識別子中の自然言語使用に基づく プログラム実装技術解析に関する調査

山下 大貴† 竹内 和広‡

†大阪電気通信大学大学院 工学研究科 ‡大阪電気通信大学 情報通信工学部

mi14a009@oecu.jp takeuchi@isc.osakac.ac.jp

1 はじめに

プログラムに含まれる自然言語には，一般的な文や句によるコメントの記述と，自然言語の語を利用した独特の記述がなされるクラス名，メソッド名，変数名といった識別子の存在がある．本研究では，プログラム中の識別子の命名にどのような語が使われているか，そのプログラムの機能や構造とどのような関係を持つかを視座に調査する．

2 本研究の位置づけ

2.1 予備調査

識別子命名に使用される語は，デザインパターンとアルゴリズムといったプログラミング技法に関する教科書 [1][2] のプログラムの予備調査により，一般語だけでなく，専門用語や略語といった様々な語が使用されることを確認した．例えば，メソッド名に append，write，変数名に index といった一般語が使用される．また，tokenizer といった専門用語や，特殊な語として args といった語が識別子命名において使用される．

そのような識別子命名の際に使用される語は，適当に選択されるわけではない．例えば，GoF デザインパターンの 1 つである Iterator パターンを採用した教科書プログラムの識別子の語は，メソッド名に next，add や，変数名に index といった語が共起して使用されている．この実装方法に対する語の関係を命名に用いることで，プログラムの構成や振る舞いを示すことが行われる．Iterator パターンは，ある対象に関する繰り返しを与えるパターンである．

本研究では，このように，開発者があるモデルや，アルゴリズムといった実装方法に基づきプログラムを実装することを考えたときに，プログラム中の識別子命名にその実装方法に関連した語の関係が用いられることを考え，識別子命名における語の関係を調査したい．

2.2 関連研究

データマイニング手法の基盤ツールが整備されてきたことから，プログラムの識別子における語の使用に基づいた分析が行われている．例えば，プログラムに含まれるメソッドあるいはコメントに出現する語から，プログラムで使用される語間の意味関係について自動同定を行い，それらの類型を示すことで，プログラム作成上の語の関係を明らかにする研究がなされている [3][4].

Yang らの研究 [3] は，コメントやコードに含まれる語のコンテキストを比較し，コンテキスト間で同じ語やフレーズが使われていれば構造的，意味的に関連していることを仮定し，プログラムにおける語の関係を抽出する研究を行っている．この研究で使用された手法は，プログラムに含まれる全てのメソッド，コメントを抽出し，それらを語のシーケンスに分割する．このシーケンス間の類似度を計算し，与えられた閾値を越えた場合，コンテキスト間で類似しているとみなし，関連語関係のペアとして抽出している．抽出した語間の関係類型を人手により検証した上で，それらの関係付けが，一般的な語の関係と異なることを示した．

また，Howard ら [4] は，プログラム中のメソッドと，そのメソッドに対応したコメント文に出現する語を利用し，プログラムにおける動詞の類義語関係を抽出することを行っている．この研究は，メソッドに対する説明が記述されるコメントで用いられる動作動詞に絞って研究がなされており，コメントにおける自然言語記述のヒューリスティックな特性を反映させ，メソッドとコメントにおける動作動詞の関連語関係の候補を形成し，動詞の類義語関係を抽出している．また，

これらの語の関係が，WordNet[5] における一般的な類義語とは異なることを人手により検証している．

Howard らの指摘である，識別子に使われる語の用法が，一般的な用法と異なるという知見は，プログラムの分析をする上で重要な観点だと思われる．しかし，それらの研究では，分析対象とする識別子の語を選定する際に，一般的な言語処理ツールを用いているため，調査対象が一般語に限定されてしまっている．すなわち，プログラム作成に用いられる専門用語や特殊語が対象とされていない点が問題である．

プログラムで使用される語の用法が，プログラムの構成や実装技術とどう関わるかに踏み込んだ研究に，柏原らの研究 [6] がある．この研究では，相関ルールマイニングを用い，メソッドとその名前を対応付ける多くのプログラムで頻出する命名の相関ルールを抽出している．この相関ルールの中から語間の共起性に基づき，述語と項の関係といった相関ルールを抽出している．相関ルールの例として、メソッド名 iterator および hasNext を呼び出しているメソッドがあれば，そのメソッド名に動詞 find を用いやすいことを示している．

