Title	圧縮プログラムを応用した著者推定
Sub Title	Authorship attribution by data compression program
Author	安形, 輝(Agata, Teru)
Publisher	三田図書館・情報学会
Publication year	2005
Jtitle	Library and information science No.54 (2005.) ,p.1- 18
Abstract	Benedetto et al. recently confirmed the validity of a method for measuring similarity using data compression software. Despite its potential, this method has not yet been applied to the field of information science. The present study proposes the use of CIR, a modified method that uses an improved ratio of compression, and describes two experiments on authorship attribution using data from modern Japanese literature. The first experiment compares the results of applying CIR and Benedetto's method to test collections of modified data (fixed length) using a procedure similar to that described by Matsuura et al. The second experiment is based on original data (variable length). The first experiment showed an average precision rate of 97.7% for CIR, while Benedetto's method gave a rate of 90.5%. The CIR method proves to be an improvement on the best method described by Matsuura et al. The second experiment confirmed the e
Notes	原著論文
Genre	Journal Article
URL	http://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00003152-00000054-0001

Library and Information Science No. 54 2005

原著論文

圧縮プログラムを応用した著者推定

Authorship Attribution by Data Compression Program

安　形　　輝
Teru AGATA

Résumé

Benedetto et al. recently confirmed the validity of a method for measuring similarity using data compression software. Despite its potential, this method has not yet been applied to the field of information science. The present study proposes the use of CIR, a modified method that uses an improved ratio of compression, and describes two experiments on authorship attribution using data from modern Japanese literature. The first experiment compares the results of applying CIR and Benedetto's method to test collections of modified data (fixed length) using a procedure similar to that described by Matsuura et al. The second experiment is based on original data (variable length).

The first experiment showed an average precision rate of 97.7% for CIR, while Benedetto's method gave a rate of 90.5%. The CIR method proves to be an improvement on the best method described by Matsuura et al. The second experiment confirmed the effectiveness of the CIR method, giving an average precision rate of 95.7%.

安形　輝：亜細亜大学，東京都武蔵野市境 5–24–10
Teru AGATA: Asia University, 5–24–10 Sakai, Musashino-shi, Tokyo
e-mail: agata@asia-u.ac.jp
受付日：2005 年 6 月 6 日　改訂稿受付日：2005 年 9 月 13 日　受理日：2005 年 10 月 30 日

I. はじめに

A. 圧縮プログラムを応用した類似データ同定

情報検索では検索式とデータ間の類似度を測定し類似度順に出力を行い，自動分類ではカテゴリとデータ間の類似度からカテゴライゼーションやデータ同士の近さからクラスタリングを行う。情報検索や自動分類が扱うデータ種は基本的にテキストデータであるため，言語的な特性を基盤とするものが多い。しかしながら，データの類似度を測定する手法には，言語的な特性からの処理を必要としないものも存在する。そのような手法の一つとして，圧縮プログラムを用いた類似データ同定手法がある。

本来，圧縮プログラムあるいはアーカイバは，データ中の冗長な部分を識別し，より短いデータに置き換えることによって全体のサイズを縮小し，外部記憶装置に占める容量を節約したり，あるいは，通信にかかる時間を短縮したりすることを目的としている。近年，圧縮プログラムを本来の圧縮用途ではなく，類似データの識別に応用する研究が行われている。

圧縮プログラムを応用した類似データの同定手法の基本的な考え方は，非常にシンプルなものである。二つのデータがあったときに，データ同士が類似していればしているほど，共通する冗長な部分が多くなると考えられる。そこで，ある二つのデータを連結する（二つのデータを単純に並置し，一つのファイルとする）ときに，圧縮プログラムがその連結データをより高い圧縮率で圧縮できるほど，つまり，生成される圧縮ファイルのサイズが小さくなればなるほど，その二つのデータは類似しているということになる。実際には，この考え方に，個別のデータごとのデータ単体での圧縮されやすさを考慮し，何らかの操作を加えることとなる。

圧縮プログラムを応用した類似データの同定には，以下のような特徴がある。

1) 一般的な圧縮プログラムを利用するため導入コストが低い。
2) テキストデータだけでなく，画像データやDNA 配列データなど，種類にかかわらず応用可能である。
3) 圧縮という計算上，非常に時間のかかる処理を行うため，大規模データには向かない。

この手法に関する最も有名な研究として，Dario Benedetto らの “Language Trees and Zipping”[1] がある。これは米国物理学会の著名な速報誌である *Physical Review Letters* 誌 2002 年 1 月 28 日号に掲載されたものである。

この文献中で，彼らは ZIP 系列の圧縮プログラムによる自動分類や類似データの同定手法を提案し（以下，Benedetto らの手法），DNA 配列の類

似度測定，言語不明データの言語識別，著者不明データの著者推定に関する実験を行った結果を簡単に紹介している。著者推定に関しては 90 文献[2]から構成されるコーパスに対して著者推定実験を行い，93.3% という高い精度を得ている。しかし，実験環境に関して詳細な記述がなく，著者推定に関する既往研究と同様の実験データを用いてもいない。そのため，実験結果の比較をすることができず，さらに彼らの行った実験の再現も難しい。

その後，同誌において手法自体の新奇性などをめぐる議論[3), 4)]が掲載された。また，この手法を磁性体のバルクハウゼンノイズの解析に用いる[5)]など，他分野での応用も活発に行われている。さらに，一般的なプログラミング雑誌である *C Magazine*[6)] やインターネットやコンピュータの話題を中心としたオンライン誌である *Wired News*[7)] で紹介されるなど，この研究は一般誌で取り上げられるほどに注目を集めてきた。しかし，掲載誌が *Physical Review Letters* 誌であったためか，今日まで，情報学分野での応用研究は少ない。

O. V. Kukushkina らは，Benedetto らに先んじて 2001 年に同様の手法で圧縮プログラムを応用したテキストの自動分類に関する実験を行っている[8)]。実験結果では，最も精度の高い圧縮プログラムは，マルコフ連鎖を応用した手法に匹敵する高精度を示した。しかし，この実験自体は，彼らが提案した手法の有効性を検証するためのものであり，マルコフ連鎖を応用した彼らの手法の記述に重点が置かれている。そのため，圧縮プログラムを応用した類似データの同定手法に関しては，付録中に参考程度に記述されているのみである。また，ロシア語文献であったため，認知度はそれほど高くなかったと考えられる。

日本語データに関しては，内山和也[9)] が Benedetto らの手法を用いて 7 人の書き手による日本語学術論文 34 件の原著者推定を行っている。Benedetto らの手法を用いた実験では，著者推定に関しては高い精度が得られた。一方で，テーマ別の識別実験では，“意味論的な識別に用いうるとする主張は，疑わしいもの”と結論づけている。Benedetto らの手法を日本語データに対して用いた点，意味的な分類への応用可能性を検討した点は評価できる。しかしながら，独自の小規模データに対して実験を行っているため，既往研究との比較ができず，その実験集合の構築方法が明らかでないため，実験を再現することができないという問題がある。

B. 著者推定に関する研究

本研究では，圧縮プログラムを応用した類似データの同定手法の検証を行うための実験対象として近代日本文学データを用いた著者推定を扱う。

著者推定とは，作者不明のデータがあった場合にデータの特徴から著者を推定することであり，計量文体学を中心として，コンピュータの登場以前から様々な手法が提案されており[10)]，継続的に研究がなされてきた比較的活発な研究領域といえる。

図書館・情報学との関係からみると，著者推定は文体的特徴から類似データを識別するが，これは情報検索や自動分類と共通の枠組みを持つといえ，その研究成果は互いに応用可能な場合が多い。例えば，佐藤進也ら[11)]はウェブ上の情報源間の自動分類に著名な著者判定手法である Tankard の手法[12)]を応用している。

