修 士 論 文

題　目　　コミットメッセージの感情推定と
　　ソースコードにおける不具合出現の関係分析

主任指導教員　　水野　修　准教授

京都工芸繊維大学大学院 工芸科学研究科

情報工学専攻

学生番号　　11622030

氏　　名　　松井　章

平成28年8月10日提出

様式 2 号

学位論文の要旨（和文）

平成 28 年 8 月 10 日

京都工芸繊維大学大学院
工芸科学研究科長　殿

工芸科学研究科　　情報工学専攻
平成 23 年入学
学生番号　　11622030
氏　　名　　松井　章　㊞

（主任指導教員　　水野　修　㊞　）

本学学位規則第 4 条に基づき、下記のとおり学位論文内容の要旨を提出いたします。

1.　論文題目

コミットメッセージの感情推定とソースコードにおける不具合出現の関係分析

2.　論文内容の要旨（400 字程度）

近年，自然言語処理の研究分野において感情推定が注目されている．これは一般的に，文章から筆者や話者の意見の極性を特定することに焦点を当てたテキスト分析である．本論文では，ソースコード管理に用いるリポジトリに蓄積されたコミットコメントに対して感情推定を行い，開発者の感情極性とソースコードに含まれる不具合混入率について調査を行った．9 つのソフトウェアプロジェクトに対して実験を適用した結果，それらのプロジェクトの間では，コミットコメントの感情極性においては差がほとんど存在しないことを発見した．また，不具合を混入させやすい開発者とそうでないものを 2 グループに分けた場合に，それぞれの感情極性において有意な差が存在すること，さらに不具合混入率の高い開発者ほどポジティブなコメントをおこなっている傾向にあることを発見した．

Investigating Emotional Plarities on commit comments of code repositories

2016 11622030 *MATSUI Akira*

Abstract

Recentlry Sentiment analysis is becoming popular in research area of natural language processing. Those are generally text analysis that are focused on

目　次

1. 緒言

ソフトウェアの開発をより効率的に行うため，様々な研究が行われている．その1つにリポジトリマイニングという手法がある．リポジトリとは，バージョン管理システムを用いたソフトウェア開発プロジェクトにおけるソースコードやコミット情報をまとめて保管することができるデータベースである．このデータベースを分析し，ソフトウェア開発における有益な知見を発見することがリポジトリマイニングの目的である．

リポジトリマイニングの研究の一環として，ソフトウェアリポジトリに含まれるソースコードなどのテキストデータに対して，テキスト分類やトピック分析などの自然言語処理の手法を適用する研究が進められている．

近年，自然言語処理の研究分野において，感情分析が注目されている．これは一般的に，文章から筆者や話者の意見の極性を特定することに焦点を当てたテキスト分析である．商品のレビューやソーシャルメディアなどに適用され，マーケティングや顧客サービスを効率的に行うことを目的に利用されている．

感情分析の手法をリポジトリ内のデータに適用することで，ソフトウェア開発者の感情と成果物の品質などの間の関係を発見することができるのではないかという考えから，本研究では，リポジトリに蓄積されたコミットコメントに対して感情分析を行い，ソースコードにおける不具合の出現と感情極性の関係の分析を行う．

具体的には

リポジトリに記録されている不具合情報から，不具合を混入させたコミットとその開発者を推定し，開発者ごとの不具合混入率と彼らが残したコミットコメントの感情極性がどのような関係にあるのか調査する．

本論文の以降の構成を述べる．2章では研究の基礎となる関連研究について述べる．3章では本研究において設定した研究設問を述べる．4章ではコミットコメントの感情推定の手法の概要について述べる．5章と6章では，コミットコメントの感情推定の手法を実際のプロジェクトに適用する実験方法とその結果及び考察を述べる．7章では本研究のまとめについて述べる．

