社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－DBS－130 (23)
2003－ＦＩ－ 71 (23)
2003／5／23

テキストからの再利用文字列の抽出と分析

相澤 彰子

国立情報学研究所

(akiko@nii.ac.jp)

現実にネットワーク上を流通するテキストの中には、複写、編集、改訂、引用等、文書の生成や伝達過程における操作により、文章や段落などの長い単位で単語列が正確に一致するものが多数存在する。これらの一致は文書間の結びつきを表す有用な情報であると捉えることができる。本稿では、上記のような単語列の反復を, 接尾辞木構造を利用してテキストから高速に検出する手法を検討し、実際に新聞記事をはじめとする複数の大規模な文書セットに適用して分析を試みた結果を報告する。

Extracting and analyzing recycled word sequences from text

Akiko AIZAWA

National Institute of Informatics

Texts distributed in real-world networks contain numbers of duplicated word sequences that are sometimes longer than sentences or paragraphs. These word sequences exactly match due to such operations as copying, editing, revising, quoting in generating or transferring documents, and can be considered as useful information that represents associations between documents. In this paper, we propose a suffix-tree based method for extracting repeated word sequences efficiently. Some analytical results are also reported where the proposed method is applied to different types of large-scale text collections such as newspaper articles and abstracts of academic papers.

1 はじめに

従来から情報検索の分野においては、テキスト文書間の類似度を、それぞれの文書中に出現する単語の分布特性の類似を用いて定義することが一般的に行われてきた。一方で、現実の文書データベースやＷｅｂ上には、引用、編集、改訂等により、文章や段落にわたる長い単位でテキストが字句どおり一致する文書が多数存在する。たとえば、Reuters などの代表的なテストコレクションを分析すると、単語の分布特性に基づく類似度が上位となる文書グループでは、グループ内の文書どうしがほぼ同一であることがわかる [1]。この性質は、単語分布に基づく重複文書検出の前提条件ともなっており、一般に成り立つものであると考えられる。

さて、文章をまたがるようなテキストの一致は、単なる偶然ではなく、著者（あるいは文書の発信元）自身の意図的な引用、あるいは編集作業や流通過程での操作により生じたものであるとみなせる。特に情報化が進む今日では、電子テキストの一部を部品として再利用することは日常的に行われているものと考えられ、全体として反復テキストの比率が高まっていることが予想される。ここで注意しなければならないのは、単語分布の統計的な類似性と、単語列の比較的長い単位にわたる一致は、互いに強い相関はあるが、本質的に異なる事象であるということである。すなわち前者は、文書どうしの話題の共通性に起因するものであるが、後者は、著者や発信者の同一性や参照・被参照関係によるつながりを示唆するものである。

しかしながら、従来のテキスト処理研究のほとん

どは、文書を順序関係を持たない単語の集合として捉えるか、特定のテキストに依存しない語の連接関係に注目するかのいずれかであり、単語 N グラムを用いる場合でも、N の値は文法的な結びつきが強い範囲（2 から高々10 程度）に設定される場合が大半であった。単語 N グラム長が 10 以上になるような単語列の一致を文書間で検出し、文書の発信者どうしの関係の抽出や話題性の分析を行うアプローチはこれまであまりなされていない。

上記の観察に基づき本稿では、複数の文書にまたがり反復される単語列を抽出し、単語列を共有する文書をグループ化するための基本的な手法について検討し、実際のテキスト文書を対象として話題や著者関連性の分析を試みた結果を報告する。検討に先立ち、文書間で反復される様々な長さの単語列を以下の 3 つのタイプに分類する。

(1) 複合語や定型句

辞書登録の対象となるような、用法が確立した語彙や言い回しで、比較的高い頻度でコーパス中に広く出現するもの。

(2) 限定的定型句

特定の話題やコミュニティの範囲で、一時的に頻出する比較的長い単位の言い回しで、著者や発信者を特定することなく使い回されるもの。

(3) 引用テキスト

著者や発信者に特有の言い回しで、基本的に反復利用にあたっては引用の断り書きが必要であるとみなされるもの。

本稿で注目するのは第二、第三のタイプ、すなわち「限定的定型句」および「引用テキスト」である。これらのタイプの単語列は従来の自然言語処理ではあまり注意が払われることがなかったが、与えられた文書の話題や発信者を強く特徴づけるものである。特に限定的定型句は、往々にして有用な情報を含むものであり、具体的には、新聞記事の場合には、ある事件に対する一連の報道で一貫して使われる組織名や人物の肩書き、複数の地方版で共通する記事本文の重要部などが該当する。

