UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

WILLYAN SCHULTZ DWORAK

UMA PERSPECTIVA SOCIAL PARA RECOMENDAÇÃO DE *EXPERTS* EM PROJETOS DE SOFTWARE LIVRE

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO - PR

2015

WILLYAN SCHULTZ DWORAK

UMA PERSPECTIVA SOCIAL PARA RECOMENDAÇÃO DE *EXPERTS* EM PROJETOS DE SOFTWARE LIVRE

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Filipe Roseiro Côgo

Co-orientador: Prof. Me. Igor Scaliante Wiese

CAMPO MOURÃO - PR

2015

Dedico esta monografia a todas as pessoas que de alguma forma me ajudaram e principalmente à minha família e demais pessoas especiais que sempre estiveram presentes me mantendo motivado a buscar novos conhecimentos.

AGRADECIMENTOS

Agradeço a Deus pela minha consciência iluminada e por me ajudar a persistir mesmo quando tudo se apresenta sem rumo. Agradeço também a todos os seres de outras dimensões que independentemente de estar ou não nessa realidade que temos como Universo, também colaboram comigo de alguma forma.

À todos os professores da Universidade que de alguma forma contribuíram com meu crescimento. Ao meu orientador Prof. Me. Felipe Roseiro Côgo que sempre esteve presente colaborando com seus conhecimentos e me auxiliando no desenvolvimento dessa pesquisa.

Ao Prof. Me. Igor Scaliante Wiese, que além de orientador também sempre foi um amigo. Sempre apoiando e dando opiniões em diversas áreas. Aconselhando a continuar na caminhada, independentemente de horários e dias da semana, todas as vezes que precisei esteve sempre presente dando forças e mostrando que mesmo que a conclusão da pesquisa fosse aparentemente impossível, teria sim um caminho para chegar ao final da jornada.

Aos meus pais Adão Arcides Dworak e Olga dos Santos Schultz Dworak que mesmo sem terem cursado uma faculdade, são para mim, um exemplo de vida, esforço e dedicação. São duas pessoas que nunca mediram esforços para me ajudar. Sempre que preciso estão presentes, me incentivando e motivando a estar sempre estudando e aprendendo coisas novas, sempre com a frase “Conhecimento não ocupa espaço! Tudo que você aprender é bem-vindo”.

À minha irmã Elaine Schultz Dworak, que sempre está ao meu lado, me motivando e apoiando em todas as decisões que tomo em minha vida. Uma pessoa que sempre posso contar.

À minha amada Dayane Anderção Gomes, que independentemente da situação, está sempre comigo, me apoiando, me orientando, me ajudando a escolher os melhores caminhos a seguir e também me ajudando a ver as coisas com mais profundidade e atenção aos detalhes.

À minha tia Salete dos Santos Schultz, que ficou com o papel de minha segunda mãe, sempre dedicando seu tempo a todos de casa oferecendo seu apoio e carinho assim como minha mãe faz, além de suas sobremesas deliciosas.

Enfim, agradeço a todas as pessoas que independentemente da forma, seja apoiando ou criticando, me ajudaram de alguma maneira a chegar até aqui e poder concluir mais essa etapa da minha vida. Um grande abraço a todos.

RESUMO

DWORAK,Willyan Schultz. UMA PERSPECTIVA SOCIAL PARA RECOMENDAÇÃO DE *EXPERTS* EM PROJETOS DE SOFTWARE LIVRE. 49 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Campo Mourão - PR, 2015.

O alto desempenho e a qualidade no desenvolvimento de projetos estão cada vez mais em foco. Isso faz com que cresça a busca por *experts* em todas as áreas. Muitas vezes a falta de conhecimento de alguns desenvolvedores dificulta e atrasa o rendimento dos projetos de software livre. A fim de colaborar com a busca e recomendação de *experts*, elaborou-se este trabalho de conclusão de curso, que tem como objetivo analisar as redes sociais existentes com base na comunicação entre os desenvolvedores e assim poder recomendar as pessoas mais indicadas para ajudar participando na discussão de um problema relatado. Com o uso da abordagem proposta, pode ser possível aumentar o desempenho da equipe reduzindo o retrabalho e aumentando a colaboração entre os membros.

Palavras-chave: Recomendação, Expertise, Projetos de Software Livre, Algoritmos

ABSTRACT

DWORAK,Willyan Schultz. A SOCIAL PERSPECTIVE FOR RECOMMENDING EXPERTS IN OPEN SOURCE SOFTWARE PROJECTS . 49 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Campo Mourão - PR, 2015.

The high performance and quality in development projects are increasingly in focus. This causes grow the search for experts in all areas. Many times the lack of knowledge of some developers hinders and delays the performance of the open source software projects. In order to assist with the search and recommendation of the experts, was drawn up this work course conclusion, which aims to analyze existing social networks based on the communication between developers and thus be able to recommend the best people to help by participating in discussion of a reported problem. Using the proposed approach, it may be possible to increase the team's performance by reducing rework and increasing collaboration among members.

Keywords: Recommendation, Expertise, Open Source Software, Algorithms

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Encontrar *experts* em assuntos específicos é uma tarefa importante em projetos que envolvam a colaboração entre as pessoas. Em muitos casos, resolver um problema particular depende de encontrar pessoas que sejam capazes de responder a determinadas perguntas relevantes para a coordenação das pessoas no projeto (MCDONALD; ACKERMAN, 2000). No desenvolvimento de software, as pessoas que trabalham em tarefas relacionadas em alguns casos colaboram a fim de coordenar suas ações. Além disso, são necessários processos que organizam as atividades de desenvolvimento, com o objetivo de evitar retrabalho, erros e inconsistências. De modo geral, esta coordenação é moderada e apoiada por pessoas que possuem um maior conhecimento sobre temas-chave do projeto (BIRD et al., 2008).

Em projetos de software livre, devida à sua natureza aberta, não se têm uma estrutura formalmente organizada ou funções, no sentido de que ninguém é obrigado a trabalhar em uma parte pré-determinada do código ou entregar uma tarefa específica. Além disso, tornar-se um colaborador do núcleo do projeto depende de gastar uma quantidade significativa de tempo para aprender os aspectos sociais e técnicos do projeto e ainda demonstrar que possui habilidades e conhecimentos necessários, para com isso ganhar reputação e ser aceito no grupo central de desenvolvimento. Assim, o sucesso de um projeto de software livre depende amplamente da qualidade do projeto e da comunidade que o suporta (SETHANANDHA et al., 2010).

Para determinar a *expertise* de cada contribuidor, neste trabalho, serão considerados os aspectos sociais existentes no projeto. Para isso serão criadas redes de comunicação com base nas trocas de mensagens entre os contribuidores. Acredita-se que se um contribuidor comenta muito sobre determinado assunto e principalmente, se ele participa de discussões com outros contribuidores que possuem conhecimento nessa área, ele é a pessoa mais indicada a ajudar resolver um problema semelhante no futuro. Assim, ficou estabelecido que o vocabulário do contribuidor pode definir a sua área de interesses dentro do projeto.

Neste trabalho o termo *expert* será utilizado para representar os contribuidores mais indicados para comentar em uma determinada *issue*. Isso com base em seu vocabulário utilizado

dentro do projeto, que foi formado com base nas *issue* (tarefa a ser feita) que o contribuidor participou. Este trabalho não entrará no âmbito de determinar áreas de conhecimento de cada contribuidor fora do projeto que esteja sendo analisado, devido a amplitude do tema.

Nesse contexto, a localização de um *expert* em um projeto de software livre é um desafio, como apontado na literatura (MORAES et al., 2010). A grande quantidade de informações e a complexidade existente nesses projetos torna difícil para os colaboradores encontrar rapidamente a informação certa. Assim, tanto para o novato que acabou de ingressar em um projeto quanto para o gerente de um sistema complexo, encontrar a melhor pessoa para ajudar, por vezes, pode ser uma tarefa difícl (KAGDI; POSHYVANYK, 2009). Identificar e recomendar pessoas com o conhecimento certo para as pessoas em dificuldades durante a implementação pode melhorar a colaboração entre a equipe, porque isso pode reduzir o tempo de espera por uma resposta, uma vez que o *expert* no assunto pode ser contatado diretamente (MORAES et al., 2010).

As contribuições para projetos de software livre podem ser feitas por diferentes meios. Os colaboradores podem interagir uns com os outros por meio de de listas de discussão, podem reportar e discutir em *issues* especificas através de sistemas de *Issue Tracking* e cooperar uns com os outros submetendo artefatos em sistemas de controle de versão de código-fonte. Assim, implementar um algoritmo para identificação de *expertise* é uma tarefa complexa, pois requer o rastreamento e recriação dessas interconexões entre esses diferentes fontes de informação.

1.1 OBJETIVOS

Com a grande aceitação dos novos sistemas de hospedagem e gerenciamento de código-fonte, como GitHub e BitBucket, foi resolvida uma parcela dos problemas de rastreamento e recriação das interconexões entre as diferentes fontes de informação existentes sobre cada projeto de software. Assim, em um mesmo local estão as listas de discussão e o repositório de codigo-fonte facilitando a comunicação entre os contribuidores. Quando esses serviços são usados corretamente no projeto, eles permitem que os colaboradores criem links entre o sistema de *Issue Tracking* e os repositórios de código-fonte. Além disso, eles apresentam um recurso que permite que um membro do projeto mencione explicitamente outros membros em suas mensagens. Estas duas características são valiosas quando se trabalha com a identificação de *expertise*, pois assim, é possível reconstruir facilmente as conexões entre as diferentes fontes e permitir a criação de redes sociotécnicas mais ricas, (MOCKUS; HERBSLEB, 2002).

A fim de colaborar com o conhecimento sobre o tópico de recomendação de *experts*

em projetos de software livre, neste trabalho será apresentada uma abordagem que faz uso do algoritmo *PageRank With Priors* (PRP) (WHITE; SMYTH, 2003) aplicado à três modelos de redes de comunicação geradas a partir das interações entre os contribuidores dos projetos. Foram minerados dados de três projetos específicos, Bootstrap [1], NodeJS [2] e Homebrew [3]. A abordagem proposta considera que o vocabulário dos contribuidores, obtidos a partir dos comentários em *issues* nos sistemas de *Issue Tracking* podem mapear a área de *expertise* de cada um dos colaboradores. Considerando isso, o objetivo principal deste estudo é responder à seguinte questão de pesquisa:

> A aplicação do algoritmo *PRP* sobre uma Rede de Comunicação gerada a partir da comunicação entre os contribuidores de um projeto de software livre, é capaz de oferecer melhores resultados na recomendação de *experts* do que as abordagens que não consideram os aspectos sociais?

Para responder a esta pergunta, foram definidos quatro objetivos específicos que devem ser alcançados:

1. Modelar três redes de comunicação entre os contribuidores, considerando em quais *issues* esses contribuidores escrevem e o conteúdo que eles escrevem nessas *issues*;
2. Comparar as três redes de comunicação geradas para definir se o uso delas é capaz de auxiliar na recomendação de *experts*;
3. Comparar o uso de algoritmos que utilizem puramente *Term Frequency (TF)* e *Term Frequency – Inverse Document Frequency (TF-IDF)* com um algoritmo que incremente os *rankings* dos contribuidores através do valor de *PRP*;
4. Observar qual a influência que o fator tempo aplicado às redes de comunicação gera nos resultados da recomendação.

Para a realização deste trabalho, foi considerado o histórico da comunicação entre os contribuidores para determinar quais assuntos estão relacionados com cada um deles. Para isso, utilizou-se o conceito de redes de comunicação para a geração do vocabulário de cada contribuidor. Também foi considerada a influência do fator tempo para a indexação e recomendação dos contribuidores. Para avaliação foram utilizados os conceitos de *Recall* e *Precision* que serão

[1] https://github.com/twbs/bootstrap
[2] https://github.com/joyent/node
[3] https://github.com/Homebrew/homebrew

vistos na Seção 4.2. Com isso foram comparadas as redes de comunicação entre si e comparadas com o modelo padrão de indexação por *TF-IDF*. Também foram comparados os resultados de intervalos de tempo diferentes em todas as redes de comunicação.

Este trabalho está organizado da seguinte forma: O Capítulo 2 apresenta algumas pesquisas feitas anteriormente por outros pesquisadores que buscaram de alguma forma contribuir com formas de recomendação de *experts* em vários tipos de projetos. O Capítulo 3 apresenta o modelo utilizado para o desenvolvimento do sistema de recomendação. No Capítulo 4 é apresentada a metodologia utilizada para a produção e avalição dos resultados. No Capítulo 5 são apresentados os resultados obtidos com base nas avaliações dos dados produzidos pelo sistema de recomendação. Por fim, no Capítulo 6 são apresentadas as conclusões e as sujetões para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Encontrar *experts* em ambientes de produção coletivos é um grande desafio (DEMARTINI, 2007). Isso inclui ambientes como Wikipedia [1], comunidades de perguntas e respostas como StackOverflow [2], além dos ambientes de desenvolvimento colaborativos que já foram citados, como GitHub e BitBucket. A seguir são apresentados os trabalhos que possuem relação com esta pesquisa. Os trabalhos serão divididos entre os que consideram apenas aspectos técnicos e entre os que, assim como este trabalho, consideram os aspectos sociais para a recomendação de *experts*.

2.1 RELACIONADOS POR ASPECTOS SOCIAIS

Nesta seção serão apresentados as pesquisas relacionadas à este trabalho onde a caracteristica principal seja a utilização de fatores sociais nas recomendações de *experts*. Para dar um embasamento teórico a este trabalho, foram escolhidos alguns trabalhos que utilizam fatores sociais em suas pesquisas.

O artigo *ArnetMiner: An Expertise Oriented Search System for Web Community*, (TANG et al., 2007) apresenta um sistema Web para a busca de *expertises (people expertise)* em comunidades de pesquisadores acadêmicos. Nesse sistema, o perfil dos pesquisadores são modelados como ontologias extraídas a partir da mineração de páginas relevantes sobre o pesquisador (como sua página pessoal, se existente) e utiliza um índice invertido para indexação da rede de colaboração. O sistema permite a busca de contribuições dos autores e utiliza ténicas de mineração para encontrar *experts* em determinados tópicos. Para essa tarefa, utiliza-se uma abordagem que leva em consideração tanto as contribuições/publicações realizadas pelos pesquisadores quanto o relacionamento entre eles. A ideia é que se uma pessoa publica muito em uma área, então ela é *expert* nessa área, além disso, se uma pessoa conhece ou se foi co-autora com muitos *experts*, então ela é *expert* nessa área.

[1] Pode ser acessado em https://www.wikipedia.org
[2] Pode ser acessado em http://stackoverflow.com

O autor não apresenta nesse artigo, resultados sobre o sistema, apenas indica que está no disponível há 1 ano (quando o artigo foi publicado) e que recebia mais de 1500 acessos mensais. O artigo é basicamente uma prova de conceito sobre o sistema, apresentando suas principais funcionalidades. Acerca da identificação de *experts*, os autores afirmam que a prosposta apresentada é melhor que o modelo base de comparação. O modelo base de comparação é a utilização do perfil dos pesquisadores combinado com PageRank. A diferença com a proposta desta pesquisa é que eles incluem a utilização do domínio da rede de pesquisadores acadêmicos e a forma de mensurar a *"expertise"*. A semelhança é que, supostamente, considerar a colaboração existente entre os pesquisadores é também considerar um aspecto social relacionado à identificação de *expertise*.

