社内文書検索システム（２）
-ステーブルオブジェクト分析による同等文書検出-

松田 勝志†
NEC サービスプラットフォーム研究所†

1 はじめに

企業や組織内には，ワープロ文書，プレゼンテーション文書，表計算文書，DTP 文書，CAD 文書等のさまざまな文書が多数存在する．これらの文書には，ある人が作成した文書を別の人がコピーした上で修正したり，作成者が修正のたびに別文書にしたり，電子メール等で多数の人に配布されて保存されたり等，同一またはほぼ同一な文書(以降，同等文書と呼ぶ)が存在する．これらの同等文書の存在は，ストレージの容量を圧迫する他，文書検索において多数の同じような文書が検索結果に表示されてしまう，といった問題を引き起こす．

本稿では，文書中で修正が行われるのは修正が容易な箇所や目立つ箇所であり，細かな箇所は修正が行われにくいという仮説のもと，そのような箇所を特定し，そこから一部の文字列を抽出し，ハッシュ化することで高速に同等文書を検出する方式について述べる．

2 同等文書

同等文書の内，ほぼ同一な文書とは，文書のコピー操作後，軽微な修正を行い，電子データ的に同一ではなくかつ内容的にほぼ同一な文書のことである．軽微な修正とは，内容を大幅に変えない修正であり，一義的に定義することは困難であるが，例えば，ある人が作成した文書をその上司が文言等を修正する，自身で文言や図表を修正する，文書の宛先や日付を修正する，等が該当する．

同等文書は様々な方法で生成される．メールでの配布やファイルサーバからのダウンロード等により同一文書が生成され，それらの同一文書から同等文書が生成される．

多数の同等文書の存在は，ファイルサーバのストレージ容量の圧迫や文書検索時に同等文書が結果に出力されてしまう，といった弊害を生じさせる．ストレージの単価が低下している現在，後者の問題の方がより深刻である．すなわち，低品質な検索結果は，検索効率の低下や検索システムへの信頼性の低下と利用率の低下という深刻な問題を引き起こす．

An In-house Documents Retrieval System (2) –Equivalent Documents Detection by Stable Objects Analysis-
†, Katsushi MATSUDA, Service Platforms Research Laboratories, NEC Corporation

3 ステーブルオブジェクト分析

大量の文書から同等文書を検出するためには，高速な検出方法が必要である．従来の単語ベクトルによる重複文書照合問題の場合，処理の重い類似度計算を大幅に削減するために prefix-filter[1][2]やそれを更に高速化する multi-level prefix-filter[3]等が用いられている．しかしながら，いずれの方法でも類似度計算を行う必要があることは同じである．

そこで筆者らは，対象文書をプレゼンテーション文書等のオフィス文書に限定することで，類似度計算処理を行わずに高速に同等文書を検出する方法を検討している．

同等文書はその性質上，ほとんどの箇所が元文書と同一である．また，元文書を修正する人や再利用する人は，文書中の修正が容易な箇所や目立つ箇所を中心に修正を行うと考えられる．逆に，修正が困難な箇所や目立たない箇所はそのまま修正されず，元文書と同一のままであると考えられる．すなわち，修正されなかった安定な部分(**ステーブルオブジェクト**)が存在する．本稿では，対象文書をプレゼンテーション文書とし，文書中の部品の密集度が高い箇所が修正困難で目立たない箇所に相当すると仮定して，ステーブルオブジェクトを特定する．

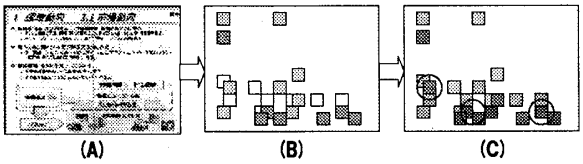

図 1．ステーブルオブジェクトの特定

ステーブルオブジェクトを特定する処理をステーブルオブジェクト分析と呼ぶ．本分析手順の概略を図 1 に示す．(A)は対象文書の例である．まず文書中の全部品を抽出し，それらの部品の面積 S_iおよび文字列長 l_i，更に飾りの数 d_iから各部品の密度 m_iを求める(B)．式(1)が密度 m_iの計算式である．本稿では，テキストを含まない部品の密度 m_iは 1 としている．また，Cは正規化のための定数であり，テキストを含む部品の密度が 1 以上になるよう調整する．

$$m_i = \begin{cases} \dfrac{C \times (l_i + d_i)}{S_i} & (l_i > 0) \\ 1 & (l_i = 0) \end{cases} \quad \cdots\cdots(1)$$

