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

2006－ＦＩ－ 84 (10)
2006－ＮＬ－175 (10)
2006／9／12

LZ78の圧縮性を利用した文書検索手法の提案

木村 洋章　渡辺 俊典　古賀 久志　張 諾

電気通信大学 大学院 情報システム学研究科
〒182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail: [kimstar,watanabe,koga,zhang]@sd.is.uec.ac.jp

あらまし 著者らは情報の圧縮性に着目した新たなマルチメディアデータ解析手法の研究を進めている．PRDC(Pattern Representation Scheme using Data Compression)[1]と呼ぶこの新概念の中では，二つのデータX，Yの類似度を，それらを圧縮辞書D1，D2，…，Dnで圧縮した時の圧縮率ベクトルの類似度で判断する．本論文ではPRDCを用いた新文書検索システムの可能性を探る．部分的ではあるが，文書分類，公知／特異句抽出，文書要約，など将来の高自立・適応文書検索システムの実現に重要な機能を実現できる可能性を提示する．

キーワード 文書解析，情報検索，要約，新句抽出，データ圧縮

A New Document Retrieval Method Using LZ78 Compression Function

Hiroaki KIMURA, Toshinori WATANABE, Hisashi KOGA, and Nuo ZHANG

Graduate School of Information Systems, University of Electro-Communications
1-5-1, Chofugaoka, Chofu City, Tokyo, 182-8585 Japan
E-mail: [kimstar,watanabe,koga,zhang]@sd.is.uec.ac.jp

Abstract We have been studying a new multimedia data analysis scheme based on the concept of compressibility. In this new concept of PRDC (Pattern Representation Scheme using Data Compression)[1], we consider two data, let them X and Y, are similar if their compressibility vectors under a set of compression dictionaries D1, D2, ..., Dn are similar. Here we investigate the possibility of new document retrieval system using the PRDC. We prove that PRDC has possibilities to solve several fundamental problems including, document classification, common/distinguished phrase extraction, and summary, that should be realized in the future highly autonomous and adaptive document retrieval systems.

Key words Document analysis, Retrieval, Summarization, New phrase detection, Data compression

1 はじめに

膨大でかつ多種多様な表現形式をもつデジタルドキュメントからユーザーの欲する情報を検索する技術が身近なものになった．PCやWebサーバー上のドキュメントを検索するシステムとしてNamazu[2]などが有名である．キーワードを入力とする全文検索を瞬時に行える．キーワードの代りに文書自体を入力することで類似した文書群を検索する文書連想検索[3]が開発されている．キーワード検索は検索漏れが少ない分，多くのノイズを拾うため一般に適合率は低くなる．そこで文書を入力することで同じカテゴリーの文書群だけに的を絞り，適合率を高くしようとする試みもある．

本論文では，言語知識を用いずに文書分類を行えるPRDC[1]を足場とした新たな文書検索システムの可能性を探る．検索結果の提示のための分類カテゴリーの自動ラベリング，文書の特徴句の抽出と要約文生成，複数の話題をもった文書からの欲する話題の抽出・表示，などの基本的機能のPRDCでの実現可能性を検討する．

2 PRDC (Pattern Representation Scheme using Data Compression) 法

2.1. 原理

A, B ふたつのデータがあったとき, A で作られた圧縮辞書を用いて B を圧縮したとき, 高圧縮であれば A と B は似ていると言える. 複数個の圧縮辞書を用いて任意のデータを多次元の圧縮率ベクトルとして特徴表現してマルチメディア情報を分析する枠組みが PRDC[1]である.

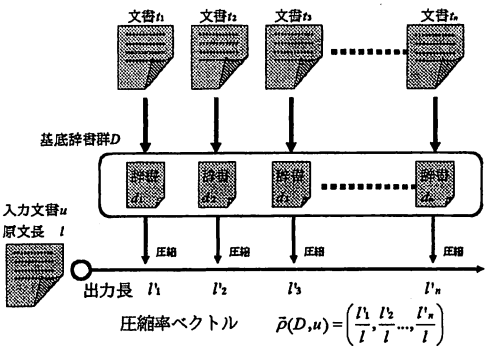

図 1： 圧縮率ベクトルの計算

入力データを圧縮率ベクトルへ置換する方法を図 1 で説明する. あらかじめ様々なジャンルの文書群 T={t1, t2, ..., tn}を収集しておき, 各文書を圧縮する過程で得られる圧縮辞書群 D={dt1, dt2, ..., dtn}を基底辞書とする. 入力データ u が与えられた時, 基底辞書により圧縮すると, 辞書数分の出力長が得られる. 各出力データ長を元のデータ長で割ったものを並べて圧縮率ベクトルとする. 元データの特徴は圧縮率ベクトルという形で数値化され, 分類や類似検索などをこのベクトルを用いて実現する.

