2005－DBS－137（Ⅱ）（52）
2005／7／14

共著関係に基づくグラフを用いた書誌情報における著者同定手法の提案と評価

鈴木　康平†, 正田　備也‡, 高須　淳宏‡, 安達 淳‡

† 東京大学 情報理工学系研究科
‡ 国立情報学研究所

学術文献データベースの著者名検索においてイニシャル等を使用するため同表記の別人を区別出来ないという問題がある．この問題に対しては従来のテキスト情報のみを用いる方法では十分な結果が出ていない．そこで我々は論文の共著関係に基づく情報を用いる手法を提案する．まず，論文における著者名をノードとして共著関係があるノードどうしにエッジを張ることによりできる共著グラフを生成する．グラフ理論においてあるノードを取り除くことによってグラフの連結成分が増える時，このノードを切断点と呼ぶ．共著グラフにおいて切断点は複数の著者に対応する可能性が高いと考えられる．そのため，切断点を取り除いて得たそれぞれの連結成分中にあるノードを基に論文を分類することで，異なる著者ごとに論文を分類することが出来ると考えられる．本稿ではまた，提案手法を実際の学術文献データベースに適用してその有効性を評価した結果を報告する．

Author Identification for Bibliographic Information using Coauthor Relationship Graph

Kouhei SUZUKI†, Tomonari MASADA‡, Atsuhiro TAKASU‡, and Jun ADACHI‡

†Graduate School of Information Science and Technology, The University of Tokyo
‡The National Institute of Informatics

The author name search on citation literature databases shows a problem when the same name refers to more than one author. To identify individual authors, the method based on only text information can't achieve a good result. Therefore, we proposed a new method that uses the information based on coauthor relationship. We report in this paper a new method for generating a coauthor graph representing author name as nodes and coauthor relationship as edges. When the number of connected components increases by removing a node, this node is called cut vertex in graph theory. We use the following intuition: cut vertices tend to correspond to multiple authors. Therefore, we can classify papers written by the same author, if we classify papers based on nodes in each connected component.The proposed method is applied to citation literature databases to examine the effectiveness.

1　まえがき

学術文献データベースは長い歴史を持つが，文献の書誌情報と引用情報の記述に誤りなどが混入し，さまざまな不整合が生じているため，利用しにくくなっている．そのため同一のレコードを自動識別するための技法もいろいろと研究されてきた．[1][3][12]．また，これを利用しインターネット上にある論文情報を自動的に収集し，引用解析をしている例も見られる.[2]．しかし，その同定もあまり十分に行えていないのが現状である．

この学術文献データベースにおいて研究者の名前で論文を検索すると同姓同名の別人による論文が混在す

る形で結果が出力される．このため，検索者が自分の目当ての著者による論文を探すのに非常に手間がかかるという問題がある．このような背景から我々は学術文献データベース上で著者を識別するために著者の共著関係を著者識別に利用することを試みた．

ここで学術文献データベース上での著者の識別問題を著者同定問題と呼ぶことにする．本稿では，まず第2節で著者同定問題について詳しく述べる．次に第3節で提案手法について説明し，第4節で予備実験の結果，そして第5節でまとめと今後の課題を述べる．

2　著者同定とは

学術文献データベースとして著名な米国 Thomson Scientific 社の SCI(Science Citation Index) では，著者名は姓，名のイニシャルの形に正規化される．例えば，本稿の第一著者の場合は suzuki,k となる．これにより，同姓の著者の識別が困難になることが容易に想像される．

名前をイニシャルにするだけで著者名がどれだけ重複するかを国立情報学研究所の論文検索ナビゲーター CiNii のデータベースに登録されているレコードの内40万件のレコードで調べた結果が表1である．

表 1: フルネームと名前がイニシャルの場合の重複する表記の数

	フルネームの時	名前がイニシャルの時
suzuki,toshio	102	2397
tanaka,katsumi	155	2365

表1が示すように名の部分をイニシャルにするだけでフルネームの時に比べて該当する論文数がそれぞれ15から20倍と大幅に増加することがわかる．因みに名をイニシャルにした際に重なる異なる著者の数を調べた結果 suzuki,t で 180 人，tanaka,k で 176 人となった．