以下、本稿の構成は次のとおりである。まず、**2.** で関連研究を概観し、次に、**3.** で反復単語列を対応する文書グループとともに抽出するための効率的な手法について述べる。様々な種類のテキストに容易に適用できるように、本稿における手法は特に単純で一般的なものとなるよう配慮している。さらに、**4.** で、上記の手法を複数の文書セットに適用して、文書に対応づけられたカテゴリや発信者情報を手がかりに、抽出された単語列を分析した結果を報告する。最後に **5.** で、今後の方向について述べる。

2 関連研究

本稿で検討の対象とする反復単語列の抽出と利用の問題は、テキスト流通の限定的な一面を捉えたものではあるが、以下に示すように、情報検索の多様な応用分野と関係が深い。

2.1 著者同定

従来より計量文献学の分野では、出典が不明、あるいは盗作や贋作の判定が必要な文献の著者同定問題に関して統計的なアプローチが適用されてきた [2, 3]。具体的には、著者の執筆のスタイルやクセを文の長さ、語彙の豊富さ、特殊な文末表現等の様々な尺度で数量化して、著者同定を行うものである。計算機の高速化や分かち書き処理技術の向上を背景に、最近の研究では、日本語文献を対象とした単語Nグラム（$N = 3 \sim 10$）なども尺度として取り入れられている [6, 7]。しかしながら、過去の研究の多くは、古典的な著作や文学作品の真贋の識別を目的としており、上記のような著者数量化のフレームワークが、一般の発信者にまで適用されはじめたのはごく最近である [4, 5]。

2.2 重複文書の検出

テキストの電子化によって多数の文書の中から重複する文書を検出する問題が、近年とりわけ重要となっている。Chowdhury らは、従来の重複文書検出手法を以下の２つに分類している [8]。第一は「shingles」と呼ばれる文字列の一致に基づく方法であり、連続した単語列など文書固有の要素集合を比較することで効率的に重複文書の検出を行おうというものである [9, 8]。第二は 類似度計算に基づく方法であり、単語分布の類似度を用いて重複文書の候補を検出しようとするものである [10, 1]。重複文書の検出では、多くの場合、単純に完全一致する文書ペアを抽出するだけではなく、あらかじめ定めた類似度の閾値を超えるものを重複と判定する。すなわち、文書間での若干の差異を許容しているが、基本的に文書全体を１つの単位として比較するものである。これに対し

て本稿では、文書中に含まれるテキストの部分的な重複に注目した検出手法を検討する。

2.3 高速文書クラスタリング

文書間で共有される単語列に基づく高速な文書クラスタリング法として、Zamir らは「Suffix Tree Clustering」（接尾辞木クラスタリング、以下 STC）を提案している [11]。STC は Web の検索結果をリアルタイムで再構成するために提案された手法で、速度および生成されるクラスタの質ともに優れた性能を示したと報告されている。接尾辞木構造を利用する点で本稿におけるクラスタリングともっとも関連が深い。ただし、STC は意味的に関連する文書集合の抽出を目的としていることから、単語列の正確な一致に注目する本稿とはクラスタリングの適用方法が異なっている。具体的には、STC は前処理で単語の正規化や文単位でのテキストの分割を適用しており、一定長以上の単語列についてはペナルティを課している。また STC では、接尾辞木構造に基づき抽出した文書クラスタを「base cluster」として、これを統合することで最終的なクラスタリング結果を生成している。本稿では正確な単語列の一致に焦点をあてているため、これらの処理は行わず大規模なテキストへの適用を目指す。

3 接尾辞木構造に基づくクラスタリング

3.1 ＳＴクラスタの定義

文書集合を $\mathcal{D}$、文書中に出現するすべての単語（異なり）の集合を $\mathcal{W}$ として、単語 N グラム（$N \geq 1$）は、N 個の単語の順序つき集合として以下で与えられる。

$$w_1^N = (w_1, \cdots, w_N) \qquad (w_i \in \mathcal{W}) \qquad (1)$$

D の接尾辞木（suffix-tree）とは、D 中のすべての単語列を含む圧縮した木構造、すなわちトライである。接尾辞木上の各ノードは、異なる単語 N グラム w_1^N に対応しており、「w_1^N を含むすべての文書」として唯一に定まる文書集合と対応づけられている。以下、このような文書集合を $S_d(w_1^N)$（$\subset \mathcal{D}$）と表記する。 ここで複数のノードが同じ文書集合に対応する場合があることに注意して、ＳＴクラスタ（suffix-tree based cluster）を、同じ文書集合に対応付けられる単語 N グラムの集合を用いて、以下のように定義する。

