Webにおけるサンプルコード検索支援手法の検討

飯田　修平†　平川　豊†

芝浦工業大学†

1. はじめに

現在、インターネットの普及によりWeb上にはプログラミング言語に関するサンプルコードが数多く存在している。これらサンプルコードはプログラミングの学習などにとても便利であるが「サンプルコードが記載されているWebページのみ欲しい」というケースでは実際にWebページを訪れないとサンプルコードの有無はわからないという問題がある。また「使用したことのないクラスの使用方法を調べる」というケースに関しては①クラスに属するメソッドを調べ、②調べたメソッドが使用されているサンプルコードを検索という2つの手順を踏むためサンプルコード検索には時間と手間がかかる。

そこで本研究では上記の「使用したことのないクラスの使用方法を調べる」というケースを想定したサンプルコード検索支援手法の検討を行う。具体的には先に説明した①と②をWebから抽出することで検索時間や負担の削減を試みる。また実際にサンプルコードの検索支援を行うシステムを作ることで、サンプルコード検索の時間削減や負担軽減ができているかの評価も行った。

2. 関連研究

抽出対象が文書中に出現する際のパターンに着目した研究には[1],[2]などがある。[1]の研究では“(呼称) こと (人物名)”というパターンを用いて人物の呼称抽出を行い、[2]では“(オブジェクト)の(外観情報)(構成要素)”というパターンによりオブジェクトの外観情報を抽出している。しかしこれらの研究では1つのパターンのみに着目し、抽出パターンの追加を考慮していないのでパターンの増加には対応ができない。またソースコード検索に関する研究[3]が存在するがこれは大規模ソフトウェア開発におけるCode Cloneを検出するというもので本研究とは対象とするドメインが異なる。

3. 本研究の提案

3.1. 概要

本研究では「使用したことのないクラスの使用方法を調べる」というケースにおけるサンプルコード検索支援を行うために

①クラスに属するメソッド名の抽出

②メソッドが使用されているサンプルコードの抽出

という2つの抽出をWebより行う。またこの際、「抽出したい対象の出現パターンや構造をルールとして登録し、登録されたルールに応じて該当するテキストを文書から抜き出す汎用的なパーサ」を用いている。

3.2. 抽出ルールを登録できる汎用的なパーサ

3.2.1.パーサの要求定義

本研究ではWebよりメソッド抽出とサンプルコード抽出を行う。一般的にWebからある対象の抽出を行う場合、その抽出対象が文書中に出現するパターンや構造には様々なものがある。例えば本研究の場合、プログラミング関連のWebページに出現する「Stringクラスのequalsメソッド」などの記述とメソッド一覧表に着目してメソッド抽出を行っている。またメソッド一覧表には<table>タグが用いられることが多いが<table>以外のタグが用いられるケースも存在するなど抽出対象の構造には様々存在する。このようにWebを用いた抽出には様々なケースに柔軟に対応できる仕組みが必要となる。

そこで抽出対象の出現パターンや構造をルールとして登録し、登録されたルールに応じて該当するテキストを文書から抜き出す汎用的なパーサを作成した。このパーサを用いることで上記2つの抽出対象を“抽出すべきルール”として統一的に扱え、また出現パターンや構造が新たに増えた場合でもルールを追加するだけで柔軟に対応できる。そして抽出ルールを変更することでサンプルコード抽出への適用も可能になる。

3.2.2.パーサに登録するルールの記述形式

メソッド抽出の際に利用するルールの例を表1に示す。

表1. メソッド抽出に使用するルールの例

R1	S→:input1:,,クラス,,$30,,:input2:
R2	S→:input1:,,クラス,,$15,,主なメソッド,,A
R3	A→$150,,<table>,,B,,$500,,</table>
R4	B→$150,,<table>,,B,,$500,,</table>
R5	B→$100,,:input2:

まず表1中に出現する単語であるが、”:input1:”はユーザ入力によって指定するクラス名、”:input2:”はメソッド名を表わすが実際には[a-zA-Z]†(メソッド|\\(.*?\\))”というメソッドに対応した正規表現となる。”$30”などの”$数値”は読み飛ばしを許容する文字数を表現する。そしてSは開始記号を、AとBは非終端記号である。

次にこのルールの意味を表1中のR1を例にとって説明する。Sは開始記号なので:input1:、つまりクラス名から始まりその直後に「クラス」という文字列が出現、そして$30,, :input2:は30文字以内にメソッド名が出現するような表記が存在すればパーサはその該当テキストを抜き出すという意味である。また登録するルールは一般的に次のように記述することができる。

$$V \rightarrow w_1,,w_2,,\cdots,,w_n$$

$$V \in \{\text{NT or T}\}$$

$$w_i \in \{\text{NT or T or \$数値 or :input}_i\text{:}\}$$

NTは非終端記号でNは終端記号、$数値は先に説明した読み飛ばし可能な文字数、そして:input$_i$:はユーザが定義できる終端記号である。このルールでは文脈自由文法のクラスの文字列を受理できるようになっている。

3.2.3.パーサの動作例

ここでは表1の抽出ルール集合をパーサがどのように適用していくのかの説明を行う。まずパーサは開始記号Sで始まるルールから適用するが、表1の場合ではR1とR2の2つが適用対象となる。このように適用ルールが複数存在する場合は上から順番(R1→R2)に1つずつ適用する。R1が全て成功した場合はそこで解析終了になるが、R1の適用が途中で失敗した場合にはR2に移行する。R2には非終端記号であるAが存在するのでR2のAまで全

A Study of Sample Code Retrieval Method on Web
†Shuhei Iida, Yutaka Hirakawa
(Shibaura Institute of Technology)