このように，学術文献データベースの検索において同一表記の著者の重なりは非常に問題となっている．フルネームで登録するとある程度は緩和されるが，依然として本質的な同姓同名の著者の問題が残っている．そのため著者同定技術は今後，論文数が増えていくにつれ，ますます必要性が高くなると思われる．

この著者同定問題に対して従来のテキスト分類手法を応用した研究が行われている [9][5][6][10]．しかしながら，引用情報に含まれるテキスト情報は論文の著者名，タイトル，発表学会名程度であることが多い．これらの研究結果より，このような少量のテキスト情報からではテキスト分類手法では精度よく著者を識別することは困難であることが示唆されている．

そこで，本研究では著者同定問題に対して，従来手法とは異なるアプローチを提案する．それは論文の共著者から著者の共著関係を抽出し，それを著者の判別に用いるというものである．

3　提案手法

3.1　概要

本研究では著者の共著関係のみに着目した．しかしながら，共著者のいない論文に対して本手法はそもそも適用することができない．そこで，最終的にはこの共著関係情報とテキスト情報を合わせることによってテキスト情報のみに頼る従来の手法の精度を向上させることができるのではないかと考えている．

図1が提案手法の概要である．この図で言うデータ集合とはある著者名が与えられたとき，元の学術文献データベースからその名前を含むレコードを抽出したものである．本手法は図のように3つの処理から構成される．1つ目の処理は本手法の特徴である共著関係を利用した分類手法を用いる．詳しいことは後述するが，この手法は共著関係を共著グラフというグラフで表す．共著グラフとは著者名をノードとしある論文において共著関係のある著者名にエッジを繋ぐことによって出来るグラフである．次に共著グラフの解析結果に基づいて分類されたレコード集合に対して各々テキスト情報による類似度判定を行うことにより更に細かくレコード集合を分割する．しかし，この状態ではデータを細かく分類し過ぎてしまっていると考えられる．そこで最後にこれらのレコード集合を一つのテキストと見なし，テキスト情報による類似度判定を用いてクラスタリングし，レコード集合の統合を行う．

3.2　共著グラフ

3.2.1　共著グラフによるレコード分類処理

共著グラフの具体例は図2のようになる．この図は q という著者名を持つ5つの論文に対する例で図中の1〜5が論文を表し，アルファベットが著者名を表す．数字の横に記してあるアルファベットがその論文の著者名である．論文1の著者名は q，a，b ということに

