

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Análise de Sentimentos de Desenvolvedores de Software

André Archanjo Nunes Coelho

JUIZ DE FORA
JULHO, 2019

Análise de Sentimentos de Desenvolvedores de Software

ANDRÉ ARCHANJO NUNES COELHO

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Bacharelado em Sistemas de Informação

Orientador: Alessandra Marta de Oliveira Julio

Coorientador: José Maria Nazar David

JUIZ DE FORA

JULHO, 2019

ANÁLISE DE SENTIMENTOS DE DESENVOLVEDORES DE SOFTWARE

André Archanjo Nunes Coelho

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM BACHARELADO EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Alessandreia Marta de Oliveira Julio
Doutora em Computação

José Maria Nazar David
Doutor em Engenharia de Sistemas e Computação

Gleiph Ghiotto Lima de Menezes
Doutor em Computação

Luciana Conceição Dias Campos
Doutora em Engenharia Elétrica

JUIZ DE FORA
09 DE JULHO, 2019

Resumo

Análise de Sentimentos é uma área que busca identificar a polaridade de uma expressão, como positiva, neutra ou negativa. Atualmente, está sendo aplicada no contexto de desenvolvimento de softwares. Isto se deve ao fato de fatores humanos, como emoções e sentimentos, influenciarem no trabalho das pessoas. Os dados utilizados para analisar os sentimentos de desenvolvedores são em sua maioria *issues* e comentários, provenientes de repositórios de códigos. Porém, após extraí-los ainda é necessário fazer um pré-processamento para serem formatados e assim utilizados. Diante disso, este trabalho tem como objetivo inicial a elaboração de um mapeamento sistemático da literatura que englobe os trabalhos no contexto de Análise de Sentimento de desenvolvedores de software. Além disso, foi implementada a ferramenta ASDS para apoiar a Análise de Sentimentos de desenvolvedores de software a partir do pré-processamento dos dados obtidos e análises da polaridade dos dados.

Palavras-chave: Análise de Sentimentos, Desenvolvedores de Software, Ferramenta de Análise de Sentimentos.

Abstract

Sentiment Analysis is the task of identifying the polarity of a expression, as positive, neutral or negative. It is currently being applied in the context of software development. This is due to the fact that human factors, such as emotions and sentiments, influence people's work. The data used to analyze the sentiments of developers are mostly issues and comments, coming from code repositories. However, after extracting them it is still necessary to pre-process them to be formatted and then used. This work has as its initial objective the elaboration of a systematic mapping of the literature that encompasses the work in the context of Sentiment Analysis of software developers. In addition, the ASDS tool was implemented to support the Sentiment Analysis of software developers from the pre-processing of the obtained data and analysis of the polarity of the data.

Keywords: Sentiment Analysis, Software Developers, Sentiment Analysis Tool.

Conteúdo

Lista de Figuras	5
Lista de Tabelas	6
1 Introdução	7
1.1 Apresentação do Tema e sua Contextualização	7
1.2 Justificativa	8
1.3 Objetivos	8
1.4 Metodologia	9
1.5 Organização do Trabalho	9
2 Fundamentação Teórica	11
2.1 Análise de Sentimentos	11
2.1.1 Método Lexical	12
2.1.2 Aprendizado de Máquina	14
2.2 Framework Conceitual	14
2.3 Ferramentas	15
2.3.1 SentiStrength	15
2.3.2 NLTK	16
2.3.3 Stanford NLP	16
2.3.4 Watson Natural Language Understanding	16
2.3.5 SentiStrength-SE	17
2.4 Considerações Finais do Capítulo	17
3 Análise de Sentimentos no Contexto de Desenvolvedores de Software: Um Mapeamento Sistemático da Literatura	18
3.1 Mapeamento Sistemático da Literatura	18
3.2 Planejamento	18
3.3 Condução do Mapeamento	23
3.4 Resultados	25
3.5 Considerações Finais do Capítulo	30
4 Framework Conceitual	32
4.1 Instanciação do Framework Conceitual	32
4.2 Características	33
4.3 Seleção da Ferramenta	35
4.4 Considerações Finais do Capítulo	36
5 ASDS - Uma Ferramenta para Analisar Sentimentos	37
5.1 Tecnologia Utilizada	37
5.2 Exemplo de Utilização	38
5.2.1 Base de Dados	38
5.2.2 Pré-Processamento	41
5.2.3 Processamento	41
5.2.4 Análise dos Resultados	43

5.3	Considerações Finais do Capítulo	47
6	Conclusão	49
6.1	Resultados Obtidos	49
6.2	Trabalhos Futuros	50
	Bibliografia	52

Lista de Figuras

2.1	Exemplo de um processo de Análise de Sentimentos utilizando Aprendizado de Máquina (Adaptado de (JAIN; DANDANNAVAR, 2016))	14
3.1	Total de artigos selecionados em cada uma das bibliotecas digitais após a aplicação dos critérios de exclusão	24
3.2	Fases da seleção dos artigos do mapeamento	25
3.3	Total de artigos publicados por ano	26
3.4	Total de utilizações das ferramentas de Análise de Sentimentos	28
3.5	Escalas de sentimentos usadas nos trabalhos selecionados a partir do mapeamento	29
3.6	Relacionamento entre os artigos	30
4.1	Fluxo do <i>Framework</i> Conceitual	33
4.2	Sumarização das Características da Ferramenta Selecionada	36
5.1	As 10 palavras com maior frequência no repositório	44
5.2	As 10 palavras positivas com maior frequência no repositório	44
5.3	Os 10 usuários mais importantes e seus respectivos números de textos escritos no repositório	45
5.4	Os 10 principais usuários e seus respectivos sentimentos distinguidos (positivo, neutro e negativo)	47
5.5	As 10 principais palavras dos 4 principais usuários	48

Lista de Tabelas

2.1	Exemplo do Método Lexical	12
3.1	PICO	20
3.2	String de Busca Geral	21
3.3	Bibliotecas Digitais Utilizadas no mapeamento sistemático	21
3.4	String de busca específica para cada uma das bibliotecas digitais utilizadas	21
3.5	Critérios de exclusão	22
3.6	Artigos de Controle	22
3.7	Total de artigos retornados para cada uma das bibliotecas digitais	23
3.8	Total de artigos por biblioteca digital após tratamento de duplicidade . . .	24
3.9	Conjuntos de diferentes emoções	29
3.10	Artigos mais citados	31
4.1	Características das Ferramentas que apoiam a Análise de Sentimentos . . .	35
5.1	Palavras e seus respectivos valores polares	46

1 Introdução

A Análise de Sentimentos tem como objetivo descobrir o sentimento, também chamado de polaridade, nos textos de seus autores (THELWALL et al., 2010) (WILSON; WIEBE; HOFFMANN, 2005), podendo classificá-lo como positivo, negativo ou neutro. É uma área de estudo recente que abrange diversos contextos, como domínios de mídias sociais, como o Twitter para a predição de eleições (TUMASJAN et al., 2010) e até desenvolvimento de softwares (SOUZA; SILVA, 2017).

1.1 Apresentação do Tema e sua Contextualização

O desenvolvimento distribuído de software é uma atividade intensiva em esforço humano. Assim, a forma como os desenvolvedores encaram suas tarefas é de suma importância. Em um ambiente de desenvolvimento em que profissionais compartilham os esforços de desenvolvimento, o estado emocional deles é fundamental (ROBINSON; DENG; QI, 2016). A experiência dos desenvolvedores pode ser definida como um meio para capturar o que eles pensam e sentem sobre suas atividades dentro de seus ambientes de trabalho, com a suposição de que a melhoria da experiência do desenvolvedor tem impactos positivos em aspectos como um melhor desempenho da equipe.

Desde que a Análise de Sentimentos se tornou um tema em voga, considerando o crescente interesse por parte da academia e da indústria, muitas ferramentas e métodos têm sido propostos para a aplicação em diversos contextos (SINHA; LAZAR; SHARIF, 2016). Entre eles, a Análise de Sentimentos dos desenvolvedores de software, durante o desenvolvimento dos projetos que estão envolvidos (JONGELING; DATTA; SEREBRENIK, 2015), (JURADO; RODRIGUEZ, 2015), (ROUSINOPOULOS; ROBLES; GONZÁLEZ-BARAHONA, 2014). A Análise de Sentimentos neste contexto se dá em bases de dados disponíveis para mineração onde os sentimentos dos desenvolvedores podem ser extraídos (FARIAS et al., 2016). Analisar o estado emocional de pessoas pelo que elas escrevem/-programam pode ser útil para propor melhorias no ambiente de desenvolvimento. Por

exemplo, tomar medidas para contornar a situação caso haja um alto valor de sentimento negativo.

1.2 Justificativa

Existem diversas ferramentas desenvolvidas para o contexto de Análise de Sentimentos (THELWALL et al., 2010) (BIRD, 2006) (SOCHER et al., 2013) (ISLAM; ZIBRAN, 2017), porém ainda existem lacunas a serem preenchidas sobre elas. Uma das lacunas seria como obter dados a serem analisados por meio de um repositório de códigos. Outra lacuna seria o formato de entrada de tais dados na ferramenta, e por fim a precisão das saídas em um nível satisfatório.

É válido ressaltar que para o conhecimento do autor, no contexto de ferramentas de Análise de Sentimentos não há uma que possibilite a extração de dados em um repositório de códigos de sua escolha. Portanto, o trabalho tem como objetivo descrever uma ferramenta com tal funcionalidade, além de pré-processar os dados extraídos para serem usados como dados de entrada em uma ferramenta de Análise de Sentimentos integrada à ferramenta desenvolvida neste trabalho.

A ferramenta desenvolvida, denominada ASDS, proporciona uma interface gráfica que auxilia o usuário em saber o estado emocional dos desenvolvedores de um projeto. Sabendo do seus sentimentos, pode-se correlacionar o seu rendimento a isso (SOUZA; SILVA, 2017). Além das funcionalidades ditas no parágrafo anterior, extração e pré-processamento dos dados e a obtenção da polaridade deles, a ferramenta proporciona ao todo cinco análises para assim obter um maior conhecimento de um repositório.

1.3 Objetivos

Este trabalho tem como objetivo inicial a elaboração de um mapeamento sistemático da literatura referente à Análise de Sentimentos no contexto de desenvolvedores de software. O propósito é conhecer as pesquisas existentes até o momento na área em questão.

Outro objetivo é a elaboração de um *framework* conceitual, necessário para auxiliar na escolha de uma ferramenta de Análise de Sentimentos, para se integrar na fer-

ramenta ASDS. A escolha é fundamental, pois a obtenção dos valores polares se dá pela ferramenta selecionada.

Finalizando, este trabalho se propõe a desenvolver a ASDS, uma ferramenta que permite a extração semiautomático de dados de um repositório de códigos. Tais dados são analisados pela ferramenta de Análise de Sentimentos previamente selecionada e os resultados são utilizados pela ASDS. Posteriormente, outras análises sobre os sentimentos encontrados nos dados podem ser realizadas.

1.4 Metodologia

Como ponto de partida, um mapeamento sistemático da literatura foi efetuado, buscando por trabalhos relacionados a Análise de Sentimentos com foco em desenvolvedores de softwares. Isto é feito com a finalidade de conhecer mais sobre o estado da arte da área, a fim de possibilitar uma visão mais abrangente da área em questão.

Posteriormente, um *framework* conceitual é elaborado e instanciado. Seu objetivo é apoiar a escolha da ferramenta de Análise de Sentimentos mais indicada para o que o trabalho se propõe.

Em seguida, a ferramenta ASDS é desenvolvida. Aplica-se a ferramenta em um exemplo de utilização posteriormente, utilizando dados provenientes de um repositório de códigos. O critério para a seleção de um repositório de códigos leva em conta a disponibilidade de um grande número de projetos *open-sources*, por possibilitarem a visualização e a extração de dados de maneira direta. Isto se dá com o objetivo de avaliar o trabalho executado.

1.5 Organização do Trabalho

Este trabalho está organizado como a seguir. O Capítulo 2 apresenta a fundamentação teórica necessária para a compreensão do trabalho. O mapeamento sistemático da literatura na área de Análise de Sentimentos no contexto de desenvolvedores de software é descrito no Capítulo 3. No Capítulo 4, o *framework* conceitual é elaborado, sua instanciagem apresentada e a escolha de uma ferramenta de Análise de Sentimentos é feita.

