DBSJ Journal, Vol.8, No.1
June 2009

順序保存ダイジェスト法による Web ページ間の部分複製検出

Sequence Preserving Digest: A New Method for Efficiently Detecting Partially Duplicated Web Pages

櫻井 俊之♥　松尾 義博♦
菊井 玄一郎♠

Toshiyuki SAKURAI　Yoshihiro MATSUO
Genichiro KIKUI

**Web ページの一部を複製して生成されるスパムブログ等の「部分複製コンテンツ」が，検索エンジンのユーザ満足度の低下などの問題を引き起こしている．
そこで我々は，部分複製を効率的に検出する手法を提案する．提案手法は Web ページを「セグメント」と呼ばれる単位に分割し，各セグメントをハッシュ値に変換して連結し，Suffix Array に格納することにより，部分複製の高速な検索を可能にするものである（順序保存ダイジェスト法）．提案手法に対して，検出速度，精度，再現率の評価実験を行った結果，有効性を確認した．**

We propose a new method for efficiently detecting partially duplicated web pages or weblogs, which often case adverse effects on search results. Our method converts each web page into a digest, which is a concatenated string of hash values, each of which derived from a segment, currently a sentence, of the original web page. Since the digest preserves the order of segments in the original web page, partial match of two digests means partial match of their original documents. Our method stores digests into suffix arrays in order to efficiently detect partial match. Experimental results show our method detects partial match correctly while still keeping space and time efficiency.

1. はじめに

近年，CMS（Content Management System）やブログサービスの発展に伴い，誰でも気軽に情報発信を行うことが可能になった．また，クリック報酬型広告やアフィリエイトプログラムの登場により，ブログから利益を得ることが可能になった．これらの要因により，特定サイトへの誘導や成果報酬を目的としたスパムブログが大量発生するようになり，検索エンジンのユーザ満足度を低下させるようになった．

スパムブログは検索結果上位に自身を表示させるために，Wikipediaやニュースサイトの記事，ECサイトの商品紹介記事，プレスリリース，歌詞等の検索ランキング上位のキーワードを含むWebページの文章を複製して生成される複製コンテンツであることが多い．そのため，複製コンテンツを効率良く検出し，それらを排除する技術が求められている．

そこで，本論文では従来技術では検出が難しかった部分複製を検出可能な手法を提案し，提案手法をシステム化し，その評価結果を述べる．

以下2章では複製コンテンツ検出技術の課題について述べ，3章では関連研究について述べる．4章では提案手法の概要について述べ，5章では提案手法を用いたシステムの構成について述べ，6章では5章で述べたシステムを用いた評価実験の結果について述べる．

2. 複製コンテンツ検出の課題

本章では，複製コンテンツの種別，検出技術に対する課題について述べる．

2.1 複製コンテンツの種別

複製コンテンツは図1に示すとおり，全体複製，包含関係，部分複製の3種類に分類される

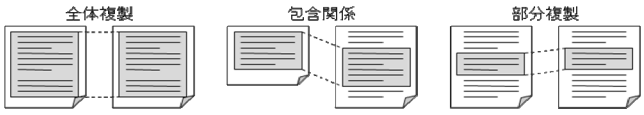

図 1　複製コンテンツの種別

Fig.1 Type of Duplicated Contents

全体複製とは，Webページ全体が他のWebページからの複製であるものを指す．包含関係とは，あるWebページ全体を別のWebページが包含しているものを指す．部分複製とは，あるWebページの一部が，別のWebページの一部として複製されているものを指す．

全体複製や包含関係の検出に関しては3章で述べるように広く研究が行われており，検出手法がいくつか提案されているが，部分複製に関しては従来手法では検出が困難であるため，部分複製の検出が可能な手法を検討する必要がある．

2.2 文書規模，検出速度の問題

スパムブログ等の複製コンテンツは前述の通りWikipediaやニュースサイトの記事等を複製して記事を生成している．そのため，これらの複製コンテンツを検出するためには，複製元となりやすい文書（原典文書）を収集し，複製検出のための原典として利用すればよい．

しかし，原典文書はWikipediaだけでも55万記事（2009年1月8日現在）存在し，Wikipediaの過去の版や各種ニュースサイト，各種ECサイト，プレスリリースまでを網羅しようとすると数千万記事にまで膨らむ．

そのため，複製コンテンツの検出を効率良く行うためには，何らかの方法でデータ量を圧縮する必要がある．

また，検出対象となるブログも，総務省情報通信政策研究所が2008年7月に公表した調査報告書[1]によると，毎月4000万～5000万記事も増加しているため，検出速度の高速化も重要な課題である．

