言語資源を利用したリポジトリマイニング基盤ツールの開発

山下大貴[†1] 千本達也[†2] 竹内和広[†1]

ソースコードのより深い解析を行うためには、自然言語による記述を考慮する必要がある。自然言語はソースコード中のコメントだけでなく、クラス名、メソッド名、変数名といった構成要素にも独特の形で出現する。我々は、ソースコードから言語表現を抽出した上で扱いやすいベクトル空間で特徴を表現するツール群を、ソースコード解析用の言語資源化を視座に開発している。また、ツールの有効性をソースコードのクラスタリング実験により評価した。

Development of fundamental repository mining tools based on natural language resources

HIROKI YAMASHITA,[†1] TATSUYA SENBON[†2]
and KAZUHIRO TAKEUCHI[†1]

For the deeper analysis on source code, we have to consider many natural language descriptions in it. Natural language is used not only for these comments, but also for variable names, class names, methods names etc. We propose a tool that extracts feature vectors from these descriptions in source code and manage the high dimensional vector space with some NLP (Ntatural Language Processing) resources and techniques. Examining the tools, we conducted a clustering experiment on source code files from some open softwares.

1. はじめに

ソースコードの詳細なマイニングのためには、ソースコード中の自然言語による記述を考慮する必要がある。ソースコードにはコメントの記述のように通常の自然言語による記述も存在するが、クラス名、メソッド名、変数名といった構成要素にも自然言語を利用した独特の記述がなされる。このような要素を扱うには、プログラミング言語の予約語の数をはるかに超える数の自然言語の語を扱わなくてはいけない問題と、語間の関係性を考慮しなくてはならない問題が生じる。

ソースコードにおける自然言語の使用は専門的であり、一般的な自然言語文章の特性をどこまで持ち合わせているかは明らかではない。柏原らは、NLP ツールである OpenNLP を用いてソースコード内の分析を行っている[2)]。OpenNLP は、整備・蓄積された自然言語の共通的な知識である辞書や新聞のような標準的なテキスト等の言語資源に基づいて作成されているため、テキストの一般的特性について解析可能である。しかし、クラス名、メソッド名、変数名には、一般的な語には見られない専門的な語との組み合わせで命名されることがあり、既存の NLP ツールだけではソースコード内に記述される語を十分に扱えない可能性がある。つまり、ソースコードの専門的な語を扱えるように、NLP ツールの拡張が必要であると考える。我々は、以上のような背景から、ソースコードの部分やファイルを特徴ベクトルに解析し、それを処理するための基礎ツール群を開発している。そのツール群は大きく分けて以下の 2 つである。

- ソースコード中の自然言語の語出現をできるだけ低次元のベクトルで表現するツール（ツール I）
- ツール I によって解析された大量のベクトルデータを扱うツール（ツール II）

ソースコード解析に必要な言語情報を、基盤資源として共有化できる仕組みを念頭に置いており、特にツール I 部分には、自然言語の語間の意味的関係性を辞書化したシソーラスの一つである WordNet を組み入れ、様々な前提で、ソースコード解析ができるように考えている。

2. ソースコードの特徴ベクトル化ツール

ソースファイルの自然言語の語彙に基づいて Bag of Words と呼ばれる最も単純な特徴ベクトルを作成す

†1 大阪電気通信大学 情報通信工学部
Osaka Electro-Communication University, Faculty of Information and Communication
†2 大阪電気通信大学 大学院工学研究科
Osaka Electro-Communication University, Graduate School of Engineering

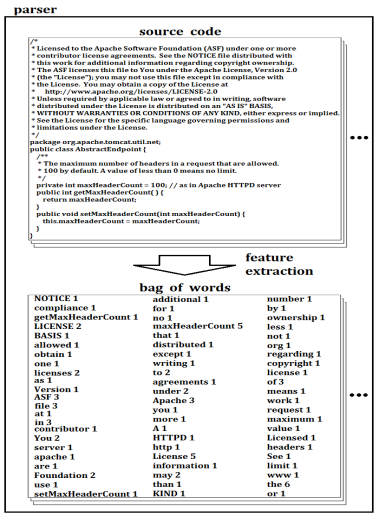

図 1　ソースコードからの Bag of words 抽出例
Fig. 1　Extracted BoW(Bag of words) from a source code