この研究は，プログラムの識別子命名における語の関係により，実装モデルや技術の特定の可能性を示しており，識別子命名における語の用法の分析が，プログラムの特徴を分析することに繋がることを示唆している．

しかし，柏原らの研究においても，分析対象の語の選定には一般的な言語処理ツールを前提とした自然言語解析ツールである OpenNLP に基づいて行われているため，プログラムで使用される専門用語や特殊語を考慮しておらず，プログラムにおける語の関係の分析として十分でない点である．このような点は，一般的な言語解析ツールを使う限りは，別の観点からプログラムのマイニングを行った際にも必ず起こりうる問題である．

このような問題を回避するためには，Yang ら，Howard らの知見であるプログラム上で一般的な語の用法とは異なる用法が存在する事実を踏まえた上で，識別子命名における語の関係を調査することが必要である．

3　調査方法

本研究では，一般語だけではなく，一般語と専門用語との関係も調査対象とする．一般語は，一般的な英語の概念辞書である WordNet を使用する．専門用語は，調査対象のプログラムの実装目的を絞り，実装目的に対応した Wikipedia 記事を使用する．また、プログラムには，様々な語の使用が予想されるが，複数プロジェクトで共通して使用される語の関係を調査する．語の関係の調査には，SVD（特異値分解）を使用する．

調査対象は，Genetic Algorithm，Neural Network，Decision Tree，HMM，CRF，SVM を実装目的としたプロジェクトを扱い，各実装目的について java で開発が行われている 6 つのプロジェクトを用いる．

識別子命名の際に使用される語に対して，一般語については WordNet により対応し，専門用語は調査対象であるプロジェクトの実装目的を説明する Wikipedia 記事により対応する．例えば，プロジェクトの実装目的が Genetic Algorithm の場合は，Wikipedia にある Genetic Algorithm の記事を説明記事として用いる．以降，専門用語を実装対象語と呼ぶ．

Wikipedia 記事には，その目的とは関係がない一般語も含んでいるため，記事上のすべての語を対象とするわけではない．そこで，テキストマイニングにおいて目的に関係する語を抽出する方法である tf-idf 法を用いて，説明記事内の語を重み付けし，この重み付け指標において，当該説明記事で特有であると判断された語を実装対象語として利用する．

具体的な調査手順は，図 1 に示すように同一目的の複数プロジェクト A，B，C，D，E，F に対して，プログラムに含まれる識別子としてクラス名，メソッド名，変数名を抽出し，キャメルケース・スネークケースに基づき，抽出した識別子を語に分割する．なお，我々は，このようなプログラム上の自然言語を抽出し，SVD 等の各種処理を行うプログラム群をツールとして整備した [7].

次に，個々のプロジェクトに対して関連語抽出手法を用いて，プログラムに含まれる語の関連語関係を同定する．手法の適用対象となるプロジェクトに対し，プロジェクト P に含まれるプログラム数が n である場合，プロジェクト P は $P = \{S_1, S_2, \cdots, S_n\}$ で表され，プロジェクト P 内のプログラム群から抽出した語の数を m としたとき，プログラム S_i は $S_i = \{w_1, w_2, \cdots, w_m\}$ の語の出現頻度ベクトルで表現される．また，プロジェクト P は行数をプログラム数，列数が語の数に相当する，$n \times m$ の非正方行列として表現される．S_i のベクトル要素 w_j はプログラム S_i 中の語 j の出現数を表している．この非正方行列に対し，手法を適用することで語の関連語集合を同定する．

