

テクニカルノート

Java メソッドの動作を表現する動詞の自動推薦手法の評価

柏原 由紀^{1,a)} 石尾 隆^{1,b)} 井上 克郎^{1,c)}

受付日 2015年3月23日, 採録日 2015年5月9日

概要: Java のメソッドの命名を支援するために, メソッドの動作を適切に表現するような動詞の候補を自動的に推薦する手法が提案されている. しかし, 既存研究の評価は推薦結果の正確さだけを基準に行われており, 実際に開発者が推薦結果を見て適切な動詞を選択できるかどうかは評価されていなかった. 本研究では, メソッドの内容と動詞のリストを被験者に提示し, メソッドの動作を表現する動詞を解答してもらう実験を行った. その結果, 開発者が提示された動詞の候補から適切な動詞を選択できることを確認した.

キーワード: 識別子, メソッドの命名, Java

Evaluation of Recommendation of Action Verbs for Java Method

YUKI KASHIWABARA^{1,a)} TAKASHI ISHIO^{1,b)} KATSURO INOUE^{1,c)}

Received: March 23, 2015, Accepted: May 9, 2015

Abstract: To support method naming for Java, there exists recommendation systems that automatically extract candidates of verbs representing method actions. Those systems are evaluated by the correctness of recommended verbs; there still be a question whether developers can recognize an appropriate verb in a recommendation list or not. In this research, we provided a method content and a list of verbs to human subjects, and asked them to describe the behavior of the method by a single verb. The result shows that developers can recognize appropriate verbs in a recommendation list.

Keywords: identifiers, method naming, Java

1. はじめに

ソースコードにおける識別子は, 開発者にとって重要な情報源である. 保守作業において作業対象を特定する手掛りの1つであるといわれている [1] ほか, 識別子はその役割を適切に表現していないようなソースコードの理解には時間がかかることが知られている [2].

オブジェクト指向プログラミング, たとえば Java においては, メソッド名は一般に, その動作と対象を表すように動詞や名詞などを組み合わせて命名される [3]. フィールドに対する読み書きであれば get や set を使用するというように, メソッドが行う処理と動詞との間には関連性が

あり [4], 既存の使用法と大きく異なるような動詞の使用は命名誤り (Naming Bug) であると主張されている [5].

開発者が適切な動詞を選択してメソッドの命名を行えるようにするために, 我々の研究グループでは, メソッド名に用いる動詞の候補リストを提示する手法を提案した [6]. この手法は, 相関ルールマイニング [7] を用いて既存のソースコードからメソッド本体の内容とメソッド名に含まれる動詞の関係を抽出しておき, 開発者から与えられたメソッドの内容に対して使用される可能性が高い動詞をランキング形式で推薦する. 同種の研究としては, Yu ら [8] によるサポートベクターマシン (SVM) を使用した動詞の推薦手法が存在している.

動詞を推薦する手法の評価は, 命名誤りの数が少ない [5] ことから, 既存のソースコードに付与されたメソッド名や, ソースコードの改版履歴から抽出した名前の変更を正解と見なした推薦の正確さによって行われている [6]. その一

¹ 大阪大学
Osaka University, Suita, Osaka 565-0871, Japan
a) k-yuki@ist.osaka-u.ac.jp
b) ishio@ist.osaka-u.ac.jp
c) inoue@ist.osaka-u.ac.jp

方で、開発者が正解を提示されたとしても、それを実際に活用できるかどうかは確認されていなかった。

本研究では、メソッドの内容と動詞の候補リストを与えられたときに、被験者がそのメソッドに適切な動詞を選択できるのかを評価する実験を行い、候補リストの情報と被験者の正答数との関係を調査した。

以降、2章では評価実験について説明する。3章で妥当性の脅威について議論した後、4章でまとめと今後の課題について述べる。

2. 評価実験

本研究では、動詞の候補が推薦されたとき開発者が適切な動詞を選択できるかを評価する。実験の設計方針は、以下の2点である。

- 動詞の推薦あり、推薦なしの2つの状態での差を調査する。命名誤りと検出されるメソッドの数は少ない[5]ことから、開発者は既存手法による支援がなくても、多くの場合は適切な動詞を選択できると考えられるためである。
- 正解が含まれないリストを与えられた場合の影響を調査する。これは、既存手法[6]、[8]がつねに正解を提供できるわけではないためである。

この方針に従って、以下の3つのリサーチクエスチョンを設定した。

RQ1: 正解を含む動詞の候補リストだけが与えられるとき、開発者は適切なメソッド名の動詞を選択できるか。

RQ2: 正解を含まない動詞の候補リストが与えられる場合があるとき、開発者は適切なメソッド名の動詞を選択できるか。

RQ3: 候補リストがあるとき、開発者が動詞を選ぶのにかかる時間は変化するか。

2.1 実験方法

本実験では、被験者に、名前が付けられていない状態のメソッドのソースコードを与え、その中身を読解し、動作を適切に表現すると思われる動詞を解答するという課題を実施してもらった。実験対象として選択したメソッドの「適切な動詞」は、文献[6]、[8]と同様に、そのメソッドに現在使われている動詞であるとした。つまり、メソッドの名前が隠された状態から、その名前を解答できれば正解とした。