（定義） S を単語 N グラム集合、D を文書集合として、$s \subset S$ なるすべての s について $S_d(s) = D$ であり、かつ、$s \not\subset S$ なるすべての s について $S_d(s) \neq D$ であるとき、$c = (S, D)$ をＳＴクラスタと呼ぶ。

たとえば図１において、ノードＡとノードＢはともに文書集合 { DOC#10, DOC#13 } に対応づけられており、あわせて１つのＳＴクラスタを構成している。

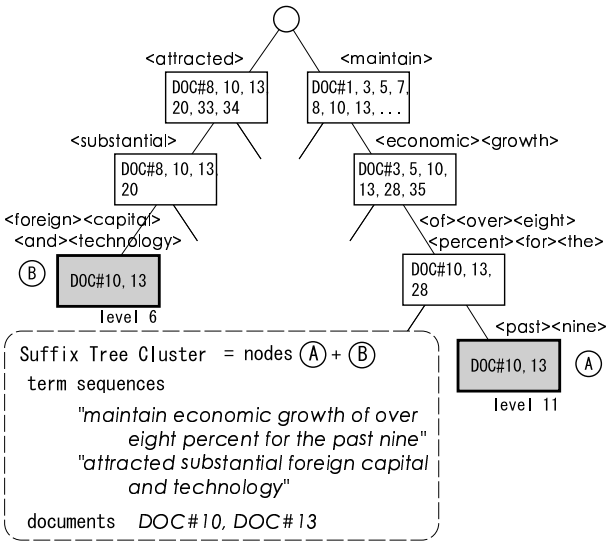

図 1: ＳＴクラスタの例

3.2 ＳＴクラスタの生成手順

ＳＴクラスタ生成の基本的な手順を以下に示す。

(1) １つの文書を１つの単語列に対応させて、各単語へのポインタを格納した配列を生成する。日本語の場合には、形態素解析を適用して分かち書き処理を行う。正規化や不要語の削除は行わない。

(2) 上記のポインタを一括してソートし、単語列の配列表現（suffix array) を生成する。このとき、接尾辞木上のノードは配列上の連続した区間に対応しており、区間内にあるポインタのアドレスから文書集合がただちに特定される。これを利用して接尾辞木を生成する。また、各ノードに対応する文書集合を配列に格納し、あらかじめ定めた順序にしたがって別々にソートしておく。

(3) 生成した接尾辞木ノードすべてを、(2) の文書配列をキーとしてソートする。ソート後のリストにおいて、同じ文書集合に対応づけられたノードどうしは隣接しており、ＳＴクラスタは連続した１区間として求められる。

3.3 コストと性能の見積り

上記手順の計算時間は、基本的にステップ (3) における接尾辞木ノードのソート時間によって定まる。ソートに要する計算時間は標準的なパッケージでも高々$O(n\ log\ (n))$ であり、現実問題としてはさほど問題とならない。コスト面で考慮が必要となるのは、メモリ上にポインタを格納するための記憶容量であり、具体的には、ステップ (2) において、接尾辞木のノードと、ノード毎の文書配列を保持するために必要な容量の見積りが主要な検討事項となる。

図 2 では、本稿の実験で用いた Reuters 記事コーパス（**4.1** 参照）を用いて、(i) 接尾辞木のノード数と、(ii) ノード毎の文書配列長の合計値を、接尾辞木の深さ（単語 N グラム長）ごとに示している。この例からも明らかなとおり、実際に多くの記憶容量を消費するのは、長さが数個以下の短い N グラムである。一方、本稿の目的は長い単位での単語列一致に注目して分析を行うことであるから、単語列長が一定値以上であるものだけを対象とすれば、コストの大幅な削減が期待できる。