♥ 正会員　NTT サイバースペース研究所
sakurai.toshiyuki@lab.ntt.co.jp
♦ NTT サイバースペース研究所
matsuo.yoshihiro@lab.ntt.co.jp
♠ NTT サイバースペース研究所
kikui.genichiro@lab.ntt.co.jp

3. 関連研究

複製コンテンツ検出に関する方法としては，類似文書検出技術を用いる方法と検索エンジンを用いる方法の2つがある．

類似文書検出技術には，Charikarの手法，Broderらの手法，Bayordoらの手法，Brinらの手法がある．

Charikarは文書毎にsimhashと呼ばれる特殊なハッシュ値を計算し，文書間の類似度をsimhashのハミング距離から算出する手法を提案した[2]．この手法におけるsimhashのハミング距離は，各文書を単語（頻度）等を次元とするベクトル空間に写像した場合の，ベクトル間のコサイン距離を近似しているため，文書全体の類似性を判定することはできるが，包含関係や部分複製を検出することができない．

Broderらは文書から得られたN-gramを基にしたShingleと呼ばれるものを定義し，2つの文書間の類似度をJaccard係数により算出する手法を提案した[3]．この手法では，N-gramが一定割合以上一致した文書の組み合わせを類似と判定するため，全体複製だけでなく，一部の包含関係も検出可能であるが，文書全体に対する複製部分の割合が少ない包含関係や部分複製を検出することができない．

Bayordoらは事前に閾値を設定し，閾値以上の文書ペアのみの類似度を計算する手法を提案した[4]．この手法は，類似度の高そうな候補を絞り込むことによって計算量を削減する手法であるが，候補を絞り込む処理における類似度関数として，コサイン類似度やJaccard係数等を用いるため，CharikarやBroderらの手法と同様の問題がある．

Brinらは共通する文の割合により類似文書を検出するCOPSというシステムを提案した[5]．COPSでは，事前に原典文書を文単位に分割しハッシュ化して登録しておき，検出対象文書も同様にハッシュ化し，共通するハッシュの数が閾値を超えると類似文書と判定する．このシステムでは，定型文など偶然一致しやすい文を多く含む文書が入力されると，複製と誤検出してしまう問題がある．

以上をまとめると，既存の類似文書検出技術は文書中に出現する単語，N-gram，文等の頻度分布が近いものを検出する手法であるため，文書全体に対する複製部分の割合が小さい場合は，類似度が低下してしまい，検出することができない．また，一致部分の出現位置や順序を考慮しないため，2つの文書間の単語，N-gram，文等の頻度分布が類似していれば，複製でなくても類似度が高くなってしまい，複製でないものまで複製と誤検出してしまう問題がある．

複製を確実に検出するためには，単語やN-gram，文等の出現順序を考慮した上で，原典文書との共通部分がある程度連続しているものを検出する仕組みが必要である．

一方，検索エンジンを用いる手法に関しては，田代らの手法がある．田代らは入力文書から文節単位のN-gramを生成し，各N-gramをクエリとして，複数の検索エンジンに入力し，各N-gramにおける検索結果の和集合を複製コンテンツ候補とし，その候補に対して複製コンテンツかどうかの判定を行う手法となっている[6]．この手法では，文節の出現順序を考慮し，入力文書と複製コンテンツ候補の共通する連続文節長の最大値を基に複製判定を行うようになっており，文書全体に対する一致部分の割合が小さくても検出することが可能なため，全体複製，包含関係，部分複製の3種類の複製コンテンツを検出可能な手法であるが，検索結果上位にのみを複製コンテンツ候補としているため，網羅性の点で問題がある．また，検索エンジンに複数回検索キーワードを入力する手法であるため，検出速度の面でも向いていないと考えられる．

4. 提案手法の概要

本章では，従来技術の問題点を解決し，Web規模にも適用可能な複製コンテンツ検出手法について述べる．

4.1 データ圧縮方法

3章で述べたとおり，全体複製，包含関係，部分複製を全て検出可能にするためには，単語，文節，文等の出現順序情報を意識した検出手法が必要である．そのため，これらの出現順序情報を保持可能なデータ圧縮手法が必要とされる．