No Capítulo 5 a ferramenta ASDS é apresentada, junto com um exemplo de utilização. Por fim, no Capítulo 6 algumas conclusões do trabalho são apresentadas juntamente com possibilidades de trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo, são abordados conceitos necessários para compreensão desta monografia. A Fundamentação Teórica aqui presente colabora para conhecimento de termos e conceitos. Desta forma tal capítulo fornece todo o material necessário para situar o leitor durante a leitura do trabalho.

O conceito apresentado na Seção 2.1 é sobre Análise de Sentimentos e suas Abordagens de Classificação. A Seção 2.2 aborda o que é um *framework* conceitual. Posteriormente, a Seção 2.3 apresenta as ferramentas para a Análise de Sentimentos. Finalizando, a Seção 2.4 traz as considerações finais deste capítulo.

2.1 Análise de Sentimentos

A Análise de Sentimentos tem por objetivo determinar a polaridade de uma expressão, como positiva ou negativa, e no caso de haver ausência de sentimento, ou seja, não ser capaz de detectar sentimentos, considerá-la neutra (WILSON; WIEBE; HOFFMANN, 2005). Também é considerada uma área de estudos de Processamento de Linguagem Natural com o objetivo de encontrar elementos subjetivos no texto (BHADANE; DALAL; DOSHI, 2015).

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: raiva (*anger*), desgosto (*disgust*), medo (*fear*), alegria (*joy*), tristeza (*sadness*) e surpresa (*surprise*) (JURADO; RODRIGUEZ, 2015).

Existem duas abordagens para a classificação do trecho de um texto, descritos a seguir: Método Lexical (*Lexical Method*) e suas variações (BHADANE; DALAL; DOSHI, 2015), e Aprendizado de Máquina (*Machine Learning*) (CHRISTOPHER, 2016).

2.1.1 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 (BHADANE; DALAL; DOSHI, 2015). A Tabela 2.1 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].

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

Tabela 2.1: 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

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 (BHADANE; DALAL; DOSHI, 2015) 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 mais algumas técnicas de pré-processamento, tais como:

- *Stemming* (LOVINS, 1968): busca retirar os prefixos e sufixos da palavra antes de analisá-la;
- *Part of Speech Tagging* (VOUTILAINEN, 2003): 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 “*NOT_good*”.

Alguns autores, como Bhadane, Dalal e Doshi (2015), 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.

2.1.2 Aprendizado de Máquina

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 (CHRISTOPHER, 2016). 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 (JAIN; DANDANAVAR, 2016).

Algumas etapas realizadas para a Análise de Sentimentos usando Aprendizado de Máquina são apresentadas na Figura 2.1. 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.

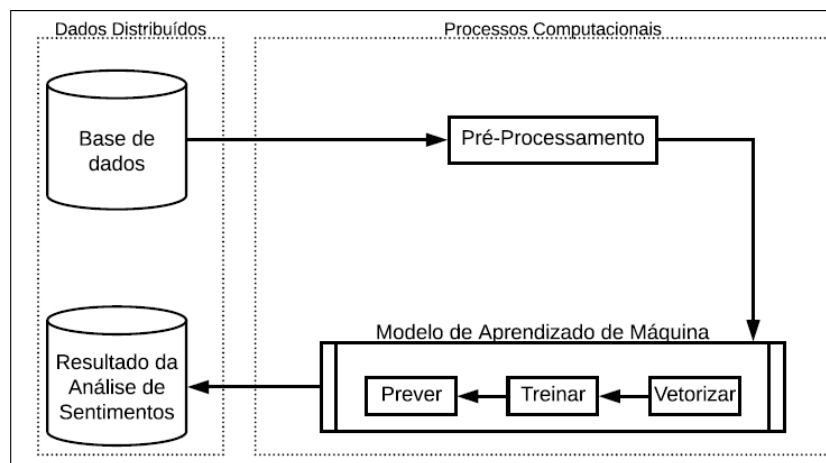


Figura 2.1: Exemplo de um processo de Análise de Sentimentos utilizando Aprendizado de Máquina (Adaptado de (JAIN; DANDANAVAR, 2016))

2.2 Framework Conceitual

Framework conceitual é uma rede de conceitos, onde cada um desempenha um papel integral, mas que estão interligados entre si com a finalidade de prover conhecimento para o entendimento de um determinado assunto. O conhecimento não é derivado de fatos, mas sim de interpretações suaves de intenções (JABAREEN, 2009). Em outras palavras, é como o pesquisador entende a rede de conceitos presente em seu estudo e como se conectam.

Pode-se dizer que o framework conceitual prepara o palco para representação do resultado de um estudo, originado de uma investigação de um determinado problema (REGONIEL, 2015). Através disso, os conceitos são obtidos, formando uma rede de conceitos.

2.3 Ferramentas

Como dito por (JONGELING; DATTA; SEREBRENIK, 2015), existem ferramentas para Análise de Sentimentos, tais como: SentiStrength (THELWALL et al., 2010), NLTK (BIRD, 2006), Stanford NLP Sentiment Analyser (SOCHER et al., 2013) e a Alchemy API, que agora faz parte da Watson Natural Language Understanding ¹. Uma outra ferramenta, derivada do SentiStrength, é a SentiStrength-SE (ISLAM; ZIBRAN, 2017). Todas as ferramentas citadas serão descritas a seguir.

2.3.1 SentiStrength

A ferramenta estima a intensidade dos sentimentos em pequenos textos até mesmo em linguagem coloquial. Os autores (THELWALL et al., 2010) afirmam que em pequenos textos de redes sociais consegue-se, através de sua utilização, ter praticamente uma precisão humana, exceto em textos políticos (válido para textos em Inglês)².

É possível obter o sentimento, que é o dado de saída da ferramenta a partir um texto, através de 4 métodos diferentes. O primeiro, e o *default*, é o dual onde há duas intensidades de sentimentos, sendo uma positiva e outra negativa. A intensidade positiva varia de 1 à 5, enquanto a negativa varia de -1 a -5. O segundo é o binário, em que a saída diz se é positivo ou negativo. O terceiro, conhecido como ternário, possibilita um resultado a mais que o binário, sendo o neutro. O quarto e último é a escala única de -4 a 4, onde menor que 0 é negativo, maior que 0 positivo e igual a 0 neutro.

A ferramenta utiliza métodos lexicais, analisando os conteúdos em um domínio geral. A entrada para análise é feita em um arquivo *.txt*, devidamente formatado de acordo com as especificações disponibilizadas pela documentação da ferramenta (THELWALL et

¹<https://www.ibm.com/watson/services/natural-language-understanding/>

²<http://sentistrength.wlv.ac.uk/>

al., 2010).

2.3.2 NLTK

NLTK (Natural Language Toolkit) é um conjunto de bibliotecas e programas para processamento de linguagem natural simbólica e estatística (NLP). Foi desenvolvido para apoiar a pesquisa e o ensino sobre NLP ou áreas estreitamente relacionadas, incluindo linguística empírica, ciência cognitiva e aprendizado de máquinas. O conjunto de ferramentas suporta funcionalidades de classificação, *tokenização*, derivação, marcação, análise e linguagem semântica, além de aceitar a entrada em texto (BIRD, 2006).

2.3.3 Stanford NLP

Com um *design* simples e uma organização dos processos que fornecem análise de linguagem natural, Stanford NLP utiliza aprendizado de máquina para lidar com os problemas da Análise de Sentimentos. A entrada é feita em texto. Como saída apresenta uma escala (- -, -, 0, +, ++), onde - - indica muito negativo, ++ muito positivo e 0 ausência de sentimento (SOCHER et al., 2013).

2.3.4 Watson Natural Language Understanding

O Alchemy API foi integrado ao Watson Natural Language Understanding³, que é uma coleção de API's que oferece uma análise textual através do processamento de linguagem natural. O conjunto de API's pode ajudar a analisar textos para encontrar sentimentos, emoções e conceitos. Com relação a detecção de sentimentos, é retornado para o fragmento de um texto, uma pontuação no intervalo de [-1 a 1], onde um valor menor que zero considera-se o texto negativo, maior um valor positivo. Caso seja zero o valor encontrado, significa que não foi possível encontrar um sentimento, o considerando neutro.

³<https://www.ibm.com/watson/services/natural-language-understanding/>

2.3.5 SentiStrength-SE

SentiStrength-SE surgiu com a necessidade de analisar sentimentos no contexto específico de Engenharia de Software, já que as ferramentas disponíveis nesse contexto apresentavam precisão muito baixa. Após um estudo para identificar os problemas responsáveis pela baixa precisão, eles foram tratados no desenvolvimento da ferramenta.

A ferramenta usa a abordagem lexical do SentiStrength com uma entrada feita em arquivos *.txt*. As polaridades de saída também podem ser tratadas em maneira binária, ternária e em pares (positivo, negativo): -1 a -5 para o negativo, +1 a +5 para positivo. Caso o par seja (+1, -1), o texto analisado é considerado neutro (ISLAM; ZIBRAN, 2017).

2.4 Considerações Finais do Capítulo

O conhecimento adquirido, com a apresentação dos conceitos abordados neste capítulo, serve de base para o correto entendimento e compreensão do que é apresentado no restante do trabalho. Por este motivo, foram apresentadas aqui algumas definições que julgou-se importante. O próximo capítulo, apresenta o mapeamento sistemático realizado, utilizando tais conceitos explicados anteriormente, além de mostrar a importância de conhecê-los.

3 Análise de Sentimentos no Contexto de Desenvolvedores de Software: Um Mapeamento Sistemático da Literatura

Neste capítulo, a Seção 3.1 explica o que é um mapeamento sistemático da literatura. A Seção 3.2 apresenta o planejamento, uma das etapas do mapeamento sistemático da literatura, enquanto a Seção 3.3 demonstra a condução do mapeamento sistemático da literatura. Os resultados obtidos são apresentados na Seção 3.4. No final, a Seção 3.5 traz as considerações finais do capítulo.

3.1 Mapeamento Sistemático da Literatura

Um mapeamento sistemático da literatura é uma forma de identificar, avaliar e interpretar trabalhos disponíveis relevantes para questões de pesquisa particulares. Uma das razões para as suas realizações é que eles resumem as evidências existentes em relação a um tratamento ou tecnologia (KITCHENHAM, 2004).

Este mapeamento sistemático da literatura foi conduzido de acordo com as etapas descritas a seguir, sendo elas: Planejamento, Condução do Mapeamento e os Resultados. Para apoiar este processo foi utilizada a ferramenta Parsif.al⁴, que é uma ferramenta *online* projetada para apoiar os pesquisadores a realizar mapeamentos sistemáticos de literatura.

3.2 Planejamento

Esta fase consiste em especificar e avaliar o protocolo a ser seguido. O protocolo apresenta o objetivo do mapeamento sistemático da literatura, as questões de pesquisa, a *string* de busca, a definição das fontes e os critérios de inclusão e exclusão dos trabalhos obtidos.

⁴<https://parsif.al/>

O objetivo 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 (QP) apresentadas a seguir:

- **QP1:** Quais itens são extraídos para a Análise de Sentimentos de desenvolvedores de software? Através dessa questão pretende-se identificar quais itens 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 na Análise de Sentimentos de desenvolvedores de software? Nessa questão busca-se identificar como é feita a extração dos itens 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 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 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 (PAI et al., 2004). 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) (KITCHENHAM; MENDES; TRAVASSOS, 2007). A atividade de revisão da *string* foi realizada por grupo de estudos semanalmente, onde era testada e os resultados obtidos apresentados, realizando incrementos necessários até chegar ao resultado final. A especificação do método PICO está presente na Tabela 3.1.

Tabela 3.1: 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 3.2.

Foram selecionadas quatro bibliotecas digitais para efetuar a busca, como mostra a Tabela 3.3.

Por conta de particularidades de duas bibliotecas digitais, Scopus e Science Direct, a *string* de busca geral foi adaptada para esses casos, como mostrado na Tabela 3.4.

Tabela 3.2: 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”*)

Tabela 3.3: Bibliotecas Digitais Utilizadas no mapeamento sistemático

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

Tabela 3.4: String de busca específica para cada uma das bibliotecas digitais utilizadas

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

Um conjunto de critérios de inclusão e exclusão foi especificado para selecionar os estudos. Esses critérios foram 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 (PETERSEN et al.,

2008), 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 3.5. Como critérios de inclusão, foram considerados os estudos que atendem a negação dos critérios de exclusão.

Tabela 3.5: Critérios de exclusão

C1	Trabalhos que não apresentam ferramentas, algoritmos, abordagens, aplicações, modelos, 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 3.6, 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.

