

# プログラム要素間の関係のグラフ埋め込みに基づく クラス名の調査

栗本真太郎<sup>†</sup> 早瀬康裕<sup>††</sup> 米内裕史<sup>†††</sup> 伊藤寛祥<sup>†††</sup> 北川博之<sup>‡</sup>

<sup>†</sup>筑波大学情報学群情報科学類 <sup>††</sup>筑波大学システム情報系

<sup>†††</sup>筑波大学システム情報工学研究科 <sup>‡</sup>筑波大学計算科学研究センター

## 1 背景

ソフトウェア開発では、プログラムを構成するクラスやメソッド、フィールドといったプログラム要素(以下、単に要素)の適切な命名が重要である[1]。なぜなら、開発者は識別子名から動作や役割を推測してプログラム理解を進める[2]ためである。しかし、適切な命名はソフトウェア開発の経験に加え、そのソフトウェアが対象としているドメインの知識を必要とする困難な作業である。そこで、開発者が識別子を命名するタスクを支援するため、プログラムの内容から識別子名を推薦する手法が提案されている。

様々な要素のうち、メソッドの識別子名推薦に取り組んだ研究はいくつか存在する。Allamanisら[3]は、Javaプログラムの抽象構文木を用いてSkip-gram[4]により分散表現(数十から数百次元程度の実数ベクトル)を生成し、それを用いてメソッド名推薦を行った。米内ら[5]は、Javaプログラムにおけるメソッドをノード、メソッド呼び出し関係をエッジとするグラフを構築し、そのグラフから生成した分散表現を用いてメソッド名を推薦した。Allamanisらの手法[3]と比べ、米内らの手法[5]はメソッドを定義した時点で分散表現を生成できる利点を持つ。

本稿では、米内らの手法[5]を拡張した手法により生成した分散表現がクラス名推薦に有効であるかの確認を目的として、クラスの分散表現とクラス名との関係を調査する。クラスの動作や性質はクラスのみ関係だけでは表すことができないと考えられるため、メ

ソッドがフィールドにアクセスしている関係といった、複数の要素及び関係を用いる拡張を行う。

## 2 調査方法

調査は3つのステップ、(1)関係の抽出(2)分散表現の生成(3)名前との対応分析で構成する。概要を図1に示す。分散表現が意味を的確に表すことが明らかになれば、分散表現の計算によりクラスの内容から適切な識別子名を算出可能であるため、分散表現がクラス名推薦に有効である確認ができたといえる。

以下、各ステップについて順を追って記述する。

### 2.1 要素間の関係の抽出とグラフの構築

グラフ埋め込みの計算に用いるデータの構築を目的として、Javaプログラムからグラフを構築する。このグラフは、次のステップにおいて要素の持つ関係が分散表現へ反映されるように、プログラムに現れる要素が他の要素とどのような関係を持っているかという情報を保持する設計とする。

上記の意図を実現するため、「クラス、メソッド、フィールドをノードとし、クラスとクラスの継承関係、クラスとメソッドの所有関係、クラスとフィールドの所有関係、メソッドとメソッドの呼び出し関係、メソッドがフィールドにアクセスする関係、メソッドと戻り値の型クラスとの関係、フィールドと型クラスの間をエッジとする」有向グラフを定義し、ノード間にエッジがあることを辞書のデータ構造により実装する。例えば、メソッドMの戻り値の型がクラスCである場合ノードMからノードCへ向かうエッジが張られており、メソッドと戻り値の型クラスとの関係を保持する辞書にキーをMとしてバリューCを追加する。また、いずれの要素についても名前解決を行っており、別プロジェクトの同一識別子名を持つ要素はグラフ上では別ノードとして扱う。

### An Investigation of Class Names Based on Graph Embedding of Relationships between Program Elements

Shintaro KURIMOTO<sup>†</sup>, Yasuhiro HAYASE<sup>†</sup>, Hiroshi YONAI<sup>†</sup>, Hiroyoshi ITO<sup>†</sup>, Hiroyuki KITAGAWA<sup>‡</sup>

<sup>†</sup>College of Information Science, University of Tsukuba

<sup>††</sup>Faculty of Engineering, Information and Systems, University of Tsukuba

<sup>†††</sup>Graduate School of Systems and Information Engineering, University of Tsukuba