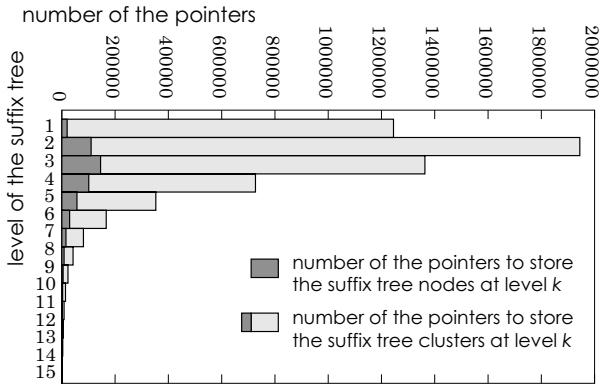

図 2: 深さ k である接尾辞木のノードと対応する文書の総数

そこで、長さ L の単語列 $w_1, \cdots, w_L$ が反復される場合の、検出率のおよその目安を計算する。まず、各単語 w_i についてランダムかつ独立に、確率 p で「反復誤り」（表記揺れ、置換、欠落など）が生じるものとする。長さが n_0 に満たない単語 N グラムは無視されるとすると、反復が検出されるためには、L 個の単語列の中で少なくとも n_0 個以上が連続して正しく再現されなければならない。

いま、k 個の誤りを含む単語列で反復が検出されない場合について考えると、k 個の誤りの間隔がすべて $(n_0 - 1)$ であるとき L は最大値となる。逆に、それよりも長い単語列については、必ず長さが n_0 以上の部分単語列が接尾辞木上に現れる。すなわち、以下を満足するとき、反復単語列の少なくとも一部が必ず検出可能である。

$$\begin{aligned} L &\geq ((n_0 - 1) \times (k + 1) + k) + 1 \\ &= n_0 \times (k + 1) \end{aligned} \tag{2}$$

言い換えると、長さ L の単語列に対して、誤り数 k が次式を満足すれば、必ず反復が検出可能である。

$$k \leq \frac{L}{n_0} - 1 \tag{3}$$

これより検出確率の下限値は次式で与えられる。

$$\epsilon(L, n_0) = \sum_{k=0}^{\lceil \frac{L}{n_0} - 1 \rceil} {}_L C_k p^k (1 - p)^k \tag{4}$$

たとえば $n_0 = 5$ とするとき、$p = 0.1$ では $L = 20$ で 87%, $L = 60$ で 99%、$p = 0.05$ では $L = 20$ で 99%以上の検出率となり、$N \leq 4$ なる N グラムを無視しても、反復候補の数え上げの精度は大きな影響を受けないことが予想される。

3.4 ＳＴクラスタの評価尺度

抽出した反復単語列の一致の強さを数量的に評価するために、「単語列一致度」（term sequence coincidence）を、（個別）相互情報量 $M(w_1^N)$ の定義にしたがって次式で求める。

$$M(w_1^N) = log \frac{P(w_1^N)}{P(w_1) \cdots P(w_N)} \tag{5}$$

すなわち、$M(w_1^N)$ は、(i) k 個の単語 $(w_1, \cdots, w_k)$ が独立に生じたと仮定する場合のエントロピーと、(ii) 実際に観測された生起に基づき求めたエントロピーの差分である。

直感的には、式 (5) の $M(w_1^N)$ の値は長い単語列の場合ほど大きくなる。単純に単語列長を尺度として用いる場合との違いは、式 (5) では頻度を考慮して特殊な事象ほど高く重みづけている点である。別途行った予備実験において、単語列長と式 (5) によ

表 1: 実験で用いたテキスト文書集合

Data source	Period	Language	#Docs	#ST-cluster	Execution time
Reuters	1996.8.20-1997.8.19	Eng	109,433	1,338,735	2644 sec.
SJM	1991.1.1 -1991.12.31	Eng	72,947	320,457	595 sec.
Mainichi	1998.1.1 -1998.12.31	Jpn	10,855	111,406	78 sec.
Nikkei	1996.1.1 -1996.12.31	Jpn	911	19,745	10 sec.
ntc-IPSJ	1988.5.19-1997.7.25	Jpn	26,796	226,640	99 sec.
ntc-JSCE	1991.9.17-1996.9.17	Jpn	21,259	180,538	70 sec.