No artigo *Pick Me! Link Selection in Expertise Search Results*, (SHAMI et al., 2008) foram analisados os fatores que influenciam se um usuário irá ou não selecionar um resultado de busca de *experts*. Foi descoberto que a ordem (*ranking*) e as conexões sociais mostradas juntas aos resultados da busca influenciam significativamente se um usuário irá explorar um resultado de busca ou não. O artigo orienta sobre questões de *design* de sistemas de busca de *experts* e baseia as direções dadas na descoberta de que as pessoas preferem consultar *experts* de seu círculo social ao invés de consultar estranhos (mesmo que sejam mais *experts*). O artigo se diferencia bastante da proposta desta pesquisa, pois ele somente demonstra uma pesquisa que identifica os fatores que influenciam se alguém irá ou não clicar em um resultado de busca de *experts*. Apesar disso, assim como esta pesquisa, a proposta do artigo considera que o aspecto social possui forte influência na busca de *experts*.

O artigo *Predicting Change Propagation in Software Systems*, (HASSAN; HOLT, 2004) apresenta uma abordagem para prever a propagação de alterações em entidades de código-fonte. Seu objetivo principal é saber como uma alteração em uma entidade de código-fonte pode se propagar à outra entidade. Para isso, analisaram os dados de grandes *CVS* (sistemas de controle de versão), observando as alterações que aconteciam em determinadas entidades de código-fonte. A semelhança entre com este trabalho, é o fato dos autores considerarem o histórico de alterações no código fonte para prever quais entidades serão alteradas no futuro. E este trabalho considera o histórico da comunicação entre os contribuidores para prever quais são os contribuidores indicados a participarem de uma determinada *issue*. Para medir seus resultados ele aplica as técnicas convencionais de recuperação de informação *Recall* e *Precision*.

2.2 RELACIONADOS POR ASPECTOS TÉCNICOS

Nesta seção serão apresentados as pesquisas relacionadas à este trabalho onde o foco foi simplesmente a recomendação de *experts* de maneira técnica, mesmo sem considerar os aspectos sociais. Esses artigos foram escolhidos porque mesmo os autores não utilizarem os aspectos sociais, eles deixaram espaços para que essas caracteristicas fossem adaptadas em trabalhos futuros, assim como este trabalho.

A recomendação de *experts* para uma solicitação de alteração é uma abordagem proposta no artigo *Who Can Help Me with this Change Request?*, (KAGDI; POSHYVANYK, 2009). A premissa básica desta abordagem é que os desenvolvedores que já contribuiram com alterações substanciais para uma parte específica do código fonte no passado, são os mais indicados para ajudar em sua alteração atual ou no futuro. Eles usam algoritmo LSI para extrair conceitos e verificar a semelhança entre uma solicitação de alteração e artefatos de código-fonte. Depois de selecionar os artefatos mais semelhantes, eles verificam quem são os desenvolvedores que têm mais *expertise* sobre esses artefatos. O *ranking* de *experts* é criadado com base na quantidade e frequência das alterações que um desenvolvedor fez um dado artefato de código fonte. Este estudo considera fatores puramente técnicos para recomendar *experts*. As variáveis sociais não são sequer mencionados, enquanto a abordagem desse trabalho foca em considerar a possibilidade de que aspectos sociais podem melhorar a recomendação ou não.

Expertise Browser (ExB) é uma ferramenta apresentada em *Expertise Browser*, (MOCKUS; HERBSLEB, 2002), que utiliza o conceito de *Experience Atoms* (*EAs*) para representar as unidades de conhecimentos coletados de repositórios de código-fonte. Eles afirmam que a *expertise* é o resultado direto da atividade de uma pessoa em relação a um produto trabalhado, alterando ou corrigindo um *bug*. EAs são usados para gerar uma rede sociotécnica envolvendo as relações entre os artefatos, pessoas e tarefas. Ele é usado para identificar *experts* e traçar seus relacionamentos. Apesar de usar um gráfico de rede social para procurar *experts*, os aspectos sociais inerentes à processos de software não são considerados pelo *ExB*.

Emergent Expertise Locator (*EEL*) é uma ferramenta apresentada em *Recommending Emergent Teams*, (MINTO; MURPHY, 2007) e visa recomendar *experts* em um determinado assunto. Ele usa o histórico de alterações do código-fonte para classificar os *experts* de um determinado artefato. Para classificar os *experts*, a ferramenta faz uso da matriz de exigência de coordenação. Esta abordagem, de certa forma, considera um aspecto social do processo de desenvolvimento de software, isso porque, uma matriz de coordenação de requisitos baseia-se nas dependências lógicas entre artefatos técnicos e as relações que indiretamente ocorreram ao

longo desses artefatos. No entanto, a abordagem não considera a estrutura social do projeto ao recomendar a *expertise* de uma determinada pessoa.

O artigo *Recommending Experts Using Communication History*, (MORAES et al., 2010) apresenta o Conscius, uma ferramenta que visa facilitar o acesso a *experts* em um determinado projeto de software. O *ranking* dos *experts* é feito através da mineração do histórico de alterações e tópicos de e-mail arquivados. A ferramenta analisa o conteúdo da comunicação, para melhorar as recomendações de usuários com base no histórico de código-fonte e o relacionamento entre o código-fonte. Embora utilizem histórico de comunicação, mais uma vez a abordagem é baseada apenas em aspectos não-sociais do projeto.

Na próxima seção é apresentado o modelo utilizado para o desenvolvimento da abordagem proposta neste trabalho, para a recomendação de contribuidores com base no histórico de comunicação entre eles. Serão apresentadas as ferramentas utilizadas, a geração das redes de comunicação e onde os valores de *PRP* foram aplicados.

3 DESENVOLVIMENTO

Para colaborar com os modelos de recomendação já existentes, optou-se neste trabalho por analisar o histórico de comunicação entre os contribuidores e modelar as redes de comunição existente entre eles. Com base nos dados obtidos a partir dessas redes, buscou-se a comparação entre a utilização das redes de comunicação com os modelos que têm como base a indexação de vocabulários apenas com *TF-IDF*. Neste capítulo será apresentado o modelo utilizado para o desenvolvimento do sistema de recomendação.

3.1 MODELO DE RECOMENDAÇÃO

A fim de resumir o processo de recomendação de *experts*, foi criada a Figura 1, que apresenta um panorama geral do processo, que vai desde a mineração dos dados até a analise dos resultados.

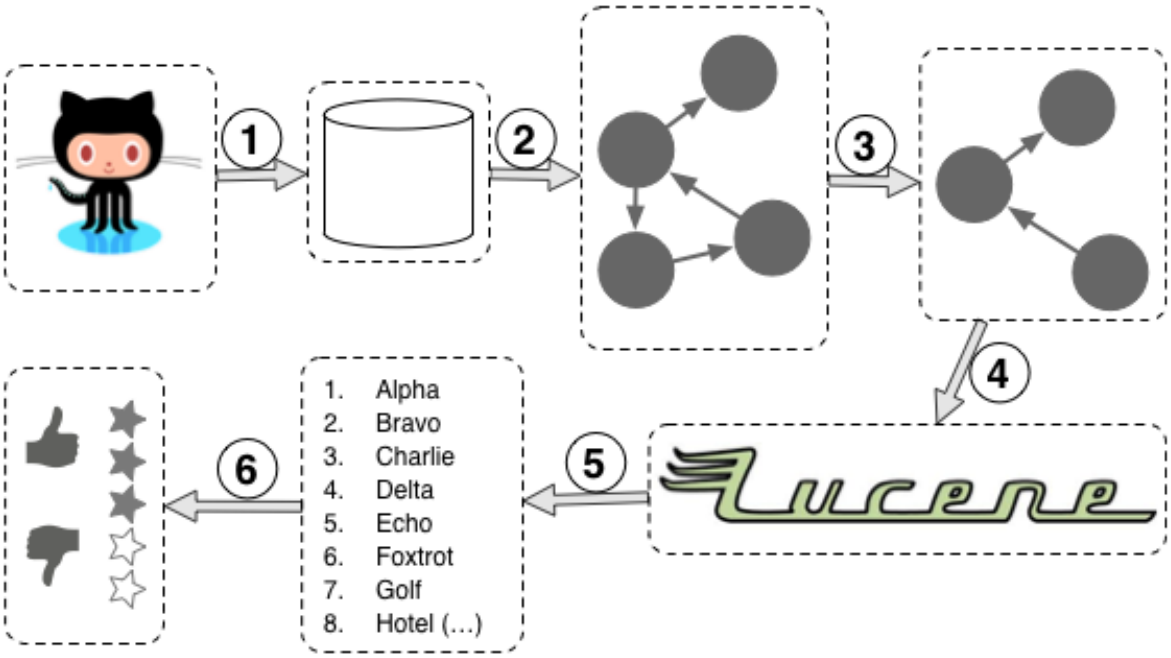

Figura 1: Criação de Redes de Comunicação Restritas.

Como pode ser visto na Figura 1, o processo inicia com a mineração dos dados do GitHub e sua persistência em uma base de dados local, no $passo_1$. Em seguida, no $passo_2$ são geradas as redes de comunicação com base na interação entre os contribuidores nas *issues* e ao mesmo tempo é feita a criação dos vocabulários. No $passo_3$ está a criação das redes restritas por termos, onde é feito o cálculo do *PRP* e então são indexados os vocabulários pelo Lucene. No $passo_4$ é passado para o Lucene como pesquisa os termos de cada *issue* do período de recomendação, onde cada uma dessas *issues* é uma pesquisa separada. No $passo_5$ o Lucene irá gerar uma lista de recomendação com os contribuidores mais indicados para participarem da *issue* que foi passada como consulta. No $passo_6$ é finalizado o processo analisando-se manualmente os resultados de *Recall*, *Precision* e *Likelihood* para saber se a recomendação foi relevante ou não, tendo como base os comentários reais da *issue* que foi passada como consulta.

3.2 ABORDAGEM PARA RECOMENDAÇÃO DE *EXPERTS*

A fim de validar e avaliar a abordagem proposta neste trabalho, foi desenvolvido um sistema chamado de JGitRecommender[1], que irá atuar como um sistema de recomendação de *experts* para os projetos de software livre usando a abordagem proposta. Sua função é a recomendação de *experts* em projetos colaborativos. No escopo deste trabalho, este sistema tem como objetivo recomendar os contribuidores que sejam mais indicados à responder uma determinada *issue* de acordo com a expertise do contribuidor. Assim, espera-se que esses contribuidores recomendados possam ajudar novos integrantes do projeto ou quaisquer outros contribuidores que estejam precisando de ajuda. Este sistema tem como base projetos de software livre que utilizam o repositório GitHub.

Para determinar a expertise de cada contribuidor o sistema leva em consideração o conteúdo das mensagens trocadas entre ele e outros contribuidores, durante um determinado período de tempo, dentro de um projeto específico. Esse conjunto de mensagens do contribuidor foi utilizado para a criação do seu vocabulário dentro do projeto. De forma geral, para a criação desse vocabulário, são considerados os seguintes conteúdos: o título e descrição das *issues* (tarefas) em que o contribuidor participou, seja respondendo ou criando a *issue*, juntamente com suas respostas. Esse processo será visto detalhadamente mais adiante, na seção da criação do vocabulário.

Outro ponto importante para a determinação da expertise dos contribuidores dentro do projeto foi a utilização do conceito de redes de comunicação. Esse conceito leva em consideração a relevância de um contribuidor com base em sua interação com outros contribuidores.

[1] A ferramenta JGitRecommender está disponível em https://github.com/wyworak/JGitRecommender

Assim, de acordo com o conteúdo dos termos de pesquisa passados para o sistema de recomendação, ele irá retornar os contribuidores com maior relevância nesses termos. A estrutura das redes de comunicação e a forma como foram criadas serão vistas em detalhes mais adiante, na seção de elaboração das redes de comunicação.

Antes de apresentar o processo de elaboração do sistema de recomendação, serão apresentadas as ferramentas que foram utilizadas em seu desenvolvimento, para que o leitor possa se familiarizar com os termos que serão apresentados nas próximas seções.

3.3 FERRAMENTAS UTILIZADAS

O desenvolvimento do sistema de recomendação utilizou como base a linguagem de programação Java. Além disso, para sua complementação, foram utilizadas algumas ferramentas para auxiliarem na geração e análises das redes de comunicação existentes no projeto. Dentre elas destacam-se as seguintes ferramentas.

3.3.1 JGITWEBMINER

O JGitWebMiner [2] é um sistema web desenvolvido em Java. Sua principal função é a mineração de dados de projetos de software livre e criação de redes de comunicação, técnicas e sócio-tecnicas. Neste trabalho o JGitWebMiner foi utilizado para a mineração de dados de projetos que estão armazenados nos repositórios do GitHub e que foram utilizados no estudo de caso.

3.3.2 GITHUB

Optou-se por coletar dados do Github porque vários projetos estão migrando seus processos e códigos fonte para esta plataforma. Inclusive a Google que possui um serviço semelhante desde 2006, o GoogleCode [3] anunciou em seu Blog [4] que a partir de Janeiro de 2016 estará encerrando suas atividades e está indicando o GitHub como subistituto para hospedagem de projetos. Os próprios projetos da Google estão sendo transferidos para o GitHub. Isso está ocorrendo porque GitHub usa o Git, que é um sistema de gerenciamento de versão distribuído e oferece vários recursos sociais que melhoram a experiência de interação entre os contribuidores.

Para realizar este trabalho, foi necessário reunir os dados extraídos dos projetos. Para

[2] A ferramenta JGitWebMiner está disponível em https://github.com/igorwiese/JGitWebMiner
[3] Disponível em: https://code.google.com/
[4] Disponível em: http://google-opensource.blogspot.com.br/2015/03/farewell-to-google-code.html

conseguir isso, usou-se a API Github [5] que permitiu a recuperação de dados sobre projetos, *issues* e colaboradores. Foi utilizada a versão 3 da APIGithub. Com ela foi possível recuperar objetos JSON contendo todos os dados sobre o assunto. O objeto JSON foi analisado e armazenado em um banco de dados local. Os dados coletados incluem data de criação, o autor, descrição da *issue* (tarefa), a lista de comentários associados à *issue* e seus respectivos autores (contribuidor que escreveu a *issue*).

Por meio dessa API foi possível minerar os dados dos repositórios do GitHub e então persisti-los em uma base de dados local. Os projetos escolhidos para este trabalho foram Homebrew, NodeJs e Bootstrap. A escolha dos projetos foi estabelecida com base na popularidade que os projetos possuem no GitHub. Para a descoberta dos projetos mais populares, foi realizada uma pesquisa [6] no GitHub e então foram escolhidos três projetos que estavam entre os dez mais populares.

