RD-001

低次元特徴空間におけるランダムフォレストを用いた書誌同一性判定の性能評価

Performance Evaluation of Bibliographic Linkage Using Random Forest in Low Dimensional Feature Space

中島　良平† Ryohei Nakashima　高須　淳宏‡ Atsuhiro Takasu　安達 淳‡ Jun Adachi

1. はじめに

電子書籍や電子ジャーナル等，世の中には電子化された文書があふれている．たとえば，国立情報学研究所が提供する学術論文情報検索サービス CiNii の収録文献数は，2014 年 8 月時点で約 1,700 万件にも及ぶ[4]．このような電子図書館サービスを快適に利用するには，検索や文書間リンクの機能が必須である[20]．文書間リンクは学術論文からそれが引用している文献へのリンクを言う．この機能を用いて利用者は学術論文の引用文献へ簡単に到達できる．

様々な記述がされる引用文献の書誌をもとに指定されたデータベース中のレコードから同一である書誌を見つけることを書誌同定と言う．書誌同定システム[17]は，様々な文献の書誌を持つデータベースから学術論文の引用文献と同一であると判定した文献の書誌を抽出し，その文献へのリンクを出力する．

書誌同定システムでは，一般的なレコード同定システム[16]と同様にセグメンテーション，正規化，ブロッキング，同一性判定の 4 つのモジュールの直列構成となっている．このうち同一性判定モジュールは，レコードペアの一致する度合いである照合スコアを求め，「照合可」「照合可能性あり」「照合不可」のいずれかのクラスに分類する．レコードペアとは，同一性判定対象のレコードのペアを示す．

「照合可能性あり」に分類されたレコードペアは，システムによる自動的な判断が不可能であるとして，人手による判定を要する．この人手による判定処理は，時間がかかり，間違いやすい[7]．そのため，この「照合可能性あり」に分類されるレコードペアを極力少なくしなければならない．したがって，同一性判定モジュールには「照合可能性あり」に分類してしまうレコードペアを減らすとともに，「照合可」「照合不可」のいずれかに正確に分類することが求められる．

書誌同定システムの性能を向上させるためには，同一性判定モジュールの性能向上が不可欠である．そこで本研究の目的を 99.99%に可能な限り近い性能を持つ同一性判定モジュールの開発としている．

本研究では，まず，書誌の同一性判定に有効な特徴量を実際の書誌データ等を参考に作成する．その特徴量を用いて Random Forest を書誌の同一性判定に適用し，既存の同一性判定モジュールやその他の分類器よりよい性能となることを示す．次に作成した特徴量の重要度を測定し，特徴量選択を行う．特徴量選択後の特徴量空間においても Ramdom Forest を用いた高精度な書誌の同一性判定が可能であることを示す．

† （株）日立製作所　情報・通信システム社, Hitachi, Ltd., Information & Telecommunication Systems Company

‡ 国立情報学研究所, National Institute of Informatics

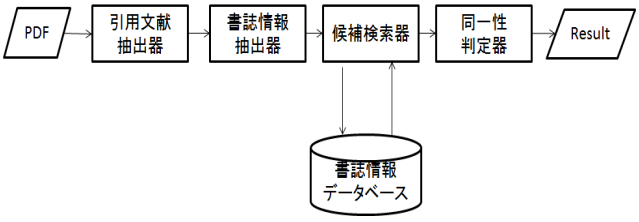


図 1　書誌同定システムの処理フロー

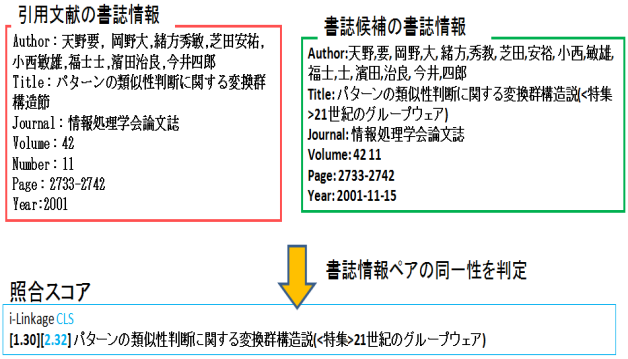


図 2　書誌の同一性判定処理例

2. 書誌同定システム

書誌同定システムは，学術論文の PDF を入力にとり，図 1 に示す処理フローを行って，照合スコアを付与した書誌の候補とそのリンクを出力する．

引用文献抽出器は，学術論文 PDF の中で引用文献が書かれた領域を特定し，引用文献文字列を書誌ごとに抽出する．

書誌情報抽出器[20]は，引用文献抽出器が抽出した引用文献文字列から書誌情報を抽出する．書誌情報抽出器は，一般的なレコード同定システムのセグメンテーションモジュールに相当する．