出現順序を保持する単位として，単語，N-gram，文等が考えられるが，できる限り大きい単位で出現順序情報を保持したほうがデータ量の大幅な圧縮が可能である．

複製は日本語としての意味をなすように，一般的には文もしくは文に準じた単位で行われる．そのため，文もしくは文に準じた単位でデータ圧縮を行えば，高いデータ圧縮率を実現しつつ，複製を検出することが可能である．

そこで我々は，上記の性質を利用したデータ圧縮方法を提案する．具体的には，まず，文もしくは文に準じた単位の切れ目となりやすい任意のセパレータ（句点，疑問符，感嘆符，HTMLタグ等）を目印にして文書を分割し，これをセグメントとする．次に，セグメント毎にハッシュ関数を用いてフィンガープリントを生成する．フィンガープリントをセグメントの出現順序を保持したまま連結し，これをダイジェストとする．Brinらのシステムとは，セグメントを連結して取り扱うという点で異なる．

上記により，セグメントの出現順序は，フィンガープリントの出現順序で表現されるため，データ量を圧縮しつつも，セグメントの出現順序は保持されている．これにより複製の検出は，原典文書と複製検出対象文書のダイジェストを比較し，ダイジェスト間の共通部分を検出する問題となる．

4.2 部分文字列照合方法

ダイジェスト間の共通部分を検出する問題は，文字列間の共通部分を検出する問題と等価である．文字列間の共通部分を総当たりで見つけようとすると，組み合わせ数が爆発してしまい現実的ではない．

そこで我々は文字列の一致部分を高速に検索することが可能なデータ構造であるSuffix Array[7]を用いることとした．Suffix Arrayは事前に検索用インデックスを作成しておく必要があるが，その計算量は$O(n\log(n))$であり，あらゆる部分文字列の検索が可能な，シンプルなアルゴリズムである．

5. 部分複製検出システム

本章では，前章の手法を用いた部分複製検出システムについて述べる．図2に本システムの機能ブロックの構成を示す．

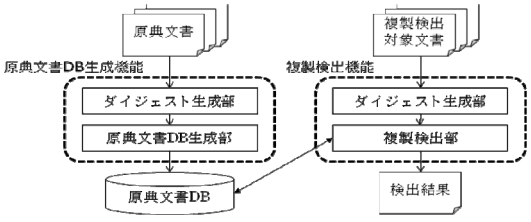

図 2　部分複製検出システムの機能ブロック図

Fig.2　Function Block Diagram

本システムは，事前に原典文書の各文書に対してダイジェストを生成し， Suffix Array の検索用インデックス（原典文書 DB）に格納する機能（原典文書 DB 生成機能）と，入力された複製検出対象文書のダイジェストを生成して，原典文書からの複製が含まれていないかを検出する機能（複製検出機能）の 2 つの機能からなる．

この 2 つの機能のうち，ダイジェストを生成する部分は共通であるため，システム全体は以下の 3 つの機能部からなる．

- ダイジェスト生成部
- 原典文書 DB 生成部
- 複製検出部

以下で，各機能部の詳細について述べる．

5.1 ダイジェスト生成部

図 3 にダイジェスト生成部の処理の流れを示す．

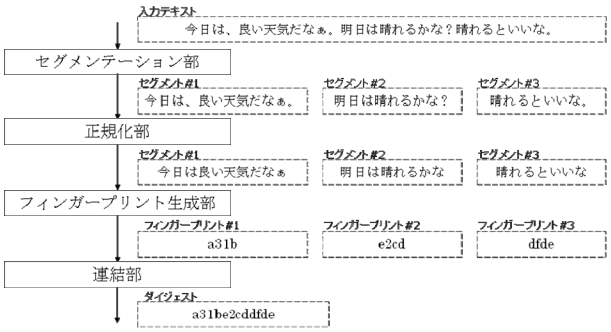

図 3　ダイジェスト生成部

Fig.3　Method of Digest Generation

まず，セグメンテーション部で入力されたテキストをセグメントに分割する．セグメントの境界はあらかじめ設定されたセパレータ（図 3 では，”。”，”？”）により決定される．

次に，正規化部で全角半角正規化，記号類の除去などの各種正規化を行う．複製の中には，記号の追加や全角半角の変更など，複製元文章に微妙な差異を付けたものが存在するため，微妙な差異を吸収するために上記の処理を行っている．正規化部では，セグメント長が短いものの除去（足切り）も行う．これは，「おはよう」，「おやすみ」など，セグメント長が短いものは，偶然の一致である確率が高く，複製でない場合が多いためである．