また，図書館が扱う資料には，著作者が不明である文献や，作品群の著作者の同一性が問題となっている文献が少なからず存在している。前者の例としては旧約聖書の著作者推定があり[13)]，後者の例としては日蓮が本当に著したのかが疑わしいとされている文献の真贋判定[14)]が挙げられる。

著者推定や真贋判定は，特に文学研究において重要な研究領域の一つであるが，それだけでなく，著名作家の未公開作の発見時の真贋鑑定[15)]，裁判における被告人の上申書と日記の作成者の同一性の検証[16)]といった著者推定の応用事例は，学術面からだけでなく実社会からの需要も高いことを示していると考えられる。

さらに，著者推定の応用領域はインターネット

第 1 表　計量文体学で用いられてきた文体的指標

		指　　標	ケニィ	村上	ホームズ	吉岡	安本	陳
			1982	1994	1994	1996	1977	2003
量	長さ	文の長さ	○	○	○	○	○	○
		単語の長さ	○	○	○	○	○	○
		音節			○			
	生起回数	単語の出現率	○	○	○	○		
		同義語		○				
		異なり語		○	○	○		
		漢字					○	○
		名詞					○	○
		接続詞						○
		接続助詞						○
		四字熟語						○
		人格語					○	○
		多出語		○		○		
		句点					○	○
		読点		○				
構文		主語，熟語，修飾語などの構文に関する情報		○				
位置	文頭	文頭に置かれる単語や品詞の出現率		○				
	文中	読点の位置		○				
	文末	文末に置かれる単語や品詞の出現率		○				
		過去止					○	○
		現在止					○	○
		不定止					○	○
表現		直喩					○	○
		声喩					○	○
		色彩語					○	○
		会話文					○	○
内容	話題							○
	引用							○

出典：石田栄美ほか 4 名．“文体からみた学術的文献の特徴分析”．2004 年度三田図書館・情報学会研究大会発表論文集，2004, p. 33

上のメールやウェブ情報源まで拡大しつつある。例えば，著者推定に機械学習手法 (Support Vector Machine) を用いた坪井祐太らの研究[17] では，メーリングリスト上のデータで学習を行い，ウェブ文書の著者推定を行っている。また，スパムメールやウェブスパム（Google のページ

ランクをあげるためのダミーページによる強リンクネットワークの構築）への対策としての著者推定も考えることができる。大手のスパムメール報告サイトの一つである SpamCop.net によれば，2004 年に報告されたスパムメールだけでも（当然報告されないスパムメールはさらに多く存在すると思われる），約 2.7 億通[18] という莫大な数であった。スパムメールはメールアドレスなどを偽装しており作成元の特定が困難な場合が多いが，本文の作成者の推定つまり著者推定が可能となれば，著者によるフィルタリングも可能となると考えられる。

石田栄美ら[19] は計量文体学の代表的な既往研究[20] において使われてきた文体的指標を，“量，構文，位置，表現，内容に関する指標に”分類し，第 1 表のようにまとめている。この表からは，計量文体学では多くの研究が文長，語長，語の出現率という解析手法，つまり，何らかの言語的，構造的，内容的な解析を必要とする手法を用いてきたことがわかる。例えば，古典的かつある程度の精度が得られる著者推定手法として，文の長さからの推定手法がある。この手法は最も簡便な推定手法の一つと考えられるが，それでも句読点や改行を手がかりに文の終端を識別する必要がある。

しかし，圧縮プログラムを応用した類似データの同定手法の場合は，テキストデータの言語，構造，内容を解析せずに，データをデータとして圧縮プログラムに投入する。そのため，どのような言語，構造，内容でも，さらにはテキストデータ以外にも対応可能となり，応用範囲は広く，汎用性が高い手法といえる。

C. 本研究の目的

本研究では，圧縮改善係数による類似データ同定手法（以下，圧縮改善係数による手法）を提案し，その有効性を検証することを目的としている。当初，本研究で検証を行う手法としては，Benedetto らの手法を用いる予定であったが，しかし，予備的な実験から二つの問題点が明らかとなったため，それらの問題点を解消した新たな圧縮改善係数による手法を提案した。

圧縮改善係数による手法の有効性の検証を目的とし，著者推定に関する実験を行っている。著者推定を実験対象として選択した理由は，内山の研究で Benedetto らの手法がテーマ別の識別よりも著者推定に対してより有効であることが指摘されており，ほぼ同様の性質を有する今回の手法の検証に適切であると考えたためである。

著者推定実験は (1) データのサイズを揃えた固定長データ，(2) 特に操作を行っていないオリジナルデータ，の二つを対象として行った。

前者 (1) の固定長データを用いた実験の目的は，既往研究と同じ環境で実験を行い，すでに有効性が認められている他の著者推定手法との比較を行うことである。日本語文献の著者推定に関する研究は，計量文体学の領域で数多くなされているが，

1) 既存の複数の手法の結果を残していること。
2) 実験用データが入手可能であること。

という二つの理由から，松浦司らによる“近代日本文学者 8 人による文章における文字 *N*-gram 分布を手がかりとする著者推定”(1999)[21] “*n*-gram の分布を利用した近代日本語文の著者推定”(2000)[22] という一連の研究を比較の対象とした。この研究中で実験が行われている著者推定手法は，松浦らが提案した非類似度評価関数 dissim, Tankard の手法，最低基準としてのダイバージェンス手法である。

この固定長データ実験集合群に対しては，二つの追加的な実験を行った。まず，第一にこの手法がサイズの小さいデータでも有効かを見るために，手がかりとなるデータのサイズを短くしていった場合に，性能がどのような形で劣化していくかを Benedetto らの手法と比較して分析した。第二に，圧縮率の変化と性能の関係を見るために圧縮プログラムの圧縮レベルを変化させた場合に性能がどのように変化するかを分析した。

後者 (2) の実験は，特に操作を加えておらずデータのサイズが統一されていないオリジナル

データに対して，この手法が有効であるかを検証するために行う。固定長データを用いた実験(1)では，松浦らの研究との比較を行うため，実験環境を揃えている。(2)ではその過程を省き，インターネット上から入手できるオリジナルデータをそのまま用いることで，このような手法が実際に応用される環境においてどの程度の性能で著者推定が可能であるかをみる。

II. 圧縮プログラムによる類似データの同定

圧縮プログラムを応用した類似データの同定は，二つのデータに共通する部分が多いほど，二つのデータを単純に並置し，一つのファイルとしたデータ（連結データ）を圧縮プログラムに投入したときに出力される圧縮ファイルのサイズが小さくなる性質を利用して行われる。ただし，個別のデータ単体での圧縮のされやすさが影響するため，その影響を考慮に入れた処理を行うこととなる。

A. Benedetto らの手法

Benedetto らの手法では，あるデータ（基準データ）と比較したいデータ（比較データ）があったときに，二つのデータを連結したときの圧縮ファイルのサイズから，比較データの圧縮ファイルのサイズの差をとることで，類似度算出を行う。このファイルサイズの差が小さいほど類似度が高いものとしている。

類似度に影響を与える要因は，連結データの圧縮サイズと比較データの圧縮サイズである。前者は二つのデータの共通部分が多いほど小さくなり，後者は比較データが圧縮されにくいほど大きくなる。つまり，大まかに意味づけを行うならば，単体では圧縮されにくい比較データを連結することで圧縮サイズが小さくなるならば，その二つのデータは類似度が高くなる，と解釈できる。

