4J-04

GitHub と Stack Overflow における開発者行動の統一的な分析

永野真知† 早瀬康裕§ 駒水孝裕‡ 北川博之‡

†筑波大学情報学群 §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 はじめに

ソフトウェア開発を支援するサービスとして GitHub[1](以下，GH) や Stack Overflow[2] (以下，SO) などのソフトウェア開発者向けウェブサイトが用いられており，それぞれのウェブサイトが開発者の活動記録を蓄積している．GH ではソースコードの変更や変更に関する議論が記録されている．SO は開発者向け Q&A サイトであり，質問と回答が分野ごとに記録されている．

GH や SO などの各ソフトウェア開発者向けウェブサイトでは使用される目的が異なるため，各サイトで得られる開発者の活動記録の性質が異なっている．GH では，開発者がどのように開発に貢献しているかという情報を得ることができる．しかし，開発者が多くのリポジトリに貢献するには多大な時間を要するため，開発者あたりのデータ件数が少ない．一方で，SO では質問や回答から開発者がどのような分野の知識を持っているかという情報が得られる．また，開発者が容易に質問や回答を投稿できるため，開発者あたりのデータ件数が多い．しかし，開発者が持っている知識を利用してどう開発に貢献してきたかがわからない．

そこで我々は, 開発者向けウェブサイトである GH と SO のデータを統合し,GH と SO にまたがる開発者の行動の分析を行う. 本研究では,GH と SO の両方 に登録している同一開発者を推定し, 開発者が 2 つの サイト間で共通の関心をもって行動をしているか調査 する. 開発者が共通の関心を持って活動していること が明らかになれば, 組み合わせることでデータ解析や 応用アプリケーションの発展につながると期待できる.

2 GitHub と Stack Overflow にまたがる開発者行動の分析手法

GH と SO にまたがる開発者行動を分析する手法について述べる．分析の手順を図 1 に示す.

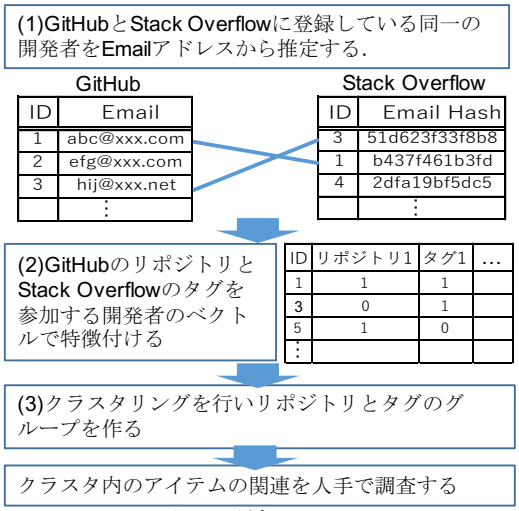

図 1: 手法チャート

使用する GH のデータを MSR Challenge 2014 [1] から，SO のデータを[3]Stack Exchange の 2015 年 8 月の公式ダンプから取得する．GH と SO のデータには，それぞれユーザ名や Email アドレスといった開発者の情報が含まれている．GH のデータにはリポジトリの情報や開発者が行ったコミットの情報などが含まれている．SO のデータには，質問や回答の情報，開発者が投稿を行った情報などが含まれている．

(1) GH と SO のデータを組み合わせるため，GH と SO 間で同一の開発者を Email アドレスに関する情報から抽出する．まず，使用する GH のデータを MSR Challenge 2014 [1] から，SO のデータを[4]Stack Exchange の 2015 年 8 月の公式ダンプから取得する．GH のデータに含まれる開発者の情報には Email アドレスが記載されている．一方で，SO のデータに含まれる開発者の情報には，Email アドレスをハッシュ化した値が記載されている．そこで，GH の Email アドレスをハッシュ化して得られる値と同じ値を持つ開発者を同一開発者として抽出する．

(2) 開発者から得られる情報を基にアイテムを特徴付けるベクトルを作成する．開発者数を n とした場合，各

User Behavior Analysis across GitHub and Stack Overflow
Machi NAGANO†, Yasuhiro HAYASE§, Takahiro KOMAMIZU‡, and Hiroyuki KITAGAWA‡
†, §,‡University of Tsukuba

[1]https://github.com/
[2]http://stackoverflow.com/

[3]http://u.brentozar.com/StackOverflow201508.7z.torrent
[4]http://u.brentozar.com/StackOverflow201508.7z.torrent