2.2. 圧縮法について

圧縮率をできるだけ厳密に求めるためには, PRDC で用いる圧縮辞書は, ある長さ n 以下の全出現文字列(n, n-1, ..., 1-gram)を含むことが理想であるが, 計算量を低く抑えるため, LZ78 圧縮法[4][5]の辞書生成や圧縮機能を便宜的に利用している[1]. 通常の LZ 系圧縮法は元データの非損失復元を保証するための種々の機構を備えているが, PRDC では任意のデータを圧縮するための辞書出力と, 圧縮後のデータ長を計算できれば良く, 辞書番号の符号長最適化も不要である. よって, これらの機能を取り外したものを使用する.

テキストデータや PCM 音声ファイル, PPM 画像ファイルといった低レベル表現のデジタルデータは 8bit 単位で符号化されることが多いため, 本研究で使用する LZ78 圧縮では 8bit を入力単位とみなす. これが以下の実験で使用するデータに関する唯一の事前知識となる.

2.3. 基礎実験

PRDC による文書の自動分類性能を検証する.

2.3.1 異言語文書群の分類実験

実験データとして Wikipedia[6]から, 同じ内容を複数言語で表現したテキストデータを得た. 言語は, 日本語(SJIS)・中国語(GB2312)・英語(ASCII)・イタリア語(ASCII)であり, 内容はそれぞれの言語で書かれた「ミサイル」・「エントロピー」・「公理」の文書, 計 12 件である. 圧縮率ベクトルとそのクラスタリング結果を図 2 に示す. 圧縮辞書はこれらからランダムに選んだ 4 件のテキストから抽出したものを用いた.

	辞書1	辞書2	辞書3	辞書4
日・ミサイル	63.0	95.2	94.8	95.1
日・エントロピー	33.8	92.3	94.4	93.6
日・公理	54.8	95.6	97.7	97.6
中・ミサイル	93.3	68.2	85.4	86.4
中・エントロピー	96.0	35.6	98.7	96.3
中・公理	95.6	51.9	96.3	96.3
英・ミサイル	92.0	86.1	29.8	42.2
英・エントロピー	91.0	83.4	40.6	41.5
英・公理	92.9	84.1	39.2	42.4
伊・ミサイル	91.1	84.2	41.7	32.0
伊・エントロピー	89.1	83.7	44.8	24.7
伊・公理	91.3	84.1	44.8	29.9

(単位 %)

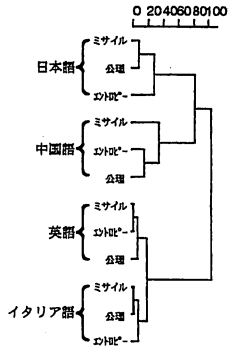

図 2： 異言語文書の分類

上位レベルで言語ごとに分類されていることがわかる. 圧縮率を見てみると, 基底辞書 1 が日本語であるため, 日本語テキストが良く圧縮され, 中・英・伊語の文書はあまり圧縮されていない. 同様のことが各言語に言える. この例では PRDC による分類は内容よりも表現言語の分離を優先することがわかる.

2.3.2 同言語, 異文字コード文書群の分類実験

2.3.1 で使用した日本語(SJIS)の文書 3 件を文字コード変換する. 変換する文字コードは JIS ・ EUC ・ UTF-8 とし, 計 12 個を用意した. クラスタリング結果を図 3 に示す.