3.3.3 LUCENE

O Apache Lucene [7] é um projeto de código aberto. O Lucene é uma biblioteca para motores de busca de alto desempenho. É totalmente escrita em Java e é adequada para praticamente toda aplicação que necessite de pesquisas por texto. O Lucene foi utilizado neste sistema para fazer a indexação dos contribuidores do projeto utilizando *TF-IDF*, facilitando na busca e posteriores cálculos.

O *TF-IDF* baseia-se na geração da frequência de termos (*TF*) e do inverso da frequência de termos (*IDF*). Com esse algoritmo, o cálculo de relevância de um termo é feito primeiramente com o (*TF*) contando quantas vezes o termo aparece nos documentos. Considerando que os termos que se repetem muito podem não ser tão relevantes quando os que aparecem menos, então, é utilizada uma função inversa que irá acrescentar o valor do inverso da frequência do termo (*IDF*), normalizando assim os valores de cada termo e definindo quais tem mais relevância (MCCANDLESS et al., 2010).

Como uma das hipóteses da pesquisa é que a comunicação está associada aos interesses dos contribuidores, o foco foi analisar as interações sociais e o texto que descreve os problemas e as observações apresentadas pelos contribuidores. Foi utilizado o *framework* Apache Lucene para indexar e consultar a estrutura textual das *issues*. A abordagem do Lucene para recuperação de documentos consiste em duas etapas principais: i) indexar os dados textuais originais

[5] A API GitHub está disponível em http://developer.github.com/

[6] Disponível em: https://github.com/search?q=stars%3A%3E1&s=stars&type=Repositories

[7] http://lucene.apache.org/

gerando estruturas interligadas que permitem a consulta com base em palavras-chave; ii) consultar os documentos indexados e classificar os resultados de acordo com a similaridade com a consulta.

3.3.4 JUNG

A *Java Universal Network/Graph Framework* [8] é uma biblioteca de software que fornece uma linguagem comum e extensível para a modelagem, análise e visualização de dados que podem ser representados como um grafo ou uma rede. Com ele foram criadas as redes de comunicação. Ele fornece uma grande quantidade de métodos para cálculos sobre redes de comunicação. Neste sistema foram utilizados os métodos para cálculo de *PRP*. Os resultados desses cálculos representam a importância de um determinado nó, neste caso um contribuidor, em relação à rede como um todo. Esses valores foram adicionados ao resultado dos cálculos de *TF-IDF*.

3.3.5 *PAGERANK WITH PRIORS*

O *PageRank* é um algoritmo utilizado atualmente pelos motores de busca do Google para calcular a relevância de uma página web em relação à outra. Foi desenvolvido pelos fundadores da empresa durante o tempo que estudavam na universidade de Stanford nos Estados Unidos.

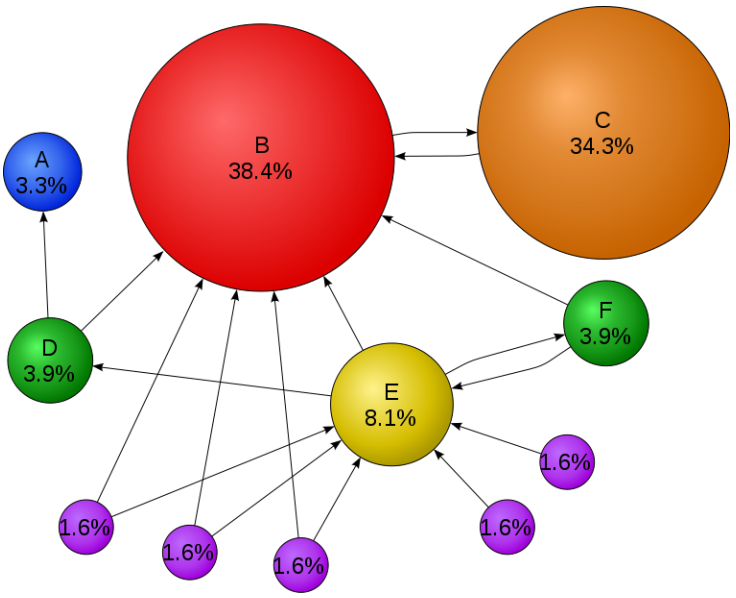

Figura 2: Representação do Sistema de PageRank.
Fonte: Wikipedia.

[8] http://jung.sourceforge.net/

A Figura 2 representa um conjunto de páginas na internet, cada nó representa uma página e o tamanho de cada um representa a sua relevância em relação às demais, assim, quanto maior o tamanho, maior o valor de *PageRank* da mesma. Com isso, quando um nó com um alto valor de *PageRank* aponta para outro qualquer, esse nó que recebeu o apontamento terá seu valor de *PageRank* aumentado.

O algoritmo utilizado neste trabalho para o cálculo da relevância de um nó nas redes de comunicação foi o *PageRank With Priors* (*PRP*) (WHITE; SMYTH, 2003). Ele é uma adaptação do algoritmo de *PageRank* original. A equação utilizada para seu cálculo é a seguinte:

$$\pi(v)^{i+1} = (1-\beta)\left(\sum_{u=1}^{d_{in}(v)} p(v \mid u)\,\pi^{(i)}(u)\right) + \beta p_v$$

O *PRP* é utilizado em redes ponderadas. Considerando um conjunto de páginas web, a probabilidade de uma pessoa ir para a página *B* estando em *A* é proporcional ao peso de cada página conectada a *B*. No caso deste trabalho, pode-se dizer que probabilidade de um contribuidor ser recomendado é proporcional ao peso de cada contribuidor que possua uma conexão com ele.

Nas próximas seções serão apresentados o processo de formação das redes de comunicação e o vocabulário dos contribuidores, como esse vocabulário é preparado e onde são utilizados os valores de *PRP*.

3.4 GERAÇÃO DAS REDES DE COMUNICAÇÃO

Para o cálculo de *PRP* é construído um grafo base para cada uma das redes de comunicação, onde cada contribuidor é um nó desse grafo e as arestas entre esses nós são as interações dos contribuidores com base nos comentários em *issues*.

A Figura 3 apresenta o conceito geral utilizado para a criação das redes de comunicação entre os contribuidores de um projeto de software livre. Na Figura 3, as setas cheias representam as *issues* que foram abertas no período de indexação, no caso abertas pelos contribuidores O_1 e O_2. Enquanto as linhas pontilhadas representam os comentários que foram feitos nas *issues* pelos contribuidores U_1, U_2 e U_3 .

Esse é apenas um panorama geral de como foram criadas as redes de comunicação, porque cada uma das três redes de comunicação possuem algumas particularidades em sua formação.

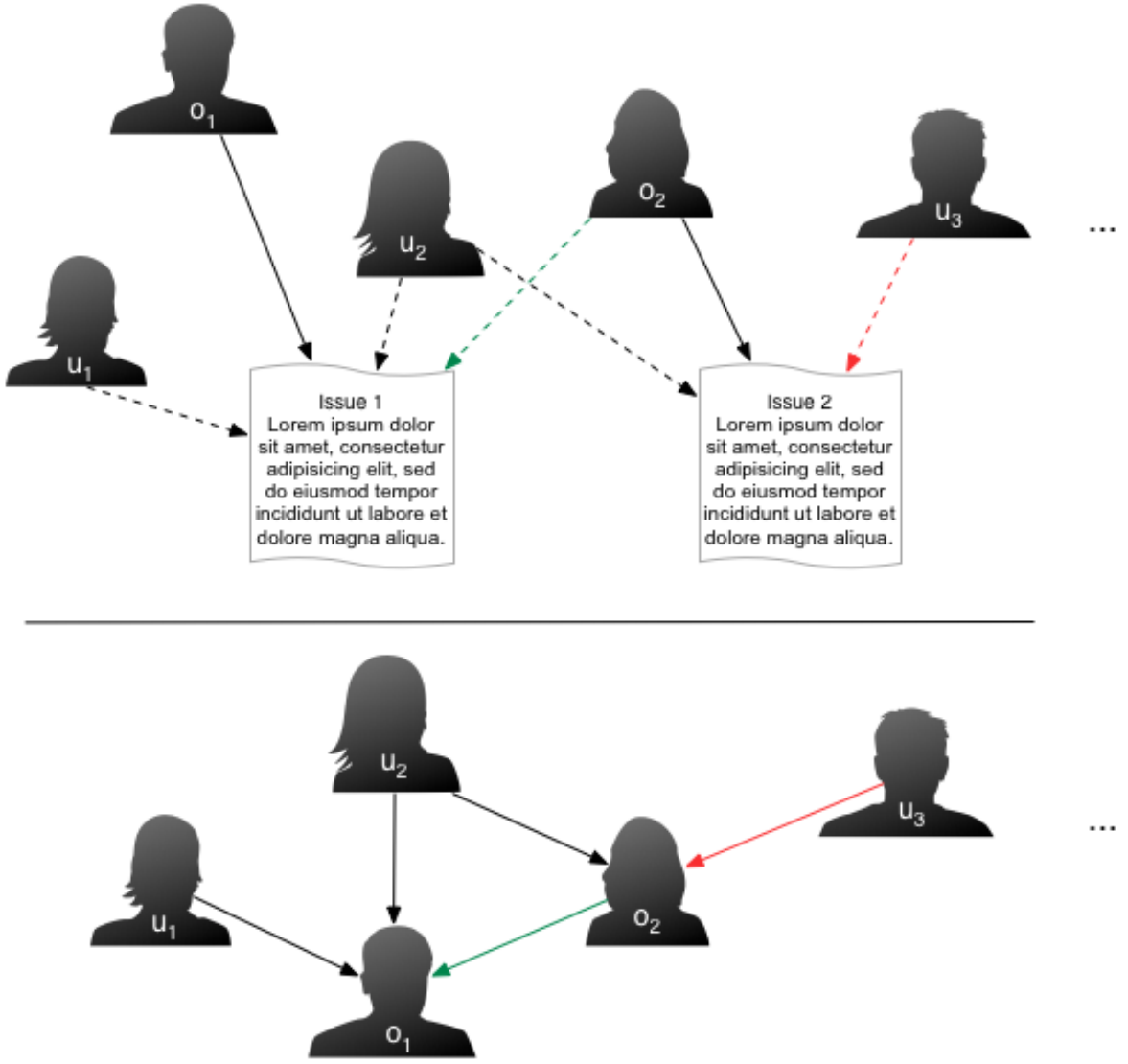

Figura 3: Conceito de Criação de Redes de Comunicação.

A seguir serão apresentados os métodos utilizados em cada uma das redes de modo específico. Para um melhor entendimento da criação dos grafos das redes de comunicação, será considerado o seguinte exemplo com personagens fictícios:

O contribuidor Malorum, que contem a ID 2 no sistema JGitRecommender, encontrou um *bug* no código do projeto em que estava participando, então entrou no GitHub e abriu uma *issue*. Logo em seguida, outros contribuidores observaram que existia uma *issue* aberta e então a responderam. Será considerado que os contribuidores responderam na seguinte ordem: primeiro o contribuidor Finibus com ID 4 respondeu a *issue*, em seguida o contribuidor Bonorum com ID 1, logo após o contribuidor Malorum com ID 2, que havia criado a *issue*, voltou para comentar e por ultimo o contribuidor Finibus com ID 4 também voltou a comentar.

Com base nesse cenário, foram criadas as três redes de comunicação que são apresen-

tadas nas seções a seguir.

3.4.1 REDE *FIRST*

A criação da rede *First* usa o seguinte conceito: a seta parte de quem abriu a *issue* para quem respondeu. No caso do exemplo acima pode ser visto que o contribuidor com ID 2 ficou centralizado no grafo porque foi ele quem abriu a *issue*. Em seguida, existe uma seta partindo de 2 para 4, isso porque o contribuidor de ID 4 comentou a *issue*. Logo após, o contribuidor 1 também comenta, logo uma seta parte de 2 para 1. Seguidamente, têm-se o contribuidor 2 comentando na sua própria *issue*. Nesse caso, não foi considerado correto colocar uma seta dele para ele mesmo. E, para finalizar a *issue*, o contribuidor 4 volta a comentar, então, novamente uma seta parte de 2 para 4. Esse exemplo é representado pela Figura 4.

Um detalhe que deve ser obervado na Figura 4 são as duas setas que partem do nó 2 para o nó 4. Elas apenas simbolizam o peso das arestas. Para o sistema o que existe é apenas uma aresta, neste caso, com peso 2. Considerando o exemplo anterior, caso o contribuidor 4 comentar em mais *issues* abertas pelos contribuidor 2, maior será o peso de sua aresta e com isso, maior influência ele terá. Esse detalhe vale para as demais redes de comunicação utilizadas neste trabalho.

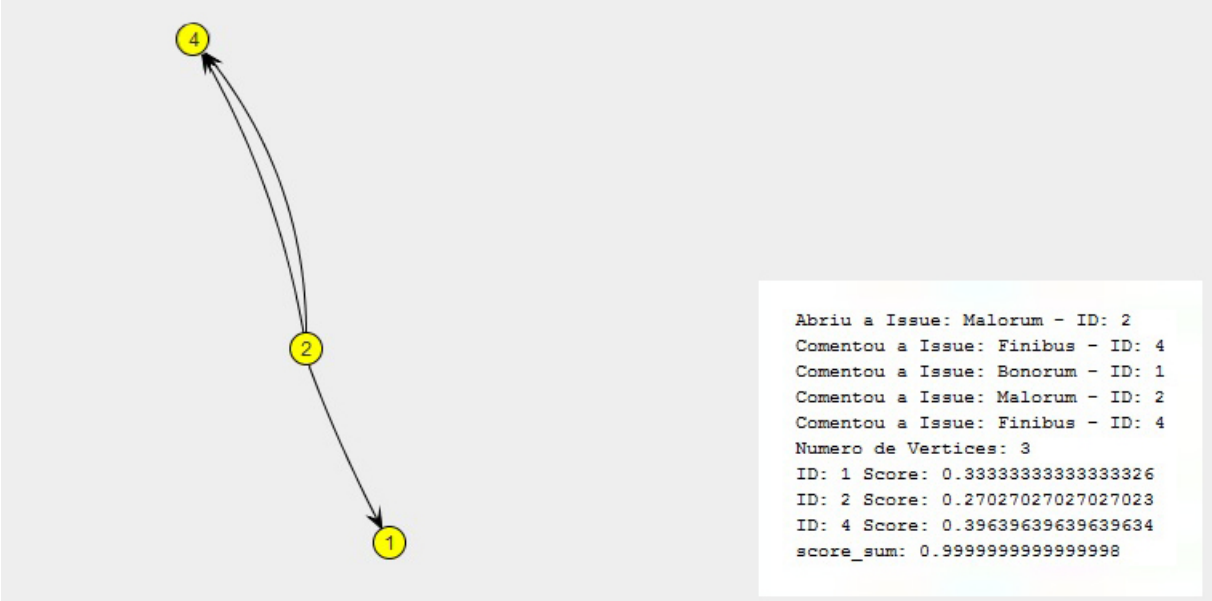

Figura 4: Representação Gráfica da Rede *First*.

Utilizando o *PagaRank with Priors*, o peso das arestas influencia na determinação da relevância de um nó na rede. Sendo assim, quanto mais setas chegam de outros nós em direção a um determinado nó, mais influente ele passa a ser. Por essa razão, quando um contribuidor

comenta em várias *issues* de outro contribuidor ou simplesmente responde mais de uma vez o mesmo contribuidor, o peso de sua aresta recebe um acréscimo.