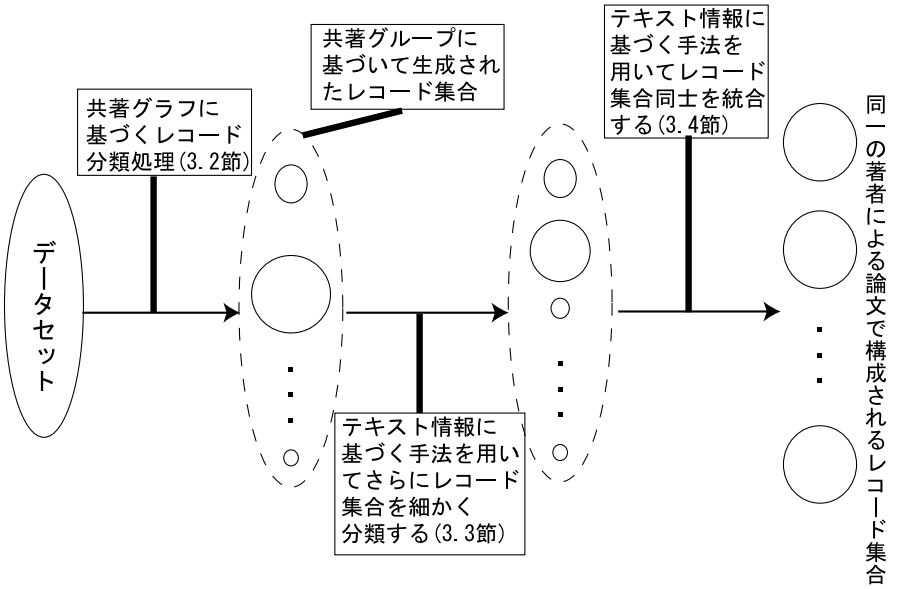

図 1:提案手法の概要

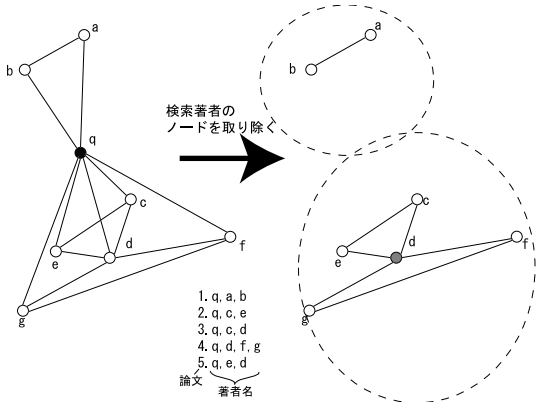

図 2:共著グラフの例

なり，これらのノードに共著関係があるのでエッジを張る．

こうしてできた共著グラフに対して検索著者名のノードをグラフ上から取り除くと，この例の場合，グラフは 2 つに分かれる．このような点をグラフ理論では切断点と呼ぶ．この例のように検索著者名ノードが共著グラフの切断点となっている時，これを取り除いてできた連結成分に対して，その中に切断点がないかどうかを探す．例の場合，下の方の連結成分に属するノード d が切断点となる．検索著者名ノードが切断点であった場合は無条件に連結成分に分けるが，連結成分中に存在した切断点に関してはある条件を満たす時のみ，さらに細かく部分グラフに分ける．また，検索著者ノードの場合とは異なり，この場合の切断点は連結成分の数だけ複製して各々の連結成分に含ませることにする．その条件とは図 3 のように切断点を 2 つ含む連結成分が出来る時である．図 3 において黒塗りの点が切断点を，実線が切断点を取ることによって単純に出来る部分グラフを表し，点線が条件によって分けられた部分グラフである．

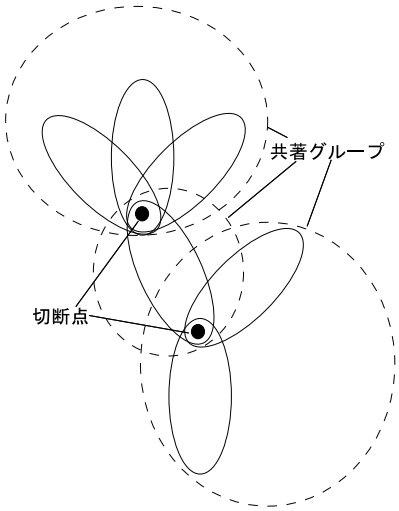

図 3:部分グラフにわける条件の例

この時のみ図 3 のように 3 つの連結成分に分ける．よって図 2 の例において切断点 d ではグラフを分けないことになり，最終的に 2 つの連結成分が作成される．よってレコードはこの連結成分に属するノードの著者名をもとに分割される．この例では a,b を含むレコード 1 で構成される集合と c,d,e,f,g を含むレコード 2, 3, 4, 5 で構成される集合の 2 つに分割される．この

ようにして分割されるレコード集合は排他的である．

3.2.2 切断点の意味

検索著者名ノードがグラフの切断点であるということ，それはこの著者名が少なくとも 2 つの共著者のグループを持つということである．名前の表記が同じである 2 人の著者がいた場合，その 2 人の著者が同じ共著者と論文を書いているということはあまりないと考えられる．つまり，検索著者名が 2 つの共著者のグループを持つということはその著者名に対応する著者が 2 人いる可能性があるということになるはずである．ここで可能性という言い方をしたのは多くの場合，研究者は複数の研究グループに所属していることがあるため，これだけでは本当に 2 人の異なる著者を持つ著者名であるのかどうかを断定することは出来ないからである．