る一致度の両者を、クラスタ内文書のカテゴリや著者のばらつきとの相関を用いて比較したところ、後者の方がよい値を示すことが確認された。

式 (5) の $P(w_1^N)$ は単語列 w_1^N の出現確率であり、w_1^N の全文書集合 $\mathcal{D}$ 中での出現頻度 $freq(w_i)$ および総語数 $F = \sum_{w_i \in \mathcal{W}} freq(w_i)$ を用いて次式で計算する。

$$P(w_1^N) = \frac{freq(w_1^N)}{F} \ . \tag{6}$$

単語 w_i の出現確率 $P(w_i)$ についても同様に、w_i を長さ１の単語列であるとみなして式 (6) から求める。ここでは未知語の確率推定を行う必要がないことから、単純さを重視して、平滑化による確率補正は適用していない。

ＳＴクラスタ (S, D) が与えられた場合、まず S の中のすべての単語列について上式 (5) による単語列一致度を計算し、クラスタ全体としての一致度は、これらの値の総和または最大値として求める。いずれを評価値とするかは、分析の目的に応じて定めるものとする。本稿の実験では統計的な特性を調べるため、一貫して最大値を尺度として用いている。

4 実験結果

4.1 対象とするテキスト文書セット

実験では表 1 に示す 6 つの文書セットを用いた。Reuters [12] および SJM (San Jose Mercury) [13] は英文のニュース記事、Mainichi [14] および NIKKEI [15] は、毎日新聞および日本経済新聞 CD-ROM 版から抽出した和文のニュース記事である。また、ntc-IPSJ および ntc-JSCE はそれぞれ、NTCIR-1[16] に収録された学会発表文献抄録データのうち、情報処理学会および土木学会における発表文献に対応している。新聞記事については、発信者が記事中に示されている記事だけを選んで実験に用いた。具体的には、Reuters および San Jose Mercury については、明示的に <byline> でタグつけされた文字列を、Mainichi および NIKKEI については、記事末尾に特定のフォーマットで付与された発信者情報（“【ワシントン１５日逸見義行】” など）を分析して用いた。

これらの文書セット中の文書は日付やＩＤ情報によって互いに区別可能であるが、テキスト部分だけに注目すると、互いの内容が同一となる「重複文書」が少数存在し、その比率は、Reuters で 3,014(2.8%)、Mainichi ではゼロ（重複なし）、ntc-IPSJ で 6 (0.06%) であった。本稿の実験では日付情報を考慮した分析のために、これらの重複文書を含んだままの形で処理を行った。日本語の分かち書き処理には形態素解析ツール茶筅 [17] を用いた。

なお表 1 では、各文書セットの収録期間、言語、文書数に加えて、前節で述べた手法を適用して抽出したＳＴクラスタの総数、および 2.8GMHz Xeon/Linux（1CPU）による実行時間を示している（ただし、比較のため形態素解析および辞書作成の時間は含んでいない）。

4.2 実験 1: 限定的定型句に注目した分析

最初の実験では、話題の重なりが少ない２つの文書セットを選び、これらに共通する単語列を抽出して、一致度の分布を調べた。ここでの主な目的は、発信者の直接参照や見聞なく再現される単語列について、対象とするテキストの違いによる一致度分布の揺らぎを調べることである。

このために、まず、(a)Reuters と San Jose Mercury (SJM)、 (b) Mainichi と Nikkei、 および (c) ntc-IPSJ と ntc-JSCE、それぞれの文書セットのペアを混合した上でＳＴクラスタを生成した。次に、その中から、Reuters と SJM のように異なる由来の文書が混在する「混合クラスタ」を抽出し、その一致度分布を調べた。ここで (a)(b)(c) それぞれのペアは、互いに年代や分野をかえることで、話題の重複がなるべく少なくなるように配慮してある。

表 2 に、混合クラスタの一致度の最大値、上位 30 個の平均値、および、抽出した混合クラスタの 95%がその値以下になるような「95%境界値」を示す。括弧内の数値は対応する単語列の長さ、すなわち一致した単語数である。また図 3 に、(a)(b)(c) の 3 ペアに関して抽出した混合クラスタの一致度分布について、面積が 1 になるよう正規化したヒストグラムの比較結果を示す。

言語や分野の違いを考慮すると、3 つの文書セットすべてについて、一致度の分布は一貫した傾向を示しているといえる。すなわち、互いに話題の重なりが少ない文書の場合、値が 50~100 よりも大きくなるような反復単語列の存在は、比較的まれである。言い換えれば、それより一致度が高い単語列の存在は、何らかの話題あるいは分野的なつながりを示すことが推察される。

