2J-02

コールグラフに基づくメソッド名の分散表現の獲得

米内裕史§　早瀬康裕†　北川博之‡

§筑波大学情報学群情報科学類　†筑波大学システム情報系
‡筑波大学計算科学研究センター

1　背景と問題

プログラムの可読性の重要な要件の一つとして，識別子がその役割や動作を示した名前を持つことが挙げられる．これは，開発者が識別子名の意味から指し示す要素の役割を推察する [1] ためである．しかし，適切な命名を行うには，そのプログラムが扱うドメインに関する語彙の知識等が必要であり，経験が浅い開発者が識別子名を考えるのは難しいという問題がある．そこで，プログラムの内容から，そのような適切な識別子名を開発者に推薦する手法が提案されている．

識別子名の評価や推薦を行う先行研究では，識別子名が指し示すボディとの関係から導き出したルールを用いていた．しかし既存の手法は，識別子名中に登場する名詞の多様性の高さから，名詞部分に関する推薦の結果の精度が低いという課題がある．

2　提案手法

2.1　概要

上記の多様性による問題を克服し，メソッドの識別子名を高い精度で推薦するため，我々はメソッド名に関する分散表現を生成し推薦に利用することを提案する．自然言語処理の分野では単語に関する推薦を行う手法として，単語に関する分散表現を生成し推薦に利用する Word2Vec[2] などが提案されている．同様にメソッド名に対して，分散表現の位置関係によってメソッド名の意味の関連を表現するような分散表現を生成できれば，その位置関係を利用することで，メソッド名の推薦に利用できると考えられる．図 1 にそのような，獲得を期待する分散表現の例を示す．Word2Vec では分散表現を生成するにあたり，注目している単語の前後で共起する単語をコンテキストとして，それを元に

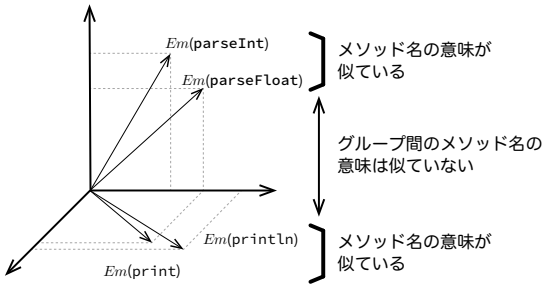

図 1:　獲得を期待する分散表現の例 (「”parseInt” と”parseFloat”，”print” と”println” にはそれぞれ意味の関連が強いが，この 2 つのグループ間では意味の関連が弱い」ことを表現)

分散表現を生成していた．我々はメソッド名に対応する分散表現を生成するにあたり，注目しているメソッドがそのボディで呼び出している他のメソッド群をコンテキストとして分散表現を生成する．これは，メソッド名はそのメソッドの動作や役割を適切に示したものであるべきだが，メソッドの動作や役割を表現する上で，ボディで呼び出しているメソッドは重要な要素であるからである．

2.2　分散表現の獲得方法

分散表現の獲得方法について，図 2 に概要図を示す．分散表現を生成するにあたり，多様なメソッドの命名と内部で呼び出しているメソッドの事例を収集するため，複数の著名なオープンソースソフトウェアのソースコードを収集した．それらのソースコードからメソッド間の呼び出し関係を得るため，宣言されているメソッドとそのボディで呼び出されているメソッドの組を抽出し，各メソッドをノードとして呼び出し元のメソッドから呼び出し先のメソッドに有向エッジが張られるようなコールグラフを生成する．

生成したコールグラフの各ノードが示すメソッド名に対応する分散表現が，そのノードを始点とするエッジの先のノードが示すメソッド群，即ちボディで呼び出

Embeddings of Method Names Based on a Call Graph
Hiroshi YONAI§, Yasuhiro HAYASE†and
Hiroyuki KITAGAWA‡
§College of Information Science, University of Tsukuba
†Faculty of Engineering, Information and Systems, University of Tsukuba
‡Center for Computational Sciences, University of Tsukuba

