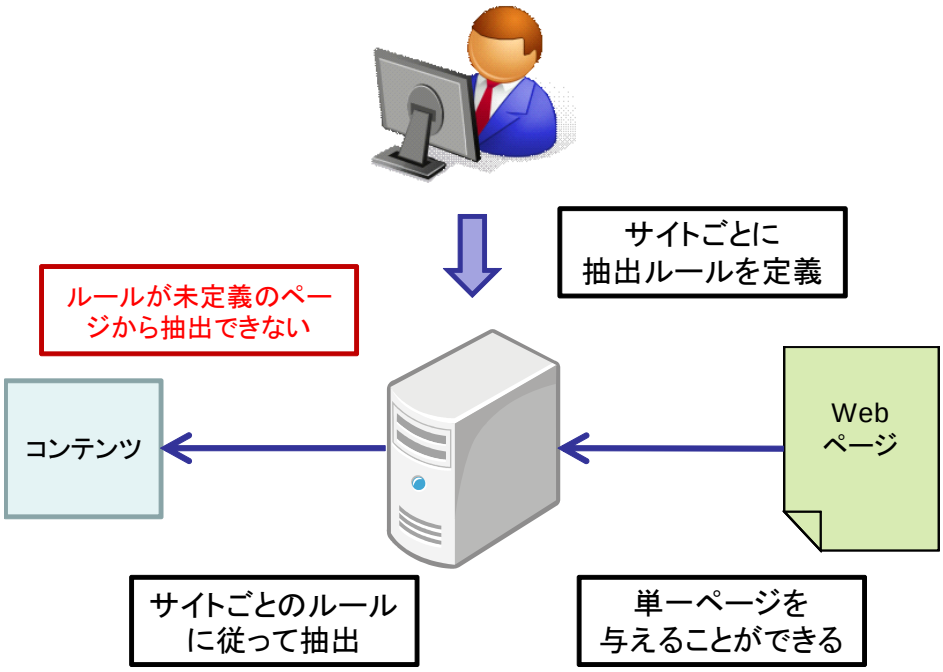

図 2.1　ルールを用いる手法の概要

作成する方法もある。このようなルールを用いる手法は，ルールが適用可能な範囲であれば，有効に機能する。さらに，他の手法よりも，軽量に作動する傾向がある。しかし，この手法で除去できる不要部分は，ルール作成者に依存するため，ルール作成者が考慮していない不要部分やウェブページには適用できない。

抽出ルールを手法の提案者に依存させない試みとして，コンテンツ抽出の基盤システムを提供し，多数のボランティアユーザによって抽出ルールを作成するプロジェクトが存在する。このようなプロジェクトとして，RSS リーダにブログ記事本文を表示することを目的とした LDR Full Feed[*2]などで使用するルールを wedata[*3]に登録するプロジェクト[*4]が挙げられる。このプロジェクトは 2008 年 4 月に開始され，2013 年 6 月 22 日の時点で，12,208 サイトのコンテンツ抽出ルールが登録されている。ただし，2013 年 10 月 23 日からアクセス不能になっており，将来における利用可能性は不透明な状態である。

[*2] `http://userscripts.org/scripts/show/22702` (accessed 2013-12-23)
[*3] `http://wedata.net/` (アクセス不能 2013-12-23)
`http://web.archive.org/web/20130729031938/http://wedata.net/` (accessed 2013-12-23)
[*4] `http://wedata.net/databases/LDRFullFeed` (アクセス不能 2013-12-23)
`http://web.archive.org/web/20130622190311/http://wedata.net/databases/LDRFullFeed` (accessed 2013-12-23)

2.1.2　経験則による手法

人手によって作成されたルールを用いる方法では，ルールの適用対象が限られているという問題が存在した。経験則による手法では，コンテンツ出現方法を検討し，あらゆるサイトに適用可能なアルゴリズムを人手によって開発する。この手法の概要を図 2.2 に示す。

Finn ら [Finn 01] は，HTML ドキュメント（ウェブページ）をテキストと HTML タグが混在した配列と見なし，コンテンツにおけるテキストの割合を最大化し，それ以外の部分におけるタグの割合を最大化するような分割位置を見つけ，コンテンツを抽出する Body Text Extraction (BTE) アルゴリズムを提案している。この手法には，ウェブページのコンテンツが 1 か所であることを前提にしているため，コンテンツと不要部分が交互に出現するようなウェブページには適用が難しい。BTE アルゴリズムの改良として，Pinto ら [Pinto 02] が提案した Document Slope Curves (DSC) アルゴリズムがあり，さらに DSC を改良した Advanced DSC (ADSC) が，Gottron[Gottron 07] によって提案されている。また，Pappas ら [Pappas 12] も着目点を同じとする抽出手法を提案している。

Gottron[Gottron 08b] は，HTML ドキュメントの行ごとのテキスト量に着目し，そのヒストグラムを作ることによりコンテンツを特定する手法を提案した。Weninger ら [Weninger 10] は，[Gottron 08b] を一般化し，HTML ドキュメントの行ごとの HTML タグの割合に着目した手法を提案した。まず，各行のタグの割合 T_i を求め（i は行番号），その割合を平滑化する（T'_i）。次に，T'_i における近傍との差 G_i を求め，その差を平滑化する（G'_i）。そして，T'_i と $|G'_i|$ を 2 次元空間に配置し，k-means[MacQueen 67] によって 3 クラスタに分割する。分割の際，1 つのクラスタの重心を原点に固定する。最後に，原点を重心とするクラスタを不要部分とし，それ以外の部分をコンテンツとして抽出する。コンテンツと不要部分が交互に出現するようなウェブページにも適用が可能であるが，コンテンツの大部分が画像であるようなウェブページには適用が難しい。

その他，Mantratzis ら [Mantratzis 05] は，HTML ドキュメントをテーブルタグ（TABLE）などで分割を行い，リンク要素（A）の多い部分を不要部分とし，それ以外をコンテンツとして抽出する手法を提案している。また，複数の経験則による手法をまとめるアプローチとして，*CombinE* system[Gottron 08a] が挙げられる。

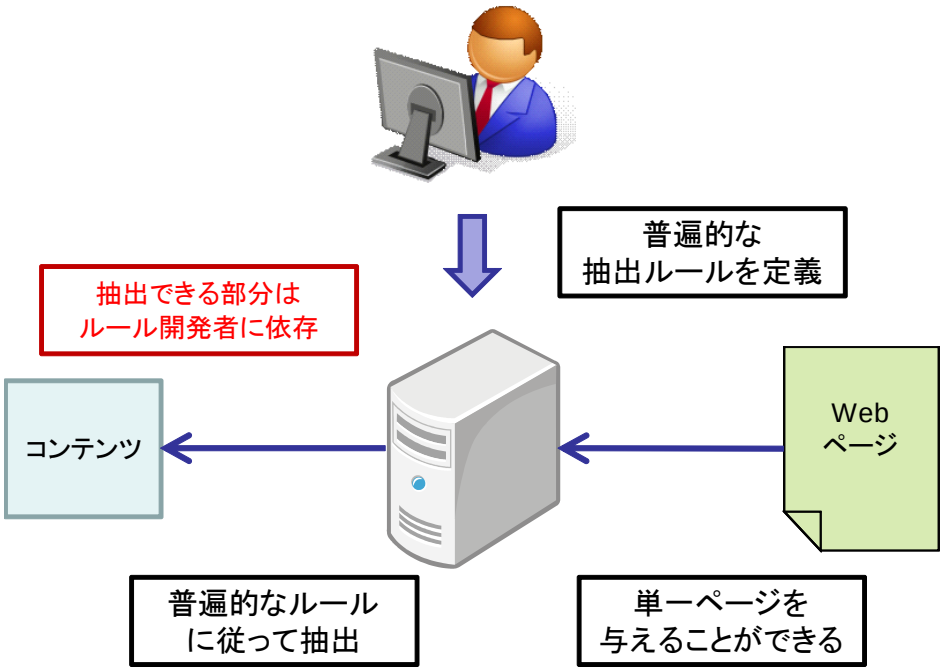

図 2.2　経験則による手法の概要

2.1.3　人手によって作成された訓練データを用いた機械学習による手法

経験則による手法は，人手によって作成されたルールを用いる手法よりも多くのサイトに適用可能であるが，提案者が考慮していない不要部分やウェブページには適用できないという問題は残る。機械学習による手法では，人手によって作成された訓練データを用いることにより，コンテンツ抽出モデルを自動獲得する。この手法の概要を図 2.3 に示す。

Bing ら [Bing 08] は，ウェブページを一連のセルと見なし，各セルに文字数，句読点数などに応じたスコアを付与した後，そのスコアの大きさを山に見立て，平均的なウェブページではどの山がコンテンツであるかを学習し抽出する手法を提案している。この手法は，機械学習によるモデル獲得を試みるものの，スコアの付与方法は，経験則に基づいている。

Cai ら [Cai 03] は，背景色，フォントサイズなどブラウザによってレンダリングされた結果をもとにセグメンテーションを行う，VIPS アルゴリズムを提案した。Song ら [Song 04] は，VIPS アルゴリズムを利用し，各セグメントの重要度を人手で付与した訓練データを用い，SVM を使った機械学習によってセグメントの重要度を判定する手法を提案した。VIPS アルゴリズムは，コンテンツと不要部分を分断しない傾向があり，Song

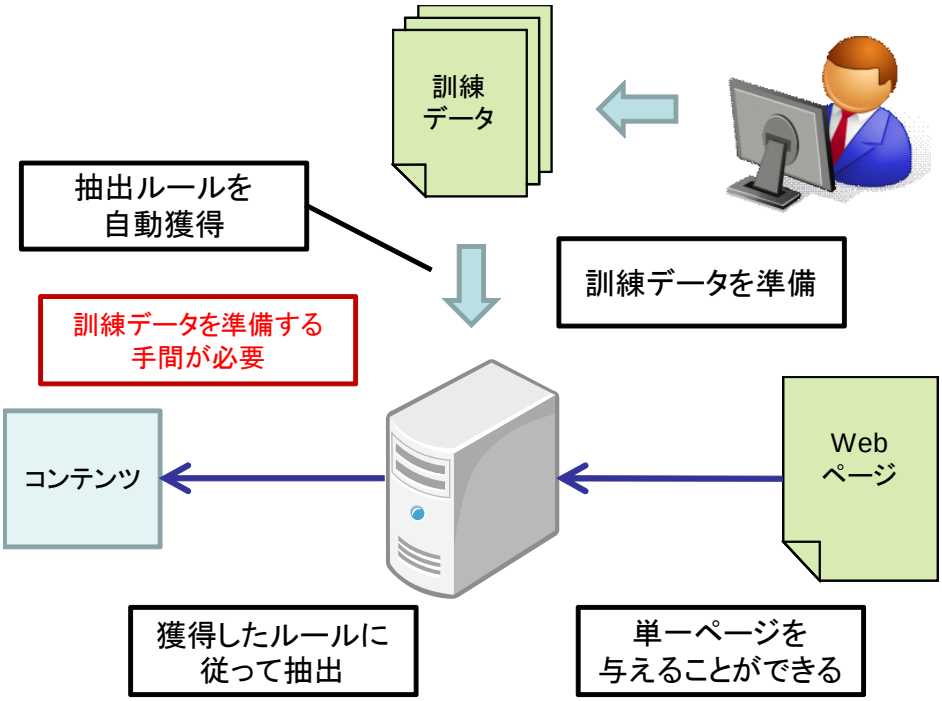

図 2.3　機械学習による手法の概要

らの手法をそのままコンテンツ抽出に適用するのは難しい。

鶴田ら [鶴田 08] は，平均的なウェブページにおいてウィンドウのどの位置にコンテンツが出現するかという情報をモデル化する手法を提案している。この手法は，1 つの識別モデルに集約するため，平均的なウェブページから外れたレイアウトを持つ場合，適用が困難であるという問題があった。モデル適用の際，レイアウト情報を加味することにより，この問題を解決した手法も提案されている [鶴田 10]。

2.1.4　ウェブページ集合から自動的に抽出する手法

機械学習による手法は，人手によって訓練データを準備する必要があるほか，訓練データの品質によって抽出性能が左右されるという問題点がある。これまで述べた 3 手法は，単一のウェブページを入力として受け付ける。大量のウェブページをマイニングする際には，事前にクローリングが必要であり，ウェブページ集合を準備することができる。ウェブページ集合を用いることにより，訓練データを必要としないコンテンツ抽出手法が提案されている。この手法の概要を図 2.4 に示す。

Lin ら [Lin 02] は，サイト内のウェブページを収集し，ページ中の部分の情報量を計算することによりコンテンツを抽出する手法を提案している。この手法では，計算量が

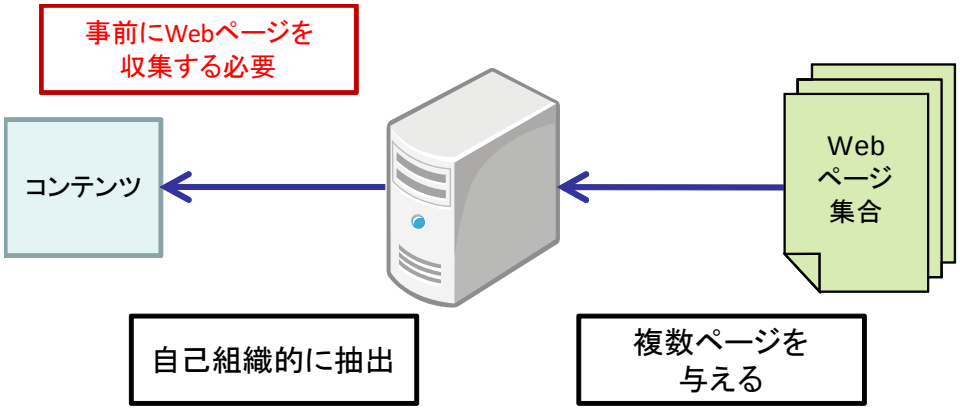

図 2.4　ウェブページ集合を用いる手法の概要

大きくなる傾向があるため，Debnath ら [Debnath 05] は，計算量を小さくした IBDF（Inverse Block Document Frequency）と呼ばれるサイト内におけるページ中の部分の重要度スコアを計算することによりコンテンツを抽出する手法を提案している。Debnath らの手法には大きくわけて 2 つの問題点が存在する。1 つ目の問題点は，コンテンツ候補の抽出に tag-set と呼ばれるコンテンツと不要部分を分断しやすいタグのリストが必要であり，このリストはウェブページデザインの流行に左右されることである。Debnath らは，ニュースページはテーブルタグ（TABLE）によってコンテンツと不要部分が分断される傾向があるため，優先的に分割するのがよいと主張している。しかし，筆者の調査では，現在，ニュースページはテーブルタグによってコンテンツと不要部分が分断されておらず，この知識が古くなっていることがわかっている。2 つ目の問題点は，IBDF を計算した後，各ウェブサイトに適した閾値を決定しコンテンツを抽出するということである。ウェブ上には無数のウェブサイトが存在しており，全てのウェブサイトに適切な閾値を決定することは困難である。

2.1.5　ポスト及びコメントを分離抽出する手法

前項までは，コンテンツ抽出一般について述べた。本項では，コンテンツを細分化し，ポスト及びコメントとして分離抽出する手法について述べる。

2.1.1 項で述べたルールを用いる方法は，ポスト及びコメントそれぞれの抽出ルールを作成することで適用可能である。しかしながら，先にも述べたように，この手法はルール作成者が考慮していない不要部分やウェブページには適用できない。同一のブログホスティングサービスを利用している記事ページに関していえば，ポストやコメントの場所が同一であると考えられるため，ブログホスティングサービスごとに抽出ルールを準備

する必要があるものの，人手によって抽出ルールを記述する方法は有効に機能する。しかし，筆者らによる予備調査では，人気のあるブログサイトの少なくとも 30% がブログホスティングサービスを利用していないことがわかった[*5]。また，2013 年 12 月 1 日現在，JP ドメインだけでも 1,353,969 ドメイン存在しており，人手によって全てのサイトのルールを作成するのは困難であると考えられる。このような状況下では，人手によって抽出ルールを記述する方法には多大な労力が必要となることは明らかであり，人手の負荷を要しない自動抽出手法を検討する必要がある。

前項までに述べたコンテンツ抽出手法をブログ記事ページに適用するとなると，これらの手法はポストとコメントを区別せずに抽出することになる。これらの手法に対し，ブログのポストまたはコメントのみを抽出する手法も提案されている。Chang ら [Chang 12] は，Weninger ら [Weninger 10] のコンテンツ抽出手法を改良した上で，“comment” という文字列が出現する位置より後方のテキストを除外することでポストの抽出を行う手法を提案している。Kao ら [Kao 10] は，コメント部分は複数の投稿で構成されることに着目し，HTML の繰り返しパターンとそのパターンに含まれる情報を用いた機械学習アプローチによってコメントを抽出する手法を提案している。Song ら [Song 10] は，ブログサイトのトップページのような複数の記事が含まれるページから，それぞれの記事を抽出する手法を提案し，コメントの抽出に応用した結果も報告している。

Cao ら [Cao 08] は，ポストとコメントを分断する境界線を見つけることにより，ポストとコメントを自動的に分離抽出する手法を提案している。この手法は，視覚情報とテキスト情報をもとにブログ記事ページのコンテンツを特定する処理と，情報量をもとにコンテンツの中でポストとコメントを分断する境界線を見つける処理の 2 段階にわけられる。彼らの手法は，コンテンツを 1 か所に特定するため，コンテンツの中に不要部分を多く含む傾向がある。また，コンテンツを特定する際，HTML ファイルのほかに，画像ファイルや外部スタイル情報（スタイルシート: Cascading Style Sheets）も必要になるため，情報検索システムなど，他のシステムに組み込むのは困難であると考えられる。

ポストとコメントを同時に自動抽出する研究は，筆者らが調査した限り，Cao ら以外に提案されていない。既存のポスト抽出手法とコメント抽出手法を組み合わせることにより，ポスト及びコメントを同時に抽出する方法が考えられるが，自動抽出が不完全であるが故に，ポスト抽出手法の結果とコメント抽出手法の結果とが一部重複する状況が考えられる。そのため，この重複部分をポストまたはコメントのいずれと見なすかの処理そのも

[*5] livedoor Reader（`http://reader.livedoor.com/`）登録者数ランキング上位 1,000 サイトから推定した。

のが，ポスト及びコメントの分離抽出手法となり得る。また，コンテンツ抽出手法とコメント抽出手法を組み合わせた場合においても，コメント抽出手法でのみ抽出できた部分を，コメントと見なすか，不要部分と見なすかの判断の余地が残る。ほかにも，それぞれの抽出手法において，コンテンツ，ポスト，コメントの最小単位が異なる点も手法の組み合わせを困難にする要因である。例えば，Cao ら [Cao 08] が提案する処理の一部（コンテンツ特定処理）を Weninger ら [Weninger 10] が提案する手法に置き換えると，ポストとコメントを分断する境界線を見つける処理に必要な情報が欠けてしまい，ポスト及びコメントに分離できなくなる。

2.2　検索クエリの分類に関する研究

検索クエリを分類する際の課題は，検索クエリ自体の情報量が少ない点である。検索クエリは単語である場合が多く，既知の単語であれば何らかのカテゴリ辞書を用いることで分類できる可能性があるが，未知の語である場合，その語が文書等でどのような文脈で出現するかを調べる必要がある。つまり，単語の情報を外部の情報によって拡張する必要がある。

検索クエリの分類に関する研究は，活用するデータに着目すると以下の 2 手法に分類できる。

- 検索エンジン内部の情報を用いる手法
- 検索エンジン外部の情報を用いる手法

2.2.1　検索エンジン内部の情報を用いる手法

辞書を用いる手法では，分類対象となる検索クエリが既知の語に限られているという問題が存在した。検索エンジン内部の情報を用いる手法では，検索エンジン事業者が保有するユーザの行動ログなどを利用し，クエリ分類モデルを自動獲得する。

ウェブ検索エンジンにおけるユーザの行動ログは様々なタスクで活用されている [Jiang 13]。Li ら [Li 08a] は検索クエリと訪問先のウェブサイトとで 2 部グラフを構築し，そのグラフに半教師あり学習手法を適用することで検索クエリを分類する手法を提案している。ある検索者があるサイトを訪問したか否かでグラフのエッジを生成しており，同じウェブサイトを訪問されるのに利用された検索クエリは，同じカテゴリに分類されるという仮説に基づいている。Baeza-Yates ら [Baeza-Yates 06] は検索クエリを訪問先の

ウェブページで拡張し，そのウェブページのテキストを素性として教師あり学習手法を適用する手法を提案している。検索クエリは高々数単語のテキストであるものの，ウェブページのテキストを追加することで，単語を素性とする教師あり学習を適用しやすくする。この手法は，検索クエリと訪問先にページが密接に関係しているという仮説に基づいている。その他にも， [Li 10, Hu 12, Cao 09, König 09] なども同様の仮説をもとにした手法である。