個々のプロジェクトで得られた関連語集合に対し，その集合内で語のペアを生成し，これらの語のペアを

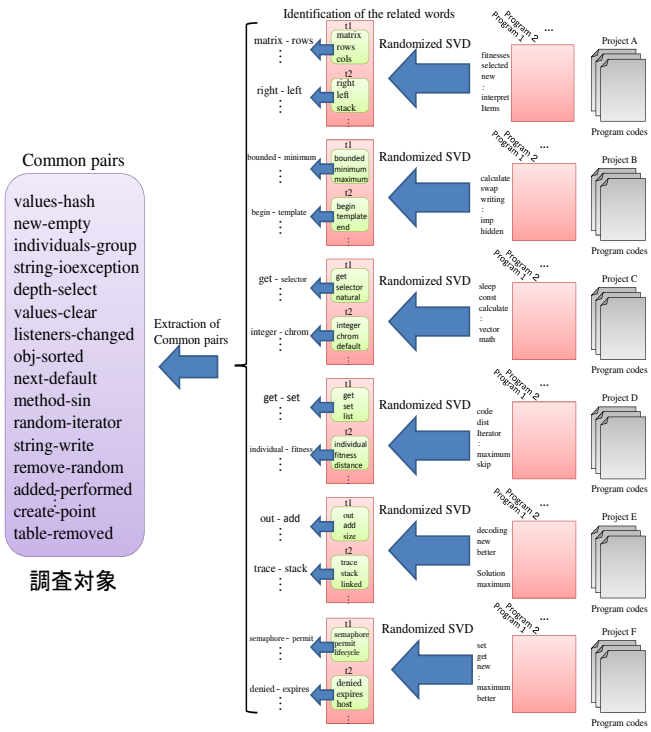

図 1: 調査手順

複数プロジェクト間で比較することにより，同一目的の複数プロジェクトで共通する語ペアを抽出する．

この語ペアに対して，ペア中の 2 つの語がプログラムの目的となっている語か否かを確認する．手順は，プロジェクトの実装目的を説明する Wikipedia 記事の語を用いて，その語ペア中の語がその実装目的に対応した実装対象語であるかを検討し，Wikipedia 記事を用いた実装対象語の対応により，プログラム中の実装目的に関連した関連語ペアの抽出を試みる．

これらの語に対し，デザインパターン・アルゴリズムの教科書プログラムから抽出した語を比較し，調査対象としているプロジェクトは，識別子命名においてどのような語を使用するのかを確認した．

語間の意味関係の発見は，対象となる行列の高次元空間をどの程度の次元まで圧縮するかが重要な問題である．そこで，本稿では行列を Randomized SVD[8] により圧縮する際に，どの程度圧縮するかを決定する基準に WordNet を用いる．

これらの語間関係を基準とすることで次元圧縮の効果を客観的に評価する．例えば，1 次元に圧縮した場合，すべての語間に意味的関係があることになり，2 次元に圧縮する場合は意味的に関連のある関連語集合が 2 つになる．これに基づき，Randomized SVD による次元圧縮の効果は意味関係を同定した集合に含まれる WordNet 上の最隣接類義語ペアの含有数で評価しつつ次元圧縮を行う．

4 調査結果・検討

GitHub から収集したプロジェクトを対象に，調査手法を適用した結果を表 1 に示す．

この表は，同一目的の複数プロジェクト間で共通する関連語ペアをプログラムの識別子命名に使用される実装対象語と一般語との関連語関係を視座に整理した．表 1 に示した各実装目的における関連語ペアのうち，一般語と特殊語に下線を示している．また，各実装目的における関連語ペアは，頻度を基準に上位から順に選択し，一部の例を載せている．

表の 2 列目と 3 列目の関連語ペアは，各実装目的に関連した実装対象語と一般語あるいは特殊語との関連語ペアになっている．先行研究では，プログラム上の識別子命名において使用される語の関係は，必ずしも WordNet の語の関係にはないことを指摘しているが，この結果は，一般語が実装対象語と関連付けられて識別子命名に用いられていることを示している．実装対象語を収集した Wikipedia 記事は，プログラム実装の際に，どのような構造により実装するのかといった情報は，多くの場合，記載されない．つまり，実装対象語と一般語の関連語関係は，この点を反映しているように思われる．

また, 表の結果は，同一目的の複数プロジェクト間で識別子命名に使用される共通の語関係の存在を示している．収集したプロジェクトのデータは，すべて開発者が異なる．すなわち，開発者が異なるにもかかわらず，複数プロジェクト間で共通の語関係が存在する理由は，開発者が当該のプログラムを実装する際に，思い浮かべるプログラム構造（構成）が存在し，その構造（構成）を持つプログラムの識別子には、共通する特定の語が使用されるからであると考えられる．例えば，表 1 の Neural Network の欄において，実装対象語 layers に next，list といった関連語関係が得られているが，それはこの layers に関するプログラムの実装に，リスト構造を用いていることが想定できる．

