

Software Developers Sentiment Analysis: A Systematic Mapping

André A. N. Coelho, Thalita T. O. Silva, Alessandreia M. Oliveira e José Maria N. David
Universidade Federal de Juiz de Fora

Rua José Lourenço Kelmer, s/n - Martelos, Juiz de Fora, Minas Gerais, Brasil

Email: {andrearchanjo, thalita.thamires, alessandreia.oliveira}@ice.ufjf.br, jose.david@ufjf.edu.br

Abstract—Sentiment Analysis is a research area that has been also focused on software developers in the latest years. However, there is a need to map the way in which this area is structured. The goal of this article is to answer research questions through a systematic mapping in order to be aware of the existing works in this area. As a result, a growth curve related to the number of publications has been perceived as well as the association between the emotional condition of a developer and his/her performance in software projects.

Keywords—Sentiment Analysis, Emotions Analysis, Software Development.

I. INTRODUÇÃO

Projetos de software estão sujeitos à influência de fatores humanos, tais como sentimentos e emoções. Diante disso, pesquisas recentes têm buscado entender como esses fatores impactam o trabalho feito por desenvolvedores de software em seus respectivos projetos [1]. A Análise de Sentimentos, no contexto de desenvolvimento de software, tem por objetivo identificar e extrair de forma automática, opiniões, sentimentos e emoções, expressos nos projetos de software [2].

A maior parte das pesquisas com relação à Análise de Sentimentos começou a partir de 2004 [2] [3] [4] [5], mas suas raízes provêm de estudos de opinião pública datadas no século XX sobre a Segunda Guerra Mundial. Uma maior disponibilidade de textos subjetivos na Web colaborou com este tema de pesquisa. Posteriormente, outras áreas começaram a ser pesquisadas, entre elas, a Engenharia de Software [6]. O foco desses trabalhos foi em revisões de artefatos [7]. Tais pesquisas visam detectar os sentimentos de desenvolvedores, a fim de entender a influência de fatores humanos em seus respectivos projetos. Por exemplo, auxiliar os envolvidos, na detecção de alguém descontente ou com sentimento negativo. Como resultado, decisões podem ser tomadas para mudar a situação. Alguns resultados foram encontrados, e estão relacionados aos seguintes temas: ligações entre os sentimentos dos desenvolvedores e a quantidade de artefatos de software modificados, tempo de correção de *issues*, linguagem de programação e distribuição geográfica de equipes [1].

Com um maior número de pesquisas nos últimos anos, este artigo propõe a coleta de trabalhos que utilizam a Análise de Sentimentos no contexto de desenvolvimento de software. A partir disso, espera-se obter um conhecimento das pesquisas existentes até o momento, através de um mapeamento sistemático [8] [9].

Diante disso, este trabalho está organizado da seguinte forma: a Seção II apresenta a Fundamentação Teórica. A metodologia utilizada no mapeamento sistemático é apresentada na Seção III. A Seção IV apresenta os resultados e analisa as questões de pesquisa. A Seção V se refere às Ameaças à Validade. Por fim, as Considerações Finais são discutidas na Seção VI.

II. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os conceitos fundamentais relacionados à Análise de Sentimentos, bem como as abordagens de classificação frequentemente utilizadas na literatura.

A. Análise de Sentimentos

A Análise de Sentimentos tem por objetivo determinar a polaridade de uma expressão, como positiva ou negativa, e no caso de não detectar sentimentos considerá-la neutra (ausência de sentimento) [10]. Também é considerada uma área de estudos de Processamento de Linguagem Natural com o objetivo de encontrar elementos subjetivos no texto [11].

A partir da polaridade é possível detectar o sentimento, mas não a emoção relacionada à expressão. Outra maneira de realizar a Análise de Sentimentos é considerar um conjunto de emoções básicas, tais como: *anger* (raiva), *disgust* (desgosto), *fear* (medo), *joy* (alegria), *sadness* (tristeza) e *surprise* (surpresa) [12].

Existem duas abordagens para a classificação de um trecho de um texto, como descrito a seguir: Método Lexical (*Lexical Method*) e suas variações [11], e Aprendizado de Máquina (*Machine Learning*) [13].

B. Método Lexical

O Método Lexical consiste na preparação de um dicionário com o propósito de armazenar a polaridade do *lexicon* (vocabulário). Ao calcular a polaridade em um texto, verifica-se palavra por palavra a sua existência no dicionário. Caso exista, é realizado um acréscimo no valor total da polaridade. Pela pontuação final é feita a classificação da polaridade do texto, ou seja, caso o resultado seja maior que 0, a polaridade é positiva, e caso contrário, negativa [11]. A Tabela I apresenta um dicionário contendo o vocabulário e suas respectivas polaridades. Considere a seguinte frase como exemplo:

With a delay[-1], comes pressure[-3]. Although with enough inspiring[+1] we can do a good work[+2].

TABLE I
EXEMPLO DO MÉTODO LEXICAL

#	Vocabulário	Polaridade
1	<i>hate</i>	-4
2	<i>good</i>	+2
3	<i>delay</i>	-1
4	<i>inspiring</i>	+1
5	<i>happy</i>	+4
6	<i>pressure</i>	-3

Neste caso, a pontuação total seria: $-1 - 3 + 1 + 2 = -1$, ou seja, polaridade negativa.

É possível notar neste exemplo que a palavra *delay* é considerada como pouco negativa por ter o valor -1, enquanto *pressure* é mais negativa com valor -3. As outras palavras, *inspiring* e *good*, são consideradas pouco positivas, tendo como valores respectivos +1 e +2. O resultado final é a soma de todos os valores encontrados das palavras presentes no dicionário, retornando no final deste exemplo -1. Como resultado, a sentença é marcada como negativa.

Algumas complementações para o Método Lexical demonstradas em [11] são descritas a seguir:

- *Baseline Approach*: o dicionário está limitado a um número fixo de palavras marcadas positivamente e negativamente;
- *WordNet*: é uma base de dados da língua inglesa. Nesta base, as palavras são agrupadas em sinônimos. Tais grupos são classificados como advérbios, pronomes, verbos e adjetivos. Como resultado, fornecem uma definição geral e representam relações semânticas entre as palavras.