検索結果の上位には，検索クエリと関係するウェブページが提示されるという経験則をもとに，検索結果のスニペット（サマリー）を利用する手法が提案されている。ユーザの行動ログを利用しないものの，検索エンジンの検索インデックスを利用することから，検索エンジンの内部の情報を用いる手法に分類できる。このような手法は，[Shen 06, Diemert 09, Broder 07] で提案されており，検索クエリに検索スニペットのテキストを追加し，教師あり学習を適用する。

2.2.2　検索エンジン外部の情報を用いる手法

検索エンジン内部の情報を用いる手法は，検索エンジン事業者にしか使えないことはもちろんのこと，十分な行動ログが存在しない場合，適用が困難であるという問題点を抱える。検索エンジン外部の情報を用いる手法は，事業者が保有するデータ以外のデータを活用する手法である。

ある検索エンジンはその他の検索エンジンから見れば，外部リソースとして見立てることができる。そのため，前項で [Shen 06, Diemert 09, Broder 07] の手法は検索エンジン外部の情報を用いる手法と見なすこともできる。KDD-Cup 2005 のように検索クエリのみを与えられたタスクにおいてしばしば利用され，また，使用する検索エンジンによって性能が大きく変化することも報告されている [Shen 06]。

検索エンジンを外部リソースとして利用することは，実用面で困難があり，現実的には外部の言語リソースが利用される。Wikipedia は膨大な言語リソースとして知られており，様々な研究課題で活用されている [中山 09]。Hu ら [Hu 09] は Wikipedia 内に存在するグラフ構造に着目し，検索クエリを分類する手法を提案している。Wikipedia のデータには，見出し語（ページ）間のリンク情報，見出し語とカテゴリのリンク情報，カテゴリ間のリンク情報などが含まれており，見出し語，カテゴリをノードとするグラフを構築することができる。ある見出し語をある検索クエリと見なすことで，半教師あり学習手法を適用することができる。Khoury ら [Khoury 11] や AlemZadeh ら [AlemZadeh 12] にもよって提案されている。

Wikipedia 以外の言語リソースとして，Q&A サイトのデータを用いた例もある。Yoon ら [Yoon 10] は検索クエリを “質問” であると仮定し，Q&A のコーパスであらかじめ分類モデルを作成しておき，その分類器に検索クエリを与えることで，クエリの分類を行う手法を提案している。彼らは，さらに検索エンジン内部のクリッキングログと組み合わせる手法も提案している。その他の外部リソースの活用として，ウェブページのリンク構造を使う手法や [Dai 11, Ji 11]，Open Directory Project のリンク構造を使う手法 [Ullegaddi 11] も提案されている。

第 3 章

ウェブページのコンテンツ並びにポスト及びコメントの自動特定

3.1　本章の概要

2.1.4 項で述べたように，ウェブページ集合を利用してコンテンツを抽出する手法には，抽出ルールや訓練データの準備など，事前に人手を必要としない利点がある。しかし，これまでに提案されてきた手法は，ページの分割に経験則が用いられたり，コンテンツの判断に閾値が導入されたりするなど，ある程度の事前準備が必要であった。このような状況では，ウェブページのデザインの流行が変わったり，多種大量のページを対象としたりすると，適用が困難になる。本研究では，開発者による経験則や，閾値決定のための事前準備を要せず，全自動でコンテンツを抽出する手法を検討する。

提案するコンテンツ抽出手法は，あるウェブページのコンテンツは他のウェブページに出現しないという仮定をもとにしたコンテンツ抽出手法である。HTML のブロックレベル要素をもとに定義した，コンテンツと不要部分のかたまり（ブロック）を単位として考え，ブロックごとにコンテンツ判定を行う。

さらにコンテンツ抽出手法を拡張し，コンテンツをポスト及びコメントに分離する手法を提案する。この手法は，複数のブログ記事ページを含むあるブログサイトにおいて，ポストは全ての記事ページに出現する一方で，コメントは一部の記事ページにしか付かない傾向があることに着目した手法である。要素識別子をもとにブロックの役割を識別して，全ての記事ページにおいて同じ役割をもつコンテンツとして認められるブロックをポストとして抽出し，残りのコンテンツとして認められるブロックをコメントとして抽出する。

3.2　コンテンツ及びポスト・コメントの定義

一般的に，ウェブページはそのページがまさに伝えたい情報**コンテンツ**（主要部分）と，直接関係しない主要でない部分（**不要部分**）から成り立っている。また，ウェブページのコンテンツは，さらにポストとコメントに大別できる場合がある。“コンテンツ” 及び “不要部分” からイメージする内容はユーザごとに異なり，“ポスト” 及び “コメント” を明示的に定義した文献も筆者による公表論文以外に存在しないため，本節では，本論文におけるそれらの定義を述べる。

本論文では，記事の書き手による記事本文を**ポスト**とする。また，記事本文に付随する記事タイトル，記事日時，著者名，写真・図，写真・図の説明文もポストとみなす。さらに，記事の読み手によるコメント本文及びトラックバック[*1]本文を**コメント**とする。また，コメント本文及びトラックバック本文に付随するタイトル，投稿日時，コメント著者名（トラックバック送信元ブログ名）もコメントとみなす。なお，ポストとコメントを区別しない場合は，両方をまとめてコンテンツと呼ぶ。

不要部分の例として，広告，メニュー，関連記事リストが挙げられる。広告は，表示されているウェブページとの関連性が高ければコンテンツになりうるが，広告除去ソフトウェア[*2]が存在するなど一般的にコンテンツと認知されていない。また，メニュー及び関連記事リストは，別のページに移動するための情報であり，その表示されているウェブページに必ずしも必要とされていない。

3.3　提案手法

本論文で提案するウェブページのコンテンツ並びにポスト及びコメントの自動特定手法（以下，本章内で**提案手法**と呼ぶ）は，コンテンツ自動抽出手法（以下，**ExtractUniqueBlock 法**と呼ぶ）とポスト及びコメントの分離抽出手法（以下，**分離抽出手法**と呼ぶ）に大別できる。

ExtractUniqueBlock 法は，図 3.1 のように，ヘッダのような不要部分はページをまたいで出現するが，あるウェブページのコンテンツは他のウェブページに出現しないという仮定をもとにしたコンテンツ抽出手法である。HTML のブロックレベル要素をもとに定義した，コンテンツと不要部分のかたまり（ブロック）を単位として考え，ブロックごと

[*1] ブログ特有のリンク通知システム
[*2] Adblock Plus (Add-ons for Firefox)

にコンテンツ判定を行う。

分離抽出手法は，図 3.2 のように，複数のブログ記事ページを含むあるブログサイトにおいて，ポストは全ての記事ページに出現する一方で，コメントは一部の記事ページにしか付かない傾向があることに着目した手法である。要素識別子をもとにブロックの役割を識別して，全ての記事ページにおいて同じ役割をもつコンテンツとして認められるブロックをポストとして抽出し，残りのコンテンツとして認められるブロックをコメントとして抽出する。

ExtractUniqueBlock 法は，特定の言語に依存せず，W3C[*3]が定義する HTML タグの分類を利用してコンテンツ抽出を行う。提案手法による「ブロック識別子付与」においても，言語依存となる手掛かり語（“ポスト”，“コメント”，“リプライ”，など）を使用せず，W3C によって定義されている情報のみを利用する。そのため，抽出ルールはもちろんのこと，手掛かり語などの特殊な情報を準備することなく，抽出処理を行うことができる。

提案手法は，次の 5 つの過程から構成される。なお，ExtractUniqueBlock 法は，Step.1 から Step.3 までの処理であり，分離抽出手法は Step.4 及び Step.5 の処理である。

Step.1　[ブログページの収集]
　ポスト及びコメントの抽出を行うブログページを収集する。

Step.2　[ブロックの抽出]
　各ブログページのコンテンツの最小単位となるブロックを抽出する。

Step.3　[コンテンツの抽出]
　ページ集合の中でユニークなブロックをコンテンツとして抽出する。

Step.4　[ブロック識別子の付与]
　要素識別子をもとに各ブロックにブロック識別子を付与する。

Step.5　[ポスト及びコメントの抽出]
　ブロック識別子をもとにコンテンツをポストとコメントに分離する。

以降，上に示した各処理の詳細を説明する。

3.3.1　ブログページの収集

ExtractUniqueBlock 法はブログ記事ページに限らず，一般のウェブページを処理対象とするが，分離抽出手法はブログ記事ページに特化しているため，ブログ記事ページを処

[*3] World Wide Web で使用される技術の標準化を進める国際非営利団体。

図 3.1　ExtractUniqueBlock 法の着目点

理対象として議論を進める。本研究において，処理対象となる**ブログページ集合**は，以下の集合とする。

1. 同じブログサイトの記事ページのみで構成される
2. 2 記事ページ以上で構成される
3. コメントが付いていない記事ページが含まれる

一般的にブログ記事に対するコメント率は低い。また，フィード[*4]を用いて最新の記事ページを投稿直後に収集すれば，少なくとも 1 ページはコメントが付く前に収集できると考える。そのため，上の条件（3）は特殊事例とならない。なお，上の条件（3）が満たされていない場合は，仮説上，コメントはポストとして抽出されることが予想される。

記事ページには画像ファイルやスタイルシートが含まれている場合も多く，Cao らの手法 [Cao 08] はそれらのファイルも必要とする。一方，提案手法はそれらのファイルを必要とせず，記事ページの HTML ファイルのみで抽出処理を行うことができる。そのため，処理に必要なデータの準備が比較的容易である。

[*4] RSS，Atom などウェブサイトの更新情報を記述した XML 文書。

図 3.2　分離抽出手法の着目点

本論文では，ブログページ集合を $\boldsymbol{S}$ として次のように表現する。

$$\boldsymbol{S} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, \ldots, \boldsymbol{D}_N\}$$

$\boldsymbol{D}_i (1 \leq i \leq N, 2 \leq N)$ は各ブログページとする。

3.3.2　ブロックの抽出

コンテンツを抽出するためには，コンテンツの最小単位を決定する必要がある。本論文では，コンテンツ及び不要部分の最小単位を**ブロック**と呼ぶ。ウェブページは，マークアップ言語を定義するための言語の一種である SGML（Standard Generalized Markup Language）に基づく HTML タグで記述されているため，その構造は DOM ツリーで表現することができる。本項では，DOM ツリーからブロックを抽出する手法を述べる。

例えば，ウェブページはリスト 3.1 のような HTML コードで記述されている。DOM ツリーとは，図 3.3 のようにタグ（要素名），テキストなどを葉とする木構造である。図 3.3 は，リスト 3.1 を DOM ツリーとして表現（ただし属性情報と改行のみのテキストは省略している）した結果である。

ウェブページのレイアウト方法の特徴及び流行を考慮せず，汎用的にコンテンツの抽出を行うためには，ブロックも汎用的に抽出する必要がある。ウェブページで利用されている HTML タグは，WWW で使われる技術の標準化を進める国際団体である W3C に

リスト 3.1　HTML コードの例

```
<body>
  <div id="i1">
    <p class="c1">Text 1</p>
    <img src="#" alt="img-alt text">
  </div>
  <div id="i2">
   <img src="#" alt="img-alt text">
   <img src="#" alt="img-alt text">
  </div>
  <div id="i3">
    <a href="#" title="a-title text">Text 2</a>
    <script>Code</script>
  </div>
</body>
```

よって定められており，この定義に従うことで汎用的な抽出が可能になる。W3C の定めた HTML タグは，ウェブページ内の見出しや段落など文書の基本構造を構成するためのブロックレベル要素（H1, P, DIV, TABLE など）と，特定の語の修飾やハイパーリンクを設置するためのインライン要素（FONT, STRONG, A など）に大別することができる [Raggett 99]。

本手法では，ブロックの抽出にブロックレベル要素を用いる。ブロックレベル要素を用いてブロックを抽出する際，ブロックがコンテンツ及び不要部分の最小単位となるよう下位ノードにブロック要素が存在しないように抽出する。例えば，図 3.4 は図 3.3 の DOM ツリーの一部であり，図 3.4 をスタイルシートなどを考慮せず簡素にレンダリングすると図 3.5 のようになる。下位ノードにブロック要素が存在しないように抽出するとは，図 3.6 のように 2 つのブロックを抽出することであり，スタイルシートなどを考慮せず簡素にレンダリングすると図 3.7 のような 2 つの領域が抽出される。

図 3.3 の DOM ツリーからブロックを抽出すると，図 3.8 のように 5 つのブロックが抽出される。ブロックレベル要素をもとにブロックを抽出する際の例外として，ブラウザにレンダリングされない SCRIPT, NOSCRIPT, STYLE の 3 要素及びそれらの下位ノードはブロック内に含めないものとする。また，BODY 要素はブロックレベル要素ではな

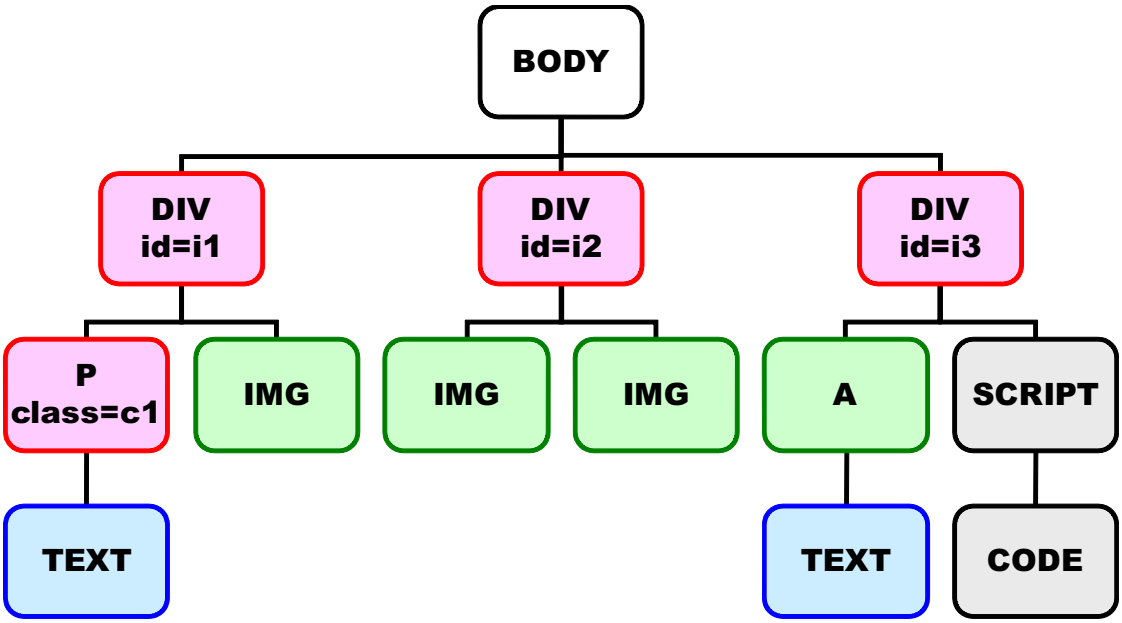

図 3.3　リスト 3.1 の HTML コードを DOM ツリーで表現した結果

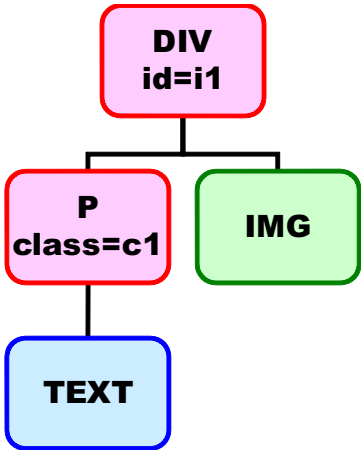

図 3.4　図 3.3 の一部

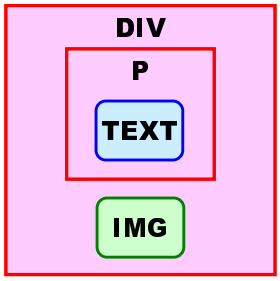

図 3.5　図 3.4 の DOM ツリーを簡素にレンダリングした結果

いが，直下にブロックレベル要素以外が存在する HTML 構造にも対応するため，例外的にブロックとして認める。なお，図 3.8 からわかるとおり，ブロックレベル要素とブロックは一対一で対応する。

本論文では，ブログページ $\boldsymbol{D}_i$ を次のように表現する。

$$\boldsymbol{D}_i = \{\boldsymbol{B}_{i1}, \boldsymbol{B}_{i2}, \boldsymbol{B}_{i3}, \ldots, \boldsymbol{B}_{iM_i}\}$$

$\boldsymbol{B}_{ij}(1 \leq j \leq M_i)$ は各ブロックとする。なお，ブロック数 M_i は，ブログページごとに変化するが有限である。

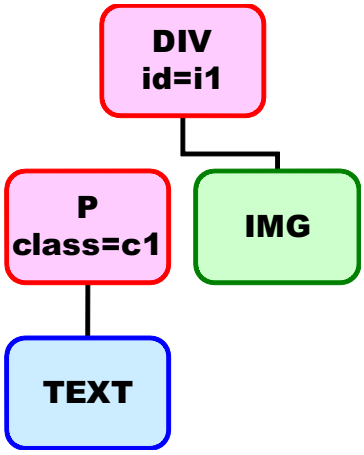

図 3.6　図 3.4 の DOM ツリーから 2 つのブロックを抽出した結果

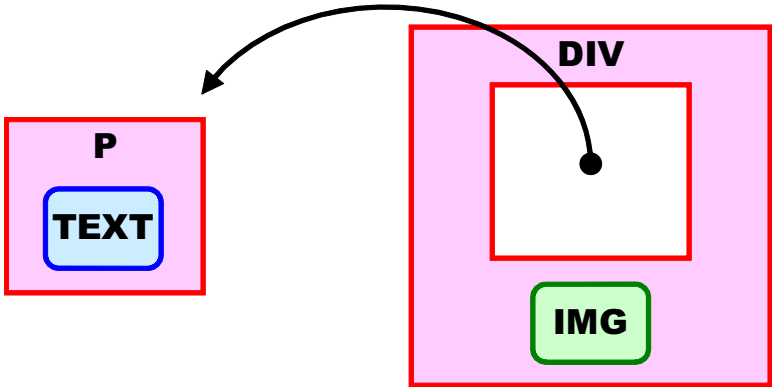

図 3.7　図 3.6 の抽出結果を簡素にレンダリングした結果

3.3.3　コンテンツの抽出

ブロックの一致を判断し，ページ集合内で 1 度だけ出現するブロックをコンテンツとして抽出する。本論文では，このブロックを**コンテンツブロック**と呼ぶ。

コンテンツの抽出処理は，ブロックを比較することによって行う。その処理はブロックのベクトル化と，ベクトルの比較の 2 過程にわけられる。以降の小々節で，それらの 2 過程を説明する。

ブロックのベクトル化

ブロック間の微小な違いを許容して一致を判断するために，ブロックをベクトルで表現する。本論文では，そのベクトルを**ブロックベクトル**と呼ぶ。本手法では，ブロックベクトルのベクトル素性としてブロックに含まれる要素名，テキスト， title, alt, src 属性の

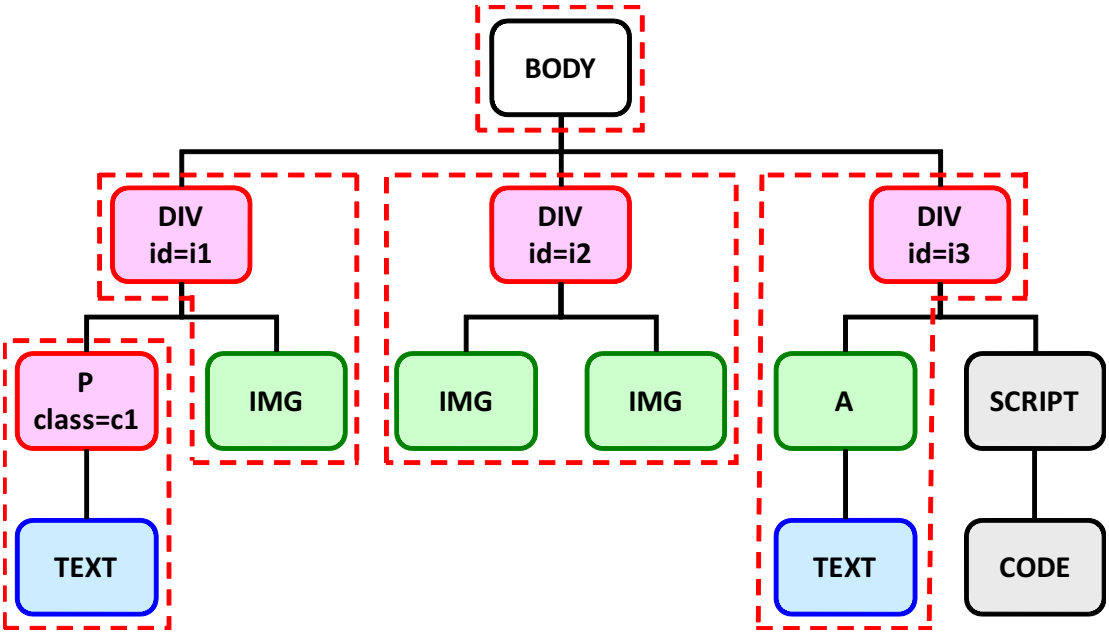

図 3.8　DOM ツリーから 5 つのブロックを抽出した例（破線部分）

属性値を用いる。素性の値はそれぞれの出現回数とする。なお，テキストの出現回数をカウントする際，テキストを改行ごとに個々のベクトル素性と見なし，英字の場合は全てを小文字に正規化して用い，空白のみのテキストは除外している。