2.1.1 課題内容

課題の作成にあたっては、我々の既存手法[6]を拡張したものを使用した。既存手法ではソースコードからの学習を行っていたが、バイトコードからの学習を可能とし、Qualitas Corpas^{*1}バージョン20130901に収録された112

個のオープンソースソフトウェアのバイナリファイル集合から相関ルールを抽出して使用した。課題の作成元とするソフトウェアは、学習セットに含まれていないソフトウェアから、BlueJ, OrderPortal, Saxon-HE, NeoDatisの4つを選択した。これらは、Hayaseら[9]によるとそれぞれデスクトップアプリケーション、Webアプリケーション、XML処理、データベースという異なるドメインに所属している。

各ソフトウェアから3つずつメソッドを選択し、12の課題とした。我々の既存手法はランキング形式で動詞を列挙することから、上位5件だけを開発者に提示するという使用法を仮定し、既存手法を実際に適用したときに正解の動詞が1位に出ていたメソッド、3位に出ていたメソッド、5位に出ていたメソッド、5位以内に含まなかったメソッドをそれぞれ3つずつとなるように選んだ。選定には、以下の基準を用いた。

- 動詞の選択が目的語に影響されないように、メソッド名が動詞1単語からなるもの。
- 被験者の負担が大きくなりすぎないように、メソッド内部の命令が5行以上かつ15行以下であるもの。
- 継承関係による影響のない（オーバーライドではない）もの。

各プロジェクトから1課題ずつ、正解の動詞が提示される位置がそれぞれ異なるように選び出したものを1セットとし、4課題ずつの3セットに分割した。正解が出現した順位によってメソッドを選択しているが、動詞のリストはいずれも既存手法で実際に提示されるものをそのまま使用している。

課題1つにつき被験者に与えられるメソッドに関する情報は、メソッド名を空欄としたメソッド本体に加えて、メソッドが定義されているクラス名、その親クラス名およびインタフェース名、メソッド本体でアクセスしているフィールドの定義である。パッケージのインポート文や、メソッド本体でアクセスされていないフィールドの定義、課題となるメソッド以外のメソッド、課題となるメソッド内外に記述されているコメントは削除した。これは、課題となるメソッド本体の動作を表す動詞を推測することを目的としており、使用法からの判断を避けるためである。また、メソッド内のコメントを削除するのは、課題ごとのコメントの有無が結果に影響を与える可能性を排除するためである。作成した課題の1つを図1に示す。

2.1.2 実施手順

被験者は、大阪大学基礎工学部情報科学科の学部4年生3人と、大阪大学大学院情報科学研究科コンピュータサイエンス専攻博士前期課程の大学院生9人の計12人である。いずれもソフトウェア工学講座に所属しており、Javaプログラミングについて最短1年、最長5年の経験がある。

1人の被験者には2セットの課題を均等になるように割

*1 <http://qualitascorpus.com/>

表 1 正解の動詞が提示されている順位と被験者が選択した動詞の順位の関係
Table 1 Relationship between ranks of correct verbs and selected verbs.

リスト内での 正解の順位	候補リストありでの解答のリスト内での順位						正答数	
	1位	2位	3位	4位	5位	リスト外	リストあり	リストなし
1位	11	0	1	0	0	0	11 / 12	10 / 12
3位	0	2	8	0	1	1	8 / 12	4 / 12
5位	0	1	4	0	7	0	7 / 12	4 / 12
リスト外	2	0	1	0	1	8	2 / 12	7 / 12

```
public class AesMd5Cypher implements IO {
    public void _____(byte b) throws IOException {
        try {
            byte[] bytes = {b};
            write(bytes, 0, 1);
        } catch (IOException e) {
            DLogger.error(e.getMessage() + " - " +
                OdbThread.currentThreadName());
            throw e;
        }
    }
}
```

動詞の候補: write, append, add, load, create

図 1 write メソッドから作成された課題の例

Fig. 1 An example task made of a write method.

