修士論文

同一文抽出に基づく
類似ページの検出と分類

指導教員　　黒橋 禎夫 教授

京都大学大学院情報学研究科
修士課程知能情報学専攻

姜ナウン

平成21年2月6日

同一文抽出に基づく類似ページの検出と分類

姜ナウン

内容梗概

近年，ウェブページが爆発的に増加しており，我々は検索エンジンを用いることにより多種多様な情報を得ることができる．しかし，ウェブページの約 40%が類似ページといわれており，検索結果に類似ページが含まれるという問題がある．

本研究では 1 億ページという大規模なウェブコレクションを対象として，類似ページ検出を行なう．本研究では類似ページを，文字列をある程度共有する 2 つのページと定義し，ミラーページなどの同一ページ，引用ページ，盗作ページなどが含まれる．

本手法はまず，各ページから長い低頻度の文を抽出する．これは，文長が長く，また，ウェブ全体での頻度が低い文を 2 ページで共有すればこれらのページは関連性が高いといえるためである．また，各ページにおいてコンテンツ領域を抽出し，コンテンツ領域にある文のみを類似ページ検出の手がかりとする．これは非コンテンツ領域にある文を共有しても 2 つのページに関連性が低いからである．以上の処理によって得られた文を共有するページペアを類似ページとみなす．

次に，類似ページを同一ページ，引用ページ，盗作ページなどに自動分類する．分類は，ページに対する類似文字列の割合である重複率，インリンク/アウトリンクの有無，URL の類似度などの様々な情報を用いて行なう．類似ページ検出の実験を行なったところ，単純な URL の正規化ではわからないミラーページや，引用ページ，様々なサイトから記事をはりあわせたようなスパムページを発見することができた．

Finding and Classifying Near-Duplicate Pages based on Identical Sentences Detection

Naun KANG

Abstract

The recent explosive increase of Web pages has made it possible for us to obtain a variety of information with a search engine. However, by some estimates, as many as 40% of the pages on the Web are duplicates of the other pages. Thus, there is a problem that some search results contain the duplicate pages.

This thesis proposes a method for detecting similar pages from a huge amount of Web pages: hundred million Japanese Web pages. Similar pages are defined as two pages that share some sentences, and are classified into mirror pages, citation pages and plagiaristic pages, etc.

First, from each page, relatively long sentences are extracted. This is because two pages tend to be relevant when they share relatively long sentences. A pair of pages that has the identical sentences is regarded as similar pages.

Next, similar pages are classified based on several information such as an overlap ratio, the number of inlinks/outlinks, and contents region extraction.

We conducted the similar page detection and classification on the large scale Japanese Web page collection, and can find some mirror pages that we cannot find by the simple URL normalization, citation pages, and plagiaristic pages.

同一文抽出に基づく類似ページの検出と分類

目次

第1章　はじめに

近年のウェブページの爆発的増加により，我々は検索エンジンを用いることにより多種多様な情報を得ることができる．それに伴ない，人々の生活が徐々にウェブに依存したものとなってきている．例えば，企業や人物のトップページや，天気予報，旅行先などすべてウェブで調べている．

検索を行なっていると，検索結果中にしばしば類似したページが現われ，検索結果の把握が阻害されてしまうことがある．例えば，ミラーページや，ブログの月別と日別ページなどがあり，その他にも掲示板の記事のコピーや商品のレビュー，ソーシャルブックマークとその元記事などがある．

ウェブには同一ページや部分的に類似したページが多数存在する．全ウェブページ中の 40%は同一ページであるという統計がある [1]．現在の検索エンジンでも類似ページの処理がある程度はマージが行なわれ,「関連ページ」として集約されているが，先ほど述べたとおり，マージが行なわれていない類似ページが存在しており，その取り扱いは十分ではない．

類似ページを判定することはユーザの検索結果把握の阻害を抑制することができるだけでなく，検索エンジンを作る側としても，ディスク容量が圧迫される，または，インデックス構築が遅くなるといった問題を軽減することができる．

また，冒頭でウェブページの爆発的増加と述べたが，それは多くの人々が情報を発信してきていること以上に,「ウェブ上の負の側面」が増えたためだと考えられる．ウェブ上の負の側面とは，スパムページ，スプログ (スパムブログ)，SEO(Search Engine Optimization)，盗作などであり，これらが爆発的に増加している．スプログとは，自動スクリプトで他のブログをコピーしてブログを大量生成させて自分へのリンクをはり，検索順位を上げることにより，特定の商業サイトに誘導するサイトのことである [2, 3]．これらは主にアフィリエイト目的のものである．盗作には他のサイトを無断コピーしたものがあり，また，上記のスプログも一種の盗作である．

以上のような背景のもと，本研究では 1 億ページという大規模なウェブコレクションを対象として，類似ページ検出を行なう．本研究での類似ページとは同一ページや段落が類似しているページも含む．基本的なアイデアは長い低頻度の文に着目し，それらを共有するページペアを類似ページとみなすものである．次に，各ページにおいてコンテンツ領域を抽出し，コンテンツ領域にある

文のみを類似ページ検出の手がかりとする．これは非コンテンツ領域にある文を共有しても 2 つのページに関連性が低いからである．以上の処理によって得られた文を共有するページペアを類似ページとみなす．そして，文の重複率，リンクなどの様々な情報を用いて，類似ページを同一ページ，引用ページ，盗作ページ，スパムページなどに自動分類する．

本論文の構成は以下の通りである．第 2 章では関連研究について述べ，第 3 章では本研究で扱う類似ページの分類と提案するシステムの概要について述べる．次に第 4 章で低頻度の長い文の抽出方法，第 5 章でコンテンツ領域抽出方法について述べる．そして第 6 章では類似ページの自動検出と分類について述べ，第 7 章で実験について，最後に第 8 章で結論を述べる．

第2章　関連研究

本研究と関連がある研究は，大きくわけて以下の3つである．

- 類似ページ検出
- コンテンツ領域抽出
- スパムページ検出

順に関連研究について述べる．

2.1　類似ページ検出

類似ページ検出に関する研究がさかんに行なわれている．その研究の目的としては，検索結果に類似ページを表示させない，検索エンジンを作る際にディスク容量を節約することができるなどがある．

Lyon らは trigram を用いて類似ページを検出している [4]．まず，文書を trigram の列に変換する．そして，$S(*)$ をページの trigram の集合とした時，類似度 R(Resemblance) を以下の式で与える．

$$R = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \tag{1}$$

また，含有率 C(containment) を以下の式で与える．

$$R = \frac{|S(A) \cap S(B)|}{|S(A)|} \tag{2}$$

実験は新聞コーパス (335 記事) とレポート (280 テキスト) を対象として行なっているが，現在扱われているテキスト量から比べるとはるかに小さいものである．

Manku らは Web ページクローリングの際にクロールしたページがすでにクロールしたページの類似ページかどうかを判定し，類似ページであればクロールしないといった手法を提案している [5]．類似ページの判定はまず各ページを f ビットの fingerprint に変換し，2 つのページの fingerprint の違いが k ビット以内であるかをチェックすることによって行なっている．

Henzinger は Broder らの手法 [6] と Charikar の手法 [7] を大規模ウェブデータを利用して評価している [8]．Broder らの手法は，まず，n の長さのトークンの sequence において，k トークンから 64bit の Rabin[9, 10] の fingerprint を作る．結果として，$n - k + 1$ の fingerprint が作られ，これは *single* と呼ばれている．

$S(d)$ をページ d の single とした時に，ページ d とページ d' の類似度を $\frac{S(d)\cap S(d')}{S(d)\cup S(d')}$ と定義する．この類似度計算を計算コストが高いので，以下のように近似する．m 個の異なる fingerprint 関数 $f_i(1 \leq i \leq m)$ に対して，$n-k+1$ の fingerprint を生成し，それぞれの i に対して最小値を求める．そして，ページ d とページ d' の類似度を fingerprint の最小値の一致度で近似する．一方，Charikar の手法は，まず，すべてのトークンを b 次元ベクトルに写像し，さらに正のエントリを 1 に，負のエントリを 0 に写像する．二つのページの cosine 類似度は写像したベクトルの bit の一致率と等しいことを利用して，類似度を計算する．1.6B という大規模なウェブコレクションを用いて二つの手法を実験したところ，Charikar の手法は Broder の手法に比べて異なるサイト間での near-duplicate なページを発見できることを示している．また，これらの二つの手法を組み合わせることにより精度，再現率ともに向上させられることを示している．

BarYossef らは，DUST(Different URLs with Similar Text) アルゴリズムを提案している [11]．このアルゴリズムはある URL のリストが与えられた時に，同じ内容である他の URL に変換されるルールを自動的に発見するものである．変換ルールは以下のように一般的に適用できるものとサイト特有のものがある．

- 一般的なもの
  - “~” → “/people”
  - “%7E” → “~”
- サイト特有のもの
  - “co.il/story_” → “co.il/story?id=”
  - “labs” → “laboratories”

また，パターンは部分文字列置換とパラメータ置換の 2 種類がある．この手法はこれまで説明した手法とは異なり，ウェブページの内容自体は全く見ず，URL リストのみから変換ルールを発見することにより，類似ページを検出する．

Xiao らは near-deplicate なレコードを効率的に発見するアルゴリズムを提案している [12]．そして，そのアルゴリズムを図 1 のような near-duplicate なウェブページ検出に適用している．

その他に盗作ページを検出するものとして，Hoad らの研究がある [13].