2つの研究設問に対する

それぞれ

ソフトウェアの開発は人間の知的活動の成果であり、開発時点での開発者の感情がソフトウェアの品質に影響を与えることは想像に難くない。

2. 関連研究

ソフトウェア工学分野では，リポジトリマイニングの隆盛に伴ってバージョン管理システムの変更履歴を用いて不具合を予測する研究が行われている．Nagappan らは，ソースコードの変更履歴から不具合を予測する研究 [1] を行った．また，Kim らは過去の不具合の修正の履歴から未発見の不具合を予測する手法 [2] [3] を提案した．本研究ではこれらの研究と同様に，バージョン管理システムの変更履歴と不具合の関係を調査する．これらの研究はバージョン管理システムに記録されている情報のみを利用しているのに対し，本研究ではバージョン管理システムに記録されている情報から開発者の感情を推定し，それらと不具合の関係を調査する．

感情推定の分野において様々な研究が行われている．Adam らは，SNS である FaceBook を対象に実験 [4] を行った．その結果，ポジティブな投稿の表示を減らした場合，そのユーザ自身のポジティブな投稿が減り，ネガティブな投稿が増えるという結果を示した．高野らは感情推定法の提案を行い，新聞社のアクセス解析へ適用 [5] した．その結果，ポジティブな記事を閲覧した人は連続でポジティブな記事を閲覧しやすく，同様にネガティブな記事を閲覧した人は連続でネガティブな記事を閲覧しやすいという結果を示した．これらの研究の結果から，ソフトウェアのソースコードに感情推定が適用できた場合，開発者が読むソースコードの感情極性によってその開発者が書くソースコードの感情極性が左右されると考えられる．従って，本研究では将来的にソースコードの感情極性からソースコードの不具合の発生の防止や，不具合を予測することを目的として，ソースコードに感情推定を適用する．ソフトウェア工学の分野に感情推定を適用した様々な研究も行われている．Michal は 56 人のソフトウェア開発者を対象に調査を行い，ほとんどの開発者はポジティブな感情を持っている時は生産性が向上し，ネガティブな感情を持っている時は生産性が向上しにくいという結果を示した [6]．Sebastian らはソフトウェア開発者を対象にバイオメトリックセンサーを用いた実験を行った [7]．その結果，開発者の感情と作業の進捗には相関があることを示した．Alessandro らは，バグ管理システムのバグ報告から自動的に感情を分析するツールを作成した [8]．Emitza らはソフトウェア開発プロジェクトのための Web 共有サービスである GitHub [9] を対象に，異なるオープンソースプロジェクトのコミットコメントに対して感情分析を行い，感情と

用いられるプログラミング言語やコミットコメントが書かれた時間・曜日などの関係を調査した [10].

何かまとめがほしい

3. 研究設問

本研究は将来的に，開発者がどのような感情をもちながら開発を行っている時に不具合が混入されやすいのかを特定し，開発者の精神的な負担の軽減や，不具合の発生を未然に防止することを目的としている．

これらの目的のために，本研究では，開発者の感情極性とコミットコメントの間に関係が存在するかを調査する．

従って，本研究では次に設定する研究設問について調査を行う．

RQ1：異なるプロジェクトにおいてコミットコメントの感情極性に差があるか．

RQ2：開発者の不具合混入率と感情極性に関係があるか．

RQ1 では複数のプロジェクトを選定し，コミットコメントから感情極性を抽出する．そして，それぞれのプロジェクトの間に有意な差が存在するかどうかを調査する．

RQ2 では，開発者ごとの不具合混入率を算出し，全てのプロジェクトを通して不具合率と感情極性の関係について調査する．

4. コミットコメントの感情推定と不具合率の算出

本研究では，以下の手順でコミットコメントの感情推定及び不具合率の算出を行う．

1. ソフトウェアリポジトリからコミットコメントを抽出する．
2. 抽出したコミットコメントから感情推定を行う．
3. ソフトウェアリポジトリとバグデータベースの不具合情報を統合する．
4. 開発者ごとの不具合混入率，感情極性の統計値を計算する．