次ぎに，周辺の部品を含めた密集度 M_i を求める(C)．本稿では，式(2)にあるように半径 D 以内にある周辺部品の密度による加重を行う．

$$M_i = m_i + \sum_j g \frac{m_i m_j}{r_{ij}^2} \quad (r_{ij} \le D) \quad \cdots\cdots(2)$$

半径 D を設けた理由は，全ての部品の組合せで計算すると計算量が多くなるためであり，文書を半径 D の倍の長さでグリッドに分割しておけば，部品 m_i の位置するグリッドおよびその周辺の最大 4 個のグリッド中の部品 m_j との組合せ計算に抑えることができる．最後に密集度が高い部品をステーブルオブジェクトとする．

4 ハッシュによる同等文書の高速検出

4.1 同等文書検出の概略

ステーブルオブジェクト分析を行うことによって，処理の重い類似度計算を行わずに同等文書を検出することが可能となる．同等文書検出処理の概略は以下の通りである．

(1) ステーブルオブジェクトを特定
(2) ステーブルオブジェクトから文章または複数の単語を抽出
(3) 抽出した文章または結合した単語で特徴文字列を生成
(4) 特徴文字列からハッシュ値を算出
(5) ハッシュ値をハッシュテーブルに登録する際に衝突が発生したら同等文書と特定

上記は，1 文書に対して 1 つのステーブルオブジェクトを特定した場合の処理を示しているが，実際には，プレゼンテーション文書の各スライド単位に数個(例えば 3 個)のステーブルオブジェクトを特定し，別々の特徴文字列とそのハッシュ値を算出し，全てのハッシュ値が同じスライドと衝突した場合にのみ同等文書と特定する．

4.2 特徴文字列生成

計算式(1)および(2)は，文書中の修正困難で目立たない箇所を特定するものと考えているが，そのような箇所に修正がされないとは限らない．ステーブルオブジェクト内のテキストの軽微な修正(例えば「てにをは」の修正等)がされた場合も考慮して，本節ではステーブルオブジェクトのテキストから文章の抽出ではなく，単語を抽出する方法について述べる．

本稿で提案する同等文書検出は高速性に注力しているため，処理の重い形態素解析等は行なわず，文字コード区分による簡易な抽出方法を行なう．具体的には，テキストを構成する文字の種類(漢字，ひらがな，アルファベット，ASCII 文字等)が変化する境界で分割し，漢字またはアルファベットの部分文字列のうち文字列長が長い文字列を抽出する．例えば，「本稿で提案する同等文書検出は高速性に注力しているため」という文字列の場合，「同等文書検出」が抽出できる．

4.3 ハッシュ値の衝突

特徴文字列のハッシュ値を計算し，対象文書集合に対して十分な大きさのハッシュテーブルに登録する．この際にハッシュ値が衝突するということは，同じ特徴文字列である可能性が高い．1 文書(もしくは 1 スライド)から複数個のハッシュ値を生成し，それぞれをハッシュ値 - 文書 ID のペアでハッシュテーブルに登録した場合，ある文書と全てのハッシュ値が偶然一致する可能性は極めて低いため，その文書を同等文書と特定しても構わない．

ハッシュテーブルを用いることで，同等文書検出は，対象文書集合を 1 回スキャンするだけで完了する．すなわち，対象文書数を n とすると，$O(n)$の計算量となる．

同等文書検出によって，同等文書の組合せが判明するため，文書検索時に同等文書を結果から削除することが可能となる．

5 おわりに

本稿では，文書中の部品が密集した箇所は修正がされにくいという仮説のもと，ステーブルオブジェクト分析とハッシュテーブルを用いて高速に同等文書を検出する方式について述べた．

現在，本方式を用いたシステムを試作中である．実装が済みしだい評価実験を行い，仮説の検証および従来の重複文書照合方式との速度比較を行なう予定である．

参考文献

[1] Sarawagi et al，Efficient Set Joins on Similarity Predicates，*Proceedings of the 2004 ACM SIGMOD International Conference on Management of data*，pp.743-754，2004.

[2] Chaudhuri et al，A Primitive Operator for Similarity Joins in Data Cleaning，*Proceedings of the 22nd International Conference on Data Engineering (ICDE06)*，pp.5-16，2006.

[3] 立石他，multi-level prefix-filter を用いた高速重複文書照合，*日本データベース学会 Letters*，Vol.5，No.4，pp.49-52，2007.