ブロックに含まれる要素名の数は，ブロックのレイアウト情報を表現している。また，ブロックに含まれるテキストの数はブロックの内容を， title, alt, src 属性の属性値の数は画像（IMG）が出現するブロックの内容を表現している。

本論文では，ブロック $\boldsymbol{B}_{ij}$ を次のようにブロックベクトルで表現する。

$$\boldsymbol{B}_{ij} = (b_{ij1} \quad b_{ij2} \quad b_{ij3} \quad \ldots \quad b_{ijL})$$

$b_{ijk}(1 \leq k \leq L)$ はベクトルの各素性の値とする。抽出を行うブログページ集合に含まれるブログページの数は N であり，テキストはその内容ごとに次元が異なるため L は非常に大きな値を取るが，ブログページ集合 $\boldsymbol{S}$ を決定した段階で固定化される。

比較抽出

ブロックの一致を判断し，他のブログページには出現しないブロック，すなわちブログページ集合の中で 1 度だけ出現するブロックにコンテンツラベルを付与し，コンテンツブロックを抽出する。

ブロックの一致判断は，前で述べたブロックベクトルを用いて，ブロックベクトル同士のコサイン類似度 [北 02a] を計算することにより行う。ブロックベクトル $\boldsymbol{B}_{ij}$ と $\boldsymbol{B}_{kl}$ の

類似度 $Sim(\boldsymbol{B}_{ij}, \boldsymbol{B}_{kl})$ は，次のように計算できる。

$$Sim(\boldsymbol{B}_{ij}, \boldsymbol{B}_{kl}) = \frac{\boldsymbol{B}_{ij} \cdot \boldsymbol{B}_{kl}}{\|\boldsymbol{B}_{ij}\|\|\boldsymbol{B}_{kl}\|} \quad (i \neq k)$$

本手法では，ブロック間の類似度 $Sim(\boldsymbol{B}_{ij}, \boldsymbol{B}_{kl})$ が 0.9 を越えたとき，それらのブロックは一致したと判断する。一致判断に類似度を導入することにより，レンダリングにほとんど影響を与えない微小な違いを許容できる。ブロック間の比較に用いる閾値に関しては，3.4.8 項で検討する。

本処理は，$Sim(\boldsymbol{B}_{ij}, \boldsymbol{B}_{kl})$ の計算結果が存在すれば $Sim(\boldsymbol{B}_{kl}, \boldsymbol{B}_{ij})$ の計算を省略するが，それら以外の高速化処理を行っておらず，全てのブロックの組み合わせを順に計算している。そのため，ブログページ集合に含まれるページ数が大きくなると，処理時間が長くなる。処理時間に関する詳細は，3.4.7 項で述べる。

本章で提案する手法のうち，ExtractUniqueBlock 法に関する処理は本項で完了する。以降の項では，分離抽出手法に必要な処理について述べる。

3.3.4　ブロック識別子の付与

分離抽出手法は，複数のブログ記事ページを含むあるブログサイトにおいて，ポストは全ての記事ページに出現する一方で，コメントは一部の記事ページにしか付かない傾向があることに着目した手法である。全ての記事ページにおいて同じ役割をもつコンテンツとして認められるブロックをポストとして抽出し，残りのコンテンツとして認められるブロックをコメントとして抽出するために，各ブロックの役割を識別する**ブロック識別子**（提案手法のために定義する識別子）を付与する。

ブログページ集合内におけるブロックの役割を識別するブロック識別子を付与するために，あらかじめページ内の各ブロックレベル要素に付与された要素識別子（Element Identifiers）を利用する。要素識別子は id 属性及び class 属性のことであり [Bos 09]，スタイルシートのセレクタなど，要素を特定するために使われる識別子であることから，ブロックレベル要素をもとにしたブロックの役割を識別するのに準用できると考えられる。しかし，ブロック識別子に要素識別子をそのまま用いると問題が発生する。

ブロック識別子に要素識別子をそのまま用いることの最大の問題は，全てのブロックレベル要素に要素識別子が付与されているとは限らない点である。この問題を解決するために，近隣のブロックは同じ役割をもつと仮定し，要素識別子を DOM ツリー上の兄ノードから弟ノードに伝播させる。親ノードよりも兄ノードを優先することにより，ブラウザでのレンダリングイメージにおいて，近隣から伝播される。例えば，図 3.9 に示すツリーに

おいて親ノードを優先して伝搬させると（A，B が要素識別子である），下位ノードは左から A，B，A となる。この場合，レンダリングイメージで見てもわかるとおり，B のノードが孤立してしまう。対して兄ノードを優先して伝搬させると，左から A，B，B となり，孤立を防ぐことができる。

あらかじめ付与された要素識別子の中には，ブログページ集合内で一般化されたブロック識別子に適さない要素識別子が 2 種類含まれる。一方は，各ページにおいて何度も出現する要素識別子である。このような要素識別子を伝播させると，隣接していないにもかかわらず同じブロック識別子が付与されてしまう。他方は，ブログページ集合内において一部のページにしか出現しない要素識別子である[*5]。このような要素識別子は，個々のページ特有の情報をもつ可能性が高く，一般化するのに適さない。

以上を踏まえ，ブロックに対し，ブログページ集合内で一般化されたブロック識別子の付与を行う。まず，ブロック識別子として伝播させる要素識別子として，以下の条件を満たす要素識別子を選択する。

a.　1 ページ中に 1 度のみ出現する
b.　ブログページ集合全てのページに出現する

そして，各ブロックと対応するブロックレベル要素を DOM ツリー上で前順（preorder）に走査し，次の優先順位に従ってブロック識別子を付与する。

1. 自身の要素識別子を付与する
2. 隣接する兄ノードのブロック識別子を付与する
3. 親ノードのブロック識別子を付与する
4. 標準値（_default_）を付与する

図 3.10 に「自身の要素識別子を付与」「隣接する兄ノードのブロック識別子を付与」の 2 処理を示し，図 3.11 に「親ノードのブロック識別子を付与」「標準値を付与」の 2 処理を示す。なお，ツリーにおいて左側が親ノード，上側が兄ノードであるものとする。図 3.10 の左ツリーの処理ノードに要素識別子が存在するため，その要素識別子 id=post をブロック識別子として付与する。右ツリーの処理ノードには要素識別子が存在しないが，兄ノードが存在するため，そのノードのブロック識別子 id=post を付与する。さらに，図 3.11 左ツリーの処理ノードに要素識別子が存在せず，さらに兄ノードも存在しないため，親

[*5] article_12345 のような記事固有の数値を含む要素識別子が，記事本文付近にしばしば出現する。

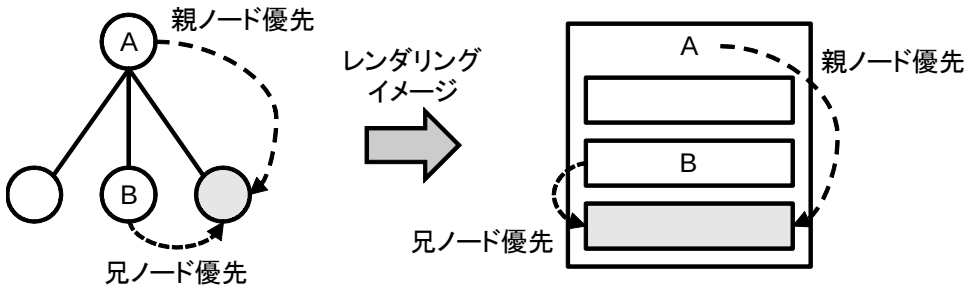

図 3.9　親ノード優先と兄ノード優先との違い

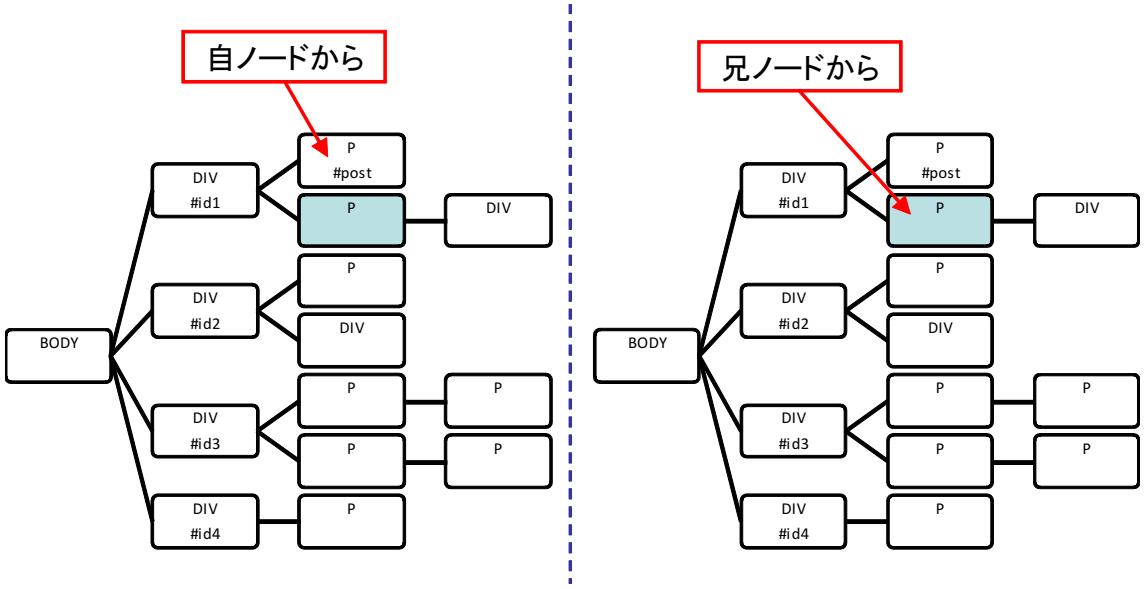

図 3.10　ブロック識別子の付与処理の例 1

ノードのブロック識別子 id=post を付与する。右ツリーの処理ノードには，要素識別子，兄ノード，親ノードのいずれも存在しないため，標準値を付与する。

図 3.12 は，DOM ツリー（要素識別子は要素内に記載）からブロック抽出を行い，各ブロック（破線矩形枠部分）にブロック識別子（実線矩形枠部分）を付与した例である。ブロック識別子上部の括弧付き数字は，上で述べたブロック識別子付与規則を表す。例えば，ブログページ 1 の「class=c1」は双方のブログページに出現しているが，ブログページ 2 において 2 度出現しているため，伝播させる要素識別子の条件 a を満たさず，このブロックに対しては規則（2）が適用される。

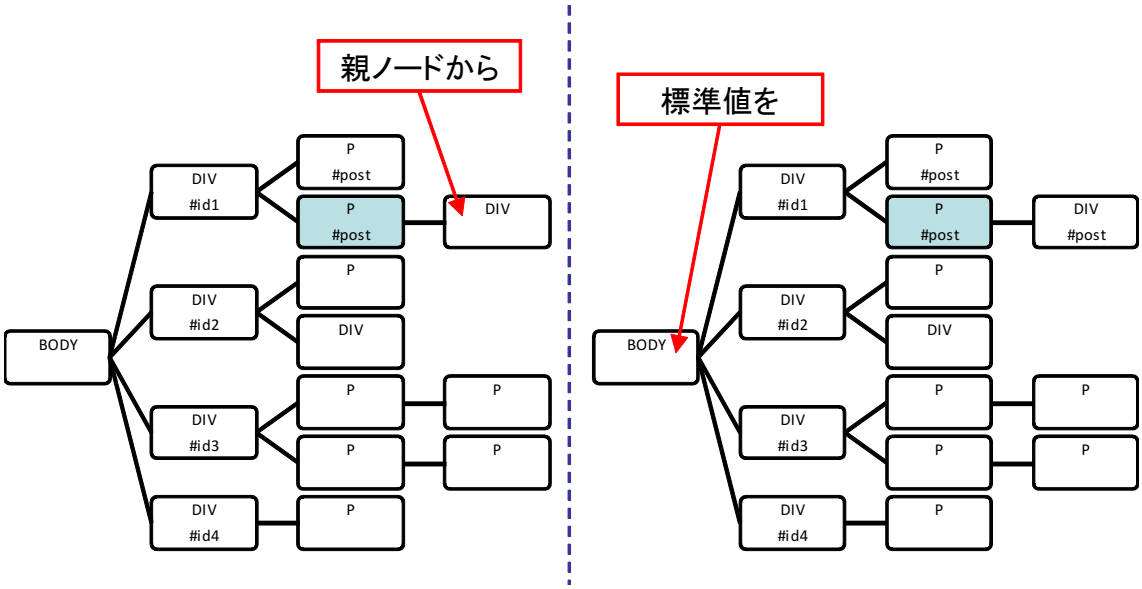

図 3.11　ブロック識別子の付与処理の例 2

3.3.5　ポストとコメントの分離抽出

3.2 節で定義したように，コンテンツはポストとコメントから成るため，ポストとコメントの分離抽出にあたり，コンテンツブロック以外のブロック（不要部分）を除外する。例えば，図 3.13 に示すレンダリングイメージにおいて，矩形をブロック，着色された矩形をコンテンツブロックとすると，着色されていないブロックを除外する。なお，A から F の英字は，各ブロックのブロック識別子を表す。

筆者らは，ポストは全ての記事ページに出現する一方で，コメントは一部の記事ページにしか付かない傾向があることに着目した。この傾向に従って，コンテンツブロックをポストとコメントに分類する。具体的には，全てのページに出現するブロック識別子が付与されたコンテンツブロックをポストとし，残りのコンテンツブロックをコメントとして抽出する。図 3.13 では，全てのページに出現するブロック識別子 B が付与されたコンテンツブロックがポストとして抽出され，ブロック識別子 C が付与された残りのコンテンツブロックがコメントとして抽出される。

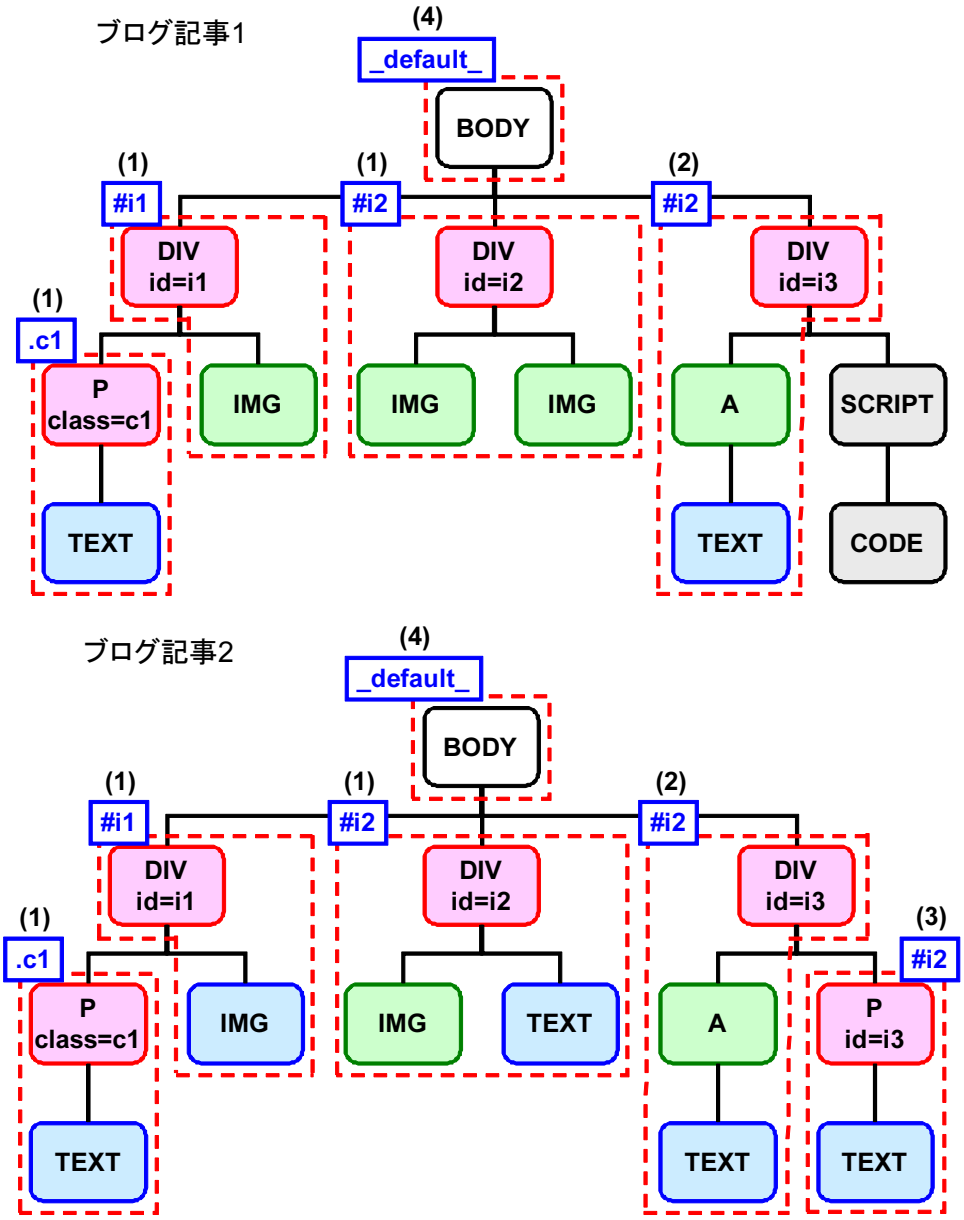

図 3.12　各ブロックにブロック識別子を付与した例

3.4　評価実験

本節では提案手法の有効性を示すための評価実験について述べる。まず，評価指標，使用データ，比較手法について述べる。そして，比較手法を交えた実験結果について述べる。また，提案手法を改善する方法を検討した後，考察を述べる。

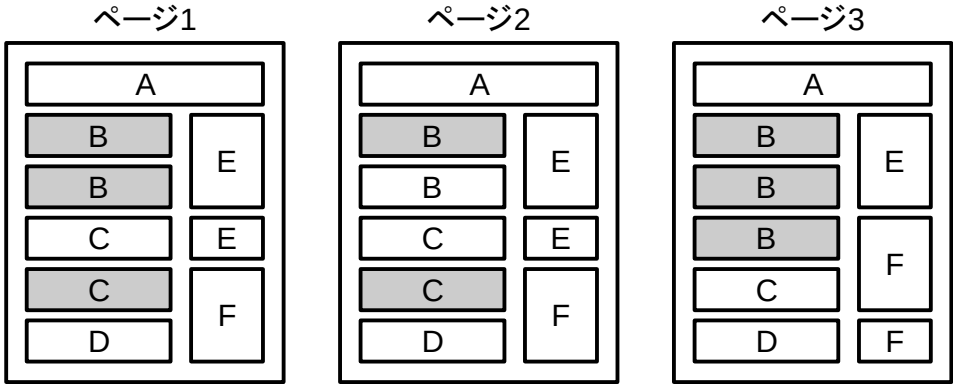

図 3.13　ポストとコメントの分離抽出の例

3.4.1　評価指標

本実験の評価指標は，3.2 節の定義に従って人手により作成したポストとコメントの正解データに対する正確度（Accuracy），適合率（Precision），再現率（Recall），F 値（F-measure）である [北 02b]。

正確度は，各ブロックがどれだけ正確に判別できたかの指標であり，以下の式で計算される。

$$Accuracy = \frac{\text{正確に判別できたブロックの数}}{\text{正解データに含まれるブロックの数}}$$

なお，コンテンツ抽出性能の正確度は各ブロックをコンテンツまたは不要部分に分類した場合の正確度を表し，ポスト・コメント抽出性能の正確度は各ブロックをポスト，コメント，不要部分に分類した場合の正確度を表す。