Além dessas complementações também existem algumas técnicas de pré-processamento, tais como:

- *Stemming* [14]: busca retirar os prefixos e sufixos da palavra antes de analisá-la;
- *Part of Speech Tagging* [15]: também conhecido como *POST* ou *POS Tagging*. É o processo de relacionar uma palavra com uma parte do texto, baseando-se no contexto inserido e na sua definição;
- *N-grams*: ao invés de levar em conta somente uma palavra por vez em uma frase, considera-se uma sequência de palavras N para encontrar a polaridade;
- *Conjunction Rules*: geralmente uma sentença expressa apenas um sentimento. Porém, quando encontra-se uma conjunção na frase, o sentimento expressado anteriormente pode ser alterado. Um exemplo para esta situação seria: *Apesar de a lanterna ser bonita, infelizmente ela é muito frágil*. Neste exemplo, é possível perceber que o principal sentimento passado sobre a lanterna é negativo. Mesmo que o texto tenha sido iniciado por uma sentença positiva, tal como, “lanterna ser bonita”, a conjunção “apesar” foi fundamental para ponderar a sentença;
- *Stop Words*: representam palavras sem valor semântico, como “eu” ou “e”. São retiradas do texto por não

proverem nenhuma ou pouca informação sobre o sentimento;

- *Negation Method*: nesta técnica adiciona-se uma marcação “NOT” em um adjetivo quando ele possui uma negação, como por exemplo “not good” fica “NOTgood”.

Alguns autores, como [11], utilizam as técnicas de pré-processamento *Negation Method*, *Stop Words* e *Stemming* a fim de melhorar os resultados a serem encontrados na Análise de Sentimentos.

O Aprendizado de Máquina é um campo da Ciência da Computação que evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em Inteligência Artificial [13]. Vários algoritmos de aprendizado de máquina como: *Maximum Entropy* (ME), *Naive Bayes* (NB) e *Support Vector Machines* (SVM) são usualmente utilizados para classificação de textos [16].

Algumas etapas realizadas para a Análise de Sentimentos usando Aprendizado de Máquina são apresentadas na Figura 1. Inicia-se com uma coleção de dados extraídos. Esses dados são pré-processados utilizando várias técnicas do Processamento de Linguagem Natural (NLP). Os conteúdos relevantes para a Análise de Sentimentos precisam ser extraídos e, finalmente, classificados pela técnica.

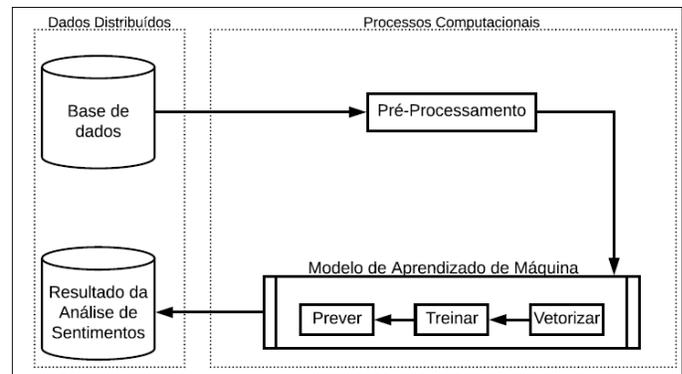


Fig. 1. Exemplo de um processo de Análise de Sentimentos utilizando Aprendizado de Máquina. (Adaptado de [16])

III. MAPEAMENTO SISTEMÁTICO DA LITERATURA

Um mapeamento sistemático é uma forma de identificar, avaliar e interpretar trabalhos disponíveis relevantes para questões de pesquisa particulares. Uma das razões para a realização de mapeamentos sistemáticos é que eles resumem as evidências existentes em relação a um tratamento ou tecnologia [17].

Este mapeamento foi conduzido de acordo com as etapas descritas a seguir. Para apoiar este processo foi utilizada a ferramenta Parsif.al¹.

A. Planejamento

Esta fase consiste em especificar e avaliar o protocolo a ser seguido. O protocolo apresenta o objetivo do mapeamento, as

¹Ferramenta online projetada para apoiar os pesquisadores a realizar revisões sistemáticas de literatura (<https://parsif.al/>)

questões de pesquisa, a *string* de busca, a definição das fontes e os critérios de inclusão e exclusão dos trabalhos obtidos.

O objetivo do trabalho é obter uma visão geral das técnicas, métodos, modelos e ferramentas relacionados à Análise de Sentimentos aplicados a desenvolvedores em um ambiente de desenvolvimento de software. Com base no objetivo, foram elaboradas 7 questões de pesquisa apresentadas a seguir:

QP1: Quais itens de configuração são extraídos para a Análise de Sentimentos de desenvolvedores de software?

Através dessa questão pretende-se identificar quais itens de configuração do ciclo de vida do desenvolvimento de software, como *commits*, *issues*, e-mail, entre outros, são selecionados para a extração e posterior Análise dos Sentimentos contidos nos mesmos.

QP2: Quais são as técnicas utilizadas para extração dos itens de configuração na Análise de Sentimentos de desenvolvedores de software?

Nessa questão busca-se identificar como é feita a extração dos itens de configuração e quais são as técnicas utilizadas para a extração destes.

QP3: Quais são as ferramentas utilizadas para a Análise de Sentimentos de desenvolvedores de software?

Com esta questão, pretende-se descobrir quais são as ferramentas mais utilizadas para apoiar a Análise dos Sentimentos contidos nos itens de configuração extraídos.

QP4: Como a Análise de Sentimentos de desenvolvedores de software está sendo aplicada na Engenharia de Software?

Através desta questão pretende-se descobrir qual o objetivo da análise do item de configuração selecionado bem como quais os itens que se relacionam com os sentimentos, como por exemplo, produtividade, desistência, erros adicionados nos códigos ou outros fatores.

QP5: Qual a classificação de sentimentos é mais usada?

Com essa questão busca-se identificar as categorias de classificação mais utilizada nos artigos (positivo e negativo ou positivo, neutro e negativo ou outro).

QP6: Quais tipos de emoções têm sido mais usadas?

Nessa questão busca-se identificar quais emoções são consideradas ao realizar as Análises de Sentimento de desenvolvedores de software.

QP7: Qual abordagem tem sido mais utilizada no contexto de Análise de Sentimentos de desenvolvedores de software?

Através desta questão busca-se identificar quais abordagens de Análise de Sentimentos (Método Lexical, Aprendizado de Máquina e híbrida) têm sido mais empregadas no contexto de desenvolvedores de software.

Para definir a expressão de busca a ser utilizada nas bibliotecas digitais foi realizado um processo de teste e refinamento,

baseado no método PICO [18]. O acrônimo PICO é usado para identificar as quatro partes críticas da expressão de pesquisa, que são: (*Population* - População (P), *Intervention* - Intervenção (I), *Comparison* - Comparação (C), *Outcome* - Resultado (O) [19]. A atividade de revisão da *string* foi realizada por especialistas. A especificação do método PICO está presente na Tabela II.