4.1 コミットコメントの抽出

バージョン管理をおこなっているソフトウェア開発プロジェクトでは，ソースコードの更新をリポジトリにコミットする際，その更新内容についてのコメントを残すことが慣習となっている．それらのコメントはコミットに紐付いて管理され，必要であれば諸々のツールで参照することができる．

本研究では Git リポジトリで管理されているオープンソースソフトウェア開発のプロジェクトを用いる

Git リポジトリでは，git コマンド群により提供される git show コマンドや git log コマンドによりコミットに関する情報を取得することができる．

Git リポジトリから取り出されるコミット情報にはそのコミットに関する様々な情報が含まれている．例えば git show コマンドでは次のようにコマンドオプションを付与することで，コミット ID・開発者・コミットコメントを整形しながら抽出することができる．

```
$ git --no-pager show --summary --date=iso \
      --pretty=format:'"%h","%an","%s"' $TAG | grep "\""

"f3ff296","Xenia M Khailenka","Update release notes"
```

4.2 感情推定

前節で得られたコミットコメントを文章とみなして感情推定を行う．本研究では NLTK [11] に実装された VADER [12] の感情極性推定手法を用いた．VADER はマイクロブログのようなソーシャルドメインに用いられるテキストに対して有効であることがわかっている．また VADER による感情推定において，学習データは不要である．

この手法では，入力した英語の文章に対して感情推定を行い，次の 3 つの感情極性を 0.0 から 1.0 の数値として抽出され，それぞれの値の総和は 1 となる．

positive：良い印象を与える度合い

negative：悪い印象を与える度合い

neutral：良し悪しの判断が付かない度合い

例えばある文章に対して VADER の感情推定を行い，positive が 0.75，negative が 0.10，neutral が 0.15 というような結果が得られた場合，この文章は比較的良い印象を与える文章であると言える．

4.3 不具合情報の統合

バージョン管理システムで管理されたソフトウェアリポジトリとバグデータベースに記録されている不具合情報を統合し，不具合を混入させたコミットを識別する．

本研究では，不具合情報の統合には SZZ アルゴリズムを用いる．SZZ アルゴリズムは，バグデータベースの情報とリポジトリに保存されたソースコード編集履歴を相互に結びつけることで，不具合を混入させたコミットを特定する．

SZZ アルゴリズムの基本的なアイデアを以下に示す．

- バグデータベースからバグの修正を行っているログを抜き出す．
- バグのログとバージョン管理システムを結びつけて，バグの修正を行ったソースコードの変更を抜き出す．
- バグの報告日以前のソースコードの変更をバグの原因となった変更として決定する．

4.4 感情極性と不具合混入率の統計量の計算

節 4.2 の方法で得られる感情極性は一つのコミットから得られるコミットコメントから算出される．本研究では開発者単位での不具合混入率について調査を行うため，開発者ごとのコミットコメントから得られた感情極性を集計し，positive・negative・neutral の平均値と標準偏差を説明変数とする．

また，SZZ アルゴリズムにより特定された不具合混入コミットを集計することで，開発者ごとの不具合混入率を得ることが出来る．

従って，ひとりの開発者について，次の 7 つの変数を得る．

M_{pos}：positive の平均値

M_{neg}：negative の平均値

M_{neu}：neutral の平均値

SD_{pos}：positive の標準偏差

SD_{neg}：negative の標準偏差

SD_{neu}：neutral の標準偏差

R_{bug}：不具合混入率

5. 適用実験 RQ1

5.1 実験方法

5.1.1 対象プロジェクトを選定する

本研究では，次の 9 つのオープンソースプロジェクトのリポジトリを対象とした．

P_1：Apache MINA

P_2：Apache OpenJPA

P_3：Apache Cayenne

P_4：Eclipse BIRT

P_5：Eclipse Platform

P_6：Eclipse ECF

P_7：Eclipse EMF

P_8：Eclipse webtool

P_9：Eclipse xpand

Emitza らの調査によれば，プロジェクトが採用するプログラミング言語によってコミットコメントの感情極性に違いがある．そのため，本研究では調査対象のプロジェクトはすべて Java を採用したものに限定した．