アイテムを n 次元の特徴ベクトル $\mathbf{x}_i = (t_0, \ldots, t_{n-1})$ で表現する．アイテム i がリポジトリの場合，開発者 u がアイテム i に対してコミットしている場合に $\mathbf{x}_i$ の u 番目の要素を 1 にする．また，アイテム i がタグの場合，開発者 u がアイテム i に関連する質問や回答の投稿・編集を行っている場合に $\mathbf{x}_i$ の u 番目の要素を 1 にする．データから得られるすべてのリポジトリを使用する場合，フォークしたリポジトリが含まれる．フォークしたリポジトリへのコミットは，フォーク元のリポジトリに反映させるために作業したものが多い．そこで，クローン数が少ない不人気のリポジトリへのコミットをフォーク元のリポジトリへのコミットとする．また，タグに関しては，開発者からの利用頻度が低いものは除外する．

(3) 作成したベクトルを用いてアイテムに対してクラスタリングを行い，開発者の活動記録から類似したアイテムを検出する．クラスタリングには階層的クラスタリングであるウォード法 [2] を利用する．階層的クラスタリングは，分類される過程を階層構造で表すことができるため，クラスタリング結果の解析に向いている．ウォード法は階層的クラスタリングの中でも分類精度が高いことが知られている．本研究では距離関数として 1 からコサイン類似度を引いた値を使用する．

3 評価実験

提案手法を実施し，各クラスタ内に含まれる GH と SO のアイテムを分析する．GH と SO のデータから得られた同一開発者は 25831 人，リポジトリは 3173 個，タグは 23521 個となった．クローン数の閾値を 5，タグの利用頻度の閾値を 10 としたところ，リポジトリ数は 88 個，タグ数は 7173 個となった．階層的クラスタリングの距離の閾値は 3 とし，同一クラスタ内にある GH と SO のアイテムが類似しているか比較する．クラスタ内に GH と SO の両方のアイテムを含んでいるクラスタは 13 個存在した．

得られたクラスタ内に含まれたリポジトリとタグが同じ分野のものであるか確かめる．GH と SO の両方のアイテムを含み，関連性のあるアイテムを分類しているクラスタの例を表 1 に示す．リポジトリ名の '/' の前はリポジトリの所有者のユーザ名を示している．下線が引いてあるタグは同じ分野のタグであると私が判断したものである．クラスタ 1 について見てみると Ruby に関連するライブラリや機能に関連するアイテムが約 40%含まれていた．リポジトリの homebrew は macOS 用のパッケージマネジャであり，プログラムは Ruby で書かれているため Ruby に関心の有る開発者たちに興味を持たれていると考えられる．同じようにクラスタ 2 とクラスタ 3 について見てみると，クラスタ 2 は Microsoft .Net Framework に関連するリポジトリとタグ，クラスタ 3 は，Scala に関連するリポジトリとタグが分類されていた．

表 1: クラスタごとの類似するアイテムの例

ID	アイテム (上段がリポジトリ，下段がタグを表す)
1	plataformatec/devise, mxcl/homebrew, thoughtbot/paperclip, rails/rails, vmg/redcarpet
	ab-testing, cedar, clearance, diaspora, dragonfly-gem, factory-girl, faye, fields-for, fog, gmaps4rails, inherited-resources, jammit, jugger-naut, machinist, merb, meta-search, minitest, mocha, oauth-ruby, padrino, phusion, polymorphic, rails-3.1, rcov, refinerycms, sqlite3-ruby, tire, tmux, single-table-inheritance, wicked-pdf
2	AutoMapper/AutoMapper, NancyFx/Nancy, ravendb/ravendb, restsharp/RestSharp, ServiceStack/ServiceStack, SignalR/SignalR,
	automated-testing, autopostback, checkin, cqrs, document-oriented-db, envdte, event-sourcing, http-status-code-405, knockout-mapping-plugin, mschart, mspec, modelstate, nancy, norm, psake, service-layer, solid-principles, specflow, ticket, tweetsharp, xunit, wcf-rest, webfarm
3	sbt/sbt, scala/scala
	casbah, dispatch, implicits, scalate, scalaquery, sub-type, tuple-unpacking, trait

4 結論

GH と SO を組み合わせたデータを分析し，開発者は GH と SO 間で共通の関心を持ち活動しているか調査した．評価実験の結果，開発者は GH と SO 間で共通の一定の関心を持って活動していることがわかった．GH と SO のデータを統合することは GH のデータで不足した情報と SO のデータで不足した情報を補い，データ解析や応用アプリケーションの発展につながると期待できる．

参考文献

[1] G. Gousios. The ghtorrent dataset and tool suite. In *Proc. the 10th Working Conf. on MSR*, MSR'13, pages 233–236, 2013.

[2] J. H. Ward. Hierarchical grouping to optimize an objective function. In *J.Am.Stat.Assoc.,Vol.58,1963*, pages 236–244, 1963.