次に連結成分中の切断点を探す理由は検索著者名だけでなく共著者の名前の中にも複数の著者に対応しているものがある可能性が考えられるからである．特に第 2 節で紹介した米国 Thomson Scientific 社の SCI(Science Citation Index) では著者名が「姓」と「名のイニシャル」という形で正規化されており，本手法においてもこの場合を想定している．そのため，第 2 節で述べたように名前をイニシャルにすると同じ表記となる著者名が増大していることが考えられる．よって前記のような可能性を考慮する必要性が十分にある．

共著者の名前も複数の著者に対応している場合，検索著者名を基にグラフを分けたとしても十分に分けることが出来なくなる．そこで，検索著者名ノードを取り除いて出来た連結成分に切断点がないかどうかを探すことなる．この切断点はグラフ理論における頂点連結度が 1 の場合に相当する．この頂点連結度を求める問題はグラフ理論において最も基本的な問題の一つであり，多くの効率的なアルゴリズムが研究されている．これらのアルゴリズムは基本的に自己ループと平行枝を持たない無向グラフを想定したものである．本手法における共著グラフではエッジは共著者に張られるものであり，自分自身へ張られることはないので自己ループを持たないという条件を満たしている．また，複数の論文で共著していようと一つの枝で共著関係を表しているので平行枝が無いという条件も満たしている．

上記に基づき現在の実装システムでは S.Even and R.E.Tarjan のアルゴリズムを用いている [4]．このアルゴリズムでは頂点連結度を求めるのに必要な時間計算量は頂点数を V，エッジの数を E とした時，$O(|V|^2 \cdot |E|)$ である．本手法においては頂点連結度が 1 の場合のみを求めていることもあり実用上十分な処理速度が得られている．

最後にこのようにして求めた切断点においてグラフを分割する際の条件付けの意味を述べる．連結成分中に切断点が一つしかない場合は共著者に研究室を取りまとめる教授と助手の先生の名前が載っているような論文が考えられる．この時，メインの著者は大抵，研究室所属の学生であり，このような論文は研究室の学生の数だけ存在する．この場合，そのまま分けてしまうとこれらの学生の数だけ連結成分が作られてしまう．しかし，これらは当然全てまとめられるべきである．つまり，このような切断点は複数の人間に対応していないと考えるわけである．

そして連結成分中に 2 つ以上の切断点があった場合は前述したように図 3 に分ける．この図の場合，これら 2 つの切断点すべてが複数の著者に対応していると考えられるため，2 つの切断点を一つずつ含むものと二つの切断点を両方含むものの 3 つの連結成分を作ることになる．

最後に共著グラフ作成からレコード集合作成までの流れは以下のようになる．

1. 検索著者名を含むすべての論文から著者名を抜き出す
2. 一つの著者名を一つのノードとして共著グラフを作る
3. 検索著者名ノードを取り除き，連結成分に分ける
4. 全ての連結成分において切断点がないかを調べる
5. 切断点があったなら，条件に当てはまるかを判別し，最終的な連結成分群を作る
6. この最終的な連結成分に含まれる著者名に基づいてレコード集合を作成する

3.3 テキスト情報によるレコードの再分割

この処理の目的は先の共著関係だけでは正しく分類出来なかった異なる著者によるレコードをテキスト情報を利用した方法で分離することである．タイトルや発表学会名などのテキスト情報と共著関係は異なる情報源である．よって，共著関係を利用した分類とテキスト情報を利用した分類を両方行うことにより相互補