5.1.2 コミットコメントの抽出

4.1 節の手順に従い，対象プロジェクトにおける全てのコミットコメントを抽出することで，コミット id・コミットを行った開発者名・コミットコメントを得た．

5.1.3 感情推定

前節で抽出した全てのコミットコメントについて感情推定を行った．Python から NLTK の nltl.sentiment.vader を次のように呼び出すことでコミットコメントの感情極性を得た．

```
>>> from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

```
>>> vader = SentimentIntensityAnalyzer()
>>> vader.polarity_scores("it's runnning better than before.")
{'pos': 0.42, 'neg': 0.0, 'neu': 0.58, 'compound': 0.4404}
```

さらに，節 4.4 で列挙した変数を開発者ごとに計算することで，次のような形式のデータを 9 つのプロジェクトの個数だけ得た．なお，pos_mean や neg_sd はそれぞれ M_{pos} や SD_{neg} を表している．

```
author, pos_mean, neg_mean, neu_mean,  pos_sd,  neg_sd,  neu_sd
Andrey,  0.12345,  0.12345,  0.12345, 0.12345, 0.12345, 0.12345
Evgeny,  0.12345,  0.12345,  0.12345, 0.12345, 0.12345, 0.12345
...
```

5.1.4 プロジェクト間の有意差の検定

以上の手順により，それぞれのプロジェクトにおける開発者ごとの感情極性の平均値と標準偏差を得た．

ここで，開発者の感情極性の平均値と標準偏差において異なるプロジェクトの間に有意な差が存在するかどうかを調べるため，次のような仮説を設定した．

H_0：それぞれのプロジェクトの間には有意な差がない．
H_1：それぞれのプロジェクトの間には有意な差がある．

本実験では検定対象であるそれぞれの標本は非等分散である仮定し，welch の t 検定を行った．数値計算については，統計分析ソフトである R [13] による実装を利用した．以下に R による Welch の t 検定のサンプルを示す．

```
> sample_a <- c(1,2,3,4)
> sample_b <- c(1,1,3,4)
> t.test(sample_a, sample_b, var.equal=F)

  Welch Two Sample t-test

data:  sample_a and sample_b
t = 0.25265, df = 5.8698, p-value = 0.8092
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.184368  2.684368
```

