様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ６ 月 １６ 日現在

機関番号：３４４０７
研究種目：若手研究(B)
研究期間：2012～2014
課題番号：２４７００９１８

研究課題名（和文）ソースコードの内容と記述特徴に着目した統合プログラミング授業支援システムの開発

研究課題名（英文）A development of an integrated programming class support system focusing on the similarity based on the content of source codes and author's coding style

研究代表者
大野　麻子（Ohno, Asako）

大阪産業大学・工学部・講師

研究者番号：９０５５０３６９

交付決定額（研究期間全体）：（直接経費）　　2,200,000円

研究成果の概要（和文）：本研究では，これまでに提案した「参照ベクトルを用いた類似性検出手法(FRef)」および「作成者の記述特徴に基づく類似性検出手法(CMアルゴリズム)」を改良し，授業課題ソースコードの特徴に合わせた採点・盗用発見および可読性評価機能としてプログラミング授業支援システムに実装した．実際の授業で提出された授業課題ソースコードを用いて評価を行った結果，盗用発見機能や可読性評価については手作業による評価に近い結果が得られることが確認されたが，採点機能については更なる精度の向上が必要であると考えられる．

研究成果の概要（英文）：In this study, I improved the two source code similarity measuring methods that I had proposed in my former study, that were, "A similarity measuring method using reference vector (FRef)" and "A similarity measuring method based on author's coding style (CM Algorithm)". I implemented the two methods to a programming class support system to achieve auto-scoring, plagiarism detection, and coding style evaluation functions. From the results of the evaluation experiments using a set of source codes produced by students in real-world programming class as test data, I confirmed high performance for plagiarism detection and coding style evaluation function while scoring function needed more improvement to be used in practice.

研究分野：知的学習システム

キーワード：知的学習システム　教育工学　授業支援システム　盗用発見　類似性検出　採点支援　授業課題ソースコード　特徴抽出

１．研究開始当初の背景

プログラミング授業における採点・盗用発見には多大な時間と労力が要求され，担当教員の授業改善への取り組みの妨げとなっている．

授業課題ソースコードは一般に，(性質 A)「行数が少ない」，(性質 B)「授業で習った範囲の知識で作成されるため互いに内容が類似している」という独特の性質を持つ．このため，採点においては(問題 A)「類似性検出に十分な特徴の抽出が難しい」，盗用発見においては(問題 B)「内容に基づく類似性検出では偶然のアルゴリズムの一致を盗用と誤判定してしまう」という問題がある．

２．研究の目的

本研究ではこれまでに，「参照ベクトルを用いた類似性検出手法（FRef）」と「作成者の記述特徴に基づく類似性検出手法(CM アルゴリズム)」という二つの手法を提案した．

前述の(問題 A)（問題 B)を解決するよう「FRef」と「CM アルゴリズム」を改良し，授業課題ソースコードに特化した類似性検出手法として提案し，これらを採点機能および盗用発見機能として実装した統合プログラミング授業支援システムを開発する．

システムの提供する採点・盗用発見機能によりプログラミング授業における教員の肉体的・精神的負担を軽減することで，授業改善に集中出来る環境を提供し，間接的にプログラミング授業の教育効果の向上に貢献することを目指す．

３．研究の方法

まず，既存のプログラミング授業支援システムについて調査し，システムに必要な機能について整理する．また，採点を自動化するにあたり，その基準については既存研究の間でも様々な提案がなされている．これらについて教員の負担等も考慮しながら検討を行う．また，既存手法を本手法と組み合わせて用いることで採点・盗用発見機能の精度の向上を図る．これらの結果を踏まえて，次の各内容を行う．

(1)　「FRef」による採点機能の実現

図 1 に示す手法（FRef）は既存手法のように一対のソースコード（正解となるソースコードと学生の提出したソースコード）を直接比較するのではなく，参照ソースコードという共通の第三者のソースコードを用いて特徴ベクトル（参照ベクトルとよぶ）を生成し，これらを比較することで間接的に類似度を求める手法である．このため，短いソースコードから生成した低次元の特徴ベクトルを用いて高速な類似性検出が行えるという特長がある．しかし，あくまでソースコードの内容（標準化された単語の分布）の類似に基づく指標であるため，授業課題ソースコードのように大きな違いが生じにくい対象から類似性を検出することは困難である．そこで本研究では，既存手法で用いられている中でも単純なソフトウェアメトリクスを元にした指標を併用することで，本手法の特長を維持しながらも類似性検出精度を向上させることを試みる．