3.4.2 REDE *PRIORS*

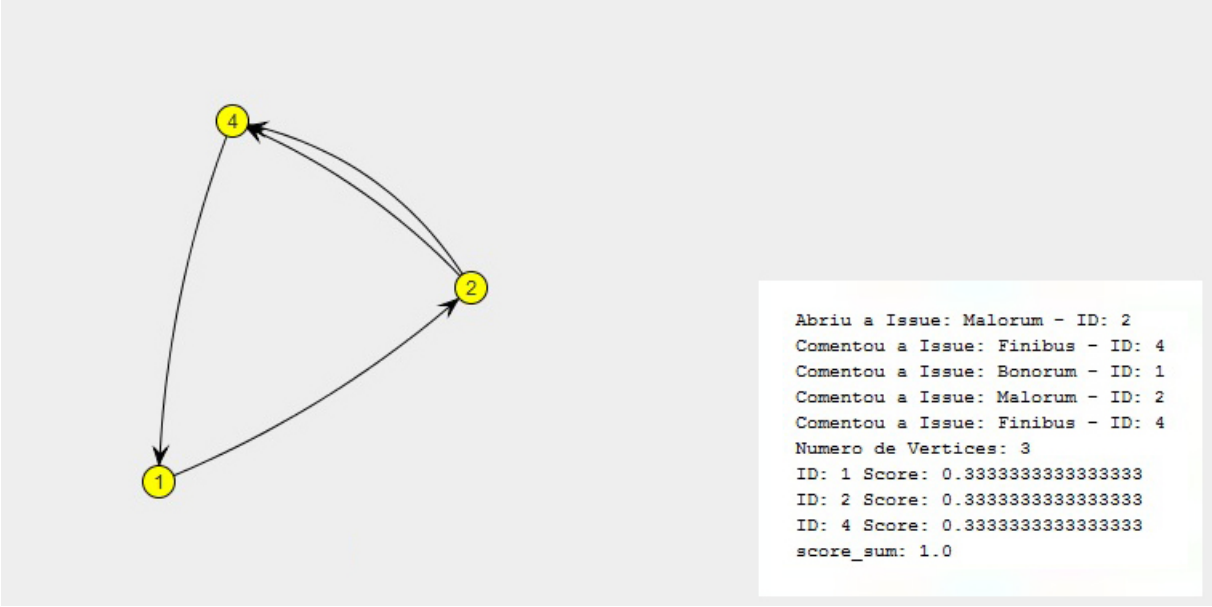

Figura 5: Representação Gráfica da Rede *Priors*.

A criação da rede *Priors* usa o seguinte conceito: a seta parte de quem abriu a *issue* para quem respondeu e após isso aponta sempre para o próximo contribuidor que responder. Considerando esse conceito e o cenário anterior, como pode ser visto na figura 5, o contribuidor com ID 2 abriu a *issue*. Em seguida, existe uma seta partindo de 2 para 4, isso porque o contribuidor de ID 4 foi o primeiro a comentar na *issue*. Logo após, o contribuidor 1 também comenta, nesse caso, uma seta parte de 4 para 1. Seguidamente, têm-se o contribuidor 2 comentando na sua própria *issue*. Agora, no caso da rede *Priors*, ele pode receber uma seta, porque não está apontando para ele mesmo, e sim, existe a seta partindo de 1 para 2. E, para finalizar a *issue*, o contribuidor 4 volta a comentar, então, novamente uma seta parte de 2 para 4. Caso um contribuidor comente duas vezes seguidas na *issue*, será considerado apenas uma aresta. Mas se ele comentar várias vezes na *issue* respondendo outros contribuidores, nesse caso, serão criadas arestas a cada novo comentário. Esse exemplo é representado pela Figura 5.

3.4.3 REDE *EVERYONE*

Para a criação da rede *Everyone* o algoritmo simplesmente passa criando as duas redes ao mesmo tempo, nesse caso têm-se a soma das duas redes ao final da execução. Considerando

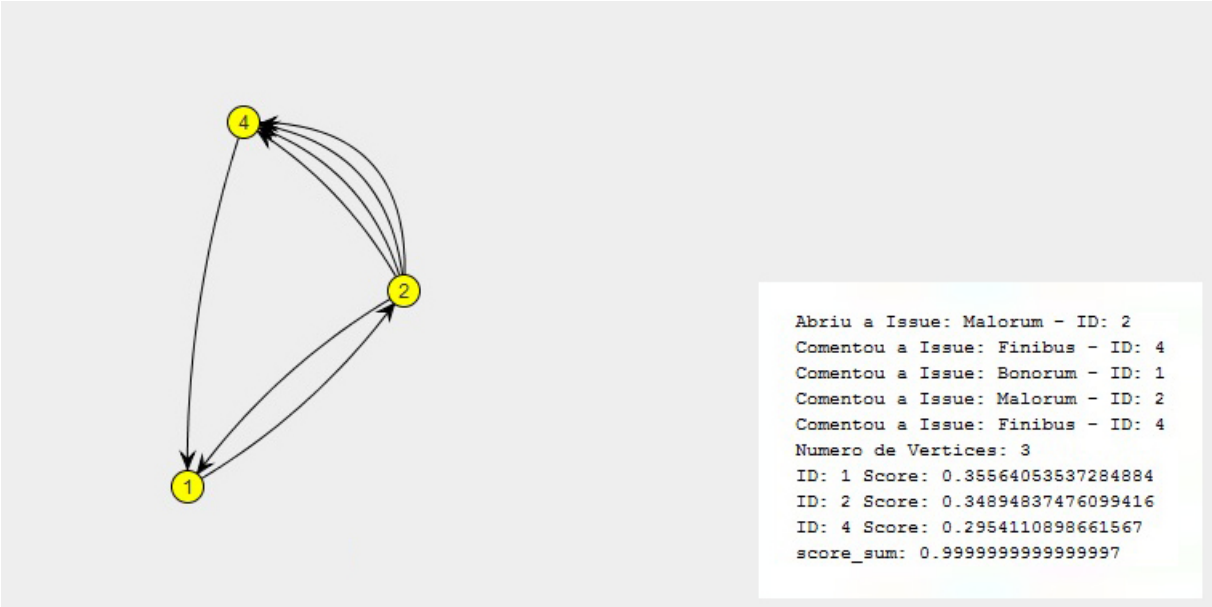

Figura 6: Representação Gráfica da Rede *Everyone*.

o cenário anterior, pode ser visto na Figura 4 que, o contribuidor com ID 2 abriu a *issue*. Em seguida, existem duas setas partindo de 2 para 4, isso porque o algoritmo cria uma seta da rede *First* e outra para a Rede *Priors*. Logo após, o contribuidor 1 também comenta, nesse caso, uma seta parte de 4 para 1 devido a rede *Priors* e ao mesmo tempo, uma seta parte de 2 para 1 devido a rede *First*. Em seguida, têm-se o contribuidor 2 comentando na sua própria *issue*. Assim, uma seta parte de 1 para 2 devido a rede *Priors*, e não exite seta de 2 para 2 devido a rede *First*. E, para finalizar a *issue*, o contribuidor 4 volta a comentar, então, novamente duas setas partindo de 2 para 4, isso porque o algoritmo cria uma seta da rede *First* e outra para a Rede *Priors*. Esse exemplo é representado pela Figura 6.

3.5 FORMAÇÃO DO VOCABULÁRIO DOS CONTRIBUIDORES

O processo para a formação do vocabulário dos contribuidores inicia-se com a definição de um período dentro do projeto no qual será analisado. Neste trabalho foram utilizados três períodos diferentes, um com doze meses e outros dois períodos com seis meses. Esses períodos serão nomeados de Período de Indexação, para os quais serão coletados os termos dos usuários para que as redes de comunicação sejam compostas. Para cada um desses períodos de indexação, foi criado outro período posterior com a mesma quantidade de tempo, os quais serão nomeados de Período de Recomendação. Esses últimos períodos serão aqueles para os quais serão recomendados os *experts*.

O período de indexação foi criado para que fosse possível reduzir o problema de reco-

mendação de contribuidores que não estão mais no projeto e também para ser utilizado como um parâmetro nas avaliações. Com a definição desse período, são consideradas como válidas apenas as *issues* que foram criadas e fechadas dentro desse intervalo de tempo. Assim, as *issues* que foram criadas dentro desse intervalo mas não foram fechadas foram desconsideradas e também, as que foram criadas fora do intervalo e foram fechadas dentro do intervalo, foram desconsideradas. Essa foi uma maneira encontrada para definir o vocabulário dos contribuidores dentro de um determinado período.

O período de recomendação foi criado para que seus dados sejam usados nos testes do sistema de recomendação e para avaliação do mesmo. Para isso, da mesma forma que no período de indexação, nesse período são consideradas apenas as *issues* que foram abertas e fechadas nesse intervalo de tempo. A criação desses períodos será vista com mais detalhes mais adiante na Seção 4.3, onde é abordada a criação dos períodos.

Partindo do princípio que o período foi definido, o próximo passo é a escolha das redes de comunicação. Nesse trabalho, optou-se por utilizar três redes de comunicação diferentes, cujos nomes foram definidos como *First*, *Priors* e *Everyone*. Depois de definidas as redes, os vocabulários foram criados de formas diferentes para cada rede. Após a geração de cada um dos vocabulários com base nas redes de comunicação, os mesmos são salvos. Ficando então um vocabulário para cada usuário, para cada rede de comunicação, para cada período.

Para um melhor entendimento, a criação de cada vocabulário será apresentada de forma detalhada a seguir.

3.5.1 VOCABULÁRIO PARA A REDE *FIRST*

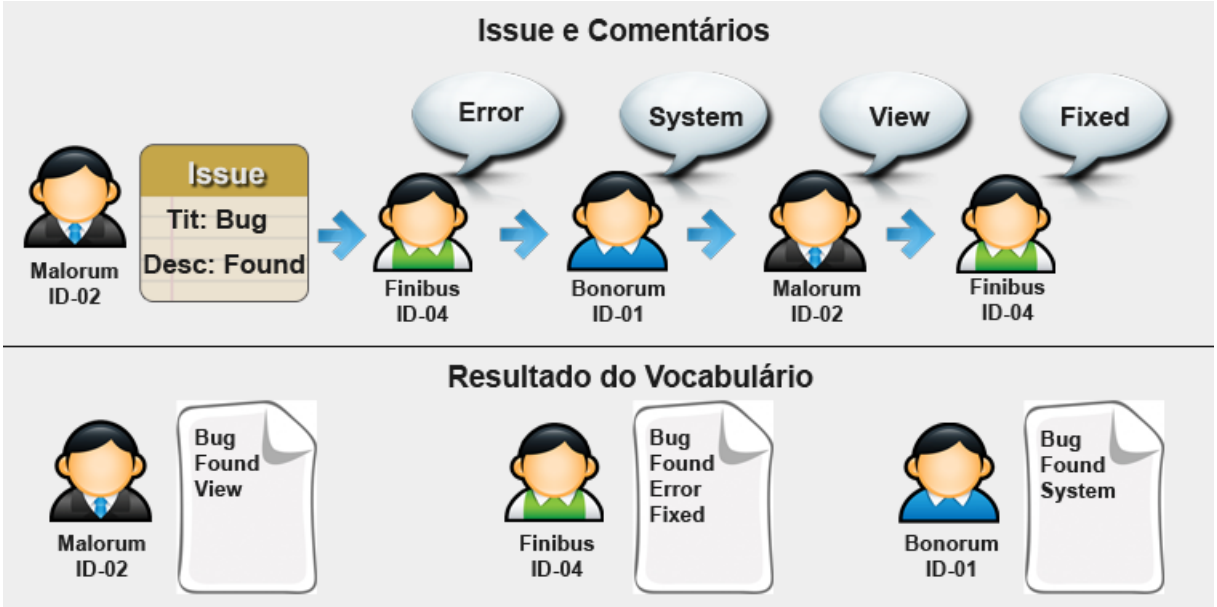

Figura 7: Formação do Vocabulário na Rede *First*.

Para a criação do vocabulário da rede *First* primeiramente foram selecionadas na base de dados todas as *issues* que foram abertas e posteriormente fechadas dentro do período definido para análise. Após isso, para cada *issue* do período foi encontrado o contribuidor que a criou e atribuído ao mesmo como vocabulário o título e a descrição dessa *issue*. Então esse contribuidor foi para uma lista de contribuidores. Feito isso, foram analisados todos os comentários de cada *issue*, onde cada contribuidor que comentou recebe em seu vocabulário o título e a descrição da *issue*, que são somadas aos seus comentários e então esses contribuidores também vão para a lista.

Nessa rede, como pode ser visto na Figura 7, é considerado apenas como vocabulário do contribuidor o título, a descrição e mais os comentários que ele fizer em cada *issue* que participar. Portanto, considera-se como conhecimento do contribuidor apenas o que ele comentou. O conteúdo da *issue* é acrescentado porque parte-se do principio que se ele respondeu é porque entende do assunto.

3.5.2 VOCABULÁRIO PARA A REDE *PRIORS*

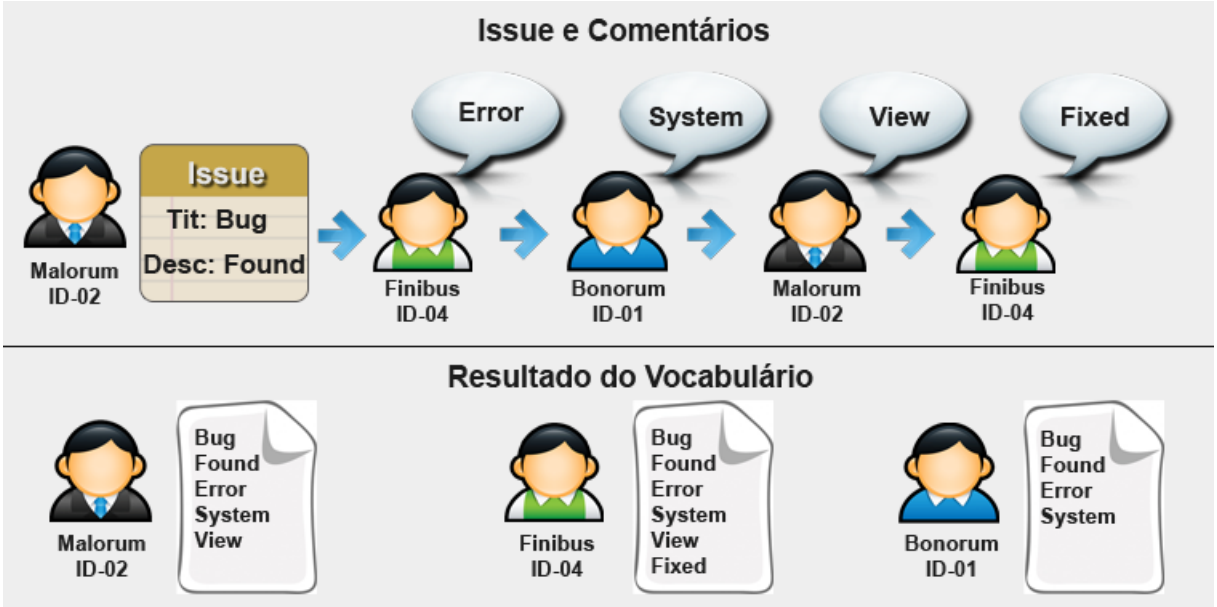

Figura 8: Formação do Vocabulário na Rede *Priors*.