```
sample estimates:
mean of x mean of y
```

一つのプロジェクトにおいて M_{pos}・M_{neg}・M_{neu}・SD_{pos}・SD_{neg}・SD_{neu} の六つの変数があり，他の八つのプロジェクトと二標本検定を繰り返すため，プロジェクトごとに 48 個の p 値をそれぞれ得た．

5.2 実験結果

それぞれのプロジェクト間において検定を行い，得られた 48 個の p 値のうち有意水準を 5% とした場合に仮説 H_0 が棄却された個数を表 5.1 に示す．

なお，各プロジェクトの感情極性の抽出結果については付録 A 参照のこと．

5.3 考察

本実験では，異なるプロジェクト間における開発者のコミットコメントの感情極性に差があるのかについて調査した．

表 5.1 によれば，総じて仮説 H_0 が棄却された数は少ない．つまり異なるプロジェクト間における感情極性には差が存在するとは言えない．

一方で，プロジェクト P_4，つまり Eclipse Platform に関しては他のものよりも多く仮説 H_0 が棄却されているため，なんらかの理由で他のプロジェクトと違う傾向を示しているのではないかと考えられる．

P_4 以外のプロジェクトを除いて再度検定を行うと，表??の結果が得られた．

棄却された H_0 のうちの 32 件が Eclipse Platfor によるものであったため，全体的に割合が小さくなったことが確認できる．

以上より，RQ1 での設問「異なるプロジェクトにおいてコミットコメントの感情極性に差があるか」は「違いは発見できない」と結論づける．

表 5.1　各プロジェクトの t 検定の結果

プロジェクト	H_0 が棄却された数	棄却率
P_0	3	0.063
P_1	10	0.208
P_2	6	0.125
P_3	5	0.104
P_4	17	0.354
P_5	5	0.104
P_6	1	0.021
P_7	3	0.063
P_8	2	0.042

表 5.2　P_4 を除いて t 検定を行った結果

プロジェクト	H_0 が棄却された数	割合
P_0	2	0.095
P_1	6	0.048
P_2	4	0.143
P_3	1	0.024
P_5	3	0.071
P_6	0	0.000
P_7	1	0.024
P_8	1	0.024

6. 適用実験 RQ2

RQ2 について調査するため，二つの実験をおこなった．

一つ目の実験では，開発者を不具合混入率の高いグループと低いグループに別け，それぞれのグループに所属する開発者の感情極性に違いがみられるかを調査した．二つ目の実験では，全ての開発者の感情極性の値と不具合混入率の相関について調査した．

6.1 実験 1 方法

6.1.1 対象プロジェクトを選定する

RQ1 についての調査の結果から，プロジェクト間の感情極性においては有意な差は存在しないと結論づけた．しかし，その中でも Eclipse Platform においては他のプロジェクトと傾向の異なる結果を示していたため，本実験では，精度を高める目的でこれを除外して実験を進めた．従って，実験に利用するプロジェクトは以下の 8 つとした．

- Apache MINA
- Apache OpenJPA
- Apache Cayenne
- Eclipse BIRT
- Eclipse ECF
- Eclipse EMF
- Eclipse webtool
- Eclipse xpand

コミットコメントの抽出及びそれらに対する感情推定については RQ1 の実験結果を流用した．

6.1.2 不具合情報の統合

4.3 節の手順に従い，研究室で開発・保守されているツールである `szz_tools` を用いて不具合情報の統合を行った．上述の 8 個のリポジトリに対して，バグデータベー

ス JIRA [14] の情報とリポジトリの更新情報を結びつけた．

このツールを対象のリポジトリに対して実行すると，4.3 節で述べたように，不具合を混入させたと考えられるコミットに BUG-(issue ID)-(fix ID)-(bug ID) という tag が付与される．

6.1.3 不具合混入率の算出

前節で得られた不具合混入 tag を集計し，それぞれの開発者の不具合混入件数を得た．また，あらかじめ得られている開発者ごとの総コミット数を除すことで不具合混入率を算出した．

6.1.4 グループ分け

不具合混入率の高さを基準に，開発者を二つのグループに分ける．それぞれのグループのサンプルサイズが等しくなるように閾値を設定し，不具合混入率の高い方のグループを H グループ，低い方のグループを L グループとした．

ここまでの手順で，次のような形式のデータが得られた．

```
author, pos_mean, neg_mean, neu_mean,  pos_sd,  bug_ratio, group
Andrey,  0.12345,  0.12345,  0.12345, 0.12345,    0.23456,     H
Evgeny,  0.12345,  0.12345,  0.12345, 0.12345,    0.01234,     L
...
```

6.1.5 Mann-Whitney の U 検定

それぞれのグループに属する開発者の感情極性に有意な差があるのかを調査するため，次の仮説を設定した．

H_0：H グループと L グループの間には有意な差がない．
H_1：H グループと L グループの間には有意な差がある．

仮説 H_0 を棄却できるかどうかを調べるため，検定を行った．ここでそれぞれの標本は非正規分布に従うことを仮定し，Mann-Whitney の U 検定を用いた．H グループと L グループの対応する変数を検定対象とし，M_{pos}・M_{neg}・M_{neu}・SD_{pos}・SD_{neg}・SD_{neu} に対する 6 つの p 値を得た．

6.2 実験 1 結果

前節で求めた H グループ及び L グループの閾値とサンプルサイズを表 6.1 に示す．また，Mann-Whitney の U 検定を行った結果を表 6.2 に示す．

6.3 実験 2 方法

6.3.1 感情極性と不具合混入率の相関

実験 2 では，再度すべての開発者を 1 つのグループに統合し，それぞれの感情極性と不具合混入率の相関について調査した．不具合混入率の分布については何も仮定しないため，本実験では Spearman の順位相関係数を採択した．数値計算については，Python のデータ解析用ライブラリである pandas による実装を利用した．以下に Spearman の相関係数の計算のサンプルを示す．