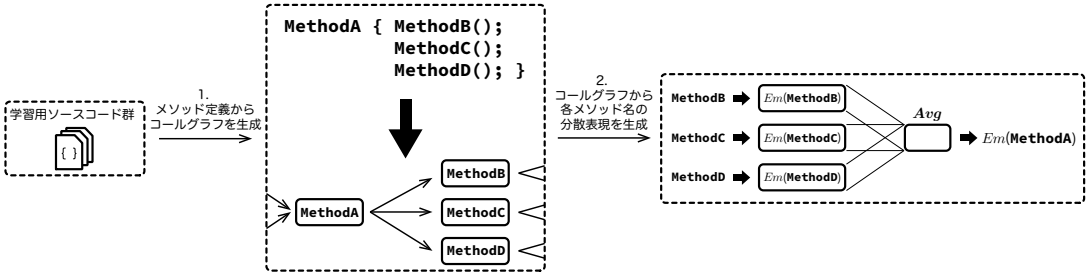

図 2: 提案手法における分散表現の獲得方法の概要図

しているメソッドの分散表現の平均となるように，各メソッドに対応する分散表現を定める．これにより，メソッドの内部の動作，即ち呼び出しているメソッドによって呼び出し元のメソッドの分散表現を決定することを表現する．

なお，コールグラフの構築にあたり，ソースコード中のメソッドの名前解決ができない場合があり，実際には完全なコールグラフを生成するのは困難である．対策として，同名のメソッド定義が複数あった場合にコールグラフ上でそれらをクラスやパッケージにより区別せずひとつのノードに統合するという処理を行う．このとき，統合されるそれぞれのメソッドのボディで呼び出しているメソッド群はすべて統合後のひとつのメソッドが呼び出しているものとする．

2.3 分散表現の学習方法

分散表現を学習によって獲得するにあたり，以下に示す通り関数 E_1 を定めた．

$$E_1 = \sum_{k=1}^{n} |Em(m_k) - avg(Em(t_{k1}) + Em(t_{k2}) + \dots)|^2$$

上記の式において，n はコールグラフ上のメソッドの数，$Em(m_k)$ は k 番目のメソッドに関する分散表現，$Em(t_{ki})$ は k 番目のメソッドのボディにおいて i 番目に呼び出されているメソッドの分散表現を示す．関数 E_1 は各メソッド名に対応する分散表現が，内部で呼び出しているメソッド群の分散表現の平均に近いほど値が小さくなるため，この関数の値を最小化するように分散表現の値を調整する．

また，上記の関数のみでは，すべてのメソッドの分散表現が零ベクトルへ向かうように分散表現が調整されてしまう可能性があるが，零ベクトルではメソッド名の意味を表現できない．対策として，以下に示すとおり関数 E_2 を定めた．

$$E_2 = \sum_{k=1}^{n} (1 - |Em(k)|)^2$$

関数 E_2 は各メソッド名に対応する分散表現のノルムが 1 に近いほど値が小さくなるため，この関数の値を最小化するように分散表現の値を調整することで分散表現が零ベクトルに向かうのを抑止することが期待される．

以上より，最終的な関数 E は以下の通りになる．

$$E = w_1 E_1 + w_2 E_2$$

w_1，w_2 はそれぞれ E_1 と E_2 の重みを決めるパラメータである．関数 E を損失関数として，この値を最小化するように分散表現の値を学習する．

また，単純に損失関数 E を最小化するように分散表現を調整した場合，すべてのメソッドの分散表現が等しくなるように調整されてしまう可能性があり，図 1 に示すような「分散表現が離れているメソッド間では意味の関連が弱い」ことを表現できない．対策として，学習の過程で各メソッドに関して呼び出し関係の無いメソッドをいくつか選択し，それらのメソッドの分散表現から離れるように注目しているメソッドの分散表現を調整する処理を行う．

3 まとめ

本稿では，メソッド間の呼び出し関係を利用してメソッド名に対する分散表現を獲得する手法を提案した．今後は，獲得した分散表現により，内部で呼び出しているメソッド群から元のメソッド名を提示できるかどうかを評価し，また，より多様なメソッド名を提案できるようにするため，メソッド名を構成単語毎に分割して提案する手法を検討する予定である．

参考文献

[1] N. Pennington. Comprehension Strategies in Programming. Empirical Studies of Programmers: 2nd Workshop, 1987.

[2] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In HLT-NAACL, 2013.