そして，フィンガープリント生成部でハッシュ関数を用いて，セグメントに対するフィンガープリントを計算する．本システムではハッシュ関数として Rabin　Fingerprint[8]の 32bit ハッシュを採用した．

最後に連結部でフィンガープリントを連結してダイジェストを生成する．

5.2 原典文書 DB 生成部

ダイジェスト生成部から出力されたダイジェストと文書 ID の組み合わせをストアし，インデックスをフィンガープリント長単位とした Suffix Array を構築する．

5.3 複製検出部

図 4 に複製検出部の処理の流れを示す．

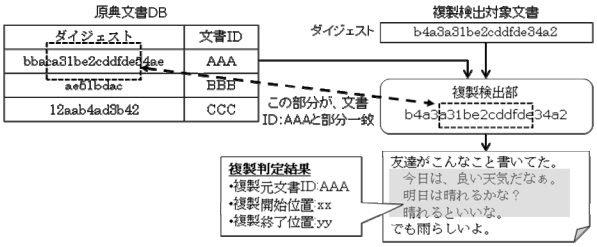

図 4　複製検出部

Fig.4　Method of Duplicate Detection

複製検出対象文書が入力されると，まず，ダイジェスト生成部を用いてダイジェストを生成する．次に，ダイジェストをクエリとして，原典文書 DB に対して，前方最長一致のエントリを検索する．もし，前方最長一致のエントリが見つかった場合は，一致長をカウントし，事前に設定した長さ（最短一致長）より長く一致していれば複製とみなし，複製部分を確定する．

6. 評価実験

本章では，実機を用いた実験結果について述べる．本評価実験では，提案手法の原典文書規模と照合速度の評価，精度の評価，再現率の評価の 3 つを行った．

なお，本評価実験では，セグメント長 5 文字未満を足切りし，最短一致長は 3 セグメントとした．

以降で，各実験の詳細について述べる．

6.1 原典文書規模と照合速度

原典文書に格納可能な文書規模を調べるため，文書数を変化させながら原典文書 DB を作成する実験を行った．

実験には，CPU: Intel Xeon E3070，メモリ: 4GB の計算機を用い，原典文書として，Wikipedia　50 万記事（http://download.wikimedia.org/jawiki/　より 20080726 版をダウンロード）とブログ 2000 万記事を用いた．

また，原典文書 DB のサイズが大きくなるに従って，照合速度がどれほど低下するかを確認するため，文書規模毎に照合速度の計測も合わせて行った．

実験結果を図 5 に示す．

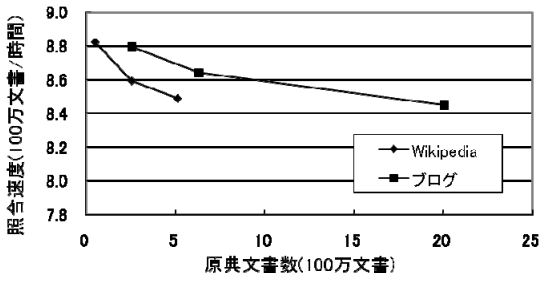

図 5　文書規模と照合速度

Fig.5　Performance vs. Scale

実験結果より，4GB 程度のメモリであっても約 2000 万文書を格納可能であることがわかった．

原典文書を分割し，それぞれを別の計算機に格納し，分散処理を行うことにより，さらに大規模な原典文書を扱うことも可能であると考えられる．しかし，具体的な検討はまだ未実施のため，今後検討する余地がある．

また，照合速度に関しては，2000 万文書をインデックスしていた場合でも，840 万文書/時間の速度での照合が可能であった．

2.2 節で述べたとおり，文献[7]の調査によると，ブログの新着記事は一ヶ月に 4000 万から 5000 万記事あるため，本システムで十分処理が可能であると考えられる．

6.2 精度

ブログにどれだけ Wikipedia からの複製が含まれているかを判定し，判定結果がどれだけ正しいかを確認することにより，本システムの精度を評価した．具体的には，原典文書として Wikipedia 50 万記事，複製検出対象としてブログ 120 万記事を用い，本システムにより複製検出を行い，複製と判定された 378 記事に対して Wikipedia からの複製が含まれていないかを目視にて確認した．