```
>>> import pandas as pd
>>> df
   pos mean  bug ratio
0        0.8        0.30
1        0.5        0.20
2        0.2        0.23
3        0.1        0.05
>>> df.corr(method='spearman')
           pos mean  bug ratio
pos mean         1.0        0.8
bug ratio        0.8        1.0
```

6.4 実験 2 結果

不具合混入率とそれぞれの感情極性の Spearman の順位相関係数を得た．実験 2 の結果を 6.3 に示す．

表 6.1　それぞれのグループのサンプルサイズと閾値

グループ	不具合混入率の閾値	サンプルサイズ
L	≤ 0.205	105
H	> 0.205	105

表 6.2　Mann-Whitney の U 検定の結果

感情極性	p 値
pos mean	0.0000139
neg mean	0.4891321
neu mean	0.1126360
pos sd	0.0000563
neg sd	0.4419391
neu sd	0.1426264

表 6.3　感情極性と不具合混入率の相関係数

項目	相関係数
pos mean	0.384
neg mean	0.021
neu mean	-0.232
pos sd	0.322
neg sd	0.007
neu sd	0.155

6.5 考察

6.5.1 実験 1 について

表 6.2 によると，pos の平均値と標準偏差において $p < 0.05$ であるため，有意水準 5% で仮説 H_0 は棄却される．つまり，開発者を不具合混入率が多いグループと少ないグループの二つに分けた場合，それぞれの開発者のコミットコメントのポジティブさの平均値及び標準偏差には違いが存在するといえる．一方でコミットコメントのネガティブさや平坦さに関しては，二つのグループにおいて違いが存在するとは言えない．

6.5.2 実験 2 について

表 6.3 によると，pos mean と不具合混入率の相関係数は 0.384，pos sd と不具合混入率の相関係数は 0.322 であることから，pos の平均値及び標準偏差には不具合混入率との間に弱い相関があるといえる．つまり，コミットコメントのポジティブさの平均値が高く，あるいはポジティブさの振れ幅の大きな開発者ほど不具合を混入させる傾向にあるということが言える．

6.5.3 仮タイトル

ここで，RQ2 での設問について考察する．

実験 1 の結果から不具合混入率の高いグループと低いグループに分けた場合に，それぞれのグループに属する開発者の感情極性には有意な差が存在することを発見した．さらに実験 2 では，それらの感情極性においては不具合率との正の相関が存在することを発見した．

従って，RQ2 での設問「開発者の不具合混入率と感情極性に関係があるか」は「関係がある」とし，さらにその関係については次の 2 点にであると結論づける．

- コミットコメントのポジティブさの平均値が高い開発者ほど不具合混入率が高い．
- コミットコメントのポジティブさの標準偏差が高い開発者ほど不具合混入率が高い．

7. 結言

本研究では，リポジトリ管理されたプロジェクトから取り出したコミットコメントを文章とみなして感情推定を行い，開発者の感情極性とソースコードにおける不具合出現の関係を調査した．9個のオープンソースソフトウェアプロジェクトに対して実験を適用し，以下の分析結果が得られた．

- プロジェクト間の感情極性には有意な差が存在しない．
- 不具合混入率の高い開発者とそうでない開発者には感情極性に有意な差が存在し，特に不具合混入率の高い開発者はポジティブなコメントをする傾向にある．

これらの結果から，コミットコメントと感情極性の間には関係が存在し，コミットコメントの感情推定は不具合の分析に有効であると考えられる．

今後の課題としては，

- 稼働の高い開発者の影響を考慮した分析を行う．

コミットメッセージに含まれる特殊な表現などを除去したより感情があらわれるであろう部分について実験を行いたい．

謝辞

本研究を行うにあたり，研究課題の設定や研究に対する姿勢，本報告書の作成に至るまで，全ての面で丁寧なご指導を頂きました，本学情報工学部門水野修准教授に厚く御礼申し上げます．本報告書執筆にあたり，業務との両立に追われていた執筆者を励ましていただいた勤務先の浅井社長，市川取締役をはじめとする同僚の皆さん，学生生活を通じて著者の支えとなった家族や友人に深く感謝致します．

参考文献

[1] N. Nagappan and T. Ball, “Use of relative code churn measures to predict system defect density,” Proceedings of the 27th international conference on Software engineering, pp.284–292, May 2005.

[2] S. Kim, K. Pan, and J.E.J. Whitehead, “Memories of bug fixes,” Proceedings of 14th ACM SIGSOFT international symposium on foundations of software engineering, pp.34–45, Nov. 2006.