候補検索器はブロッキング[16]を行う．書誌同定システムでは CiNii と CrossRef[5]に収録されている学術論文の書誌情報を持つデータベースを用いる．候補検索器は一般的なレコード同定システムの正規化モジュールとブロッキングモジュールに相当する．

同一性判定器は書誌情報抽出器が抽出した書誌情報と候補検索器が選別した書誌候補が持つ書誌情報のペアの同一性を表す照合スコアを出力する．処理例を図 2 に示す．照合スコア欄に示した数値が入力の書誌に対する照合スコアである．同一性判定器の実装は，i-Linkage[17]と CLS [19]の二つがある．同一性判定器は，一般的なレコード同定システムの同一性判定モジュールに相当する．

3. 関連研究

相澤ら[17]は，引用文献文字列などのテキストを大規模データベースのレコードに動的に対応付ける「i-Linkage」を提案した．また，蔵川ら[21]は，異なるデータセットで i-Linkage の性能評価を行った．日立製作所中央研究所(日立中研) [19] は CLS を提案し，性能評価を行った．

我々は既存実装である i-Linkage ， CLS の問題点を特徴量の次元数，特徴抽出，書誌の同一性判定性能，分類器の 4 つの観点で分析した．特徴量の次元数は i-Linkage ， CLS はそれぞれ 23 次元，60 次元であるが，次元の呪いを考慮すると次元数はもっと低い方がよい．特徴抽出では約物，大文字から小文字への正規化処理を行わないで抽出した特徴量があるが，これは編集距離の精度に影響する．また，引用文献の書誌と書誌候補の書誌のうちどちらか片方の書誌のみに依存した特徴量があるが，これらは書誌の同一性を判定するのに役に立つかわからない．i-Linkage と CLS の書誌の同一性判定性能である F 値と精度はともに約 95%前後あるが，人手判定の回数をさらに減らすにはより高い性能が必要となってくる．分類器については，i-Linkage ，CLS は分類器として SVM を用いている．しかし，書誌同定問題と同じくレコード同定問題の小問題である著者の曖昧性解消問題において Treeratpituk ら[14]の研究で Random Forest が SVM やその他の分類器より有効であることが示されている．

4. 提案手法

4.1 概要

既存実装の問題点を解決するため本研究では，まず，書誌の同一性判定に必要な特徴量を再検討する．その特徴量に引用文献の書誌と書誌候補の書誌のうちどちらか片方の書誌のみに依存した特徴量と両方の書誌を比較した特徴量を含める．また，特徴量計算の前処理として約物や大文字から小文字への正規化処理を導入する．その作成した特徴量をいくつかの分類器を用いて書誌の同一性判定に適用し，評価実験を行う．実験結果より作成した特徴量が i-Linkage や CLS のものより優位であることを示す．次に作成した特徴量の重要度を測定する．その結果から片方の書誌のみに依存する特徴量の必要有無の考察と特徴量選択を行い特徴量の次元数削減を図る．

また，書誌の同一性判定の分類器として Random Forest を適用し，Random Forest が書誌の同一性判定において既存実装やその他の分類器よりよい性能であることを示す．

4.2 特徴量

分類器の入力となる特徴量は i-Linkage が用いている特徴量， i-Linkage と CLS の同一性判定誤りのパターンを分析した結果，i-Linkage と CLS が用いている学習データを参考に 25 次元の特徴量ベクトルを得た．25 次元の特徴量ベクトルの詳細は付録 A に記載する．

5. 作成した特徴量の優位性を示すための実験

5.1 使用した分類器

使用した分類器とアンサンブル学習に使用した学習器およびその数を表 1 に示す．表 1 の linear は線形カーネル，poly は多項式カーネルを示す．実装は全て python で行った．SVM はライブラリ libsvm[6]を用いて実装した．アンサンブル学習手法は全てライブラリ scikit-learn[15]を用いて実装した．既存実装である i-Linkage と CLS で SVM は利用されているが，同じ特徴量を使用したときの SVM の性能も確認するため， SVM も比較手法として取り入れた．Bagging において Decision Tree を利用していないのは，Random Forest が Decision Tree を用いた Bagging の改良であり，それの性能を上回ることが文献[3]により検証されているためである．アンサンブル学習の手法における学習器の数は， Decision Tree はライブラリで設定されている初期値を用いた．SVM は学習器の数が 10 を超えると性能が一定の値となったため，学習器の数を 2~10 まで変更し，学習器の数ごとの性能を 5 分割交差検定を用いて確認した．その中でも最も性能が良かった 3 を採用した．