評価を行った結果，378 記事中 374 記事が Wikipedia からの複製という結果を得た．これにより精度は 98%となった．

不正解の内訳は以下の通りである．

（1） Wikipedia 自体が複製
（2） 文章が Wikipedia と偶然一致

（1）に関しては，複製元と複製先の判別は人間でさえも難しいため，原典文書に混入した複製コンテンツを除去するのは難しい課題である．

（2）に関しては，定型文，固有名詞等の羅列が偶然一致してしまったことが原因である．足切り条件を今回の実験値より大きくすることにより，偶然一致する可能性を低下させることは可能である．しかし，検出漏れが増大する可能性があるため，文字列の情報量を考慮する等，検討の余地がある．

6.3 再現率

あらかじめ目視により作成した正解データをどれだけ再現することができるか評価を行った．具体的には，ブログ 10204 記事に対して，目視により複製が含まれているかどうかを判定し，複製元記事（82 記事）を収集し，これを原典文書とした上で，上記ブログを複製検出対象として本システムに入力し，何記事検出できるか評価した．

評価を行った結果，66 記事に複製が含まれているという結果を得た．これにより再現率は 80%となった．

不正解の内訳は以下の通りである．

（A） 複製範囲が短い
（B） セグメンテーション誤り

（A）に関しては，複製判定基準（実験では連続 3 セグメント以上）を短くすれば，検出漏れを防ぐことも可能であるが，偶然一致する確率が高まり誤検知してしまう恐れがある．そのため，判定条件を一律に短くするのではなく，セグメントの出現しにくさ（情報量）を加味するなどして，改善していく必要がある．

（B）に関しては，改行文字をセパレータに含めていなかったため，箇条書きのような，複数の項目が改行文字によって分割されている場合，各項目を独立したセグメントとして認識することができなかったことが原因である．改行文字をセパレータに含めれば，このような誤りは発生しなくなるが，そうすると，今度は改行位置を変更しただけの複製を検出できなくなってしまう．従って，文体や係り受け解析結果等に応じて，分割位置を柔軟に変更できるようにする等の検討をする余地がある．

7. まとめ

本研究では，大規模かつ高速な複製検出を行うための手法を考案し，プロトタイプを実装した．実験結果から，大規模かつ高速な複製検出が可能であることを確認した．また，精度と再現率の実験を行い，効果を確認した．今後は，セグメンテーション方法や正規化処理の検討を行い，誤検出や検出漏れを改善していく予定である．

[文献]

[1] 総務省情報通信政策研究所調査研究部: "ブログの実態に関する調査研究の結果". http://www.soumu.go.jp/iicp/chousakenkyu/data/research/survey/telecom/2008/2008-1-02-2.pdf, 2008年7月.
[2] M. Charikar: "Similarity Estimation Techniques from Rounding Algorithms". ACM STOC 2002, 2002.
[3] A. Broder, S. Glassman, M. Manasse and G. Zweig: "Syntactic Clustering of the Web". WWW6, 1997.
[4] R. J. Bayardo, Y. Ma, and R. Srikant: "Scaling up all pairs similarity search". WWW2007, 2007.
[5] S. Brin, J. Davis and H. Garcis-Molina: "Copy Detection Mechanisms for Digital Documents". ACM SIGMOD Annual Conference, 1995.
[6] 田代崇，上田高徳，堀奏佑，平手勇宇，山名早人，"Webページを対象とした著作権違反自動検出システム"，信学技報DE2006-54, 2006.
[7] U. Manber and G. Myers: "Suffix arrays: a new method for on-line string searches". SODA 1990, 1990.
[8] Michael O. Rabin: "Fingerprinting by random polynomials". Center for Research in Computing Technology, Harvard Univ, TR-15-81, 1981.

櫻井 俊之 Toshiyuki SAKURAI
1998 年神奈川大学電気工学科卒．2000 年同大学大学院工学研究科電気工学専攻博士前期課程修了．同年東日本電信電話（株）入社，次世代 IP ネットワーク，情報セキュリティ，自然言語処理に関する研究開発に従事．日本データベース学会，電子情報通信学会各会員．

松尾 義博 Yoshihiro MATSUO
1988 年大阪大学理学部物理学科卒．1990 年同大大学院研究科博士前期課程修了．同年日本電信電話（株）入社．機械翻訳，自然言語処理の研究に従事．言語処理学会，情報処理学会各会員．

菊井 玄一郎 Genichiro KIKUI
1984 年京都大学工学部電気工学科卒．1986 年同大学大学院工学研究科博士前期課程修了．同年日本電信電話（株）入社．1990～94 および 2001～06（株）国際電気通信基礎技術研究所（ATR）に出向．言語処理に関する研究開発に従事．博士（情報学）．言語処理学会，情報処理学会，電子情報通信学会，人工知能学会各会員．