表 2: 話題の重なりが少ない場合の一致度の値

Data Source	Top score	Top 30 Avr.	95% Th.
Reuters-SJM	162.6 (9)	95.7 (9.8)	38.7 (6)
Mainichi-Nikkei	116.8 (12)	93.2 (10.6)	47.7 (7)
IPSJ-JSAI	133.8 (12)	102.3 (13.8)	45.5 (8)

図 4 に、Reuters の各文書に対して付与されたカテゴリを用いて、クラスタ内の文書が共通カテゴリを持つ場合の一致度分布を求めた結果をに示す。比較のため、クラスタ内文書の日付偏差が 3 日以内、1 日以内となる場合についての値もあわせて示している。このように、一致度の分布は、話題が高い文書ほど大きな偏りを示すことが確認できる。ただし、特定の文書間で話題が一致したからといって、必ずしも単語列一致度が高い値を示すわけではないので、上記の 95% 境界値は、話題の一致不一致を判別するためのものではないことに注意が必要である。

また、一致度の高い文字列は多くの場合、文書クラスタの内容を推察する手がかりとなる情報を含むことが観察された。たとえば、(a)Reuters および SJM pairs における上位 3 個は、(i) "Kate, Larry, Mindy, Nicholas, Odette, Peter, Rose, Sam, Teresa," (score 162.6; 台風の名称)、(ii) "said David Jones, chief economist at Aubrey G. Lanston & Co." (score 124.8)、(iii) "the Golan Heights, which Israel captured from Syria in the 1967 Middle" (score 113.6) であった。将来的には、このような反復文字列を文書クラスタの要約に活用するアプローチも考えられる。

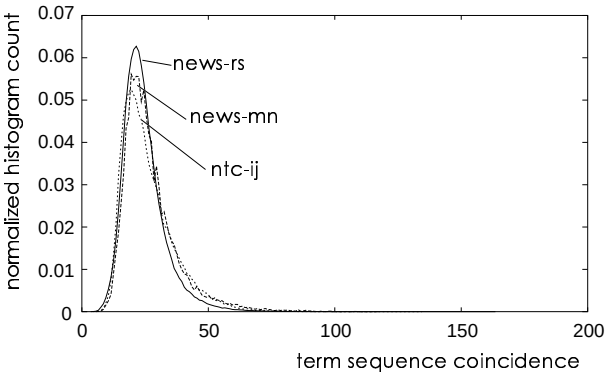

図 3: 話題の重なりが少ない場合の一致度の分布

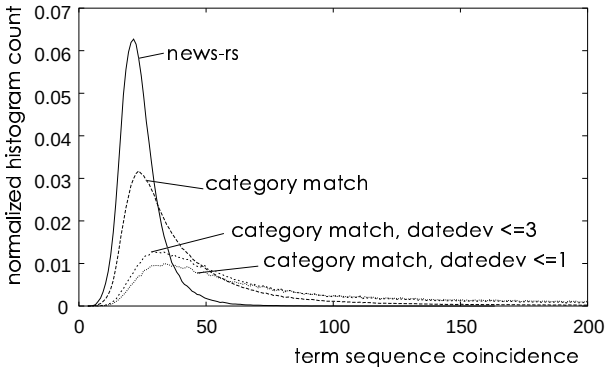

図 4: Reuters のカテゴリに対応する一致度の分布

4.3 実験 2: 引用テキストに注目した分析

次の実験では、Reuters, Mainichi, ntc-IPSJ の 3 つの文書セット各々について、クラスタ内文書の著者が互いに異なるＳＴクラスタを抽出し、単語列一致度の分布を調べた。ここで著者が「異なる」とは、クラスタ内の文書すべてに共通であるような著者（共著者を含む）が存在しない場合を指している。

表 3 に、著者が異なるクラスタの 95%境界値を示す。参考のため、表 3 では、各文書セットごとにテキストを文単位に分割し、式 (5) による一致度の文あたり平均値もあわせて示している。また図 5 に、正規化したヒストグラムによる一致度分布の比較を示す。表 2、図 3 の場合と比較すると、対象文書セットによる違いが見られることがわかる。

表 3: 著者が異なるＳＴクラスタの単語列一致度の値

Data Source	category	Sentence Avr.
Reuters	907.4 (84)	330 (24.5)
Mainichi	164.2 (20)	394 (33.6)
ntc-IPSJ	110.2 (15)	420 (32.4)