Para a criação do vocabulário da rede *Priors*, assim como na rede *First*, foram selecionadas na base de dados todas as *issues* que foram abertas e posteriormente fechadas dentro do período definido para análise. Novamente, cada contribuidor que abriu uma *issue* recebe o conteúdo do título e descrição da *issue* em seu vocabulário e vai para uma lista de contribuidores. Feito isso, são analisados todos os comentários de cada *issue*, onde cada contribuidor que comentou recebe em seu vocabulário o título, a descrição, o seu comentário e mais os comentários anteriores ao seu naquela *issue*.

Nessa rede, como pode ser visto na Figura 8, é considerado que cada vez que o contribuidor comentou em uma *issue* ele adquiriu conhecimento sobre o assunto, sendo assim tem em seu vocabulário o título com a descrição e todos os comentários anteriores nas *issues* onde ele participou.

3.5.3 VOCABULÁRIO PARA A REDE *EVERYONE*

Para a criação do vocabulário da rede *Everyone* são utilizados os dois métodos das redes anteriores. Nesse método, é levado em consideração que todos que participaram de uma *issue* adquiriram o conhecimento naquele assunto, sendo assim, todos que participaram recebem o título, a descrição e todos os comentários daquela *issue*. Uma representação gráfica da formação do vocabulário na rede *Everyone* pode ser vista na Figura 9.

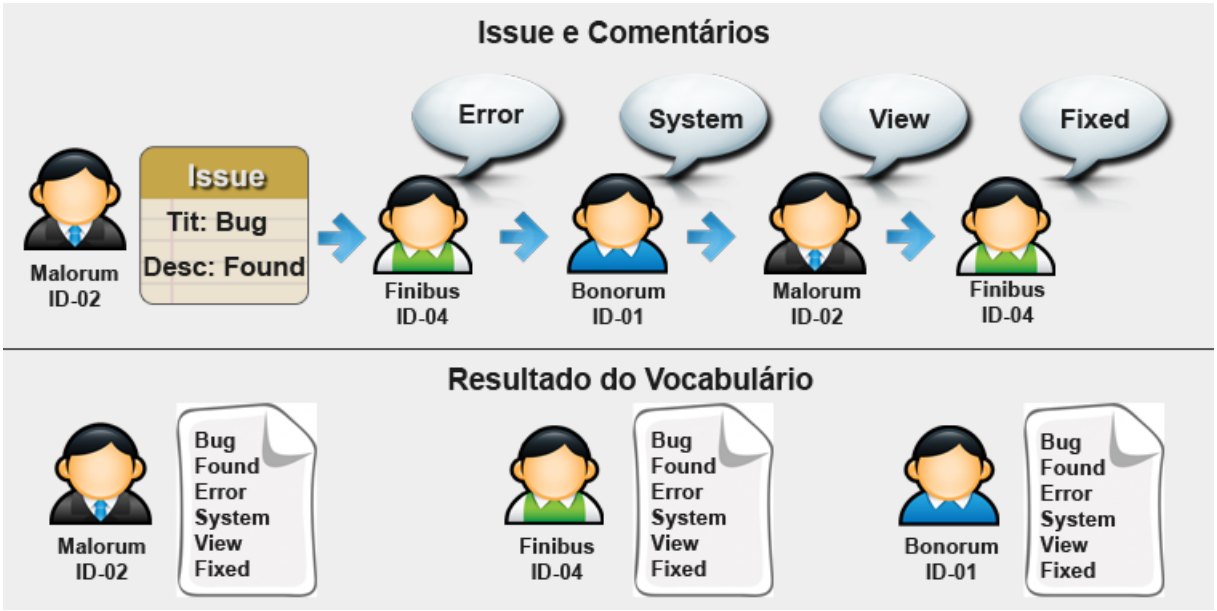

Figura 9: Formação do Vocabulário na Rede *Everyone*.

Na rede *Everyone*, todos os contribuidores que participaram de uma *issue* ficam com os todos os termos que estão relacionados a ela. Esse modelo de vocabulário foi criado para testar se pode harver melhores resultados considerando que os contribuidores adquiriram conhecimento por participarem de determinada *issue*.

3.6 OTIMIZAÇÃO DO VOCABULÁRIO

Para otimizar a indexação e a recomendação de contribuidores, algumas ações são importantes no momento de gerar o vocabulário. Uma delas foi o corte dos termos que aparecem apenas uma vez no período. Para isso, foi feito um mapa com todos os termos e a quantidade de vezes que eles aparecem no período. Assim, ao indexar cada contribuidor, se o termo não aparecer mais de uma vez, é removido do vocabulário.

Outra ação para otimização foi o uso de um analisador customizado do Lucene. Nesse analizador são removidos termos chamados de *StopWords*, que são termos irrelevantes para os motores de busca e também são cortados os termos com menos de quatro caracteres. Utiliza-se também a técnnica de *stemming*, que consiste em reduzir as palavras ao seu radical. Com isso, todas as variações da palavra se tornam uma palavra só.

Com o vocabulário otimizado, parte-se então para o processo de análise de *PRP* de cada termo.

3.7 REDES RESTRITAS E CÁLCULO DO *PRP* POR TERMOS

Com base nos grafos criados anteriormente, o cálculo do *PRP* para os termos, de modo geral, é feito da seguinte forma:

A partir dos grafos base de cada uma das redes de comunicação, é gerada uma rede de comunicação restrita para cada um dos termos do projeto que foram salvos anteriormente. Para isso, compara-se o termo com o vocabulário dos contribuidores, quando um contribuidor não possui o termo analisado em seu vocabulário, ele é removido do grafo. Ao final da comparação de cada termo, ficam no grafo apenas os contribuidores que possuem esse termo em seu vocabulário. Com isso em mãos, é calculado o *PRP* desse termo para cada contribuidor e então guardado em uma lista para que seja persistido na base de dados para uma futura indexação dos contribuidores.

A Figura 10 representa a criação de uma rede de comunicação restrita. Nela, o termo utilizado para a restrição da rede é *bug*. Nesse caso, o sistema irá analisar todos os que possuem o termo *bug* em seu vocabulário e então as conexões entre eles serão mantidas enquanto os contribuidores que não possuem o termo *bug* serão excluídos da rede. Feito isso, o sistema calcula o *PRP* dos contribuidores que ficaram na rede utilizando a classe *PageRankWithPriors* [9] do Jung, então esse valor é aplicado ao termo *bug* de cada um que sobrou na rede restrita e então o termo é indexado.

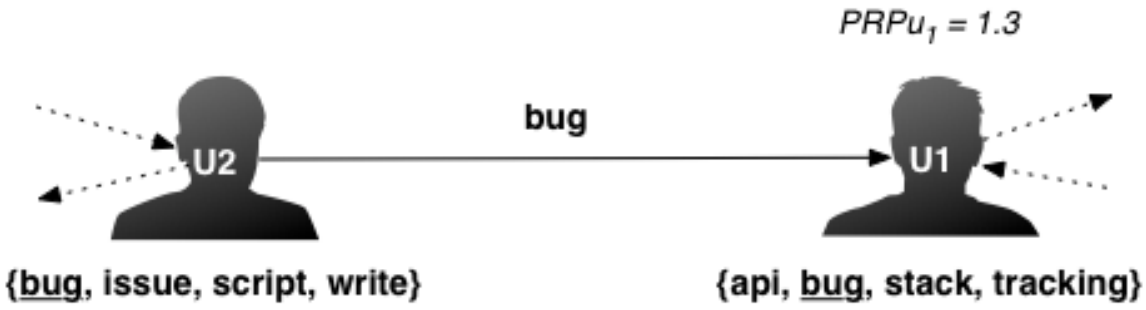

Figura 10: Criação de Redes de Comunicação Restritas.

3.8 INDEXAÇÃO DOS CONTRIBUIDORES E SEUS VOCABULÁRIOS

Depois de definidas as redes, os vocabulários foram indexados de duas formas diferentes para cada uma das redes, para fins de serem comparados no futuro. Em uma das formas,

[9] Disponível em http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/scoring/PageRankWithPriors.html

os vocabulários são indexados apenas considerando-se o *TF-IDF* e outra na qual foi acrescentado ao termo o valor do *PRP*, calculado anteriormente em cada uma das redes de comunicação. Para isso, utilizou-se o método do Lucene $termo.setBoost(floatprp)$. Isso faz com que o termo receba um ganho e com isso seja considerado mais relevante para o motor de buscas. Fazendo isso, acredita-se que quanto mais interações sociais um contribuidor tenha dentro do projeto, maior sua relevância em determinados termos.

Ao final da indexação têm-se seis vocabulários diferentes indexados para cada um dos contribuidores:

1. ***First*:** vocabulário criado com base na rede *First* e com valores de *PRP* somados aos pesos dos termos indexados com o uso de *TF-IDF*;
2. ***Priors*:** vocabulário criado com base na rede *Priors* e com valores de *PRP* somados aos pesos dos termos indexados com o uso de *TF-IDF*;
3. ***Everyone*:** vocabulário criado com base na rede *Everyone* e com valores de *PRP* somados aos pesos dos termos indexados com o uso de *TF-IDF*;
4. ***First-TF-IDF*:** vocabulário criado com base na rede *First* e sem valores de *PRP* somados aos termos. Indexado apenas com o uso de *TF-IDF*;
5. ***Priors-TF-IDF*:** vocabulário criado com base na rede *Priors* e sem valores de *PRP* somados aos termos. Indexado apenas com o uso de *TF-IDF*;
6. ***Everyone-TF-IDF*:** vocabulário criado com base na rede Everyone e sem valores de *PRP* somados aos termos. Indexado apenas com o uso de *TF-IDF*.

Após a apresentação do processo de desenvolvimento do sistema de recomendações, na próxima seção será apresentada a metodologia utilizada para todo esse processo possa acontecer. Nela são apresentadas como são geradas e avaliadas as recomendações.

4 METODOLOGIA

Buscando medir e analisar os dados obtidos, foi criada uma metodologia para a apresentação dos resultados. Nesta seção será apresentada a forma como os dados são analisados e as métricas utilizadas. Será apresentado como foi elaborado o modelo de recomendação de contribuidores, como essas recomendações foram analisadas e a importância da divisão do projeto em períodos.

4.1 RECOMENDAÇÃO DE CONTRIBUIDORES

Para fazer a recomendação de contribuidores, é passado como consulta para o sistema de recomendação, os termos do título e da descrição de uma *issue* do período de recomendação, que é um período posterior ao período em que os contribuidores foram indexados. Com isso, o sistema retorna uma lista de contribuidores que possuem um *rank* mais elevado para os termos que foram passados como pesquisa. São passados também como parâmetros, o período de indexação de onde se deseja fazer a busca e qual a rede de comunicação, para que o sistema de recomendação faça a busca nos índices corretos.

Para os testes realizados neste trabalho, foram executadas uma busca para cada *issue* do período de recomendação em cada um dos seis índices de vocabulários dos contribuidores, de acordo com as redes de comunicação que foram indexadas.

Mais detalhes sobre a divisão dos períodos de recomendação e indexação podem ser encontrados na Seção 4.3.

4.2 MODELO DE AVALIAÇÃO

Com base no modelo de avaliação proposto por (HASSAN; HOLT, 2004), para a avaliação dos resultados, serão utilizadas três variáveis para poder determinar quais são os melhores resultados obtidos e assim poder avaliar as redes de comunicação. A seguir são apresentadas as variáveis e o modelo de cálculo de cada uma delas.

4.2.1 TOP N

Essa é uma variável que foi utilizada para definir o número máximo de contribuidores que seriam recomendado pelo sistema. Foram feitos testes de recomendação utilizando como valores de corte 10, 7, 5 e 3 contribuidores. Assim os cálculos de *recall* e *precision* puderam ser feitos.

4.2.2 ACERTOS DO RECOMENDADOR

Neste trabalho, foi consiredado como acerto do recomendador quando o sistema recomendou um contribuidor e o mesmo apareceu em comentários na *issue* em que foi recomendado.

4.2.3 *RECALL*

Essa variável será utilizada para calcular a sensibilidade do recomendador. Calcula o índice de acertos do recomendador em relação ao número de contribuidores que realmente comentaram na *issue*. Sua formula de cálculo é:

$$Recall = \frac{A}{O}$$

Onde, A é o número de acertos realizados pelo recomendador e O é o número de contribuidores observados nos comentários das *issues*.

Exemplo: Na lista de recomendação existe uma lista de contribuidores, onde o recomendador acertou quatro e nos comentários da *issue* existem cinco contribuidores. Nesse caso, *Recall* seria:

$$Recall = \frac{4}{5} = 0,8$$

Sendo assim, para a avaliação, quanto mais próximo de 1 for o valor de *Racall* melhor, o que significa que o recomendador acertou mais contribuidores.

4.2.4 *PRECISION*

Essa variável será utilizada para calcular a precisão do recomendador. Calcula o índice de acertos do recomendador em relação ao número de contribuidores que foram recomendados. Sua formula de cálculo é:

$$Precision = \frac{A}{R}$$

Onde, A é o número de acertos realizados pelo recomendador e R é o número de contribuidores recomendados pelo sistema.

Exemplo: Na lista de recomendação existem quatro recomendados e o recomendador acertou três. Nesse caso, *Precision* seria:

$$Precision = \frac{3}{4} = 0,75$$

Sendo assim, para a avaliação, quanto mais próximo de 1 for o valor de *Precision* melhor, o que significa que o recomendador teve uma precisão maior.

Como o número de contribuidores que realmente comentaram nas *issues* é variável, utilizou-se um valor de corte no número de contribuidores recomendados pelo *Top N*, apresentado anteriormente. Assim foi possível ter um melhor resultado nos valores de *precision*.

4.2.5 *LIKELIHOOD*

Em português, *LikeliHood* é a probabilidade de alguma coisa acontecer. Essa variável foi utilizada para calcular quantas *issues* dentro de um determinado período possuem pelo menos um contribuidor que foi recomendado pelo sistema.

Ela foi aplicada nas avaliações devido ao fato da média de contribuidores comentando em casa *issue* ser baixa. Com isso, o valor de *Precision* tende a ser baixo também. Então *LikeliHood* entra na avaliação para fazer a correção desse índice e mostrar que no cenário geral o fato de pelo menos um acerto por *issue* pode ter sido um resultado relevante.