適合率，再現率，F 値は，コンテンツ，ポスト，コメントがどれだけ抽出できたかの指標であり，例えば，ポストに対するこれらの指標は以下の式で計算される。

$$Precision = \frac{\text{“ポスト” と正解した数}}{\text{“ポスト” と判定した数}}$$

$$Recall = \frac{\text{“ポスト” と正解した数}}{\text{正解データに含まれる “ポスト” の数}}$$

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

コンテンツまたはコメントの適合率，再現率，F 値も同様に計算する。

3.4.2　正解データの作成・内容

実験に使用する正解データ（テストデータ）は，筆者の開発したアノテーションツール（図 3.14）を用いて作成した。このアノテーションツールは，コンテンツの最小単位となるブロックを自動で認識し，ブラウザ上にてマウスの操作のみで容易にポスト及びコメントをアノテートすることができる。HTML コードや複雑な DOM ツリーを見ながらアノテートを行う必要がないため，短時間で，より正確に正解データの作成を行うことができる。アノテートの基準は，3.2 節の定義に従う。

アノテートの対象としたブログ記事ページは，livedoor Reader の登録数ランキング上位からブログ形式の 9 サイト，100SHIKI[*6]，Engadget Japanese[*7]，ネタフル[*8]，404 Blog Not Found[*9]，IDEA*IDEA[*10]，TechCrunch Japan[*11]，Life is beautiful[*12]，My Life Between Silicon Valley and Japan[*13]，Going My Way[*14] から 2009 年 4 月 11 日に収集した計 206 ページである（表 3.1）。これらの 9 サイトは，提案手法の汎用性を示せるよう，ブログホスティングサービスが重複しないように選択されている。選択した 9 サイトのうち 6 サイトはブログホスティングサービスを利用せず，独自に開設されたブログサイトであった。なお，様々な観点で評価できるよう，全てのページにコメントが付いているブログサイト（My Life Between Silicon Valley and Japan）及びコメントが 1 件も付いていないブログサイト（Going My Way）がテストデータに含まれる。

表 3.1 に示すとおり，ポスト及びコメントの候補となるブロックは 35,216 ブロック存在し，正解データを人手によって作成したところ，ポストと認められるブロックは 2,932 ブロック，コメントと認められるブロックは 973 ブロックであった。また，38% のページにコメントが付いていた。

[*6] http://www.100shiki.com/
[*7] http://japanese.engadget.com/
[*8] http://netafull.net/
[*9] http://blog.livedoor.jp/dankogai/
[*10] http://www.ideaxidea.com/
[*11] http://jp.techcrunch.com/
[*12] http://satoshi.blogs.com/
[*13] http://d.hatena.ne.jp/umedamochio/
[*14] http://kengo.preston-net.com/

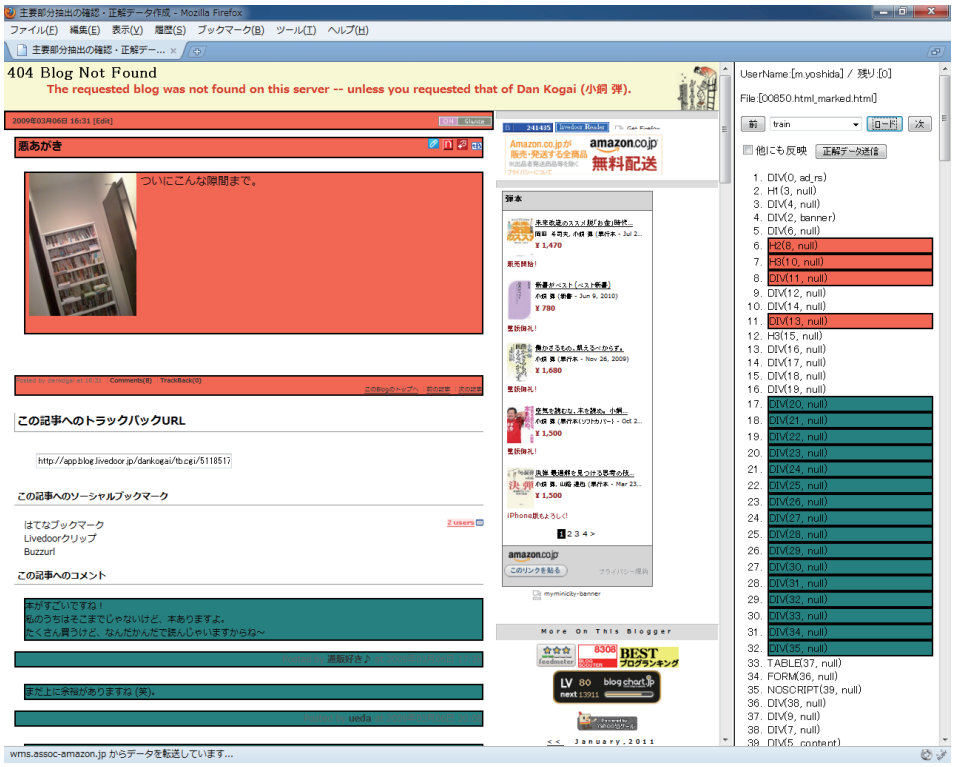

図 3.14　正解データ作成用のアノテーションツール

3.4.3　比較手法

提案手法の有効性を示すため，次の 3 つの手法と比較する。2.1.5 項で述べたように，提案手法と従来手法とではコンテンツの最小単位が異なるため，比較手法によって抽出されるコンテンツブロックは，コンテンツによって包含される全てのブロックのうち，テキストを含むブロックとする[*15]。

比較 1（Cao ら [Cao 08]）

Cao ら [Cao 08] によって提案された，ポストとコメントを同時に自動抽出する手法である。

比較 2（Pappas ら [Pappas 12]）

Pappas ら [Pappas 12] の手法はコンテンツ抽出手法であるが，彼らが公開している実装[*16]にはポストとコメントを同時に抽出する機能を有している。この実装で

[*15] テキストを含まないブロックもコンテンツブロックと見なした場合，再現率に変化が見られず，適合率が著しく低下した。

[*16] `https://github.com/nik0spapp/webpage_segmentation`

表 3.1　データセットの詳細

ブログ名	記事数	ブロック	ポスト	コメント
100SHIKI	10	725	153	12
Engadget Japanese	40	6,163	200	36
ネタフル	30	2,494	451	4
404 Blog Not Found	50	19,264	1,008	750
IDEA*IDEA	9	569	121	6
TechCrunch Japan	20	1,951	182	8
Life is beautiful	12	1,525	130	111
My Life Between ...	5	575	166	46
Going My Way	30	1,950	521	0
合計	206	35,216	2,932	973

は，抽出したコンテンツの中に “comment” や “reply” など，コメント出現特有の文字列が現れた場合，その周辺部分をコメントと見なし，それら以外をポストと見なしている。なお，この実装には特定の言語に依存する処理が含まれていたため，日本語のページにも適用できるように改良した[*17]。

比較 3（Weninger ら [Weninger 10] と Song ら [Song 10]）

コンテンツ抽出手法である Weninger ら [Weninger 10] の手法[*18]とコメント抽出手法である Song ら [Song 10] の手法を組み合わせる。Weninger らの手法による抽出分はポスト，Song らの手法による抽出分はコメント，双方の抽出が重複する場合はコメントと見なす[*19]。

3.4.4　分離抽出の実験結果

まず，正確度の比較結果を表 3.2 に示す。表中の「提案」は提案手法を表し，「比較 1〜3」は前項で説明した各比較手法を表す。また，太字は最も性能が高かった手法を表す。

[*17] “comment”，“reply”，“.”，など特定の文字列を探索する処理をそれらの日本語訳（“コメント”，“リプライ”，“返信”，“。”，など）も探索するようにした。

[*18] 以下のウェブページで公開されている実装を利用した。
`http://www.cse.nd.edu/~tweninge/cetr/`

[*19] 予備実験を行ったところ，Song らの手法が Weninger らの手法よりも高い適合率を示したため，抽出結果の一部が不要部分であるか否かの判定は必要ないと判断した。

それぞれのブログサイトにおいて，コンテンツ抽出性能，ポスト・コメント抽出性能のそれぞれに対して符号検定を行ったところ，提案手法と各比較手法との間に危険率 1% で有意差を確認した。「My Life Between Silicon Valley and Japan」を除き，提案手法の方が有意に正確度が高いことが確認できた。3.4.2 項で述べたように，「My Life Between Silicon Valley and Japan」には全てのページにコメントが付いており，3.3.1 項で述べた提案手法の前提を満たしていない。提案手法の前提を満たすブログサイトに限れば，いずれのサイトにおいても提案手法が最高性能を示している。

本指標は，不要部分の特定も含めた正確度を算出するため，ブロック数に対しコンテンツブロック数が少ない場合，全てのブロックを不要部分とすることで高い性能を達成できる場合がある。例えば，比較手法 2 は，多くのブロックを不要部分と判定することで，比較的高い正確度を達成していた。

抽出したポスト及びコメントの活用を想定すれば，それらに対する適合率，再現率，F 値の評価も必要である。適合率と再現率から算出した F 値による比較結果を表 3.3 に示す。表中の太字は最も性能が高かった手法を表す。「Going My Way」は正解データにコメントが含まれないため，コメント抽出性能の F 値が計算できず「-」とした。「My Life Between Silicon Valley and Japan」を除き，提案手法の方が抽出性能が高いことが確認できた。先の結果同様，提案手法の前提を満たすブログサイトに限れば，いずれのサイトにおいても提案手法が最高性能を示している。提案手法のコメント抽出性能に関しては，正解ブロック数（コメント）が少ないため実験結果が安定していないが，「404 Blog Not Found」「Life is beautiful」からわかるとおり，十分なコメントがあれば高い性能で安定している。

表 3.2 正確度（Accuracy）による分離抽出性能の比較

	コンテンツ抽出性能				ポスト・コメント抽出性能			
ブログ名	提案	比較 1	比較 2	比較 3	提案	比較 1	比較 2	比較 3
100SHIKI	**94.2**	85.9	78.6	84.6	**92.8**	85.9	77.5	82.3
Engadget Japanese	**98.5**	86.5	93.0	77.0	**98.5**	85.3	92.4	76.5
ネタフル	**94.7**	66.4	81.2	86.0	**90.6**	66.1	81.1	85.7
404 Blog Not Found	**98.8**	96.6	91.6	81.5	**98.5**	87.2	91.2	75.2
IDEA*IDEA	**94.7**	77.9	73.8	85.4	**93.7**	77.9	73.8	85.1
TechCrunch Japan	**97.0**	81.3	90.1	94.9	**96.8**	68.3	90.1	94.6
Life is beautiful	**97.3**	89.0	87.1	91.1	**96.0**	75.7	81.1	88.5
My Life Between ...	**95.0**	79.7	61.7	83.5	80.3	70.6	58.3	**83.1**
Going My Way	**94.4**	85.0	68.5	80.3	**89.7**	45.6	68.5	80.3
マイクロ平均	**97.8**	90.1	88.5	82.3	**96.7**	81.0	87.8	78.6
マクロ平均	**96.1**	83.1	80.6	84.9	**93.0**	73.6	79.3	83.5

表 3.3 F 値による分離抽出性能の比較

	コンテンツ抽出性能				ポスト抽出性能				コメント抽出性能			
ブログ名	提案	比較 1	比較 2	比較 3	提案	比較 1	比較 2	比較 3	提案	比較 1	比較 2	比較 3
100SHIKI	**87.3**	75.1	28.6	72.3	**87.8**	86.6	26.3	70.4	**48.3**	29.3	0.0	0.0
Engadget Japanese	**79.9**	34.8	12.2	23.7	**84.5**	58.6	5.1	21.8	**62.6**	0.0	6.9	35.3
ネタフル	**85.8**	52.0	9.3	68.4	**83.0**	63.7	8.6	67.9	**7.8**	0.0	0.0	0.0
404 Blog Not Found	**93.3**	84.0	28.2	43.2	**90.0**	20.7	36.1	27.4	**92.8**	47.2	0.0	0.0
IDEA*IDEA	**88.7**	65.8	0.0	73.0	**90.0**	79.3	0.0	73.8	**35.3**	15.4	0.0	0.0
TechCrunch Japan	**83.9**	49.6	26.1	73.7	**84.1**	31.4	26.9	73.7	**60.0**	3.2	0.0	0.0
Life is beautiful	**92.1**	72.7	40.6	76.1	**82.5**	58.4	20.3	60.1	**95.2**	0.0	0.0	86.2
My Life Between ...	**93.5**	66.5	8.3	73.8	**83.2**	59.6	0.0	69.0	0.0	26.2	0.0	**87.8**
Going My Way	**89.3**	76.0	41.0	47.3	**84.1**	24.6	41.0	47.3	-	-	-	-
マイクロ平均	**90.1**	67.6	26.4	47.9	**86.2**	49.0	26.7	38.5	**81.8**	26.9	0.4	23.3
マクロ平均	**88.2**	64.1	21.6	61.3	**85.5**	53.7	18.3	56.8	**50.3**	15.2	0.9	26.2

表 3.4　提案手法によるポストとコメントの分離抽出性能詳細

	ポスト抽出性能			コメント抽出性能		
ブログ名	抽出数	適合率	再現率	抽出数	適合率	再現率
100SHIKI	150	88.7	86.9	17	41.2	58.3
Engadget Japanese	174	90.8	79.0	63	49.2	86.1
ネタフル	371	91.9	75.6	99	4.0	100.0
404 Blog Not Found	832	99.5	82.1	715	95.1	90.7
IDEA*IDEA	128	87.5	92.6	11	27.3	50.0
TechCrunch Japan	158	90.5	78.6	12	50.0	75.0
Life is beautiful	156	75.6	90.8	120	91.7	99.1
My Life Between ...	233	71.2	100.0	0	-	0.0
Going My Way	456	90.1	78.9	46	-	-
合計（マイクロ平均）	2,658	90.7	82.2	1,083	77.7	86.4

提案手法による分離抽出性能の詳細を表 3.4 に示す。「My Life Between Silicon Valley and Japan」はコメントが自動抽出されなかったため，コメント抽出性能の適合率が計算できず「-」とした。コメントが自動抽出されなかった原因は，3.3.1 項で述べた提案手法の前提による。「My Life Between Silicon Valley and Japan」には全てのページにコメントが付いており，自動的に分離抽出した結果，全てのコメントがポストとして抽出された。また，「Going My Way」には 1 件のコメントも付いておらず，仮説通りであれば全てのコンテンツがポストとして抽出されるが，コンテンツ抽出の失敗により，コメントとして抽出されるブロックが存在した。

コンテンツ自動抽出に失敗したポスト及びコメントは，その後の処理である分離処理で抽出不可能となるため，表 3.4「ポスト抽出性能」「コメント抽出性能」の再現率には上限値が存在する。コンテンツ自動抽出の結果から正解データに含まれないブロック（不要部分）を除外し，残りのブロックを完璧に分離抽出した場合の再現率は，それぞれ 87.1% と 91.6% であった。提案手法による性能（再現率）は，それぞれ 82.2% と 86.4% である。このことから，提案する分離処理は，コンテンツ自動抽出が不完全であっても，その範囲内でかなり正確に分離できていることがわかる。

3.4.5 分離処理のみの性能評価実験

前項で評価した分離抽出性能は ExtractUniqueBlock 法に依存しているため，本項では，コンテンツを過不足なく抽出できたと仮定し，分離処理のみの評価を行う。比較手法 1 及び 3 に関しては，それらの手法によって抽出されたコメントの集合を C，正解データを T としたとき，ポスト及びコメントをそれぞれ $T-(T\cap C)$ 及び $T\cap C$ であると見なして評価する[*20]。比較手法 2 に関しては，表 3.3 が示すようにコメント抽出性能が低かったことを踏まえ，HTML ファイルから “コメント” や “comment” などの文字列を探索して分割し，後方に含まれるブロックの集合を C と見なして比較手法 1 及び 3 と同様に評価する[*21]。

分離処理のみの比較結果（F 値）を表 3.5 に示す。提案手法のための前提を満たさない「My Life Between Silicon Valley and Japan」及びコメントがポストの 1% 未満しか存在しない「ネタフル」を除き，提案手法が最高性能を示している。コンテンツ自動抽出が理想的に行われた場合，提案手法は非常に高い性能を達成できることが明らかになった。

[*20] ポストの集合を P とし，$T\cap P$ 及び $T-(T\cap P)$ と見なした場合も試したが，本文で述べた条件の方が F 値が高かった。

[*21] “コメント” や “comment” などの探索文字列が複数回出現する場合，最初の出現位置で分割する。探索文字列は 3.4.3 項と同じものを利用した。

表 3.5 分離抽出処理のみの性能比較（F 値）

	ポスト抽出性能				コメント抽出性能			
ブログ名	提案	比較 1	比較 2	比較 3	提案	比較 1	比較 2	比較 3
100SHIKI	100.0	100.0	100.0	96.2	100.0	100.0	100.0	0.0
Engadget Japanese	100.0	91.5	0.0	93.7	100.0	0.0	26.5	40.0
ネタフル	99.6	99.6	100.0	99.6	0.0	0.0	100.0	0.0
404 Blog Not Found	99.3	24.0	0.0	72.9	99.0	59.3	59.8	0.0
IDEA*IDEA	100.0	100.0	100.0	97.6	100.0	100.0	100.0	0.0
TechCrunch Japan	100.0	46.2	0.0	97.8	100.0	9.9	8.1	0.0
Life is beautiful	100.0	69.7	97.6	90.6	100.0	0.0	97.4	86.2
My Life Between ...	87.8	87.1	0.0	97.1	0.0	32.4	35.7	87.8
Going My Way	100.0	41.6	0.0	100.0	-	-	-	-

表 3.6　コンテンツ再抽出の有無による平均抽出性能の比較（マイクロ平均）

		適合率	再現率	F 値	正確度
コンテンツ再抽出あり	コンテンツ	90.6	92.2	91.4	98.1
	ポスト	90.4	85.1	87.7	96.9
	コメント	74.6	93.2	82.9	
コンテンツ再抽出なし（ExtractUniqueBlock 法）	コンテンツ	92.1	88.2	90.1	97.8
	ポスト	90.7	82.2	86.2	96.7
	コメント	77.7	86.4	81.8	

3.4.6　コンテンツ抽出の改善処理

ExtractUniqueBlock 法は，各々のブロックの位置を加味せずコンテンツを判定するため，連続するコンテンツブロックの一部を取りこぼす場合がある。図 3.15 の破線部分は，コンテンツとして抽出されて欲しいのに，抽出できなかった例である。そこで，コンテンツ抽出（3.3.3 項）で不要部分と判断されたブロックのうち，コンテンツブロックと同一のブロック識別子と要素名をもつブロックを新たにコンテンツブロックとして再抽出することにより，コンテンツ自動抽出の性能向上を試みた。

表 3.6 は平均抽出性能（マイクロ平均）を比較した結果である。この実験結果より，コンテンツ再抽出ありは再抽出なし（ExtractUniqueBlock 法）に比べて全体的に性能が向上していることがわかる。再抽出ありによるコンテンツ抽出性能は，再抽出なしに比べ，適合率は少々低下したが，再現率は大幅に向上し，F 値が向上している。正確度に関しても，コンテンツ抽出性能とポスト・コメント抽出性能のいずれも向上した[*22]。図 3.16 はコンテンツ抽出の改善処理を含め，分離抽出手法による抽出を行ったブログ記事ページ[*23]の例である（破線着色部分がポスト，実線着色部分がコメントを示す）。この例は，図 3.15 と同じページの例であり，取りこぼしていたコンテンツ部分が適切に再抽出されていることが確認できる。

[*22] 符号検定を行ったところ，危険率 1% において 5 サイトで有意差を確認した。
[*23] http://blog.livedoor.jp/dankogai/archives/51185176.html

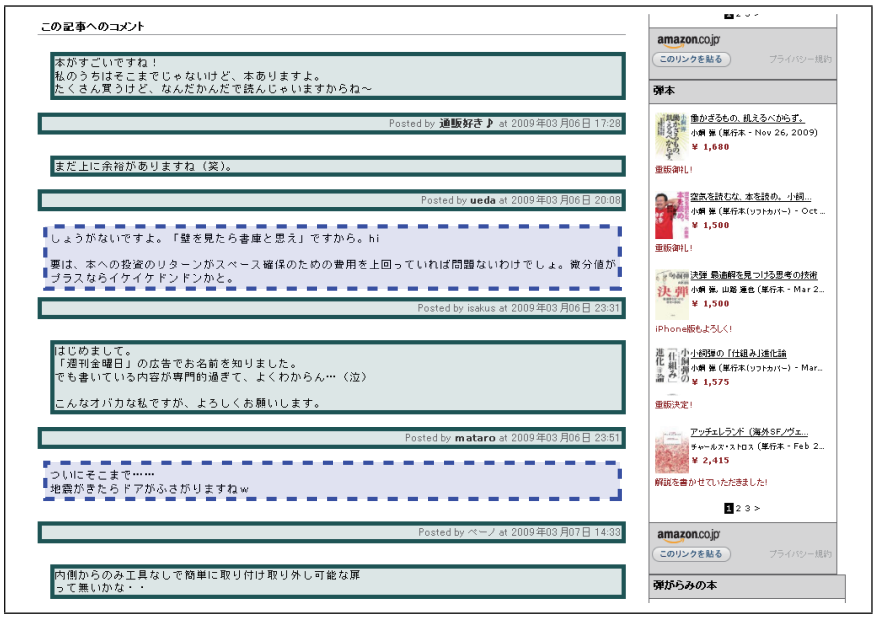

図 3.15　連続するコンテンツ部分の抽出失敗例（破線部分）

3.4.7　処理時間改善の検討

3.3.3 項で述べたように，コンテンツを抽出する際のブロック間の類似度は，全てのブロックの組み合わせを順に計算している。そのため，ブログページ集合に含まれるページ数が大きくなると，処理時間が長くなる。本項では，ブログページ集合に含まれるページ数と処理時間の関係を評価した。