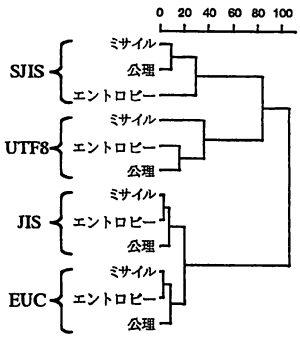

図 3 :異コード文書の分類

2.3.1 の結果と似て，文書内容よりも表現コードの分離を優先することがわかる．

2.3.3 同言語，同文字コード文書群の分類実験

日本語の迷惑メールを SJIS コードで表現したものを用いる．人が目視で選んだ迷惑メール 30 件から基底辞書を作成し，未知メール 69 件(内 38 が迷惑メール)の分類実験を行う，比較としてベイズ型学習迷惑メールフィルタを搭載した Mozilla Thunderbird[7]により，同様の実験を行った．30 個の迷惑メールを初期に学習させ，未知のメール 69 個の分類を行った．

表 1 :迷惑メールの分類

	PRDC	Thunderbird
全メール数	69	
全迷惑メール数	38	
迷惑メールと判定した数	33	35
判定の正答数	25	32
再現率	0.66	0.84
適合率	0.76	0.91

表 1 の結果より，PRDC による分類はベイズ法には劣るものの，文書を内容ごとに分類する能力を持つことがわかる．

2.4. 分類精度向上の可能性:不要語自動抽出

PRDC による迷惑メールの分類では，内容が異なる 2 つの文書の圧縮率が近くなる例が見られた．その原因究明のために，圧縮に寄与した文字列を調査した．図 4 にその一例を示す．

•ついて	•ている	•なけれ	•問題を
•それ	•れる	•もの	•関係
•でも	•ることが	•こと	•対して、
•ない	•るため	•可能	•場合の
			•考えら

図 4 :圧縮に寄与した文字列の一例

これらは日本語に頻繁に現れるが文書を特定する能力の低い，情報検索の世界で "不要語"と呼ばれるものである．伝統的手法では，不要語は文書からあらかじめ除去される[8]．上記の実験では，言語知識を持たない PRDC が不要語の使用頻度が似た文書を似ていると判断し，人の分類と異なる結果を出したと考えられる．よって，PRDC による検索システムの高性能化には不要語抽出と除去の機能が必要となることも判った．

2.5. 考察

以上，2.3.1，2.3.2 で，PRDC の言語分類，文字コード分類能力を確認し，2.3.3 で文書の内容分類能力を確認した．これらはすべて PRDC の圧縮率ベクトルをクラスタリング処理することで実現されている．

3 PRDC による文書の自動要約

ここでは，PRDC の枠組みだけを用いた文書自動要約の可能性を検討する．出発点において要約とは何か？という問題に直面する．そこでまず次の定義を導入する．

3.1. 文書要約機能の定義

文書の要約とは，その文書を適当に分割した断片の中から，その文書の論点を近似表現する少量の断片を選出することである．

3.2. 選出する断片

さまざまな選出法が想定される．ここでは以下を考える．

3.2.1 原文に相似な縮小文書の選出(サムネイル文書)

圧縮率ベクトルの特徴として，次のようなものがある．あるテキスト A の圧縮率ベクトルを $\rho(A)$とする．テキスト A を n 個の断片に分割したものを SEG(A,n)={sA1, sA2,...,sAn}とするとき，ベクトルの重心(Centroid) G(sA1, sA2,...,sAn) $\fallingdotseq \rho(A)$が成立する(図 5)．これは 2 つのテキスト sA1, sA2 を連接したテキスト sA1.sA2 の圧縮率ベクトル ρ(sA1.sA2) は ρ(sA1)と ρ(sA2)の重みつき線形和となるという定理[1]から得られる．

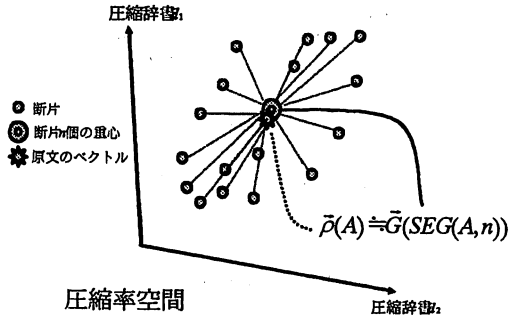

図 5 :圧縮率空間

図 6 はある日本語(SJIS, 4,270byte)を入力テキストとした断片数と重心誤差の関係を示したものである．断片数 n を

横軸，ρ(A)とG(SEG(A,n))との誤差を縦軸とした．

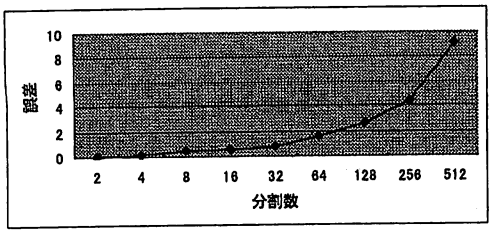

図6：断片数と重心誤差

512分割時においては一つの断片サイズが9byteとなるが誤差9程度にとどまっている．なお実験は5次元で行っており，誤差最大幅は223(1辺の長さが100の5次元立方体の対角線長)である．