Benedetto らの手法による類似度順出力の具体的な手順は以下のとおりである。

1) 基準データ X，比較データ A_i があるとき，候補となるすべての比較データ A_i について，A_i と X の連結データを作成する。
2) 比較データ A_i 単体，比較データ A_i と基準データ X の連結データから圧縮ファイルをそれぞれ作成する。
3) LZ_{A_i+X} を，連結データの圧縮ファイルのサイズ，LZ_{A_i} を比較データ A_i 単体で圧縮したファイルのサイズとしたときに，$LZ_{A_i+X}-LZ_{A_i}$ を算出する。
4) 値の小さな順に比較データ A_i を出力する。

この手法では，連結データを圧縮したサイズと比較データを圧縮したサイズの差が小さい順に並び替えることで，類似度順出力を実現している。しかし，この手法を用いた予備的な実験からは，

1) 比較データだけでなく基準データの単体での圧縮されやすさがデータを連結したもののサイズに影響すること。
2) 連結データを連結する順序が圧縮サイズに影響すること。

の二つの問題点が明らかとなった。

B. 圧縮改善係数からの推定手法

Benedetto らの手法の二つの問題点を考慮し，連結データの圧縮率からデータ単体での圧縮率の影響とデータの連結順序の影響を排除する目的で，以下の数式で表される圧縮改善係数を考案した[23]。

$$\text{圧縮改善係数}=\left(\frac{LZ_X}{L_X}+\frac{LZ_{A_i}}{L_{A_i}}\right)-\left(\frac{LZ_{X+A_i}+LZ_{A_i+X}}{L_{X+A_i}}\right) \quad (1)$$

ここで，L はファイルサイズを示し，L_X は基準データ X のファイルサイズを，L_{X+A_i} は基準データ X と比較データ A_i を連結したファイルサイズを表している。LZ は圧縮ファイルのサイズを示しており，LZ_X は X を圧縮した場合のファイルサイズを，LZ_{X+A_i} は基準データ X を先に，比較

データ A_i を後として連結した場合の圧縮ファイルサイズを，LZ_{A_i+X} は逆に連結した場合の圧縮ファイルサイズをそれぞれ表している。

式 (1) は，前半が各データ単体での圧縮されやすさを，後半が連結データの圧縮されやすさを表現しており，全体として，データ単体と比較してデータを連結したことで，どの程度，圧縮率が上がったかを表している。後半部で LZ_{X+A_i} と LZ_{A_i+X} の二つを算出する理由は，圧縮プログラムのアルゴリズムと実装（バッファの大きさなど）を考慮した場合に，二つのデータをどの順序で投入するかが与える影響を排除するためである。

この式 (1) を基準データ，比較データのサイズが異なった場合を考慮に入れて改良したものが，式 (2) である。

圧縮改善係数

$$=2\cdot\left(\frac{LZ_X}{L_X}\cdot\frac{L_X}{L_{X+A_i}}+\frac{LZ_{A_i}}{L_{A_i}}\cdot\frac{L_{A_i}}{L_{X+A_i}}\right)-\left(\frac{LZ_{X+A_i}+LZ_{A_i+X}}{L_{X+A_i}}\right)$$

$$=2\cdot\frac{LZ_X+LZ_{A_i}}{L_{X+A_i}}-\frac{LZ_{X+A_i}+LZ_{A_i}{}^{+X}}{L_{X+A_i}} \quad (2)$$

以下の実験ではこの式 (2) を採用している。式 (2) は，式 (1) の前半部をファイルサイズで正規化することで，サイズが異なる場合にも対応させたものである。

圧縮改善係数はデータを連結したときの圧縮されやすさがデータ単体と比較してどの程度改善されたかを示しており，この値が高ければ高いほど，類似度が高いことを意味している。そのため，あるデータに対する類似度順の出力は，基準データと各比較データのすべての組み合わせについて圧縮改善係数を算出し，値が高いものから順に比較データを並べるという手順となる。

圧縮プログラムに投入するデータが，Benedetto らの手法では比較データと基準データの連結データおよび比較データ単体の二つであったのに対し，圧縮改善係数による手法では比較データと基準データを連結したもの，その逆順に連結したもの，比較データ単体，基準データ単体の四つとなり倍増している。そのため，単純に考えれば，圧縮改善係数による手法は，計算処理上，負荷がとても高いといえる。

しかし，圧縮プログラムによる処理は時間がかかるため，実際にはすべてのデータ単体とすべての組み合わせの連結データに対して圧縮サイズの算出を行ったのちに類似度の算出を行うことになる。これはブール型情報検索システムにおいてあらかじめ索引ファイルを作成しておくのと同様の処理といえる。結果として，計算処理上の負荷はどちらの手法でも実質的に同程度となる。

C. 圧縮プログラムを応用したシステム

1. Zip 形式

Benedetto らによる手法あるいは圧縮改善係数を用いた手法のどちらでも，圧縮サイズが得られるならばどのような圧縮プログラムであれ応用可能である。

しかし，圧縮プログラムを応用した類似データの同定手法は，原理上，圧縮率の高い圧縮プログラムを用いるほど，類似データの識別力が高くなると考えられる。そのため，どの圧縮プログラムを採用するかが重要となる。ただし，圧縮プログラムの性能は対象データの種類や特性によって変化するため，データの種類や特性に合わせた圧縮プログラムを用いる必要がある。

また，本手法の重要な特徴として，実装が容易であるという利点がある。例えば，Benedetto らの手法は Perl 言語では 10 行程度のプログラムで実現できるほど応用が容易である[23]とされている。この特性を生かすためには，汎用性の高い圧縮プログラムを採用することが適当であると考えられる。

本研究では圧縮プログラムとして Zip 形式を選択したが，その主な理由は以下のとおりである。

1) Zip はテキストデータに対する圧縮率が高いとされる。
2) Zip は MS-Windows で初期装備されるなど，最も標準的な圧縮形式で実装が容易で

ある。

3) Benedetto の手法を用いた先行研究の多くで採用している圧縮形式である。

2. Zip の原理

Zip の圧縮処理は，入力されたデータに対して，まず LZ77 符合化を用い，その結果に対してさらに Huffman 符号化を行うという二段階で行われる。テキストデータ用に考案された圧縮アルゴリズムを組み合わせて使うことで非常に高い圧縮率を実現している。ここで，二つの圧縮アルゴリズムについて簡単に触れておく。

LZ77 符号化[24] は辞書を使った圧縮アルゴリズムとして最も有名なものの一つである。繰り返し出現するデータ列をより短いデータ列で置き換える辞書式圧縮はテキストデータに向き，高い圧縮率が得られるが，大量のデータを扱う場合や短期記憶が少ない場合に辞書データを保持する方法が問題になってくる。そこで，読み込んだデータ列でバッファに入っているものを辞書として扱うことで，その問題を解消したのが LZ77 符号化である。圧縮時には，データを読み込むにつれて辞書とする領域がスライドするため，スライド辞書法とも呼ばれる。

Huffman 符号化[25] の基本的な原理は以下のとおりである。一般的なテキストデータにおいて，各文字を表現するビット数は同じである。たとえば，ASCII コードにおいて文字は 8 ビット，日本語のシフト JIS コードであれば 16 ビットである。頻繁に現れる文字をより短いビット数で表現し，一方で，あまり出現しない文字をより長いビット数で表現すれば，結果として，全体のサイズを小さくすることができる。このような考え方に基づいて，全体が最も小さくなるような符号を求めるのが Huffman のアルゴリズムである。