4.3 DIVISÃO DO PROJETO EM PERÍODOS

Para a realização das análises, optou-se por dividir o projeto em mais de um período. Essa divisão foi feita após os primeiros testes com a ferramenta de recomendação por duas razões:

1. **Tempo de Processamento e Custo com Hardware:** Com o grande volume de dados que um projeto pode ter, tentar analisa-lo a partir de um computador com configurações de uso residencial, usando a metodologia aplicada neste trabalho e sem dividi-los em intervalos menores de tempo pode ser inviável devido ao tempo gasto com o processamento ou até mesmo devido às limitações de hardware;

2. **Recomendação de Contribuidores que Saíram do Projeto:** A rotatividade dos contribuidores foi um dos pontos que chamou a antenção quando foram realizados os primeiros testes com o sistema de recomendação. Como base nisso, surgiu a hipótese de que essa rotatividade poderia influenciar na recomendação de contribuidores. Assim, optou-se por dividir o projeto em intervalos menores de tempo para que isso pudesse ser verificado. Testando assim a hipótese de que, restringindo-se o intervalo de tempo para a indexação dos contribuidores pode-se conseguir melhores resultados na recomendação de contribuidores. Os resultados dessa analise são apresentados na Seção 5.1

Os períodos utilizados para as análises nesse trabalho são apresentados na Tabela 1:

Tabela 1: Períodos de Indexação e Recomendação dos Projetos

Período	Duração	Indexação	Recomendação
6-1M	12	01/01/2013 a 30/06/2013	01/07/2013 a 31/12/2013
3-1M	6	01/01/2013 a 31/03/2013	01/04/2013 a 30/06/2013
3-2M	6	01/04/2013 a 30/06/2013	01/07/2013 a 30/09/2013

Em todos os três projetos, foram analisados três intervalos de tempo diferentes, um com doze meses de duração e outros dois intervalos com seis meses, para observar o quanto o fator tempo influência na recomendação de contribuidores e tentar entender porque esse fator pode vir a influenciar nas recomendações.

Optou-se por dividir um intervalo maior de tempo (neste caso, o período de doze meses) em dois para que fosse possível analisar o impacto nas recomendações e também na rotatividade dos contribuidores. Quanto ao estudo ser com base em dados do ano de 2013, isso se deu devido ao fato desses projetos já terem sidos minerados anteriormente com a ferramenta JGitWebMiner.

Cada intervalo apresentados na Tabela 1 foi subdivido em dois. Assim, por exemplo, o período $6-1M$ que contém um intervalo de 12 meses do projeto foi subdividido em dois períodos, o **Período de Indexação** com seis meses e o **Período de Recomendação**, também com seis meses. Por essa razão o período foi nomeado de $6-1M$. Essa divisão foi utilizada porque cada um dos dois períodos tem uma função distinta no sistema de recomendação, como pode ser visto abaixo:

1. **Período de Indexação:** nesse período são gerados os vocabulários dos contribuidores, as redes de comunicação existentes entre eles e todo o processo de indexação desses contribuidores. Esse processo de indexação é explicado com detalhes na Seção 3.8;

2. **Período de Recomendação:** esse período é utilizado para os testes no sistema de recomendação. A partir das *issues* desse período, são geradas as consultas que serão passadas ao recomendador. Para avaliação das recomendações de cada um dos períodos é feita uma comparação entre a lista de contribuidores geradas pelo recomendador e os contribuidores que comentaram nas *issues* do perído de recomendação. E então, é aplicado o modelo de avaliação apresentado na Seção 4.2. Mais detalhes sobre o processo de recomendação de contribuidores pode ser visto na Seção 4.1

A seguir, no Capitulo 5 são apresentados os resultados que foram obtidos com a utilização da metodologia apresentada neste trabalho. Serão apresentadas dados estatísticos sobre os projetos, a rotatividade dos contribuidores, as comparações entre as redes de comunicação, comparações entre períodos utilizados e comparações entre a indexação via *TF-IDF* com a indexação utilizando *PRP*.

5 RESULTADOS E DISCUSSÃO

Nesta seção serão apresentadas as informações levantadas com base nos dados de três projetos de software livre que foram escolhidos para esse trabalho, Node, Bootstrap e Homebrew. Para um melhor embasamento, serão apresentados nessa seção alguns dados estatísticos de cada projeto.

Os primeiros passos dados com a ferramenta JGitRecommender, que foi desenvolvida para gerar as recomendações, basearam-se em analisar o projeto desde o início até o período de tempo atualonde o teste foi realizado, para que assim gerasse as recomendações. Alguns pontos foram observados com isso. O primeiro, seria o custo de hardware para processar todas essas informações, visto que alguns projetos estão sendo desenvolvidos há vários anos. Outro ponto que foi analisado foi a rotatividade dos contribuidores dentro do projeto.

Não se tinha uma dimensão exata sobre o percentual de rotatividade, mas ao longo da pesquisa surgiu a hipótese de que a rotatividade dos contribuidores poderia influenciar em grandes níveis a recomendação de *experts*. Visto que se fosse indexado todo o projeto para depois fazer as recomendações, corria-se o risco do sistema de recomendação gerar uma lista com contribuidores que já deixaram o projeto. A apresentação dos dados sobre a rotatividade dos contribuidores será apresentada na Seção 5.1

5.1 RESUMO DOS PROJETOS E A ROTATIVIDADE DE CONTRIBUIDORES

A seguir serão apresentados os projetos que foram analisados, com dados estatísticos sobre os projetos e seus respectivos resultados com base na hipótese da influência da rotatividade de contribuidores nas recomendações. Para facilitar a leitura serão separados por projetos.

5.1.1 PROJETO NODE

O projeto joyent/node, segundo sua documentação [1], é uma plataforma que foi desenvolvida em Javascript e tem como objetivo construir aplicações de rede rápidas e escaláveis. Atualmente, em junho de 2015, o projeto conta com 615 colaboradores, 258 releases e mais de 10 mil *commits* e mais de 8 mil *forks*, que são cópias do repositório do projeto para os repositórios dos contribuidores. Geralmente usa-se o *fork* para gerar uma versão própria do projeto, podendo fazer alterações no código fonte. Na Tabela 2 são apresentadados os dados sobre um dos períodos analisados no projeto Node.

Tabela 2: Dados Sobre o Período Analisado no Projeto Node

Nome do Projeto	**Node**		
Período	**6-1M**		
Duração (meses)	**12**		
Data de Início de Indexação	**01/01/2013**	Data de Início de Análise	**01/07/2013**
Data de Fim de Indexação	**30/06/2013**	Data de Fim de Análise	**31/12/2013**

O projeto Node é o menor dos três projetos analisados nesse trabalho. Como pode ser observado na Tabela 3, durante o período de indexação de seis meses, o projeto Node contou com a participação de 701 contribuidores e teve 1004 *issues* abertas e concluídas dentro desse intervalo de tempo. Vale a pena lembrar que foi chamado de contribuidor todos que participaram de alguma *issue* no projeto, enquanto os colaboradores são os que além de participarem de discussões em *issues*, também fazem alterações no código e fazem *commits*. Essa é a razão pela qual os projetos possuem mais contribuidores do que colaboradores nas estatísticas.

Tabela 3: Dados do Perído de Indexação do Projeto Node

Contribuidores	**701**
Issues	**1.004**
Comentários	**4.309**
Contribuições Únicas	**2.353**
Média Comentários/Issue	**4,29**
Média Contribuidores/Issue	**2,34**
Média Comentários/Contribuidores	**6,15**
Média Contribuições Únicas/Contribuidores	**3,36**

[1] Disponível em https://nodejs.org

Depois de fazer a indexação dos dados do projeto Node, um dos resultados obtidos com a ferramenta JGitRecommender foram estatísticas sobre cada período. Algumas dessas estatísticas são apresentadas nas Tabelas 2, 3, 4 e 5. Para uma melhor compreenção das tabelas de estatísticas do projeto Node, foi criada uma lista com algumas das legendas:

1. **Contribuidores:** Número total de contribuidores que participaram de alguma *issue* no período;
2. ***Issues*:** Número total de discussões sobre algum tópico no período;
3. **Comentários** Número total de comentários no período;
4. **Contribuições Únicas:** Número total de comentários sem repetir contribuidor em uma *issue* (Se comentar mais de uma vez na *issue* só conta uma vez);
5. **Contribuidores Ativos:** contribuidores que participaram em ambos períodos;
6. **Percentual de Contribuidores Ativos:** Percentual de contribuidores que podem ser recomendados e possuem potencial para colaborar;
7. **Entrada de Contribuidores:** O número total de contribuidores que entraram no projeto durante o período de recomendação;
8. **Saída de Contribuidores:** O número total de contribuidores que participaram no período de indexação mas apareceram em discussões no período de recomendação.

Na Tabela 4 são apresentadados os dados sobre o período de recomendação do projeto Node. O que pode ser observado nessa tabela é uma redução nas atividades do projeto. Nesse período foram abertas menos *issues* e com isso, menos contribuidores participaram.

Tabela 4: Dados do Perído de Recomendação do Projeto Node

Contribuidores	**513**
Issues	**757**
Comentários	**2.976**
Contribuições Únicas	**1.733**
Média Comentários/Issue	**3,93**
Média Contribuidores/Issue	**2,29**
Média Comentários/Contribuidores	**5,80**
Média Contribuições Únicas/Contribuidores	**3,38**

Uma das coisas que pode ser observada na Tabela 5 é a alta rotatividade de contribuidores no projeto. Arredondando os valores, observa-se na linha *Percentual de Contribuidores Ativos*, uma retenção de apenas 16% dos contribuidores. Com essas informações é possível considerar que, mesmo existindo um grande número de contribuidores no projeto, apenas uma pequena parte desse grupo estará disponível para o próximo período. O que dá uma fundamentação para hipótese de que o fator tempo influencia na recomendação dos contribuidores. Visto que contribuidores que estavam em um período e não estão em outro, caso um deles seja recomendado, mesmo esse possuindo todas as características necessárias para poder colaborar, não irá contribuir por não estar no projeto.

Tabela 5: Alterações Entre os Períodos de Indexação e Recomendação no Projeto Node

Contribuidores Ativos	**111**
Entrada de Contribuidores	**402**
Saida de Contribuidores	**590**
Percentual de Entrada de Contribuidores	**78,36%**
Percentual de Saida de Contribuidores	**84,17%**
Percentual de Contribuidores Ativos	**15,83%**

A mesma análise foi feita nos três períodos e seu dados foram simplificados na Tabela 6. O objetivo de observar períodos com intervalos diferentes era ver se a rotatividade dos contribuidores seria igual se os períodos fossem mais curtos. O que foi observado é que para os três casos, a rotatividade foi muito parecida.

Tabela 6: Rotatividade de Contribuidores no Projeto Node

Período	6-1M	3-1M	3-2M
contribuidores Ativos	111	69	61
Entrada de Contribuidores	402	274	210
Saida de Contribuidores	590	327	282
Percentual de Entrada de Contribuidores	78%	80%	77%
Percentual de Saida de Contribuidores	84%	83%	82%
Percentual de Contribuidores Ativos	16%	17%	18%

Considerando que a rotatividade seja alta independentemente do tamanho do intervalo de tempo, a Tabela 6 sugere que a recomendação onde o objetivo é recomendar alguém para participar de uma *issue*, seja feita utilizando períodos menores. Mas antes de determinar o tamanho desse período deve ser analisado com qual intervalo de tempo o vocabulário dos contribuidores ficará consistente o suficiente para produzir uma recomendação relevante.

5.1.2 PROJETO BOOTSTRAP

O projeto Bootstrap, segundo sua documentação [2], é um *framework* para desenvolvimento responsivo de páginas *Web*, sem a necessidade do programador se preocupar em fazer alterações no código para que o site seja visualizado corretamente tanto em dispositivos móveis quanto em computadores. Atualmente, em junho de 2015, o projeto conta com 655 colaboradores, 33 releases e mais de 11 mil *commits*. Na Tabela 7 são apresentadados os dados sobre um dos períodos analisados no projeto Bootstrap.

Tabela 7: Dados Sobre o Período Analisado no Projeto Bootstrap

Período	**6-1M**		
Duração (meses)	**12**		
Data de Início de Indexação	**01/01/2013**	Data de Início de Análise	**01/07/2013**
Data de Fim de Indexação	**30/06/2013**	Data de Fim de Análise	**31/12/2013**

Assim como no projeto Node, foram elaboradas para o projeto Bootstrap as Tabelas 7, 8, 9 e 10, que contém as estatísticas de seis meses do projeto, nas quais são apresentados o número de contribuidores, a quantidade de *issues* e de comentários, o número de novatos e de desistentes no projeto, dentre outras informações relevantes.

Tabela 8: Dados do Perído de Indexação do Projeto Bootstrap

Contribuidores	**1.928**
Issues	**1.772**
Comentários	**4.872**
Contribuições Únicas	**3.688**
Média Comentários/Issue	**2,75**
Média Contribuidores/Issue	**2,08**
Média Comentários/Contribuidores	**2,53**
Média Contribuições Únicas/Contribuidores	**1,91**

Em comparação ao projeto Node no mesmo período, pode ser observado na Tabela 8 que o projeto Bootstrap apresentou um maior número de contribuidores, chegando a ter quase três vezes mais contribuidores. Mas o número de *issues* não aumentou na mesma escala.Mesmo possuindo mais contribuidores, tanto o número de comentários quanto o número de contribuidores por *issue* foi menor, o que gera uma redução nas redes de recomendação.

[2] http://getbootstrap.com/

Ao contrário do projeto Node, o Bootstrap apresentou um aumento nas atividades do projeto no período de recomendação. Como pode ser visto na Tabela 9, houve um aumento no número de *issues* abertas e também no número de contribuidores.

Tabela 9: Dados do Perído de Recomendação do Projeto Bootstrap

Contribuidores	**2.703**
Issues	**3.620**
Comentários	**10.690**
Contribuições Únicas	**7.109**
Média Comentários/Issue	**2,95**
Média Contribuidores/Issue	**1,96**
Média Comentários/Contribuidores	**3,95**
Média Contribuições Únicas/Contribuidores	**2,63**

Uma informação interessante que pode ser retirada da Tabela 10 é a rotatividade de contribuidores. Como pode ser visualizado, no projeto Bootstrap o percentual de retenção de contribuidores foi de 12%. Esse foi um valor bem menor se comparado ao projeto Node, onde esse valor foi de 16%. Apenas comparando os percentuais esses valores não parecem representar tanta diferença, mas se for comparado o número de contribuidores de cada projeto, então passa a ter mais relevância. No projeto Node, o número de contribuidores foi de 701, enquanto que no Bootstrap esse número foi de 1928, quase três vezes a quantidade de contribuidores do outro projeto.

Tabela 10: Alterações Entre os Períodos de Indexação e Recomendação no Projeto Bootstrap

Colaboradores Ativos	**227**
Entrada de Colaboradores	**2.476**
Saida de Colaboradores	**1.701**
Percentual de Entrada de Colaboradores	**91,60%**
Percentual de Saida de Colaboradores	**88,23%**
Percentual de Colaboradores Ativos	**11,77%**

Dos três projetos analisado o Bootstrap foi o que apresentou o pior percentual de retenção de contribuidores. Como pode ser visto na Tabela 11, chegou a ter 93% de desistência dos contribuidores que entraram no período 3-1M,. Novamente reforçando a hipótese de influência do intervalo de tempo nas recomendações de *experts*.