従来研究はタイムスタンプやメニューが異なるくらいを類似ページと呼んでいるが本研究は引用ページなどのように部分的に類似しているものも含める．

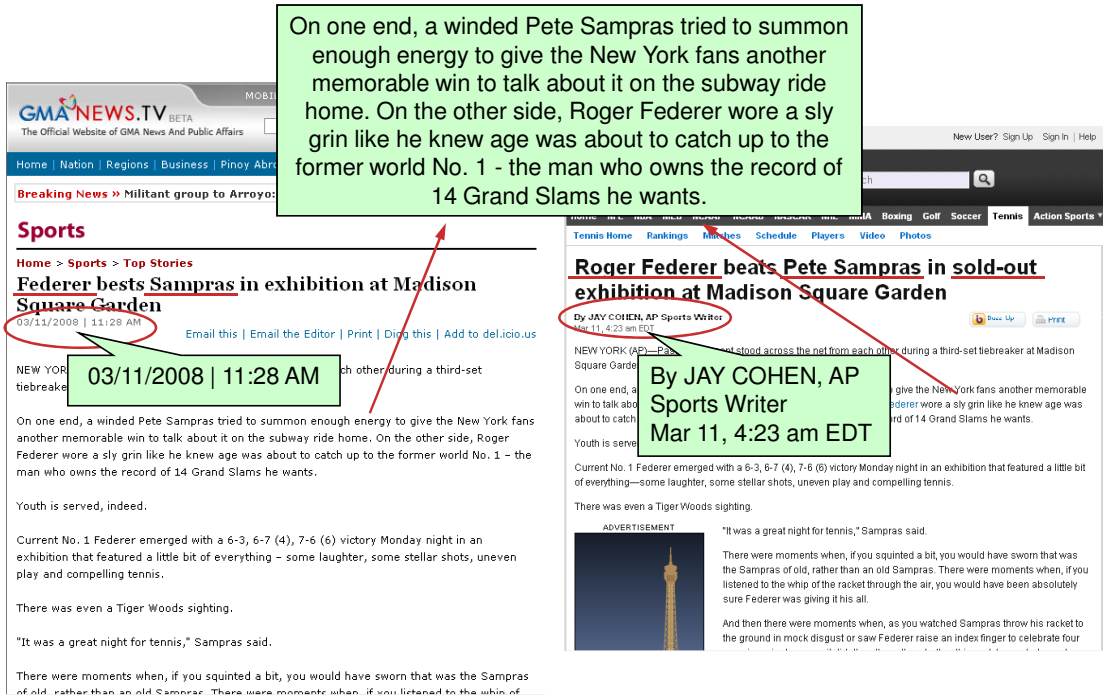

図 1: 類似ページ検出

2.2 コンテンツ領域抽出

ウェブページにおいてコンテンツ領域と非コンテンツ領域を分離する研究が行なわれている．研究の目的としては，非コンテンツ領域を検索インデックスから除外することにより速度向上やディスク容量の節約を行なえることや，データマイニングやテキストマイニングにおいて非コンテンツ領域を除外することにより処理時間を短縮することなどがある．

Lin らはウェブページから informative なコンテンツを自動抽出する手法を提案している [14]．<table>タグが用いられているページのみを対象とし，まず，<table>タグに基づき，ページをブロックにセグメンテーションする．そして各ブロックに対して，キーワード抽出を行ない，ページクラスタ (サイト) 内でのエントロピーを計算する．計算されたエントロピーの値が閾値以下であれば非コンテンツ領域としている．

Debnath らはコンテンツ領域を検出する二つのアルゴリズム，*FeatureExtractor* と *K-FeatureExtractor*(*FeatureExtractor* に K-means クラスタリングを加えたもの) を提案している [15]．この手法は Lin らの手法とは異なり，ウェブページに<table>タグがなくてもよい．実験により，上記で説明した Lin らの手法よ

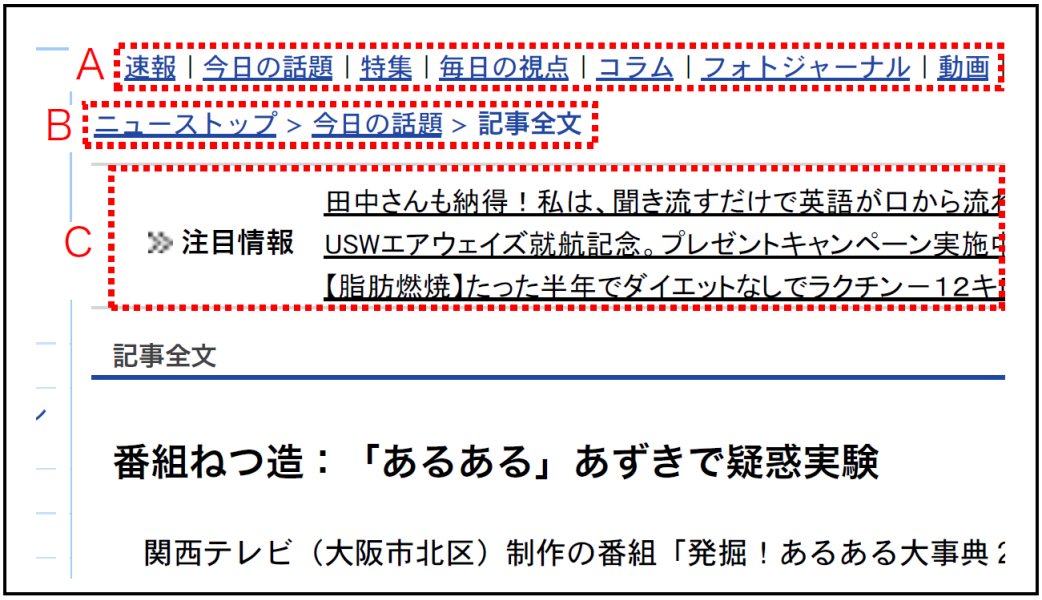

図 2: 非コンテンツ領域の例 (A や B の領域はナビゲーション目的のリンクであり，C の領域は広告である．非コンテンツ領域の定義は後続するアプリケーションにより異なると考えられ，例えばウェブ検索エンジンにとっては A や B の領域は非コンテンツ領域であるが，リンク構造を解析する場合は A や B の領域は非コンテンツ領域とすべきではない．)

りも精度，実行時間ともに上回っていることを示している．

中村らはウェブページにおける非コンテンツ領域を自動検出している [16]．図 2 に例を示す．まず，チャンキング問題として定式化した教師あり学習により，ウェブページをテキストユニットに分割する．非コンテンツ領域を示唆するキーワード (「広告」,「検索」,「著作権」,「ホーム」などの 47 個のキーワード)，テキスト長 (文字数が少なければ非コンテンツ領域になりやすい)，動詞または形容詞を含むかどうか (動詞や形容詞を含めばコンテンツ領域になりやすい)，DOM ツリーから得られる HTML タグなどを素性とし，非コンテンツ領域を検出するモデルを自動学習している．

2.3 スパムページ検出

日本のブログの 4 割はスパムといわれており[1)]，近年爆発的に増加しているスパムページやスプログに関する分析・自動検出に関する研究が行なわれてい

[1)] http://www.gamenews.ne.jp/archives/2008/03/4_47.html

る．スパムページやスプログは検索結果の順位を狂わせることからその検出が重要な課題となっている．

小野らはウェブスパムの中でも特に関連サイト同士で密にリンクを張ることによりランキング向上を図るリンクスパムの分布について調査している [2]．すべてのノードが互いにエッジによって相互連結された部分グラフであるクリークに着目し，極大クリークを大規模ウェブコレクションから抽出している．

佐藤らはスパムブログの収集と分析を行なっている [3]．まず，キーワードによって検索されるブログサイトの生起数の推移を観測することによりバースト現象を検出し，バースト日において特に一日の投稿記事数が多いブログサイトを中心にサイトの収集を行なっている．また，収集されたサイトに対して，スパムかどうかの判定を行ない，リンクやコピーされた文書などの情報に基づき，スパムブログの類型化を行なっている．

第3章　類似ページの分類とシステムの概要

本章ではまず，本研究で検出する類似ページとその分類について述べる．そして，本研究で提案するシステムの概要について述べる．

3.1　類似ページの分類

本研究では，ページ間である程度文字列を共有しているものを類似ページと定義する．したがって，類似ページには，完全に文字列が同じである同一ページや，いくつかの文を共有したページが含まれる．本研究で対象とする類似ページを文を共有する割合，仕方によって以下のように分類する．また，図3に例とともに示す．

- **同一**: 2つのページが同一のものである．例えば，ミラーページや，定型ページ (apacheが生成するページなど)，盗作ページなどがある．
- **包含**: 1つのページが他方のページに包含される場合である．例えば，ブログの月別ページと日別ページなどがある．
- **部分共有**: 2つのページでその部分を共有するページである．例えば，引用ページと被引用ページ，文集合を共有するページ，スパムページ，盗作ページと被盗作ページなどがある．引用ページの例を図4に示す．この例では右側のページが左側のページを引用しており，右側のページには，“[引用サイト] http://www.sittakaburi.jp/” と引用元が明記されている．次に，文集合を共有するページを図5に示す．左側のページ，右側のページともに個人情報保護法第18条第4項を引用しているが，このページ間にはリンクがない．

3.2　システムの概要

本研究ではウェブページコレクションとして，検索エンジン基盤TSUBAKI[17]で検索対象となっている日本語1億ページを用いた．検索エンジン基盤TSUBAKIとは，研究用途に主眼をおいた検索エンジンであり，ブラウザによる検索[1)]を提供するとともに，APIを介して誰でも自由かつ無制限にその検索結果を取得することが可能である．日本語1億ページは2007年5月から7月にかけてクロールされたものである．本研究ではこの1億ページを対象として類似ページ検出

[1)] http://tsubaki.ixnlp.nii.ac.jp/index.cgi

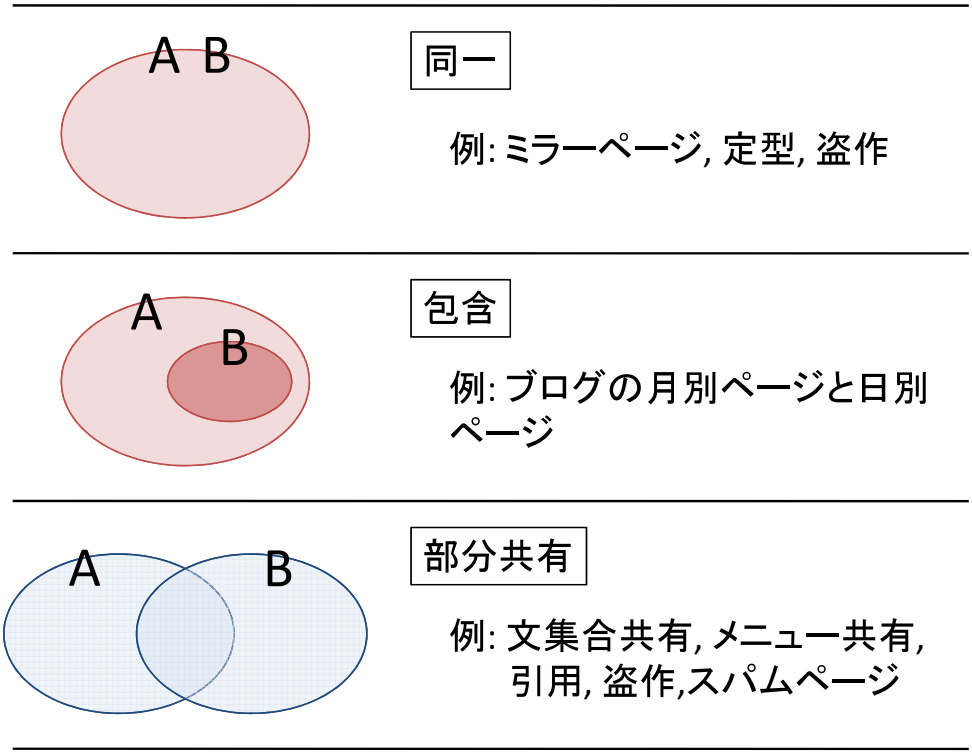

図 3: 類似ページの分類

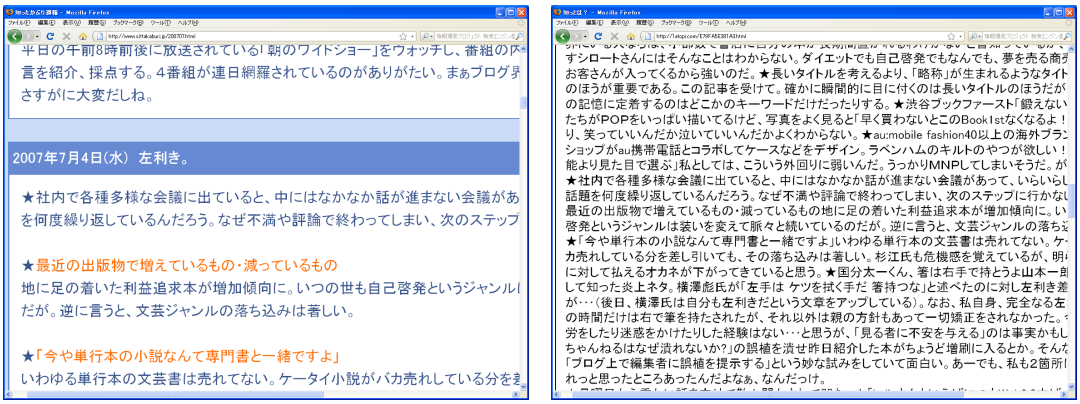

図 4: 引用ページの例

を行なう.