ただし，Huffman 符号化をそのまま実装すると，データを最初に読み込み，各文字の表現ビット数を計算し，その結果に基づき圧縮を行うためにもう一度最初からデータを読み込む必要がある。そのため，Zip ではデータを読み込みながら Huffman 木を構築していく動的 Huffman 符号化のバリエーションが用いられることが多い（今回の実験で用いたシステムの Zip 実装には，このバリエーションの一つが用いられている）。

3. Zip の実装と設定

Zip 形式は定評がある有名な圧縮形式であるため，いくつかの亜種が存在するが，今回のシステム構築においては，Java 言語の開発環境 J2SDK[26] に付属するクラスライブラリである java.util.zip 以下のクラスを用いた。これらの実装は RFC 1950[27], RFC 1951[28] に準拠した Info-Zip[29] の動的ライブラリを元にしている。

Zip の片翼を担う圧縮アルゴリズムである LZ77 符号化では，スライド辞書や最大データ長のバッファサイズによって圧縮率が変化する。一般に，それらのサイズを大きくすると圧縮率は高くなる一方で，圧縮に時間がかかるようになる。また，辞書内の位置情報を表現するための符号のサイズも大きくなるため，結果的に圧縮率が下がる結果となってしまう。そのため，実装ではスライド辞書や最大データ長のサイズが無制限に大きくとられることはない。

java.util.zip の圧縮用クラスライブラリでは，圧縮レベルを 0 から 9 に設定可能である。圧縮レベルが 0 の場合，圧縮をせずにデータをそのまま格納するため，このレベルは考慮しない。圧縮レベル 1 から 9 については，レベル 1 では圧縮速度が最も速いが圧縮率が最も低く，レベル 9 では圧縮速度が最も遅いが圧縮率が高くなるとされている。このレベル分けはバッファサイズの大きさに差をつけることで行われている。今回の設定では圧縮レベルを分析した一部の実験を除き，最も圧縮率が高いとされるレベル 9 に設定した。LZ77 符号化において圧縮率に影響を与えるスライド辞書のサイズは Info-Zip の動的ライブラリでは圧縮レベル 9 で 32K バイトとなっている。

III. 既往研究との比較実験

A. 実験環境

1. 実験テキスト

実験対象データ集合の構築は，松浦らの研究と

まったく同じ手順によって行った。ただし，ここで作成された実験集合群は，構築手順の一部に無作為な選択を含む部分があるため，まったく同じ実験集合群ではなく，ほぼ同じ性質を有すると考えられる実験集合群となる。

本研究で実験集合群構築に用いたのは，著作権の切れた作品のデジタル化を行っている青空文庫[30]から入手した，岡本綺堂，芥川龍之介，梶井基次郎，菊池寛，国木田独歩，水野仙子，樋口一葉，有島武郎の 8 人の近代日本文学者による 92 作品のテキストデータである。

各作品データについては，著者推定実験に用いるため，本文以外の著者，タイトル，執筆年月日などの書誌事項を除去した。また，先行研究と同様に，改行，空白は原則として一文字としているが，改行後の空白は冗長であるため除去し，半角英数記号は全角に変換している。

これらの作品は明治から昭和初期にかけて執筆された作品であり，歴史的仮名遣いで書かれたものと現代仮名遣いに改めた作品が混在しているが，手法の頑強性をも検証するため，先行研究と同様に，あえて統一はしていない。同一著者内で歴史的仮名遣いと現代仮名遣いの作品が混在するものは，芥川龍之介，有島武郎，国木田独歩の 3 著者であり，他の著者については，すべての作品が歴史的仮名遣い，現代仮名遣いのどちらかであった。

使用した全 92 作品は，72 本の小説，9 本のエッセイ，5 本の書簡形式文章，3 本の戯曲，2 本の日記，1 本の談話から構成される（第 2 表）。

2. 実験集合の作成

松浦らの研究と同様の実験環境を構築するために，青空文庫からの 92 作品データを基にして固定長データからの 50 の実験集合群を作成した。圧縮改善係数からの手法ではデータが固定長である必要はないが，先行研究との比較を行うため，ここでも固定長のデータを作成した。固定長データを作成する場合にはサイズが設定した長さに満たない小さなデータをつなげていく必要があるが，「作品データのつなぎ合わせ方によって，著者推定精度が変動することが考えられるので，ランダムに 50 通りの作品のつなぎ合わせ方を用意し」[21]，50 の実験集合を作成した。

各実験集合の作成手順は以下のとおりである。

1) 92 作品のデータを作品プールとする。そこから擬似乱数（Mersenne Twister 法[31]）によって作品を一つ選択する。
2) 作品中のテキストが 30,000 文字よりも多い場合は，先頭の 30,000 文字を取り出し，実験集合に追加する。該当作品を作品プールから削除する。
3) 30,000 文字よりも少ない場合，同じ著者の 30,000 文字未満の作品群から一つずつ作品を選択し選んだ順に連結する。30,000 文字を超えた時点で，テキストの先頭 30,000 文字を一つの実験テキストとし，実験集合に追加する。連結したすべての作品を作品プールから削除する。
4) 作品プールに作品が残っている場合，1) に戻る。作品がない場合には 5) へ進む。
5) 実験集合に作品が一つしか登録されなかった著者の場合は，著者推定が不可能となるためその著者の作品を除去する。また，著者による偏りをなくすために，一著者あたりの最大実験テキスト数を 5 とし，一つの実験集合の作成を終了する。

このような手順で作成された 50 の実験集合群の総計は第 3 表のとおりである。実験集合群に含まれるデータはすべて 30,000 文字の固定長データとなる。日本語は全角文字であり 1 文字が 2 バイトであるため，バイトに換算した場合には 60,000 バイトの大きさのデータとなる。

実験集合群の各集合に含まれるデータの平均異なり著者数は 7.9 人である。第 3 表からは松浦らのデータと同様の手順で作成したにもかかわらず，特に水野仙子の値が異なっていることがわかる。無作為抽出がデータ作成手順に含まれるため，10 回データ集合を作成し，先行研究と同様の性質になるかを試行したが，そのような実験集合