処理時間の評価は，テストデータに含まれる 404 Blog Not Found のブログページを用いて行った。5 ページずつ増やした際の処理時間（秒）及び ExtractUniqueBlock 法におけるコンテンツ抽出性能のグラフを図 3.17 に示す。グラフからわかるとおり，ページ数が増えると処理時間が長くなり，1 ページあたりの処理時間も長くなる。しかし，コンテンツ抽出性能（F 値）は 95% 前後で安定している。このことから，大きなページ集合を分割して処理することで，コンテンツ抽出性能に大きく影響を与えずに処理時間を削減できると考えられる。

3.4.8　ブロック間の比較に用いる閾値の検討

3.3.3 項で述べたように，ブロック間の一致を判定するためにコサイン類似度を用いている。そして，類似度が 0.9 を超えたとき，それらのブロックは同じであると認めている。本項では，使用する閾値について評価した。

図 3.16　実験結果のブログ記事ページ例（分離抽出手法）

閾値の評価は，テストデータに含まれる 404 Blog Not Found のブログページを用いて行った。閾値と ExtractUniqueBlock 法におけるコンテンツ抽出性能の関係を示したグラフが図 3.18 である。このグラフより，閾値が 0.8 を以上になると抽出性能（F 値）が高い値で安定しており，本手法が閾値にほとんど依存していないことがわかる。また，ベクトルの完全一致（閾値 1.0）においても高い性能を示しており，類似度のかわりにハッシュ法を利用することで，処理時間も改善する可能性がある。

3.4.9　考察

ExtractUniqueBlock 法は，あるウェブページのコンテンツは他のウェブページに出現しないという仮定をもとにしたコンテンツ抽出手法である。そのため，重複ページが含まれる場合，コンテンツが抽出されなくなると考えられる。Chowdhury ら [Chowdhury 02] や Manku ら [Manku 07] の手法を活用するなど，ウェブページを収集する時点でページの重複を検知する必要がある。

3.2 節では投稿日時もコンテンツであると定義したが，投稿日時の表現方法は限られ

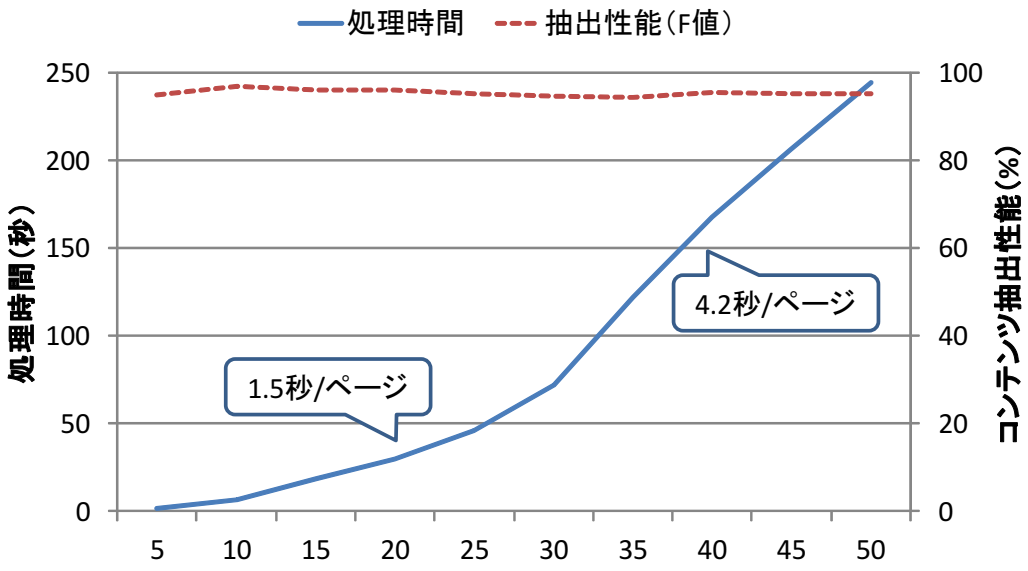

図 3.17　ページ数と処理時間及びコンテンツ抽出性能との関係

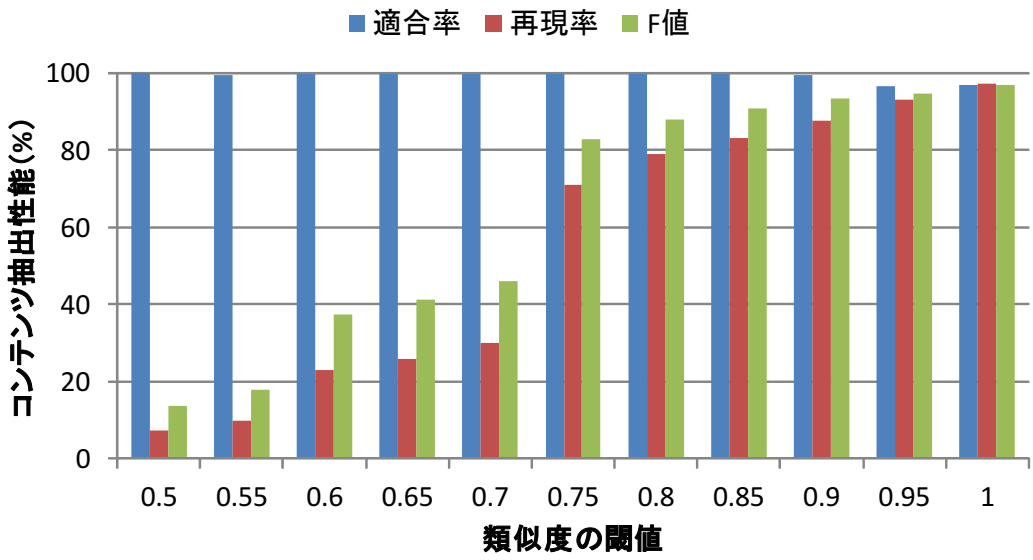

図 3.18　コサイン類似度の閾値とコンテンツ抽出性能（%）との関係

ている。使用したテストデータには含まれていなかったが，「2014 年 1 月 6 日」のように年月日しか公開しないウェブサイトがある場合，そのウェブサイト内で同じ日に 2 記事以上公開されると，投稿日時の抽出が困難になると考えられる。これ解決するためには，あらかじめ日付フォーマットを学習したモデルが必要になる。ただし，代表的な日付フォーマットが限られているため，モデル作成は小さな労力で作成できる可能性がある [南野 04]。

分離抽出手法の性能は，ExtractUniqueBlock 法の性能に依存している。ポストの適合率に比べてコメントの適合率が低くなる傾向があるのは，ポストが完全に抽出されないこと，不要部分を誤って抽出することが影響している。ExtractUniqueBlock 法の性能を改善できれば，分離抽出手法の性能も向上する可能性が高い。

ブログページ集合に構造が乱れた HTML（Valid でない HTML）で構成される記事ページが含まれている場合，ブロック識別子が適切に付与されない場合がある。3.4.5 項の実験で「404 Blog Not Found」が理想的な結果とならなかった原因は，そのような記事ページが 1 ページ含まれていたからである。構造が乱れた HTML であっても，ウェブブラウザで表示すると適切に表示される場合が多い。これは，構造が乱れた HTML であってもブラウザが柔軟に解釈するからである。このことから，ウェブブラウザのような HTML パーサを使用することにより，構造が乱れた HTML にも対応できると考えられる。

分離抽出手法は，コメントが付いていない記事ページがブログページ集合に含まれることを前提としている。そのため，全てのページにコメントが付いていると，コメントもポストとして抽出してしまう。3.4.5 項の実験で「My Life Between Silicon Valley and Japan」が理想的な結果とならなかった原因は，提案手法の前提に反し，全てのページにコメントが付いていたからである。この問題は，3.3.1 項で述べたとおり，フィードを用いて記事ページを収集することで解決できるが，既に収集済みの記事ページに適用するなど，過去の資源（データ）を有効活用するためには収集方法に依存せずに解決することも望まれる。コメント部分のブロック識別子はどのブログページ集合でも似ている傾向があること（ブロック識別子に「comment」を含むなど）に着目し，コメントが 1 件も抽出できない場合は，他のブログページ集合でコメントに分類されたブロックのブロック識別子を利用することにより，適切に抽出できる可能性がある。

スタイルシートを使用せずにデザインしているブログサイトに対しては，分離抽出手法を適用することはできず，要素識別子を用いずにブロックの役割を識別する必要がある。コンテンツブロックを包含するような DOM ツリーの構造パターンを獲得し，その構造をブロック識別子と見なすことで，要素識別子に頼らずブロックの役割を識別できる可能性がある。ウェブページ内において繰り返し出現する構造パターンの獲得は，コメントを抽出する手法 [Song 10, Kao 10] でも用いられているが，獲得範囲をページ集合に拡張することで，分離抽出手法を拡張できるかもしれない。なお，松本ら [松本 13] の調査によれば，一般のウェブページも含めスタイルシートの利用率が年々増加しており（2002 年に 21.3% だった利用率が 2012 年には 80.9% まで増加），今後も要素識別子を用いる分離抽

出手法は有効に機能すると考えられる。

3.5　本章のまとめ

本章では，事前に抽出ルールや学習データを準備する必要のないアルゴリズムで，ブログページ集合の記事ページからコンテンツ並びにポスト及びコメントを抽出する手法を提案した。日本語ブログサイトを対象とした実験により，提案手法の有効性を示した。提案手法は，ブログページ集合を与えさえすれば自動的にコンテンツ並びにポスト及びコメントを抽出するため，非常に小さな労力で記事ページからコンテンツ並びにポスト及びコメントを抽出できる。

ポスト及びコメントを対象とする分離抽出手法の性能は，コンテンツ抽出手法である ExtractUniqueBlock 法の性能に依存しており，コンテンツ抽出性能の改善のみで分離抽出性能が改善できることが示唆された。さらに，分離抽出手法を実現するために提案したブロック識別子を活用することで，コンテンツ抽出性能が改善することを示した。

第 4 章

集中的に検索されるクエリを対象としたカテゴリ特定

4.1 本章の概要

これまで提案されてきたクエリ分類問題は，主に恒常的に検索される検索クエリを対象とし，検索クエリの検索時期と分類時期に制約が設けられていなかった。このような環境下では，ユーザが実際に検索を行っている時期に分類せず，時期外れに分類するということが起こりえる。しかし，図 4.1 のように集中的に検索されるような状況がしばしば起こり，分類時期の制約は実用化に関する重要な課題である。本研究では，集中的に検索されるクエリを適切な時期に分類を行う問題に取り組む。

集中的に検索されるクエリを分類する際の課題は，分類のための素性が欠乏している点である。検索クエリの分類では，ユーザの検索ログ及びクリッキングログ[*1]を利用する場合が多いが，集中的に検索されるクエリに対しては，分類したい時点において，それらのログが十分に存在しないことがある。Wikipedia などの外部リソースを用いる手法も提案されているが，新製品の名称など，未知の検索クエリには適用が難しいと考えられる。

本研究では，素性が欠乏する問題に対し，ソーシャルメディアの 1 つである Twitter[*2]のデータを活用することで対応する。また，Twitter に投稿されたテキストと検索クエリ及びカテゴリとの類似性のみにとどまらず，投稿ユーザや投稿テキストに含まれるハッシュタグ[*3]の情報も利用する。提案手法は，あるカテゴリの情報を好むユーザはそのカテゴリ

[*1] 検索を行ったユーザが，検索結果のどのページに訪れたかを記録したログ。入力された検索クエリと，訪問先のページとの関連性が高いという仮説のもとで利用される。

[*2] `https://twitter.com/`

[*3] 投稿テキスト中にあらわれる，「#FCB」のようなハッシュサインから始まる文字列。ユーザが投稿ツ

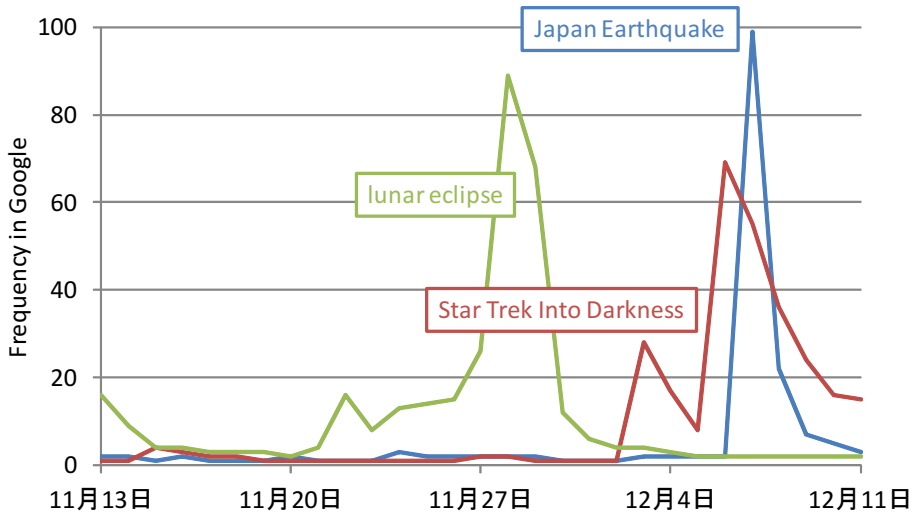

図 4.1　Google トレンドで調査した検索頻度（2012 年）

に属する単語を含むツイートを行う傾向がある，という仮説をもとにする。クエリ，ユーザ，ハッシュタグをノードとするグラフを構築し，グラフベースの半教師あり学習手法であるラベル伝搬アルゴリズムによって，クエリ分類問題を解く。なお，Twitter のデータを活用したクエリの分類は，Twitter のツイートの分類やハッシュタグの分類に似ており，それらに取り組んでいる例は存在する（[Sriram 10, Wang 11, Quercia 12, Posch 13] ほか）。ただし，筆者が調べた限り，分類時期に着目し，制約を設けている研究は存在しない。

4.2　集中的に検索されるクエリの例

本研究では，Windows Internet Explorer[*4]にインストールされた Bing Bar[*5]を通じて検索された検索クエリの中から集中的に検索されるクエリを選択し，それらの種別特定（カテゴリ分類）を行う。検索クエリの語数に制限は設けず，「筑波 研究」のような複数の語からなる検索クエリも対象とする。なお，ある検索クエリが現在，集中的に検索されているか否かの検知は本研究の対象外であり，集中的に検索されている時期及び検索クエリは，何らかの方法で与えられるものとする。

本節では，集中的に検索されたクエリを取り上げ，その検索頻度及び外部リソースにおける言及状況などについて検討する。外部リソースはニュース記事，ソーシャルメディ

イートのトピックを明示するために，任意に利用される。

[*4] http://windows.microsoft.com/en-US/internet-explorer/products/ie/home

[*5] http://toolbar.discoverbing.com/

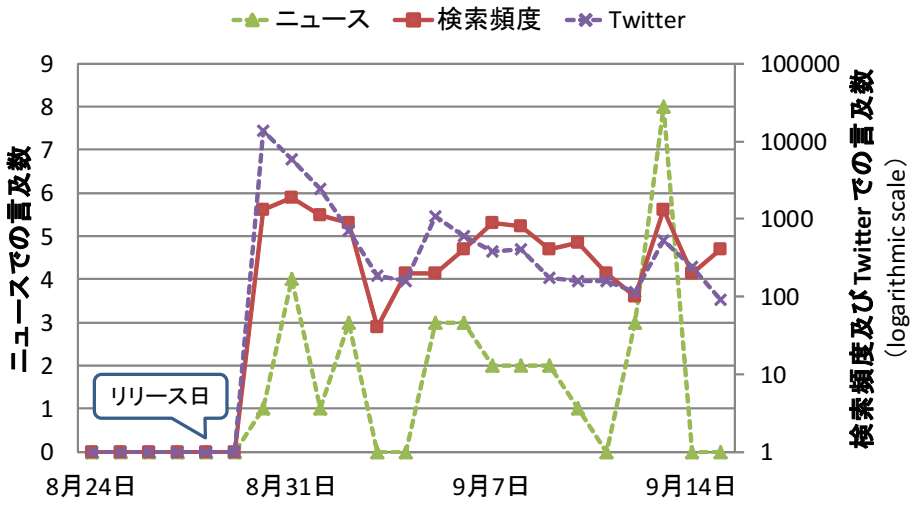

図 4.2　クエリ「カレログ」の検索頻度及び外部リソースの状況（2011 年）

ア，Wikipedia，はてなキーワード[*6]を利用する。時事情報の伝達手段であるニュース記事には新しい話題をふんだんに含んでおり，時宜にかなった言語リソースとしてしばしば利用される。本研究では全国紙及び地方紙を広くカバーする Ceek.jp News[*7]が収集したデータを利用する。また，ソーシャルメディアは即時性の高い言語リソースとして注目を集めており，本研究では Twitter のデータを利用する。Wikipedia 及びはてなキーワードはオンライン辞書サービスである。

図 4.2 は集中的に検索されたクエリ「カレログ」の検索頻度及び外部リソースの状況である[*8]。カレログは 2011 年 8 月 28 日にリリースされた携帯電話向けアプリケーションの名称であり，恋人を監視するために提供された。マルウェア同等の機能を有することから，ウェブ上で急に話題になった。Wikipedia では 2011 年 9 月 7 日，はてなキーワードでは 2011 年 8 月 30 日夜に「カレログ」に関する項目が作成された。クエリが初めて集中的に検索されたとき，同様に Twitter での言及が急増していることが確認できる。しかし，ニュース記事においては 1 記事でしか言及されていなかった。多数のニュース記事で言及されたのは 2 週間後であり，その日は，Twitter の 2 度目の急増が確認できる。このことから，Twitter は流行に敏感であり，集中的に検索されるクエリを分類するための素性として適していることが示唆される。

[*6] http://d.hatena.ne.jp/keyword/

[*7] http://news.ceej.jp/

[*8] 検索頻度及び Twitter での言及数は対数軸を利用しており，実値が 0 の場合，対数軸上の 1 にプロットしている。

4.3 提案手法

本研究は，n 個の検索クエリ集合 $Q = \{q_1, \ldots, q_n\}$ が与えられたとき，それぞれの検索クエリ q_i に，あらかじめ定めたカテゴリ集合 $C = \{c_1, \ldots, c_m\}$ のいずれが当てはまるかを解く問題である。通常，集中的に検索されるクエリは，新製品のリリースや映画の公開など，特定の話題及び時期によって引き起こされるため [Kulkarni 11]，あるクエリの属するカテゴリは 1 カテゴリのみであると仮定する。従って，あるクエリに対して最も見込みのあるカテゴリが，そのクエリの属するカテゴリであると見なす。

本研究では，即時性の高い外部リソースとしてソーシャルメディア（Twitter）の投稿に着目している。また，Twitter に投稿されたテキストと検索クエリ及びカテゴリとの類似性のみにとどまらず，投稿ユーザや投稿テキストに含まれるハッシュタグの情報も利用する。これらの情報を利用するにあたり，本研究では，あるカテゴリが既知な検索クエリを頻繁に言及する傾向のあるユーザはそのカテゴリの情報をフォローする傾向がある，また，あるカテゴリが既知な検索クエリと頻繁に共起するハッシュタグはそのカテゴリの検索クエリと共起しやすい，という仮説を立てた。

本研究では，Twitter のデータから QueryTwitterGraph $G = \{V, E, W\}$ を構築し，グラフベースの半教師あり学習手法でクエリ分類問題を解く。ここで，V は n 個のクエリノード V_q，l 個のユーザノード V_u，k 個のハッシュタグノード V_h で構成されるものとする（$N = n + l + k$）。E はノード間のエッジの有無（1 または 0）を示し，W はエッジの重み（スコア）を表す行列（$N \times N$）である。もし，ノード v_i と v_j の間にエッジがないならば $W_{ij} = 0$ となる。

4.3.1 グラフの構築

クエリとユーザの関係

図 4.3 のように Twitter はあるツイートがどのユーザによって投稿されたかの情報を提供しており，ツイートとユーザの関係が明確である[*9]。さらに，ツイートに含まれるクエリに着目すれば，そのクエリとユーザを関係付けることもできる。それらの関係を図 4.4 に示す。この図の上にあらわれるユーザは，音楽に属するクエリを含むツイートを多数投稿しており，ユーザは音楽を好む（音楽カテゴリに属する）と推測することができる。さらに，下にあらわれるユーザのように，あるクエリを含むツイートがスポーツを好むユー