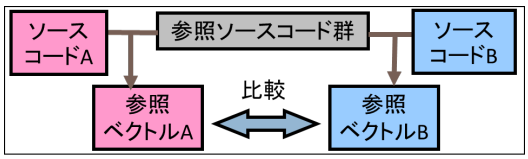


図 1 参照ベクトルを用いた類似性検出手法（FRef）

(2)　「CM アルゴリズム」による盗用発見機能実現および可読性評価手法の提案

図 2 に示す手法(CM アルゴリズム)では対象となる作成者の記述特徴を複数のソースコードからあらかじめ抽出し記述スタイルモデルという隠れマルコフモデルに学習させておく．記述スタイルモデルの表す特徴と学生の提出した課題ソースコードから新たに抽出した記述特徴を比較することで作成者認証を行い，盗用発見機能を実現する．本研究では，記述スタイルモデルを一部改良し，より詳細な類似性検出を可能にするとともに，本手法を用いた可読性評価手法についての提案を行う．

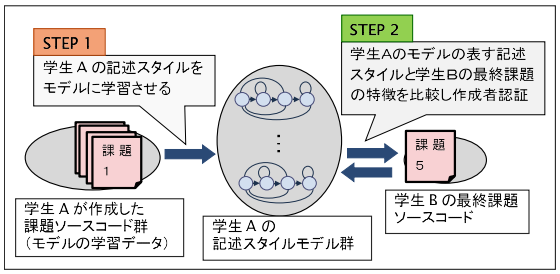


図 2 作成者の記述特徴に基づく類似性検出手法（CM アルゴリズム）

(3)　統合プログラミング授業支援システムの開発

図 3 は提案手法を実装したシステムの概要である．本システムは「データベースを有するサーバ」と「教員・学生がブラウザから

アクセスするクライアント」により構成される．本システムのサーバには次のデータが格納される．(1)学生の作成した課題ソースコード，(2)(1)から抽出した記述スタイル特徴，(3)(2)を学習させたモデル，(4)(3)を用いて検出した最終課題ソースコードとの記述スタイル特徴の類似性（盗用発見結果），(5)毎回の課題の模範解答，(6)(5)と学生の課題ソースコードの類似性（採点結果），(7)(5)と学生の記述スタイル特徴の類似性検出結果（可読性評点）．学生用クライアントからは課題ソースコードの提出と採点結果の閲覧を行うことができる．教員用クライアントからは採点，盗用発見結果の閲覧を行うことができる．教員・学生クライアント用には Web ベースの GUI，研究者用には CUI が提供される．可読性評価結果と採点結果の相関の分析はシステムの機能ではなく研究者が直接行う．

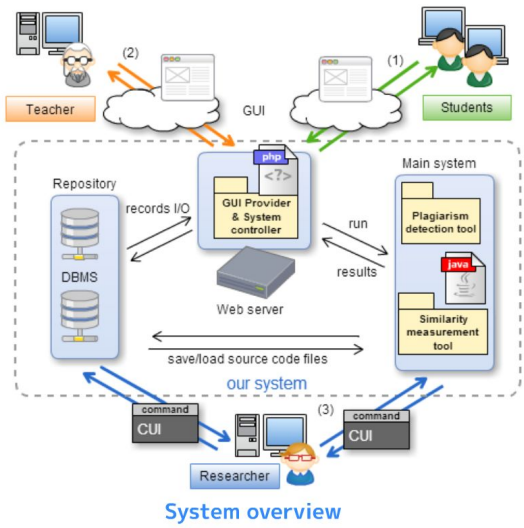


図 3 システム概要

(4) その他（多言語化，他形式データへの適用，精神的負担の軽減効果について）

二つの提案手法は Java 言語で書かれたプログラムソースコードを対象としていたが，C 言語で書かれたソースコードにも対応可能とするための改良を行う．

また，CM アルゴリズムは系列データとみなせるものであればソースコード以外のデータにも理論上適用が可能である．他形式のデータへの適用を通して本手法の評価を行う．

さらに，CM アルゴリズムを用いた盗用発見が教員・学生の精神的負担を軽減可能であるかについてアンケートや聞き取り調査の結果を元に議論する．

4．研究成果

本研究では図 1 に示す「参照ベクトルを用いた類似性検出手法（FRef）」および図 2 に示す「作成者の記述特徴に基づく類似性検出手法（FRef）」を改良し，それぞれの手法を盗用発見機能および採点機能として実装する授業支援システムの開発を行った．