TABLE II
PICO

PICO	Termos	Descrição
P	“ <i>software developer</i> ” OR “ <i>developer</i> ” OR “ <i>development teams</i> ” OR “ <i>programmer</i> ” OR “ <i>implementer</i> ”	desenvolvedores de software
I	“ <i>sentiment analysis</i> ” OR “ <i>emotion analysis</i> ”	Análise de Sentimentos
C		Não se aplica
O	“ <i>tool</i> ” OR “ <i>technique</i> ” OR “ <i>algorithm</i> ” OR “ <i>method</i> ” OR “ <i>model</i> ” OR “ <i>approach</i> ” OR “ <i>application</i> ”	Ferramentas, técnicas, abordagens, aplicações, métodos, algoritmos e modelos

As palavras-chave foram combinadas usando o operador OR dentro de cada um desses três elementos da estrutura PICO utilizados no mapeamento. Como a comparação não é aplicável a este estudo, ela foi ignorada. Como resultado, tem-se a *string* de busca apresentada na Tabela III.

TABLE III
STRING DE BUSCA GERAL

(“*software developer*” OR “*developer*” OR “*development teams*” OR “*programmer*” OR “*implementer*”) AND (“*sentiment analysis*” OR “*emotion analysis*”) AND (“*tool*” OR “*technique*” OR “*algorithm*” OR “*method*” OR “*model*” OR “*approach*” OR “*application*”)

Foram selecionadas quatro bibliotecas digitais para efetuar a busca, como mostra a Tabela IV.

TABLE IV
BIBLIOTECAS DIGITAIS UTILIZADAS NO MAPEAMENTO SISTEMÁTICO

Bibliotecas Digitais	Endereços Eletrônicos
IEEE Xplore Digital Library	http://ieeexplore.ieee.org
Enginnering Village	http://www.engineeringvillage.com
Scopus	http://www.scopus.com
Science Direct	http://www.sciencedirect.com

Por conta de particularidades de cada biblioteca digital, a *string* de busca geral foi adaptada a cada uma delas, sofrendo alterações, sendo mostradas na Tabela V.

Um conjunto de critérios de inclusão e exclusão foi especificado para selecionar os estudos. Esses critérios foram

TABLE V
STRING DE BUSCA ESPECÍFICA PARA CADA UMA DAS BIBLIOTECAS DIGITAIS UTILIZADAS

Biblioteca Digital	String de Busca
IEEE Xplore	<i>("software developer" OR "developer" OR "development teams" OR "programmer" OR "implementer") AND ("sentiment Analysis" OR "emotion analysis") AND ("tool" OR "technique" OR "algorithm" OR "method" OR "model" OR "approach" OR "application")</i>
Engineering Village	<i>("software developer" OR "developer" OR "development teams" OR "programmer" OR "implementer") AND ("sentiment Analysis" OR "emotion analysis") AND ("tool" OR "technique" OR "algorithm" OR "method" OR "model" OR "approach" OR "application")</i>
Scopus	<i>TITLE-ABS-KEY (("software developer" OR "developer" OR "development teams" OR "programmer" OR "implementer") AND ("sentiment Analysis" OR "emotion analysis") AND ("tool" OR "technique" OR "algorithm" OR "method" OR "model" OR "approach" OR "application"))</i>
Science Direct	<i>(((((("software developer" OR "developer" OR "development teams" OR "programmer" OR "implementer") AND ("sentiment Analysis" OR "emotion analysis") AND ("tool" OR "technique" OR "algorithm" OR "method" OR "model" OR "approach" OR "application"))))wn KY and (72* wn CL)))</i>

cadastrados na Parsif.al. Esta ferramenta importa padrões BibTex dos artigos retornados nas bibliotecas digitais, facilitando a visualização dos artigos e seus determinados atributos. Além disso, também possibilita cadastrar os critérios de inclusão e exclusão, e no momento de realizar a seleção dos estudos, eleger um dos critérios e um status (*Accepted, Rejected, Unclassified ou Duplicated*) para identificar cada estudo.

Foi necessário também descrever os critérios de exclusão [20], com o objetivo de verificar se um artigo é um potencial candidato a ser selecionado ou a ser excluído do mapeamento sistemático. Neste trabalho, os critérios de exclusão utilizados são apresentados na Tabela VI. Como critérios de inclusão, foram considerados os estudos que atendem a negação dos critérios de exclusão.

TABLE VI
CRITÉRIOS DE EXCLUSÃO

C1	Trabalhos que não apresentam ferramentas ou algoritmos ou abordagens ou aplicações ou modelos ou métodos ou técnicas de Análise de Sentimentos de desenvolvedores de software
C2	Trabalhos que o acesso ao texto completo não está disponível
C3	Trabalhos que são relacionadas a palestras, tutoriais e/ou cursos
C4	Trabalhos que tenham versões mais recentes

A avaliação da *string* foi feita através da definição de três artigos de controle, listados na Tabela VII, e de sua verificação no retorno nas buscas. Uma revisão da literatura convencional anterior obteve tais artigos de controle. Eles foram úteis para

fornecer uma compreensão inicial da área bem como para definir palavras-chave de busca.

TABLE VII
ARTIGOS DE CONTROLE

#	Ref	Artigo
1	[2]	E. Guzman and B. Bruegge. Towards emotional awareness in software development teams. In Proceedings of the 2013 9th joint meeting on foundations of software engineering, pages 671–674. ACM, 2013.
2	[12]	F. Jurado and P. Rodriguez. Sentiment analysis in monitoring software development processes: An exploratory case study on github project issues. Journal of Systems and Software, 104:82–89, 2015.
3	[21]	M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, and B. Adams. The emotional side of software developers in jira. In Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on, pages 480–483. IEEE, 2016.

O trabalho apresentado em [2] aplica a Análise de Sentimentos para melhorar a consciência emocional em equipes de desenvolvimento distribuídas. A abordagem proposta pelos pesquisadores extrai e resumiu as emoções expressas em artefatos de colaboração, e posteriormente exibiu os resumos aos líderes dos projetos analisados. Jurado e Rodriguez [12], os pesquisadores propuseram técnicas de Análise Sentimentos para identificar e monitorar os sentimentos contidos no texto escrito por desenvolvedores. Foi realizado um estudo de caso com objetivo de identificar a polaridade e encontrar pistas emocionais em problemas de desenvolvimento de 9 projetos. Por último, o trabalho descrito em [21] propôs uma classificação manual de 2.000 comentários de problemas e 4.000 frases escritas por desenvolvedores com emoções como amor, alegria, surpresa, raiva, tristeza e medo. O conjunto classificado foi vinculado a artefatos de software relatados em um conjunto de dados previamente publicado. O conjunto de dados enriquecido apresentado em [21], permite posteriores investigações do papel dos artefatos no desenvolvimento de software e sua relação com as emoções ali contidas.

B. Condução do Mapeamento

As buscas utilizando as *strings* adaptadas nas bibliotecas digitais foram realizadas na primeira quinzena do mês de janeiro de 2018. É importante ressaltar que todas as etapas do mapeamento foram feitas por duas pessoas, com objetivo de validar as seleções feitas.