表 1　使用した分類器とアンサンブル学習に使用した学習器およびその数

手法	弱学習器	学習器の数
i-Linkage		
CLS		
SVM(linear)		
SVM(poly)		
AdaBoost	SVM(linear)	3
	SVM(poly)	3
	Decision Tree	50
Bagging	SVM(linear)	3
	SVM(poly)	3
Random Forest	Decision Tree	10

5.2 評価データ

評価データは 2 つある．以下に各評価データを説明する．

(1)情報処理学会論文誌 PDF の引用文献を i-Linkage でブロッキングした結果に人手で正解，不正解のラベルを付与したものである．正例が 1,244 件，負例が 795 件，合計 2,039 件である．これを original と呼ぶ．

(2) i-Linkage と CLS の同一性判定誤りの事例を集めた．そのデータに手作業で正解判定ラベルを付与したデータを評価データとする．すなわち，既存実装である i-Linkage と CLS が苦手とするデータである．正例が 138 件，負例が 1849 件，合計 1987 件である．これを weakData と呼ぶ．

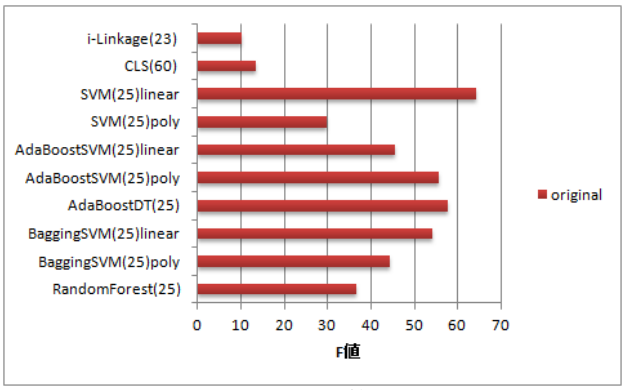
5.3 結果

original を学習データとし weakData に対する同一性判定性能を評価した．original を学習データとしたときの weakData に対する同一性判定性能を評価した結果から算出した F 値を図 3 の(a)，精度を図 3 の(b)に示す．図 3 中の linear は線形カーネル，poly は多項式カーネルを示す． 図 3 中の()内は特徴量ベクトルの次元数を示す．

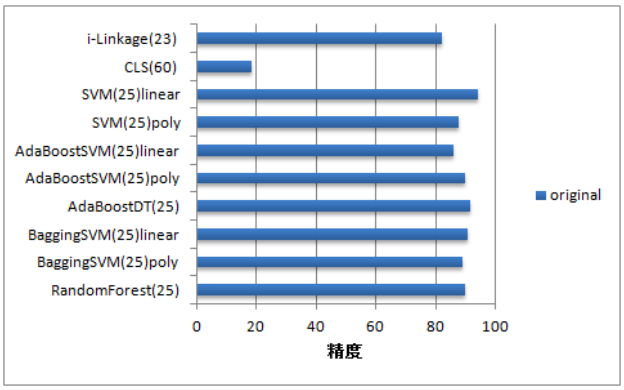
5.4 考察

結果は，i-Linkage と CLS よりどの手法も F 値が 15%以上，精度が 3%以上優位であった．このことより作成した 25 次元の特徴量ベクトルが i-Linkage や CLS の特徴量より有効であることを示すことができたと言える．

本実験において書誌の同一性判定を誤っている書誌情報ペアの多くは，url のみしか書誌情報を持たないような書誌情報ペアであった．実際の例を表 2 に示す．これは，書誌情報がないことにより特徴抽出ができないことが原因で