上記より，圧縮率ベクトル空間では(n個の断片の重心ベクトル)≒(原文書の圧縮率ベクトル)が成立する．これを利用し，nより出来るだけ小さいmでG(SEG(A, m)) ≒ ρ(A) となるm個の断片集合SEG(A,m)を見出して原文Aの縮小版であるとみなせる．以下これをCentroid要約と呼ぶ(図7)．

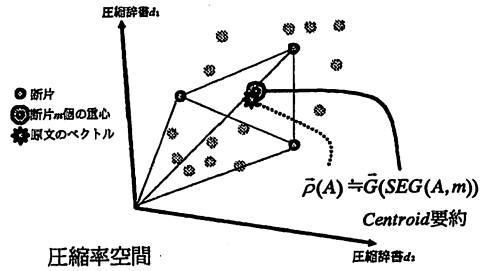

図7：Centroid要約

SEG(A,m)の探索は，全断片を重心ベクトルへの射影効果の高いものから順次ピックアップしてゆく発見的手法を用いる．

3.2.2 原文に固有な断片の選出

圧縮率ベクトルの特徴として，次のようなものがある．テキストAの圧縮率ベクトルρ(A)の要素{cvA1, cvA2, ..., cvAn}が原点Oに近い，即ちテキストAはどの基底辞書でもよく圧縮される場合，テキストAはごく一般的な内容であると考えられる．

次に，テキストBの圧縮率ベクトルρ(B)が原点Oから遠い，即ちBはどの基底辞書でも圧縮できない場合，新規性が高いか，異コードあるいは異言語の文書であると考えられる．

ここで簡単な実験を行う．無作為に選んだ5件の英語文書t1~t5を基底辞書とする．英語の一般的な単語を集めた不要語リスト[9]をテキストA，無作為に選んだ日本語文書をテキストB，さらに無作為に選んだ英語文書5個をテキストC,D,E,F,Gとして，テキストA~Gの圧縮率ベクトルを求める．結果を表2に示す．

表2：予備実験の結果

	辞書t0	辞書t1	辞書t2	辞書t3	辞書t4	辞書t5	原点との距離
A	32.6	32.4	30.7	31.3	31.4	32.7	78.0
B	98.7	98.7	98.7	98.7	98.7	98.7	241.6
C	37.2	34.8	34.8	34.4	34.7	34.3	85.8
D	37.6	35.2	34.7	34.6	35.0	35.3	86.8
E	36.6	34.5	34.2	34.0	34.1	34.3	84.8
F	38.0	36.6	36.3	35.9	36.3	36.4	89.6
G	36.1	34.4	34.4	34.1	34.4	34.1	84.7

圧縮率ベクトルが原点に近いもの(A)は，どの文書にもよく出現する不要語であることがわかる．

日本語テキストBの圧縮率ベクトルは原点から最遠位置にある．基底辞書t1~t5はすべて英語であるため，英語世界にとって未知の日本語は新規性が高いとみなされている．

次に，代表的な断片を取り出すために，ある一つの文書から得られた全断片の圧縮率ベクトル集合の凸包の頂点に着目する．

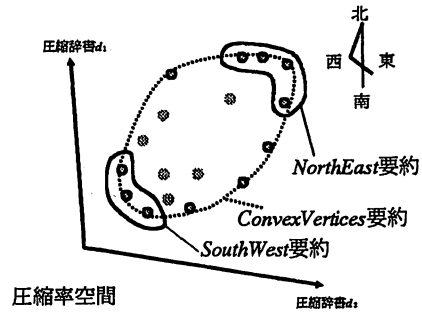

図8：3種の要約法

簡単のために2次元の場合を考察すると，頂点断片は周囲の断片を結ぶ線分上にない．もし線分上にあれば，両端点の重み付き和で説明される(両端点断片の内容のミクスチャ)断片となり，代表的とは言えなくなる．逆に端点を代表として採用しておけば，それらを結ぶ線分上の断片をすべて重み付き和で説明できる．よって凸包の端点断片のみを考察対象とし，それらの位置に着目して3タイプの要約を定義する．

SW(SouthWest)要約：図8の原点に近い断片集合であり，多くの文書で多用される句を含み，一般性の高い内容

である.

