

Investigando a Relação das Refatorações de Código com os Sentimentos de Mensagens de *Commit*

Jordão M. de Souza¹, Ticiania L. Coelho da Silva¹, Criston P. de Souza¹,
Carla Ilane Moreira¹, Lincoln Rocha¹, José Antônio F. de Macêdo¹

¹Universidade Federal do Ceará (UFC)
Ceará – Brasil

jordao05@alu.ufc.br, jose.macedo@lia.ufc.br,

{ticianalc, criston, carlailane, lincolnrocha}@ufc.br

Abstract. *This paper presents an analysis of feelings in messages expressed by developers through commits in code repositories. In this work, six open source projects were analyzed, and a total of 12,113 commit messages. In the first phase, commit messages were classified into two categories (positive or negative). As a second phase, the paper investigates whether feelings are related to code refactoring activities, and concludes that when working with refactorings, the tendency is to express fewer negative feelings.*

Resumo. *Este trabalho apresenta um estudo a partir da análise de sentimentos em mensagens expressas pelos desenvolvedores por meio de commits em repositórios de código aberto. Foram analisados seis projetos de código aberto e um total de 12.113 mensagens de commit. Para o estudo foram classificadas as mensagens dos commits em duas categorias: positivo ou negativo. Em seguida, é realizada uma investigação se os sentimentos estão relacionados às atividades de refatoração de código. Como resultados deste estudo, tem-se indícios que quando se trabalha com refatorações a tendência é a expressar menos sentimentos negativos.*

1. Introdução

Métodos de análise de sentimentos foram inicialmente desenvolvidos para extrair a polaridade de sentimentos em textos curtos ou comentários em redes sociais onde existe grande interação do público, como em *tweets* [Thelwall et al. 2012]. Pesquisas recentes associam fatores humanos, como humor e emoções, com resolução de problemas [Graziotin et al. 2014], com linguagens de programação [Guzman et al. 2014] e com tarefas de refatoração de código [Singh and Singh 2017].

Estas pesquisas têm mostrado que as emoções afetam a qualidade do produto desenvolvido, a produtividade, a criatividade e a satisfação dos desenvolvedores [De Choudhury and Counts 2013]. Guzman *et al.* (2014) concluíram que *commits* feitos nas segundas-feiras e projetos desenvolvidos na linguagem de programação Java tendem a ter mais sentimentos negativos. Singh e Singh (2017) mostraram que, geralmente, os desenvolvedores expressam com mais frequência sentimentos negativos que positivos enquanto realizam a refatoração de código.

Neste trabalho, são analisados os sentimentos expressos pelos desenvolvedores em mensagens de *commits* em projetos de código fonte aberto. A questão a ser investigada neste trabalho é: “Quando se trabalha com refatoração de código, os sentimentos expressos tendem a ser positivos?”. A forma como este trabalho abordou essa questão difere da forma abordada em [Singh and Singh 2017], pois aqui foram analisados todos os *commits* que tiveram refatorações de código, e não apenas uma amostra.

Seis projetos de código fonte aberto foram utilizados neste trabalho. Para verificar o relacionamento dos dados de refatoração de código, de não refatoração e os sentimentos coletados, são utilizados os seguintes métodos estatísticos para validar os resultados: *Student's t-Test* [De Winter 2013] e *Wilcoxon Test* [Gehan 1965]. Como os *commits* com refatorações são disjuntos dos *commits* sem refatorações, os testes na tabela 4 foram feitos assumindo que as amostras são independentes (não pareadas). Foram encontradas evidências que mostram que quando se trabalha com tarefas de refatoração de código, os sentimentos negativos tendem a diminuir.

O restante do artigo é dividido como segue: a Seção 2 apresenta a metodologia para execução do estudo, indicando os projetos selecionados, como foi feita a análise dos sentimentos e a coleta de refatorações. A Seção 3 mostra os resultados encontrados e, por fim, a Seção 4 apresenta as conclusões e trabalhos futuros.

2. Metodologia do Estudo

Esta Seção apresenta a metodologia executada no estudo para investigação da relação das refatorações de código e sentimentos em projetos de código fonte aberto. É descrito o conjunto de dados analisados dos projetos extraídos do *GitHub*. Além disso, é apresentada como foram identificadas nos projetos as refatorações de código e realizada a análise dos sentimentos dos *commits* dos projetos.