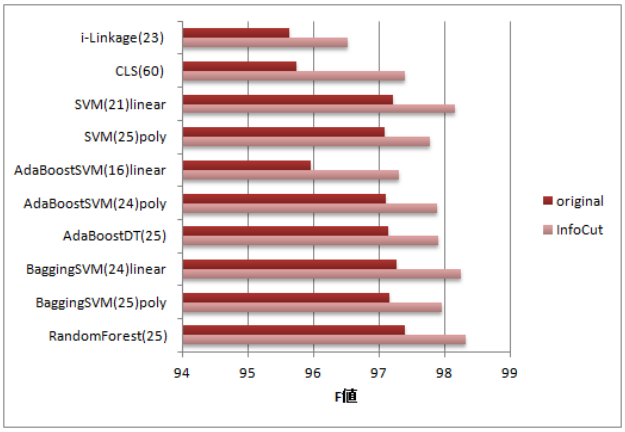


(a)F 値

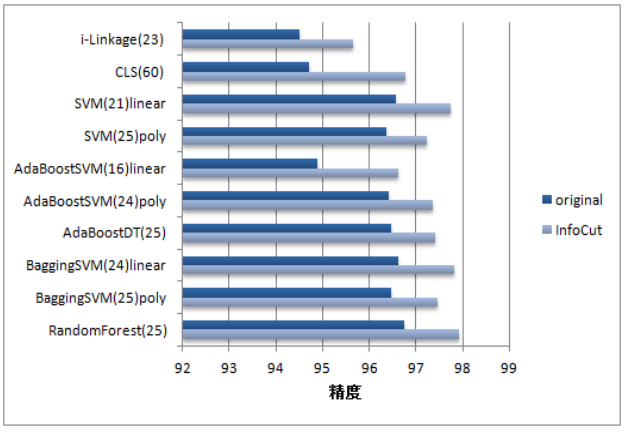


(b)精度

図 3　original を学習データとしたときの各分類器の weakData に対する性能



(a)F 値



(b)精度

図 4　original と InfoCut を用いた各分類器の 5 分割交差検定の結果

表 2　情報量の少ない書誌情報の例

	引用文献の書誌情報	書誌候補の書誌情報
書誌情報	内容	内容
title	null	The Secure Shell (SSH) Transport Layer Encryption Modes
author	null	null
journal	null	null
url	http://www.ietf .org/rfc/rfc4303.txt	http://www.ietf.org/rfc/rfc4344.txt
pubDate	null	null

あると考えられる．また，人手正解ラベル誤りの書誌情報ペアが学習データである original に存在していたことも書誌の同一性判定誤りの原因の一つであると考えられる．そのため，まず，original に含まれる人手正解ラベルを誤っている書誌情報ペアの人手正解ラベルを修正する．次に url しか書誌情報を持たない書誌情報ペアは分類対象外としてデータから削除し，新たな評価データを作成する．

その他の問題点として本実験の評価データは weakData のみであることがあげられる．そのため，どの分類器が最も汎化能力が高いか判断できない．そこで，分類器の汎化能力を測定するため，交差検定を行う．

6. 汎化能力の高い分類器を決定するための実験

6.1 評価データ

評価データの詳細を以下に示す．

(1)作成した特徴量の優位性を示すための実験で用いた original を用いる．

(2)original には明らかに人手正解ラベルが誤っているデータが 34 件あった．これをまず正しい人手正解ラベルに修正する．次に書誌情報として URL のみしか持たない書誌情報ペア 59 件を削除したものを評価データとする．正例が 1,214 件，負例が 766 件，合計 1,980 件である． これを InfoCut と呼ぶ．

6.2 結果

評価は original と InfoCut を用いて表 1 にあるそれぞれの分類器で 5 分割交差検定を行った．5 分割交差検定を行った結果から算出した F 値を 図 4(a)，精度を図 4 (b)に示す．

6.3 考察

どちらのデータセットにおいても F 値，精度ともにラ Random Forest が最高値であった．これより汎化性能が最も高い分類器は，Random Forest であると言える．Info Cut における Random Forest の性能は F 値 98.33%と精度 97.93%であり i-Linkage，CLS より 1%以上優位性がある．

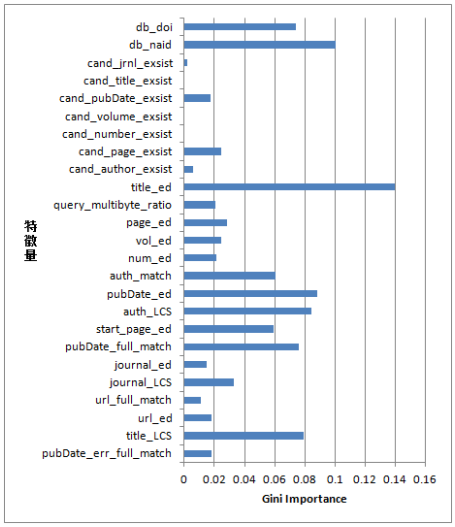


図 5　Gini Importance を用いた各特徴量の重要度

既存実装である i-Linkage と CLS よりよい性能となったがまだ性能改善の余地がある．Random Forest は，Segal[13]により弱学習器である個々の木の深さを調節することにより性能がわずかながら改善されることが示されている．これより Random Forest のパラメタチューニングを行えば性能改善の可能性がある．