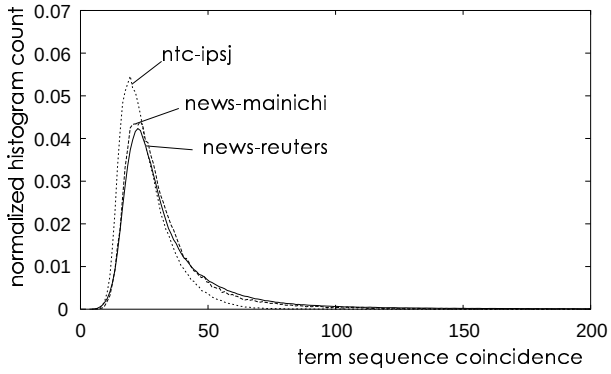

図 5: 著者が異なるＳＴクラスタの単語列一致度の分布

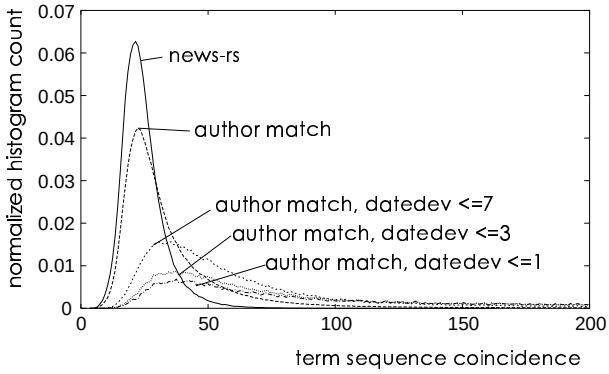

図 7: Reuters における一致度分布の日付偏差による変化

この文書セットによる違いをさらに明らかにするために、クラスタ内文書の日付偏差による 95%境界値の変化を調べた。日付偏差 t を、0 ～ 30 日の範囲で変化させ、偏差 t 以下となるクラスタについて 95%境界値を求めた結果を図 6 に示す。また、Reuters の各文書に対して、クラスタ内の文書の日付偏差が、7, 3, 1 日の各値以下である場合の一致度分布を図 7 に示す。これらの分析結果から、Reuters における一致度の値は、記事の日付に依存しており、数日以内に発信される記事の間では記者名によらず反復が行われていることがわかる。このことからテキスト反復利用の形態は、メディアや分野に依存して様々であることが推測される。

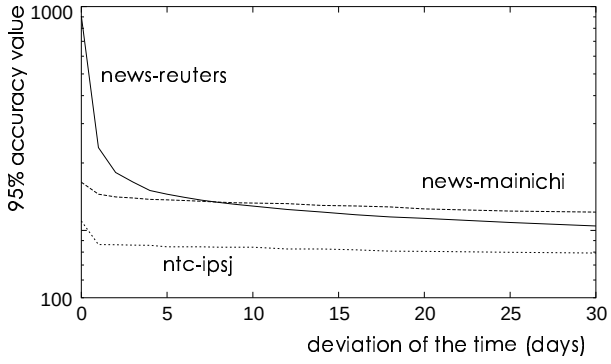

図 6: 文書セットごとの日付偏差の影響

4.4 実験３: 反復利用比率の測定

最後の実験では、Reuters、Mainichi、ntc-IPSJ の３つの文書セットについて、テキストの「反復利用比率」を測定した。具体的には、単語列の一致度が、与えられた閾値 c を越えるものについて、２度目以降の出現による単語数を求め、全体ののべ単語数に対する比率を求めた。c の値を $10 \leq c \leq 600$ の範囲で変化させた場合の反復利用比率を図 8 に示す。ntc-IPSJ について反復利用比率 0.05 弱、Reuters および Mainichi については 0.15 ～0.25 と比較的高く、この数値は Web などの環境ではさらに高いことが予想される。