Foram feitas buscas em 4 bibliotecas digitais usando as *strings* de busca adaptadas, citadas anteriormente, considerando apenas os campos: título do artigo, resumo e palavras-chave. Caso a biblioteca digital tivesse a opção de determinar a área na qual os artigos se enquadravam, esta opção também foi levada em consideração. A quantidade de artigos obtidos foi 218. A Tabela VIII sumariza a quantidade de artigos retornados por biblioteca digital.

Para realizar o tratamento de duplicidade entre artigos em diferentes bibliotecas digitais, foi adotada a seguinte regra: quando constatada a duplicidade de determinado artigo,

TABLE VIII
TOTAL DE ARTIGOS RETORNADOS PARA CADA UMA DAS BIBLIOTECAS DIGITAIS A PARTIR DA BUSCA ESPECÍFICA

Bibliotecas Digitais	Artigos Retornados
IEEE Xplore	66
EI COMPENDEX	27
Science Direct	54
Scopus	71

escolheu-se uma biblioteca digital para o artigo permanecer. Em relação às demais bibliotecas digitais o artigo foi eliminado. Essa escolha de qual biblioteca digital o artigo seria excluído foi realizada levando em conta o número de artigos que cada biblioteca digital retornou. O artigo era excluído, prioritariamente, das bibliotecas digitais que retornaram um maior número de artigos. Os artigos seriam eliminados caso estivessem duplicados da biblioteca digital Scopus, IEEE Digital Library, ScienceDirect e EI Compendex nesta ordem de prioridade. Após a realização desta regra foi obtido um total de 169 artigos, como mostra a Tabela IX.

TABLE IX
TOTAL DE ARTIGOS RETORNADOS PARA CADA UMA DAS BIBLIOTECAS DIGITAIS A PARTIR DA BUSCA ESPECÍFICA APÓS TRATAMENTO DE DUPLICIDADE

Bibliotecas Digitais	Artigos Retornados
IEEE Xplore	55
EI COMPENDEX	27
Science Direct	54
Scopus	33

A partir desse ponto, foi executada uma primeira seleção, que consistiu em aplicar, manualmente, os critérios de exclusão e inclusão no título, resumo e palavras-chave dos trabalhos. Nesta etapa foram selecionados 36 trabalhos. Foi realizada uma segunda avaliação dos artigos pelos critérios de seleção agora aplicados no resumo, introdução e conclusão. Como resultado, foram obtidos 23 artigos. Por fim, uma terceira e última avaliação dos artigos foi feita, aplicando os critérios em seu conteúdo, resultando em 18 artigos aceitos, conforme apresentados na Tabela X. Nela, as colunas **#**, **Ref**, **Artigo**, **Bibliotecas Digitais** referem-se, respectivamente, à sequência numérica da tabela, à referência encontrada neste artigo, o nome do artigo e a(s) biblioteca(s) digital(is) onde foi encontrado.

A distribuição de artigos por bibliotecas digitais está ilustrada na Figura 2. Nela se percebe que a maioria foi proveniente da biblioteca digital EI COMPENDEX, e que não houve artigos do Science Direct aceitos perante os critérios.

A Figura 3 resume as fases de seleção dos artigos do mapeamento com o total obtido em cada fase.

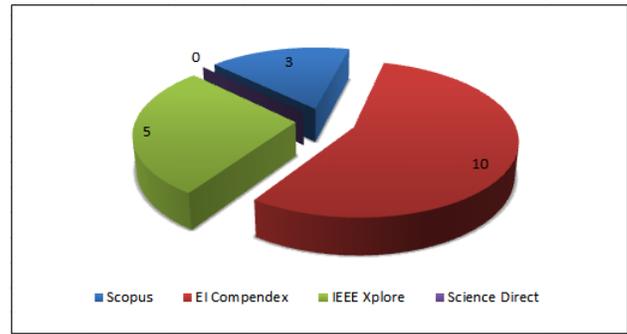


Fig. 2. Total de artigos selecionados em cada uma das bibliotecas digitais após a aplicação dos critérios de exclusão

IV. RESULTADOS

A partir do conjunto final retornado da *string* de busca, teve-se como resultado 18 artigos publicados em um intervalo de 5 anos. É possível constatar um crescimento de publicações sobre o tema a partir do ano de 2013, como pode ser observado na Figura 4. Analisando os mesmos, foi possível compreender a Análise de Sentimentos de desenvolvedores de software, com o objetivo de responder as questões de pesquisa elaboradas para este mapeamento.

A primeira questão de pesquisa (**QP1**) pretende descobrir quais itens dos desenvolvedores de software são extraídos para posterior análise. Constatou-se que dos 18 artigos, 8 artigos escolheram como item *issues* ou logs de *commits* a serem extraídos de um repositório. O GitHub foi o repositório mais utilizado para a extração dos projetos e itens, sendo usado em 6 artigos. Provavelmente, isso se deve ao fato de o GitHub possuir uma API REST que permite a extração destes dados [12], além de ser o maior *host* de códigos do mundo, com milhões de programadores colaborando em mais de 10 milhões de repositórios [5]. Outro item buscado em grande quantidade foi as questões e respostas do StackOverflow², sendo usado por 5 artigos. Isso se deve ao fato de muitos programadores o utilizarem para externarem dúvidas. Para tanto, fazem assim perguntas para as quais esperam respostas gerando um grande número de dados.

A segunda questão de pesquisa (**QP2**) pretende encontrar os métodos utilizados para a extração dos itens da questão anterior. Os artigos tendem a utilizar os seguintes métodos:

- 1) API's para a extração dos seus dados, como por exemplo a API REST do GitHub
- 2) Boa [33], uma infraestrutura com um domínio de linguagem específico
- 3) API's públicas a fim de facilitar a mineração de repositórios de softwares

Uma forma alternativa é utilizar um projeto privado, com a devida permissão do gestor, criando um software para a extração dos itens necessários [28].