Tabela 3.6: Artigos de Controle

#	Ref	Artigo
1	(GUZMAN; BRUEGGE, 2013)	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	(JURADO; RODRIGUEZ, 2015)	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	(ORTU et al., 2015)	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 (GUZMAN; BRUEGGE, 2013) aplica a Análise de Sentimentos para melhorar a consciência emocional em equipes de desenvolvimento distribuídas. A abordagem proposta pelos pesquisadores extraiu e resumiu as emoções expressas em artefatos de colaboração e, posteriormente, exibiu os resumos aos líderes dos projetos analisados. Os líderes concordaram que os resultados apresentados foram úteis para criar uma consciência emocional. Jurado e Rodriguez (2015), propuseram técnicas de Análise Sentimentos para identificar e monitorar os sentimentos contidos no texto es-

crito 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. Com isso, encontrou-se que em textos de desenvolvedores, sentimentos são encontrados, e através destas informações seria possível analisar o processo de desenvolvimento de projetos. Por último, o trabalho descrito em (ORTU et al., 2015) 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 (ORTU et al., 2015), permite posteriores investigações do papel dos artefatos no desenvolvimento de software e sua relação com as emoções ali contidas.

3.3 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 3.7 sumariza a quantidade de artigos retornados por biblioteca digital.

Tabela 3.7: Total de artigos retornados para cada uma das bibliotecas digitais

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

Para realizar o tratamento de duplicidade entre artigos em diferentes bibliotecas digitais, foi adotada a seguinte regra: quando constatada a duplicidade de determinado artigo, escolheu-se uma biblioteca digital para o artigo permanecer. Em relação às de-

mais 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 3.8.

Tabela 3.8: Total de artigos por biblioteca digital 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.

A distribuição de artigos por bibliotecas digitais está ilustrada na Figura 3.1. 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.

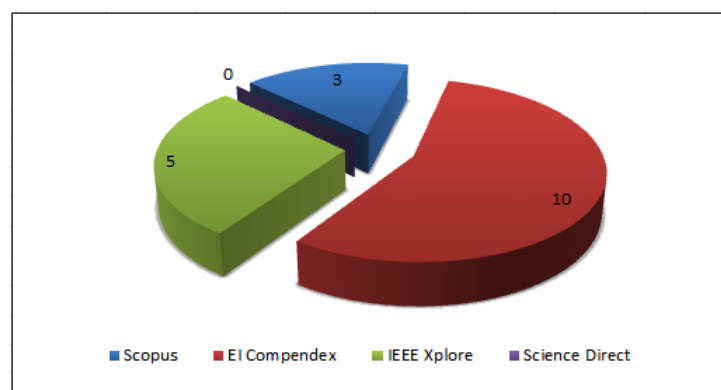


Figura 3.1: Total de artigos selecionados em cada uma das bibliotecas digitais após a aplicação dos critérios de exclusão

A Figura 3.2 resume as fases de seleção dos artigos do mapeamento com o total obtido em cada fase.

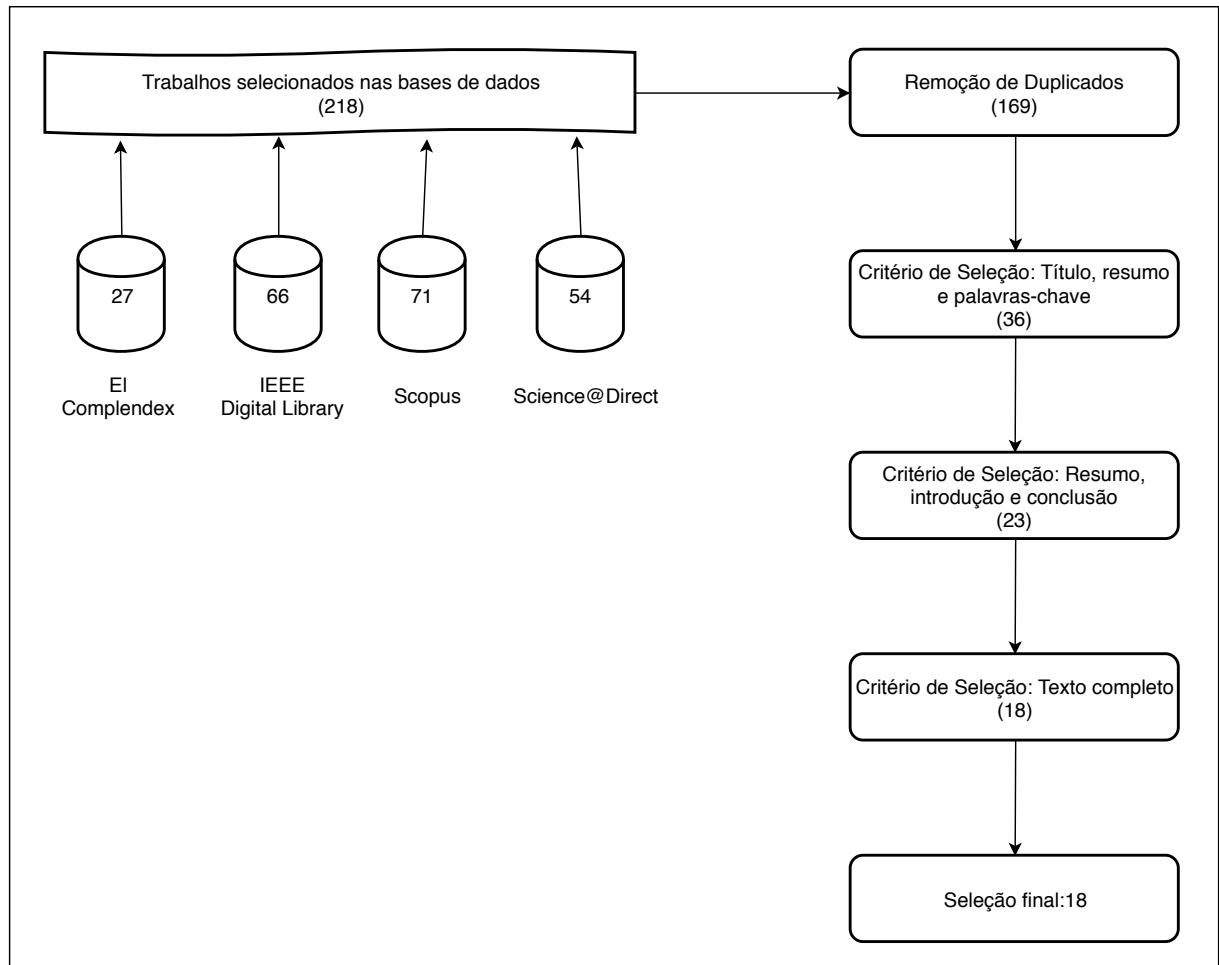


Figura 3.2: Fases da seleção dos artigos do mapeamento

3.4 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 3.3. 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

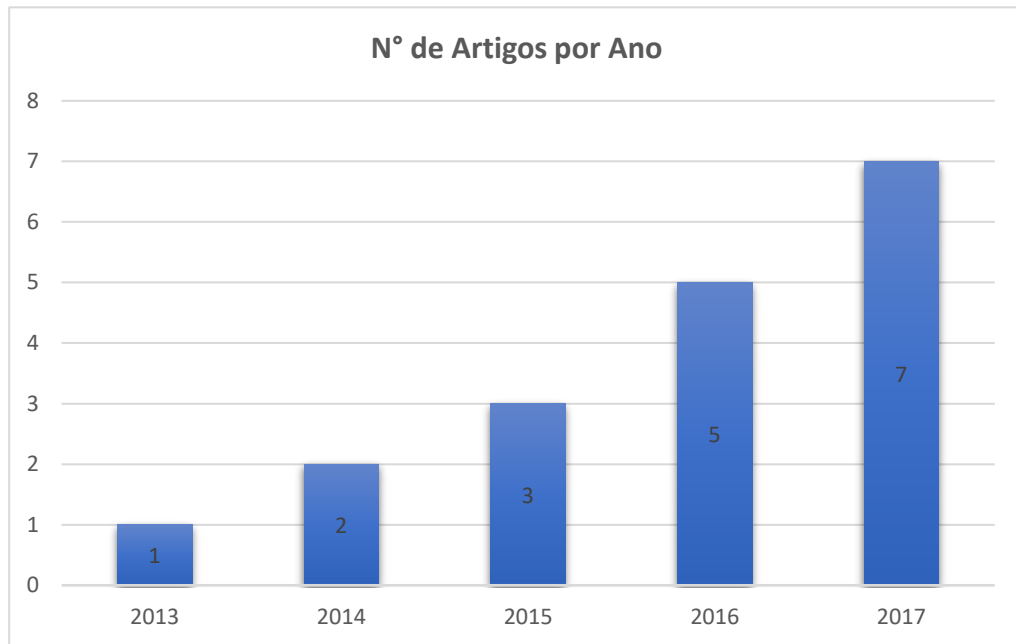


Figura 3.3: Total de artigos publicados por ano

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 (JURADO; RODRIGUEZ, 2015), além de ser o maior *host* de códigos do mundo, com milhões de programadores colaborando em mais de 10 milhões de repositórios (PLETEA; VASILESCU; SEREBRENIK, 2014). Outro item buscado em grande quantidade foi as questões e respostas do StackOverflow⁵, 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 (DYER et al., 2013), uma infraestrutura com um domínio de linguagem es-

⁵O Stack Overflow é um site americano gratuito de perguntas e respostas sobre desenvolvimento de software. (<https://stackoverflow.com/>)

pecífico

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 (PATWARDHAN, 2017).

A terceira questão de pesquisa (**QP3**) pretende descobrir quais ferramentas são utilizadas para reconhecer os sentimentos nos itens extraídos. Para a criação da Figura 3.4, considerou-se 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 (THELWALL et al., 2010), 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 (BIRD, 2006) (Natural Language Toolkit), uma vasta biblioteca implementada na linguagem de programação Python (ROSSUM; JR, 1995), 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. (JONGELING; DATTA; SEREBRENIK, 2015) (JONGELING et al., 2017), 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 (AHMED et al., 2017) (GOYAL; SARDANA, 2017) (CALEFATO et al., 2017), ou até mesmo classificar manualmente os seus textos (ORTU et al., 2016).

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

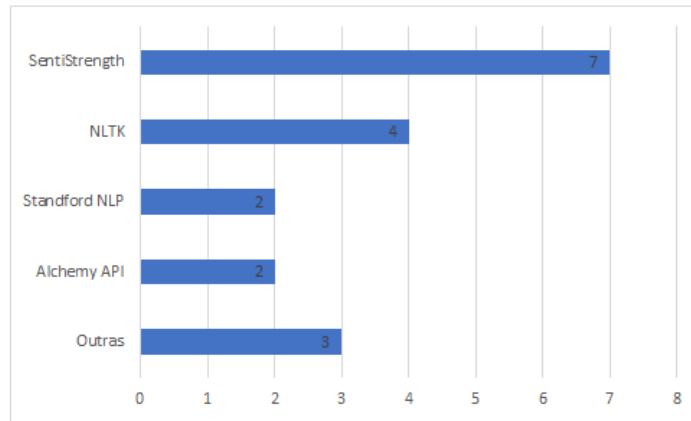


Figura 3.4: Total de utilizações das ferramentas de Análise de Sentimentos

das ferramentas de Análise de Sentimentos verificando o impacto de se escolher uma (JONGELING; DATTA; SEREBRENIK, 2015) (JONGELING et al., 2017) e construir uma ferramenta de Análise de Sentimentos designada especificamente para comentários de códigos (AHMED et al., 2017).

A questão (**QP5**) pretende descobrir quais as classificações de sentimentos mais usadas. Através desta pergunta foi constatado, como mostra a Figura 3.5, 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 desenvolvedores de software (JONGELING; DATTA; SEREBRENIK, 2015). 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, 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 3.9. 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.