圧縮プログラムを応用した著者推定

第２表　作品リスト

著者名	タイトル	著者名	タイトル
岡本綺堂	化け銀杏	菊池寛	恩讐の彼方に
岡本綺堂	弁天娘	菊池寛	勝負事
岡本綺堂	菊人形の昔	菊池寛	出世
岡本綺堂	狐と僧	菊池寛	忠直卿行状記
岡本綺堂	帯取りの池	菊池寛	父帰る
岡本綺堂	お照の父	菊池寛	藤十郎の恋
岡本綺堂	津の国屋	菊池寛	若杉裁判長
岡本綺堂	柳原堤の女	菊池寛	ゼラール中尉
岡本綺堂	幽霊の観世物	国木田独歩	源おじ
芥川龍之介	あばばばば	国木田独歩	牛肉と馬鈴薯
芥川龍之介	アグニの神	国木田独歩	非凡なる凡人
芥川龍之介	秋	国木田独歩	恋を恋する人
芥川龍之介	あの頃の自分の事	国木田独歩	武蔵野
芥川龍之介	或阿呆の一生	国木田独歩	怠惰屋の弟子入り
芥川龍之介	或敵打の話	国木田独歩	酒中日記
芥川龍之介	或旧友へ送る手記	国木田独歩	たき火
芥川龍之介	或日の大石内蔵助	国木田独歩	運命論者
芥川龍之介	浅草公園―或シナリオ―	国木田独歩	少年の悲哀
芥川龍之介	一塊の土	国木田独歩	石清虚
梶井基次郎	愛撫	水野仙子	響
梶井基次郎	ある崖上の感情	水野仙子	輝ける朝
梶井基次郎	ある心の風景	水野仙子	神樂阪の半襟
梶井基次郎	泥濘	水野仙子	道―ある妻の手紙―
梶井基次郎	冬の蠅	水野仙子	女
梶井基次郎	冬の日	水野仙子	四十餘日
梶井基次郎	筧の話	水野仙子	嘘をつく日
梶井基次郎	過古	樋口一葉	十三夜
梶井基次郎	器楽的幻覚	樋口一葉	にごりえ
梶井基次郎	K の昇天―或は K の溺死	樋口一葉	大つごもり
梶井基次郎	交尾	樋口一葉	たけくらべ
梶井基次郎	檸檬	樋口一葉	うつせみ
梶井基次郎	のんきな患者	樋口一葉	わかれ道
梶井基次郎	路上	樋口一葉	ゆく雲
梶井基次郎	桜の樹の下には	有島武郎	小さき者へ
梶井基次郎	雪後	有島武郎	二つの道
梶井基次郎	城のある町にて	有島武郎	片信
梶井基次郎	蒼穹	有島武郎	卑怯者
梶井基次郎	闇の絵巻	有島武郎	広津氏に答う
梶井基次郎	橡の花―或る私信―	有島武郎	一房の葡萄
菊池寛	青木の出京	有島武郎	小作人への告別
菊池寛	入れ札	有島武郎	水野仙子氏の作品について
菊池寛	勲章を貰う話	有島武郎	溺
菊池寛	身投げ救助業	有島武郎	宣言一つ
菊池寛	M 侯爵と写真師	有島武郎	想片
菊池寛	無名作家の日記	有島武郎	私の父と母
菊池寛	大島が出来る話	有島武郎	火事とポチ

第 3 表　実験集合の総計

著者名	50 セット中の合計	松浦ら (2000)
岡本綺堂	218	203
芥川龍之介	100	100
梶井基次郎	170	160
菊池寛	241	222
国木田独歩	147	129
水野仙子	88	48
樋口一葉	100	84
有島武郎	100	98
総　計	1,164	1,044

群は作成されなかった。

データ集合の特性に差異が見られた要因としては,

(1) 青空文庫のデータに対して 1999 年時点から修正が加えられこと。

(2) 無作為抽出のための擬似乱数として本研究では Mersenne Twister 法を用いていること（松浦らの研究ではどのような擬似乱数を用いたかは公開されていない)。

が考えられるが，既往研究で公開されているデータではこれ以上の分析は行うことができない。

結果として，実験集合群の特性に若干違いは出ているが，松浦らのデータに比べ各集合に含まれる平均著者数が増加しており，著者推定の精度からはより厳しい条件となったといえる。著者「水野仙子」のテキストデータは，松浦らのデータでは半数以下の集合にのみ含まれるが，今回の集合には 2/3 以上のデータに含まれている。

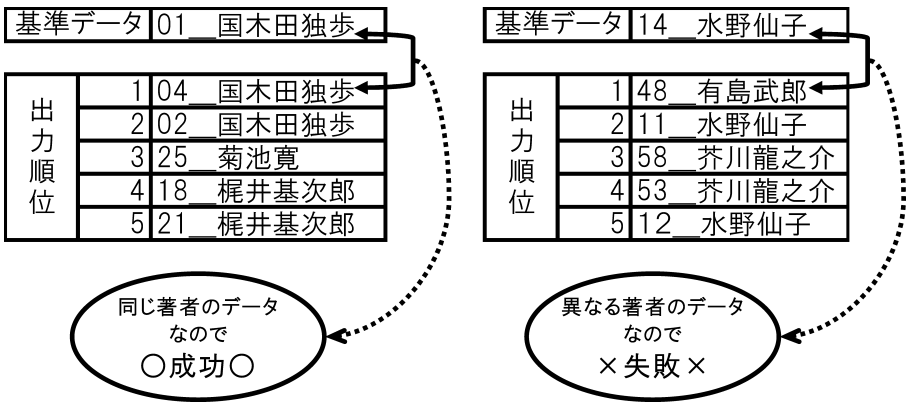
3.　著者推定実験の評価尺度

著者推定実験の評価は，先行研究と同様の手順で行った。ある著者推定手法によってある基準データと集合内の他のデータを比較し，類似度順出力を行ったとき，同じ著者の他のデータが順位 1 位に出力されれば，著者推定に成功したものとし，逆に 2 位以下に出力された場合には失敗したものとする（第 1 図)。全推定試行数に対して，著者推定の成功数の割合を算出している。これを平均成功率と呼び，式で表すと以下のとおりである。

$$\text{平均成功率} = \frac{\text{成功例数}}{\text{全推定数}}\ (\%) \qquad (3)$$

4.　平均成功率の最低基準

ここでは 50 セットから構成されるデータ集合群の統計的な特徴から，もしある作品を基準データとして選択した場合に，それに対応する類似度順出力を完全に無作為に行うシステムの平均成功率を確率的に算出することで，平均成功率の最低



第 1 図　著者推定の成功と失敗の例

基準を考える。

実験集合群の各集合に含まれるデータの平均異なり著者数は 7.9 人である。また，実験集合群の全データが 1,164 件であるため，各集合の平均データ数は 23.28 件である。著者推定の一回の試行を考えた場合に，ある作品を選択すると，その 1 作品は比較データから外すため，集合中の平均データ数は 22.28 件となる。そこで，一人の著者あたりの平均データ数は $22.28 \div 7.9 \fallingdotseq 2.82$ 件となる。ここからどの著者に対してもその著者のデータを出力する確率は $2.82 \div 22.28 \fallingdotseq 12.66\%$ となるため，無作為にデータを出力するシステムがあれば，そのシステムの平均成功率は 12.66% となる。

この値は平均成功率の最低基準となるため，著者推定実験における平均成功率の値は，絶対的な成功率以外に，この最低基準である 12.66% からどの程度改善されたかで相対的に判断することもできる。

B. 固定長データに対する著者推定実験

固定長 30,000 文字のテキストデータから構成される 50 実験集合群を使い，著者推定を行った実験の結果を，第 4 表に示した。表中における松浦らの平均成功率について補足的な説明をすると，彼らは実験のなかでテキストデータから n-gram でデータを取り出しているが，n の値をさまざまに変化させ，それぞれの平均成功率を出している。この表では，それらの平均成功率で最も高かった n についての値を比較対象として転記している。

まず，平均成功率の最低基準である 12.65% と比較した場合に，すべての手法の平均成功率は大幅に高い値となった。これは，どの手法も著者推定に対して，程度の多少はあれ，有効であることを示している。

第 4 表　既往研究との比較

推計手法			平均成功率
松浦ら (2000)*	dissim	3-gram	96.00%
	Takard の手法	2-gram	77.40%
	ダイバージェンス	1-gram	52.50%
Benedetto らの手法			90.46%
圧縮改善係数による手法			97.68%

