

IssueQuality: Uma extensão para a análise da qualidade do relato de uma issue

Objetivo

O objetivo da extensão é analisar de maneira automatizada a qualidade do relato de uma *issue*¹ em repositórios do GitHub. O resultado da análise é um conjunto de dicas para o responsável por redigir a *issue* com o intuito de melhorar a informação fornecida no relato.

Funcionamento

O funcionamento da extensão consiste em criar de forma automatizada um comentário em uma nova issue com um conjunto de dicas que visam melhorar a qualidade da informação fornecida. O comentário é produzido com base no texto do relato o qual é avaliado com base nos seguintes atributos:

- *Etapas para Reproduzir*: Verifica se reportador incluiu uma lista, na forma de itens, descrevendo as etapas executadas até a ocorrência da falha.
- *Arquivos Anexados*: Avalia a existência de arquivos anexados à issue, como por exemplo screenshots ou stack trace.
- *Fragmentos de Código*: Verifica se algum fragmento de código foi adicionado ao relato da issue.
- *Compleitude de Palavras-Chaves*: Utilizando o conjunto de dados fornecido por Andy Ko e outros² foi construído a distribuição de frequências das palavras que ocorrem em uma issue. Em uma primeira etapa, removemos as palavras de parada (stopwords), reduzimos as palavras e selecionamos as 100 palavras com maior frequência. Em seguida, categorizamos as palavras escolhidas nos seguintes grupos:
 - action items (doe, work, open)
 - build-related (build, task)
 - documentation-related (support, help, content)
 - expected and observed behavior (fail, error, crash)
 - project-related (management, list)
 - source code-related (java, code, method)
 - user interface elements (menu, display, button)

¹No contexto dos repositórios do Github uma Requisição de Mudança (RM) recebe o nome de issue.

²A. J. Ko, B. A. Myers, and D. H. Chau. A linguistic analysis of how people describe software problems. In Proceedings of the 2006 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2006), pages 127–134, 2006

- *Legibilidade do Texto:* Avalia o nível legibilidade do texto com base em testes já existentes na literatura. Neste estudo utilizamos os testes de legibilidade Flesch–Kincaid, Automated Readability Index - ARI e Dale–Chall Readability Formula. Os testes foram selecionados por apresentarem metodologias distintas para determinar a legibilidade do texto. O Flesch–Kincaid é baseado no número de sílabas das palavras que compõem as sentenças do texto. O ARI considera o número de caracteres de cada palavra. Por outro lado, o teste Dale–Chall é baseado em um conjunto mínimo de palavras. Para o texto Flesch–Kincaid a legibilidade será considerada baixa para uma pontuação *menor do 50*. Para os outros testes, a legibilidade será considerada ruim se o número de anos de estudos necessário para o entendimento *for maior ou igual a 13*, conforme utilizado no trabalho de Bettenburg e outros³.

Para cada issue analisada a extensão cria um *vetor de características* que armazena uma pontuação para cada um dos itens listados anteriormente. Estes podem ser binários (por exemplo, anexo presente ou não) ou contínuo (por exemplo, legibilidade do texto). A análise dos atributos utilizam da sintaxe da linguagem de marcação Markdown, que é o padrão para as issues dos repositórios no GitHub.

O comentário produzido é composto de três partes: *cabeçalho*, *corpo* e *dicas*. O cabeçalho apresenta um texto padrão que é personalizado com o nome do usuário (login) no Github do reportador. Utilizando esta sintaxe o próprio Github se encarrega de enviar um e-mail notificando o usuário sobre o comentário. O formato do cabeçalho pode ser visualizado na Figura 1.

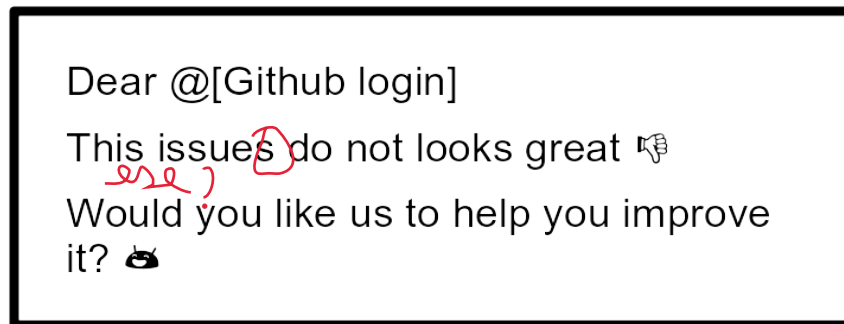


Figure 1: Cabeçalho padrão do comentário

Ao final do comentário é incluído um conjunto de dicas com objetivo de apresentar ao reportador os benefícios que a melhoria da qualidade do relato pode ter na

³Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R., & Zimmermann, T. (2008, November). What makes a good bug report?. In Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering (pp. 308-318). ACM.

solução de sua *issue*, como por exemplo dizendo que issues que são mais fáceis de serem lidas possuem um tempo de solução menor. Estas dicas foram obtidas de trabalhos acadêmicos sobre o assunto, especialmente o trabalho Bettenburg e outros⁴. A Figura 2 exibe o conteúdo das dicas que fazem parte do comentário.

Did you know:

- Issues containing *stack traces* get fixed sooner.
- Issues that are *easier to read* have lower lifetimes.
- Including *code samples* in your issue increases the chances of it getting fixed.

O corpo é parte mais dinâmica do comentário. Ele é construído incluindo fragmentos de texto quando certos critérios de aceitação não foram atendidos. Por exemplo, caso não seja detectado a presença de “*etapas para reproduzir*” no relato de uma issue o seguinte fragmento de texto é incluído no corpo do comentário: *Add step to reproduce*. A tabela a seguir resume os critérios de aceitação e os fragmentos de texto incluídos para cada um dos atributos avaliados na *issue*. O exemplo de utilização da extensão pode ser visualizado no repositório *vagnerclementino/flask* clique aqui.

Atributo	Critério de Aceitação	Forma de Análise	Mensagem
Etapas para Reproduzir	Existência de pelo menos uma lista na forma itemizada representando as etapas executadas até ocorrência do erro.	Expressão Regular	Add step to reproduce.
Arquivos Anexados	Pelo menos um arquivo anexado a issue.	Expressão Regular	To Attach files with a screenshots, or stacktraces.
Fragmento de Código	Existência de pelo menos um fragmento de código no relato da issue.	Expressão Regular	To include a code block.

⁴Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R., & Zimmermann, T. (2008, November). What makes a good bug report?. In Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering (pp. 308-318). ACM.

Atributo	Critério de Aceitação	Forma de Análise	Mensagem
Completeness de relato da issue Palavras-Chaves	As palavras que compõem o deve fazer parte de pelo menos duas das categorias que foram utilizadas para análise.	Busca e Categorização	To include a code block.
Legibilidade do Texto	Dois dos três testes de legibilidade apresentarem valores acima dos limites definidos	Testes Flesch–Kincaid, Automated readability index e Dale–Chall readability formula	To improve readability of text.