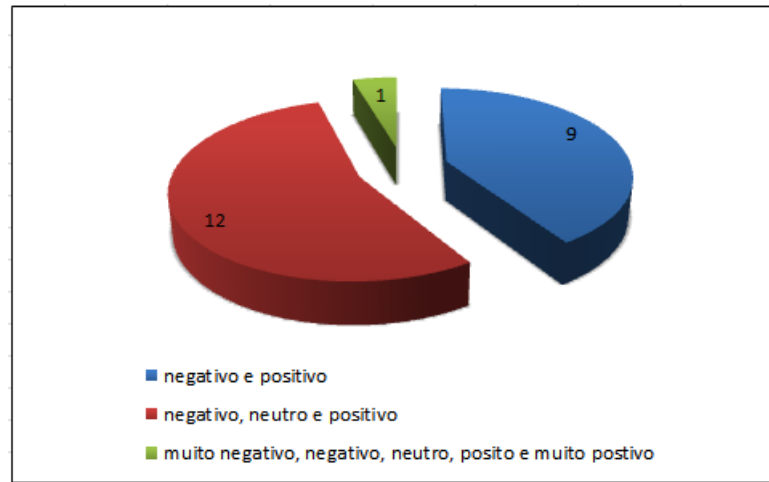


Figura 3.5: Escalas de sentimentos usadas nos trabalhos selecionados a partir do mapeamento

Tabela 3.9: Conjuntos de diferentes emoções

#	Emoções
1º	<i>joy, love, surprise, anger, fear</i>
2º	<i>love, joy, sadness</i>
3º	<i>love, joy, surprise, anger, fear and sadness</i>
4º	<i>love, joy, anger and sadness</i>
5º	<i>anger, anticipation, disgust, fear, joy, sadness, surprise, and trust</i>
6º	<i>anger, surprise and happiness</i>
7º	<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 (THELWALL et al., 2010), 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 3.6 os trabalhos mais citados estão destacados em azul, Guzman et al. (GUZMAN; BRUEGGE, 2013), Pletea et al. (PLETEA; VASILESCU; SEREBRENIK, 2014), Jongeling et al. (JONGELING; DATTA; SEREBRENIK, 2015) e Novielli et al. (NOVIELLI; CALEFATO; LANUBILE, 2015), citados 9, 8, 7 e 5 vezes

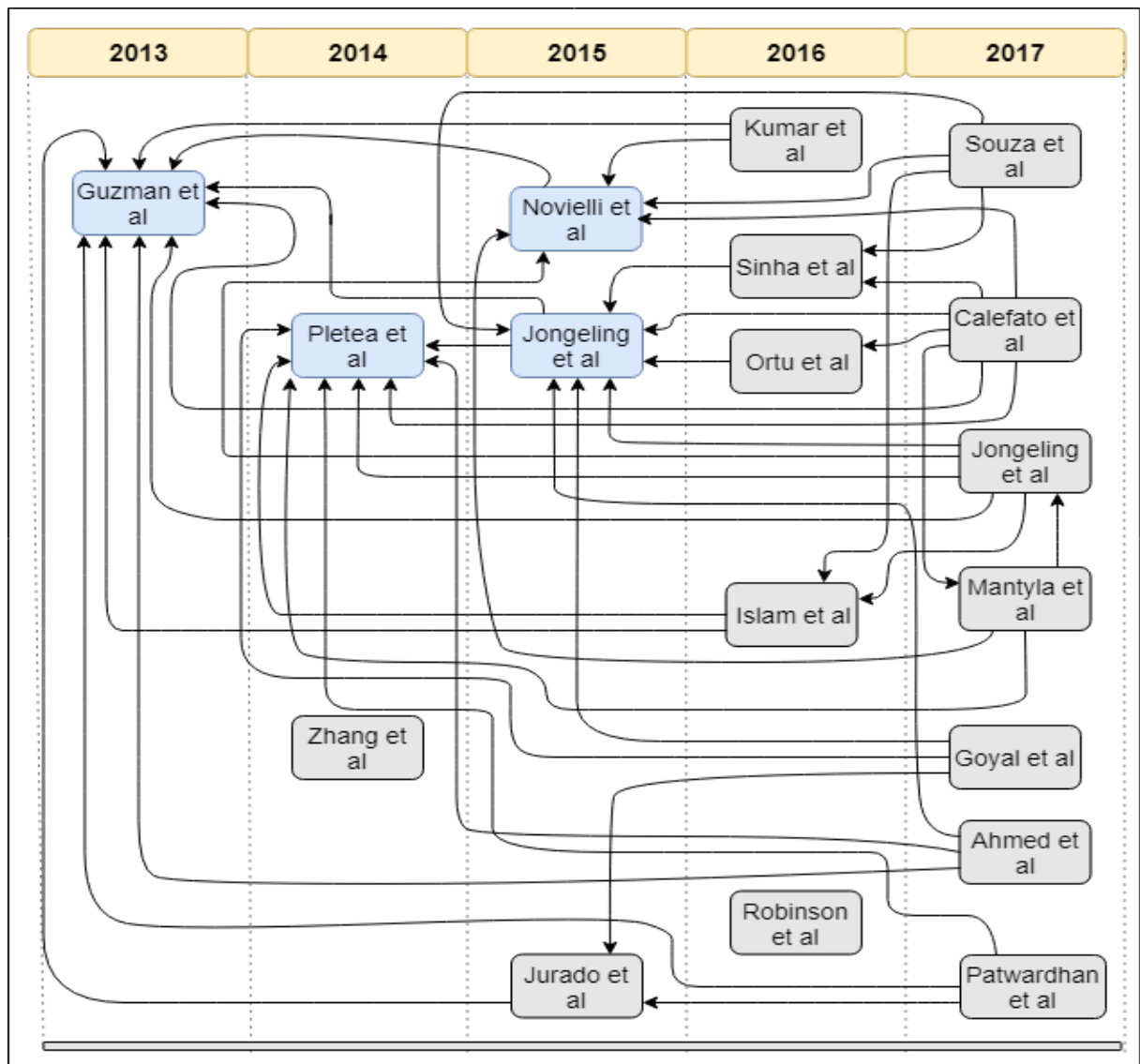


Figura 3.6: Relacionamento entre os artigos

respectivamente, como mostra a Tabela 3.10. Isso mostra possível influência destes trabalhos sobre os demais. Vale ressaltar que (GUZMAN; BRUEGGE, 2013) é o primeiro a abordar o tema de Análise de Sentimentos de desenvolvedores de software, se tornando o mais citado.

3.5 Considerações Finais do Capítulo

Este capítulo apresentou o mapeamento sistemático adaptado de (COELHO et al., 2018) publicado na conferência do CLEI-SLISW 2018.

Sumarizando os resultados encontrados, pode-se constatar que o número de trabalhos publicados na área apresentou uma curva de crescimento a partir de 2013, chegando

Tabela 3.10: Artigos mais citados

#	Ref	Artigo
1º	(GUZMAN; BRUEGGE, 2013)	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.
2º	(PLETEA; VASILESCU; SEREBRENİK, 2014)	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.
3º	(JONGELING; DATTA; SEREBRENİK, 2015)	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.
4º	(NOVIELLI; CALEFATO; LANUBILE, 2015)	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.

ao seu ápice em 2017, demonstrando assim o interesse da comunidade acadêmica. Outra constatação diz respeito aos itens extraídos e as técnicas que são utilizadas para a extração dos mesmos. Pela análise dos trabalhos selecionados, grande parte dos itens 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 no caso de não detectar sentimentos, neutro.

4 Framework Conceitual

Neste capítulo é proposto um *framework* conceitual, com a finalidade de apoiar a escolha da ferramenta mais adequada para o contexto de Análise de Sentimentos, descrevendo pontos positivos e negativos das características de cada ferramenta. É importante ressaltar que no caso deste estudo, cada conceito é uma característica a ser analisada. Na Seção 4.1 é apresentado a instanciação do *framework* conceitual. A Seção 4.2 apresenta as características a serem levadas em conta para a seleção. A Seção 4.3 compara as ferramentas entre si, levando em conta as características de cada ferramenta. Por fim, a Seção 4.4 traz as considerações finais do capítulo.

4.1 Instanciação do Framework Conceitual

Para uma melhor visualização, foi elaborada a Figura 4.1 a fim de mostrar o passo a passo da instanciação do *framework* conceitual, permitindo um melhor entendimento de todo o processo que ocorre.

Primeiramente, antes da escolha da ferramenta mais adequada para a Análise de Sentimentos, deve-se verificar de qual base de dados ocorrerá a extração das informações desejadas, como pode ser visto na Figura 4.1. Pode-se vir de repositórios de códigos como o GitHub, JIRA ou de um repositório privado, além do site de perguntas e respostas sobre programação, o Stack Overflow.

Após a escolha de uma base de dados, ocorrerá a extração dos dados, seja por uma API ou *html scraping*⁶, e com isso o seu armazenamento em um arquivo *.txt*. Após essa etapa, os dados passam por um pré-processamento para que possam ser usados em algumas das ferramentas, na etapa seguinte.

Com os dados pré-processados, uma ferramenta é selecionada para obter os dados desejados, o valor polar de informações desejadas de um projeto de software, podendo ser *issues* e comentários de seus desenvolvedores. Com estes dados em mãos, uma análise

⁶Varredura de um documento html onde através de um script é feita a coleta dos dados que na mesma constam.

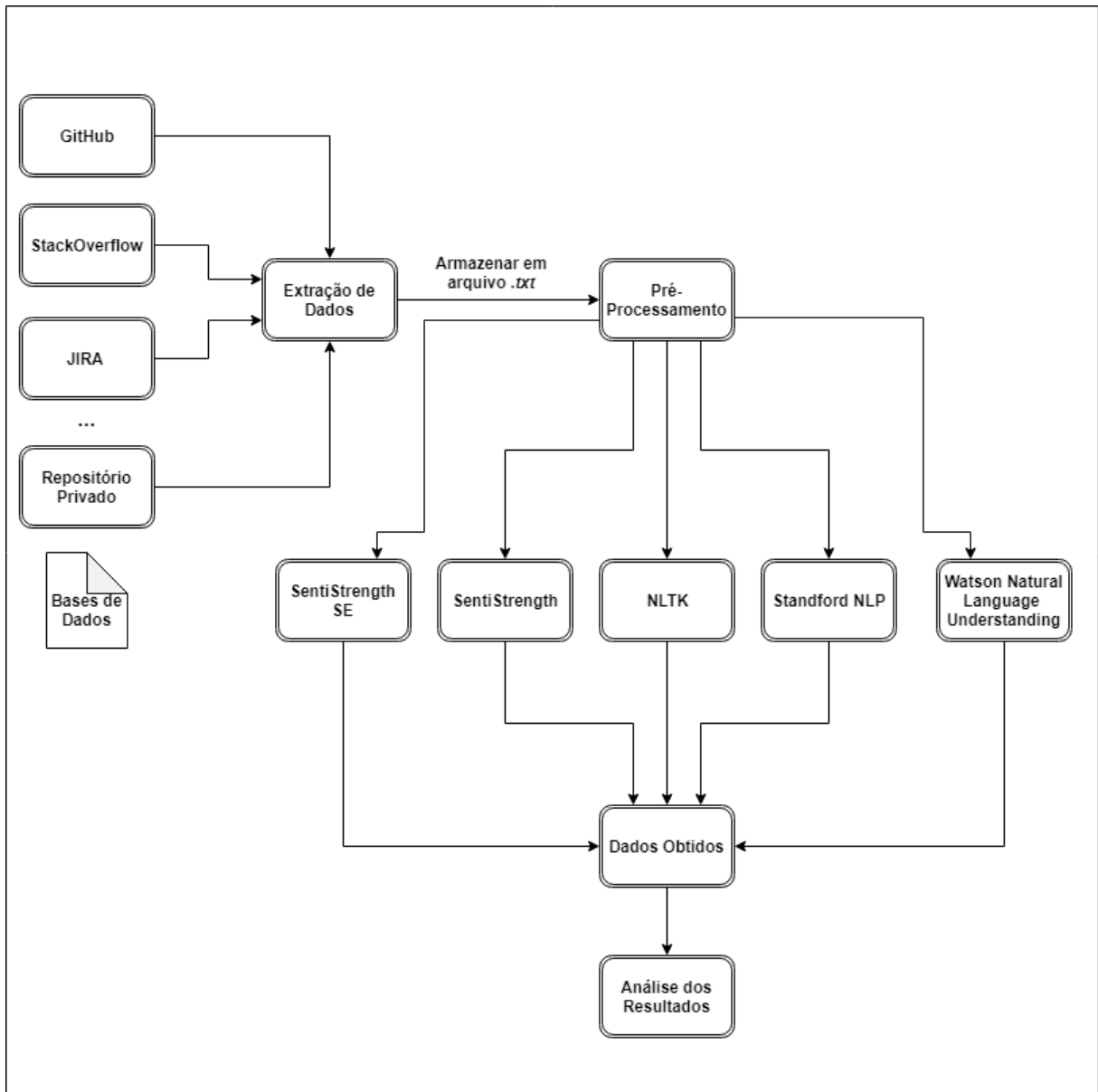


Figura 4.1: Fluxo do *Framework* Conceitual

dos resultados é realizada para obter informações relevantes, a fim de contribuir com o desenvolvimento do projeto.