り当てる。被験者はそのうち1セットを動詞の候補リストがある状態で、もう片方を動詞の候補リストがない状態で解く。被験者12人のうち、6人は先に動詞の候補リストがある1セットを、残り6人は先に動詞の候補リストがない1セットを解いてもらった。1セット内での課題の順番は、同一順序で課題を割り当てられている被験者はいないように割り当てた。1人の被験者が受け取る8つの課題の状態をまとめると、以下のとおりである。

- 動詞の候補リストが与えられた4課題。正解の動詞を1位、3位、5位に提示するリストと、正解を含まないリストを1つずつ用いる。
- 動詞の候補リストが与えられない4課題。

実験は、被験者が1つの部屋に集合した状態で、紙を用いて実施した。課題の説明文1ページと、8つの課題各1ページ、Javaの開発経験などを問うアンケート1ページからなる問題用紙を渡し、5分間の説明ののち、各課題に順に解答してもらった。解答方法は動詞を解答欄に記述する形であり、課題に解答して次の課題に進んだ後の解答の修正は許さなかった。時間制限は特に定めなかったが、解答の様子はビデオカメラで撮影し、問題用紙のページをめくる間隔を各課題に要した時間として計測した。

2.2 実験結果

正解が提示されている順位が異なる候補リストが与えられたとき、被験者がどの順位のもの、あるいはリスト外の動詞を解答したかを集計したものを表1に示す。行は正解の

表 2 候補リスト内に正解が含まれている場合の解答の正確さ
Table 2 Correctness of answers for recommendation lists including correct verbs.

	正解	不正解	合計
候補リストあり	26	10	36
候補リストなし	18	18	36
合計	44	28	72

動詞が提示されている順位、列は被験者が選んだ動詞が提示されている順位を表しており、たとえば、1位に正解が提示されているリストに対しては、1位に提示されている動詞が解答されたのが11件、3位に提示されている動詞が解答されたのが1件であった。また、表の右端の列に被験者の正答数を示しており、これらの動詞に対してはリストありでの12件の解答のうち11件、リストなしでの12件の解答のうち10件が正解であったことを示している。候補リストの3位、5位に正解が提示されている場合は、リストを用いた場合の正答数がリストを用いない場合を上回った。これらのメソッドは、被験者にとって判断が難しく、動詞の推薦が有効に働いたと考えられる。候補リストに正解がなかった場合は、12件中8件でリスト外の動詞が解答されているが、正しい動詞が解答されたのは2件だけである。リストに適切な動詞がないことを認識するのは容易であるが、それが判断に何らかの影響を与えている可能性がある。

2.2.1 RQ1: 正解を含むリストだけが与えられる場合

表1の結果から候補リストに正解の動詞が含まれている場合の正答数を集計した結果を表2に示す。この表に対してフィッシャーの正確確率検定を適用し、候補リストの有無が正答数に有意な差をもたらしているかどうかを調べた。その結果、 $p = 0.045$ となり、有意水準 $\alpha = 0.05$ で有意な差があったといえる。このことから、動詞の推薦が正しく行われていれば、開発者が正しい判断を行える可能性が向上することが分かる。

2.2.2 RQ2: 正解を含まないリストが与えられる場合

解答された動詞の正答数を、正解を含まない候補リストの結果まで含めて集計したものを表3に示す。候補リストがある場合のほうが被験者が正解を選んでいる数が多いが、正答数に差がないという帰無仮説に対してカイ二乗検定を行うと $p = 0.54$ となり、有意な差ではなかった。メ

表 3 すべての候補リストを含めた場合の解答の正確さ

Table 3 Correctness of answers for all the recommendation lists.

	正解	不正解	合計
候補リストあり	28	20	48
候補リストなし	25	23	48
合計	48	48	96

表 4 課題にかかった時間の統計的な指標値

Table 4 Statistics of time spent for tasks.

	平均値	中央値	標準偏差	変動係数
候補リストあり	190 秒	187 秒	72 秒	0.38
候補リストなし	214 秒	185 秒	112 秒	0.53

ソッドに対して適切な動詞を選択できるようにするには、75%のメソッドに対して正解の動詞を5位以内に出せるという正確さでは不十分であったと考えられる。

2.3 RQ3：開発者が動詞を選ぶのにかかる時間

候補リストの提示がある場合とない場合で、解答にかかる時間に差があるのかを調査した。候補リストがある場合とない場合の解答時間の統計的な指標値を表4に示す。全体として、解答にそれほど多くの時間は費やされていなかった。候補リストがあるほうが解答時間のばらつきが少ないことから、被験者が動詞の選択に悩んだときに、適切な動詞の候補を検討する助けになったと考えられる。

3. 妥当性への脅威

本研究では、実験の設計上、「適切な動詞」の正解をただ1つと仮定した。既存研究でも同様の仮定が用いられているが、類義語など、明確な誤りとはいえない動詞が複数存在することから、既存手法の効果を実際よりも低く見積もっている可能性がある。