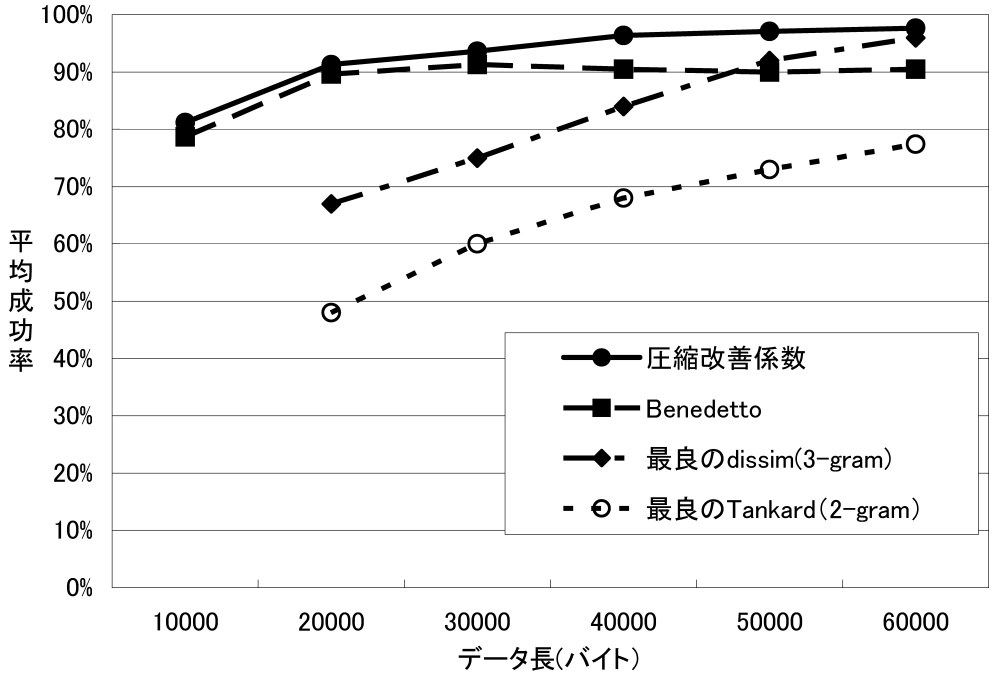
本研究で提案した圧縮改善係数による手法は，すべての手法の中で最も成功率が高く，97.68%（1164 試行中の 1137 回成功）という，ほぼ 100% に近い非常に高い精度を得ることができた。また，50 実験集合中の 29 集合ではすべての試行において正解著者を同定しており，少なくともそれらの集合に関しては，完全に成功しているといえる。

圧縮改善係数による手法は，松浦らの研究において最も性能がよかった dissim の最高値 96.00% よりもさらに 1.68% 平均成功率の値が高い結果となった。松浦らの提案した手法は n-gram の n 値を変えて最適点を見つけるという精緻化を行った結果の性能であるのに対して，圧縮改善係数による手法では外部プログラムである Zip に対して圧縮レベルを最高の 9 にしている以外は特別な操作を行わず用いた結果である。つまり，簡便性という面から見た場合，圧縮改善係数による手法は松浦らの手法よりもはるかに優れているといえる。

圧縮プログラムを応用した手法同士の比較では，Benedetto らの手法でも 90.46% という高精度を得られているが，圧縮改善係数による手法には 10% 程度及ばない。また，すべての試行が成功した実験集合は 50 集合中の 2 集合のみであった。先行研究との比較では Benedetto らの手法は dissim よりは低い成功率ではあるが，定評のある Tankard の手法よりは 12.06% 高い成功率が得られている。

C. データ長を変化させた場合の性能劣化

前節では 30,000 文字/60,000 バイトという固定長データを用いて比較実験を行ってきた。しかし，書簡などの短いテキストに関する著者推定の場合，必ずしも十分な長さのデータが得られるとは限らない。ここでは，基準データ，比較データともに同じ割合で短くしていった場合，つまり，



第２図　データ長と平均成功率

手がかりをより少なくした場合に，どの程度の性能劣化が起こるかを実験した。

実験手順としてはこれまでと同様の 50 セットから構成される実験集合群を使ったが，その際，各データを 10,000 バイトから 60,000 バイトまで 10,000 バイトずつ変化させた場合の平均成功率を算出した。実験結果は，成功率を縦軸に，データのサイズを横軸にとったグラフでは第２図のようになった。このグラフからは，dissim や Tankard の手法はデータが短くなるにつれ，性能が徐々に落ちているが，Benedetto らの手法や圧縮改善係数による手法による著者推定の場合，20,000 バイト/10,000 文字程度のデータであっても９割を超える成功率が得られていることがわかる。

また，二手法については性能の落ち込みが極めて緩やかであるため，既往研究では行われなかった 10,000 バイト/5,000 文字という非常に短いデータを使っての著者推定も行ったが，ここまで手がかりが少なくなったデータにおいても，圧縮改善係数による手法では 80% を超える成功率となっている。

Benedetto の手法は 60,000 バイトからデータ長を短くしていったときに，20,000 バイト時点まではほぼ横ばいであり，性能低下は見られず，逆に少しだけ精度が上がっている箇所も見られた。最も短い 10,000 バイト時点で大きな性能低下が見られ，平均成功率は 80% を若干下回るものとなった。

D.　圧縮レベルによる違い

圧縮プログラムの圧縮レベルを低く設定すれば，原則的にデータの圧縮率は低くなり，類似データの識別力が低くなることが予想される。逆に圧縮レベルを高く設定すれば類似データの識別力が高くなることが予想される。ここでは，圧縮改善係数による手法について圧縮レベルを変化させた場合に，著者推定の精度がどのように変化するかを検証するために実験を行った。

具体的には，B 節と同様の環境で，圧縮改善係数による手法を用いて，固定データ長 60,000 バイトの 50 実験集合群に対して，圧縮レベルをレ

第 5 表　圧縮レベル別成功率

圧縮レベル	平均圧縮率	平均成功率
1	51.59%	94.24%
2	49.92%	96.65%
3	48.44%	94.33%
4	47.75%	97.51%
5	46.81%	97.77%
6	46.36%	97.68%
7	46.29%	97.59%
8	46.27%	97.68%
9	46.27%	97.68%

ベル 1 からレベル 9 まで変化させた場合の平均成功率の値を算出した。Info-Zip のライブラリでは圧縮レベル 0 からレベル 9 に設定可能であるが，圧縮レベル 0 は前述の理由から除外している。

実験結果は第 5 表のとおりとなった。

この表で中央の平均圧縮率は 60,000 バイトの固定長のデータを圧縮したときに平均してどの程度圧縮されたかを示している。平均圧縮率については，原則的には圧縮レベルに比例し，平均圧縮率が高くなっている。ただし，個別のデータについてはレベル 5 から 7 がレベル 8, 9 よりも高い圧縮率を示している場合も見られた。レベル 8 と 9 についてはどのデータについてもサイズが同一であるため，圧縮レベルによる圧縮パラメータの違いはないと考えられる。

表で右列に平均成功率があるが，全体としては，どの圧縮レベルでも 94% を超える高い平均成功率が得られている。圧縮レベルが 1 から 3 のときには，4 以上の圧縮レベルよりも 1〜3% 平均成功率が低くなっている。一方で，圧縮レベルが 4 から 9 の場合にはわずかな差はあるが，すべて 97% 以上となっている。

圧縮レベルを高くしても平均成功率が必ずしも高くならなかった理由としては，以下のような原因が考えられる。

(1) 全体として 94% を超える高い精度が得られており， Zip を使った場合の性能限界であるため，データのごくわずかな違いで精度が変わってしまうこと。

(2) 圧縮レベルは圧縮を行うときのバッファの大きさを指定するもので，基本的にはバッファが大きいほど，圧縮率が上がるとされているが，バッファを大きく取りすぎても符号自体が長くなり，圧縮率は下がることもあるため，圧縮レベルを高くしたときに必ずしも圧縮率が高くなるとは限らないこと[32]。

IV. オリジナルデータを対象とした著者推定

A. 実験環境

既往研究との比較ではどのデータのサイズも同じとし，一部のデータは複数の文献を連結するという操作をした固定長データを用いた実験集合を対象とした実験であった。ここでは，オリジナルの 92 作品に何ら操作を加えることなく，そのままの形で用いて著者推定実験を行った。