Tabela 11: Rotatividade de contribuidores no Projeto Bootstrap

Período	6-1M	3-1M	3-2M
contribuidores Ativos	227	66	100
Entrada de Contribuidores	2476	656	1721
Saida de Contribuidores	1701	903	622
Percentual de Entrada de Contribuidores	92%	91%	95%
Percentual de Saida de Contribuidores	88%	93%	86%
Percentual de Contribuidores Ativos	12%	7%	14%

Para finalizar a apresentação dos projetos, o último a ser apresentado é o projeto Homebrew. Os dados estatísticos e as analises feitas com base nesse projeto são apresentadas a seguir.

5.1.3 PROJETO HOMEBREW

O projeto Homebrew, segundo sua documentação [3], é um gerenciador de pacotes para sistemas operacionais OS X da Apple. Ele atua de forma semelhante aos gerenciadores de pacotes do Linux, auxiliando na instalação de programas. Atualmente, em junho de 2015, o projeto conta com 4862 colaboradores e quase de 49 mil *commits*. Até o momento ele não apresenta nenhuma release. Na Tabela 12 são apresentadados os dados sobre um dos períodos analisado no projeto Homebrew.

Tabela 12: Dados Sobre o Período Analisado Projeto Homebrew

Nome do Projeto	**Homebrew**		
Período	**6-1M**		
Duração (meses)	**12**		
Data de Início de Indexação	**01/01/2013**	Data de Início de Análise	**01/07/2013**
Data de Fim de Indexação	**30/06/2013**	Data de Fim de Análise	**31/12/2013**

Para manter um padrão de apresentação dos projetos, as Tabelas 12, 13, 14 e 15, apresentam as estatísticas do projeto Homebrew em um período de doze meses.

Entre os três projetos analisado, o Homebrew é o maior de todos. Como pode ser visto na Tabela 13, tanto o número de contribuidores quanto o número de *issues* foi superior aos

[3] http://brew.sh/

demais projetos analisados nesse trabalho. Com isso, este projeto tem redes de comunicação maiores do que os outros que foram vistos anteriormente.

Tabela 13: Dados do Perído de Indexação do Projeto Homebrew

Contribuidores	**2.586**
Issues	**3.820**
Comentários	**11.819**
Contribuições Únicas	**6.216**
Média Comentários/Issue	**3,09**
Média Contribuidores/Issue	**1,63**
Média Comentários/Contribuidores	**4,57**
Média Contribuições Únicas/Contribuidores	**2,40**

Este projeto, assim como o projeto Bootstrap, teve um aumento nas atividades no período de recomendação. Como pode ser visto na Tabela 14, praticamente todos os campos da tabela aumentaram de um período para o outro. O único que teve uma leve queda foi o índice de $Contribuidores/Issue$.

Tabela 14: Dados do Perído de Recomendação do Projeto Homebrew

Contribuidores	**2.986**
Issues	**4.293**
Comentários	**15.550**
Contribuições Únicas	**7.796**
Média Comentários/Issue	**3,62**
Média Contribuidores/Issue	**1,82**
Média Comentários/Contribuidores	**5,21**
Média Contribuições Únicas/Contribuidores	**2,61**

Um ponto interessante em comparação com os outros projetos é que no Homebrew, mesmo ele sendo o maior dos três projetos analisados, a rotatividade de contribuidores foi a menor, como pode ser visto na Tabela 15. Com esses dados em mãos, o que foi observado é aparentemente, não existe uma relação entre a quantidade de contribuidores e a alta rotatividade.

Tabela 15: Alterações Entre os Períodos de Indexação e Recomendação no Projeto Homebrew

Contribuidores Ativos	**549**
Entrada de Contribuidores	**2.437**
Saida de Contribuidores	**2.037**
Percentual de Entrada de Contribuidores	**81,61%**
Percentual de Saida de Contribuidores	**78,77%**
Percentual de Contribuidores Ativos	**21,23%**

Em comparação aos demais projetos já analisados nesse trabalho, o nível de permanência dos contribuidores foi maior no Homebrew. Como pode ser visto na Tabela 16, no período de doze meses, o projeto conseguiu manter 21% dos contribuidores do período de indexação para o período de recomendação, enquanto no Node foi de 16% e no Bootstrap foi de 12% no mesmo período.

Tabela 16: Rotatividade de contribuidores no Projeto Homebrew

Período	6-1M	3-1M	3-2M
contribuidores Ativos	549	241	246
Entrada de Contribuidores	2437	1136	1053
Saida de contribuidores	2037	1040	1131
Percentual de Entrada de Contribuidores	82%	82%	81%
Percentual de Saida de Contribuidores	79%	81%	82%
Percentual de Contribuidores Ativos	21%	19%	18%

Na próxima seção serão apresentados os resultados das análises feitas com os dados gerados pelo sistema de recomendação. Nela serão apresentados os comaprativos entre projetos, redes de comunicação entre outras coisas que foram apresentadas na Seção 4.

5.2 ANÁLISE DO SISTEMA DE RECOMENDAÇÃO

Nessa seção serão analisados os dados de saída do sistema de recomendação de *experts* que foi desenvolvido ao longo desta pesquisa. Como já foi mencionado anteriormente, esse sistema foi nomeado de JGitRecommender e tem seu foco em encontrar contribuidores que sejam capazes de responder uma *issue*. Aqui serão analisadas as diferenças entre as redes de comunicação, procurando entender se existe alguma que apresente melhores resultados. A diferença entre a utilização de ganho com *PRP* em relação à indexação convencional do vocabulário dos

contribuidores. E também a diferença entre considerar todos os contribuidores ou considerar apenas os contribuidores ativos ao gerar as listas de recomendação.

Para as análises dessa seção serão utilizadas as tabelas 17, 18 e 19, com os valores de *Recall*, *Precision* e *LikeliHood* médios de cada um dos três projetos. Cada uma dessas tabelas possui quatro conjunto de informações que serão utilizados para as comparações necessárias. Nelas temos:

1. Grupo com *Recall Precision* utilizando *PRP* das Redes de Comunicação. Neste grupo estão as redes *First*, *Priors* e *Everyone*. Este por sua vez é dividido em dois:
   (a) grupo de dados onde são considerados os contribuidores ativos e;
   (b) grupo que considera todos os contribuidores para a recomendação.
2. Grupo com *Recall Precision* utilizando apenas *TF-IDF*. Neste grupo estão as redes *TF-IDF-First*, *TF-IDF-Priors* e *TF-IDF–Everyone*. Esse grupo utilizou o conceito de Redes de Comunicação apenas para a produção do vocabulário dos contribuidores. Da mesma forma que o grupo anterior, este também é dividido em dois:
   (a) grupo de contribuidores ativos e;
   (b) grupo que considera todos os contribuidores para a recomendação.

5.2.1 COMPARATIVO ENTRE A INDEXAÇÃO COM *PRP* EM RELAÇÃO AO *TF-IDF*

Com base na metodologia utilizada nesse trabalho e nos dados fornecidos pelas tabelas 17, 18 e 19, pode ser observado que a utilização do *PRP*, obtido através da geração de redes de comunicação criadas a partir da interação entre os contribuidores nas *issues*, apresentou uma pequena vantagem em relação à indexação realizada apenas com *TF-IDF* .

Tabela 17: Médias de *Recall Precision* projeto Node

NodeJS							
Rede	**Issues**	**T. Periodos - T. Desenv.**			**T. Periodos - Desenv. Ativos**		
		Recall	**Precision**	**LikeliHood**	**Recall**	**Precision**	**LikeliHood**
First	5820	0,581	0,217	86,48%	0,697	0,219	86,89%
Priors	5820	0,578	0,216	86,01%	0,693	0,218	86,48%
Everyone	5820	0,577	0,214	86,24%	0,692	0,216	86,58%
TF_IDF_First	5820	0,555	0,208	83,45%	0,673	0,212	84,55%
TF_IDF_Priors	5820	0,555	0,208	83,45%	0,673	0,212	84,55%
TF_IDF_Everyone	5820	0,509	0,192	78,35%	0,632	0,198	80,58%

Tabela 18: Médias de *Recall Precision* projeto Bootstrap

Bootstrap							
Rede	**Issues**	**T. Periodos - T. Desenv.**			**T. Periodos - Desenv. Ativos**		
		Recall	**Precision**	**LikeliHood**	**Recall**	**Precision**	**LikeliHood**
First	23384	0,622	0,212	89,12%	0,814	0,215	89,94%
Priors	23384	0,623	0,212	89,20%	0,814	0,215	90,01%
Everyone	23384	0,622	0,212	89,18%	0,811	0,214	89,78%
TF_IDF_First	23384	0,599	0,203	86,27%	0,798	0,209	88,36%
TF_IDF_Priors	23384	0,599	0,203	86,27%	0,798	0,209	88,36%
TF_IDF_Everyone	23384	0,570	0,191	82,74%	0,774	0,201	86,03%

Comparando os piores casos de *TF-IDF*, que foram os que utilizaram a rede *Everyone* para a geração do vocabulário e são representados nas tabelas como *TF-IDF-Everyone*, com os melhores resultados que utilizam *PRP*, observou-se a uma diferença que variou de 7% a 18% dependendo do projeto. O projeto mais impactado com a utilização do *PRP* foi o projeto Homebrew. Como pode ser visto na tabela 19, o valor de *LikeliHood*, que indica quantas vezes o recomendador acertou pelo menos um contribuidor que comentou na *issue*, teve uma certa relevância, isso porque, das 22332 *issues* do período, utilizando apenas *TF-IDF* o sistema acertou 15841 enquanto com o uso do *PRP* na indexação, chegou a acertar pelo menos um contribuidor em 19443 *issue* na rede *First*, resultando em 18% de ganho.

Tabela 19: Médias de *Recall Precision* projeto Homebrew

Homebrew							
Rede	**Issues**	**T. Periodos - T. Desenv.**			**T. Periodos - Desenv. Ativos**		
		Recall	**Precision**	**LikeliHood**	**Recall**	**Precision**	**LikeliHood**
First	22332	0,595	0,238	87,02%	0,776	0,247	87,24%
Priors	22332	0,585	0,231	85,96%	0,763	0,240	86,16%
Everyone	22332	0,592	0,234	86,72%	0,791	0,248	86,90%
TF_IDF_First	22332	0,489	0,194	73,98%	0,681	0,214	76,40%
TF_IDF_Priors	22332	0,489	0,194	73,98%	0,681	0,214	76,40%
TF_IDF_Everyone	22332	0,469	0,184	70,93%	0,657	0,204	73,73%

O projeto Node, considerando a mesma análise feita na tabela do projeto Homebrew, teve um ganho de 10% no acerto de pelo menos um contribuidor por *issue*. Se for considerado apenas o aumento no número de acertos, pode ter-se a impressão que não houve relevância, isso porque o projeto Node é o menor dos três analisados, então com isso o número de *issues*

e contribuidores pode ser menos da metade dos outros projetos. Assim sendo, é melhor olhar para o percentual de ganho.

O projeto Bootstrap, que como visto anteriormente, foi o que apresentou o maior nível de rotatividade de contribuidores, nesse caso foi o que apresentou o menor índice de relevância entre indexar o vocabulário com *PRP*. Este projeto teve apenas 8% de diferença entre os dois modelos de indexação. Uma possível explicação para isso pode ser o fato de um grupo menor de contribuidores ser o responsável por comentar nas *issues* do período, fazendo com que o vocabulário desses contribuidores fique mais relevantes em relação aos demais e com isso utilizar um ganho nos termos fez menos diferença.

Mas antes de afirmar que é relevante utilizar o valor de *PRP* na indexação do vocabulário, devem ser analisados os outros dados, no caso, o filtro de contribuidores ativos. Essa é a análise que será apresentada ao longo desse trabalho.

5.2.2 COMPARATIVO ENTRE OS MODELOS DE REDES DE COMUNICAÇÃO

A análise das redes de comunicação ficará limitada em observar apenas qual dos modelos de geração de vocabulário apresenta o melhor resultado na recomendação de contribuidores. Questões como, se é relevante ou não utilizar as redes de comunicação serão abordadas em outras analises ao longo desse trabalho.

No que se diz respeito ao modelo de geração dos vocabulários, a rede de comunicação que apresentou o melhor resultado, ainda que pouco expressivo em relação às demais, foi a rede *First*. Esta rede, levando em consideração seu processo de criação, é a mais simples de ser implementada, com isso, sugere-se a sua utilização. Como o ganho com as demais redes foi inferior, mesmo que muito próximo, o custo da implementação dessas outras redes não se justifica.

Para essa análise, foram criadas algumas tabelas parecidas com as tabelas 17, 18 e 19, para cada um dos três projetos, para analise de *Recall*, *Precision*, *LikeliHood* e *Top N*. Onde foram analisados os três períodos de tempo. O que foi observado ao longo da pesquisa é que em alguns casos onde os períodos eram maiores, considerando os períodos de seis meses, a rede *Everyone* chegou se apresentar superior e em outros casos em períodos de tempo menores, no caso, três meses, a rede *Priors* apresentou algum ganho. Mas em ambos os casos, a diferença ficou em no máximo um decimo, o que não justifica sua utilização. Vale ressaltar que esses ganhos não aconteceram em todos os períodos e sim em alguns casos.

Uma possível explicação para essa rede ter apresentado melhores resultados é o fato

de que com ela, o contribuidor tem em seu vocabulário apenas os termos que ele mesmo usou nas discussões, com o acréscimo do título e descrição da *issue*. O que faz que ele contenha em seu vocabulário os termos do problema e seus termos apresentados como solução.

5.2.3 COMPARATIVO ENTRE A UTILIZAÇÃO DE UM FILTRO PARA CONTRIBUIDORES ATIVOS EM RELAÇÃO À INCLUSÃO DE TODOS CONTRIBUIDORES DO PERÍODO

Ao longo da pesquisa surgiu a hipótese de que se fossem cortados todos os contribuidores que estavam no período de indexação, mas que não possuem nenhuma interação com outros contribuidores no período de recomendação, isso aumentaria a taxa de acertos do recomendador. Esse teste foi realizado e resultou em alguns dados interessantes.

O que foi observado é que houve um aumento nas três variáveis utilizadas, *Recall*, *Precision* e *LikeliHood*. Com os dados obtidos através das 39 tabelas, foi observado que em o número de acertos do recomendador não foi tão expressivo quando comparado apenas com a utilização do *PRP*. Mas foi bem expressivo quando comparado a indexação por TF-IDF.

Olhando para a tabela 18, onde são apresentadas as médias do projeto Bootstrap, pode ser visto que o ganho no *Recall* chegou a dois décimos, o que representa 20% do total do *Recall*. Outro ponto interessante baseando-se nesses dados é o fato de apenas por filtrar os contribuidores ativos, teve-se um ganho em relação à apenas utilizar o *PRP* mantendo todos os contribuidores.

Mas ainda assim, a indexação com *PRP* e a aplicação do filtro de contribuidores ativos foi a que apresentou os melhores resultados de todos, chegando a ter em algumas das tabelas de intervalos de três meses do projeto Bootstrap, valores de *Recall* superiores a 0,9, o que significa que praticamente todos os recomendados pelo sistema participaram das discussões nas *issues*.