2.1. Conjunto de Dados

Para realizar esse estudo, foram analisados projetos de código aberto codificados na linguagem Java extraídos do *GitHub*¹. Os projetos foram escolhidos com base na quantidade de *commits* que continham. Foram selecionados os primeiros seis projetos que continham entre 4400 e 7850 *commits* dos *top-projects* na linguagem Java no *GitHub*. Os projetos analisados foram: *Dropwizard*, *Guava*, *Kafka*, *Mockito*, *RxJava* e *Tutorials*. A quantidade total de *commits* em cada projeto é ilustrada na Tabela 1. No entanto, nem todos os *commits* coletados foram analisados, como é explicado a seguir.

Para coleta das refatorações realizadas em cada um dos projetos, foi utilizada a ferramenta *RefactoringMiner* [Tsantalis et al.]. Para que essa ferramenta pudesse analisar as refatorações de um *commit*, este não poderia ser um *commit* de *merge*, pois este tipo de *commit* geralmente não tem alterações no código, apenas a junção de dois ou mais outros *commits*.

2.2. Análise de Sentimentos

Para analisar os sentimentos das mensagens de *commit* foi utilizada a ferramenta *SentiStrength* [Thelwall et al. 2012], que permite a extração de sentimentos a partir de textos

¹<https://github.com/>

Projeto	Total de commits	Refatorações	
		Com	Sem
<i>dropwizard</i>	4482	101	951
<i>guava</i>	4676	346	1759
<i>kafka</i>	4871	392	3296
<i>mockito</i>	4653	516	1844
<i>RxJava</i>	5330	295	1542
<i>tutorials</i>	7817	147	924
Total	31829	1797	10316

Tabela 1. Quantidade total de *commits* por projeto

Score	Mensagem	Score Final
{4, -1}	Follow @fleaflicker's excellent[4]advice .	3
{3, -2}	Relax[3]the constraints[-2]on ConfiguredBundle .	1
{1, -5}	Definitely hating [-4][-1 booster word]#320 .	-4
{2, -2}	Correct default[-2]value[2] of rootPath	0

Tabela 2. Exemplos de mensagens de *commit*

curtos e de baixa qualidade. Esta ferramenta foi escolhida pela facilidade que tem de extrair sentimentos de textos curtos, e também porque já foi utilizada por estudos anteriores [Sinha et al. 2016, Singh and Singh 2017, Guzman et al. 2014].

SentiStrength utiliza um dicionário de *tokens*, que associa a cada *token* um *score*. Palavras com sentimentos negativos recebem uma pontuação entre -1 e -5, tal que -1 indica baixo sentimento negativo e -5 extremamente negativo. Já as palavras com sentimentos positivos recebem uma pontuação entre 1 e 5, sendo 1 baixo sentimento positivo e 5 extremamente positivo. Para analisar uma frase, a ferramenta atribui pontuações às palavras da frase que estão presentes no dicionário, e a pontuação final da frase é um par contendo o maior valor positivo e o maior valor absoluto negativo. Por exemplo, na frase “*I love you but I hate the current political climate*”, a ferramenta associará à palavra “*love*” uma pontuação igual a 3 e a “*hate*”, uma pontuação igual a -4. Logo, o *score* final da frase será {3, -4}. A Tabela 2 apresenta alguns *commits* dos projetos analisados e seus respectivos *scores* associados.

Para encontrar o sentimento final de uma mensagem do *commit*, foi realizada a soma da pontuação positiva e da pontuação negativa fornecida pela ferramenta *SentiStrength*, assim como é realizado em [Sinha et al. 2016]. Dessa forma, uma mensagem de *commit* pode ser classificada como: positiva quando o *score* final é maior que 0; negativa quando o *score* final é menor que 0; ou neutra quando o *score* final é igual a 0. Isso pode ser visto na Tabela 2. A coluna *Score final* mostra a pontuação final para cada exemplo.

Como a ferramenta SentiStrength é limitada ao seu dicionário e este deve ser específico para o contexto dos dados utilizados, isto pode ameaçar a qualidade da análise de sentimentos.

2.3. Refatorações

Segundo Fowler e Kent (1999), refatoração de código é uma forma de mudar um sistema de software, melhorando a estrutura interna de uma forma que não mude seu comportamento externo. Ainda segundo eles, atividades de refatoração também minimizam os riscos de introdução a *bugs*.