²O Stack Overflow é um site americano gratuito de perguntas e respostas sobre desenvolvimento de software. (<https://stackoverflow.com/>)

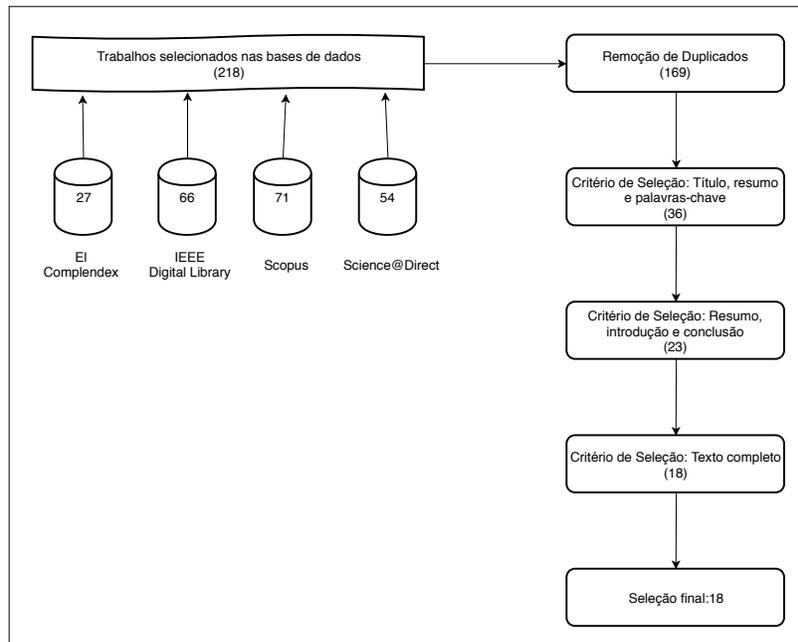


Fig. 3. Fases da seleção dos artigos do mapeamento

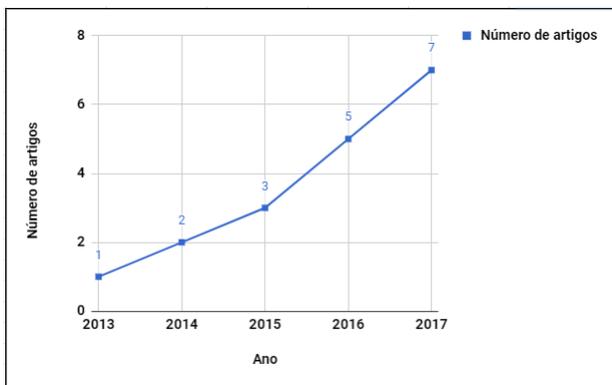


Fig. 4. Total de artigos publicados por ano

A terceira questão de pesquisa (**QP3**) pretende descobrir quais ferramentas são utilizadas para reconhecer os sentimentos nos itens extraídos. Para a criação da Figura 5, considerouse apenas os artigos que utilizaram ou implementaram alguma ferramenta, além de ser possível utilizar mais de uma em um único trabalho. Com ela, é possível perceber que a ferramenta mais utilizada e a mais bem avaliada é a SentiStrength [34], uma ferramenta de mineração de opinião e Análise de Sentimentos que realiza uma análise nos textos. Utiliza-se como base para classificação mais de 16.000 textos previamente computados e classificados. Em seguida, temos a ferramenta NLTK [35] (Natural Language Toolkit), uma vasta biblioteca implementada na linguagem de programação Python [36], constituindo um conjunto de ferramentas para a análise automática de textos nos mais diferentes níveis. Porém, um problema de ambas as ferramentas é que elas foram treinadas com textos

fora do contexto de Engenharia de Software. Segundo os trabalhos de Jongeling et al. [6] [3], ambas as ferramentas não apresentam resultados satisfatórios por não concordarem entre si, e também com a classificação manual feita por pessoas, por apresentarem resultados muito distintos. Com isso, alguns trabalhos procuraram criar uma nova ferramenta [26] [27] [29], ou até mesmo classificar manualmente os seus textos [24].

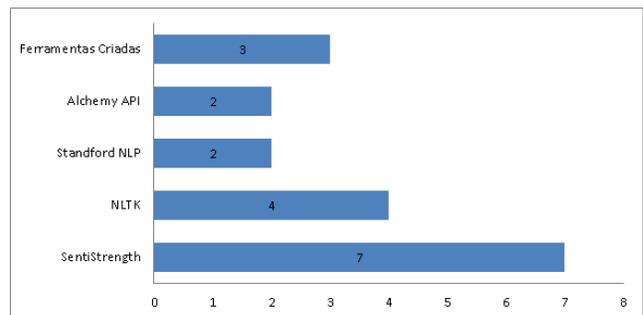


Fig. 5. Total de utilizações das ferramentas de Análise de Sentimentos

A quarta questão de pesquisa (**QP4**) pretende descobrir o motivo da Análise de Sentimentos ser utilizada no contexto proposto pela publicação. Os motivos foram bem variados, mas todos tentavam relacionar o estado emocional do desenvolvedor com o projeto. Houve estudos que tinham objetivos mais gerais. Eles buscavam saber qual emoção ocorria mais rotineiramente, buscavam descobrir a relação do desenvolvedor com os dias da semana, e se o número de mudanças feitas afeta o desenvolvedor. Alguns estudos tinham um objetivo mais específico, como por exemplo: avaliar o desempenho das ferramentas de Análise de Sentimentos verificando o impacto de se escolher uma [6] [3], construir uma ferramenta