今回，評価や学習に用いたデータは約 2,000 件である．しかし，もっと膨大な数のデータを用いて分類器の性能を比較すると異なる結果となる可能性がある．これより学習データの数の違いによる性能評価を行うべきである．また，学習データを増やせば性能が向上することが期待される．

他には情報処理学会論文誌以外の異なる分野の論文データを用いて分類器の性能を比較すると異なる結果となる可能性がある．これより異なる分野の論文データで分類器の比較をする必要がある．

7. 特徴量重要度と特徴量選択

汎化性能が高かった評価データである InfoCut を用いて 5 分割交差検定を行った際にどの特徴量の影響が強かったのかを調査した．重要度の指標として Gini Importance[2]を用いた．特徴量重要度の調査結果を図 5 に示す．

図 5 の結果より特徴量選択条件ごとに InfoCut を用いて Random Forest の交差検定を行った．その結果を表 3 に示す．

図 5 の結果から cand_number_exist，cand_volume_exist，cand_title_exist は値が 0 となっており必要のない特徴量であることが分かる．これらが必要のない特徴量となってしまった原因は，特徴量の値が全ての学習データにおいて一様であったことが考えられる．実際に 3 つの全ての特徴量を確認したところ，3 つとも一様の値となっていた．この 3 つの特徴量は，どれも書誌候補にのみ依存する特徴量である．また，Gini Importance の高い Top10 を見ると片方の書誌のみに依存した特徴量でランクインしているのは，2 つだけである．これより，片方の書誌のみに依存する特徴量の多くは書誌の同一性判定に有効とは言えない．

表 3 より特徴量選択後は，全ての特徴量を用いた場合より 10 次元も少ない特徴量ベクトルで，全ての特徴量を用いた場合の性能に近い性能を出すことができている．同時に既存実装の i-Linkage や CLS の特徴量の次元数より 8 次元低い次元数にすることができた．

表 3　特徴量選択後の InfoCut を用いた Random Forest の 5 分割交差検定の結果

特徴量選択条件	使用した特徴量	次元数	F 値	Acc
Gini Importance が 0.02 以上の特徴量のみ	1,2,8,10～19,21,24	15	98.23	97.83
全ての特徴量	1~25	25	98.79	97.93

8. おわりに

本研究では，99.99%に可能な限り近い性能を持つ同一性判定モジュールを開発するために，書誌の同一性判定の分類器として Random Forest を用いることを提案した．書誌同定システムの同一性判定モジュールの既存実装やその他の分類器と比較した結果，Random Forest が最もよい性能となった．また，関連研究により著者の曖昧性解消問題においても Random Forest は有効であることが示されているため，一般的なレコード同定問題においても有効であると言えるだろう．また，既存実装が用いている特徴量ベクトルより次元数を削減，特徴量抽出において片方のみの書誌に依存する特徴量が有効でないことを確認するため，作成した特徴量の重要度を測定し，特徴量選択を行った．特徴量重要度の測定の結果，重要度の高い Top10 に片方の書誌のみに依存する特徴量が 2 つ含まれた．これより片方の書誌のみに依存する特徴量の多くは書誌の同一性判定に有効とは言えない．また，特徴量選択の結果，15 次元の特徴量で 25 次元の特徴量とほぼ同等の性能をだすことができた．結果として既存実装より 8 次元低い次元数にすることができた．

今後は，Random Forest のパラメタチューニングによる性能向上，学習データの数の違いによる性能評価，異なる分野の論文データでの性能評価などの課題に取り組む．

参考文献

[1] R. Baeza-Yates and B. Ribeiro-Neto:Modern Information Retrieval [2nd ed.], Addison-Wesley Professional,pp.223(2011).
[2] L. Breiman, and A. Cutler, "Random Forests":http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm(2015.3.6 取得)
[3] L. Breiman. Random forests. Machine Learning, Vol.45, No.1, pp.5-32,(2001).
[4] CiNii Articles：http://support.nii.ac.jp/ja/cia/cinii_articles (2014.11.28 取得).
[5] CrossRef：http://www.crossref.org/ (2014.11.28 取得)
[6] C. Chang and C. Lin. Libsvm: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/, (2015.3.4 取得).
[7] P. Christen.:A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, Vol.24, No.9, pp.1537-1555(2012).
[8] I. Guyon, M. Nikravesh, Steve Gunn and Lotfi A. Zadeh:Feature Extraction Studies in Fuzziness and Soft Computing,Vol.207,pp.315-324(2006).
[9] J. Han, M. Kamber, and Jian Pei:Data Mining: Concepts and Techniques [3rd ed.] Morgan Kaufmann, pp.378-386(2012).
[10] T. Hastie, R. Tibshirani and J. Friedman:統計的学習の基礎, 共立出版(2014)