4.2 Características

Com a instanciação do *framework* conceitual explicada, seguem as características levantadas. Elas são julgadas relevantes para a seleção da ferramenta, sendo explicitadas a seguir.

A primeira característica é a técnica disponibilizada pela ferramenta, que pode ser Lexical ou Aprendizado de Máquina (AM), ambas descritas no Capítulo 2.

A segunda característica é o Domínio que a ferramenta será aplicada, e para isso existem 2 opções:

- Engenharia de Software (ES): quando a ferramenta é específica para o domínio de ES;
- Geral: a ferramenta não possui configuração de domínio, ou seja, não possui configuração específica de ES.

Já o tipo de entrada é uma característica que pode determinar a facilidade de utilização da ferramenta, estando dividida em:

- Texto: permite a entrada de um texto único por vez;
- Arquivo *.txt*: permite a entrada de um arquivo com vários textos para serem processados de uma vez;
- Repositório de Códigos (RC): a ferramenta permite extrair dados direto de um RC, como por exemplo o GitHub.

Outra característica é o tipo de saída da ferramenta, que é dividida em 2 tipos: polaridade e emoções. A saída de polaridade está subdividida em:

- Dual: é dada uma intensidade de positivo e negativo;
- Binária: a saída é “positivo” ou “negativo”;
- Ternária: a saída além de conter “positivo” e “negativo” possui o “neutro”;
- Escala: a saída é um valor, por exemplo, entre -1 e 1, onde -1 é negativo e 1 positivo;
- Probabilidade: é dado um valor de probabilidade de ser positivo, neutro e negativo.

Já a saída do tipo de emoções está subdividida como:

- 6 emoções básicas propostas por (EKMAN; DAVIDSON, 1994): raiva, desgosto, medo, alegria, tristeza e surpresa;
- 5 emoções básicas: são as 6 básicas exceto a emoção de surpresa, por poder causar ambiguidade, pelo fato dela poder em uma situação ser positiva e em outra negativa.

Tabela 4.1: Características das Ferramentas que apoiam a Análise de Sentimentos

		SentiStrength	SentiStrength-SE	Watson Natural Language Understanding	NLTK	Stanford NLP	
Técnica	Lexical	X	X	-	X	-	
	AM	-	-	X	-	X	
Domínio	ES	-	X	-	-	-	
	Geral	X	-	X	X	X	
Entrada	Texto	-	-	X	X	X	
	Arquivo .txt	X	X	-	-	-	
	RC	-	-	-	-	-	
Saída	Polaridade	Dual	X	X	-	-	
		Binária	X	X	-	-	
		Ternária	X	X	-	X	-
		Escala	X	X	X	-	X
		Probabilidade	-	-	-	X	-
	Emoções	6 básicas	-	-	-	-	-
		5 básicas	-	-	X	-	-

Para prover uma melhor visualização das características listadas acima e mostrar quais estão presente em cada ferramenta, foi criada a Tabela 4.1.

4.3 Seleção da Ferramenta

Para a seleção da ferramenta, há dois pontos principais para se levar em conta na escolha. Primeiramente, dado o objetivo do estudo que é construir uma ferramenta para auxiliar a Análise de Sentimentos dos desenvolvedores, em um ou mais projetos, e relacionar essa análise com outras métricas. Por exemplo, levando em conta a contribuição de cada desenvolvedor para o projeto, existe um cenário com um domínio específico, que é a área de Engenharia de Software. Além disso, é importante analisar a polaridade dos comentários dos desenvolvedores contidos no repositório de códigos.

Utilizando o *framework* e a Tabela 4.1 para escolher a ferramenta, a primeira característica analisada é a saída esperada. Como neste caso o intuito é analisar a polaridade do sentimento, ou seja, se é positivo, negativo ou neutro (ausência de sentimento), e não comparar a precisão das técnicas, todas as ferramentas permanecem nas possibilidades. Já na entrada, como o objetivo é analisar os comentários de desenvolvedores em um repositório de códigos, o ideal seria a própria ferramenta permitir a extração dos comentários direto dessa fonte de dados. Entretanto, não há nenhuma ferramenta que contenha esta característica, portanto a entrada por arquivo *.txt* foi escolhida, pelo motivo de facilitar a execução de testes em detrimento de utilizar um texto por vez, onde para contornar esta deficiência seria necessário usar a recursividade, para permitir a leitura de todos os textos

de uma única vez. Por fim, restaram duas ferramentas que englobam até o momento as características desejadas, a SentiStrength e SentiStrength-SE. A técnica utilizada se torna irrelevante neste caso, pelo simples fato de ambas as ferramentas utilizarem a mesma, a técnica Lexical. Por último, o domínio desejado para o que é proposto pelo estudo, é da área de Engenharia de Software e com isso a ferramenta escolhida foi a SentiStrength-SE.

Para sumarizar a escolha da ferramenta SentiStrength-SE e suas características selecionadas, formulou-se a Figura 4.2.

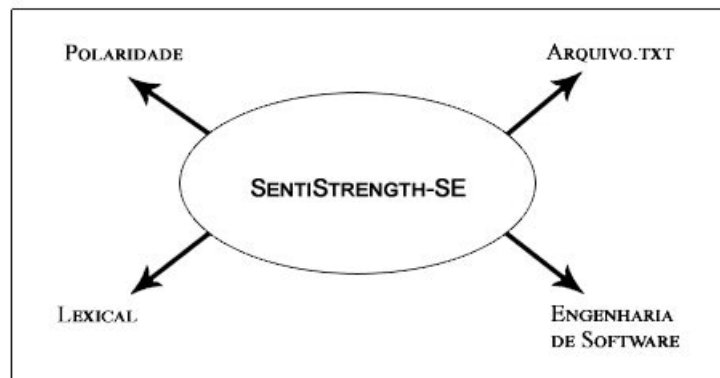


Figura 4.2: Sumarização das Características da Ferramenta Seleccionada

Sendo assim, a escolha do SentiStrength-SE neste estudo significa integrá-la a uma nova ferramenta desenvolvida a fim de apoiar a Análise de Sentimentos no contexto de desenvolvedores de software.

4.4 Considerações Finais do Capítulo

Este capítulo apresentou o *framework* conceitual elaborado para auxiliar a Análise de Sentimentos, ao apoiar a escolha da ferramenta mais adequada. A Tabela 4.1 foi construída para ajudar a visualizar as características levantadas para a seleção da ferramenta, e posteriormente uma análise foi realizada, onde chegou-se a conclusão que a ferramenta SentiStrength-SE deveria ser selecionada.

5 ASDS - Uma Ferramenta para Analisar Sentimentos

Este capítulo propõe discutir a implementação da nova ferramenta desenvolvida no contexto do trabalho, chamada ASDS, a partir da instanciação do *framework* conceitual e da ferramenta para Análise de Sentimentos escolhida, o SentiStrength-SE. A Seção 5.1 descreve as tecnologias utilizadas para a construção da ferramenta. Na Seção 5.2 é apresentado um exemplo de utilização da ferramenta, a partir de dados extraídos de um repositório de códigos. Por fim, a Seção 5.3 contém as considerações finais do capítulo.

5.1 Tecnologia Utilizada

Para a construção da ferramenta, utilizou-se a linguagem de programação orientada a objetos Java (ARNOLD; GOSLING; HOLMES, 2005). Para apoiar o desenvolvimento, utilizou-se o Apache Maven, ou apenas Maven (MILLER; VANDOME; MCBREWSTER, 2010), sendo uma ferramenta de automação e compilação com seu uso primário em projetos em java. A ferramenta tem a finalidade de reunir as tecnologias necessárias para a construção do projeto de uma maneira mais simples.

Para conectar o projeto com uma base de dados, sendo esta um repositório de códigos, integrou-se uma java API para o GitHub⁷ ao projeto. Para a conexão ocorrer, é necessário utilizar a chave de acesso, conhecida como *token*, da sua conta de usuário do GitHub para permitir um maior número de requisições a sua API. Isto possibilita a extração de dados de repositórios públicos, tais como *issues* e seus comentários, nome e login dos autores, a data da postagem, se a *issue* está aberta ou fechada, entre outras informações. No caso de uma *issue* aberta, significa que o problema ainda não foi resolvido e caso fechada o problema teoricamente já alcançou uma solução.

Os dados extraídos vem no formato javaScript object notation(JSON), um for-

⁷<https://github.com/kohsuke/github-api>

mato leve para transferência de dados. Portanto, são armazenados em um arquivo *.json*, onde a manipulação destes dados ocorre pela biblioteca java GSON⁸, desenvolvida pela Google. Pode ser usada para converter objetos em Java para a sua representação em JSON ou para converter uma JSON *string* para o objeto em java equivalente.

Com os dados obtidos, utilizou-se a ferramenta para Análise de Sentimentos SentiStrength-SE como uma biblioteca, a partir de seu arquivo *.jar* para integrá-la ao projeto, com a finalidade de executar uma análise polar sobre os dados extraídos a partir da java API para o GitHub.

Para gerar análises em relação aos valores polares obtidos, foram utilizadas expressões regulares. A fim de visualizar os resultados obtidos, uma nova biblioteca foi integrada, chamada de JFreeChart⁹ que permite a geração de gráficos.

5.2 Exemplo de Utilização

Para exemplificar o processo que ocorre ao utilizar o programa, são demonstradas nas seguintes subseções as etapas passo a passo. A Subseção 5.2.1 aborda a etapa da Base de Dados, onde os dados são extraídos de um repositório público. A Subseção 5.2.2 é apresentada a etapa de pré-processamento, onde os dados são manipulados de forma a serem utilizados no SentiStrength-SE. O processamento dos dados, abordado na Subseção 5.2.3, demonstra como a ferramenta analisa o texto a ser processado, determinando o valor polar de suas sentenças. Por fim, a Subseção 5.2.4 analisa de forma geral, as informações provenientes e adquiridas do repositório.

5.2.1 Base de Dados

A base de dados escolhida é proveniente do GitHub, sendo o repositório selecionado o Slim Framework¹⁰. A escolha se deu de forma aleatória, onde encontrou-se no GitHub um repositório, em busca de *issues*, onde cada uma tivesse o seu próprio número de comentários. O repositório escolhido tinha, no momento de sua extração 12.929 dados,

⁸<https://github.com/google/gson>

⁹<http://www.jfree.org/jfreechart/>

¹⁰<https://github.com/slimphp/Slim>

sendo considerado um bom número para se utilizar no exemplo de utilização.

Foi realizada a extração dos dados, que foram armazenados em um arquivo *.json*. Os dados extraídos se referem ao repositório, *issues* e comentários. É apresentado na lista a seguir, os dados referentes ao repositório, *issue* e comentário.

- **Repositório:**

- ***Title***: Nome do repositório;
- ***Description***: Descrição do repositório;
- ***Language***: Linguagem de programação principal do repositório;
- ***Issues***: *Issues* do repositório.

- ***Issue***:

- ***Body***: Texto descrevendo a *issue*;
- ***Title***: Título da *issue*;
- ***State***: Estado da *issue*, se está aberta ou fechada;
- ***CreatedAt***: : Data em que a *issue* foi criada;
- ***UpdatedAt***: Data em que a *issue* foi atualizada (caso não tenha sido atualizada, a data é a mesma em que foi criada);
- ***ClosedAt***: Data em que a *issue* foi fechada (caso não tenha sido fechada, não se busca tal informação);
- ***OpenedBy***: Login do autor da *issue*, de quem a abriu;
- ***NumberOfComments***: Número total de comentários referente à *issue*;
- ***Comments***: Comentários da *issue*.

- **Comentário:**

- ***Login***: Login do autor do comentário;
- ***CreatedAt***: Data em que o comentário foi postado;
- ***UpdatedAt***: Data em que o comentário foi atualizado (caso não tenha sido atualizado, a data é a mesma em que foi criado);

- **AuthorAssociation:** Maneira em que o usuário é associado ao repositório;
- **Body:** Texto descrevendo o comentário.