(1) 採点機能実現のための「FRef」の改良

表 1 はソフトウェアメトリクスを元にした特徴量を算出するために使用したメトリクスの一覧である．

表 1 本研究で使用したメトリクス

メトリクス名	内容
予約語	”import”，”protected”，”public”，”private”，”if”，”for”，”while”，”switch”，”IOException”，”try” の 10 種類の予約語の出現数
構造 1	ネストの深さ
構造 2	メソッド数
サイズ	ソースコードの行数

上記により 13 次元のメトリクスベクトルを生成し，ベクトル間の標準化距離により「正解」との類似度を算出した．

表 2 は FRef およびメトリクスを用いたときの採点結果（類似度スコア），類似順位と手作業による採点結果である．類似度スコアは 0〜1 の値をとり，正解と同一の場合 1 となる．手作業による採点結果は 0〜100%の値をとる．

表 2 FRef およびメトリクスを用いた採点結果と手作業による採点結果の比較

file name	similarity[1] (FRef)	rank (FRef)	similarity (metrics)	rank (metrics)	score	
a_8.c	1.00	1	1.00	1	100%	Exemplar source code
s1_8.c	0.58	12	0.54	11	69%	
s2_8.c	0.75	6	0.72	6	100%	
s3_8.c	0.31	81	0.06	117	63%	
s4_8.c	0.75	5	0.74	4	88%	
s5_8.c	0.74	8	0.69	8	100%	
s6_8.c	0.79	3	0.77	2	100%	
s7_8.c	0.00	128	0.00	124	0%	
s8_8.c	0.70	9	0.63	9	94%	
s9_8.c	0.57	13	0.48	13	100%	
s10_8.c	0.53	14	0.39	14	100%	
s11_8.c	0.68	10	0.62	10	100%	
s12_8.c	0.76	4	0.72	7	94%	
s13_8.c	0.61	11	0.54	12	88%	
s14_8.c	0.75	7	0.74	4	100%	
s15_8.c	0.80	2	0.75	3	100%	

FRef，メトリクスの双方において，手作業で行った採点結果に近い結果を得られることが確認された．しかしながら，現時点の評価指標では正解ソースコードの特徴ベクトルと各学生の作成した採点対象ソースコードの特徴ベクトルとの距離に基づく総合的な類似判定を行うのみであり，正解と大きく異なる冗長な構造を持ったソースコードや，既習得範囲以外の要素をもつソースコード

に対し低い類似度が算出される可能性がある．また，内容に関する詳細な評価を行い，学生に適切な助言をあたえるためには，目視による確認作業が要求される．この対策として，課題の内容に応じ使用するメトリクスを決定し，ベクトルを算出する方法や，出題意図を確認する出題を併用する方法が考えられるが，いずれも教員の労力増大とのトレードオフが懸念される．

(2) 手法 CM アルゴリズムの改良

本手法では作成者から抽出した記述特徴を隠れマルコフモデルの構造やパラメータにより表現する．モデルの例を図 4 および図 5 に示す．「{」や「(」など，あらかじめ定義した 14 種類の記号を基準としてその前後における特定の記号（「空白」，「タブ」，「改行」）の出現傾向を表している．

図 4 のモデルは記号「{」の前方，図 5 は後方における記号の出現傾向を表している．ここで，「空白」の文字数を 1～4 文字と分け，それぞれについて観測確率を求めたことで，より詳細な記述特徴の表現が可能となっている．