現在の動詞が与えられておらず動詞を推測するという課題は、メソッド抽出リファクタリングのようにメソッドに含めたい処理内容が先に定まっている状態に相当する。メソッドの名前変更リファクタリングでは、適切な候補がないときに現状を維持するという選択肢をとることも可能であることから、本実験における「不正解」がただちに命名誤りにつながるとは限らない。また、本研究における実験では被験者が知らないメソッドを課題としたが、実際の開発環境では開発者は周辺のメソッドとの呼び出し関係やそのメソッドの役割について知っている可能性が高い。より現実のソフトウェア開発に近い形での実験は、今後の課題である。

本研究では、候補リストとして上位5件という想定をおき、順位づけされたリストを用いた。上位10件、20件というように、提示する個数や提示方法による影響は、本研

*2 <http://www.gpower.hhu.de/>

究では評価していない。

本実験では被験者の所属がソフトウェア工学講座に限られている。また、RQ1の検定では、カイ二乗検定に必要な標本数 ($G^*Power3^{*2}$ で 2×2 の分割表において効果量 0.3, 有意水準 0.05, 検定力 0.8 の場合を計算すると 88 標本) に満たなかったためフィッシャーの正確確率検定を用いた。被験者を増やして検定を行うと結果が変わるかもしれない。

4. まとめ

本研究では、動詞の候補リストが推薦されたときに開発者が適切な動詞を選択できるかどうかについて調査した。その結果、開発者は候補リストに正解が提示されているとき、候補リストがないときよりも正解の動詞を選択できるが、候補リストに正解が含まれない場合まで考慮すると有意差はなかった。このことから、動詞の候補を推薦する手法自体は有効に働くが、その正確さが非常に重要であるといえる。

今後の課題として、正確に動詞の候補を推薦できる手法の構築と、より現実のソフトウェア開発に近い作業、特にメソッドの名前変更作業を課題とし、複数の正解がありうることを考慮しての被験者実験の実施があげられる。

謝辞 本研究は、科研費 (課題番号 2522003, 26280021) の助成を得たものである。

参考文献

- [1] De Lucia, A., Di Penta, M., Oliveto, R., Panichella, A. and Panichella, S.: Using IR Methods for Labeling Source Code Artifacts: Is It Worthwhile?, *Proc. ICPC*, pp.193–202 (2012).
- [2] Lawrie, D., Morrell, C., Feild, H. and Binkley, D.: What's in a Name? A Study of Identifiers, *Proc. ICPC*, pp.3–12 (2006).
- [3] McConnell, S.: *Code Complete, Second Edition*, Microsoft Press (2004).
- [4] Høst, E.W. and Østvold, B.M.: The Programmer's Lexicon, Volume I: The Verbs, *Proc. SCAM*, pp.193–202 (2007).
- [5] Høst, E.W. and Østvold, B.M.: Debugging Method Names, *Proc. ECOOP*, pp.294–317 (2009).
- [6] Kashiwabara, Y., Ishio, T., Hata, H. and Inoue, K.: Method Verb Recommendation Using Association Rule Mining in a Set of Existing Projects, *IEICE Trans. Information and Systems*, Vol.E98-D, No.3, pp.627–636 (2015).
- [7] Agrawal, R., Imielinski, T. and Swami, A.: Mining Association Rules between Sets of Items in Large Databases, *Proc. SIGMOD*, pp.207–216 (1993).
- [8] Yu, S., Zhang, R. and Guan, J.: Properly and Automatically Naming Java Methods: A Machine Learning Based Approach, *Advanced Data Mining and Applications, Lecture Notes in Artificial Intelligence*, Vol.7713, pp.235–246, Springer Berlin Heidelberg (2012).
- [9] Hayase, Y., Kashima, Y., Manabe, Y. and Inoue, K.: Building Domain Specific Dictionaries of Verb-Object Re-

lation from Source Code, *Proc. CSMR*, pp.93–100 (2011).



柏原 由紀

2012年大阪大学基礎工学部情報科学科卒業。2014年同大学大学院情報科学研究科博士前期課程修了。同年株式会社エヌ・ティ・ティ・データ入社。在学中，ソフトウェアの可読性とリファクタリングに関する研究に従事。



石尾 隆 (正会員)

2003年大阪大学大学院基礎工学研究科博士前期課程修了。2006年同大学院情報科学研究科博士後期課程修了。同年日本学術振興会特別研究員(PD)。2007年大阪大学大学院情報科学研究科コンピュータサイエンス専攻助教。

博士(情報科学)。プログラム解析，プログラム理解に関する研究に従事。



井上 克郎 (フェロー)

1956年生。1979年大阪大学基礎工学部情報工学科卒業。1984年同大学大学院博士課程修了。同年同大学基礎工学部助手。1984～1986年ハワイ大学マノア校情報工学科助教授。1989年大阪大学基礎工学部講師。1991年同

助教授。1995年同教授。2002年大阪大学大学院情報科学研究科教授。2012年同大学院情報科学研究科・研究科長。工学博士。ソフトウェア工学，特に，ソフトウェア開発手法，プログラム解析，再利用技術の研究に従事。