[11] B.H.Menze, B.M.Kelm, R.Masuch, U.Himmelreich,P.Bachert, W.Petrich and F.A.Hamprecht. :A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC bioinformatics ,Vol.10, No.1(2009).
[12] M. Ohta, D. Arauchi, A. Takasu and J. Adachi, Empirical Evaluation of CRF-Based Bibliography Extraction from Reference Strings, *In Proc. of IAPR DAS 2014*, pp. 287-292, (2014).
[13] M.Segal:Machine Learning Benchmarks and Random Forest Regression,Technical report, Scholarship Repositry , University of Califrnia(2004).
[14] P. Treeratpituk and C. Lee Giles:Disambiguating authors in academic publications using random forests. *In Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries (JCDL '09)*, pp.39-48(2009).
[15] scikit-learn: http://scikit-learn.org/stable(2015.3.5 取得)
[16] 相澤彰子, 大山敬三, 高須淳宏, 安達淳：「レコード同定問題に関する研究の課題と現状」, 電子情報通信学会論文誌, Vol.J88 , No.3, pp.576-589 (2005).
[17] 相澤彰子, 高久雅生, 大山敬三：「大規模データベースを利用したリンケージシステムの提案と実装」,日本データベース学会 Letters , Vol.6, No.4, pp.17-20 (2008).
[18] 阿部重夫：「サポートベクトルマシン入門」, 森北出版株式会社(2011).
[19] 株式会社日立製作所中央研究所：「書誌同定システムにおける SVM 学習に関する評価報告書」, (2011)(非公開).
[20] 川上尚慶, 太田学, 高須淳宏, 安達淳 ：「少量学習データによる参考文献書誌情報抽出」, WebDB Forum (2014)
[21] 蔵川圭, 孫媛, 相澤彰子：「書誌リンケージに基づく研究分野マッピングの適合率検証」,情報処理学会第 76 回全国大会論文集,pp.433-434(2014)
[22] 徳永健伸：「情報検索と言語処理」, 東京大学出版会 (1999).

付録 A. 特徴量仕様

本稿で使用した特徴量の詳細仕様を以下に示す．(1)~(11)の特徴量は i-Linkage が持つ特徴量仕様を参考に，(12)以降の特徴量は実際のデータを確認し適当なものを作成した．引用文献文字列から書誌情報抽出器により抽出される書誌情報のうち本稿で使用した書誌情報を表 4 に示す．特徴量計算で利用する関数の仕様を表 5 に示す．また，title，journal，author は，文字列正規化処理をしたものとする．文字列正規化処理対象の文字列と正規化後の状態を表 6 に示す．なお，特徴量計算で使用する引用文献と書誌候補は次のように定義する．引用文献$\mathrm{r_{qi}} = \{\mathrm{b_{q1}}, \ldots, \mathrm{b_{qm}}\}$は書誌情報$\mathrm{b_{qj}}$の集合とする．引用文献$\mathrm{r_{qi}}$をブロッキング処理の入力とし，書誌情報データベースから抽出してきた書誌候補を$\mathrm{c_{ql}} = \{\mathrm{b_{c1}}, \ldots, \mathrm{b_{cm}}\}$とする．

(1)db_doi
書誌候補$\mathrm{c_{ql}}$ の抽出元のデータベースが CrossRef なら 1，それ以外は 0 となる．

(2)db_naid
書誌候補$\mathrm{c_{ql}}$ の抽出元のデータベースが CiNii なら 1，それ以外は 0 となる．

(3)cand_jrnl_exsist
書誌候補$\mathrm{c_{ql}}$ が持つ書誌情報に journal が存在すれば 1，それ以外は 0 となる．

(4)cand_title_exsist
書誌候補$\mathrm{c_{ql}}$ が持つ書誌情報に存在すれば 1，それ以外は 0 となる．

(5)cand_pubDate_exsist
書誌候補$\mathrm{c_{ql}}$ が持つ書誌情報に pubDate が存在すれば 1，それ以外は 0 となる．

(6)cand_volume_exsist
書誌候補$\mathrm{c_{ql}}$ が持つ書誌情報に volume が存在すれば 1，それ以外は 0 となる．