[*9] https://twitter.com/ceekz/status/395039799091814400

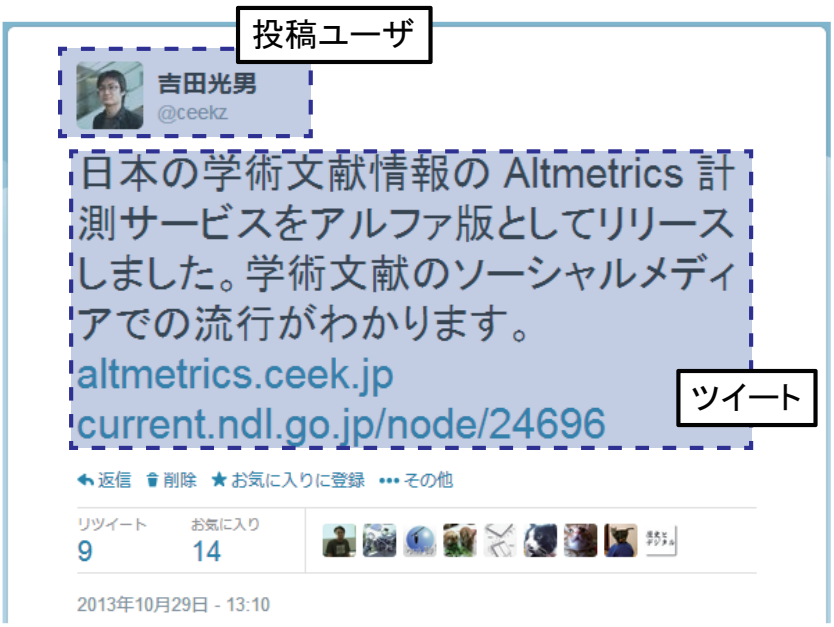

図 4.3　Twitter におけるツイートの例

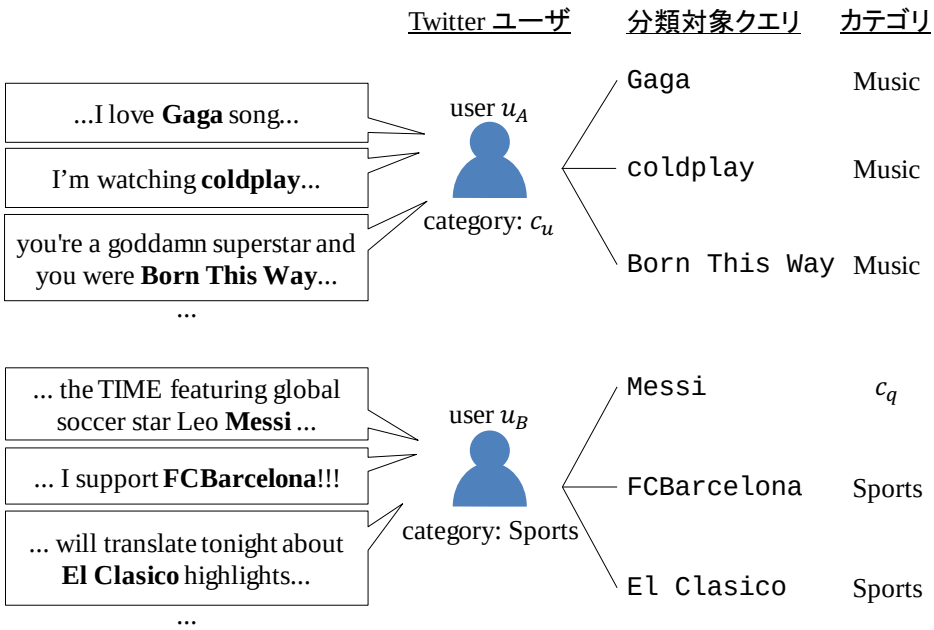

図 4.4　クエリとユーザの関係

ザによって投稿されているならば，そのクエリはスポーツに属する可能性が高いと推測できる。なお，クエリは 1 個以上の単語で構成されており，「クエリを含むツイート」とは，クエリを構成する単語全てが出現するツイートを意味する[*10]。

[*10] 出現するか否かの照合は，ツイートを分かち書きせず，単純な部分一致処理によって行う。

クエリとユーザの関係を利用するために，まず，ユーザノード集合 V_u を Twitter データから抽出する必要がある。検索クエリ q_i を含むツイート集合 T_i を Twitter データから抽出する。それぞれのツイートは投稿ユーザと関係付けられるため，ユーザ集合 U_i を構築できる。全てのクエリ集合にからユーザ集合 $U_{all} = \{U_1, \ldots, U_n\}$ を構築すると，重複するユーザが出現するため，重複を除いたユーザ集合 U_Q をユーザノード集合 V_u として利用する。

次に，クエリノード集合 V_q とユーザノード集合 V_u とをエッジ E で繋ぎ，それぞれの重みを表す行列 W を計算する。あるユーザがあるクエリを含むツイートを投稿すればするほど，そのクエリはそのユーザが属するカテゴリの可能性が高いと見立てる。この見立てをもとに， W を構成するそれぞれの重み W_{qu} は，ユーザ u が投稿したツイートの中に，クエリ q がどれだけ出現したかをもとに計算する。ユーザ側から見たとき $\psi(u \to q)$ はユーザがクエリに言及する可能性を意味し，クエリ側から見たとき $\psi(q \to u)$ はクエリがユーザに言及される可能性を意味する。これらを次のように計算する。

$$\psi(u \to q) = \frac{count(u,q)}{\sum_{q' \in Q} count(u,q')}, \psi(q \to u) = \frac{count(u,q)}{\sum_{u' \in U_Q} count(u',q)}$$

$count(u,q)$ はユーザ u のツイートにクエリ q が出現した回数を表す。ユーザから見るか，クエリから見るかで重みが異なるため，最終的なクエリ q とユーザ u との重みを以下のように計算する。

$$W_{qu} = \frac{\psi(u \to q) + \psi(q \to u)}{2}$$

クエリとハッシュタグの関係

Twitter の特徴として，ハッシュタグの存在が挙げられる。ハッシュタグは「#FCB」のようなハッシュサイン（「#」）から始まる文字列であり，投稿ユーザは自身のツイートの話題（トピック）を明示するために，自由に利用できる。しばしば，複数のユーザが同じハッシュタグを利用し，自身のツイートに付与する。この場合，ある話題に関する全てのツイートを閲覧しやすくする。筆者による調査では，約 10% のツイートにハッシュタグが含まれていた[*11]。クエリとユーザの関係と同様に，あるハッシュタグと共起する任意のクエリは，それぞれ同じカテゴリに属すと推定され（図 4.5），ハッシュタグはクエリ分類に有用な情報源であると考えられる。

クエリとハッシュタグの関係を利用するために，まず，検索クエリ q_i を含むツイート集合 T_i の中で出現したハッシュタグ集合 H_i を構築する。クエリとユーザの関係同様に，

[*11] 本研究で利用するために収集したツイート集合における割合。

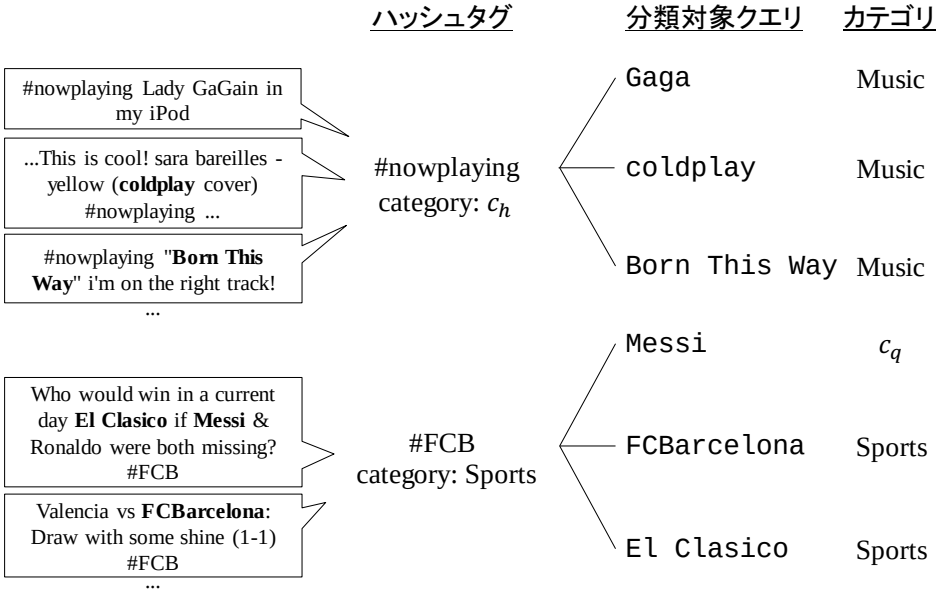

図 4.5　クエリとハッシュタグの関係

全てのクエリ集合にからハッシュタグ集合 $H_{all} = \{H_1, \ldots, H_n\}$ を構築すると，重複するユーザが出現するため，重複を除いたハッシュタグ集合 H_Q をハッシュタグノード集合 V_h として利用する。クエリとユーザの関係と同様に，クエリとハッシュタグとの間の重みをそれぞれ以下のように計算する。

$$\psi'(h \to q) = \frac{count'(h,q)}{\sum_{q' \in Q} count'(h,q')}, \psi'(q \to h) = \frac{count'(h,q)}{\sum_{h' \in H_Q} count'(h',q)}$$

$count'(h,q)$ はハッシュタグ h とクエリ q が共起した回数を表す。ハッシュタグから見るか，クエリから見るかで重みが異なるため，最終的なクエリ q とハッシュタグ h との重みを以下のように計算する。

$$W_{qh} = \frac{\psi'(h \to q) + \psi'(q \to h)}{2}$$

クエリ間の類似度

クエリとユーザ及びハッシュタグの関係のほかに，クエリが出現するツイートをもとにしたクエリ間の類似度を考慮する。これは，同じカテゴリのクエリは，同じような文脈で出現するという仮説に基づく。例えば，スポーツカテゴリに属するクエリであれば，「サッカー」や「トーナメント」などと共起しやすいことが挙げられる。クエリ q と q' との重み $W_{qq'}$ を，類似度 $sim(q,q')$ をもとに算出する。

類似度 $sim(q, q')$ を算出するために，検索クエリ q_i を含むツイート集合 T_i に対し，bag-of-words モデルを適用し，ベクトルを生成する。ベクトルの素性はツイート集合 T_i にあらわれる単語であり，各素性の値は TF-IDF [徳永 99] によって計算する。IDF の計算の際，各ツイートをそれぞれ 1 文書であると見なす。クエリ q と q' との類似度 $sim(q, q')$ は，それぞれのベクトル間のコサイン類似度とする[*12]。ただし，著しく類似度が低い関係を除外するため，閾値 ρ を導入し，$sim(q, q') < \rho$ のときは $W_{qq'} = 0$ であると見なす。

コサイン類似度をそのまま利用すると，あるノードがどのノードに対しても高い類似度を持つような状況が発生する可能性があり，各ノードが持ちうる重みを均一にするための正規化を行う。この正規化は，q に着目すると次の式で行われる。

$$sim'(q \to q') = \frac{sim(q, q')}{\sum_{q^* \in \{q^* | edge(q, q^*) \in E\}} sim(q, q^*)}.$$

q に着目した場合と，q' に着目した場合とで重みが異なるため，最終的な q と q' との重みは以下とする。

$$W_{qq'} = \frac{sim'(q \to q') + sim'(q' \to q)}{2}$$

重みの調整

グラフの構築作業において，クエリとユーザの関係，クエリとハッシュタグの関係，クエリ間の類似度をもとにそれぞれエッジの重みを定義した。図 4.6 はグラフ構築作業によって作成されるグラフの形状である。それぞれの重み W_{qu}，W_{qh}，$W_{qq'}$ は異なる仮説をもとにしており，どの重みを優先するかで，分類性能が変化する可能性がある。そこで，最終的なグラフ G を構築する際には，パラメータ α を導入し，重みの調整を行う[*13]。

$$W'_{qu} = \alpha W_{qu}, W'_{qh} = \alpha W_{qh}, W'_{qq'} = (1 - \alpha) W_{qq'}. \tag{4.1}$$

パラメータ α は 0 から 1 の値を取り，$\alpha = 0$ のとき，クエリ間の類似度に基づく重み $W_{qq'}$ のみが使われることを意味する。

[*12] コサイン類似度のほかに，ジャッカード係数，ダイス係数，シンプソン係数なども試したが，分類性能はコサイン類似度を利用したときが最も良かった。

[*13] 予備実験で W_{qu} 及び W_{qh} に別の重みを与えても結果が大きく変動しなかったため，同様に扱うこととした。

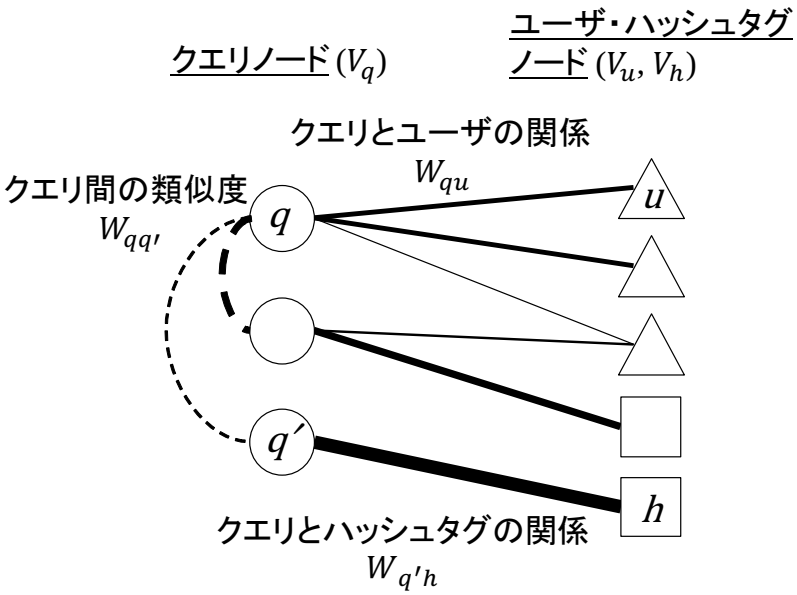

図 4.6　構築されるグラフ G の形状

4.3.2　グラフ伝搬アルゴリズム

前項までの処理により，Twitter のデータをもとにしたグラフ QueryTwitterGraph が構築できている。今回，グラフを利用した半教師あり学習手法でクエリ分類問題を解く。今回の半教師あり学習の枠組みでは，あらかじめ n_0 個のクエリノード集合 V_{q0} に，既定のカテゴリ集合 C に含まれるカテゴリが割り当てられているものとする。これらは，いわゆる学習データである。残りの n_1 個のクエリノード集合 V_{q1} にはカテゴリが割り当てられておらず（$n = n_0 + n_1$），これらはいわゆるテストデータである。本処理の目的は，ユーザ及びハッシュタグノードを通じて，クエリノード集合 V_{q0} から V_{q1} にカテゴリ情報を伝搬させることである。

グラフを利用した半教師あり学習手法は，ラベル伝搬（Label Propagation）アルゴリズムとして提案されている [Zhu 03, Zhou 04]。今回，ラベル伝搬アルゴリズムとして Talukdar ら [Talukdar 08, Talukdar 09, Talukdar 10] によって提案された手法を利用する。この手法は，大規模なグラフにおいて少数のシードのみでラベル伝搬を行う手法であり，少数のクエリノードに対し，大量のユーザ及びハッシュタグノードが存在する本研究との相性が良いと考えられる。実験の際は，Talukdar が提供している実装[*14]を利用する。

[*14] `https://github.com/parthatalukdar/junto` (Junto v1.2.2)

4.4　評価実験

本節では提案手法の有効性を示すための評価実験について述べる。まず，使用データ，実験手順（実験環境），比較手法について述べる。そして，比較手法を交えた実験結果について述べる。また，使用したデータの効用をグラフ構造から考察する。

4.4.1　データセット

実験に使用する検索クエリは，2011 年 7 月 1 日から 9 月 30 日までの間に Bing Bar で検索されたクエリのうち，日々の平均の 3 倍以上の頻度で検索された日（ピーク日）が存在するクエリを選択した[*15]。その際，検索頻度の少ないクエリは除外した[*16]。この結果，選択されたクエリは 5,721 クエリであった。

実験にあたり，それぞれのクエリに正解カテゴリを付与する必要がある。正解カテゴリの付与は，オンライン辞書サービスであるはてなキーワード[*17]のカテゴリと照合することで行った。有名なオンライン辞書サービスとして，はてなキーワードのほかに Wikipedia を挙げることができるが，Wikipedia はカテゴリ構造が複雑であり，1 カテゴリに限定して照合することが困難であったため，はてなキーワードを利用した。はてなキーワードは，図 4.7 の項目ページ[*18]のように，各見出し語に対して 1 カテゴリのみが付与されており，1 カテゴリに限定して照合するのが容易であった。はてなキーワードと照合したところ，2,923 クエリに対して 17 個のカテゴリが使われた（図 4.8）。本実験では，これら 2,923 クエリを利用する。

使用する Twitter データは，使用する検索クエリと同期間（2011 年 7 月 1 日から 9 月 30 日まで）に投稿された日本語ツイートである。Twitter Stream API[*19]を通じて 2,510 万件のツイートを収集し，その中から実験に利用するクエリを含む 450 万ツイートを抽出し，それらを利用した。また，クエリ間の類似度を算出する際は，bag-of-words モデルを利用するため，MeCab[*20] [Kudo 04] でツイートを分かち書きした。

[*15] 平均の 3 倍以上の頻度で検索される日が複数ある場合，検索頻度が最大だった日を “ピーク日” と見なす。

[*16] 今回，1% サンプリングされた検索ログを利用しており，ピーク日における頻度が 10 未満のものを除外した。

[*17] `http://d.hatena.ne.jp/keyword/`

[*18] `http://d.hatena.ne.jp/keyword/%A5%AB%A5%EC%A5%ED%A5%B0`

[*19] `https://dev.twitter.com/docs/streaming-apis/streams/public`

[*20] `http://code.google.com/p/mecab/` (MeCab v0.98 for MS-Windows)

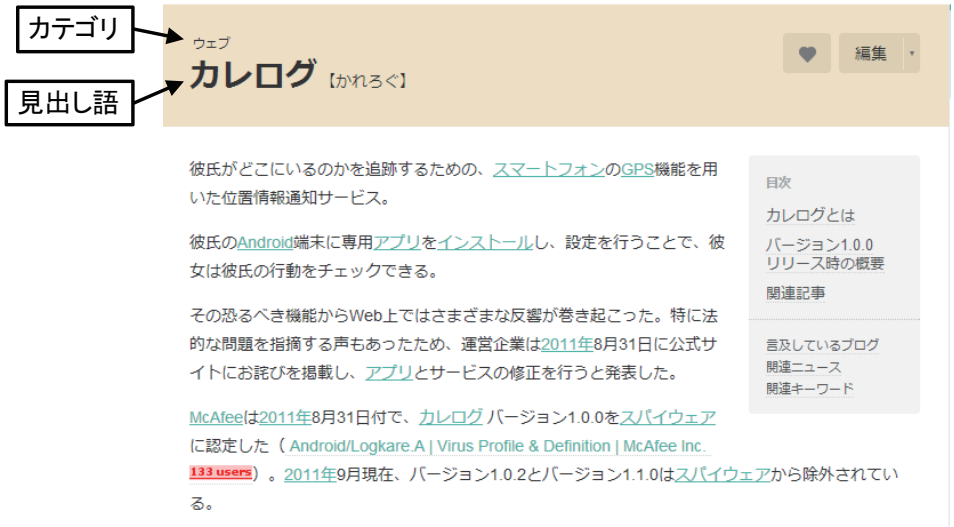

図 4.7　はてなキーワードの項目ページ例（カレログ）

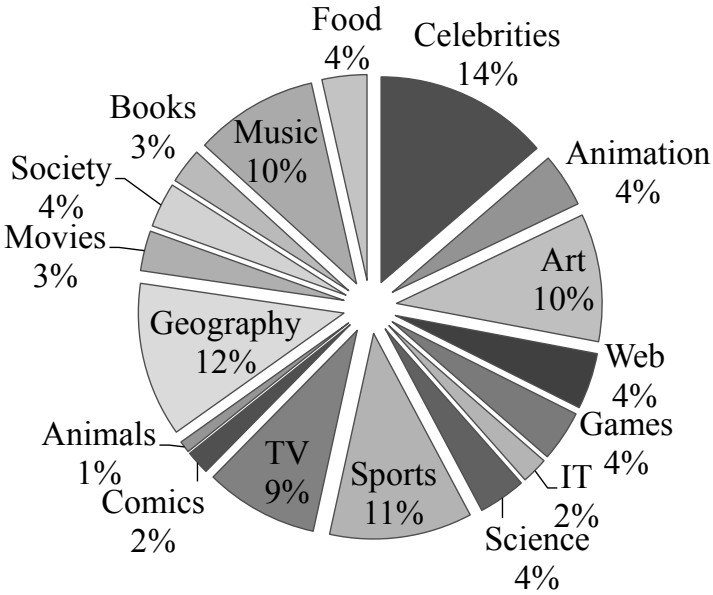

図 4.8　使用するクエリのカテゴリ分布

4.4.2　実験手順

ピーク日で適切に分類できているか確認するため，図 4.9 のように d 日間のウィンドウを動かしながら評価する。グラフを構築するために利用できるデータはウィンドウ内でピークがある検索クエリ，及びそのクエリを含む Twitter データのみである。ウィンドウの初日から $d-1$ 日目までのクエリに初期カテゴリを割り当て，ラベル伝搬アルゴリズムによって d 日目（テスト日）がピーク日であるクエリを分類する。