ると**図 1** のようになる。我々のツール I では、まず、ソースコード内のコメントやメソッド名、識別子など、自然言語で記述しうる部分要素を選択して特徴ベクトルを抽出できるように設計している。例えば、メソッド名に関してはキャメルケース、スネークケースのそれぞれに対応して語彙に分割する。

図 1 のような bag of words は、何も処理を行わなければ特徴ベクトルは一般に非常に大きな次元となり、機械処理が難しくなる。そこで、語彙資源と自然言語処理の手法により高次元ベクトル空間に対応する次元圧縮をツール I 内に実装した。次元圧縮ツールについては以下の 2 つの方向性から開発している。

(1)　WordNet を用いた次元圧縮
(2)　randomized SVD[1] を用いた次元圧縮

上記の (1) では WordNet を用いて辞書引きを行い、ソースコードから抽出した単語が定義されていればその語を特徴として用いることで一般的に扱われる語のみを特徴として次元圧縮する方法である。(2) では行と列方向で乱択化による行列圧縮を行い、効率的に SVD を処理をすることができる randomized SVD を実装している。

3. 実　　験

SVD は特徴ベクトルの次元を圧縮する。例えば、1 次元に圧縮したとき、すべての語が同義語となり、2 であれば類似単語集合は 2 つになる。SVD による次元圧縮の効果を、類似単語集合に WordNet 上の最隣接類義語のペアが含まれる数で評価した結果を図 2 に示す。この結果は、WordNet の一般的な類義語関係

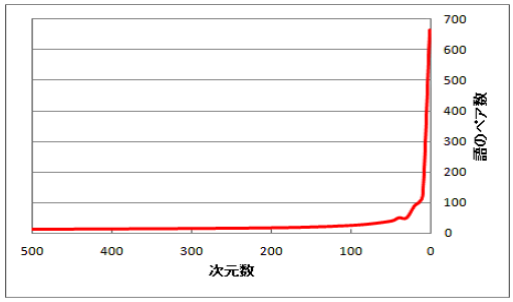

図 2　randomized SVD による次元圧縮
Fig. 2　Dimentional Reduction with SVD

がリポジトリマイニングでは必ずしも成立しないことを示唆する。

次に、**表 1** にクラスタリング対象のソースコード情報を示す。これをツール I によって圧縮した特徴ベクトルを用いてツール II によってクラスタリングする実験を行った。その結果、やはり WordNet のクラスタリングは SVD に比べて効果が低いという知見を得た。これは、ソースコード中の専門用語やシステム固有の用語を知識として整理する必要があることを示す。また、コメントの記述より、その他の部分の自然言語情報がクラスタリングに寄与していることが判明している。これは、プログラムのメソッド名他に出現する自然言語がより明確にソースコードの特徴を示す可能性があり興味深い。

4. 議　　論

一般的な語彙知識である WordNet は、そのままではソースコード解析に有効に機能しない。今後は、計算的手法で得られた知見を専門的ドメインでの語彙知識として整理することを考えている。

参 考 文 献

1) 岡野原 大輔: 全部分文字列のクラスタリングとその応用, 言語処理学会 第 17 回年次大会 発表論文集 (2011 年 3 月)
2) 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 (2013)

表 1　クラスタリングに用いたオープンソースプロジェクト
Table 1　Target projects for source code clustering

オープンソース	Jmeter	apache ant	Jedit	JDOM	Junit
ファイル数	899	733	514	200	142
次元	17205	15423	14070	6061	3773
異なり語数	3458	3353	2872	1625	610
WordNet 含有語	2146	1989	1957	899	540