<sup>‡</sup>Center for Computational Sciences, University of Tsukuba

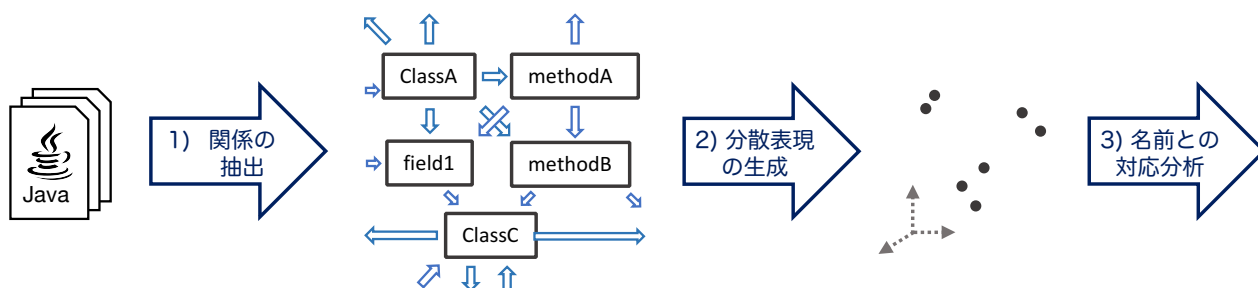


図 1: 調査の概要図

## 2.2 要素に対応する分散表現の生成

クラスの性質を保持した分散表現の獲得を目的として、要素間の関係グラフを用いることでクラス、メソッド、フィールドそれぞれについて分散表現を生成し、それらを統合することで要素間の関係を反映したクラスの分散表現を生成する。

クラスの分散表現は、継承クラスの分散表現、所有メソッド群の分散表現の平均、所有フィールド群の分散表現の平均、以上3つに重みをつけて和をとった分散表現となるように学習を行う。これはクラスと、継承クラス、所有メソッド群及び所有フィールド群との間に存在する意味的な関連を表現できることを期待したものである。

同様に、メソッドの分散表現は、呼び出しメソッド群の分散表現の平均、アクセスしているフィールド群の分散表現の平均、戻り値の型クラスの分散表現、以上3つに重みをつけて和をとった分散表現となるように学習を行い、フィールドの分散表現は、型クラスの分散表現となるように学習を行う。

上述の手順により、クラス、メソッド、フィールドの分散表現を定める。最終的なあるクラスの分散表現は、継承クラスの学習済み分散表現、所有メソッド群の学習済み分散表現の平均、所有フィールド群の学習済み分散表現の平均、以上3つに重みをつけて和をとった分散表現とする。

## 2.3 分散表現の抽出と識別子名との対応分析

これまでのステップで生成した分散表現によりクラス名推薦が期待できるか明らかにすることを目的として、生成したクラスの分散表現同士のベクトル空間上での距離と、対応するクラス名同士の意味的な近さがどのように対応しているか調べる。意味的に近いクラスの分散表現はベクトル空間上でも近くに存在し、意味的に離れているクラスの分散表現はベクトル空間上

でも離れているというように、クラスの性質の類似性を分散表現が保持していれば、クラス名推薦への応用が期待できるものとする。

まず、分散表現に対しクラスタリングを行い、クラスターを分析する。相対的に近い要素同士が近く遠い要素同士が遠くなるため、各クラスターがどのような要素によって構成されているか調べることによって、どのような場合に分散表現がベクトル空間上で近くなるまたは遠くなるかを明らかにする。

次に、クラス名の類似性とクラスの分散表現の距離との関係について調査する。部分一致する単語同士や類義語の関係にある単語同士は意味的に近いといえるため、識別子名が部分一致するクラス同士や識別子名に含まれる単語が類義語の関係にあるクラス同士のベクトル空間上での距離を調べることにより、クラス名に類似性がある場合に分散表現にその類似性が距離として反映されているかを明らかにする。

## 3 まとめ

本稿では、要素間の関係を反映したクラスの分散表現とクラス名との関係について調査を行なった。今後は、調査を元の実験を行い評価し、クラス名推薦手法を考案する予定である。

## 参考文献

- [1] A. Von Mayrhauser and A. Marie Vans. Program comprehension during software maintenance and evolution. *Computer*, No. 8, pp. 44–55, 1995.
- [2] A. De Lucia et al. Using ir methods for labeling source code artifacts: Is it worthwhile? In *Proceedings of ICPC*, pp. 193–202. IEEE, 2012.
- [3] M. Allamanis et al. Suggesting accurate method and class names. In *Proceedings of FSE*, pp. 38–49. ACM, 2015.
- [4] T. Mikolov et al. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*, pp. 3111–3119, 2013.
- [5] 米内ら. メソッド呼び出し関係に基づくメソッド名の予測. SES 2018 論文集, Vol. 2018, pp. 34–43, 2018.