TABLE X
LISTAGEM FINAL DOS ARTIGOS SELECIONADOS

#	Ref	Artigo	Bibliotecas Digitais
1	[2]	E. Guzman and B. Bruegge. Towards emotional awareness in software development teams. In Proceedings of the 2013 9th joint meeting on foundations of software engineering, pages 671–674. ACM, 2013.	EI COMPENDEX e Scopus
2	[5]	D. Pletea, B. Vasilescu, and A. Serebrenik. Security and emotion: sentiment analysis of security discussions on github. In Proceedings of the 11th working conference on mining software repositories, pages 348–351. ACM, 2014.	Scopus
3	[22]	W. N. Robinson, T. Deng, and Z. Qi. Developer behavior and sentiment from data mining open source repositories. In System Sciences (HICSS), 2016 49th Hawaii International Conference on, pages 3729–3738. IEEE, 2016.	EI COMPENDEX, IEEE Xplore e Scopus
4	[23]	M. V. Mantyla, N. Novielli, F. Lanubile, M. Claes, and M. Kuutila. Bootstrapping a lexicon for emotional arousal in software engineering. In Mining Software Repositories (MSR), 2017 IEEE/ACM 14th International Conference on, pages 198–202. IEEE, 2017.	EI COMPENDEX, IEEE Xplore e Scopus
5	[24]	M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, and B. Adams. The emotional side of software developers in jira. In Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on, pages 480–483. IEEE, 2016.	IEEE Xplore
6	[1]	R. Souza and B. Silva. Sentiment analysis of travis ci builds. In Proceedings of the 14th International Conference on Mining Software Repositories, pages 459–462. IEEE Press, 2017.	IEEE Xplore e Scopus
7	[25]	V. Sinha, A. Lazar, and B. Sharif. Analyzing developer sentiment in commit logs. In Proceedings of the 13th International Conference on Mining Software Repositories, pages 520–523. ACM, 2016	EI COMPENDEX e Scopus
8	[26]	T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi. Senticr: A customized sentiment analysis tool for code review interactions. In Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, pages 106–111. IEEE Press, 2017.	IEEE Xplore
9	[6]	R. Jongeling, S. Datta, and A. Serebrenik, “Choosing your weapons: On sentiment analysis tools for software engineering research,” in 2015 IEEE International Conference on Software Maintenance and Evolution(ICSME). IEEE, 2015, pp. 531–535.	EI COMPENDEX, IEEE Xplore e Scopus
10	[3]	R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik. On negative results when using sentiment analysis tools for software engineering research. Empirical Software Engineering, 22(5):2543–2584, 2017.	Scopus
11	[27]	A. Goyal and N. Sardana. Nrfixer: Sentiment based model for predicting the fixability of non-reproducible bugs. e-Infomatica Software Engineering Journal, 11(1), 2017.	Scopus e IEEE Xplore
12	[28]	A. Patwardhan. Sentiment identification for collaborative, geographically dispersed, cross-functional software development teams. In Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on, pages 20–26. IEEE, 2017.	IEEE Xplore
13	[4]	M. R. Islam and M. F. Zibran. Towards understanding and exploiting developers’ emotional variations in software engineering. In Software Engineering Research, Management and Applications (SERA), 2016 IEEE 14th International Conference on, pages 185–192. IEEE, 2016.	IEEE Xplore
14	[29]	F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli. Sentiment polarity detection for software development. Empirical Software Engineering, pages 1–31, 2017.	EI COMPENDEX e Scopus
15	[30]	N. Novielli, F. Calefato, and F. Lanubile. The challenges of sentiment detection in the social programmer ecosystem. In Proceedings of the 7th International Workshop on Social Software Engineering, pages 33–40. ACM, 2015.	EI COMPENDEX e Scopus
16	[12]	F. Jurado and P. Rodriguez. Sentiment analysis in monitoring software development processes: An exploratory case study on github project issues. Journal of Systems and Software, 104:82–89, 2015.	EI COMPENDEX e Scopus
17	[31]	A. K. M. M. M. Rahman and C. K. Roy. Embedded emotion-based classification of stack overflow questions towards the question quality prediction. 2016.	EI COMPENDEX e Scopus
18	[32]	Y. Zhang, B. Shen, and Y. Chen. Mining developer mailing list to predict software defects. In Software Engineering Conference (APSEC), 2014 21st Asia-Pacific, volume 1, pages 383–390. IEEE, 2014.	EI COMPENDEX e IEEE Xplore

de Análise de Sentimentos designada especificamente para comentários de códigos [26].

A questão (QP5) pretende descobrir quais as classificações de sentimentos mais usadas. Através desta pergunta foi constatado, como mostra a Figura 6, que a escala mais usada é *positivo*, *neutro* e *negativo*, representando 54,5% dos artigos selecionados. A segunda escala mais usada é *positivo* e *negativo*, representando 40,5% dos artigos. Ainda houve uma terceira escala pouco explorada: *muito negativo*, *negativo*, *neutro*, *positivo* e *muito positivo*, que é utilizada pela

ferramenta *Stanford NLP*. Esta ferramenta foi pouco utilizada por ter resultados inferiores a outras ferramentas de Análise de Sentimentos no contexto de Desenvolvimento de Software [6]. Ela apresenta dificuldade em realizar a análise sobre trechos de códigos e URLs.

A sexta questão de pesquisa (QP6) pretende descobrir quais são as emoções consideradas ao realizar a Análise de Sentimentos. A minoria dos estudos considera as emoções dos desenvolvedores, a maioria utiliza apenas a classificação de sentimentos positivos e sentimentos negativos. Nessa minoria,



Fig. 6. Escalas de sentimentos usadas nos trabalhos selecionados a partir do mapeamento

não há um consenso sobre o conjunto de emoções que deve ser utilizado para esse contexto. Foram usados 7 conjuntos diferentes de emoções, mostrados na Tabela XI. Os pesquisadores ainda dividiram esses conjuntos em subconjuntos de emoções positivas e emoções negativas. A respeito dessa subdivisão há um impasse sobre a classificação da emoção *surprise* que pode ser considerada como positiva e negativa.

TABLE XI
CONJUNTOS DE DIFERENTES EMOÇÕES

#	Emoções
1°	<i>joy, love, surprise, anger, fear</i>
2°	<i>love, joy, sadness</i>
3°	<i>love, joy, surprise, anger, fear and sadness</i>
4°	<i>love, joy, anger and sadness</i>
5°	<i>anger, anticipation, disgust, fear, joy, sadness, surprise, and trust</i>
6°	<i>anger, surprise and happiness</i>
7°	<i>anger, disgust, fear, joy, sadness, and surprise</i>

A sétima questão de pesquisa (QP7) pretende descobrir qual abordagem tem sido mais usada para realizar a Análise de Sentimentos de desenvolvedores de software. Há três abordagens utilizadas para a Análise de Sentimentos: lexical, Aprendizado de Máquina e a combinação das duas, ou seja, híbrida. No contexto de desenvolvedores de softwares a mais utilizada é a abordagem lexical. Isto ocorre porque a ferramenta mais utilizada nos estudos é a SentiStrength [34], uma ferramenta que utiliza a abordagem lexical que pontua o texto de duas formas, uma positiva de 1 a 5 e uma negativa de -1 a -5.

Além de responder as questões de pesquisas, procurou-se identificar os autores que publicaram na área de Análise de Sentimentos em Engenharia de Software e as suas relações de pesquisa. Na Figura 7 os trabalhos mais citados estão destacados em cinza, Guzman et al [2], Pletea et al [5], Jongeling et al [6] e [30], citados 9, 8, 7 e 5 vezes respectivamente, como mostra a Tabela XII. Isso mostra possível influência destes trabalhos sobre os demais. Vale ressaltar que [2] é o primeiro a abordar o tema de Análise de Sentimentos de desenvolvedores

TABLE XII
ARTIGOS MAIS CITADOS

#	Ref	Artigo
1°	[2]	E. Guzman and B. Bruegge. Towards emotional awareness in software development teams. In Proceedings of the 2013 9th joint meeting on foundations of software engineering, pages 671–674. ACM, 2013.
2°	[5]	D. Pletea, B. Vasilescu, and A. Serebrenik. Security and emotion: sentiment analysis of security discussions on github. In Proceedings of the 11th working conference on mining software repositories, pages 348–351. ACM, 2014.
3°	[6]	R. Jongeling, S. Datta, and A. Serebrenik, “Choosing your weapons: On sentiment analysis tools for software engineering research,” in 2015 IEEE International Conference on Software Maintenance and Evolution(ICSME). IEEE, 2015, pp. 531–535.
4°	[30]	N. Novielli, F. Calefato, and F. Lanubile. The challenges of sentiment detection in the social programmer ecosystem. In Proceedings of the 7th International Workshop on Social Software Engineering, pages 33–40. ACM, 2015.

de software, se tornando o mais citado.