本研究で提案するシステムの概要を図 6 に示す. システムは大きくわけて以下の 3 ステップからなる.

1. 低頻度の長い文の抽出

   類似ページを検出する手がかりとして文を用いる. 単に文を共有しているだけではそのページ間が類似しているかどうかを判断できないので, 低頻

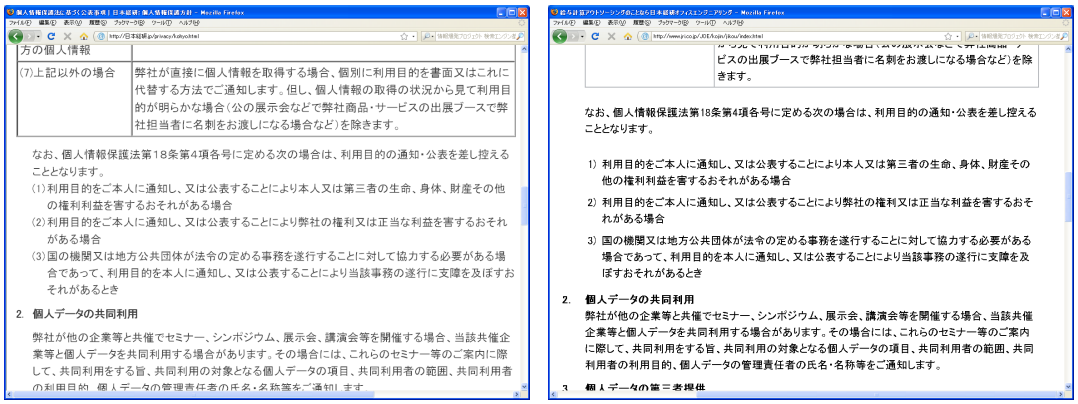

図 5: 文集合共有ページの例

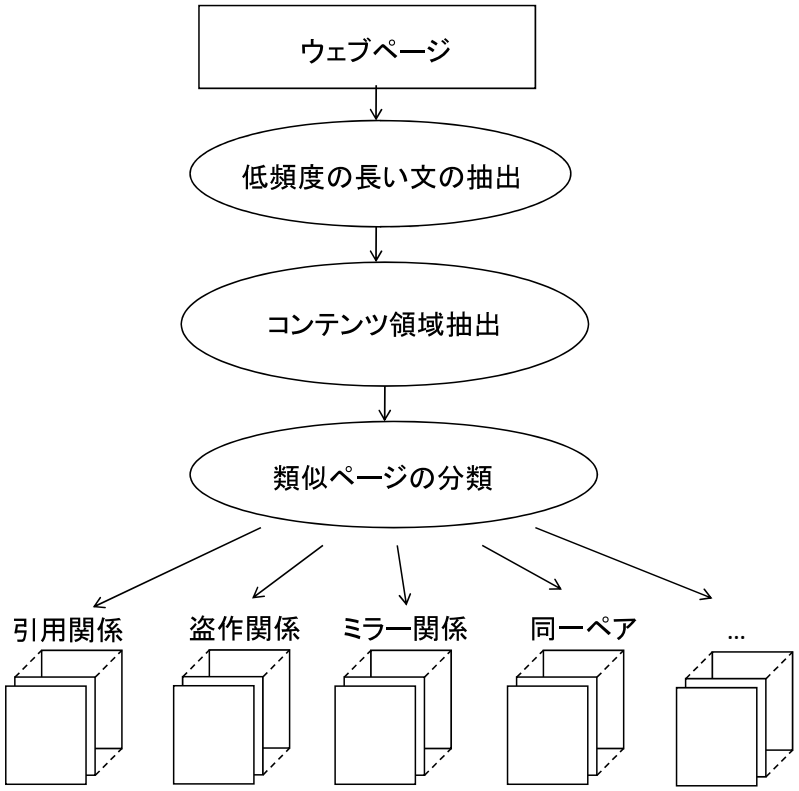

図 6: システムの概要

度の長い文に着目する．低頻度とは Web 全体での文書頻度が低い文を意味する．すなわち，高頻度の文というのはどの文書にでも現れる一般的な文であり，類似ページを検出する手がかりとはすることができず，低頻度の文を共有していればそのページ間には関連性がある可能性が高いといえる．また，あらゆる文のうち比較的長い文 (文字数の多い文) に着目する．たと

え低頻度の文であっても文が短いと偶然そのページに現れた可能性が高く，逆に文が比較的長いとそれらのページに関連性がある可能性が高くなるといえる．以上より，各文書から低頻度の長い文を抽出し，これを手がかりにして類似ページを検出する．

2. コンテンツ領域抽出

   ウェブページには主要なコンテンツ以外にメニューバーやカウンタといった非コンテンツ領域が存在する．上記で抽出した文が非コンテンツ領域に存在した場合，その文はそのページ内での重要性が低いと考えられ，類似ページ検出の手がかりにはなりにくい．一方，コンテンツ領域に存在すれば，その文はページ内での内容にあたる部分となり，重要性が高いと考えられ，類似ページ検出の手がかりとなる．

3. 類似ページ分類

   上記までの手順で抽出した各類似ページペアに対して，3.1 節で述べた分類のどれにあたるかを自動分類する．分類に用いる手がかりとしては重複率，URL の類似度，リンクなどを用いる．

第4章　低頻度の長い文の抽出

本章では類似ページ検出の手がかりとする低頻度の長い文を各ページから抽出する手法について述べる．1億ページのあらゆるページ間の類似度を計算するのは計算量が膨大となる ($O(n^2)$ の計算量)．そこで各ページから特徴量を抽出し，それが同一であるものを類似ページとみなす．従来研究では特徴量として trigram などが利用されているが，trigram であると関連のないページ間でのマッチが大量に生じる可能性があるので，本研究では意味的に完結している文 (句点から句点まで) を用いる．文を最小単位とすると，助詞などの一部が改変された文の類似性を検出することができないが，これについては今後の課題で述べる．

まず，検索エンジン基盤 TSUBAKI で提供されている標準フォーマットと呼ばれるデータについて説明し，そこから文を抽出する手法について述べる．そして，低頻度の長い文を選出する方法について述べる．

4.1　標準フォーマットからの文抽出

ウェブコーパスを研究に利用する場合に少なくとも以下のような処理を行なう必要がある．

- ウェブページのクローリング
- 各ページにおいて HTML タグの除去，文整形
  図7に示すように，赤字のタグは文区切りに，青字のタグはレイアウトのために利用されており，文境界が不明瞭である．

ウェブコーパスは新聞コーパスに比べて大規模，かつ，多様な話題を含むことにより近年利用されることが多くなってきたが [18, 19]，各研究者が上記のような処理を行なう必要があるとなるとウェブコーパスの研究利用の敷居が高くなってしまう．

そこで，検索エンジン基盤 TSUBAKI ではブラウザでの検索・API を提供するとともに，様々な処理をウェブ文書に適用したものを無制限に提供している．各ページは HTML タグの削除・文抽出などの処理が行なわれ，標準フォーマット [20, 21] と呼ばれる XML 形式に変換されて管理されており，URL・文字コードなどのメタ情報，文集合，インリンク (このページにリンクしているページ集合)，アウトリンク (このページがリンクしているページ集合) などの情報，言

```
しかしさすがに電池の持ちが悪くなってきたのと、<br>
たまたまキャンペーンをやっていて無料で機種変できるみたいだったから<br>

愛着のわいた携帯を手放すことにした。</p>

<p>で、折角変えるならまた長く使えるのがいいじゃない？<br>
すごいいいデザインのがあって（しかもロゴがsoftbank!!）<br>
これに決めた！！</p>

<p>と思ったら…</p>

<p>なんと品切れ。他の店舗に問い合わせてもどこも品切れ。<br>
唯一あったのが電車で40分かかる所。</p>

<p>もちろん行きました。</p>

<p>そこまでして手に入れた携帯だから前以上に既に愛着がわいてますw<br>
また5年間使い続けるぞい！</p>
                    </div></div>
                <p class="entry-footer">
                  <span class="post-footers">
                      投稿者: KN006 日時: 2006年10月16日 22:05
                  </span>
```

図 7: HTML タグの例 (赤字のタグは文区切りに，青字のタグはレイアウトのために利用されている．)

語解析結果が含まれている． 図 8 に標準フォーマットの例を示す．<Title>タグで囲まれた部分がそのページのタイトル，<Text>タグで囲まれた部分がそのページの本文から抽出された情報であり，<S>タグが本文から抽出された文の情報 (先頭からのオフセット，文長，文の ID などの情報)，<RawString>タグで囲まれた部分が HTML から抽出された文の生文字列を示す．標準フォーマットは API(http://tsubaki.ixnlp.nii.ac.jp/api.cgi) でアクセスすることができ，ユーザは前処理を行なうことなく研究をすることが可能となっている．

ウェブページから標準フォーマットへの変換手順を以下に示す．

1. 日本語ページ判定

   検索エンジン基盤 TSUBAKI では日本語ウェブページを検索対象とするために，日本語ページのみを収集している．そこで，ウェブページが日本語で書かれたものであるかの判定を行なう．ウェブページ中の charset 属性，

または，perl の Encoding::guess_encoding() 関数を用いてページの文字コードを調査し，以下の文字コードを日本語ページの候補とする．

euc-jp, x-euc-jp, iso-2022-jp, shiftjis, windows-932, x-sjis, shiftjp, utf8

上記で得られたページのうち助詞 (「が」,「を」,「に」,「は」,「の」,「で」) の含有率が 0.5%以上のページを日本語ページとみなす．それ以外のページは破棄する．