NE(NorthEast)要約:図 8 の原点から最遠付近の断片集合であり, その文書にだけ現れる新規性の高い句を含む.

CV(ConvexVertices)要約:図 8 の凸包端点を網羅的に選出した断片集合であり, 上記の SW, NE などを全て含む.

索引語の重み付けに関して TF・IDF 概念がある[10]. TF(term frequency)は上記の SW 要約, IDF(inverse document frequency)は NE 要約と関連する概念である.

なお, 凸包端点の探索は, ランダムに発生させた法線ベクトルを持つ超平面で凸包をはさみこむことを繰り返す確率的手法を用いる. このため, 突出度の高い端点ほど繰り返し多数回, 発見される傾向を持つ.

3.3. 考察

以上, 合計 4 種類の断片抽出法を導入した. これらのうち要約能力の高いものは NE 要約と CV 要約であると予想される. これらは人の関心を引く新規性の強い文書断片を提示するからである. 逆に SW 要約は, 公知内容や不要語を提示するため, あまり好まれないと予想する.

3.4. 要約実験

人工的に作成した擬似文書と実文書とを用いた実験をおこなう. 原文の何%程度の要約文書を出力するかを指示する要約率をユーザが指定する. 与えた文書を断片化し, それぞれの圧縮率ベクトルを求める. その後, 凸包端点断片を探索する. 端点は発見回数順に並べ, 上位から要約率を満たすまで採用する.

3.4.1 擬似文書による実験

松崎等の方法で擬似言語の文書データを作成した[11]. アスキーコード中の表示可能記号 0x20 ～ 0x40 , 0x61 ～ 0x7A(小文字アルファベット)を文字集合 C とする. 一文書につき 10,000byte を文字集合 C からランダムに抽出し, 100 個の文書を生成した. 続いて擬似キーワード, 不要語を上書きにより挿入した. 表 3 に擬似的キーワードを示す. 出現頻度とは 100 文書中, 当該キーワードを含む文書数である.

表 3 :擬似文書の性質

	全文書中の出現確率	1文書中の出現回数
ALFA	100%	20回
BRAVO	100%	1回
CHARLIE	20%	20回
DELTA	70%	40回
ECHO	30%	2回
FOXTROT	10%	5回

ALFA のように全文中に多く出現するキーワードは不要語を模擬しており, ECHO のように一部の文書に多く出現するキーワードは重要な新規語を模擬している.

要約率 5%, 断片サイズを 16byte とする要約実験を行い擬似キーワードの再現率を 4 つの要約タイプで求めた. その結果, すべての擬似キーワード含む断片を検出できることがわかった. 表 4 に各擬似キーワードの再現率を示す.

表 4 :擬似キーワードの再現率

擬似キーワード \ 要約タイプ	SW	NE	Cent.	CV
ALFA	100%	4%	100%	100%
BRAVO	3%	28%	3%	16%
CHARLIE	5%	100%	100%	100%
DELTA	72%	84%	99%	100%
ECHO	7%	67%	0%	74%
FOXTROT	10%	80%	0%	100%

結果は以下のように予想を裏付けるものとなった.

SW 要約は全文中に出現する不要語キーワード抽出し, 新規語を抽出しない.

NE 要約は新規語を抽出し, 不要語を抽出しない.

Centroid 要約は全文書中での文書の出現頻度によらず, 一文書中の出現回数が多いキーワードを抽出する傾向がある.

CV 要約は不要語と新規語をともに抽出する.

よって, 原文の全体的な内容が知りたい場合は, 一般的な内容と新規性の高い内容を両方抽出する Centroid 要約と CV 要約が有効と思われる。他の文書にない新規な断片を得たい場合は, NE 要約が有効であり, 逆に常識的, 一般的な内容や, 不要語を得たい場合は SW 要約が有効である.