完的な効果が期待できると考えられる．

テキスト情報を利用するこの処理と次のレコード集合の統合処理についての具体的な方法は現在検討中である．

3.4 テキスト情報によるレコード集合の統合

この処理では，2 つの情報に基づいて細分化されたレコード集合を同一の著者によるレコードを一つのレコード集合にまとめる形で統合することが目的である．先のレコード分割処理において，一つのレコード集合中に含まれるレコードは，単一の著者によるものに統一されていることが期待できる．そのため，レコード集合に含まれるレコードのテキストをまとめて単一の著者に関する一つのテキストとして扱うことが出来る．このことにより，レコード単体に比べて含まれる情報の量が増え，従来手法の問題点であった有効なテキスト情報の不足を解消することが出来ると考えられる．そして情報の量が増えるので，これらのレコード集合どうしを著者に関してクラスタリングする時にレコード単体に比べて，精度の良いクラスタリングが期待出来る．

4 実験

4.1 実験について

提案手法である共著グラフを用いたレコード分類処理の性能を評価するための実験を行った．本実験では国立情報学研究所の論文検索ナビゲーター CiNII のデータベースに登録されているレコードの内 40 万件をデータとして用いた．これらのレコードの著者名は全て姓と名前のイニシャルの形に変換した上で，40 万件のデータから登録されているレコードの数が 1500 件程度，500 件程度，300 件程度の著者名をそれぞれ 5 つずつ合計 15 個抽出し，それらの著者名に対してレコード分類実験を行った．

CiNII のデータは著者を識別するためのタグなどが一切ないため，今回は著者名をフルネームとした時に同じ表記になる著者名を同じ著者を指すものとして正解とした．

評価に関しては M,Rosell らの提案した purity を用いた [11]．これは precision をもとにした評価関数で n_{ij} を i 番目のレコード集合中に含まれる j 番目の著者によるレコードの数，n_i を i 番目のレコード集合中に含まれるレコードの総数とした時，レコード集合ごとの purity，ρ_i は $\rho_i = \max_j \left\{ \frac{n_{ij}}{n_i} \right\}$ となり，レコード集合全体の purity，ρ は $\rho = \sum_i \frac{n_i}{n} \rho_i$ となる．ここで n は全レコード集合に含まれるレコードの総数である．

4.2 実験結果

分類実験の結果を検索著者名ごとにその著者名でヒットしたレコードの総数，フルネームにした際に重複していた名前の数，レコードを一つしか含まないレコード集合の数 S_{n1}，最も多くのレコードを含むレコード集合に含まれるレコードの数 S_{nmax}，purity，最大のレコード集合を除いて計算した purity である n-purity でまとめたのが表 2 である．ただし，検索著者名でヒットしたレコードであっても共著者のいない論文については本手法では分類の対象としていないので著者名でヒットしたレコードの総数には含んでいない．purity についてはレコードを一つしか含まないレコード集合は当然 100 %になるため計算には含めていない．

なお，実験を行った共著グラフによるレコード分類処理は第 3 節で述べたように同じ著者によるレコードのみを含む集合を作ることであり，同じ著者によるレコードを全て一つの集合にまとめることが目的ではないため，recall による評価は行わなかった．

さらに表 3，表 4 が著者名 watanabe,k と yajima,h のレコード集合のサイズの内訳を示したものであり，レコード集合のサイズごとに，レコード集合の個数 N_s と，同サイズのレコード集合群だけで計算した purity である s-purity を示している．

表 3 : watanabe,k のレコード集合の内訳

レコード集合のサイズ	N_s	s-purity
800	1	0.105
21	1	0.905
14	1	1.0
12	1	0.667
11	1	0.273
8	2	0.750
7	7	0.735
6	8	0.729
5	9	0.711
4	16	0.703
3	41	0.854
2	61	0.869

表 2:検索著者名ごとのレコード分類の結果

検索著者名	レコード総数	重複する著者名の数	S_{n1}	S_{nmax}	purity	n-purity
watanabe,k	1574	164	233	800	0.384	0.797
tanaka,m	1437	145	209	707	0.401	0.854
sato,k	1417	208	220	693	0.394	0.855
yamamoto,t	1390	160	201	719	0.424	0.868
suzuki,h	1286	111	165	699	0.480	0.879
fukuda,t	503	62	84	175	0.721	0.951
okada,t	466	102	101	107	0.712	0.888
okamoto,y	447	60	67	243	0.603	0.927
kawai,t	412	57	67	115	0.745	0.909
takeda,y	381	63	81	126	0.690	0.937
yajima,h	250	30	30	188	0.927	0.938
fujimoto,s	244	36	33	146	0.540	0.969
matsumura,m	231	32	43	68	0.867	0.967
hamada,y	215	42	44	33	0.959	0.957
koike,k	189	40	34	63	0.768	0.902