2. ページから日本語文の抽出
   まず，HTML タグ，改行を利用して段落を認識する．HTML タグとしては，<p>や ，<table>といったブロックタグを利用する．そして，抽出したテキストに対して以下の後処理を行なう．
   - HTML タグの消去
   - HTML エンティティのデコード
   - 全角に変換
   - 漢字間の空白を削除
   - 文字の正規化
3. 段落内の文分割
   文区切り文字 (「.」,「？」,「！」,「♪」,「…」) を手がかりとして文に分割する．上記の文字で分割すると過分割をおこす場合があるので，ルールにより過分割を修正する．例えば，文末が「！」や「♪」で次の文の文頭が「と」「っ」「です」であれば，文を連結する．

   (1)　１年ってあっという間！と思う．

   また，連続するアンカーテキストを 1 文にするかどうかを判定する．この判定は，連続するアンカーテキストの文字列を複合名詞に連結し，その複合名詞が Web 全体で閾値以上出現するかどうかで行なう．

   (a) 京都市左京区

   (b) 福地寿樹捕鯨問題

   上の例では，(a) の「京都市左京区」の頻度は閾値以上なので 1 文としてつなげるが，(b) の「福地寿樹捕鯨問題」の頻度は閾値以下なので別の文とする．

4. リンク情報の埋めこみ
   まず，アウトリンクを各ページから抽出し，標準フォーマットにアウトリンクの情報を付与する．全ウェブページからアウトリンクの情報を抽出した後，各ページにインリンクの情報を付与する．インリンク・アウトリンクともに文書 ID，URL，アンカーテキストの組からなる．
5. 日本語文の言語解析
   抽出した日本語文の言語解析もうめこむことにより，標準フォーマットを利用するユーザは各自で言語解析を行なう必要がなくなる．言語解析ツールは任意であるが，現在のところ，形態素解析器 JUMAN[1)] ・構文解析器 KNP[2)] ・同義表現解析 SynGraph[22] による言語解析結果がうめこまれており，<Annotaion>タグで囲まれている．

以上のような処理によって変換された標準フォーマットという形でウェブページが管理されているため，ユーザは HTML から簡単に文を抽出することができる．すなわち，標準フォーマットの<S>タグ内にある<RawString>タグで囲まれた文字列を抽出すればよい．また，<S>タグの Id 属性はその文が先頭から何文目かを表しており，この情報も合わせて抽出する．

4.2 低頻度の長い文の選出

3.2 節で述べた日本語 1 億ページには文が約 60 億文あり，図 9 に文の頻度の分布を示す．また，図 10 に文長の頻度分布を示す．

本研究では類似ページを検出する手がかりとして文を用いる．単に文を共有しているだけではそのページ間が類似しているかどうかは判断できないので，低頻度の長い文に着目する．

低頻度とは Web 全体での文書頻度が低い文を意味する．高頻度の文というのはどの文書にでも現れる一般的な文であり，類似ページを検出する手がかりとはすることができず，低頻度の文を共有していればそのページ間には関連性がある可能性が高いといえる．表 1 に各長さごとの文とその頻度の例を示す．例えば,「情報検索，読者レビュー，書店売上ランキングや，コミックや書籍の購入をご案内している，本の総合サ...」という文は Web で 3,800 回現れているが，

1) http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html
2) http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html

この文を含む 2 つのページは必ずしも関連があるとはいえない．一方で,「むしろ「開始直後というのは，夜間開催で生じる影響，異様さを住民が実感するチャンス」と見る向きもある.」という文は Web で 3 回しか出現しておらず，この文を含む 2 つのページは関連がある可能性が高い．

さらに，あらゆる文のうち比較的長い文 (文字数の多い文) に着目する．たとえ低頻度の文であっても文が短いと偶然そのページに現れた可能性が高く，逆に文が比較的長いとそれらのページに関連性がある可能性が高くなるといえる．

予備実験の結果，長さ 20 文字以上の頻度 10 回以下の文を用いることとし，これを手がかりにして類似ページを検出する．

```
<?xml version="1.0" encoding="utf-8" ?>
<StandardFormat Url="http://tabitano.main.jp/7kiyomizu.html" Original
Encoding="shiftjis">
<Header>
<Title Offset="512" Length="39">
 <RawString>清水寺【京都旅楽トラベル】</RawString>
</Title>
<OutLinks>
 <OutLink>
  <RawString>錦天満宮</RawString>
  <DocIDs>
   <DocID Url="tabitano.main.jp/7nisikitenjin.html">0006141064</DocID>
  </DocIDs>
  </OutLink>
</OutLinks>
<InLinks>
<InLink>
 <RawString>清水寺</RawString>
 <DocIDs>
  <DocID Url="http://tabitano.main.jp/7hanatoro2.html">043533898</DocID>
  <DocID Url="http://tabitano.main.jp/7nisikitenjin.html">006141064</DocID>
 </DocIDs>
</InLinks>
<Text Type="default">
 <S Offset="1271" Length="30" is_Japanese_Sentence="1" Id="1">
  <RawString>京都旅楽【たびたの】</RawString>
  <Annotation Scheme="SynGraph">
   <![CDATA[
* 2A <BGH:楽/がく|楽だ/らくだ><文頭><サ変><体言><係:同格連体>...
+ 1D <文節内><係:文節内><文頭><地名><体言><正規化代表表記:京都/きょうと>
京都 きょうと 京都 名詞 6 地名 4 * 0 * 0 "疑似代表表記 代表表記:京都/きょうと" <疑
似代表表記><代表表記:京都/きょうと>...
...
  ]]>
  </Annotation>
 </S>
 <S Offset="1342" Length="80" is_Japanese_Sentence="1" Id="2">
  <RawString>京都観光，修学旅行，世界遺産清水寺はいつも参拝者で賑わっている.
</RawString>
 </S>
  ...
 </Text>
</StandardFormat>
```