A coleta das refatorações realizada pela ferramenta *RefactoringMiner* detecta 11 tipos de atividades de refatoração: *Extract Method*, *Inline Method*, *Move*

Tabela 3. Sentimentos entre *commits*

Sentimento	Score final do Sentimento	Com Refatoração		Sem Refatoração	
		Número de <i>Commits</i>	Porcentagem do Sentimento	Número de <i>Commits</i>	Porcentagem do Sentimento
Negativo	-4	0	39.73%	2	51.98%
	-3	7		36	
	-2	119		781	
	-1	588		4543	
Neutro	0	358	19.92%	1587	15.38%
Positivo	1	702	40.35%	3204	32.64%
	2	23		158	
	3	0		4	
	4	0		1	
Total	-	1797	-	10316	-

Method/Attribute, *Pull Up Method/Attribute*, *Push Down Method/Attribute*, *Extract Superclass/Interface*, *Move Class*, *Rename Class*, *Rename Method*, *Extract and Move Method* e *Change Package*). Mais detalhes sobre essas atividades de refatoração podem ser encontrados em [Tsantalis et al.]. *RefactoringMiner* implementa uma versão do algoritmo UMLDiff [Xing and Stroulia 2005] para modelos orientados a objetos. Esse algoritmo é usado para inferir o conjunto de classes, métodos e campos adicionados, excluídos ou movidos entre revisões de código sucessivas. Depois de executar esse algoritmo, um conjunto de regras é usado para identificar diferentes tipos de refatoração. A ferramenta *RefactoringMiner* pode ser utilizada independente de sistema operacional ou IDE. [Tsantalis et al. 2013] identificou uma precisão de 96,4% no uso da ferramenta, e mostra que há uma taxa muito baixa de falsos positivos.

3. Resultados e Discussões

Nas seções seguintes, é apresentada uma análise dos resultados encontrados.

3.1. Análise de Sentimentos e Refatoração de Código

Para encontrar a relação entre os sentimentos e as refatorações de código, foram analisadas separadamente as mensagens de *commits* com e sem refatorações. A quantidade de *commits* analisada para cada um dos 6 projetos é mostrada na Tabela 1, e a porcentagem de *commits* encontrada para os sentimentos negativos, positivos e neutros é mostrada na Tabela 3.

De modo similar ao procedimento de Sinha et al. (2016), os *commits* com scores -1 e 1 (que resultariam em score final igual a 0) foram descartados. Desta forma, foi considerado como *commits* positivos aquele com score pelo menos 2, e *commits* negativos aqueles com score no máximo -2. Quando se trata de *commits* com refatorações, os sentimentos são balanceados entre negativos e positivos, com uma porcentagem de 39,7% e 40,4%, respectivamente. Já no restante dos *commits* (sem refatorações), o sentimento negativo domina, aparecendo em 52,0% dos *commits*, enquanto os sentimentos positivos apareciam em 32,6% das vezes. De fato, o Wilcoxon test [Gehan 1965] rejeitou a hipótese de que a distribuição dos scores dos *commits* sem refatoração é simétrica em relação ao zero ($p\text{-value} < 2,2e-16$), embora também rejeite esta simetria para os *commits* com refatoração ($p\text{-value} < 0,019$) se for levado em conta 95% de significância. Além disso,

Tabela 4. Testes por projetos nos *commits* com e sem Refatorações

<i>Projetos</i>	Com refatorações			Sem refatorações		
	Média	Wilcoxon signed-rank test (p-value)	Student's T-Test (p-value)	Média	Wilcoxon signed-Rank test (p-value)	Student's T-Test (p-value)
Dropwizard	-0.16	0.12	0.1	-0.33	<2.2e-16	<2.2e-16
Guava	-0.36	8.291e-10	2.918e-10	-0.38	<2.2e-16	<2.2e-16
Kafka	-0.47	2.231e-15	<2.2e-16	-0.6	<2.2e-16	<2.2e-16
Mockito	0.35	<2.2e-16	<2.2e-16	0.24	<2.2e-16	<2.2e-16
RxJava	0.23	1.639e-05	1.183e-05	0.04	0.13	0.13
Tutorials	-0.18	0.03	0.03	-0.24	1.98e-12	9.825e-13
Média	-0.06	0.02	0.02	-0.26	<2.2e-16	<2.2e-16

foi aplicado o *Student's t-Test* [De Winter 2013] para testar se o score médio dos *commits* positivos é igual ao score médio dos *commits* negativos, e com significância de 95% esta hipótese foi rejeitada nos dois caso (com e sem refatoração).