É importante ressaltar que no momento do armazenamento não houve qualquer tipo de pré-processamento dos dados, ou seja, estão no formato exato em que a API do GitHub retornou. A Listagem 5.1 demonstra um fragmento do formato do armazenamento dos dados extraídos. É possível notar uma *issue*¹¹ e seus respectivos dados, e os comentários pertencentes a esta *issue*, onde cada comentário tem os seus próprios dados.

```
1 ...
2 "issues": [
3   {
4     "body":"We need to tidy up our test suite for Slim 4...",
5     "title": "Test Suite Proposed Changes",
6     "state": "OPEN",
7     "createdAt": "Feb 19, 2019, 1:13:19 PM",
8     "updatedAt": "Feb 19, 2019, 2:33:19 PM",
9     "openedBy": "l0gicgate",
10    "numberOfComments": 4,
11    "comments": [
12      {
13        "login": "akrabat",
14        "createdAt": "Feb 19, 2019, 6:26:04 AM",
15        "updatedAt": "Feb 19, 2019, 6:26:04 AM",
16        "authorAssociation": "MEMBER",
17        "body": "Pimple should be mocked out for 4.x
18        testing."
19      },
20      {
21        "login": "akrabat",
22        "createdAt": "Feb 19, 2019, 6:30:29 AM",
23        "updatedAt": "Feb 19, 2019, 6:30:29 AM",
24        "authorAssociation": "MEMBER",
25        "body": "Agreed re PHPUnit upgrade."
26      },
27      ...
28    ]
29  },
30  ...
```

Listagem 5.1: Formato dos dados extraídos da API do GitHub

¹¹<https://github.com/slimphp/Slim/issues/2585>

5.2.2 Pré-Processamento

O pré-processamento é realizado a partir do arquivo *.json* com o propósito de convertê-lo em um arquivo *.txt*, pelo fato do SentiStrength-SE permitir como arquivo de entrada este formato. Porém, outras mudanças são realizadas nos dados. Uma delas é apenas o armazenamento do login do usuário, concatenado com sua respectiva *issue* e/ou comentário. Isto possibilita uma análise polar dos sentimentos dos textos e possibilita relacioná-los aos autores de maneira mais simples.

Outra modificação está relacionada à formatação do arquivo *.txt*. Para tanto, cria-se duas colunas. A primeira está relacionada ao número correspondente de cada texto, indo de 1 até 12.929 neste exemplo. A segunda coluna contém o login do autor e o seu texto.

Para uma melhor visualização, é retirada parte do conteúdo do arquivo *.txt* pré-processado, sendo selecionados alguns textos. Estes textos vão do número 13 ao 20, ou seja, 8 textos de um total de 12.929 proporcionado pelo arquivo. Ilustrado na Listagem 5.2.

```
1 13   akrabat: I agree that we can improve this. I'm not sure
2     about extending Router and App from a common class though.
3 14   l0gicgate: @akrabat we're not extending Router, that
4     component stays the same. What ...
5 15   skillaug: just fixing spelling errors
6 16   coveralls: [![Coverage
7     Status](https://coveralls.io/builds/21686958/badge)](https:// ...
8 17   l0gicgate: Closing as this is already fixed by #2555.
9     Thank you for your contribution @skillaug
10 18   bnf: ..which is (or may be) used for performing (real) sub
11     requests.
12 19   coveralls: [![Coverage
13     Status](https://coveralls.io/builds/21603956/badge)](
14     https://coveralls.io/builds/21603956)Coverage remained ...
15 20   bnf: Use the `$(callable)` syntax instead.
16     See https://trowski.com/2015/06/20/php-callable-paradox/
```

Listagem 5.2: Pré-processamento dos dados extraídos pela API do GitHub

5.2.3 Processamento

Tendo a posse do arquivo *.txt*, e com a correta manipulação dos dados, este está pronto para ser utilizado pelo SentiStrength-SE como dado de entrada, para a realização da

análise polar de seus textos. Desta forma, um novo arquivo *.txt* é gerado com os dados de saída, com um total de 4 colunas. A primeira coluna representa o valor positivo do texto, enquanto a segunda coluna representa o valor negativo do texto. O texto contendo a análise polar das palavras está presente na terceira coluna e, por fim, a quarta coluna apresenta a explicação, informando porque o texto é positivo, negativo ou neutro.

Portanto, para demonstrar os três possíveis resultados de saída da análise polar do SentiStrength-SE são apresentadas as saídas de cada um deles em listagens. Para um melhor entendimento, a formatação do arquivo *txt* será adaptado nas listagens, descritas a seguir. Os valores pertencentes a primeira e segunda coluna estão presentes na primeira linha. O conteúdo da terceira coluna está presente na segunda linha. Por fim, o texto da quarta coluna aparece na terceira linha.

É possível notar pela Listagem 5.3 que os valores positivo e negativo encontrados no texto são os mesmos, pelo fato de a comparação ser feita de forma modular, $|positivo| = |negativo|$ ($1 = |-1|$). Isto evidencia que o valor polar encontrado é neutro, revelando a ausência de sentimento, por não encontrar palavras que contenham sentimento.

```

1 1      -1
2 akrabat: Agreed re PHPUnit upgrade.
3 akrabat :Agreed re PHPUnit upgrade. [sentence: 1,-1] [result: max +
   and - of any sentence][trinary result = 0 as pos=1 neg=-1]

```

Listagem 5.3: Exemplo de um texto com sentimento neutro

Na Listagem 5.4, o valor positivo é maior do que o valor negativo no texto, $|positivo| > |negativo|$ ($3 > |-1|$), pela palavra *fine* ter um valor polar de +3. Portanto, é evidenciado que o valor polar encontrado é positivo.

```

1 3      -1
2 l0gicgate: This is a proposed fix for #2362 @akrabat I'm not sure if
   you're okay with using '@mcrypt_' to silence the warnings since of
   the behavior in 7.1 (7.2 is fine).
3 l0gicgate :This is a proposed fix for #2362 @akrabat I'm not sure if
   you're okay with using '@mcrypt_' to silence the warnings since of
   the behavior in 7 .[sentence: 1,-1] 1 (7 .[sentence: 1,-1] 2 is
   fine[3] ).[sentence: 3,-1] [result: max + and - of any sentence][
   overall result = 1 as pos>-neg]

```

Listagem 5.4: Exemplo de um texto com sentimento positivo

Por fim, nota-se na Listagem 5.5 que o valor positivo é menor do que o valor

negativo no texto, $|positivo| < |negativo|$ ($1 < |-2|$), por conta da palavra *stability* ter um valor polar de -2 . Com isto, o valor polar final é dado como negativo.

```

1 1      -2
2 briannesbitt: Removed minimum stability setting... there are no
  requirements on other packages.This setting dictates the stability
  level of packages you require.  Read it as "are you ok with
  composer installing dev level packages".  Users of slim would put
  this in their composer.json file if they wanted to access dev
  packages of slim, like "dev-master".
3 briannesbitt :Removed minimum stability[-2] setting ...[sentence:
  1,-2] there are no requirements on other packages .[sentence:
  1,-1] This setting dictates the stability[-2] level of packages
  you require .[sentence: 1,-2] Read it as 'are you ok with composer
  installing dev level packages '.[sentence: 1,-1] Users of slim
  would put this in their composer .[sentence: 1,-1] json file if
  they wanted to access dev packages of slim ,like 'dev -master '.[
  sentence: 1,-1] [result: max + and - of any sentence][overall
  result = -1 as pos<-neg]
```

Listagem 5.5: Exemplo de um texto com sentimento negativo

Com isso, o arquivo está corretamente processado, tendo todos os dados necessários para assim realizar uma análise a partir deles. Tal análise é apresentada na seção a seguir.

5.2.4 Análise dos Resultados

A partir dos dados obtidos na etapa anterior, algumas análises são executadas com a finalidade de angariar um maior conhecimento sobre o repositório e seus usuários. Tais análises são demonstradas e explicitadas a seguir. Porém, é importante ressaltar que para cada análise a ser executada, é necessário carregar o arquivo *.txt* desejado para cada uma delas.

A primeira análise é relacionada à frequência de palavras com um valor polar em relação às *issues* e seus respectivos comentários. Uma seleção das 10 palavras com maior frequência são apresentadas, como pode ser visto na Figura 5.1. Nela é possível observar que a palavra *thanks* possui a maior frequência encontrada.

Outro ponto a se notar nesta análise é a presença de palavras tanto com valor polar positivo quanto negativo na Figura 5.1. Entretanto, caso necessário é possível selecionar apenas as palavras com um valor positivo ou negativo. Há um total de três opções,

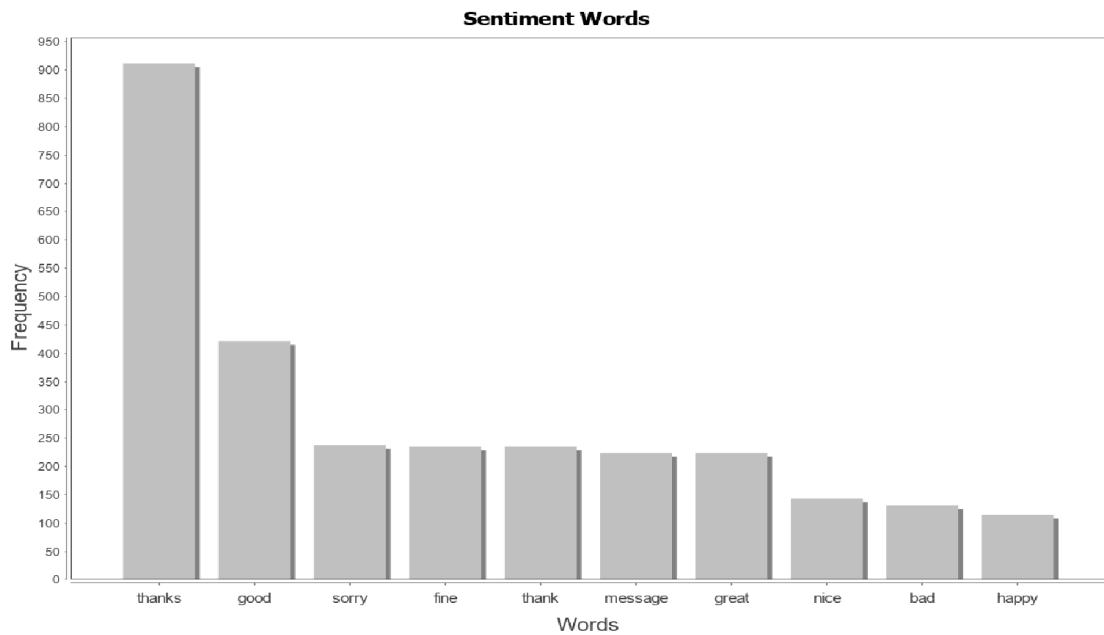


Figura 5.1: As 10 palavras com maior frequência no repositório

Positive, *Negative*, *All*. Selecionando a primeira opção, apenas palavras com um valor positivo aparecem no gráfico, mas caso a segunda opção seja escolhida somente as palavras com um valor negativo estão presentes. Ao ser selecionada a terceira opção, palavras tanto positivas quanto negativas compõem o gráfico, como visto na Figura 5.1. Para exemplificar tais opções, a Figura 5.2 apresenta a opção *Positive* selecionada e contendo assim apenas palavras com valores positivos.