図 8: 標準フォーマットの例

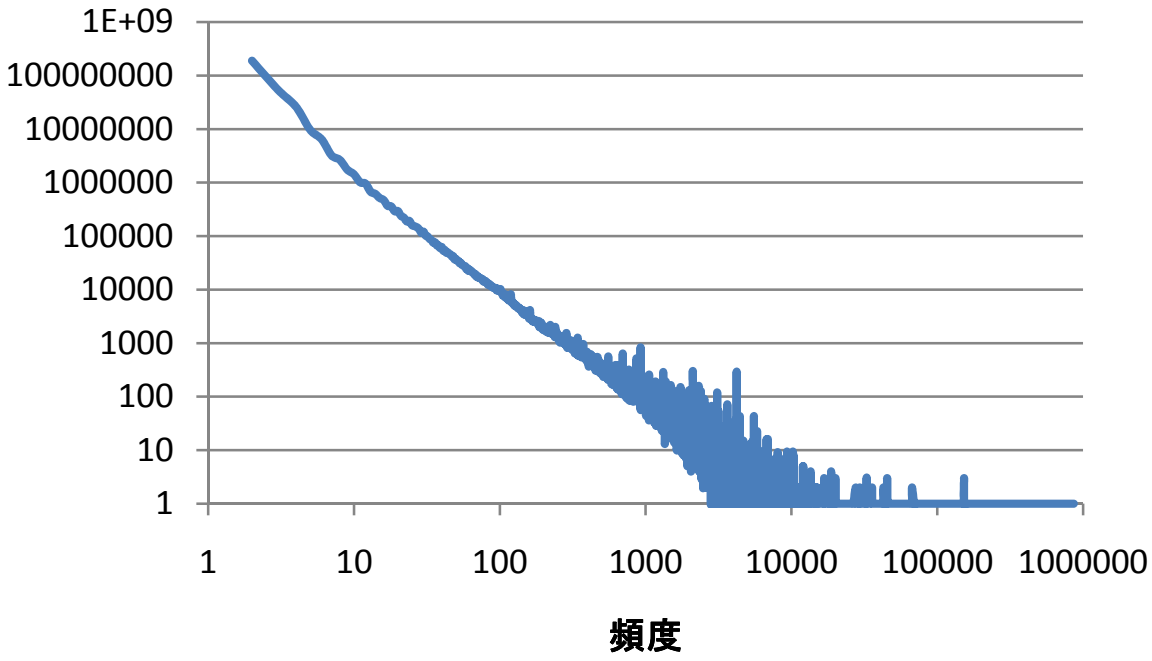

図 9: 頻度の分布

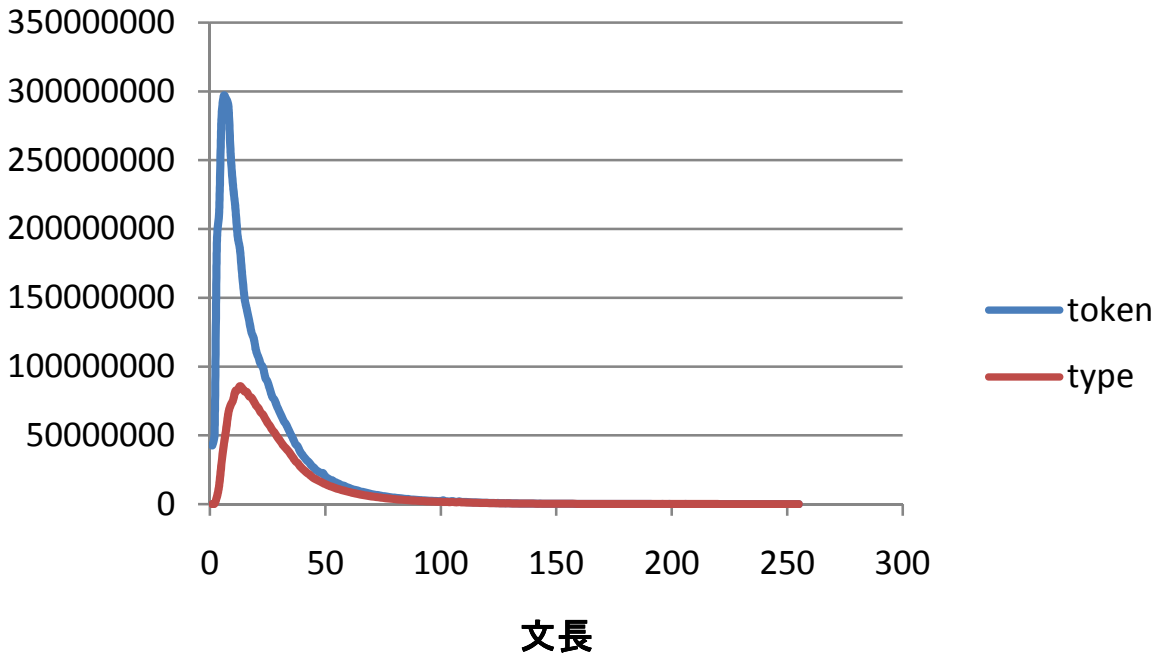

図 10: 文長の頻度分布

表 1: 各長さごとの文とその頻度の例

文長	頻度	文
5	18	ううう〜〜〜
5	10062	プチトマト
5	81200	こんにちは
10	15072	ごちそうさまでした.
10	152790	このページの TOP へ
20	9	ｍｉｘｉの更新情報をまとめて見るツール.
20	100	ルイジ・グロンバール選手，ＧＦ３８に加入
20	1001	プロフィールを作って，メールを送るだけ！
50	3	むしろ「開始直後というのは，夜間開催で生じる影響，異様さを住民が実感するチャンス」と見る向きもある.
50	9	そのまま王選手のペースかと思いきや，山本選手が冷静にカットを攻め好試合となるが，最後は王選手に軍配.
50	100	よくある質問集にあるＱ＆Ａを読んでも問題が解決しない場合は，こちらのページからお問い合わせください.
50	3800	情報検索，読者レビュー，書店売上ランキングや，コミックや書籍の購入をご案内している，本の総合サ...
...		
100	2	帰りは，環状八号を使ってＲ２４６に合流のつもりが，方向はあっているのだけど迷い迷い，川崎駅のターミナルにつっこんだり，よくわからん道クネクネしたりしな溝口あたりでＲ２４６に合流し，帰宅.
100	9	この大辞典は，語源やそのほかの周辺情報（文化的背景など）に関する相当詳しいデータが載っていますが，音声データは付属しておらず，名詞の可算・非可算の区別などの英語初学者が必要とする事項は載っていません.
100	100	橿原市を中心に奈良県中南和の売買や賃貸不動産物件情報と取引，新築注文一戸建住宅の建築や建設・格安リフォーム改装・増改築・改造情報を多数掲載また物件探しや売却，建築・リフォーム時の注意点も紹介，相談無料
100	10462	（いままで，ここでコメントしたとがないときは，コメントを表示する前にこのブログのオーナーの承認が必要になることがあります．承認されるまではコメントは表示されません．そのときはしばらく待ってください.）
...		

第5章　コンテンツ領域抽出

本章ではウェブページからコンテンツ領域を抽出する手法について述べる．ウェブページにはページの主な内容を含む領域 (**コンテンツ領域**) 以外に，有用な情報を含まない領域 (**非コンテンツ領域**) がある．非コンテンツ領域の例としては以下のようなものがある．

広告，メニュー，ツールバー，カウンタ，著作権表示

非コンテンツ領域を含むページの例を図 11 に示す．このページでは左側にはメニュー，ページ中央には広告が含まれており，これらは直接ページの内容とは関連がない非コンテンツ領域である．

関連研究の 2.2 節でも述べたとおり，非コンテンツ領域を検索インデックスから除外することにより速度向上やディスク容量の節約を行なえることや，データマイニングやテキストマイニングにおいて非コンテンツ領域を除外することにより処理時間を短縮することなどを目的として，コンテンツ領域と非コンテンツ領域の識別が行なわれている．

本研究では，コンテンツ領域と非コンテンツ領域の検出結果を類似ページの関連性を測る上で用いる．具体的には，共有している文がどちらのページでもコンテンツ領域であれば 2 つのページの関連性が高く，引用ページまたは盗作ページの可能性が高い．共有している文が一方のページのみでコンテンツ領域であったり，また，いずれのページでも非コンテンツ領域であれば，それらの 2 つのページの関連性は低いと考えられる．

まず，ページをブロックに分割する．ブロックへの分割は DOM 木に基づいて，ページ全体に対する文字数の割合や HTML タグを考慮しながら行なう．その後，分割された各ブロックに対して，コンテンツ領域か非コンテンツ領域かの判断を行なう．この判断は，ブロック内のリンクの割合や含まれる文の最大文字数などを考慮して行なう．

5.1　ブロックへの分割

ページを，コンテンツ領域か非コンテンツ領域の判断を行なう最小単位であるブロックに分割する．以下に手順を示す．

1. ページを DOM 木 (Document Object Model)[1)]に変換する．DOM 木の例を

[1)] http://www.w3.org/DOM/

図 11: 非コンテンツ領域の例 (ページ上部には広告，ページ左部にはメニューがある．)

図 12 に示す．DOM 木への変換には perl モジュール HTML::TreeBuilder[1)] を用いた．

2. DOM 木に基づいてページをブロックに分割する．
   BODY タグを起点として，深さ 1 であるノードのリストを得る．各ノードについて，その部分木に含まれる文字数が全体の文字数の 50%以上であれば，さらに一段深いノードのリストを得る．すべてのノードにおいて，部分木に含まれる文字数が全体の文字数の 50%以下であるようになるまでこの操作を繰り返す．この時点で得られたノードのリストがブロックの候補となる．
3. 後処理
   自分のノードの深さと同じ他のノードと HTML タグが一致している場合，そ

[1)] http://search.cpan.org/~petek/HTML-Tree-3.23/lib/HTML/TreeBuilder.pm

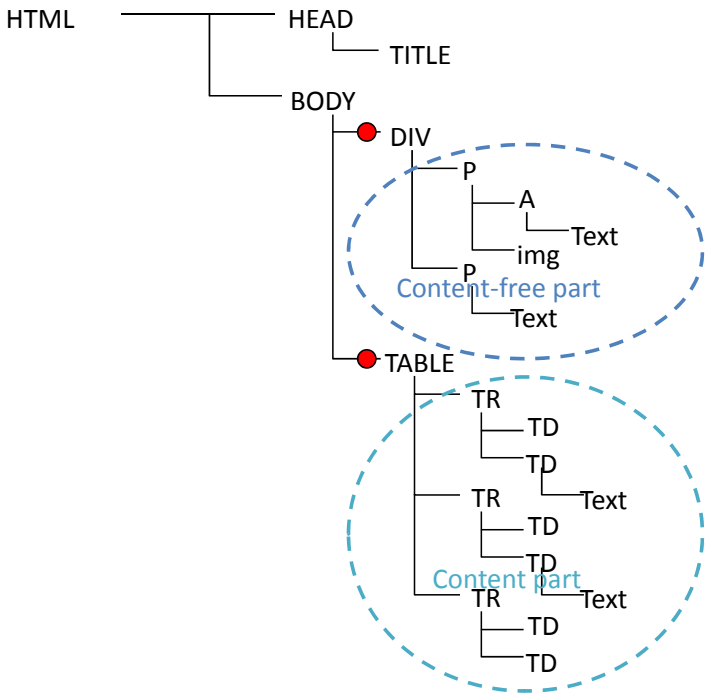

図 12: DOM 木の例

れらのブロックを連結する処理を行なう．これは並列しておかれているノードの場合，それらはすべてコンテンツ領域または非コンテンツ領域になるというヒューリスティックによるものである．例えば，あるノードが<table>タグで，それと同じノードの深さである他のノードも<table>タグであればそれらは一つにまとめたブロックとする．

5.2 コンテンツ領域の抽出

ページをブロックに分割した後，各ブロックに対してコンテンツ領域か非コンテンツ領域かを判断する．以下の条件の OR を非コンテンツ領域にする．

- リンクの割合が閾値以上である．
  - 閾値は予備実験により 50%とした．例を図 13 に示す．この領域はリンクの割合が閾値以上であるため非コンテンツ領域とみなされる．
- 文の最長の文字数が閾値以下である．
  - 閾値は予備実験により 10 文字とした．例を図 14 に示す．この領域は文の最長文字数が閾値以下であるため非コンテンツ領域とみなされる．

ただし，見出しを表わす<h>タグ (<h1>, <h2>, ···, <h6>) に囲まれた文字列は，リンクがはられていることや，文字数が少ないことが多いものの，コンテ

```
     <table width="100%" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td>
            <div class="side_text">
<a href="/XXX.html"><img src="/XXX.gif" align="absmiddle" border="0">情報労連
がウルトラ警備隊の隊員募集</a></td>
            </div>
        </tr>
        <tr>
          <td>
            <div class="side_text">
<a href="/XXX.html"><img src="/XXX.gif" align="absmiddle" border="0">狙われた
赤城徳彦農相</a></td>
            </div>
        </tr>
        <tr>
          <td>
            <div class="side_text">
<a href="/XXX"><img src="/XXX.gif" align="absmiddle" border="0">最薄の液晶テレ
ビが８月に発売</a></td>
            </div>
        </tr>
        <tr>
          ...
        </tr>
      </table>
    </td>
  </tr>
</table>
```

図 13: 非コンテンツ領域の例 (リンクの割合が閾値以上)

ンツ領域である可能性が高いので，上記の条件には含めないこととする．例えば，<h>タグに囲まれた場合として以下のようなものがあり，どちらも上記の条件を満たすが，非コンテンツ領域とみなさない．

- `<div><h1><a href="http://example.com">Economics, Technology & Media</a></h1></div>`
- `<h2>自動車保険</h2>`

```
<table cellpadding="0" cellspacing="0" width="180">
<tr>
<td align="center">

<table cellpadding="1" cellspacing="0" width="100%" align="center">
<tr>
<td class="calTitleCSS">
<img src="mIcon.gif?sdt=2007314223958" align="absmiddle">  カレンダー
</td>
</tr>
<tr>
<td class="calNaiyoTopCSS">
<table cellpadding="1" cellspacing="0" width="100%">
<tr>
<td colspan=7>
<table border=0 width="100%">
  <tr>
<td width="15%"><a href="?curYear=2007&...">&lt;&lt;</a></td>
<td width="70%">2007 年 5 月</td>
<td width="15%">
</td>
  </tr>
</table>
</td>
</tr>
<tr>
<td align="center" width="14%">
<font color="#CC0000">日</font>
</td>
<td align="center" width="14%" class="calNaiyoWeekCSS" >
月
</td>
....
</table>
```

図 14: 非コンテンツ領域の例 (文の最長文字数が閾値以下)

<TR><TD width="204" height="18"></TD>
<TD rowspan="2" valign="top" width="536"><h2><B><FONT size="+0">クルクミン</FONT></B></h2>

カレー粉の黄色のもと、ウコン（ターメリック）に含まれる色素で、ポリフェノールの一種です。強い抗酸化作用を持ちます。
ウコンには、春ウコンと秋ウコンがありますが、クルクミンを多く含んでいるのは秋ウコンです。ウコンは、昔からインドや中国、
日本などで黄疸や肝臓、胃腸の薬として利用されてきました。
クルクミンは、赤血球の酸化を抑え、溶血（赤血球が破壊されること）を
抑制します。この結果、血中コレステロールを減少させて、血液をサラサラにする効果を期待することができます。また、肝臓の機能を助
けアルコールの分解中に出てくるアセトアルデヒトと呼ばれる有害物質の分解を促進したり、胆汁の分泌を促します。胆汁は、膵臓から分
泌される膵液とともに脂肪分解酵素であるリパーゼの働きを助け、脂肪の吸収を促進する働きがあります。このことから、体脂肪や体重を減
らす効果があると言われています。