Ainda, observando a Figura 7, outro dado encontrado é que os pesquisadores Alexander Serebrenik, Nicole Novielli e Filippo Lanubile possuem três publicações na área, sendo que os dois últimos trabalharam em conjunto em todas as oportunidades, enquanto A. Serebrenik trabalhou 2 vezes com Robert Jongeling e Subhajt Datta.

V. AMEAÇAS À VALIDADE

Durante o planejamento deste estudo, buscou-se evitar ameaças que pudessem impactar ou limitar a validade dos resultados obtidos [37]. No entanto, não é possível garantir que tais ameaças não tenham afetado os resultados. Desta forma, as ameaças identificadas no contexto deste estudo são descritas a seguir.

As etapas de seleção dos estudos e extração de dados foram realizadas por dois autores, com o intuito de diminuir uma possível subjetividade. Os resultados encontrados por cada um eram comparados para detectar um possível viés, e quando havia um desacordo, os autores conversavam e expunham seus argumentos, a fim de se chegar a um consenso.

Problemas terminológicos na *string* de busca podem ter levado à ausência de alguns estudos primários. Para minimizar estes problemas, a *string* de busca foi testada e os resultados encontrados foram apresentados a um grupo de estudos para análise. Com o auxílio recebido, incrementos à *string* de busca eram realizados e, por fim, chegou-se ao resultado final.

Considerar apenas 4 bibliotecas digitais, apesar dessas fontes de dados serem referenciadas na literatura, pode significar não indexar todo o conteúdo disponível na Web. Por este motivo é possível que alguns trabalhos relevantes não tenham sido contemplados por este mapeamento.

VI. CONSIDERAÇÕES FINAIS

Este estudo buscou analisar os relatos de experiência e publicações sobre Análise de Sentimentos de desenvolvedores de software, com o objetivo de caracterizar as abordagens

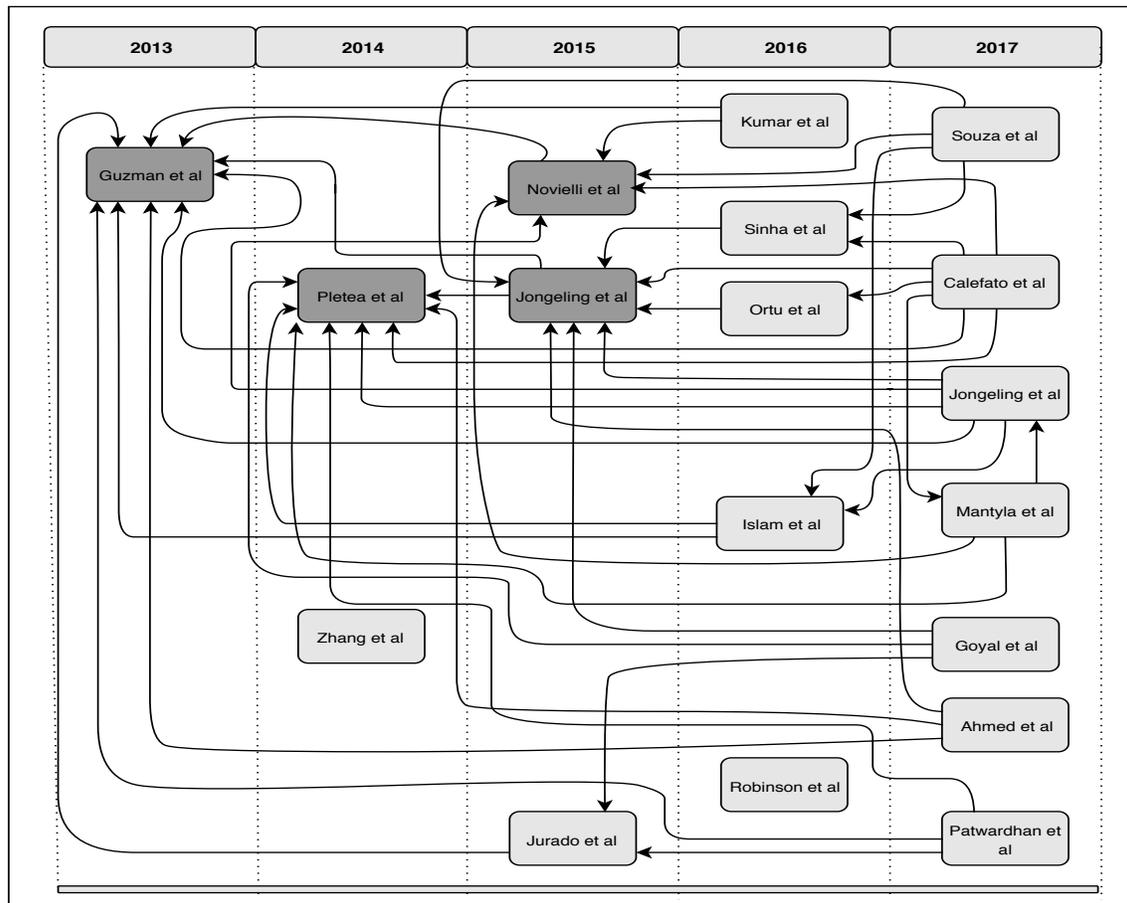


Fig. 7. Relacionamento entre os artigos

neste contexto. O mapeamento sistemático é útil para identificar e consolidar os trabalhos relacionados até o momento e orientar pesquisas futuras. As conclusões são importantes para ampliar o conhecimento sobre este tema de pesquisa.

Vale ressaltar que este estudo permitiu a descrição do problema alvo abordado, a especificação de questões de pesquisa, o desenvolvimento do mapeamento sistemático e, finalmente, a resposta às questões. Inicialmente, 218 artigos foram encontrados a partir de 4 bibliotecas digitais. Após a condução do mapeamento, 18 artigos relacionados foram selecionados e utilizados para responder as questões de pesquisas levantadas.

Com a sumarização, foi possível constatar que o número de trabalhos publicados na área apresentou uma curva de crescimento a partir de 2013, chegando ao seu ápice em 2017, demonstrando assim o interesse da comunidade acadêmica. Outra constatação diz respeito aos itens de configuração extraídos e as técnicas que são utilizadas para a extração dos mesmos. Pela análise dos trabalhos selecionados, grande parte dos itens de configuração escolhidos são provenientes de repositórios de códigos, sendo eles *issues* e *logs* de *commits*, extraídos por meio de API's existentes ou feitas pelos próprios autores.

Sobre as abordagens de classificação dos sentimentos, o método lexical é o mais utilizado pelos trabalhos relacionados.

A ferramenta SentiStrength, classificada como a mais utilizada pelos trabalhos selecionados usa tal abordagem. No mapeamento constatou-se que as emoções dos desenvolvedores foram levadas em conta em uma minoria dos estudos. A maior parte considera apenas a polaridade, ou seja, sentimentos positivos, negativos e ausência de sentimentos (neutro).