表 4:yajima,h のレコード集合の内訳

レコード集合のサイズ	N_s	s-purity
188	1	0.926
5	1	1.0
3	3	1.0
2	9	0.889

4.3 実験結果の分析

表 2 より，レコードの総数の大小に関わらず半数近くのレコードが一つのレコード集合に固まっていることがわかる．このようなレコード集合はほとんどが多数の異なる著者によるレコードで構成されており，そのためにレコード集合全体の精度が低下してしまっていた．このような多数の著者によるレコードを含む大規模なレコード集合によって精度が低下することを調べるため，最も大きいサイズのレコード集合を除外して purity を計算した．これともとの purity を見比べると明らかに精度が大幅に向上することがわかる．そのため，本手法の精度を高めるためにはこのような大きなサイズのレコード集合が出来ないように工夫する必要があると考えられる．

しかしながら，大きなサイズのレコード集合が必ずしも精度低下の原因となっているわけではない．3，表 4 は著者名 watanabe,k と yajima,h のレコード集合の内訳を示したものである．もともとのレコードの総数の違いがあるもののどちらもレコード総数の半分以上を占める大きなレコード集合を持つが，その精度が全く異なっている．この yajima,h の場合のようにサイズの大きいレコード集合の purity が高いこともある．

では，多数の異なる著者によるレコードが一つに固まってしまうのは何故だろうか．一つ考えられるのは多数の共著者を持つ論文の存在である．著者名を姓と名のイニシャルという形にするとその著者名に対応する著者が爆発的に増えることは第 2 節で述べた．そのため，一つの論文に共著者が多数居る場合，それらの共著者の名前に別の論文における異なる著者名が重なってしまうことが共著者の数だけ多くなると考えられる．こうなると共著グラフにおいてノード間の枝が複雑に絡みあい，サイズの大きい連結成分が出来てしまうことになる．サイズの大きい連結成分にはそれだけ多くのノード，つまり著者名が含まれるため，これをもとに作られるレコード集合のサイズは当然大きくなる．そこで改良案としてはある人数以上の共著者いる場合はその共著関係を共著グラフに反映させないことが考えられる．実験で用いた 40 万件のデータ中に共著者が 5 人以上いる論文について調べたところ，93180 件もあることがわかった．ただ，あまり共著グラフに反映させる論文を減らしてしまうと，今度は十分な共著関係を得られなくなる可能性があるため，何

人以上の共著者がいる論文を除外するかは今後実験により適当な値を決める必要があると考えられる．

次に問題となるのがレコードを一つしか含まないレコード集合の存在である．これを減らすためには検索著者名以外の切断点でグラフを分割しない，もしくはその条件をもっと緩くするという手段しかないと思われる．しかし，このようなことをすると，極端に大きな集合ができやすくなり，分類精度が低下する．本手法においては最後にテキスト情報を用いたレコード統合処理を考えている．表 2 の n-purity を見ればわかるように極端に大きなレコード集合を除いた場合，共著グラフを用いたレコード分類処理の前処理としての性能は高く，これにテキスト情報を用いた分類処理を追加で行うことにより更なる精度の向上が見込まれる．この時，同一の著者のみで構成されたレコード集合がいくつか存在すれば，レコードを一つしか含まないレコード集合があったとしても第 3 節で述べたように個々のレコードどうしでクラスタリングを行うよりも高い精度が期待できると考えられる．