<FONT size="-1" color="#666666">スポンサード　リンク
</FONT></TD></TR>

コンテンツBLOCK

<TR>
<TD valign=“top” width=“204” height=“737”><FONT size=“-1”>
<A href=“http://www.ykah.com/shoku/”>食と健康に関する辞典TOP</A>
メニュー
（食材の栄養成分と働き）
<A href="yasai.html">野菜類</A>
<A href="kokurui.html">穀類・豆類</A>
<A href="kazitu.html">果実・種実類</A>
<A href="fish.html">魚介類・海藻類</A>
<A href="niku.html">肉類・卵・乳製品</A>
<A href="tyoumiryou.html">調味料・加工品</A>
<A href="herb-spice.html">ハーブ・スパイス</A>
<A href="drinks.html">飲料</A>

<A href="eiyouseibun.html">栄養成分・サプリメント（あ～こ）</A>
<A href="eiyouseibun2.html">栄養成分・サプリメント（さ～の）</A>
<A href="eiyouseibun3.html">栄養成分・サプリメント（は～ろ）</A>
<A href="tenkabutu.html">添加物</A>
<A href="cal.html">カロリー・栄養成分早分り</A>
<A href="sick-yobou.html">病気予防と食べ物</A></FONT></TD>
</TR>

非コンテンツBLOCK

図 15: 非コンテンツ領域抽出の例

第6章　類似ページの自動分類

本章では類似ページの自動分類を行なう．類似ページの分類は 3.1 節で述べたものであり，同一ページ，引用ページなどに分類する．

第 4 章で抽出した低頻度の長い文のうち，第 5 章で抽出したコンテンツ領域に属する文を手がかりとし，それらを共有するページを類似ページとする．

まず，6.1 節ではページ分類に用いるリソースや手がかりについて述べ，その後，6.2 節では得られた類似ページに対してページの自動分類を行なう手法について述べる．

6.1　ページ自動分類のためのリソース・手がかり

ページ自動分類に用いるリソースや手がかりを以下に示す．

- 重複率と包含率
- URL の類似度
- リンク

以下ではそれぞれについて順に説明する．

6.1.1　重複率と包含率

2 ページにおける文の重複もしくは包含している様子を重複率と包含率によって表す．

ページ p に含まれる 20 文字以上の文の集合を $S(p)$ とし，文 S の長さを $|S|$ と表わす時，2 つのページ p_1 と p_2 における重複率 R を以下のように定義する．

$$R(p_1, p_2) = \frac{2 \times |S(p_1) \cap S(p_2)|}{|S(p_1)| + |S(p_2)|} \tag{3}$$

定義より重複率は 0 から 1 までの値をとる．また，分子，分母ともに 20 文字以上の文だけを対象に計算しており，20 文字未満の文は考慮に入れない．

また，2 つのページ p_1 と p_2 の包含率を Simpson 係数によって定義する．Simpson 係数は以下のように計算される．

$$Simpson(p_1, p_2) = \frac{|S(p_1) \cap S(p_2)|}{min(|S(p_1)|, |S(p_2)|)} \tag{4}$$

重複率と同じく 0 から 1 までの値をとり，この係数は，2 つのページの包含関係を捉えるために利用する．

以上の二つの値を利用することにより，ページペアの分類を行なう．

6.1.2 URLの類似度計算

2つのページのURLが類似していれば，それらの2ページは関連がある可能性が高く，また，類似していなければ，関連がない可能性が高いといえる．以下に2つのURLの類似度計算を行なう手法について述べる．

まず，URLをドメイン名部分とディレクトリ部分で分け，ドメイン部分，ディレクトリ部分にわけて処理する．ドメイン名部分とディレクトリ部分は以下のように定義する．

http://ドメイン名部分/ディレクトリ部分

例えば，`http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html`というURLの場合，`nlp.kuee.kyoto-u.ac.jp`がドメイン名，`nl-resource/juman.html`がディレクトリ部分となる．

以下では2つのURLの類似度を計算する手法について述べる．類似度計算において，ドメイン名の処理，ディレクトリ名の処理をわけて考える．

1. 文字の正規化・削除
   - ドメイン名部分
     - 先頭にある「www」を含む文字列や「jp」を削除する
     - 末尾にある「net」,「com」,「biz」,「org」,「co.jp」,「ac.jp」,「ne.jp」,「or.jp」を削除する
   - ディレクトリ部分
     - 先頭の「~」や「%7E」を削除する
     - 末尾のhtmlをhtmに置換する
     - 末尾の文字列がindexを含んでいる場合，それを削除する
2. 文字列の分割
   - ドメイン名部分の処理
     - 「.」または「-」で区切る
   - ディレクトリ部分の処理
     - 「/」で区切る
3. 類似度の計算

   ドメイン名部分とディレクトリ部分の類似度を別々に計算する．それぞれにおいて，Simpson係数で類似度を計算する．そして，それらの平均をとることにより，2つのURLの類似度とする．

表 2: URL の類似度計算の例

類似度	URL
0.50	http://www.gipc.kanazawa-u.ac.jp/whatsnew/host/kh23102.htm http://133.28.23.100/whatsnew/host/kh23102.htm
1.00	http://www.softic.or.jp/lib/cases/Nikkei_v_Comline.html http://mail.softic.or.jp/lib/cases/Nikkei_v_Comline.html
0.38	http://www.dd.iij4u.or.jp/~yamano/dokyo/saigoku/saigoku_17.htm http://yamanokanata.net/dokyo/saigoku/saigoku_17.htm
1.00	http://www.fdev.ce.hiroshima-cu.ac.jp/~terauchi/SEIKEI.HTM http://va620v.fdev.ce.hiroshima-cu.ac.jp/~terauchi/SEIKEI.HTM
1.00	http://www.keiju.co.jp/data2/tushinbo.htm http://keijyulion.keiju.co.jp/data2/tushinbo.htm

表 2 に URL の類似度計算を行なった例を示す.

6.1.3 リンク

2 ページ間にリンクがあればそれらは関連があるといえる．また，リンクには方向性があり，例えば，ページ A からページ B へのリンクがあれば，ページ A がページ B の内容を参照している．また，検索エンジンがページランクを用いてランキングを行なっていることから，リンクをはることによってランキングをあげようとしている場合もある．

4.1 節で述べた標準フォーマットには OutLink と InLink の情報がうめこまれており，この情報を利用する．上記で述べたとおり，リンクには方向性があるので，それも利用する．

6.2 自動分類

6.1 節で述べたリソースを使って，同一文字列を含むページペアを，3.1 節で述べた分類 (同一, 類似, 引用, 盗作ページなど) に分けることを行なう．ページの自動分類の概要を図 16 に示す.

まず，2 ページ間の重複率 R と包含率 $Simpson$ を計算し，その値に応じて同一，包含，部分共有に分類する．まず，$R > 0.6$ であれば同一とする．R が 1 の時のみではなく，0.6 以上となっているのは以下の理由による．R の分母はページに含まれる 20 文字以上の文が対象であるが，R の分子は Web 全体での頻度が 10 回以下のものしか対象とはならず，ページに Web 全体での頻度が 10 回より

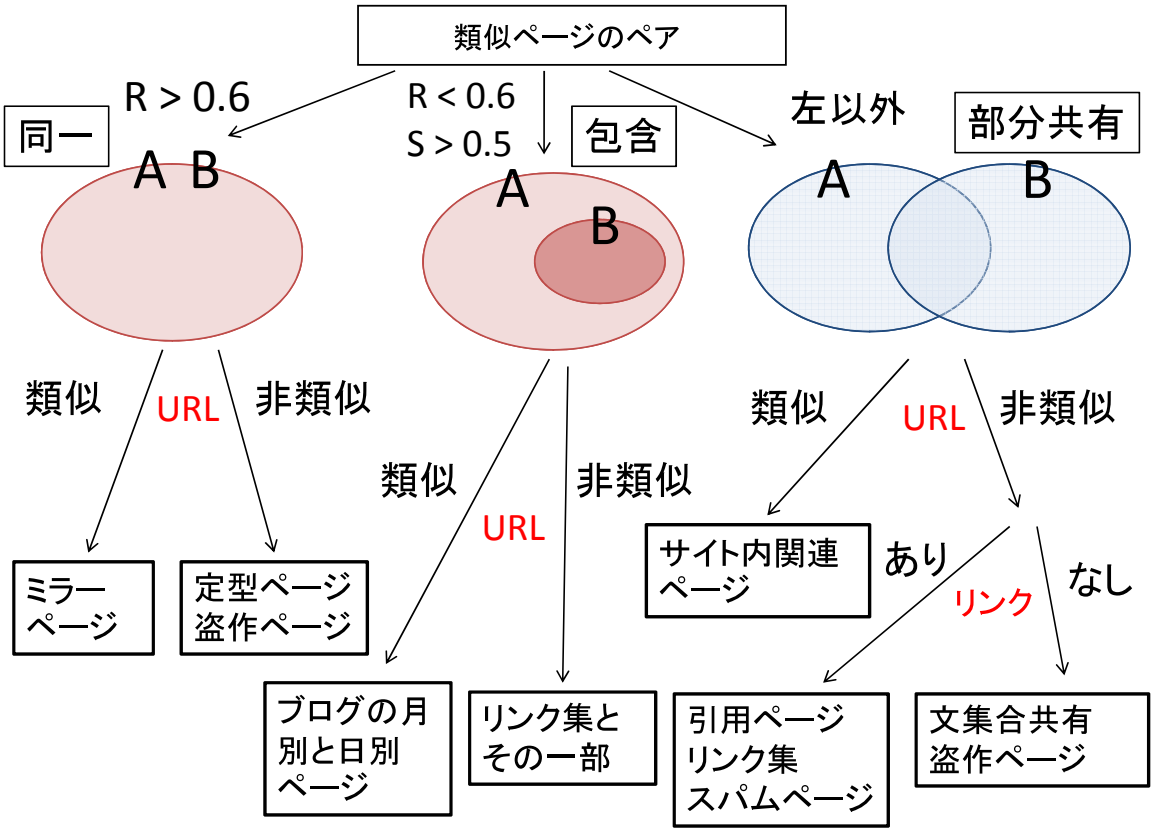

図 16: ページの自動分類の概要

大きい文があれば，その文を共有しても分子にカウントされない．よって，全く同じページであっても必ず R が 1 となるわけではない．予備実験により，R が 0.6 より大きければ，その 2 ページに含まれる Web 全体での頻度が 10 回より大きい文をすべて共有することが多く，その値に設定した．

そして，$R < 0.6$ かつ $Simpson > 0.5$ であれば，包含とする．条件が $Simpson = 1$ とならないのは上記の理由と同じものである．そして，これら以外を部分共有とする．