(7)cand_number_exsist
書誌候補$\mathrm{c_{ql}}$ が持つ書誌情報に number が存在すれば 1，それ以外は 0 となる．

(8)cand_page_exsist
書誌候補$\mathrm{c_{ql}}$ が持つ書誌情報に page が存在すれば 1，それ以外は 0 となる．

表 4　利用した書誌情報

書誌情報	概要
title	タイトル
author	著者
pubDate	ジャーナルの出版年
journal	ジャーナル名
page	ページ
volume	巻
number	号
url	URL
db	抽出元のデータベース

表 5　関数の仕様

関数	仕様
ed(a,b)	文字列 a と文字列 b の編集距離を計算し，区間[0,1]に正規化した値を出力する． a と b の内文字列長の大きい方で除算することで区間[0,1]に正規化する．
BM(a,b)	文字列 b に文字列 a と完全一致するものが含まれれば 1，含まれなければ 0 を出力する
LCS(a,b)	文字列 a と文字列 b の最長共通部分文字列を出力する
len(a)	文字列 a の長さを出力する
max(a,b)	数値 a と数値 b のうち大きい方を出力する

表 6　正規化対象の文字列と正規化後の状態

正規化対象の文字列	正規化後
大文字	小文字
.(ピリオド)	削除
,(カンマ)	削除
_(アンダーバー)	削除
-(ハイフン)	削除
空白	削除
)	削除
(	削除

(9)cand_author_exsist

書誌候補c_{ql}が持つ書誌情報に author が存在すれば 1，それ以外は 0 となる．

(10)title_ed

引用文献r_{qi}が持つ title と書誌候補c_{ql}が持つ title の編集距離を元にした類似度． r_{qi}_title, c_{ql}_titleはそれぞれ引用文献r_{qi}が持つ title と書誌候補c_{ql}が持つ title を示す．

$$\text{title\_ed} = \begin{cases} 0 & \text{if } \nexists \text{title} \in c_{ql} \text{ or } \nexists \text{title} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_title}, c_{ql}\text{\_title}) & \text{if otherwise} \end{cases}$$

(11)query_multibyte_ratio

引用文献r_{qi}のマルチバイト文字の比率.

(12)page_ed

引用文献r_{qi}が持つ page と書誌候補c_{ql}が持つ page の編集距離を元にした類似度． r_{qi}_page, c_{ql} _pageはそれぞれ引用文献r_{qi}が持つ page と書誌候補c_{ql}が持つ page を示す．

$$\text{page\_ed} = \begin{cases} 0 & \text{if } \nexists \text{page} \in c_{ql} \text{ or } \nexists \text{page} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_page}, c_{ql}\text{\_page}) & \text{if otherwise} \end{cases}$$

(13)vol_ed

引用文献r_{qi}が持つ volume と書誌候補c_{ql}が持つ volume の編集距離を元にした類似度． r_{qi}_volume, c_{ql}_volumeはそれぞれ引用文献r_{qi}が持つ volume と書誌候補c_{ql}が持つ volume を示す．

$$\text{vol\_ed} = \begin{cases} 0 & \text{if } \nexists \text{volume} \in c_{ql} \text{ or } \nexists \text{volume} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_volume}, c_{ql}\text{\_volume}) & \text{if otherwise} \end{cases}$$

(14)num_ed

引用文献r_{qi}が持つ number と書誌候補c_{ql}が持つ number の編集距離を元にした類似度． r_{qi}_number, c_{ql}_numberはそれぞれ引用文献r_{qi}が持つ number と書誌候補c_{ql}が持つ number を示す．

$$\text{num\_ed} = \begin{cases} 0 & \text{if } \nexists \text{number} \in c_{ql} \text{ or } \nexists \text{number} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_number}, c_{ql}\text{\_number}) & \text{if otherwise} \end{cases}$$

(15)auth_match

引用文献r_{qi}が持つ author の総数 n に対する書誌候補 c_{ql}が持つ author に引用文献r_iが持つ author に完全一致するものが含まれる割合． r_{qi}_author_j, c_{ql}_authorはそれぞれ引用文献に含まれる書誌r_{qi}が持つ author と書誌候補c_{ql}が持つ author を示す．

$$\text{auth\_match} = \frac{\sum_j^n \text{BM}(r_{qi}\text{\_author}_j, c_{ql}\text{\_author})}{n}$$