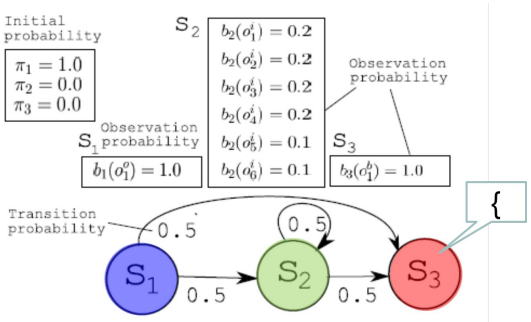


図 4 前方記述スタイルモデルの例

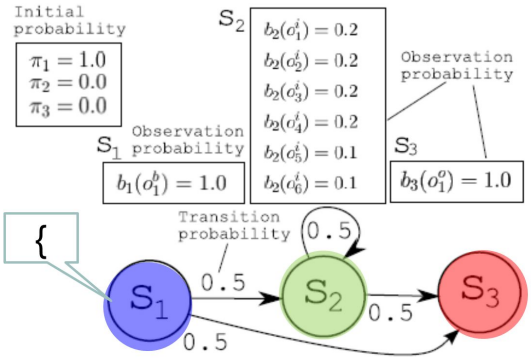


図 5 後方記述スタイルモデルの例

図 6～図 9 は CM アルゴリズムによる可読性チェック結果の例である．ここで，「可読性の高さ」の指標はあらかじめ定義した記述スタイルに基づき作成したソースコードの記述スタイル（手本）との距離である．

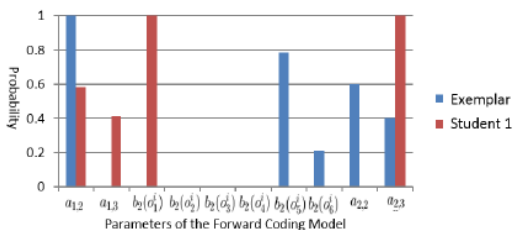


図 6 手本(青)と学生 1(赤)の前方記述スタイル比較

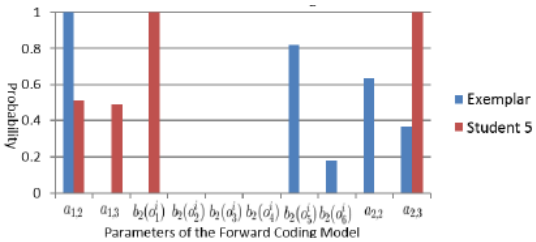


図 7 手本(青)と学生 5(赤)の前方記述スタイル比較

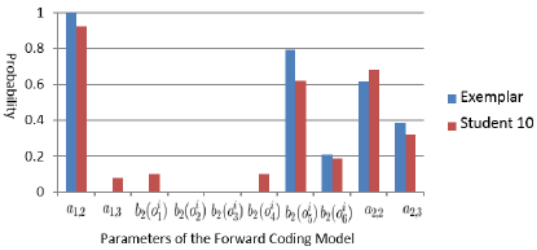


図 8 手本(青)と学生 10(赤)の前方記述スタイル比較

学生 1（図 6 赤）および学生 5（図 7 赤）は前方記述スタイルが類似しており，学生 10（図 8 赤）とは大きく異なること，学生 10 のみ手本（図 6～図 8 青）と類似していることが分かる．また，各棒グラフは図 4 に示す前方記述スタイルモデルのパラメータに対応しており，これにより例えば学生 10 は記号「{」の直前にそれぞれ 10%の確率で 1 文字または 4 文字スペース，60%の確率でタブ，20%の確率で改行を入れていることがわかる．この情報を利用して，各学生に個別の助言を与えることが可能となる．例えば，①10%の確率で起きている「何の記号も入れない」ケースをなくして 100%何らかの記号を挿入するようにすること，②挿入する記号の種類については，これまで使用していた 1 文字・4 文字スペースをタブに変更することを助言することで学生の記述スタイルを手本の記述スタイルに近づけ，可読性向上を目指させることができる．

図 9 は学生 1 の後方記述スタイル特徴と見本を比較したものである．記号「{」の後方については特に修正の指導を行う必要がないことがわかる．

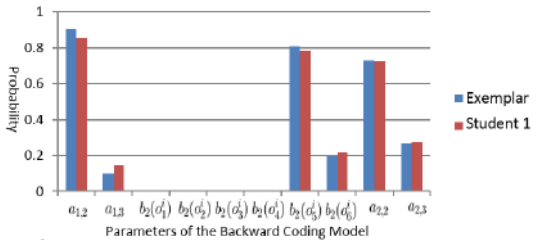


図 9 手本(青)と学生 1(赤)の後方記述スタイル比較

このように，CM アルゴリズムを用いて記述スタイル特徴を可視化することで，これまで画一的に行われてきたコーディングスタイル教育において，個人の癖に応じた指導を行うことが可能となった．