Concluiu-se que em ambos os casos, os sentimentos expressos pelos desenvolvedores tendem a ser negativos, porém quando se trabalha com atividades de refatoração, a tendência é de expressar menos sentimentos negativos do que quando não se trabalha com refatoração. Esses resultados diferem de Singh e Singh (2017), pois os autores encontraram apenas que quando se trabalha com refatorações de código, os desenvolvedores expressam mais sentimentos negativos que positivos, sem fazer uma comparação com o sentimento expresso nas atividades que não tiveram refatorações de código.

3.2. Análise de Sentimentos e Refatoração de Código por projeto

Os resultados encontrados para alguns projetos não foram os mesmos encontrado no geral. A Tabela 4 mostra os resultados de *p-value* encontrados nos testes e as médias para cada um dos projetos separadamente. Quando o *p-value* aparece em negrito, significa que a hipótese nula foi rejeitada (*p-value* < 0.05).

Em relação aos *commits* com refatorações, no projeto *Dropwizard* não foi possível rejeitar as hipóteses nulas em ambos os testes. Nos outros casos as hipóteses nulas foram refutadas, porém nos projetos *Mockito* e *RxJava*, as médias são positivas, o que indica que nesses casos, os *commits* quando há refatorações tendem a ter sentimentos positivos. Observando os dois projetos na Tabela 4 de *commits* sem refatorações, pode-se verificar que as médias continuam positivas, porém menores que suas respectivas médias quando há refatorações. Dessa forma, pode-se concluir que nesses dois projetos os sentimentos expressos em geral são positivos, porém quando se trabalha com atividades de refatoração, os sentimentos tendem a ser mais positivos ainda. Para os demais casos, os resultados acompanham o resultado encontrado no geral.

4. Conclusões

Este trabalho apresentou um estudo de análise de sentimentos em mensagens de *commit* produzidas por desenvolvedores em projetos Java de código fonte aberto e sua relação com refatorações do código. Assim como em Sinha et al. (2016), foram identificadas evidências que mostram que em geral, os sentimentos dos projetos tendem a ser negativos. No entanto, quando se trabalha com tarefas de refatorações, os *commits* tendem a

ser menos negativos. Também foram identificados casos de projetos onde a média dos sentimentos é positiva, essa média se torna ainda mais positiva quando se trabalhava com atividades de refatorações. Como trabalhos futuros, pretende-se coletar dados de mais projetos e relacionar os sentimentos de mensagens de *commit* com outros fatores, como a aparição de mau cheiro no código (ou do inglês, *bad smells*) [Fowler and Beck 1999].

Referências

- De Choudhury, M. and Counts, S. (2013). Understanding affect in the workplace via social media. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13*, pages 303–316, New York, NY, USA. ACM.
- De Winter, J. C. (2013). Using the student's t-test with extremely small sample sizes. *Practical Assessment, Research & Evaluation*, 18(10).
- Fowler, M. and Beck, K. (1999). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.
- Gehan, E. A. (1965). A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52(1-2):203–224.
- Graziotin, D., Wang, X., and Abrahamsson, P. (2014). Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ*, 2:e289.
- Guzman, E., Azócar, D., and Li, Y. (2014). Sentiment analysis of commit comments in github: An empirical study. In *Proceedings of the 11th Working Conference on Mining Software Repositories, MSR 2014*, pages 352–355, New York, NY, USA. ACM.
- Singh, N. and Singh, P. (2017). How do code refactoring activities impact software developers sentiments? – an empirical investigation into github commits. In *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, pages 648–653. IEEE.
- Sinha, V., Lazar, A., and Sharif, B. (2016). Analyzing developer sentiment in commit logs. In *Proceedings of the 13th International Conference on Mining Software Repositories*, pages 520–523. ACM.
- Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the Association for Information Science and Technology*, 63(1):163–173.
- Tsantalis, N., Guana, V., Stroulia, E., and Hindle, A. (2013). A multidimensional empirical study on refactoring activity. In *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research*, pages 132–146. IBM Corp.
- Tsantalis, N., Mansouri, M., Eshkevari, L., Mazinanian, D., and Dig, D. Accurate and efficient refactoring detection in commit history. In *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018*.
- Xing, Z. and Stroulia, E. (2005). UmlDiff: an algorithm for object-oriented design differencing. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, pages 54–65. ACM.