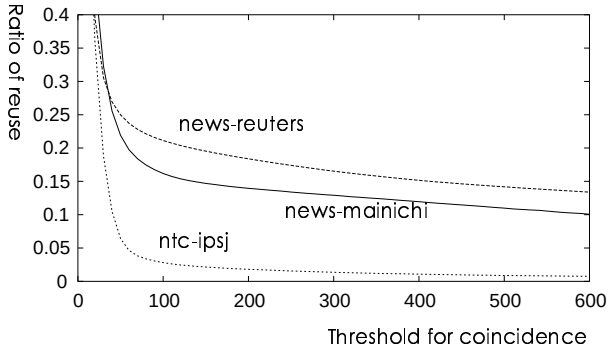

図 8: 文書セットごとの反復利用比率

5 おわりに

本稿では、発信者、日付、カテゴリ等があらかじめ付与されたテキスト文書セットを用いて、長い単位での単語列の一致について統計的な分析を試みた。本稿での結果はまだ予備的検討の域を出ていないが、実験を通して、提案する分析の枠組みを Web 等現実的な規模の文書セットに適用するための手法の確立とスケーラビリティの確認が行えたものと考えている。今後の課題としては、文書中に埋め込まれた引用テキストの抽出による情報ナビゲーションや、発信者、日付、カテゴリ等のタグを利用したイベント特有表現の抽出等のアプリケーションへの適用が考えられる。また、無記名のテキストに対して、著者

や話題の単一性をどのように数量化したらよいかという問題は、今後のテキスト流通において重要であり、さらなる検討と分析が必要である。

謝辞

本研究を行うにあたり数々の有益なコメントを頂きました国立情報学研究所の大山敬三氏に感謝いたします。

参考文献

[1] Mark Sanderson: "Duplicate Detection in the Reuters Collection," Technical Report of the Department of Computing Science at the University of Glasgow, TR-1997-5 (1997).

[2] 村上征勝. 「行動計量学シリーズ：真贋の科学」朝倉書店. 1994.

[3] Tony McEnery and Michael Oakes: "Authorship Identification and Computational Stylometry," in Handbook of Natural Language Processing, Marcel Dekker Inc., 545–562 (2000).

[4] Yuta Tsuboi and Yuji Matsumoto: "Authorship Identification for Heterogeneous Documents," SIG Notes of Information Processing Society in Japan, SIG-NL-148, 17–24 (2002).

[5] 佐藤進也、原田昌紀、風間一洋: "文字列出現頻度比較による情報源間の類似性判定", 情報処理学会研究会報告、情報学基礎研究会 FI-66-16, 119–126 (2002).

[6] 松浦司, 金田康正: "近代日本小説家 8 人による文章の n-gram 分布を用いた著者判別", 情報処理学会研究会報告, 自然言語処理 NL-137-1, 1–8 (2000).

[7] 吉田篤弘, 延澤志保, 平石智宣, 斎藤博昭: "著者判別に有効な特徴量の推定", 情報処理学会研究会報告, 自然言語処理 NL-145-13, 83–90 (2001).

[8] Abdur Chowdhury, Ophir Frieder, David Grossman, and Mary Catherine McCabe: "Collection Statistics for Fast Duplicate Document Detection," ACM Trans. on Information Systems, 20(2), 171–191 (2002).

[9] Andrei Z.Broder, Steven C.Glassman, Mark S.Manasse, and Geoffrey Zweig: "Syntactic Clustering of the Web," Proc. of the Sixth International World Wide Web Conference, 391–404 (1997).

[10] Héctor García-Molina, Luis Gravano, and Narayanan Shivakumar: "dSCAM: Finding Document Copies Across Multiple Databases," Proc. of Fourth International Conference on Parallel and Distributed Information System, 68–79 (1996).

[11] Oren Zamir and Oren Etzioni: "Web Document Clustering: A Feasibility Demonstration," Proc. of SIGIR'98, 46–54 (1998).

[12] Reuters: "Reuters Corpus, Volume 1, English language, 1996-08-20 to 1997-08-19" (2000).

[13] Donna Harman and Mark Liberman: "TIPSTAR Complete" Linguistic Data Consortium (1993).

[14] 毎日新聞社1999. CD-毎日新聞 98 年版.

[15] 日本経済新聞社: "1996～2000 年版 日経全文記事データベース" (2001).

[16] National Center for Science Information Systems: "NTCIR Test Collection 1" (1999).

[17] 松本 裕治, 北内 啓, 山下 達雄, 平野 善隆, 松田 寛, 浅原 正幸: "日本語形態素解析システム「茶筌」Version 2.0 使用説明書 第 2 版", NAIST Technical Report, NAIST-IS-TR99012, 奈良先端科学技術大学院大学 (1999).