この考え方は，プログラムが採用したデータ構造やアルゴリズムだけではなく，デザインパターンなどのプログラムの構成に関しても同様であると考えられる．すなわち，このような多くのプロジェクトで共通して使用される語の用法は，実装において開発者が持つ経験に基づき，繰り返し採用されるある種の知識であると考えている．

表 1: 各実装目的から抽出した関連語ペア

実装目的	実装対象語 - 実装対象語	実装対象語 - 教科書語		Others 関連語ペアの 1 語のみ実装対象語	
CRF	sequence-feature	sequence-print	sequence-iter	previous-tag	model-eval
	model-sequence	feature-get	feature-instance	labels-path	transition-identifiers
	pattern-training	feature-hash	sequence-string	sequence-seq	labels-score
	parameters-likelihood	feature-iterator	sequence-iterator	sequence-yprev	gradient-pos
Decision Tree	class-attribute	class-set	best-split	best-builder	class-true
	gain-best	attribute-equals	attribute-count	class-entry	attribute-counter
	information-gain	best-list	attribute-exception	categorical-numeric	categorical-ignored
	greater-best	other-merge	categorical-array	categorical-splits	nodes-submit
Genetic Algorithm	gene-chromosome	chromosome-get	chromosome-integer	fitness-mutate	component-condition
	crossover-chromosome	chromosome-next	best-size	mutation-arrays	generation-selector
	fitness-chromosome	fitness-compare	fitness-get	component-event	crossover-evolve
	genetic-chromosome	crossover-get	chromosome-value	chromosome-calculate	mutation-comparator
HMM	backward-forward	contains-put	states-sum	likelihood -alpha	parts-pos
	values-parameters	contains-map	next-observation	parameters-entry	viterbi-prev
	contains-word	word-hash	words-integer	replace-pos	pattern-matches
	parts-replace	states-add	states-update	parameters-calculate	probability-rand
Neural Network	learning-rate	activation-set	neural-clone	linear-idx	process-failures
	forward-layers	layers-list	layers-next	layer-build	threshold-thresh
	layer-hidden	functions-init	layers-exception	connections-conns	threshold-null
	functions-activation	layers-index	element-read	network-build	delta-diff
SVM	linear-kernel	epsilon-error	solution-sum	classifier-step	model-leave
	cross-validation	class-count	alpha-get	model-predict	epsilon-param
	linear-model	optimal-update	classifier-begin	model-estimates	multiclass-decision
	classifier-model	classifier-new	optimal-select	class-estimates	optimal-bound

5 おわりに

本稿ではプログラムの識別子命名における自然言語の使用が，プログラム構成といった実装技術にどのように関連するのかを調査した．本研究で得られた知見は，あるプログラムに対して、どのようなモデル・アルゴリズムに基づいて，作成されているのかを識別子命名において用いられる自然言語の用法により、特定できる可能性を秘めている．今後は，この知見に基づき，与えられたプログラムのアルゴリズム・モデルの構成を特定する研究をしていきたい．

謝辞

この研究の一部は科研費 (基盤 (C) 課題番号:24501158) の支援を受けている．

参考文献

[1] 奥村晴彦，首藤一幸，杉浦方紀，土村展之，津留和生，細田隆之，松井吉光，光成滋生． Java によるアルゴリズム事典．技術評論社 (2003)

[2] 結城 浩．増補改訂版 Java 言語で学ぶデザインパターン入門．ソフトバングパブリッシング (2004)

[3] J. Yang and L. Tan. Inferring Semantically Related Words from Software Context. MSR 2012 , pp.161-170 (2012)

[4] M. J. Howard, S. Gupta, L. Pollock, and K. Vijay-Shanker. Automatically Mining Software-Based, Semantically-Similar Words from Comment-Code Mappings. MSR 2013, pp.377-386 (2013)

[5] WordNet, http://wordnet.princeton.edu.

[6] 柏原由紀, 鬼塚勇弥, 石尾隆, 早瀬康裕, 山本哲男, 井上克郎．相関ルールマイニングを用いたメソッドの命名方法の分析, 日本ソフトウェア科学会, ソフトウェア工学の基礎 XX, pp.25-34 (2013)

[7] 山下大貴，竹内和広，WordNet および Wikipedia と連携するソースコード上の関連語マイニングツール．ソフトウェアエンジニアリングシンポジウム 2014，pp.194-195 (2014)

[8] N. Halko, P. G. Martinsson, and J. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM REVIEW Vol.53, No.2, pp.217-288 (2011)