Vale ressaltar que nesses períodos, os maiores valores de *Recall* foram 0,92 onde foram aplicados, o filtro e a indexação com *PRP*, mas ao mesmo tempo, onde o vocabulário foi indexado apenas com *TF-IDF* e também foi aplicado o filtro, os valores de *Recall* foram de 0,91.

Essas informações acabam colocando em dúvida a real vantagem na utilização dos valores de *PRP* na indexação do vocabulário dos contribuidores. Como pode ser visto, apenas cortando os contribuidores inativos do projeto já houve um ganho significativo nas recomendações.

5.2.4 CONSIDERAÇÕES SOBRE OS RESULTADOS

Olhando para os resultados e focando apenas nos valores de *Precision*, tem-se a impressão que o recomendador não se saiu bem, mas se for levado em consideração os valores de *LikeliHood*, nota-se que o resultado melhorou, uma vez que nos três projetos, os valores de *LikeliHood* ficaram acima de 80%. No caso especifico do projeto Bootstrap usando o filtro de usuários ativos, a média de vezes que o recomendador acertou pelo menos um contribuidor recomendado por *issue* foi de 89%.

Uma explicação para os baixos valores de *Precision* é resultado da baixa média de contribuidores únicos comentando em cada *issue*. Como foi visto nas estatísticas de cada projeto, a média foi de aproximadamente 2 contribuidores por *issue*. Considerando que o recomendador gerou listas com 10, 7, 5 e 3 contribuidores, o índice de *Precision* de um modo geral tende a ser baixo.

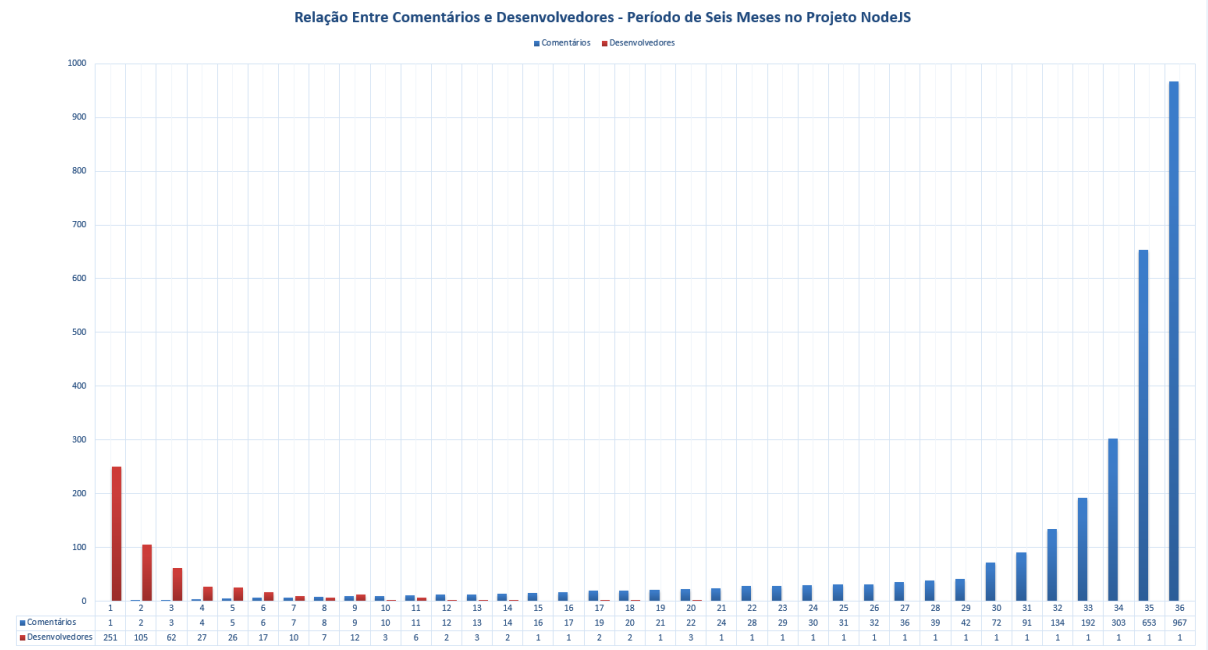

Figura 11: Relação Entre Comentários e contribuidores.

Outro dado interessante encontrado com a pesquisa, foram os valores de comentário por contribuidor. Como pode ser visto na Figura 11 que representa os dados do projeto Node em um período de seis meses, existe um grande número de contribuidores que possuem quantidades consideráveis de comentários e ao mesmo tempo existe um grande número de contribuidores que comentam apenas uma vez e não interagem mais.

Ao ver esse histograma, surgiu a hipótese de que os 20% dos contribuidores que mais comentam no período são os contribuidores que se mantem no projeto por mais tempo. O que

se observou, com uma inspeção no GitHub, foi que nos três projetos analisados, pelo menos três dos contribuidores que mais comentaram em 2013, que foi o ano utilizado para a pesquisa, ainda são os contribuidores com o maior número de *commits* nos projetos, o que dá a entender que ainda estão ativos no projeto.

Olhando para a Figura 11, pode ser visto que pelo menos metade dos contribuidores possuem menos de cinco comentários durante o período de indexação. A partir desta informação surge a hipótese de que esses contribuidores podem ser cortados do sistema de recomendação desde o início, reduzindo assim os custos de processamento de informação. Mas essa hipótese não foi testada nesse trabalho, fica a critério de novos pesquisadores testarem se isso oferece ou não melhores resultados.

6 CONCLUSÕES E TRABALHOS FUTUROS

Com base nos objetivos específicos estabelecidos no início deste trabalho, pode-se dizer que ele apresentou um bom resultado. Embora alguns resultados tenham sido diferentes do que se esperava no início da pesquisa, eles foram interessantes por esclarecer alguns pontos. Um dos resultados obtidos com a pesquisa que apresentou-se diferente do esperado foi do primeiro objetivo específico. O objetivo era comparar o uso de algoritmos que utilizem puramente *Term Frequency (TF)* e *Term Frequency – Inverse Document Frequency*,*(TF-IDF)*, com um algoritmo que incremente os *rankings* dos contribuidores através do valor de *PRP*.

Esperava-se que o resultado fosse uma grande diferença entre as duas abordagens, onde a utilização do *PRP* apresentasse melhores resultados. O que se percebeu é que existe sim um ganho com sua utilização, mas não foram valores que justifiquem, de acordo com a metodologia utilizada, a geração das redes de comunicação para cálculo de *PRP*, devido ao custo de processamento de dados. O mais indicado é a utilização das redes de comunicação para a formação do vocabulário dos contribuidores, visto que apresentou diferenças na recomendação entre os modelos de redes.

Para a obtenção dos resultados, o segundo objetivo específico proposto foi modelar uma rede de comunicação entre contribuidores considerando em quais *issues* eles escrevem e o conteúdo que eles escrevem nessas *issues*. Essas redes foram criadas e completaram esse objetivo. Como foi visto ao longo deste trabalho, as redes de comunicação criadas foram *First*, *Priors* e *Everyone*.

Após a criação dessas redes serem finalizadas entra o terceiro objetivo que era comparar as três redes de comunicação geradas para definir se o uso delas é capaz de auxiliar na recomendação de *experts*, caso seja, apresentar qual teve o melhor resultado. Como já discutido acima, o resultado da utilização das redes não foi significativo. Mas observando as redes entre si, observou-se que a rede *First* apresentou melhores resultados em relação às demais redes, o que é um ponto positivo, visto que a sua implementação é a mais simples de todas.

Com base nos dados obtidos, acredita-se que a maneira mais eficiente de geração do

vocabulário dos contribuidores, de acordo com a metodologia utilizada neste trabalho, seja utilizando a rede *First*. Aparentemente, manter no vocabulário de cada contribuidor apenas os termos utilizados por ele mesmo somado com o título de descrição da *issue* apresenta melhores resultados na recomendação.

O último dos objetivos específicos era observar qual a influência que o fator tempo inserido no processo de indexação gera nos resultados da recomendação. O que observou-se foi que o intervalo de tempo tem forte influência na recomendação, em vista à grande rotatividade de contribuidores. Nesse ponto duas coisas foram analisadas, uma delas é a necessidade da definição de um intervalo de tempo para a definição de contribuidores ativos e outra é o custo do processamento de dados em períodos muito longos. Para analisar um projeto inteiro pode ser gasto muito tempo, dependendo do tamanho do projeto e corre-se o risco de gerar uma lista de recomendados que não corresponde ao grupo de contribuidores que está ativo no projeto para poder ajudar.

Enfim, para finalizar a pesquisa foram destacadas algumas pesquisas que podem dar continuidade a esse trabalho e produzir melhores resultados. Essas sugestões estão na seção de Trabalhos Relacionados. E alguns pontos que limitam este trabalho estão a seguir na seção de Limitações.

6.1 LIMITAÇÕES

Um ponto que deve ser lavado em consideração é o fato do recomendador não saber que alguns novatos podem pessoas indicadas à recomendação. O modelo apresentado leva em consideração um determinado período de tempo para a geração do vocabulário dos contribuidores. Com isso, apenas os contribuidores que estavam no projeto durante o período de indexação poderão ser recomendados pelo sistema. Isso limita as recomendações, porque dentro do período de testes podem existir contribuidores que sejam capazes de responder as dúvidas de outros contribuidores, mas o sistema não tem essa informação.

6.2 TRABALHOS FUTUROS

Durante o decorrer da pesquisa foram surgindo ideias de implementação que poderiam apresentar melhores resultados. Mas para não perder o foco dos objetivos específicos estabelecidos no início da pesquisa, essas ideias ficaram apenas na lista de trabalhos futuros.

Como já foi comentado, existe um grande número de contribuidores que abandonam os

projetos de software livre, mas ao mesmo tempo também existe um grande número de novatos que estão entrando nos projetos. Para exemplificar, a figura 12 representa a rotatividade dos contribuidores nos três projetos analisados neste trabalho, considerando um período de doze meses de projeto.

Figura 12: Rotatividade de contribuidores.

Com base nesses dados, uma das indicações para uma futura pesquisa, seria excluir os contribuidores inativos desde o início da geração e indexação dos vocabulários. E ao mesmo tempo, também fazer uma exclusão dos contribuidores que possuem apenas um comentário, já que esses apresentam pouca influência sobre o projeto. Isso reduzirá de forma significativa o tempo de processamento dos dados do projeto. Dessa forma, torna-se viável a indexação do vocabulário utilizando os valores de *PRP*.

Visto que nessa pesquisa o recomendador foi testado acertando ou não os contribuidores que comentaram nas *issues*, outra pesquisa que pode apresentar bons resultados seria a utilização do sistema de recomendação proposto nessa pesquisa em um projeto na prática, enviando uma notificação para o contribuidor recomendado, seja para participar em uma *issue* ou simplesmente ajudar um novato com dúvidas. Feito isso, testar se os recomendados são ou não as pessoas certas para a recomendação.

Essa sugestão vem da hipótese de que não é porque o contribuidor recomendado pelo sistema não comentou na *issue* que ele não esteja apto para ajudar a resolve-la. Imagina-se que se o contribuidor recomendado receber uma notificação de que ele foi recomendado, talvez ele possa vir a ajudar, seja comentando em uma *issue* ou ajudando um novato.

REFERÊNCIAS

BIRD, C.; PATTISON, D.; D'SOUZA, R.; FILKOV, V.; DEVANBU, P. Latent social structure in open source projects. In: **Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering**. New York, NY, USA: ACM, 2008. (SIGSOFT '08/FSE-16), p. 24–35. ISBN 978-1-59593-995-1. Disponível em: `http://doi.acm.org/10.1145/1453101.1453107`.

DEMARTINI, G. Finding experts using wikipedia. In: **Proceedings of the Workshop on Finding Experts on the Web with Semantics (FEWS2007) at ISWC/ASWC2007, Busan, South Korea**. [S.l.: s.n.], 2007. p. 33–41.

HASSAN, A.; HOLT, R. Predicting change propagation in software systems. In: **Software Maintenance, 2004. Proceedings. 20th IEEE International Conference on**. [S.l.: s.n.], 2004. p. 284–293. ISSN 1063-6773.

KAGDI, H.; POSHYVANYK, D. Who can help me with this change request? In: **Program Comprehension, 2009. ICPC '09. IEEE 17th International Conference on**. [S.l.: s.n.], 2009. p. 273–277. ISSN 1092-8138.

MCCANDLESS, M.; HATCHER, E.; GOSPODNETIC, O. **Lucene in Action, Second Edition: Covers Apache Lucene 3.0**. Greenwich, CT, USA: Manning Publications Co., 2010. ISBN 1933988177, 9781933988177.

MCDONALD, D. W.; ACKERMAN, M. S. Expertise recommender: a flexible recommendation system and architecture. In: **Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work**. New York, NY, USA: ACM, 2000. (CSCW'00), p. 231–240. ISBN 1-58113-222-0. Disponível em: `http://doi.acm.org/10.1145/358916.358994`.

MINTO, S.; MURPHY, G. Recommending emergent teams. In: **Mining Software Repositories, 2007. ICSE Workshops MSR '07. Fourth International Workshop on**. [S.l.: s.n.], 2007. p. 5–5.

MOCKUS, A.; HERBSLEB, J. D. Expertise browser: A quantitative approach to identifying expertise. In: **Proceedings of the 24th International Conference on Software Engineering**. New York, NY, USA: ACM, 2002. (ICSE '02), p. 503–512. ISBN 1-58113-472-X. Disponível em: `http://doi.acm.org/10.1145/581339.581401`.

MORAES, A.; SILVA, E.; TRINDADE, C. da; BARBOSA, Y.; MEIRA, S. Recommending experts using communication history. In: **Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering**. New York, NY, USA: ACM, 2010. (RSSE '10), p. 41–45. ISBN 978-1-60558-974-9. Disponível em: `http://doi.acm.org/10.1145/1808920.1808929`.

SETHANANDHA, B. D.; MASSEY, B.; JONES, W. Managing open source contributions for software project sustainability. In: **Proceedings of the 2010 Portland International Conference on Management of Engineering & Technology PICMET 2010)**.

Bangkok, Thailand: [s.n.], 2010. Disponível em: http://www.cs.pdx.edu/~bart/papers/picmet-patch.pdf.

SHAMI, N. S.; EHRLICH, K.; MILLEN, D. R. Pick me!: Link selection in expertise search results. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2008. (CHI '08), p. 1089–1092. ISBN 978-1-60558-011-1. Disponível em: http://doi.acm.org/10.1145/1357054.1357223.

TANG, J.; ZHANG, J.; ZHANG, D.; YAO, L.; ZHU, C.; LI, J. Arnetminer: An expertise oriented search system for web community. In: **Proceedings of the Semantic Web Challenge 2007 co-located with ISWC 2007 + ASWC 2007, Busan, Korea, November 13th, 2007**. [s.n.], 2007. Disponível em: http://ceur-ws.org/Vol-295/paper01.pdf.

WHITE, S.; SMYTH, P. Algorithms for estimating relative importance in networks. In: **Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining**. New York, NY, USA: ACM, 2003. (KDD '03), p. 266–275. ISBN 1-58113-737-0. Disponível em: http://doi.acm.org/10.1145/956750.956782.