3.4.2 日本語実文書による実験

WEB のニュースサイトから無作為に選んだ 12 件の記事 (1 文書あたり 500 ～ 4000 字)を用いた. 各文書から上記 4 タイプの要約文を生成し, 被験者 8 人による主観評価を行なった.

各被験者には 1 つの原文記事と, 上記 4 タイプの要約文が提示される. 各タイプでの要約の善し悪しを 5 段階評価する. さらに 4 タイプのうち最も良いタイプを一つだけ選ぶ.

12 個の記事に対して上記を繰り返した. 最後に各被験者からの総評を得た.

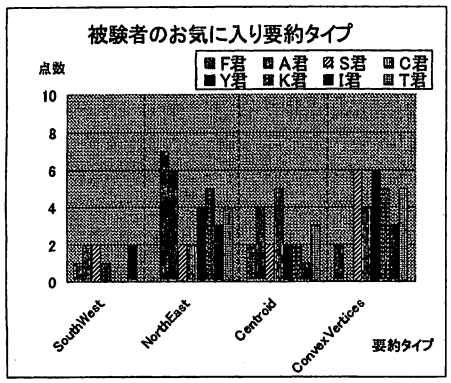

図 9 :実文書の要約評価結果

図 9 にタイプ別の評価結果と好みの結果をグラフで示す. 被験者によって評価にばらつきがあるが, SW 要約が悪く, NE および CV 要約の評価が良かった. 総評でも「SW 要約は何をいってるのかわからない」,「NE 要約や CV 要約が個人的には好きだ」などの意見が多数を占めた.

4 まとめ

情報の爆発的増加に見舞われる現代において検索システムの重要性は論を待たない. 我々は, 膨大かつ時間とともに変化する情報を対象とする検索システムには例えば下記のような機能が必要と考えるが, 未だ十分には実現されていない.

(1)辞書作成などの人手を要せず, 統一かつ簡便な枠組みで分類や認識を実現すること, (2)新規情報の自動検出を実現できること, (3)分類カテゴリを代表する文書の要約が実現できること.

本論文では, 圧縮性に基づいて情報を特徴付けるパラダイムである PRDC を用いて, これらの問題を解決する可能性を探った. その結果, (1)については, 文書の圧縮性特徴量の利用という単一の原理で文書分類や類似性判定を, 言語解析知識や人手を使用せずに実現する可能性を備えている, (2)については, 既存文書から自動抽出される基底辞書による圧縮空間で分類できない情報を新規であると自動判定できる, (3)については, 文書の NE 要約などで対応できる, などの有望な結果を得ることができた. これらの結果を踏まえ, 自立性、適応性の高い文書検索システムの可能性を引き続き探究したい.

謝辞

本研究の推進において, 文書表現の母体となるコード体系の差異が PRDC の性能に及ぼす影響についてコメントやアドバイスをいただいた北村浩客員助教授(NEC ネットワークス研究所) に感謝します.

参考文献

[1] Toshinori Watanabe, Ken Sugawara, and Hiroshi Sugihara, "A New Pattern Representation Scheme Using Data Compression", IEEE TPAMI, Vol.24, No.5, MAY.2002.

[2]「全文検索システム Namazu 」,"http://www.namazu.org/"

[3] 高野明彦, 丹羽芳樹, 西岡真吾, 岩山真, 今一修, 久光徹, 「汎用連想計算エンジン GETA 」,"http://geta.ex.nii.ac.jp/"

[4] Jacob Ziv and Abraham Lempel; A Universal Algorithm for Sequential Data Compression, IEEE Transactions on Information Theory, May 1977.

[5] 植松友彦, 文書データ圧縮アルゴリズム 入門, CQ 出版社, 1994.

[6] Wikipedia, "http://www.wikipedia.org/"

[7] Mozilla.org, "http://www.mozilla.org/"

[8] 北研二, 津田和彦, 獅子堀正幹, 情報検索アルゴリズム, 共立出版, 2002.

[9] freeWAIS-sf 2.x & SFgate 5.x User Guide, "http://www-fog.bio.unipd.it/waishelp/waishlp3.html"

[10] Salton, G. and Buckley, C.: "Term-weighting approaches in automatic text retrieval", Information Processing and Management, 24, pp.513-523, 1988.

[11] 松崎大輔, 渡辺俊典, 古賀久志, 張 諾, "圧縮性に着目した文書の関係分析手法", 情報処理学会 第 84 回情報学基礎研究会.