92 作品のうち，最もサイズが小さいものは梶井基次郎の「過古」で，書誌事項を除いたテキスト部分は 3,184 バイトであり，最もサイズが大きいものは岡本綺堂の「半七捕物帳　津の国屋」で，66,928 バイトであった。92 作品のサイズの平均値は 20,783 バイトである。

オリジナルデータに対しても，前章の固定長データの実験と同様に，実験集合中から 92 作品のうち 1 作品を選択し，それ以外のデータに関して類似度順出力を行った。同じ著者のデータが出力順位 1 位に出力された場合には成功とし，平均成功率を算出した。さらに，各データの先頭から何バイトまでを圧縮処理の対象とするのかについて，10,000 バイト単位で変化させる試行も行った。

B. 実験結果

オリジナルデータに対する著者推定実験の結果は第 6 表のとおりである。

第 6 表　オリジナルデータに対する著者推定

		Benedetto		圧縮改善率	
		成功数（92 試行中）	成功率	成功数（92 試行中）	成功率
データ長（バイト）	60,000	79	85.9%	88	95.7%
	50,000	79	85.9%	88	95.7%
	40,000	81	88.0%	88	95.7%
	30,000	80	87.0%	88	95.7%
	20,000	87	94.6%	88	95.7%
	10,000	86	93.5%	88	95.7%

この表から，右列に記されている圧縮改善係数による手法では，手がかりとするデータの先頭からのサイズに関係なく 92 作品中 88 作品において著者推定が成功しており，95.7% という高い平均成功率を示したことがわかる。また，左列の Benedetto らの手法も全体的には 80% 以上の高い平均成功率であった。ただし，先頭からのデータサイズを変化させた場合，60,000 バイトと 50,000 バイト時の平均成功率が最も低く，手がかりとなるデータのサイズを短くするほど，性能が高くなるという現象が見られた。

C. 失敗事例

オリジナルデータを対象とした実験では，圧縮改善係数による手法は，4 作品においてのみ著者推定に失敗した。この 4 作品を基準データとして行った試行について，類似度が高いと判定された上位 5 作品までを示したものが第 7 表である。

この表から，失敗 4 事例のうち事例 1, 2, 3, の三つは出力順位 2 位が正解著者となっていることがわかる。

さらに，失敗事例 2 は基準データが有島武郎「水野仙子氏の作品について」で出力順位 1 位には水野仙子の作品が出力されており，また，失敗事例 3 は基準データが水野仙子の作品で出力順位 1 位に有島武郎「水野仙子氏の作品について」が出力されている。「水野仙子氏の作品について」はタイトルからもわかるように，有島武郎による水野仙子の作品に対するエッセイである。つまり，明示的な引用部分や，明示的な引用ではないが同じ語彙を使っている箇所が多く存在すると考えられる。結果として失敗事例 2, 3 は，著者の作品同士よりも，ある著者の作品とその作品が言及していた作品との共通部分が多く，結果として誤った判定がされてしまった事例であることが理解できる。

失敗事例 4 については出力順位 5 位までに同じ著者の作品は出力されず，出力順位 7, 8, 10 に同じ著者の作品が出力されていた。出力順位 1 位 2 位だけでなく，上位に同じ著者の作品が出力されなかったという点で，この事例は著者推定に完全に失敗したと考えられる。基準データと出力順位の上位の作品を比べると，基準データである菊

第 7 表　失敗事例

		1		2		3		4	
		著者	作品名	著者	作品名	著者	作品名	著者	作品名
基準データ		国木田独歩	少年の悲哀	有島武郎	水野仙子氏の作品について	水野仙子	輝ける朝	菊池寛	勝負事
出力順位	1	有島武郎	卑怯者	水野仙子	輝ける朝	有島武郎	水野仙子氏の作品について	有島武郎	小作人への告別
	2	国木田独歩	非凡なる凡人	有島武郎	広津氏に答う	水野仙子	嘘をつく日	有島武郎	私の父と母
	3	菊池寛	恩讐の彼方に	芥川龍之介	或旧友へ送る手記	水野仙子	響	梶井基次郎	器楽的幻覚
	4	梶井基次郎	愛撫	水野仙子	女	水野仙子	神樂阪の半襟	梶井基次郎	過古
	5	梶井基次郎	過古	水野仙子	神樂阪の半襟	水野仙子	四十餘日	芥川龍之介	一塊の土

池寛の「勝負事」は括弧で囲まれた会話の多い作品であるが，集合中にある菊池寛のほかの作品は会話の少ない物語か戯曲であった。そのため，独白形式の有島武郎の「小作人への告別」が一番に出力されたと考えられる。

Ⅴ．まとめと課題

A. 実験のまとめ

本研究では圧縮プログラムを用いた類似データの同定手法が有効であるかどうかを日本語データの著者推定実験から検証することを試みた。事前実験から既往研究で使われてきた Benedetto らの手法には二つの問題点が明らかとなったため，それらを改善した圧縮改善係数による類似データの同定手法を提案した。

既往研究との比較実験では，既往研究とほぼ同様の環境を用意し，圧縮プログラムを用いた二手法を使って実験し，既往研究の実験結果との比較を行った。圧縮プログラムを応用した同定手法は，従来の著者推定手法（と同程度あるいはそれ）以上の高い平均成功率が得られた。特に，今回提案した圧縮改善係数による手法は，最も高い性能を示した。また，手がかりとなるデータ長が短い場合でも他の手法に比べ，性能劣化が少ないことが明らかとなった。圧縮レベルを変化させた場合の実験では，低圧縮レベルのグループは高圧縮レベルのグループと比較して，平均成功率が若干ではあるが低かった。このことから，圧縮率が高くなれば，類似データの同定の精度が上がることが示唆された。

青空文庫のデータを加工せずに行った実験でも 95.7% と高い成功率が得られ，データのサイズが異なるデータについても十分に応用可能であると考えられる。この実験の失敗事例の分析からは 4 事例のうち 3 事例では出力順位 2 位に正解著者の作品が出力されており，完全なる判定失敗は少ないことが明らかとなった。

B. 今後の課題

1. 他の圧縮プログラム

本研究における類似データの同定手法は外部の圧縮プログラムを用いている。そのため，それらのプログラムが，公開してあるアルゴリズム以外に非公開の特殊な操作を行っていれば分析不可能なブラックボックスとしてしか扱うことができない。そのような欠点はあるが，一方で，圧縮プログラムに関する技術が進み圧縮率が向上すれば，この手法の改善に直結するという利点がある。今回は用いる圧縮プログラムを最も標準的な Zip としたが，異なる圧縮アルゴリズム，特に Huffman 符号化，LZ77 符号化以外のものを使った圧縮プログラムを用いれば，今回とは異なる結果が得られることが予想される。

2. 可逆圧縮と非可逆圧縮

圧縮アルゴリズムは，可逆圧縮と非可逆圧縮に大別することができる。今回用いた Zip は可逆圧縮（無歪）アルゴリズムを用いた圧縮プログラムの一種であり，テキストデータなどの圧縮に多く用いられ，圧縮されたデータから元データを完全な形で復元可能である。一方で，非可逆圧縮の圧縮アルゴリズムは画像や音声の圧縮に使われることが多い圧縮手法である。特徴的な部分を残し，あまり認識されない瑣末なデータを省略することで，高い圧縮率が得られるが，圧縮データからは元データを正確には復元できない。Benedetto らの手法，圧縮改善係数による手法は，両方ともに非可逆圧縮の圧縮アルゴリズムを用いることも理論上は可能である。今後は，この手法の更なる検証のために，異なる圧縮プログラムを可逆圧縮，非可逆圧縮を問わずに組み込んで実験を行っていくことを検討している。