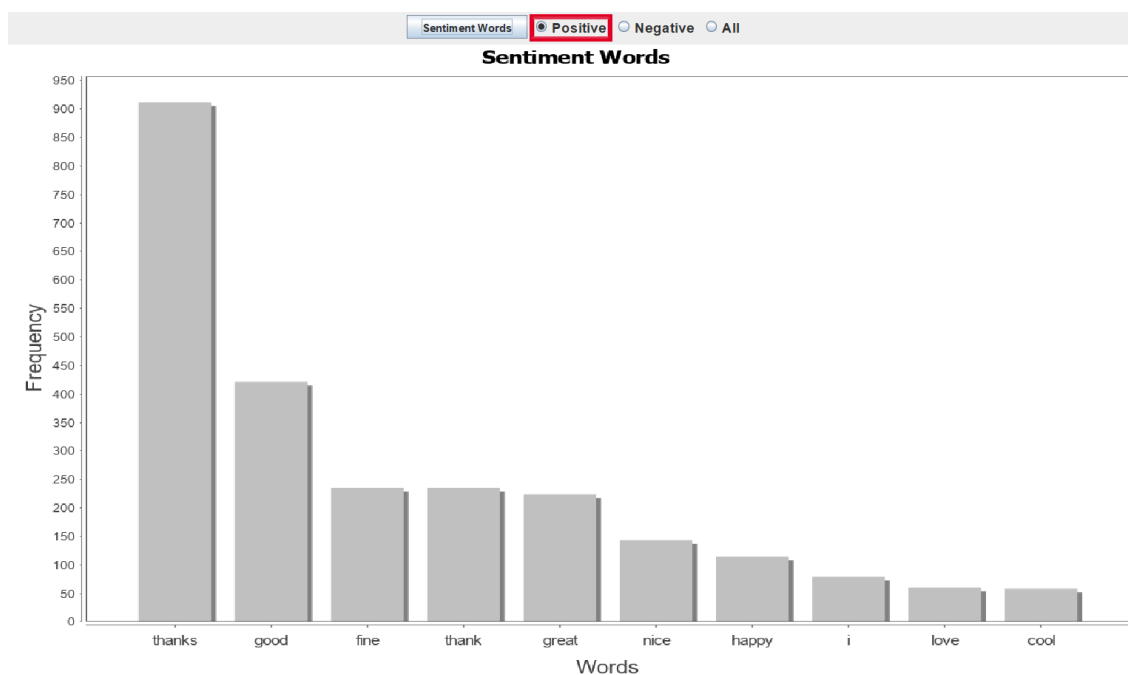


Figura 5.2: As 10 palavras positivas com maior frequência no repositório

Nesta primeira análise, levando em conta a Figura 5.1, é possível perceber que as palavras positivas possuem uma frequência consideravelmente maior que as negativas. É possível perceber isso comparando a frequência das palavras *thanks* e *good* com *bad*, por exemplo, dando a entender que há um maior sentimento positivo do que negativo no repositório, algo a ser confirmado pelas próximas análises.

A segunda análise, mostrada na Figura 5.3, apresenta a relação entre os usuários e o número de textos postados por eles em *issues* e seus respectivos comentários. É selecionado um total de 10 usuários com o maior número de textos (*issues* e/ou comentários escritos referente a cada usuário). Isto denota a importância deles para o repositório, pois representa uma maior relevância para o mesmo por terem uma maior presença.

É possível perceber pela Figura 5.3, que os dois principais usuários são akrabat e codeguy, baseado no fato de terem os maiores números de textos dentre o grupo de 10, portanto os mais importantes do repositório. Em vista disso, é possível perceber que estas análises com relação aos sentimentos de usuários deverão levar em conta os dois pelo menos.

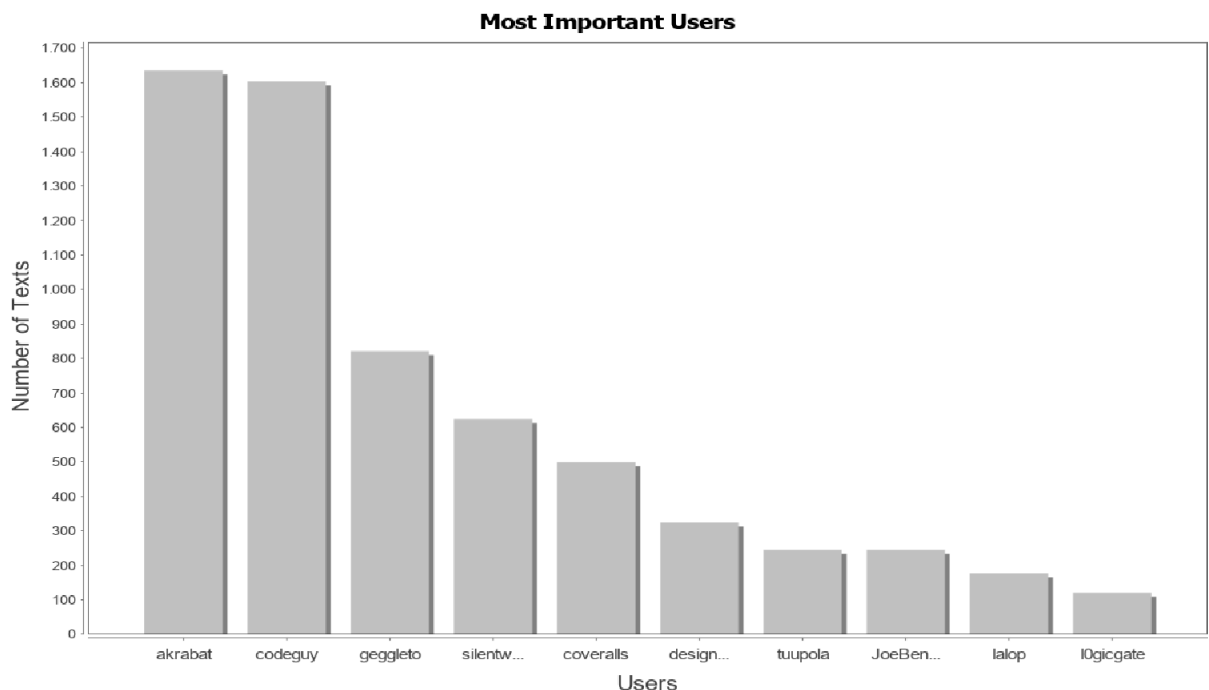


Figura 5.3: Os 10 usuários mais importantes e seus respectivos números de textos escritos no repositório

A terceira análise apresenta as palavras encontradas nos textos que possuem um sentimento, tanto positivo quanto negativo, e os seus respectivos valores polares. É

possível notar pela Tabela 5.1, uma grande quantidade de palavras com sentimentos. Porém, é perceptível que boa parte delas não possuem uma frequência grande o bastante de aparição nos textos, levando em conta o resultado encontrado na Figura 5.1.

Tabela 5.1: Palavras e seus respectivos valores polares

#	<i>Words</i>	<i>Value</i>
1	<i>funny</i>	4.0
2	<i>wow</i>	4.0
...
3	<i>fine</i>	3.0
4	<i>love</i>	3.0
...
5	<i>favorite</i>	2.0
...
6	<i>fools</i>	-2.0
7	<i>bother</i>	-2.0
...
8	<i>evil</i>	-3.0
9	<i>ugly</i>	-3.0
10	<i>terrible</i>	-3.0
...

A quarta análise diz respeito aos 10 principais usuários, encontrados anteriormente na segunda análise, que distingue os sentimentos de cada um em seus respectivos textos. Tais sentimentos podem ser positivos, negativos ou neutros. A Figura 5.4 demonstra os resultados encontrados desta análise.

É possível notar pela Figura 5.4 que na maior parte dos casos o número de textos com ausência de sentimento (neutro) são maiores do que os sentimentos positivos e negativos. A única exceção neste caso é o do usuário *coveralls*, que apresenta apenas sentimentos positivos, algo raro e destoante dos demais.

Outro resultado a se perceber, sem exceção, são os sentimentos positivos de cada usuário serem maiores que seus respectivos sentimentos negativos. Isto pode demonstrar uma possível relação com o resultado encontrado na primeira análise, onde é possível perceber uma maioria de palavras positivas levando em conta todas as palavras, considerando tanto sentimentos positivos quando negativos.

A quinta análise alude a frequência de 10 palavras referentes aos 4 principais usuários do repositório. Essas 10 palavras não foram escolhidas de forma aleatória, mas sim pela importância para um ou mais usuários através de suas frequências. Os qua-

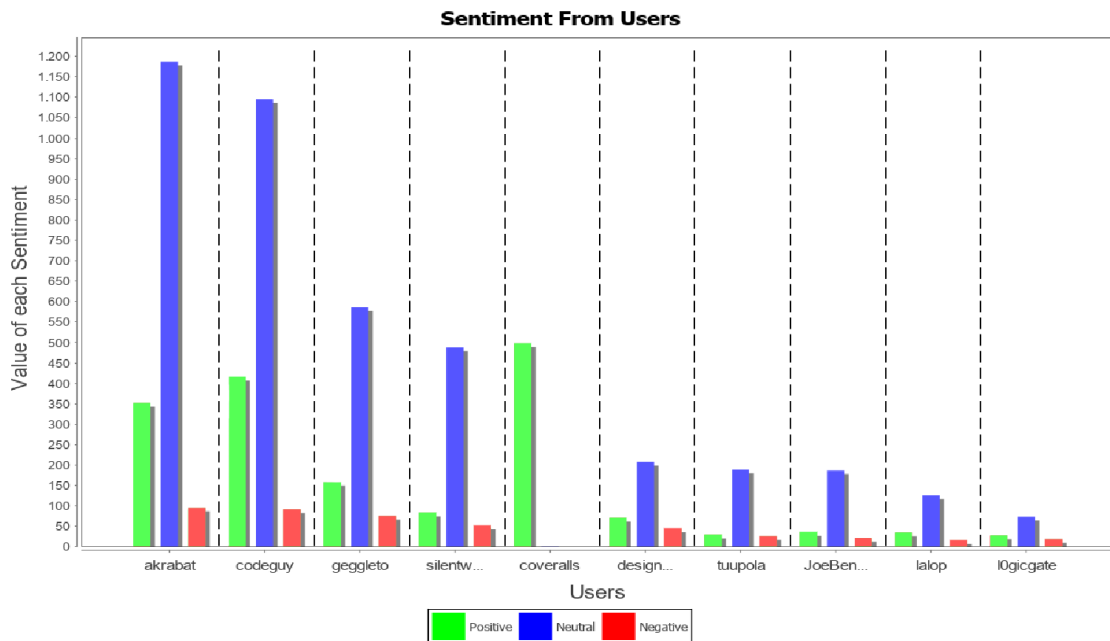


Figura 5.4: Os 10 principais usuários e seus respectivos sentimentos distinguidos (positivo, neutro e negativo)

tro principais usuários são codeguy, akrobat, geggleto e silentworks, levando em conta o resultado mostrado na Figura 5.3. A Figura 5.5 demonstra o resultado obtido.

É possível notar que os usuários possuem palavras em comum, sendo a única compartilhada por todos é a palavra *bad*, que possui um sentimento negativo. Outro ponto é a alta frequência da palavra *happy* pelo usuário codeguy, corroborando para um alto número de textos com valor positivo. Conhecendo a frequência das palavras dos usuários permite conhecer o hábito de cada um, se tem mais palavras positivas ou negativas em seus textos.

5.3 Considerações Finais do Capítulo

Este capítulo descreveu as funcionalidades da ferramenta ASDS desenvolvida para este trabalho, onde explicitou as etapas de seu funcionamento. As etapas descritas são, base de dados, pré-processamento, processamento de dados e por fim análise de resultados.

Acredita-se que essa ferramenta tem a capacidade de apoiar a Análise de Sentimentos, demonstrando assim o seu valor e a sua importância na área. Tais conclusões são corroboradas pelo exemplo de utilização mostrado.

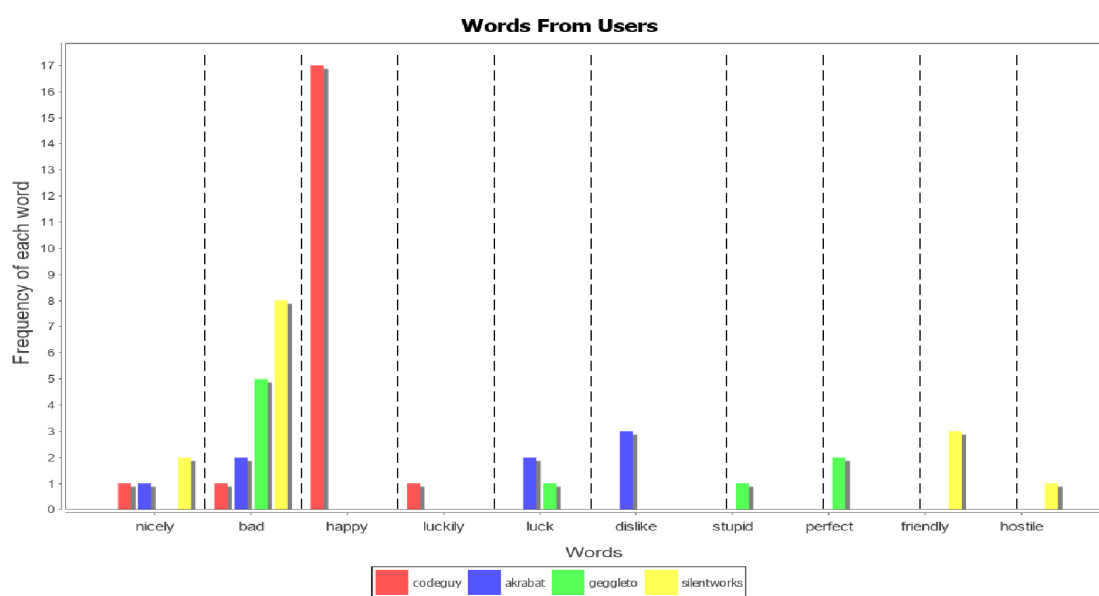


Figura 5.5: As 10 principais palavras dos 4 principais usuários

6 Conclusão

Neste trabalho foram vistos conceitos necessários para a sua compreensão da Análise de Sentimentos e as abordagens existentes, o Método Lexical e suas demais variações e o Aprendizado de Máquina. Abordou o que é um *framework* conceitual e as principais ferramentas de Análise de Sentimentos existentes.