最後に表 5 が著者名 watanabe,k の 3 つのレコードを含むレコード集合中のレコードの例である．フルネームの異なる著者が混ざっているが，注目すべきはタイトルとジャーナル名である．この 2 つのテキスト情報からこの二人は分類しやすいと予想される．この例のように共著グラフによる分類処理で作成されたレコード集合に対してテキスト情報を用いた分類処理を個々のレコード集合ごとに追加で行うことにより更に分類精度を向上させることが出来ると考えられる．

5 まとめと今後の課題

同表記の別人の著者による論文を区別するための方法として論文における共著関係に基づく情報を利用する方法を提案した．

本稿においてはこの共著関係に基づく情報を利用する手段として共著グラフを作り，そこから共著グループを抽出し，それに基づいて論文を分類する方法について述べた．またこの方法の有効性を確かめるための実験を行った．

その結果，本手法は，1 つしかレコードを含まないレコード集合と，極端に大きなサイズの 1 つのレコード集合を除く，他のレコード集合は，高い purity を示すことがわかった．また，これらのレコード集合のうち，フルネームの異なる著者が混ざっているものであっても，論文のタイトルやジャーナル名を利用したテキスト情報による分類処理によって，フルネームが同じ著者を含むレコードの集合へと分割することが見込めるとわかった．

その上で今後は極端に大きなサイズのレコード集合を出来ないようにするために共著者の人数が多い論文については共著グラフに反映させないということを試してみる予定である．そして，テキストベースの類似度判定処理を実装し，システム全体の性能評価を行いたいと考えている．

他に考えられる課題としては，共著グラフによる分類処理の高速化が挙げられる．現在は Even ら [6] のアルゴリズムによって，切断点を検出した．だが，永持らが，任意の頂点対の頂点連結度を保存したまま，枝の本数を減らす手法を提案している [7][8]．この手法によって，共著グラフの枝数を減らしておくと，一つの頂点対について，切断点を求める時間計算量が，$O(|E|)$ から $O(|V|)$ に減る．よって，永持らの方法を実装することで共著グラフによる分類処理の高速化ができると考えている．

参考文献

[1] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 39–48, 2003.

[2] K. Bollacker, S. Lawrence, and C. L. Giles. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Proceedings of the Second International Conference on Autonomous Agents*, pp. 116–123, New York, 1998. ACM Press.

[3] P. Christen and T. Churches. Febrl - freely extensible biomedical record linkage. *Computer Science Technical Reports, TR-CS-02-05, Australian National University*, 2002.

[4] S. Even and R. E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, Vol. 4, pp. 507–518, 1975.

[5] H. Han, C. L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of JCDL*, pp. 296–305, 2004.

表 5 : watanabe,k のレコード集合中のレコードの例

著者名	タイトル	ジャーナル名
watanabe,kouhei	高専におけるインターネット活用と情報倫理教育	電子情報通信学会
watanabe,kouhei	電子教科書を用いたプログラミング教育	電子情報通信学会
watanabe,koichiro	Continuity in Phase Transition Behavior between Normal and Diffuse Phase Transitions in Complex Perovskite Compounds	物理系学術誌

[6] H. Han, H. Zha, and L. Giles. Name disambiguation in author citations using a k-way spectral clusering method. In *Proceedings of JCDL*, pp. 334–343, 2005.

[7] H. Nagamochi. Graph algorithms for network connectivity problems. *Journal of the Operations Research Society of Japan*, Vol. 47, No. 4, pp. 199–223, 2004.

[8] H. Nagamochi and T. Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, Vol. 7, pp. 583–596, 1992.

[9] B. W. On, D. Lee, J. Kang, and P. Mitra. Comparative study of name disambiguation problem using a scalable blocking-based framework. In *Proceedings of JCDL*, pp. 344–353, 2005.

[10] S. Oyama and C. D. Manning. Using feature conjunctions across examples for learning pairwise classifiers. In *Proceedings of 15th European Conference on Machine Learning*, pp. 322–333, Pisa, Italy, 2004.

[11] M. Rosell, V. Kann, and J. E. Litton. Comparing comparisons: Document clustering evaluation using two manual classifications. *In ICON*, 2004.

[12] 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-D1, No. 3, pp. 576–589, 2005.