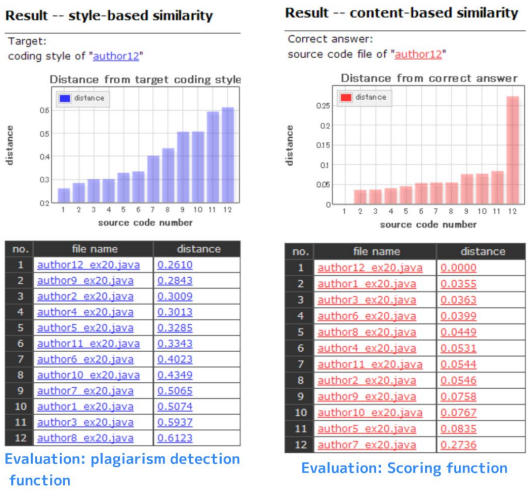
盗用の発見についても前述と同様の手続きにより行う．すなわち，ある学生の記述スタイルをモデルに学習させ，手本のスタイルとする．新たに提出された課題ソースコードから記述スタイルを抽出し，手本のスタイルと比較することにより作成者認証を行い，盗用か否か判定を行う．

(3) 統合プログラミング授業支援システムの開発

本研究において改良を行った二つの類似性検出手法は Java のツールとして開発し，データベース操作および入出力とツールの連携は MySQL と PHP により行った．

図 10 は教員用クライアント GUI における採点および盗用発見結果表示の例である．盗用発見（左）では，12 名の学生のうち 1 名の学生の記述スタイルモデルの表す特徴とそれ以外の学生のソースコードから抽出した特徴を比較した結果である．（右）は採点結果であり，正解ソースコードとの距離が点数として表示されている，いずれも特徴ベクトル間のユークリッド距離が 1～12 位まで順位付けされて表示されている．いずれも目視による確認に近い結果が得られているが，採点機能については既習得範囲以外の文法を使用した解答や冗長な構造をとるソースコードの評価が低くなることがあった．これを改善するためには，出題ごとに使用するメトリクスを定義することや出題意図に応じた設問への回答を採点項目に追加することなどが考えられる．

また，記述スタイルの部分的な類似については，可読性評価と同じくモデルのパラメータをグラフで比較することで容易にチェックが可能である．採点結果の詳細については，総合評価（FRef により算出された正解との距離：全体的な類似度）と出題意図依存の部分的な評価（メトリクスベクトルの値）個別に参照することが必要となる．



no.	file name	distance
1	author12_ex20.java	0.2610
2	author9_ex20.java	0.2843
3	author2_ex20.java	0.3009
4	author4_ex20.java	0.3013
5	author5_ex20.java	0.3285
6	author11_ex20.java	0.3343
7	author6_ex20.java	0.4023
8	author10_ex20.java	0.4349
9	author7_ex20.java	0.5065
10	author1_ex20.java	0.5074
11	author3_ex20.java	0.5937
12	author8_ex20.java	0.6123

Evaluation: plagiarism detection function

no.	file name	distance
1	author12_ex20.java	0.0000
2	author1_ex20.java	0.0355
3	author3_ex20.java	0.0363
4	author6_ex20.java	0.0399
5	author8_ex20.java	0.0449
6	author4_ex20.java	0.0531
7	author11_ex20.java	0.0544
8	author2_ex20.java	0.0546
9	author9_ex20.java	0.0758
10	author10_ex20.java	0.0767
11	author5_ex20.java	0.0835
12	author7_ex20.java	0.2736

Evaluation: Scoring function

図 10 盗用発見および採点結果の例

(4) その他（多言語化，他形式データへの適用，精神的負担の軽減効果について）

CM アルゴリズム，FRef ともに前処理の改良を行い，Java に加え C 言語で書かれたソースコードにも対応可能とした．

CM アルゴリズムの自然言語文書への適用可能性について示した．具体的には，①学会関連のメール文書②ゲームシナリオ③法律文書の記述スタイルをそれぞれモデルに学習させ，これを用いて文書データの簡単な分類・検索を行えることを明らかにした．