その後，同一，包含，部分共有の各カテゴリにおいて，URL の類似度，リンクなどの情報でさらに分類する．同一ページの場合，URL が類似していればミラーページに，類似していなければ定型ページ，盗作ページに分類する．ここで URL の類似は 6.2 節で説明した URL の類似度が正であることとする．また，包含ページの場合，URL が類似していればブログの月別，日別ページに，類似していなければリンク集とその一部とする．部分共有ページはまず，URL が類似していればサイト内関連ページ，URL が類似していない場合，2 ページ間に

リンクがあれば引用ページ，リンク集，スパムページに，リンクがなければ文集合共有ページ，または，盗作ページとする．

次節では，この分類に基づいて自動分類を行なった結果を示す．

第7章　実験と結果

本章では，第6章で述べた自動分類について実験結果を示す．まず，7.1節で類似ページ検出に関する統計を示す．そして，7.2節でページの分類結果とページ例について述べる．

7.1　類似ページ検出

第4章で得られた低頻度の長い文のうち，第5章で検出したコンテンツ領域に属していた文の異なり数は約3億文であった．これらの文を1文以上共有しているページペアを類似ページとみなしたところ，約1.8億ページペアの類似ページが得られた．この1.8億ページペアに現われたページ数は約5,600万ページであった．

7.2　自動分類結果と例

上記で得られたページペアからランダムに1,000ページペアを選び，それに対して，第6章 で述べた分類を行なった．各カテゴリのページ数ならびに分類精度を表3に示す．

まず，ページペア数の分布は，同一関係が約1割，包含関係も約1割，残り8割弱が部分共有関係であった．また，分類精度に関してはどのカテゴリも80%以上の精度が得られた．

この結果発見された類似ページの例を以下にあげる．まず，ミラーページの例を図17にあげる．また，重複率計算にはコンテンツ領域内の文しか考慮していないので，図18の例のように，メニューバーなどの非コンテンツ領域のみが異なるページペアもある．次に，ブログの月別ページと日別ページを図19に示す．

引用ページの例を図20に示す．左側のページから右側のページにリンクがある．また，図21と図22にいろいろなページをつなぎあわせただけのページの例を示す．これはスパムページであり，ページ内の記事に関連性がないものである．図23に盗作ページの例を示す．右側のページが左側のページを盗作している．右側のページから左側のページへのリンクはなく，また，少し改変してある．

表 3: ページ分類数

カテゴリ			数 (分類精度)	合計
同一	URL 類似		107 (87%)	108
	URL 非類似		1 (100%)	
包含	URL 類似		100 (93%)	107
	URL 非類似		7 (86%)	
部分共有	URL 類似		498 (83%)	
	URL 非類似	リンクあり	9 (89%)	785
		リンクなし	278 (90%)	
				1,000

図 17: 同一ページ (ミラーページ)

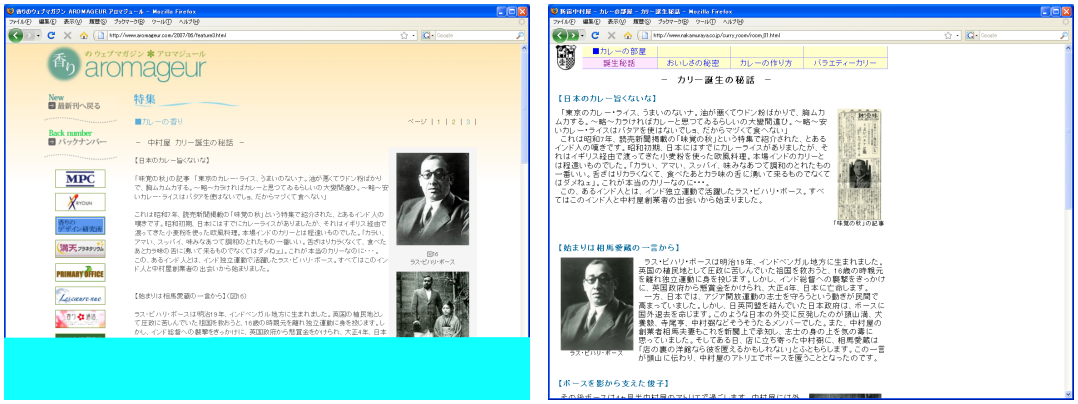

図 18: 同一ページ (コンテンツ領域が同一)