Como trabalhos futuros pretende-se aplicar uma busca manual e *snowballing* nos artigos selecionados, com o intuito de ampliar o conhecimento sobre a área bem como desenvolver uma ferramenta de Análise de Sentimentos específica para o domínio de Desenvolvimento de Software.

Outro trabalho futuro, já em andamento, leva em consideração a existência de diversas ferramentas de Análise de Sentimentos presentes no meio de pesquisa, sendo algumas utilizadas na área de Engenharia de Software, área a que este artigo tem interesse, mesmo que nem todas sejam aplicadas especificamente no domínio da área em questão. Para essa situação, um *framework* conceitual foi especificado, cujo o objetivo é levantar pontos positivos e negativos em algumas das ferramentas de Análise de Sentimentos selecionadas que tenham relevância para o domínio. Após a consolidação do *framework*, um experimento será realizado com dados reais de algum projeto *open source* à fim de validá-lo.

Este trabalho é parcialmente apoiado pela UFJF.

REFERENCES

- [1] R. Souza and B. Silva, "Sentiment analysis of travis ci builds," in *Proceedings of the 14th International Conference on Mining Software Repositories*. IEEE Press, 2017, pp. 459–462.
- [2] E. Guzman and B. Bruegge, "Towards emotional awareness in software development teams," in *Proceedings of the 2013 9th joint meeting on foundations of software engineering*. ACM, 2013, pp. 671–674.
- [3] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," *Empirical Software Engineering*, vol. 22, no. 5, pp. 2543–2584, 2017.
- [4] M. R. Islam and M. F. Zibran, "Towards understanding and exploiting developers' emotional variations in software engineering," in *Software Engineering Research, Management and Applications (SERA), 2016 IEEE 14th International Conference on*. IEEE, 2016, pp. 185–192.
- [5] D. Pletea, B. Vasilescu, and A. Serebrenik, "Security and emotion: sentiment analysis of security discussions on github," in *Proceedings of the 11th working conference on mining software repositories*. ACM, 2014, pp. 348–351.
- [6] R. Jongeling, S. Datta, and A. Serebrenik, "Choosing your weapons: On sentiment analysis tools for software engineering research," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2015, pp. 531–535.
- [7] M. Mäntylä, D. Graziotin, and M. Kuutilla, "The evolution of sentiment analysis - a review of research topics, venues, and top cited papers," 12 2016.
- [8] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, 2014, p. 38.
- [9] B. A. Kitchenham, D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further research—a participant-observer case study," *Information and Software Technology*, vol. 53, no. 6, pp. 638–651, 2011.
- [10] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 2005, pp. 347–354.
- [11] C. Bhadane, H. Dalal, and H. Doshi, "Sentiment analysis: Measuring opinions," *Procedia Computer Science*, vol. 45, pp. 808–814, 2015.
- [12] F. Jurado and P. Rodriguez, "Sentiment analysis in monitoring software development processes: An exploratory case study on github project issues," *Journal of Systems and Software*, vol. 104, pp. 82–89, 2015.
- [13] M. B. Christopher, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2016.
- [14] J. B. Lovins, "Development of a stemming algorithm," *Mech. Translat. & Comp. Linguistics*, vol. 11, no. 1-2, pp. 22–31, 1968.
- [15] A. Voutilainen, "Part-of-speech tagging," *The Oxford handbook of computational linguistics*, pp. 219–232, 2003.
- [16] A. P. Jain and P. Dandannavar, "Application of machine learning techniques to sentiment analysis," in *Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on*. IEEE, 2016, pp. 628–632.
- [17] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.
- [18] M. Pai, M. McCulloch, J. D. Gorman, N. Pai, W. Enanoria, G. Kennedy, P. Tharyan, and J. J. Colford, "Systematic reviews and meta-analyses: an illustrated, step-by-step guide." *The National medical journal of India*, vol. 17, no. 2, pp. 86–95, 2004.
- [19] B. A. Kitchenham, E. Mendes, and G. H. Travassos, "Cross versus within-company cost estimation studies: A systematic review," *IEEE Transactions on Software Engineering*, vol. 33, no. 5, 2007.
- [20] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *EASE*, vol. 8, 2008, pp. 68–77.
- [21] M. Ortu, G. Destefanis, B. Adams, A. Murgia, M. Marchesi, and R. Tonelli, "The jira repository dataset: Understanding social aspects of software development," in *Proceedings of the 11th international conference on predictive models and data analytics in software engineering*. ACM, 2015, p. 1.
- [22] W. N. Robinson, T. Deng, and Z. Qi, "Developer behavior and sentiment from data mining open source repositories," in *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. IEEE, 2016, pp. 3729–3738.
- [23] M. V. Mäntylä, N. Novielli, F. Lanubile, M. Claes, and M. Kuutilla, "Bootstrapping a lexicon for emotional arousal in software engineering," in *Mining Software Repositories (MSR), 2017 IEEE/ACM 14th International Conference on*. IEEE, 2017, pp. 198–202.
- [24] M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, and B. Adams, "The emotional side of software developers in jira," in *Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on*. IEEE, 2016, pp. 480–483.
- [25] V. Sinha, A. Lazar, and B. Sharif, "Analyzing developer sentiment in commit logs," in *Proceedings of the 13th International Conference on Mining Software Repositories*. ACM, 2016, pp. 520–523.
- [26] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi, "Sentier: A customized sentiment analysis tool for code review interactions," in *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. IEEE Press, 2017, pp. 106–111.
- [27] A. Goyal and N. Sardana, "Nrfixer: Sentiment based model for predicting the fixability of non-reproducible bugs," *e-Infomatica Software Engineering Journal*, vol. 11, no. 1, 2017.
- [28] A. Patwardhan, "Sentiment identification for collaborative, geographically dispersed, cross-functional software development teams," in *Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on*. IEEE, 2017, pp. 20–26.
- [29] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Software Engineering*, pp. 1–31, 2017.
- [30] N. Novielli, F. Calefato, and F. Lanubile, "The challenges of sentiment detection in the social programmer ecosystem," in *Proceedings of the 7th International Workshop on Social Software Engineering*. ACM, 2015, pp. 33–40.
- [31] A. K. M. M. M. Rahman and C. K. Roy, "Embedded emotion-based classification of stack overflow questions towards the question quality prediction," 2016.
- [32] Y. Zhang, B. Shen, and Y. Chen, "Mining developer mailing list to predict software defects," in *Software Engineering Conference (APSEC), 2014 21st Asia-Pacific*, vol. 1. IEEE, 2014, pp. 383–390.
- [33] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen, "Boa: A language and infrastructure for analyzing ultra-large-scale software repositories," in *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013, pp. 422–431.
- [34] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the Association for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.
- [35] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.
- [36] G. Van Rossum and F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
- [37] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.