さらに，RFID を用いて取得した顧客動線データを CM アルゴリズムのモデルに学習させ，顧客の分類に用いる試みを行った．

このように提案手法を他形式のデータへ適用し，学習データの特徴と学習済みモデルのパラメータの関係について分析を行うことを通して，本研究における記述スタイルモデルの評価に役立てた．

また，学生を対象として実施したアンケートの結果および教員を対象とした聞き取りの結果から，CM アルゴリズムを用いた「作成者のこれまでの書き方の癖との違い」に基づく盗用発見が「他の学生との類似」に基づく盗用発見に比べ精神的に負担を与えにくいことが示唆された．

5．主な発表論文等

（研究代表者，研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 5 件）

(1) Asako Ohno, Takahiro Yamasaki, and Kin-ichiroh Tokiwa, "Application and Evaluation of Content-and-Style Based Source Code Similarity Measuring Methods

towards Programming Education Support System", ICIC Express Letters, Part B, vol.6, no.5, pp.1405-1410, 2015, **査読有**.

(2) Yoshihiro Ohata, Asako Ohno, Takahiro Yamasaki, and Kin-ichiroh Tokiwa, "An Analysis of the Effects of Customers' Migratory Behavior in the Inner Areas of the Sales Floor in a Retail Store on Their Purchase", Procedia Computer Science, vol.35, pp.1505-1512, 2014, **査読有**, DOI:10.1016/j.procs.2014.08.233.

(3) Asako Ohno, Takahiro Yamasaki, and Kin-ichiroh Tokiwa, "Modeling and Discussion towards Customer Classification According to Shoppers' In-Store Behavior", ICIC Express Letters, vol.8, no.4, pp.1013-1018, 2014, **査読有**.

(4) **稲元 勉, 大野 麻子, 村尾 元「買い物経路のベクトル化に基づく顧客判別アプローチおよび主成分回帰を用いた適用例」,『電気学会論文誌 C』**, vol.132, no.12, pp.2051-2058, 2012, **査読有**.

(5) Asako Ohno and Hajime Murao, "WM algorithm: A New Similarity Measure for Natural Language based on Author's Writing Style", ICIC Express Letters, vol.6, no.4, pp.871-877, 2012, **査読有**.

〔学会発表〕（計 4 件）

(1) Asako Ohno, Takahiro Yamasaki, and Kin-ichiroh Tokiwa, "An Improvement and Application of a Source Code Similarity Measuring Method for Programming Education Support System", The 9th International Technology, Education and Development Conference, Madrid (Spain), Mar. 3, 2015.

(2) Asako Ohno, Takahiro Yamasaki, and Kin-ichiroh Tokiwa, "An Online System for Scoring and Plagiarism Detection in University Programming Class", The 22nd International Conference on Computers in Education, Nara (Japan), Dec.1, 2014.

(3) Asako Ohno, "A Methodology to Teach Exemplary Coding Style Considering Students' Coding Style Feature Contains Fluctuations," The 43rd IEEE/ASEE Frontiers in Education Conference, pp.1908-1910, Oklahoma (U.S.), Oct. 26, 2013.

(4) **大野 麻子,「参照ベクトルを用いたソースコード類似性検出手法へのメトリクスの適用に関する検討」, 電子情報通信学会教育工学研究会(ET), 愛媛大学 (愛媛県松山市), 2013 年 3 月 29 日**.

〔図書〕（計 1 件）

(1) Asako Ohno and Hajime Murao, "Modeling of a Coding Style Using Left-to-Right HMM: an Author-Based Similarity Measure for Text Classification and Retrieval", in K. Yada ed., How Data Mining Can Revitalize Your Business, Springer, (2014 **年採録決定済, 未発行**).

6．研究組織

(1)研究代表者

大野　麻子（OHNO, Asako）
大阪産業大学・工学部電子情報通信工学科・講師
研究者番号：90550369

(2)研究分担者

（　　）

研究者番号：

(3)連携研究者

（　　）

研究者番号：