[3] S. Kim, T. Zimmermann, J.E.J. Whitehead, and A. Zeller, “Predicting faults from cached history,” Proceedings of the 29th international conference on Software Engineering, pp.489–498, May 2011.

[4] A.D.I. Kramer, J.E. Guillory, and J.T. Hancock, “Predicting faults from cached history,” Experimental evidence of massive scale emotional contagion through social networks, vol.111, pp.8788–8790, 2014.

[5] 高野憲悟，萩原将文，“感情関連語を用いた感情推定法の提案とニュースサイトのアクセス解析への応用,” Proceedings of the 29th international conference on Software Engineering，vol.11，no.3，pp.495–502，2012.

[6] M.R. Wrobel, “Emotions in the software development process,” Proceedings of the 6th International Conference on Human System Interaction, pp.518–523, 2013.

[7] S.C. Muller and T. Fritz, “Stuck and frustrated or in flow and happy: Sensing developers’ emotions and progress,” Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, pp.688–699, 2015.

[8] A. Murgia, P. Tourani, B. Adams, and M. Ortu, “Do developers feel emotions? an exploratory analysis of emotions in software artifacts,” Proceedings of the 11th Working Conference on Mining Software Repositories, pp.262–271, 2014.

[9] Github, Inc., Github, （オンライン）, 入手先〈http://github.com〉（参照 2016-08-06）.

[10] E. Guzman, D. Azocar, and Y. Li, “Sentiment analysis of commit comments in

github: An empirical study," Proceedings of the 11th Working Conference on Mining Software Repositories, pp.325–355, 2014.

[11] NLTK: Natural Language Tool Kit, （オンライン）, 入手先〈http://www.nltk.org/〉（参照 2016-08-06）.

[12] C.J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," Association for the Advancement of Artificial Intelligence, 2014.

[13] R. Ihaka and R. Gentleman, "R: A language for data analysis and graphics," Journal of Computational and Graphical Statistics, vol.5, no.3, pp.299–314, 1996.

[14] Atlassian, JIRA, （オンライン）, 入手先〈https://ja.atlassian.com/software/jira〉（参照 2016-08-06）.

付録 A.　各プロジェクトの感情極性の抽出結果

表 A.1　各プロジェクトの感情極性（平均値）

projects	pos mean	neg mean	neu mean
P_1	0.038	0.047	0.914
P_2	0.068	0.041	0.891
P_3	0.046	0.056	0.897
P_4	0.048	0.040	0.912
P_5	0.025	0.082	0.894
P_6	0.042	0.031	0.927
P_7	0.048	0.034	0.918
P_8	0.056	0.034	0.910
P_9	0.039	0.041	0.920

表 A.2　各プロジェクトの感情極性（標準偏差）

projects	pos sd	neg sd	neu sd
P_1	0.080	0.098	0.122
P_2	0.110	0.089	0.137
P_3	0.105	0.117	0.156
P_4	0.093	0.089	0.130
P_5	0.073	0.116	0.138
P_6	0.083	0.081	0.125
P_7	0.094	0.069	0.130
P_8	0.092	0.074	0.130
P_9	0.088	0.087	0.136