注・引用文献

1) Benedetto, Dario et al. Language trees and zipping. Physic al Review Letters. vol. 88, no. 4, 2002, p. 048702-1–048702-4
2) Liber liber. “the Manuzio plan”. 〈http://www.liberliber.it/〉, ［最終確認日：2005-5-31］
3) Khmelev, Dmitry V.; Teahan, William J. Comment on ‘Language trees and zipping’. Physical Review Letters. vol. 90, no. 8, 2003, p. 089803-1.
4) Benedetto, Dario et al. Benedetto, Caglioti, and

Lorento reply. Physical Review Letters. vol. 90, no. 8, 2003, p. 089804-1.
5) Guendel, A. et al. Effect of stress on the entropy calculated by applying the zipping method to Barkhausen noise. Journal of Magnetism and Magnetic Materials. vol. 290–291, 2005, p. 1165–1167.
6) 奥村晴彦. コラム 3 Physical Review Letters に載った gzip. C Magazine. vol. 14, no. 7, 2002, p. 35.
7) Anderson, Mark K. ファイル圧縮技術の応用でテキストの筆者を推定. Wired News. 〈http://hotwired.goo.ne.jp/news/news/technology/story/20020208302.html〉［最終確認日：2005-5-31］
8) Kukushkina, O. V.; Polikarpov, A. A.; Khemelev, D. V. Using letters and grammatical statistics for authorship attribution. Problems of Transmitting of Information. vol. 37, No. 2, 2001, p. 172–184（http://www.philol.msu.ru/~lex/articles/grco_e.htm より入手可能）
9) 内山和也. スタイルの計量に関する覚え書き：文体論の視点から. 計量国語学. vol. 23, no. 7, 2002, p. 347–352.
10) 村上征勝. 真贋の科学：計量文献学入門. 東京, 朝倉書店, 1994. 154 p.
11) 佐藤進也ほか. 文字列出現頻度比較による情報源間の類似性判定. 情報処理学会研究報告. vol. 2002, no. 028 (FI-066), 2002, p. 119–126.
12) Tankard, J. The Literary detective. BYTE. vol. 11, no. 2, 1986, p. 231–238.
13) フリードマン, リチャード・エリオット. 旧約聖書を推理する：本当は誰が書いたか. 松本英昭訳. 滋賀, 海青社, 1989, 355 p.（原書の再版が 1997 年に刊行されている）
14) 村上征勝. 著者を探る古文書の計量分析. 電子情報通信学会誌. vol. 85, no. 3, 2002, p. 158–161.
15) 細江光. 谷崎の作品ではなかった偽作「誘惑女神」をめぐって. 国文学 解釈と教材の研究. vol. 33, no. 8, 1988, p. 134–137.
16) 森直久. ある刑事事件の供述資料における作成者同一性の心理学的検討. 札幌学院大学人文学会紀要. vol. 69, 2001, p. 13–36.
17) 坪井祐太, 松本祐治. 異なるタイプのドキュメントに対する著者推定. 情報処理学会研究報告. vol. 2002, no. 20 (NL-148), 2002, p. 17–24.
18) Ironport Systems Inc. “SpamCop statistics”. SpamCop.net. 〈http://www.spamcop.net/spamgraph.shtml?spamyear〉,［最終確認日：2005-5-31］
19) 石田栄美ほか. 文体からみた学術的文献の特徴分析. 2004 年度三田図書館・情報学会研究大会発表論文集, 2004, p. 33–36.
20) 石田らが調査した文献は以下の六文献である.
①ケニィ, アンソニー. 文章の計量：文学研究のための計量文体学入門. 吉岡健一訳. 東京, 南雲堂, 1996. 244 p.（原著は 1982 年）
②村上征勝. 真贋の科学：計量文献学入門. 東京, 朝倉書店, 1994. 154 p.
③Holmes, D. I. Authorship attribution. Computers and Humanities. vol. 28, 1994, p. 87–106.
④吉岡健一. “計量文体学研究の展望”. 文章の計量. 東京, 南雲堂, 1996. p. 196–234.
⑤安本美典. 語彙の量的構造. 数理科学. vol. 15, no. 6, 1977, p. 44–49
⑥陳志文. 新聞の各紙面に見られる文体の類型：主成分分析法による朝日新聞と読売新聞の分析から. 国語学研究. no. 42, 2003, p. 54–44.
21) 松浦司, 金田康正. 近代日本文学者 8 人による文章における文字 *N*-gram 分布を手がかりとする著者推定. 情報処理学会研究報告. vol. 99, no. 95 (NL-134), 1999, p. 31–38.
22) 松浦司, 金田康正. *n*-gram の分布を利用した近代日本語文の著者推定. 計量国語学. vol. 22, no. 6, 2000, p. 225–238.
23) 安形輝. 圧縮プログラムによる類似データの同定. 2004 年度日本図書館情報学会研究大会発表要綱, 2004, p. 65–68.
24) Ziv, Jacob; Lempel, Abraham. A universal algorithm for sequential data compression. IEEE Transactions on Information Theory. vol. IT-23, no. 3, 1977, p. 337–343（http://citeseer.ist.psu.edu/ziv77universal.html より入手可能）
25) Huffman, D. A. A method for the construction of minimum redundancy codes. Proceedings of the Institute of Radio Engineers. vol. 40, no. 9, 1952, p. 1098–1101（http://compression.graphicon.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf より入手可能）
26) Java Developer’s Kit（サン・マイクロシステムズが開発した特定の OS に依存しないプログラミング言語である Java 言語の標準的な開発環境：http://java.sun.com/j2se/ より入手可能）
27) Deutsch, P. “ZLIB Compressed Data Format Specification version 3.3”. IETF. 〈http://www.ietf.org/rfc/rfc1950.txt〉,［最終確認日：2005-5-31］（日本語訳が http://www.futomi.com/lecture/japanese/rfc1950.html より入手可能）
28) Deutsch, P. “DEFLATE Compressed Data Format Specification version 1.3”. IETF. 〈http://www.ietf.org/rfc/rfc1951.txt〉,［最終確認日：2005-5-31］（日本語訳が http://www.futomi.com/lecture/japanese/rfc1951.html より入手可能）
29) Info-ZIP. “Info-ZIP Application Note 970311

(java.util.zip クラスの元になる Info-ZIP 形式の詳細な説明)". 〈ftp://ftp.uu.net/pub/archiving/zip/doc/appnote-970311-iz.zip〉[最終確認日: 2005-5-31]

30) 青空文庫. "青空文庫". 〈http://www.aozora.gr.jp/〉, [最終確認日: 2005-5-31]

31) Mersenne Twister 法とはきわめて長い周期 2^{19937}-1 を持ち, 623 次元以下で一様に分布する擬似乱数実験集合群に生成手法である. モンテカルロ法に向く乱数とされる. その詳細は以下の文献に詳しい.
Matsumoto, M.; Nishimura, T. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Transaction on Modeling and Computer Simulation. vol. 8, no. 1, 1998, p. 3–30.

32) 実験集合群で 60,000 バイトの固定長データについては, 圧縮レベルに比例して平均圧縮率が高くなっていた。しかしながら, より短いデータについて平均圧縮率を算出すると, 10,000 バイト, 20,000 バイト, 40,000 バイトのデータについてはごく僅かであるがレベル 7 の方がレベル 8, 9 よりも高い圧縮率を示しており, 必ずしも圧縮レベルと平均圧縮率が連動するものでないことがわかる。