(16)pubDate_ed

引用文献r_{qi}が持つ pubDate と書誌候補c_{ql}が持つ pubDate の編集距離を元にした類似度 r_{qi}_pubDate, c_{ql}_pubDateはそれぞれ引用文献c_{ql}が持つ pubDate と書誌候補c_{ql}が持つ pubDate を示す．

$$\text{pubDate\_ed} = \begin{cases} 0 & \text{if } \nexists \text{pubDate} \in c_{ql} \text{ or } \nexists \text{pubDate} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_pubDate}, c_{ql}\text{\_pubDate}) & \text{if otherwise} \end{cases}$$

(17)auth_LCS

まず引用文献r_{qi}が持つ author の文字列長に対する引用文献r_{qi}が持つ author と書誌候補c_{ql}が持つ author の Longest Common Subsequent[1]の割合の総和の引用文献r_{qi}が持つ author の総数 n に対する割合

$$\text{auth\_LCS} = \frac{\sum_j^n \frac{\text{len}(\text{LCS}(r_{qi}\text{\_author}_j, c_{ql}\text{\_author}))}{\text{len}(r_{qi}\text{\_author}_j)}}{n}$$

(18)start_page_ed

引用文献r_{qi}が持つ開始ページと書誌候補c_{ql}が持つ開始ページの編集距離を元にした類似度．spage は開始ページを示す． r_{qi}_spage, c_{ql}_spageはそれぞれ引用文献r_{qi}が持つ spage と書誌候補c_{ql}が持つ spage を示す．

$$\text{start\_page\_ed} = \begin{cases} 0 & \text{if } \nexists \text{spage} \in c_{ql} \text{ or } \nexists \text{spage} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_spage}, c_{ql}\text{\_spage}) & \text{if otherwise} \end{cases}$$

(19)pubDate_full_match

引用文献r_{qi}が持つ pubDate と書誌候補c_{ql}が持つ pubDate が完全一致していたら 1，それ以外は 0 となる．

(20)journal_ed

引用文献r_{qi}が持つ journal と書誌候補c_{ql}が持つ jouranl の編集距離を元にした類似度． r_{qi}_jouranl, c_{ql}_jouranlはそれぞれ引用文献r_{qi}が持つ jouranl と書誌候補c_{ql}が持つ jouranl を示す

$$\text{start\_page\_ed} = \begin{cases} 0 & \text{if } \nexists \text{journal} \in c_{ql} \text{ or } \nexists \text{journal} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_jouranl}, c_{ql}\text{\_jouranl}) & \text{if otherwise} \end{cases}$$

(21)journal_LCS

引用文献r_{qi}が持つ journal の文字列長と書誌候補c_{ql}が持つ journal の文字列長の大きい方に対する引用文献r_iが持つ journal と書誌候補c_{ql}が持つ journal の Longest Common Subsequent の文字列長の割合.

$$\text{journal\_LCS} = \frac{\text{len}(\text{LCS}(r_{qi}\text{\_jouranl}, c_{ql}\text{\_jouranl}))}{\max(\text{len}(r_{qi}\text{\_journal}), \text{len}(c_{ql}\text{\_journal}))}$$

(22)url_full_match

引用文献r_{qi}が持つ URL と書誌候補c_{ql}が持つ URL が完全一致していたら 1，それ以外は 0 となる．

(23)url_ed

引用文献r_{qi}が持つ url と書誌候補c_{ql}が持つ url の編集距離を元にした類似度． r_{qi}_url, c_{ql}_urlはそれぞれ引用文献r_{qi}が持つ url と書誌候補c_{ql}が持つ url を示す．

$$\text{url\_ed} = \begin{cases} 0 & \text{if } \nexists \text{url} \in c_{ql} \text{ or } \nexists \text{url} \in r_{qi} \\ 1 - \text{ed}(r_{qi}\text{\_url}, c_{ql}\text{\_url}) & \text{if otherwise} \end{cases}$$

(24)title_LCS

引用文献r_{qi}が持つ title の文字列長と書誌候補c_{ql}が持つ title の文字列長の大きい方に対する引用文献r_{qi}が持つ title と書誌候補c_{ql}が持つ title の Longest Common Subsequent の文字列長の割合.

$$\text{title\_LCS} = \frac{\text{len}(\text{LCS}(r_{qi}\text{\_title}, c_{ql}\text{\_title}))}{\max(\text{len}(r_{qi}\text{\_title}), \text{len}(c_{ql}\text{\_title}))}$$

(25)pubDate_err_full_match

引用文献r_{qi}が持つ pubDate に±1 した値と書誌候補c_{ql}が持つ pubDate が完全一致していたら 1，それ以外は 0 となる．