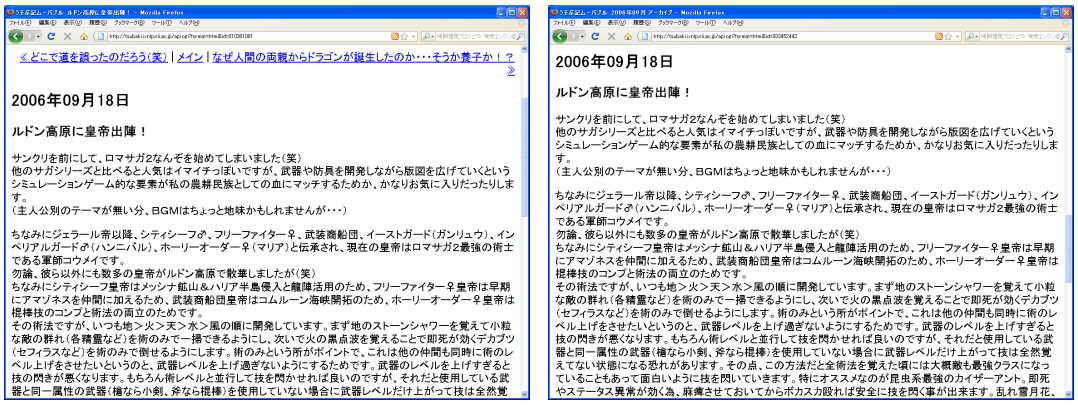

図 19: ブログの月別ページと日別ページ

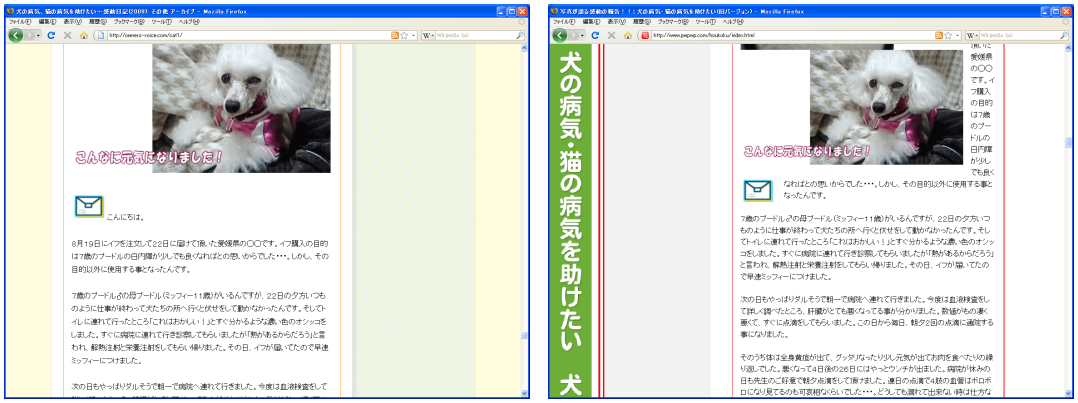

図 20: 引用ページの例

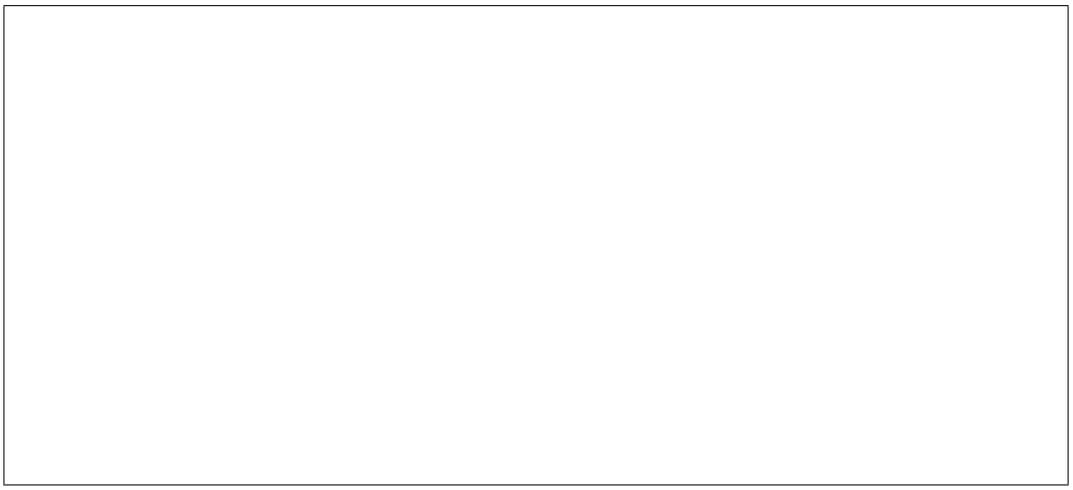

図 21: いろいろなページをつなぎあわせただけのページの例

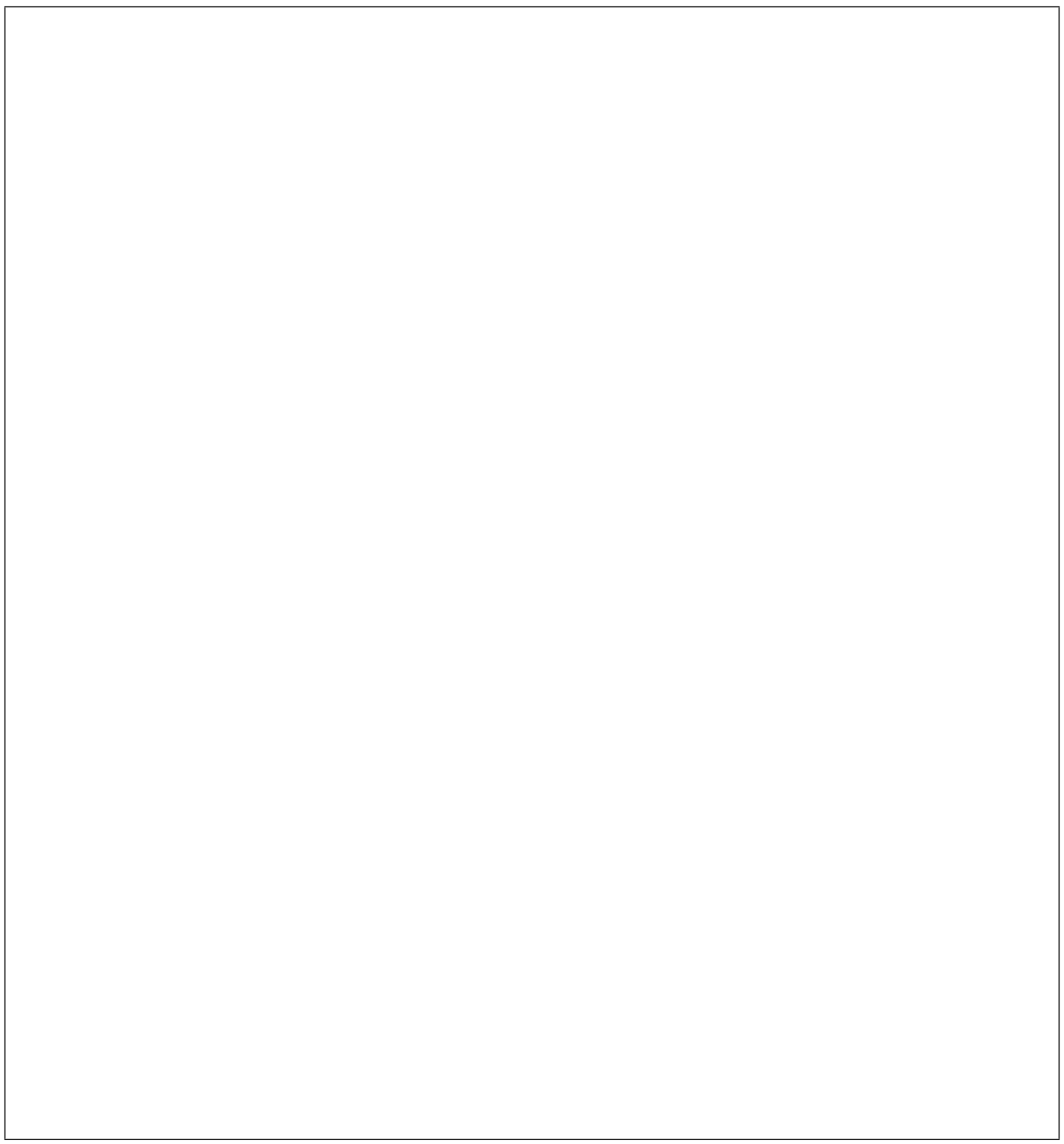

図 22: いろいろなページから引用しているページ例 (一番上のページは下の 4 つのページを引用している)

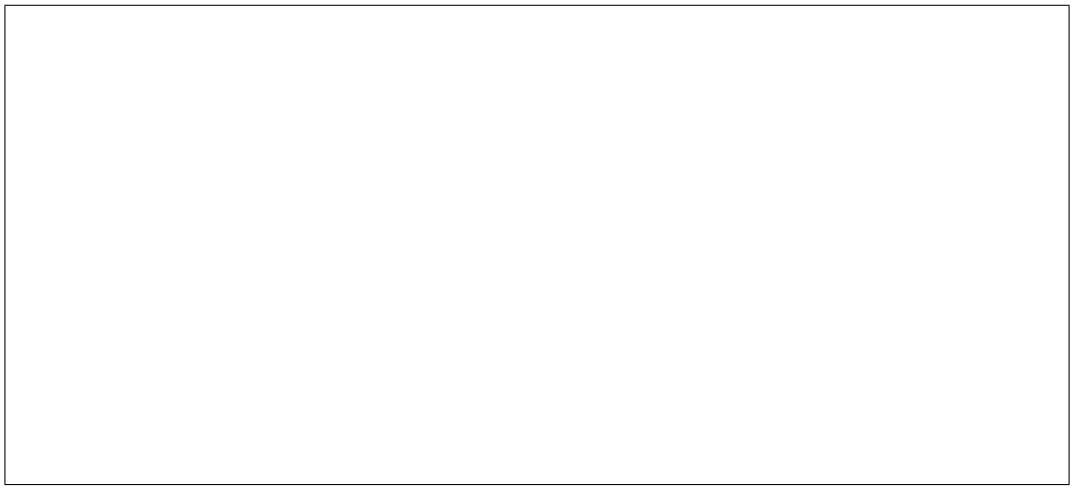

図 23: 盗作ページの例

第8章　おわりに

本研究では1億ページという大規模なウェブコレクションを対象として，類似ページ検出を行なった．まず，類似ページを，文字列を共有する2つのページと定義し，ミラーページなどの同一ページ，引用ページ，盗作ページなどに分類した．

まず，各ページから長い低頻度の文を抽出し，それらを共有するページペアを類似ページとみなした．そして，重複率，リンク，URLの類似度などの様々な情報を用いて，類似ページを同一ページ，引用ページ，盗作ページ，スパムページなどに自動分類した．実験を行なったところ，単純なURLの正規化ではわからないミラーページや，引用ページ，様々なサイトから記事をはりあわせたようなスパムページを発見することができた．

今後の課題としては，類似ページの検出，分類精度をあげるとともに，本研究での類似ページ検出結果を検索エンジン基盤TSUBAKIに反映させることがあげられる．

- 同一ページの除去
  現在のTSUBAKIでは検索結果を表示する際にURLやタイトル，検索スコアなどの情報に基づいて動的に類似ページ検出を行なっているが，これは検索スピードの低下を招いている．本研究での成果を導入し，インデックス構築時に同一ページを除外することにより，インデックス構築の時間，検索スピードともに改善することができると考えられる．また，現在動的に行なっている類似ページ検索では見つからないような同一ページを発見できているので，より同一ページのマージが行なえる．
- スパムページの除去
  現在のTSUBAKIの検索結果にスパムページが含まれていることがあり、これは検索結果の文書の質を低下させている。本研究で検出されたスパムページをインデックス構築時に除外することにより、文書の質を向上させる予定である。
- 非コンテンツ領域をインデックスから除去
  本研究ではコンテンツ領域検出を類似ページ検出タスクに利用したが、関連研究で行なわれているようにインデックス構築時に非コンテンツ領域のテキストをインデックスに含めないことが考えられ、今後行なう予定である。

本研究では同一文を手がかりに類似ページの検出を行なったが，特に盗作ページの場合，助詞などを少し改変されることが多く，そのような場合，本研究の手法では検出することができない．この問題に関しては類似文字列検出プログラムを用いることにより対処する予定である．

謝辞

本研究を進めるにあたり，終始熱心にご指導くださいました黒橋禎夫教授に心からお礼申し上げます.

また，数々の論文執筆から日々の研究姿勢に至るまで，あらゆる面から終始懇切丁寧に御指導，御相談下さいました柴田知秀助教に，心から感謝いたします.

本研究に関して数々な側面から助言をいただきました，同課程 2 回生の小谷通隆氏，玉城伸仁氏，原島純氏に深く感謝いたします.

また，研究室での日常生活の様々な面で面倒を見てくださいました芦原裕子秘書にとりわけ感謝いたします.

最後に，本研究に関して援助して下さった，黒橋研究室のみなさんに感謝いたします.

参考文献

[1] Manning, C. D., Raghavan, P. and Schütze, H.: *Introduction to Information Retrieval*, Cambridge University Press (2008).

[2] 小野拓史, 豊田正史, 喜連川優: リンク解析を用いたウェブ上のスパム発見手法に関する一考察, 電子情報通信学会第 17 回データ工学ワークショップ 第 4 回日本データベース学会年次大会 (DEWS2006), Vol. 3B, No. o2 (2006).

[3] 佐藤有記, 宇津呂武仁, 福原知宏, 河田容英, 村上嘉陽, 中川裕志, 神門典子: キーワードの特性を利用したスパムブログの収集と分析, 第 22 回人工知能学会全国大会, Vol. 3E2, No. 1 (2008).

[4] Lyon, C., Malcolm, J. and Dickerson, B.: Detecting Short Passages of Similar Text in Large Document Collections, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp. 118–125 (2001).

[5] Manku, G. S., Jain, A. and Sarma, A. D.: Detecting Near Duplicates for Web Crawling, *Proceedings of the 16th International Conference on World Wide Web*, pp. 141–150 (2007).

[6] Broder, A. Z., Glassman, S. C., Manasse, M. S. and Zweig, G.: Syntactic clustering of the Web, *Proceedings of the 6th International Conference on World Wide Web*, pp. 1157–1166 (1997).

[7] Charikar, M. S.: Similarity estimation techniques from rounding algorithms, *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380–388 (2002).

[8] Henzinger, M.: Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms, *Proceedings of 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrival*, pp. 284–291 (2006).

[9] Rabin, M. O.: Fingerprinting by Random polynomials, Technical report, Center for Research in Computing Technology (1981).

[10] Broder, A. Z.: Some applications of Rabin's fingerprinting method, *Sequences II: Methods in Communications, Security, and Computer Science*,

pp. 143–152 (1993).

[11] BarYossef, Z., Keidar, I. and Schonfeld, U.: Do Not Crawl in the DUST: Different URLs with Similar Text, *Proceedings of WWW2007*, pp. 111–120 (2007).

[12] Xiao, C., Wang, W., Lin, X. and Yu, J. X.: Efficient similarity joins for near duplicate detection, *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 131–140 (2008).

[13] Hoad, T. C. and Zobel, J.: Methods for Identifying Versioned and Plagiarised Documents, *Journal of the American Society for Information Science and Technology*, Vol. 54, pp. 203–215 (2002).

[14] Lin, S.-H. and Ho, J.-M.: Discovering Informative Content Blocks from Web Documents, *In Proceedings of ACM SIGKDD'02*, pp. 588–593 (2002).

[15] Debnath, I., Mitra, P. and Giles, C. L.: Identifying content blocks from web documents, *In Proceedings of the 15th ISMIS 2005 Conference*, pp. 285–293 (2005).

[16] 中村達也, 白井清昭: ウェブページにおける非コンテンツ領域の検出, 第 13 回言語処理学会年次大会, pp. 234–237 (2007).

[17] Shinzato, K., Shibata, T., Kawahara, D., Hashimoto, C. and Kurohashi, S.: TSUBAKI: An Open Search Engine Infrastructure for Developing New Information Access Methodology, *Proceedings of Third International Joint Conference on Natural Language Processing (IJCNLP2008)*, pp. 189–196 (2008).

[18] Sekiguchi, Y. and Yamamoto, K.: Improving Quality of the Web Corpus, *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pp. 201–206 (2004).

[19] Kawahara, D. and Kurohashi, S.: Case Frame Compilation from the Web using High-Performance Computing, *Proceedings of LREC-06* (2006).

[20] 新里圭司, 橋本力, 河原大輔, 黒橋禎夫: 自然言語処理基盤としてのウェブ文書標準フォーマットの提案, 言語処理学会第 13 回年次大会論文集, pp. 602–605 (2007).

[21] Shinzato, K., Kawahara, D., Hashimoto, C. and Kurohashi, S.: A Large-Scale Web Data Collection as a Natural Language Processing Infrastruc-

ture, *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC08)* (2008).

[22] Shibata, T., Odani, M., Harashima, J., Oonishi, T. and Kurohashi, S.: SYNGRAPH: A Flexible Matching Method based on Synonymous Expression Extraction from an Ordinary Dictionary and a Web Corpus, *Proceedings of Third International Joint Conference on Natural Language Processing (IJCNLP2008)*, pp. 787–792 (2008).