て成功した場合、次はAで始まるルールであるR3が適用対象となる。R3の途中にもBが存在するのでR4とR5という順番で適用していく。R4で失敗しR5で成功するとR3に戻り、残りの終端記号である"$500,,</table>"を適用し全て成功したら解析終了という流れになる。

3.3. パーサを用いたメソッド抽出

表1のルールが登録されたパーサを用いてWebよりメソッド抽出を行った。この結果を表2に示す。この表の適合率と再現率であるが、Yahooの提供するウェブ検索APIを用いて上位15件のWebページを取得しこれらより抽出すべきメソッドの正解集合を作成し算出している。事前実験として<table>タグで記述されたメソッド表のみからのメソッド抽出を行った際の再現率は0.73であるのに対して、今回の再現率は0.83と10%上昇している。これはパーサを用いることによって<table>タグ以外のメソッド表にも対応するルールと「Stringクラスのequalsメソッド」のような記述にも対応するルールが追加可能となったためである。

表2. パーサを用いたメソッド抽出結果

テストしたクラス数	適合率	再現率
30	0.88	0.83

3.4. Web上におけるサンプルコード抽出

使用方法を知りたいメソッドが使用されているサンプルコードをWebより抽出する。Web上のサンプルコードは<pre>タグ内で記述される性質があるので①<pre>タグを抽出するようにルール登録したパーサを用いてWebページ上の<pre>タグで囲まれたテキストを全て抜き出し、②目的とするメソッドが使用されているもののみを抽出することによってサンプルコード抽出を行う。このようにパーサの抽出ルールを変更することによって抽出対象も変えることができる。この抽出結果を表3に示す。この抽出実験でも同様に正解集合をあらかじめ人手により作成している。

表3. サンプルコード抽出の結果

テストしたクラスとメソッドのペア	適合率	再現率
20	0.85	0.84

4. サンプルコード検索支援システム

4.1. システムの概要

本研究はサンプルコード検索の時間削減やユーザへの負担減少を目的としているので

1. クラスに属するメソッドの抽出
2. メソッドが使用されているサンプルコードの抽出

を自動で行う検索支援システムを作成し、検索までの作業時間や負担を測定することでシステム評価を行った。

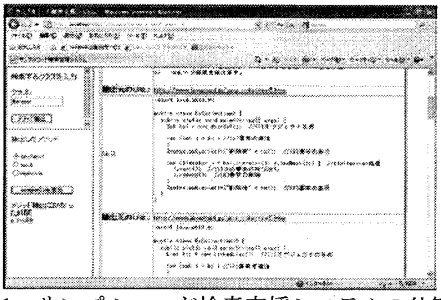

図1. サンプルコード検索支援システムの外観

実装にはJavaのEnterprise Editionである Servlet/JSP、アプリケーションサーバにはTomcatを用いた。

4.2. システム評価実験

システム評価として、従来(Yahoo!の検索エンジンを用いた)手法と提案システムを用いた際のサンプルコード検索に要する作業時間とクリック数の測定を行った。測定方法は「使用するクラスとメソッドが分かっている」という状況を想定し、与えられた10個のテストケース(クラスとメソッドのペア)のサンプルコードを各々1つ見つけるというものである。この測定を5人の被験者に行った際の結果を表4に示す。この表の作業時間は1つのテストケースのサンプルコードを見つけるまでに要した平均時間である。従来手法と比べてクリック回数は約6割、サンプルコードを発見するまでの作業時間は約5割の削減となりサンプルコード検索までの作業時間や負担の減少ができていると言える。

次にメソッド抽出とサンプルコード抽出にかかる時間を測定し、システムの数値評価を行う。測定はYahooの上位15件のWebページを対象とし、テストケースは3.3と3.4と同じものを用いた。この測定結果を表5に示す。メソッド抽出とサンプルコード抽出にかかる時間は順に平均9.2秒と6.9秒で、これを1ページ当たりの時間に換算すると0.86秒と0.63秒である。この数値は人がWebページを読む時間より十分早い時間内で処理できていると考えられる。

表4. クリック数と作業時間の比較

	従来手法	提案システム
クリック回数(回)	11	4
作業時間(秒)	75.9	35.5

表5. メソッドとサンプルコード抽出にかかる時間

	メソッド抽出	サンプルコード抽出
抽出時間(秒)	9.2	6.9

5. まとめと今後の課題

本研究ではWeb上におけるサンプルコード検索支援手法の検討を行った。メソッド表の様々な構造には抽出ルールを追加することで対応し、サンプルコード抽出では抽出ルールを変更することで抽出対象の変更にも対応が可能である。また検索支援システムを用いることで従来手法と比べてユーザの作業時間や負担を半減することができた。今後はさらにプログラミング関連のサイトを調査し、メソッド表以外の記述パターンや構造をルール登録していくことで対応できるクラスの幅を増やしていく。検索支援システムに関してもより使いやすいものへと改良していく予定である。

参考文献

[1]外間　智子, 北川　博之, "Webコーパスを用いた人物の呼称抽", 日本データベース学会 Letters, Vol. 5, No. 2, 2006年

[2]服部　峻, 手塚　太郎, 田中　克己, "オブジェクトの外観情報のWebマイニング", DEWS2007 L4-6

[3]Ducasse,S.;Rieger,M.;Demeyer,S.;"A Language Independent Approach for Detecting Duplicated Code", Software Maintenance,1999.IEEE International Conference:109-118

[4]飯田　修平,"Webにおけるサンプルコード検索支援手法の検討",第8回情報科学技術フォーラム(FIT 2009)