Foi apresentado o mapeamento sistemático da literatura, realizado com o objetivo de identificar e obter uma visão geral das técnicas, métodos, modelos e ferramentas ligadas a Análise de Sentimentos no contexto de desenvolvedores de software. Para tal, foram elaboradas 7 questões de pesquisas e uma *string* de busca. Posteriormente, selecionou-se quatro bibliotecas digitais. Os critérios de exclusão foram elaborados e depois os artigos de controle escolhidos. Por fim, os resultados obtidos pelo mapeamento sistemático da literatura foram descritos e analisados.

O *framework* conceitual foi explanado, desde sua instanciação até a seleção de qual ferramenta de Análise de Sentimentos foi escolhida para ser possível integrá-la na ferramenta ASDS. Para a seleção, ocorreu uma comparação entre as 5 ferramentas, explicadas no Capítulo 2, levando em conta os pontos positivos e negativos de cada uma.

Por fim, foi especificada a ferramenta desenvolvida no contexto desse trabalho, a ASDS. A descrição levou em conta a linguagem de programação utilizada e quais bibliotecas foram integradas. As etapas são de base de dados, pré-processamento, processamento e, por fim, análise dos resultados.

6.1 Resultados Obtidos

Foram obtidos dois principais resultados que agregam para a comunidade científica. O primeiro, uma visão ampla da área de Análise de Sentimentos no contexto de software por conta do mapeamento sistemático da literatura realizado na área. Com isso, há conhecimento para saber o que foi pesquisado, além de proporcionar possíveis novos caminhos para futuros trabalhos. Este objetivo gerou a publicação do seguinte trabalho:

- COELHO, A. et al. Software developers sentiment analysis: A systematic mapping. In:IEEE.Computer Conference (CLEI), 2018 XLIV Latin American. [S.l.], 2018.

Outro resultado obtido foi o desenvolvimento da ferramenta ASDS, com o intuito de apoiar a Análise de Sentimentos no contexto de desenvolvedores de software. Além de proporcionar um local para a extração de dados provenientes de uma base de dados de um repositório como o GitHub, é possível rearranjar os dados de forma a servirem como dados de entrada no SentiStrength-SE. Após isso, ainda é possível realizar uma análise a partir da polaridade, os valores polares, das expressões de texto contidas do repositório. As possíveis análises são: encontrar as 10 principais palavras com sentimento presentes no repositório, podendo levar em conta somente as positivas, as negativas ou ambas. Descobrir os 10 principais usuários do repositório, aqueles que mais possuem textos. Os valores e as palavras que apresentam sentimento. A distinção dos sentimentos, sendo eles positivos, negativos ou neutros, nos textos dos 10 principais usuários. E, por fim, as principais palavras com sentimento expressas pelos 4 principais usuários do repositório.

Porém, é importante frisar que a ferramenta possui as suas limitações, sendo uma delas o tempo de extração de dados provenientes do GitHub. Para não ser sobrecarregado com um alto número de requisições através de usuários, o GitHub estipula um total de requisições por hora, sendo de 5.000. Assim, caso o número de dados a se extrair de um repositório de códigos for alto, pode-se exigir um maior número de requisições, tendo que esperar o tempo necessário para reiniciar a contagem para poder continuar a realizar a coleta de dados. Portanto, o tempo para a execução desta tarefa pode ser longo. Outra limitação seria o formato de saída das análises, onde não é permitido por parte do usuário a sua customização para gerar gráficos de acordo com suas próprias especificações.

6.2 Trabalhos Futuros

Uma primeira sugestão para trabalho futuro é realizar um *snowballing*¹² (WOHLIN, 2014) dos artigos encontrados pelo mapeamento sistemático da literatura da Análise de Sentimentos no contexto de desenvolvedores de software. O mapeamento sistemático foi exe-

¹²Refere-se a usar a referência bibliográfica de um trabalho ou as citações ao trabalho para encontrar outros trabalhos, e assim auxiliar o mapeamento sistemático

cutado no início do ano de 2018. Desde desta data outros trabalhos foram publicados e portanto não fazem parte deste estudo.

Outra sugestão de um trabalho futuro é a realização de novas análises dos dados obtidos, com a finalidade de obter mais conhecimento sobre repositórios. Isto servirá para entender melhor o ambiente e como os usuários interagem uns com os outros.

Por fim, cabe minerar dados à procura de padrões consistentes, como por exemplo procurar similaridades entre dois repositórios diferentes, detectando assim um novo subconjunto de dados. Tais padrões poderiam levar em conta o número total de *issues* e comentários, os sentimentos encontrados por seus usuários e as principais palavras com sentimentos presentes nos repositórios.

Bibliografia

- AHMED, T. et al. Senticr: A customized sentiment analysis tool for code review interactions. In: IEEE PRESS. *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. [S.l.], 2017. p. 106–111.
- ARNOLD, K.; GOSLING, J.; HOLMES, D. *The Java programming language*. [S.l.]: Addison Wesley Professional, 2005.
- BHADANE, C.; DALAL, H.; DOSHI, H. Sentiment analysis: Measuring opinions. *Procedia Computer Science*, Elsevier, v. 45, p. 808–814, 2015.
- BIRD, S. Nltk: the natural language toolkit. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the COLING/ACL on Interactive presentation sessions*. [S.l.], 2006. p. 69–72.
- CALEFATO, F. et al. Sentiment polarity detection for software development. *Empirical Software Engineering*, Springer, p. 1–31, 2017.
- CHRISTOPHER, M. B. *Pattern Recognition and Machine Learning*. [S.l.]: Springer-Verlag New York, 2016.
- COELHO, A. et al. Software developers sentiment analysis: A systematic mapping. In: IEEE. *Computer Conference (CLEI), 2018 XLIV Latin American*. [S.l.], 2018. p. 1–2.
- DYER, R. et al. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In: IEEE PRESS. *Proceedings of the 2013 International Conference on Software Engineering*. [S.l.], 2013. p. 422–431.
- EKMAN, P. E.; DAVIDSON, R. J. *The nature of emotion: Fundamental questions*. [S.l.]: Oxford University Press, 1994.
- FARIAS, M. A. de F. et al. A systematic mapping study on mining software repositories. In: ACM. *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. [S.l.], 2016. p. 1472–1479.
- GOYAL, A.; SARDANA, N. Nrfixer: Sentiment based model for predicting the fixability of non-reproducible bugs. *e-Informatica Software Engineering Journal*, v. 11, n. 1, 2017.
- GUZMAN, E.; BRUEGGE, B. Towards emotional awareness in software development teams. In: ACM. *Proceedings of the 2013 9th joint meeting on foundations of software engineering*. [S.l.], 2013. p. 671–674.
- ISLAM, M. R.; ZIBRAN, M. F. Leveraging automated sentiment analysis in software engineering. In: IEEE PRESS. *Proceedings of the 14th International Conference on Mining Software Repositories*. [S.l.], 2017. p. 203–214.
- JABAREEN, Y. Building a conceptual framework: philosophy, definitions, and procedure. *International Journal of qualitative methods*, SAGE Publications Sage CA: Los Angeles, CA, v. 8, n. 4, p. 49–62, 2009.

- JAIN, A. P.; DANDANAVAR, P. Application of machine learning techniques to sentiment analysis. In: IEEE. *Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on*. [S.l.], 2016. p. 628–632.
- JONGELING, R.; DATTA, S.; SEREBRENIK, A. Choosing your weapons: On sentiment analysis tools for software engineering research. In: IEEE. *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. [S.l.], 2015. p. 531–535.
- JONGELING, R. et al. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, Springer, v. 22, n. 5, p. 2543–2584, 2017.
- JURADO, F.; RODRIGUEZ, P. Sentiment analysis in monitoring software development processes: An exploratory case study on github project issues. *Journal of Systems and Software*, Elsevier, v. 104, p. 82–89, 2015.
- KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004.
- KITCHENHAM, B. A.; MENDES, E.; TRAVASSOS, G. H. Cross versus within-company cost estimation studies: A systematic review. *IEEE Transactions on Software Engineering*, IEEE, v. 33, n. 5, 2007.
- LOVINS, J. B. Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics*, v. 11, n. 1-2, p. 22–31, 1968.
- MILLER, F. P.; VANDOME, A. F.; MCBREWSTER, J. Apache maven. Alpha Press, 2010.
- NOVIELLI, N.; CALEFATO, F.; LANUBILE, F. The challenges of sentiment detection in the social programmer ecosystem. In: ACM. *Proceedings of the 7th International Workshop on Social Software Engineering*. [S.l.], 2015. p. 33–40.
- ORTU, M. et al. The jira repository dataset: Understanding social aspects of software development. In: ACM. *Proceedings of the 11th international conference on predictive models and data analytics in software engineering*. [S.l.], 2015. p. 1.
- ORTU, M. et al. The emotional side of software developers in jira. In: IEEE. *Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on*. [S.l.], 2016. p. 480–483.
- PAI, M. et al. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. *The National medical journal of India*, v. 17, n. 2, p. 86–95, 2004.
- PATWARDHAN, A. Sentiment identification for collaborative, geographically dispersed, cross-functional software development teams. In: IEEE. *Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on*. [S.l.], 2017. p. 20–26.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *EASE*. [S.l.: s.n.], 2008. v. 8, p. 68–77.
- PLETEA, D.; VASILESCU, B.; SEREBRENIK, A. Security and emotion: sentiment analysis of security discussions on github. In: ACM. *Proceedings of the 11th working conference on mining software repositories*. [S.l.], 2014. p. 348–351.

- REGONIEL, P. Conceptual framework: A step by step guide on how to make one. Retrieved from *SimplyEducate*. Me: <http://simplyeducate.me/2015/01/05/conceptual-framework-guide>, 2015.
- ROBINSON, W. N.; DENG, T.; QI, Z. Developer behavior and sentiment from data mining open source repositories. In: IEEE. *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. [S.l.], 2016. p. 3729–3738.
- ROSSUM, G. V.; JR, F. L. D. *Python tutorial*. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
- ROUSINOPOULOS, A.-I.; ROBLES, G.; GONZÁLEZ-BARAHONA, J. M. Sentiment analysis of free/open source developers: Preliminary findings from a case study. *Revista Electronica de Sistemas de Informação*, Faculdade Cenecista de Campo Largo-FACECLA, v. 13, n. 2, p. 1, 2014.
- SINHA, V.; LAZAR, A.; SHARIF, B. Analyzing developer sentiment in commit logs. In: ACM. *Proceedings of the 13th International Conference on Mining Software Repositories*. [S.l.], 2016. p. 520–523.
- SOCHER, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2013. p. 1631–1642.
- SOUZA, R.; SILVA, B. Sentiment analysis of travis ci builds. In: IEEE PRESS. *Proceedings of the 14th International Conference on Mining Software Repositories*. [S.l.], 2017. p. 459–462.
- THELWALL, M. et al. Sentiment strength detection in short informal text. *Journal of the Association for Information Science and Technology*, Wiley Online Library, v. 61, n. 12, p. 2544–2558, 2010.
- TUMASJAN, A. et al. Predicting elections with twitter: What 140 characters reveal about political sentiment. In: *Fourth international AAAI conference on weblogs and social media*. [S.l.: s.n.], 2010.
- VOUTILAINEN, A. Part-of-speech tagging. *The Oxford handbook of computational linguistics*, Oxford University Press Oxford, p. 219–232, 2003.
- WILSON, T.; WIEBE, J.; HOFFMANN, P. Recognizing contextual polarity in phrase-level sentiment analysis. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the conference on human language technology and empirical methods in natural language processing*. [S.l.], 2005. p. 347–354.
- WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: CITESEER. *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. [S.l.], 2014. p. 38.