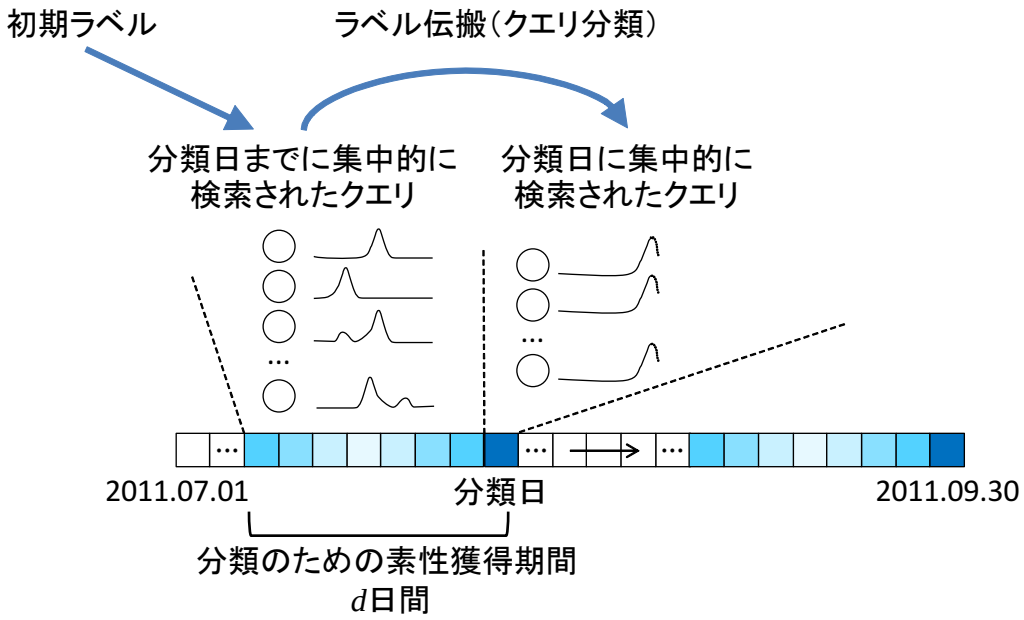

図 4.9　実験方法

今回，曜日の影響を排除するため，$d = 28$（$= 4$ 週間）として評価実験を行う。この環境では 65 のウィンドウで評価を行うことになり，最初のテスト日は 7 月 28 日である。最初の 28 ウィンドウをデベロップメント区間とし，パラメータの獲得を行う。残りの 37 ウィンドウをテスト区間として性能を評価する。

評価指標は，検索クエリの分類タスクとして有名な KDD-Cup 2005 と同様に，適合率（Precision），再現率（Recall），F 値（F-measure）を用いる [Li 05]。

$$Precision = \frac{\sum_i \text{カテゴリ } c_i \text{として正解したクエリの数}}{\sum_i \text{カテゴリ } c_i \text{として分類されたクエリの数}}$$

$$Recall = \frac{\sum_i \text{カテゴリ } c_i \text{として正解したクエリの数}}{\sum_i \text{カテゴリ } c_i \text{が割り当てられている正解クエリの数}}$$

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

4.4.3　比較手法

本実験は，クエリの分類時期を集中的に検索された日（ピーク日）に限定しているため，比較手法も同様の制約下で準備する必要がある。しかし，クエリの分類時期を限定した研究はこれまでに行われておらず，2.2 節で述べた既存研究との比較は困難である。本実験では，ベースラインとして Twitter データのかわりにニュースデータを利用する手法を準

備した。このベースラインとの比較により，Twitter データを利用することの有用性を評価できる。また，Twitter データを利用したクエリ分類において，どのような素性が有用であるかを評価するため，提案手法のサブグラフを準備した。

NewsGraph （ベースライン）

Twitter のデータのかわりにニュース記事のデータを利用する。ユーザの情報が含まれないため，クエリ間の類似度で構築されたグラフにより分類を行う。3 ヶ月の実験区間で使用されるニュース記事は，40.2 万記事（103 サイト）である。

QueryGraph

Twitter のデータを利用し，クエリ間の類似度で構築されたグラフにより分類を行う。提案手法のサブグラフである。

UserGraph

Twitter のデータを利用し，クエリとユーザ・ハッシュタグの関係で構築されたグラフを利用して分類を行う。提案手法のサブグラフである。

QueryTwitterGraph （提案手法）

QueryGraph 及び UserGraph を混合したグラフ G を利用して分類する。

4.4.4 実験結果

デベロップメント区間で選択されたパラメータは，NewsGraph では $\rho = 0.4$，QueryGraph では $\rho = 0.2$，QueryTwitterGraph では $\rho = 0.2$ 及び $\alpha = 0.8$ であった。これらのパラメータを利用し，テスト区間の評価を行った。

表 4.1 に実験結果を示す。太字は最高性能の手法を表す。適合率 50.9%，再現率 50.1%，F 値 50.5% と，いずれの指標においても提案手法が最高性能を示した。また，符号検定により，危険率 1% で有意差を確認した。NewsGraph と比較し，再現率が 20 ポイント向上しており，Twitter データは時宜にかなった分類に適していると推察される。ニュース記事は専門家（新聞記者）によって投稿されるのに対し，Twitter のツイートにはそのような制約はなく，Twitter のデータには比較的ノイズが多いと推察されるが，適合率が 5 ポイント向上している。UserGraph が NewsGraph よりも 3.4 ポイント高い適合率を示していることから，ユーザ及びハッシュタグノードによる影響であると考えられる。

QueryGraph と UserGraph を比較したとき，UserGraph の方が適合率は 3 ポイント程度，再現率は 4 ポイント程度，高い性能を示している（危険率 2% で有意差あり）。UserGraph はクエリとユーザ及びハッシュタグの関係のグラフであり，クエリ

表 4.1　提案手法によるクエリ分類性能（%）

グラフの種類	適合率	再現率	F 値
NewsGraph（ベースライン）	45.5	30.3	36.4
QueryGraph	46.1	42.1	44.0
UserGraph	48.9	45.7	47.3
QueryTwitterGraph（提案手法）	**50.9**	**50.1**	**50.5**

の分類に関し，ツイートテキスト以外の情報が有用であることを示している。また，QueryTwitterGraph の分類性能が QueryGraph 及び UserGraph の分類性能よりもさらに良くなっており，これら 2 つのグラフは性能を改善するために補完し合っていると認められる。

4.4.5　パラメータの評価

まず，クエリ間の類似度を用いたグラフ（QueryGraph）のエッジの量を調整するパラメータ ρ の影響を評価する。図 4.10 はデベロップメント区間において ρ を調整したときの適合率，再現率，F 値の変動グラフである。$\rho = 0.2$ のとき，F 値（41.6%）及び再現率（39.7%）が最高性能を示し，適合率（43.7%）も 2 番目に高い値になる。ρ の値を大きくすると，エッジの量が減少し，徐々に再現率が低下する。ρ が 0.8 を超えるとラベルが伝播されなくなり，適合率及び再現率がともに 0 になる。

パラメータ α はクエリとユーザ及びハッシュタグの関係，クエリ間の類似度のそれぞれの影響を調整する（式 4.1）。α が 0.5 よりも大きいとき，クエリとユーザ及びハッシュタグの関係がクエリ間の類似度よりも優先されることを意味する。図 4.11 はデベロップメント区間において α を調整したときの適合率，再現率，F 値の変動グラフである（$\rho = 0.2$）。α の値を大きくすると分類性能も良くなる。最も分類性能が良いのは，$\alpha = 0.8$ のときで，適合率（45.4%），再現率（45.1%），F 値（45.2%）のいずれも最高性能を示す。この結果は，クエリとユーザ及びハッシュタグの関係を利用することの有用性を示唆する。

4.4.6　クリッキングログとの比較

検索エンジン利用者が検索結果のどのページを訪問したかというクリッキングログは，検索クエリを拡張するのに有用な情報である。この情報は，先行するクエリ分類手法でも

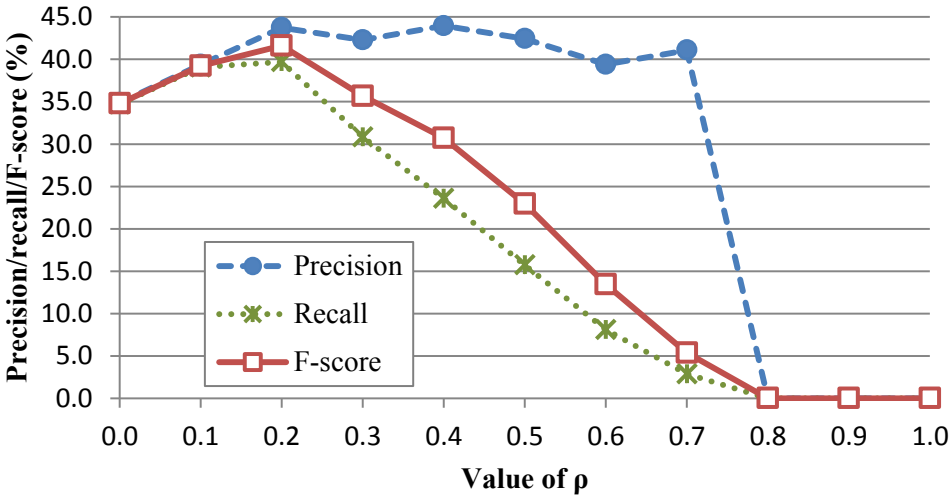

図 4.10　パラメータ ρ の調整 (QueryGraph)

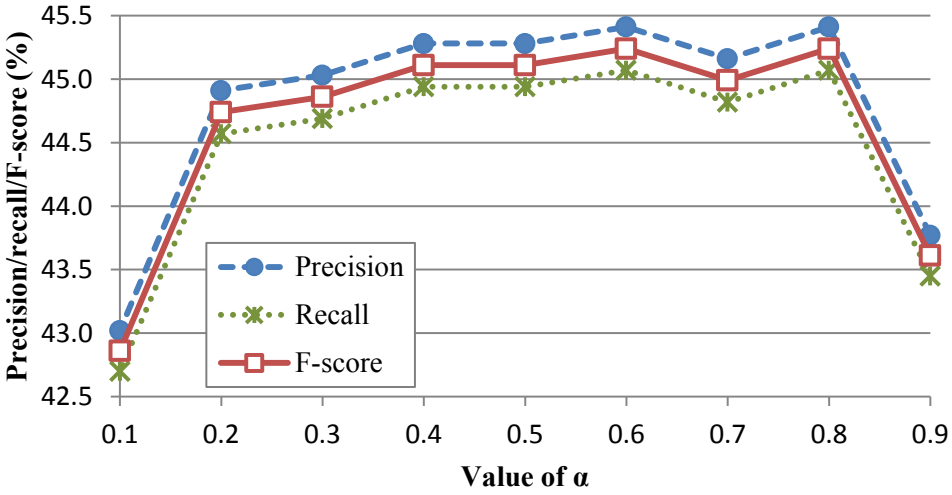

図 4.11　パラメータ α の調整 (QueryTwitterGraph, $\rho = 0.2$)

利用されており，高い分類性能を達成する。ここでは，クリッキングログと Twitter データとの分類性能の比較を行う。ただし，この実験では，クリッキングログ準備の都合上，全体の区間を 2011 年 9 月 3 日から 9 月 30 日に短縮している。本区間において利用可能なクエリは 809 クエリであった。

クリッキングログをもとにしたグラフ（ClickGraph）は，NewsGraph と同様にクエリ間の類似度をもとに構築する。使用するページは，各クエリにおいて最もアクセスされた 10 ページである[*21]。実験区間の短縮により，$d = 14$ と設定し，14 ウィンドウを準

[*21] ウェブページの収集は 2011 年 12 月に行ったため，ピーク日よりもあとに更新された情報か混入する可能性がある。

表 4.2　クリッキングログとの比較実験結果（%）

グラフの種類	適合率	再現率	F 値
ClickGraph（ベースライン）	35.4	32.1	33.7
QueryGraph	**38.5**	25.8	30.3
UserGraph	23.3	17.6	20.1
QueryTwitterGraph（提案手法）	36.8	**33.3**	**35.0**

備した。前半 7 ウィンドウはパラメータを調整するためのデベロップメント区間であり，後半 7 ウィンドウは性能評価のためのテスト区間である。QueryGraph，UserGraph，QueryTwitterGraph も同様の条件とした。

表 4.2 は ClickGraph（$\rho = 0.4$），QueryGraph（$\rho = 0.3$），UserGraph，QueryTwitterGraph（$\rho = 0.3, \alpha = 0.5$）のそれぞれの適合率，再現率，F 値である。太字は最高性能の手法を表す。QueryTwitterGraph は ClickGraph に匹敵する分類性能である確認できた[*22]。この結果は，提案手法が有望であり，Twitter データがクリッキングログに相当する資源であることを示唆している。

4.4.7　グラフ構造の評価

本項では，QueryTwitterGraph を構成する UserGraph と QueryGraph がどのように機能したかを理解するため，それらのグラフ構造を観察する。グラフ構造を可視化するためのツールとして Gephi[Bastian 09] を利用し，クラスタリングアルゴリズムは，大規模なグラフを高速に処理できる OpenOrd[Martin 11] を利用した。

図 4.12 はクエリとユーザ及びハッシュタグのノードのみで構成されたグラフ（UserGraph）の構造である。着色されたノードはクエリを表し，それぞれの色は正解カテゴリを表す。また，着色されていない白いノードはユーザ及びハッシュタグを表す。ノードのサイズは，あるクエリ並びにユーザ及びハッシュタグがツイートデータ中にどの程度出現したかを表す（大きい方が高頻度である）。なお，特定のテスト日におけるグラフであるが，構造を見やすくするために，ノードをサンプリングしてから Gephi で可視化した。この図からは，同じカテゴリに属すクエリが関係するユーザ及びハッシュタグとともにクラスタを作っていることが確認できる。これは，あるユーザまたはハッシュタグが，特定のカテゴリと強力に関係していることを意味する。また，異なるカテゴリに属すクエリ同士

[*22] 有意差は確認できなかった。

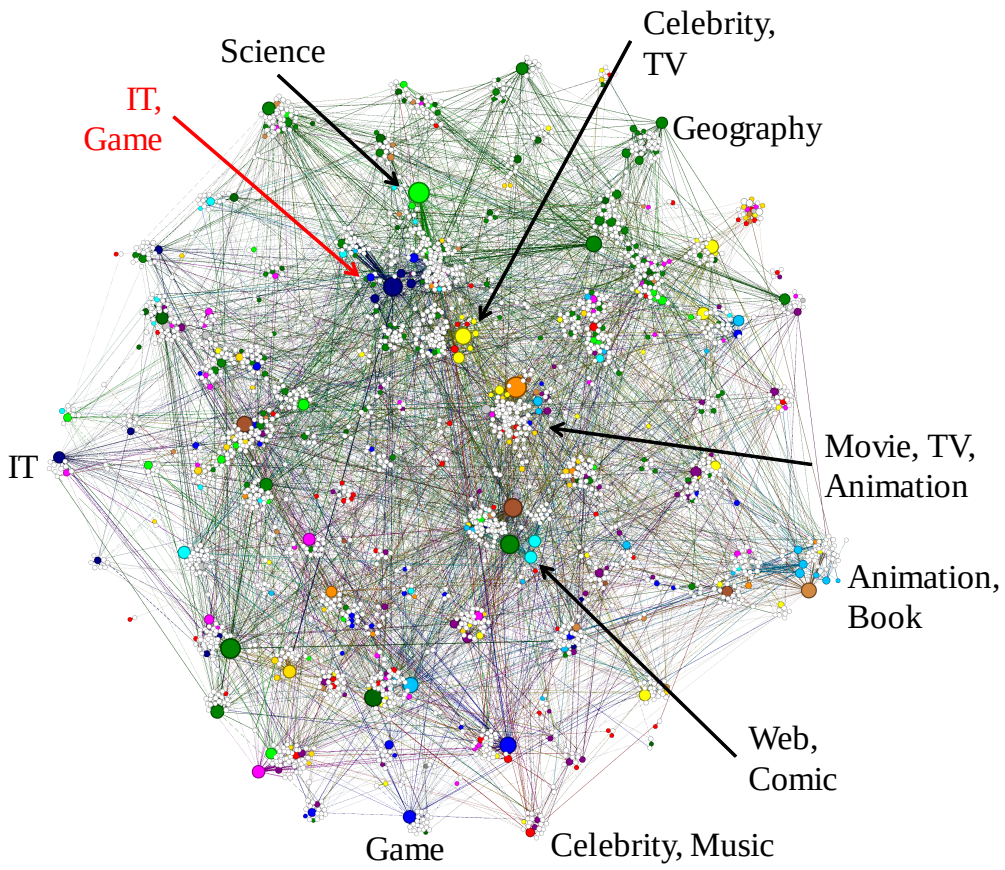

図 4.12　クエリとユーザ及びハッシュタグとのグラフ構造（UserGraph）

でクラスタを作っているのも確認できる。これは，ユーザが複数のカテゴリに属していることを示唆する。例えば，Game に興味を持つユーザは IT にも興味を持つ場合が挙げられる。

図 4.13 はクエリのみで構成されたグラフ（QueryGraph）の構造である。ノードの色は正解カテゴリを表し，ノードのサイズはツイート集合中にどの程度出現したかを表す。この図から，同じカテゴリに属すクエリ同士でクラスタを作っていることが確認できる。これは，クエリの出現するツイートの文脈が似ているのであれば，それらのクエリは同じカテゴリに属する傾向があることを示す。また，類似するカテゴリ同士でクラスタを作ることも確認できる。例えば，Celebrity，TV，Music の混合クラスタが挙げられる。これは，ある人物名が，買い物などの生活に関係する単語と共起する（Celebrity），出演したテレビに関する語と共起する（TV），リリースした CD に関する語と共起する（Music）といったように，ある語が様々なカテゴリを横断して出現することがしばしば発生するからである。

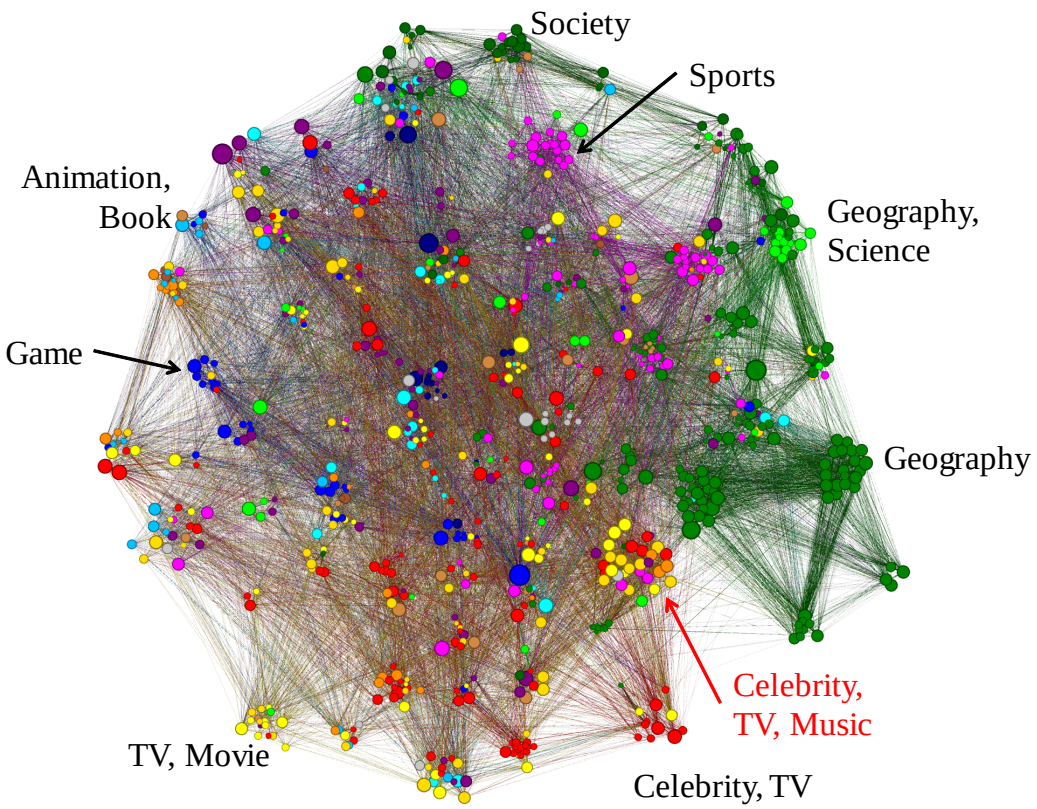

図 4.13　クエリ間のグラフ構造（QueryGraph）

UserGraph と QueryGraph の双方において，単一のカテゴリでクラスタを作る，また，複数のカテゴリでクラスタを作る様子が確認できた。複数のカテゴリで構成されるクラスタはクエリ分類に悪い影響を与えるが，双方のグラフを混合することで（QueryTwitterGraph）分類に適したグラフ構造になったと考えられる。

4.5　本章のまとめ

本章では，集中的に検索されるクエリを対象としたクエリ分類手法を提案した。このようなクエリを分類する際の問題は，分類に必要な素性が欠乏する点であるが，提案手法では，Twitter のデータを活用することで，この問題に対処した。分類時期を適切に設定した実験環境下において，ツイート，ユーザ，ハッシュタグのそれぞれの情報を組み合わせることで，ニュースデータを利用するよりもよりよい分類性能を達成した。特に，クエリとユーザ及びハッシュタグの関係に関する情報が有益であることを確認した。

第 5 章

おわりに

本研究では，ウェブ検索エンジンのテキスト種別特定に関する 2 つの課題に着目し，新たな手法を提案した。

1. ウェブページのコンテンツ並びにポスト及びコメントの自動特定
2. 集中的に検索されるクエリを対象としたカテゴリ特定

課題 1 では，あるウェブページのコンテンツは他のウェブページに出現しないという仮定をもとにしたコンテンツ抽出手法を提案した。さらにこのコンテンツ抽出手法を拡張し，コンテンツをポスト及びコメントに分離する手法を提案した。この手法は，複数のブログ記事ページを含むあるブログサイトにおいて，ポストは全ての記事ページに出現する一方で，コメントは一部の記事ページにしか付かない傾向があることに着目した手法である。これら 2 つの手法を組み合わせることで，人手による抽出ルールの作成，あるいは機械学習のための正解データの作成を行わず，HTML をテキスト処理するだけでコンテンツ並びにポスト及びコメントを抽出できる。実験では，従来手法及び従来手法を組み合わせた手法と比較を行い，提案手法の方が高精度に抽出できることを示した。

課題 2 では，集中的に検索されるクエリを適切な時期に分類する手法を提案した。ソーシャルメディア（Twitter）のデータを利用した手法であり，投稿されたテキストデータのほかに，ユーザやハッシュタグの情報も取り入れている。実験では，検索のピーク日に識別を試みるという実用的な環境において，ニュースデータやクリッキングログを利用する手法に対し，提案手法の方が高精度に識別できることを示した。

本研究で取り組んだ課題は，ウェブ検索エンジンを改良するための要素技術であり，ウェブ検索エンジンを通じた最終的なユーザ満足度の調査は今後の課題である。また，クエリ分類手法に関し，分類精度向上に寄与するユーザの発見及び傾向の調査，オンライン

学習などを取り入れた効率的な分類器の研究などを通じ，さらなる性能向上を目指したいと考えている。

参考文献

[AlemZadeh 12] AlemZadeh, M., Khoury, R., and Karray, F.: Query Classification using Wikipedia's Category Graph, *Journal of Emerging Technologies in Web Intelligence*, Vol. 4, No. 3, pp. 207–220 (2012)

[Alpert 08] Alpert, J. and Hajaj, N.: We knew the web was big..., http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html (2008)

[Baeza-Yates 06] Baeza-Yates, R., Calderón-Benavides, L., and González-Caro, C.: The Intention Behind Web Queries, in *String Processing and Information Retrieval*, Vol. 4209 of *Lecture Notes in Computer Science*, pp. 98–109 (2006)

[Bastian 09] Bastian, M., Heymann, S., and Jacomy, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks, in *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, ICWSM '09, pp. 361–362 (2009)

[Bhattarai 09] Bhattarai, A., Rus, V., and Dasgupta, D.: Characterizing Comment Spam in the Blogosphere through Content Analysis, in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Cyber Security*, CICS '09, pp. 37–44 (2009)

[Bing 08] Bing, L., Wang, Y., Zhang, Y., and Wang, H.: Primary content extraction with Mountain Model, in *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, CIT '08, pp. 479–484 (2008)

[Bos 09] Bos, B., Çelik, T., Hickson, I., and Lie, H. k. W.: Selectors, in *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*, chapter 5 (2009)

[Broder 02] Broder, A.: A taxonomy of web search, *SIGIR Forum*, Vol. 36, No. 2, pp. 3–10 (2002)

[Broder 07] Broder, A. Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V.,

and Zhang, T.: Robust classification of rare queries using web knowledge, in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pp. 231–238 (2007)

[Broder 08] Broder, A. Z., Ciccolo, P., Fontoura, M., Gabrilovich, E., Josifovski, V., and Riedel, L.: Search Advertising Using Web Relevance Feedback, in *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pp. 1013–1022 (2008)

[Cai 03] Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y.: VIPS: a Vision-based Page Segmentation Algorithm, Technical Report MSR-TR-2003-79, Microsoft Research (2003)

[Cao 08] Cao, D., Liao, X., Xu, H., and Bai, S.: Blog Post and Comment Extraction Using Information Quantity of Web Format, in *Proceedings of the 4th Asia Infomation Retrieval Symposium*, AIRS '08, pp. 298–309 (2008)

[Cao 09] Cao, H., Hu, D. H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., and Yang, Q.: Context-aware Query Classification, in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pp. 3–10 (2009)

[Chang 12] Chang, C.-H. and Chen, J.-M.: Automatic Extraction of Blog Post from Diverse Blog Pages, in *Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, WI-IAT '12, pp. 129–136 (2012)

[Chowdhury 02] Chowdhury, A., Frieder, O., Grossman, D., and McCabe, M. C.: Collection Statistics for Fast Duplicate Document Detection, *ACM Trans. Inf. Syst.*, Vol. 20, No. 2, pp. 171–191 (2002)

[Clark 99] Clark, J. and DeRose, S.: *XML Path Language (XPath)* (1999)

[Dai 11] Dai, N., Qi, X., and Davison, B. D.: Bridging Link and Query Intent to Enhance Web Search, in *Proceedings of the 22Nd ACM Conference on Hypertext and Hypermedia*, HT '11, pp. 17–26 (2011)

[Debnath 05] Debnath, S., Mitra, P., Pal, N., and Giles, C. L.: Automatic Identification of Informative Sections of Web Pages, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 9, pp. 1233–1246 (2005)

[Diemert 09] Diemert, E. and Vandelle, G.: Unsupervised query categorization us-

ing automatically-built concept graphs, in *Proceedings of the 18th international conference on World wide web*, WWW '09, pp. 461–470 (2009)

[Finn 01] Finn, A., Kushmerick, N., and Smyth, B.: Fact or Fiction: Content Classification for Digital Libraries, in *Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries* (2001)

[Gottron 07] Gottron, T.: Evaluating Content Extraction on HTML Documents, in *Proceedings of the 2nd International Conference on Internet Technologies and Applications*, ITA '07, pp. 123–132 (2007)

[Gottron 08a] Gottron, T.: Combining content extraction heuristics: the CombinE system, in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '08, pp. 591–595 (2008)

[Gottron 08b] Gottron, T.: Content Code Blurring: A New Approach to Content Extraction, *19th International Conference on Database and Expert Systems Application*, pp. 29–33 (2008)

[Gupta 03] Gupta, S., Kaiser, G., Neistadt, D., and Grimm, P.: DOM-based content extraction of HTML documents, in *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pp. 207–214 (2003)

[Gupta 05a] Gupta, S., Kaiser, G. E., Grimm, P., Chiang, M. F., and Starren, J.: Automating Content Extraction of HTML Documents, *World Wide Web*, Vol. 8, No. 2, pp. 179–224 (2005)

[Gupta 05b] Gupta, S., Kaiser, G., and Stolfo, S.: Extracting context to improve accuracy for HTML content extraction, in *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pp. 1114–1115 (2005)

[Hemenway 04] Hemenway, K. and Calishain, T.: Spidering hacks—ウェブ情報ラクラク取得テクニック 101 選 (2004)

[東川 09] 東川 玲：インターネット検索エンジンの現状と市場規模等に関する調査研究, Technical report, 総務省情報通信政策研究所 (IICP) (2009)

[Hu 07] Hu, M., Sun, A., and Lim, E.-P.: Comments-oriented blog summarization by sentence extraction, in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pp. 901–904 (2007)

[Hu 08] Hu, M., Sun, A., and Lim, E.-P.: Comments-Oriented Document Summarization: Understanding Documents with Readers' Feedback, in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 291–298 (2008)

[Hu 09] Hu, J., Wang, G., Lochovsky, F., Sun, J.-t., and Chen, Z.: Understanding user's query intent with wikipedia, in *Proceedings of the 18th international conference on World wide web*, WWW '09, pp. 471–480 (2009)

[Hu 12] Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., and Zheng, Q.: Mining Query Subtopics from Search Log Data, in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pp. 305–314 (2012)

[Ji 11] Ji, M., Yan, J., Gu, S., Han, J., He, X., Zhang, W. V., and Chen, Z.: Learning Search Tasks in Queries and Web Pages via Graph Regularization, in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pp. 55–64 (2011)

[Jiang 13] Jiang, D., Pei, J., and Li, H.: Mining Search and Browse Logs for Web Search: A Survey, *ACM Trans. Intell. Syst. Technol.*, Vol. 4, No. 4, pp. 57:1—-57:37 (2013)

[Kao 10] Kao, H.-A. and Chen, H.-H.: Comment Extraction from Blog Posts and Its Applications to Opinion Mining, in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, LREC '10, pp. 1113–1120 (2010)

[Khoury 11] Khoury, R.: Query classification using Wikipedia, *International Journal of Intelligent Information and Database Systems*, Vol. 5, No. 2, pp. 143–163 (2011)

[北 02a] 北 研二, 津田 和彦, 獅々堀 正幹：索引語の抽出と重み付け, 情報検索アルゴリズム, 第 3 章, pp. 27–49 (2002)

[北 02b] 北 研二, 津田 和彦, 獅々堀 正幹：情報検索の適用と評価, 第 2 章, pp. 9–26 (2002)

[König 09] König, A. C., Gamon, M., and Wu, Q.: Click-through Prediction for News Queries, in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pp. 347–354 (2009)

[Kudo 04] Kudo, T., Yamamoto, K., and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, in *Proceedings of the 2004 Con-*

ference on Empirical Methods in Natural Language Processing, EMNLP '04, pp. 230–237 (2004)

[Kulkarni 11] Kulkarni, A., Teevan, J., Svore, K. M., and Dumais, S. T.: Understanding temporal query dynamics, in *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pp. 167–176 (2011)

[Li 05] Li, Y., Zheng, Z., and Dai, H. K.: KDD CUP-2005 report: facing a great challenge, *ACM SIGKDD Explorations Newsletter*, Vol. 7, No. 2, pp. 91–99 (2005)

[Li 07] Li, B., Xu, S., and Zhang, J.: Enhancing Clustering Blog Documents by Utilizing Author/Reader Comments, in *Proceedings of the 45th Annual Southeast Regional Conference*, ACMSE '07, pp. 94–99 (2007)

[Li 08a] Li, X., Wang, Y.-Y., and Acero, A.: Learning query intent from regularized click graphs, in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pp. 339–346 (2008)

[Li 08b] Li, Z., Shi, S., and Zhang, L.: Improving Relevance Judgment of Web Search Results with Image Excerpts, in *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pp. 21–30 (2008)

[Li 10] Li, X., Wang, Y.-Y., Shen, D., and Acero, A.: Learning with Click Graph for Query Intent Classification, *ACM Trans. Inf. Syst.*, Vol. 28, No. 3, pp. 12:1—-12:20 (2010)

[Lin 02] Lin, S.-H. and Ho, J.-M.: Discovering informative content blocks from Web documents, in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pp. 588–593 (2002)

[Ma 12] Ma, Z., Sun, A., Yuan, Q., and Cong, G.: Topic-Driven Reader Comments Summarization, in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pp. 265–274 (2012)

[MacQueen 67] MacQueen, J.: Some methods for classification and analysis of multivariate observations, in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 281–297 (1967)

[Manku 07] Manku, G. S., Jain, A., and Das Sarma, A.: Detecting Near-duplicates for Web Crawling, in *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pp. 141–150 (2007)

[Mantratzis 05] Mantratzis, C., Orgun, M., and Cassidy, S.: Separating XHTML content from navigation clutter using DOM-structure block analysis, in *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, HYPERTEXT '05, pp. 145–147 (2005)

[Martin 11] Martin, S., Brown, W. M., Klavans, R., and Boyack, K. W.: OpenOrd: an open-source toolbox for large graph layout, in *Proceedings of Visualization and Data Analysis 2011* (2011)

[松本 13] 松本 章代, 今村 真浩, 小西 達裕, 高木 朗, 小山 照夫, 三宅 芳雄, 伊東 幸宏：日本語ウェブ文書を対象とした 10 年間の実態調査（2002 年〜2012 年）, 第 5 回データ工学と情報マネジメントに関するフォーラム, DEIM '13 (2013)

[中山 09] 中山 浩太郎, 伊藤 雅弘, Erdmann, M., 白川 真澄, 道下 智之, 原 隆浩, 西尾 章治郎：Wikipedia マイニング:Wikipedia 研究のサーベイ, 情報処理学会論文誌. データベース, Vol. 2, No. 4, pp. 49–60 (2009)

[南野 04] 南野 朋之, 鈴木 泰裕, 藤木 稔明, 奥村 学：blog の自動収集と監視, 人工知能学会論文誌, Vol. 19, No. 6, pp. 511–520 (2004)

[奥村 04] 奥村 学：自然言語による情報アクセス技術：3. テキスト自動要約, 情報処理, Vol. 45, No. 6, pp. 574–579 (2004)

[Pappas 12] Pappas, N., Katsimpras, G., and Stamatatos, E.: Extracting Informative Textual Parts fromWeb Pages Containing User-Generated Content, in *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '12, pp. 4:1–4:8 (2012)

[Parapar 10] Parapar, J., López-Castro, J., and Barreiro, A.: Blog Snippets: A Comments-Biased Approach, in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pp. 711–712 (2010)

[Pinto 02] Pinto, D., Branstein, M., Coleman, R., Croft, W. B., King, M., Li, W., and Wei, X.: QuASM: a system for question answering using semi-structured data, in *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, JCDL '02, pp. 46–55 (2002)

[Posch 13] Posch, L., Wagner, C., Singer, P., and Strohmaier, M.: Meaning As Collective Use: Predicting Semantic Hashtag Categories on Twitter, in *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13

Companion, pp. 621–628 (2013)

[Quercia 12] Quercia, D., Askham, H., and Crowcroft, J.: TweetLDA: Supervised Topic Classification and Link Prediction in Twitter, in *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pp. 247–250 (2012)

[Raggett 99] Raggett, D., Hors, A. L., and Jacobs, I.: The global structure of an HTML document, in *HTML 4.01 Specification*, chapter 7 (1999)

[佐伯 09] 佐伯 千種, 岩間 健宏：ブログの実態に関する調査研究, Technical report, 総務省情報通信政策研究所 (IICP) (2009)

[澤 08] 澤 菜津美, 森嶋 厚行, 杉本 重雄, 北川 博之：HTML ラッパ自動構築手法の提案, 日本データベース学会論文誌, Vol. 7, No. 1, pp. 263–268 (2008)

[Shen 06] Shen, D., Sun, J.-T., Yang, Q., and Chen, Z.: Building bridges for web query classification, in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pp. 131–138 (2006)

[Song 04] Song, R., Liu, H., Wen, J.-R., and Ma, W.-Y.: Learning block importance models for web pages, in *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pp. 203–211 (2004)

[Song 10] Song, X., Liu, J., Cao, Y., Lin, C.-Y., and Hon, H.-W.: Automatic Extraction of Web Data Records Containing User-Generated Content, in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pp. 39–48 (2010)

[Sriram 10] Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M.: Short Text Classification in Twitter to Improve Information Filtering, in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pp. 841–842 (2010)

[Talukdar 08] Talukdar, P. P., Reisinger, J., Pasca, M., Ravichandran, D., Bhagat, R., and Pereira, F.: Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 582–590 (2008)

[Talukdar 09] Talukdar, P. and Crammer, K.: New Regularized Algorithms for Transductive Learning, in *Machine Learning and Knowledge Discovery in Databases*, Vol. 5782 of *Lecture Notes in Computer Science*, pp. 442–457 (2009)

[Talukdar 10] Talukdar, P. P. and Pereira, F.: Experiments in Graph-Based Semi-Supervised Learning Methods for Class-Instance Acquisition, in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pp. 1473–1481 (2010)

[徳永 99] 徳永 健伸：情報検索の基礎, 情報検索と言語処理, 第 2 章, pp. 11–67 (1999)

[鶴田 08] 鶴田 雅信, 増山 繁：未知のサイトに含まれる Web ページからの主要部分抽出手法, 言語処理学会第 14 回年次大会発表論文集, NLP 2008, pp. 197–200 (2008)

[鶴田 10] 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)

[Ullegaddi 11] Ullegaddi, P. V. and Varma, V.: Learning to Rank Categories for Web Queries, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pp. 2065–2068 (2011)

[Wang 11] Wang, X., Wei, F., Liu, X., Zhou, M., and Zhang, M.: Topic Sentiment Analysis in Twitter: A Graph-based Hashtag Sentiment Classification Approach, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pp. 1031–1040 (2011)

[Weninger 10] Weninger, T., Hsu, W. H., and Han, J.: CETR: Content Extraction via Tag Ratios, in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pp. 971–980 (2010)

[Yi 03] Yi, L., Liu, B., and Li, X.: Eliminating Noisy Information in Web Pages for Data Mining, in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD '03, pp. 296–305 (2003)

[Yoon 10] Yoon, S., Jatowt, A., and Tanaka, K.: Intent Feature Discovery Using Q&A Corpus and Web Data, in *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, ICUIMC '10, pp. 40:1—-40:7 (2010)

[Zhou 04] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B.: Learning with Local and Global Consistency, in *Advances in Neural Information Processing Systems 16* (2004)

[Zhu 03] Zhu, X., Ghahramani, Z., and Lafferty, J.: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions, in *Proceedings of the Twentieth International Conference on Machine Learning*, ICML '03, pp. 912–919 (2003)

博士論文に関する論文

査読付き学術雑誌論文

本論文における第 3 章の主内容は下記として公表済みである。

1. 吉田光男, 乾孝司, 山本幹雄.
   ブログページ集合からのポストおよびコメント自動分離抽出手法.
   情報処理学会論文誌, vol.54, no.12, pp.2502-2512, 2013 年 12 月.
2. 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 年 6 月.

査読付き国際会議論文

本論文における第 4 章の主内容は下記として公表済みである。

1. Mitsuo Yoshida and Yuki Arase.
   Exploiting Twitter for Spiking Query Classification.
   Proceedings of the 8th Asia Information Retrieval Societies Conference (AIRS 2012), LNCS, vol.7675, pp.138-149, Dec. 2012.

謝辞

山本幹雄教授には本研究はもちろんのこと，学群所属時から懇切なるご指導を承りました。これまでに，多くの無理を許していただいた上に，素晴らしい研究環境を与えてくださいました。謹んで感謝の意を表します。

徳永隆治教授，狩野均教授，滝沢穂高准教授，天笠俊之准教授，乾孝司助教の諸先生方には，本学位論文の審査を通じ，様々な観点から多くの助言を頂きました。厚く御礼申し上げます。

Microsoft Research Asia の荒瀬由紀博士は，大学とは別に研究環境を用意してくださいました。Microsoft Research Asia 滞在中はもちろんのこと，帰国後も本研究に関する相談に乗っていただきました。深く感謝いたします。

津川翔助教をはじめとする，知能情報研究室の皆さまには研究に限らず多方面においてご支援を頂きました。本当にありがとうございます。

本研究は，筑波大学大学院システム情報工学研究科および Microsoft Research Asia で行われたものであり，多くの方々のご支援のもと，遂行することができました。また，本研究の一部は JSPS 科研費・特別研究員奨励費 11J01016 の助成を受けています。